

Gráfelméleti algoritmusok

London András

Pluhár András

A tananyag az EFOP-3.5.1-16-2017-00004 pályázat támogatásával készült.

Tartalomjegyzék

Előszó	vii
1. Bevezetés	1
1.1. Gráfelméleti alapfogalmak	3
1.2. Fák	5
1.3. Gráfrepresentációk	5
2. Gráfbejárások és alkalmazásaik	11
2.1. Néhány egyszerű alkalmazás	12
2.2. Robbins tétele	13
2.3. Földrajzi játék, Slither	14
2.4. Train sets	15
2.5. 2-SAT	16
2.6. Expander feltétel és hosszú utak	17
2.7. Laplace tétel és alkalmazásai	18
3. Legrövidebb utak és alkalmazásaik	21
3.1. Bellman algoritmusa	21
3.2. Topologikus rendezés	24
3.3. Kritikus út (ütemezés)	25
3.4. Maximális valószínűségű út	27
3.5. Fázisterek	28
4. Totálisan unimoduláris mátrixok	31
4.1. TU mátrixok	31
4.2. TU mátrixok karakterizációi	34

5. Hálózati problémák	35
5.1. Transshipment probléma	35
5.2. Az étkeztető (caterer) probléma	37
5.3. Legrövidebb út probléma	39
5.4. Szállítási (transportation) feladat	40
5.5. Hozzárendelési (assignment) probléma	40
5.6. Maximális folyam	42
5.7. A külszíni fejtés probléma	47
6. Intervallumgráfok és rokonaik	51
6.1. Intervallumgráfok	51
6.2. Metszetgráfok	53
6.3. Háromszögezett gráfok	55
6.4. Összehasonlítási gráfok	63
6.5. Perfekt gráfok	66
7. Exponenciális algoritmusok	71
7.1. Hálózati megbízhatóság	71
7.2. Approximáció gráfproblémákra	72
7.3. A maximális vágás probléma	76
8. Mohó algoritmusok	81
8.1. Matroidok	81
8.2. Stabil párosítások	90
8.3. Shannon kapacitás	96
9. Síkgráfok	101
9.1. Síkgráfok és lerajzolásaik	101
9.2. A $\Delta - Y$ transzformáció	103
9.3. A discharging módszer	104
10. A gráfalapú adatbányászat alapjai	109
10.1. Vizualizáció	110
10.2. Kisvilág gráfok	111
10.3. Gráfok klaszterezése	112

10.4. Információ terjedés gráfokon	117
10.5. Gráfok R^d -beli reprezentációi	119

Előszó

A jegyzet a Szegedi Tudományegyetemen előadott hasonló című kurzuson alapul illetve annak segédanyagának szántuk. A *Gráfelméleti algoritmusok* MSc kurzus, amely előtt diszkrét matematika, gráfelmélet, operációkutatás, sztochasztika, stb. kurzusok hangzanak el. Ezek részletes ismeretét nem feltételezzük, ugyanakkor a felmerülő problémákat csak vázlatosan kezeljük. Az anyagot viszont kibővítettük, hogy az azonos című PhD kurzushoz is használható legyen.

A gráfelmélet az alkalmazásokból született és az alkalmazások jelölték ki a fejlődésének az irányait. Ez hol különböző gráfparaméterek kiszámítását, hol szerkezeti problémák megoldását célozta. Változott a konkrét input gráfok köre is, síkgráfok, perfektn gráfok, véletlen vagy éppen kisvilág gráfok kaptak kiemelt figyelmet.

A terület megítélése is sokat változott a kezdetek óta. Euler még a matematika részének sem tekintette, Whitehead is csak alig, mondván "scum of topology." Az eltelt években viszont a gráfok összefonódtak az optimalizálás, geometria sőt a számelmélet alapvető kérdéseivel, így a szerepük még a tiszta matematikában is vitathatatlan.

Mi alapján a különböző modellekben felhasználható algoritmusokra és struktúrákra koncentrálnunk, nem szükségképpen a klasszikus elméletre.

Mind történeti mind algoritmikus szempontból az első helyen a gráfok különböző összefüggősége, bejárásai és a feszítőfák fogalmai állnak. Ezek segítségével meglepően sok probléma oldható meg, melyek egymással látszólag kevés rokonságban állnak.

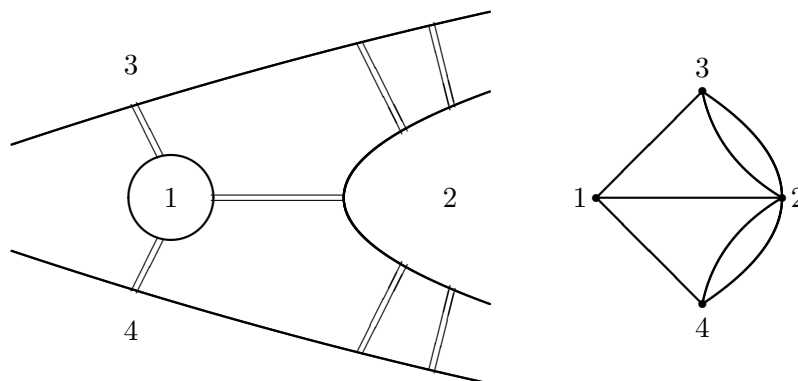
A hálózati folyamatok és a lineáris programozás kapcsolata az egyik első rendező elv, amely sok fontos eredmény között teremt kapcsolatot. Hasonló szerepet játszik a matroidok és a perfektn gráfok fogalmai, melyet szintén érintünk. Kitérünk még néhány a véletlenekkel, síkgráfokkal, adatbányászattal stb. kapcsolatos problémára.

A jegyzet elkészítéséhez elsősorban saját előadás és gyakorlat jegyzeteinket, továbbá a jegyzet végén hivatkozott szakirodalmat használtuk fel.

1. fejezet

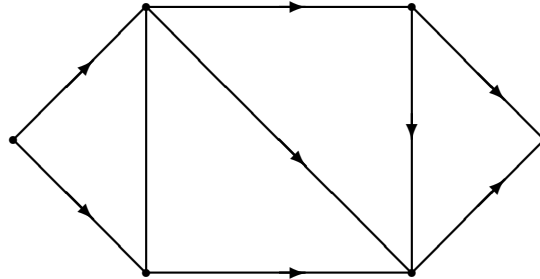
Bevezetés

A formális definíciók előtt lássuk, mik motiválják a bevezetendő fogalmainkat. A XVIII. században vetődött fel az ún. *Königsbergi probléma*, melyet Euler 1736-ban általánosan is megoldott. Königsberg városát a Pregel folyó osztotta részekre, melyet az ábrán látható módon kötöttek össze hidak. Felmerült a kérdés, lehet-e olyan sétát tenni, amelyben minden hídon pontosan egyszer haladunk át.

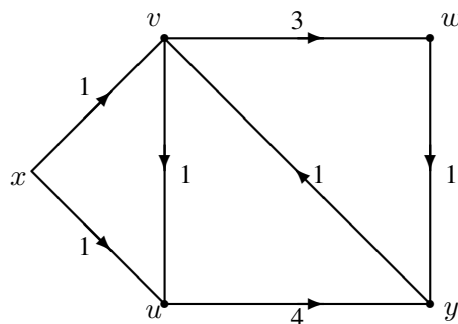


A probléma nyilvánvalóan ekvivalens azzal, hogy a jobboldalon álló objektum lerajzolható-e egyetlen vonallal (azaz a ceruza felemelése nélkül). Az ilyen objektumokat, melyek pontokból és az őket összekötő vonalakból, *élekből* állnak nevezzük majd *gráfnak*. Könnyen látható különben, hogy a probléma megoldhatatlan, mert három olyan pont van (1, 3 és 4), amelyhez páratlan számú él csatlakozik. Ezt általában is igen hasznos elnevezni; egy x ponthoz csatlakozó élek száma az x

pont fokszáma. Az élekre irányítást tehetünk, pl. ha olyan úthálózatot modellezünk, ahol egyirányú utak is előfordulnak.



Ezt a felfogást továbbvívve (azaz úthálózatnak tekintve a gráfot) felmerül a pontok közti távolság kérdése, illetve egy x és y pont közötti *legrövidebb út* megkeresése. Előfordul, hogy a gráfok éleire számokat írunk. Ezzel jelölhetjük az adott él *hosszát*, esetleg áteresztő képességét *kapacitását* vagy éppen *költségét*.



A fenti ábrán, ha a számokat távolságként interpretáljuk, x és y közt a legrövidebb út az (x, u, w, y) , hossza 3. Ha kapacitásként, akkor az x -ből y -ba 2 egységnyi anyag szállítható maximálisan. Jelenthetnek a gráf pontjai egy adott helyzetben alternatívákat is, ahol az „ x jobb, mint y ”-t egy $y \rightarrow x$ irányított éllel jelölünk. Használhatjuk a gráfokat térképszínezésre, raktározási problémák vizsgálatára (lásd *kromatikus szám*), vagy munkafolyamatok szétosztására (lásd *párosítás*), és ezen felül ezer más probléma modellezésére. Lássuk hát a példáink által sugallt fogalmak definícióját!

1.1. Gráfelméleti alapfogalmak

Egy G gráf egy $G = (V(G), E(G), I(G))$ hármast jelent, ahol $V(G)$ -t a gráf *pontjainak*, $E(G)$ -t pedig a gráf *éleinek* nevezzük. $I(G)$ egy függvény, mely G éleihez G pontjainak egy, vagy két elemű rendezett vagy rendezetlen halmazait rendeli. Szemléletesen a G gráf egyes pontjait élek (*irányított* vagy *irányítatlan*) kötik össze. Megengedünk továbbá *többszörös éleket* és *hurkokat* is. Az utóbbin természetesen olyan $e \in I(G)$ élt értünk, amelyhez rendelt halmaznak egy eleme van.

Egy hurokél mentes G gráf pontjának *fokszámán* azon élek számát értjük, amelyekhez $I(G)$ a v -t tartalmazó halmazt rendeli, a jele $d(v)$; másképp mondva $d(v)$ a v csúcsra *illeszkedő* élek száma. Tetszőleges G gráf esetében a hurokéleket duplán számítjuk be a fokszámba. Irányított gráfok esetén megkülönböztethetjük a *befutó* és a *kifutó* éleket, ezek számát $d_-(v)$ és $d_+(v)$ jelöli. Természetesen $d_-(v) + d_+(v) = d(v)$.

Az $I(G)$ által az $e \in E(G)$ élhez rendelt x és y pontot (ha kételemű halmazról van szó) az e él *összeköti*, vagy x és y *illeszkedik* az e élre. Továbbá k hosszú *sétának* nevezzünk egy $(x_1, e_1, \dots, e_k, x_{k+1})$ sorozatot, ha $x_i \in V(G)$, $i = 1, \dots, k+1$, valamint az $e_i \in E(G)$ összeköti x_i -t és x_{i+1} -et, ha $i = 1, \dots, k$. Az előbbi séta *irányított*, ha az e_i -hez x_i és x_{i+1} az (x_i, x_{i+1}) irányítással van hozzárendelve. Ha a sétában $x_1, \dots, x_{k+1}$ különbözőek, akkor *útról*, ha $x_1, \dots, x_k$ különbözőek és $x_1 = x_{k+1}$, akkor *körről* beszélünk. (Irányított séta esetén *irányított útról*, illetve *irányított körrel*.)

Beszélhetünk *súlyozott* gráfokról. Egy *pontsúlyozás* alatt egy $w : V(G) \rightarrow \mathbb{R}$, míg *élsúlyozás* alatt egy $w : E(G) \rightarrow \mathbb{R}$ függvényt értünk. Egy S halmaz *súlya*, $w(S) := \sum_{x \in S} w(x)$. Az élsúlyozott G gráfban egy $(x_1, e_1, \dots, e_k, x_{k+1})$ út hossza $\sum_{i=1}^k w(e_i)$. (A w jelölés helyett többnyire ℓ -et használunk, ha távolságot vagy valamilyen alsó korlátot, c -t ha költséget, és u -t ha egy kapacitás felső korlátját kívánjuk jelölni.)

Azt mondjuk, hogy H a G gráf *részgráfja* ha $V(H) \subseteq V(G)$ és $E(H) \subseteq E(G) \cap \binom{V(H)}{2}$.

Egy G gráf *összefüggő*, ha bármely két pontja között van út. Jelöljük $x \sim y$ -nal ha x és y pontok között van út. Könnyen belátható, hogy $\sim$ ekvivalenciareláció. A $\sim$ által meghatározott ekvivalencia osztályokat a gráf *komponenseinek* nevezzük. Ha bármely két pont között irányított út is van akkor a gráf *erősen összefüggő*. Ha $x \rightleftharpoons y$ -nal jelöljük két pont között az oda-vissza (irányított) elérhetőséget, akkor $\rightleftharpoons$ ekvivalenciareláció, és az általa meghatározott ekvivalencia osztályokat a gráf *erősen összefüggő komponenseinek* nevezzük.

1.1.1. További alapvető fogalmak

A G gráf pontjainak egy S halmaza *független*, ha S elemeit nem köti össze él. Egy független S halmaz *mag*, ha minden $x \in V(G) \setminus S$ estén van olyan $y \in S$, melyre $(y, x) \in E(G)$.

Egy G gráf pontjainak K részhalmaza *klika*, ha K bármely két pontja összekötött.

Egy $M \subseteq E(G)$ halmazt *párosításnak* nevezünk, ha bármely $x \in V(G)$ M -nek legfeljebb egy elemére illeszkedik. M *maximális*, ha nem bővíthető, *s teljes*, ha minden $x \in V(G)$ pontra van olyan $e \in M$, hogy x illeszkedik az e élre.

Egy G gráf *kromatikus száma*, $\chi(G)$, azon színek számának minimuma, melyekkel kifesthető a gráf ponthalmaza úgy, hogy az egymással összekötött pontok különböző színűek.

1.1. Feladat. Bizonyítsuk be, hogy egy G gráf esetén

1.
$$\sum_{v \in V(G)} d(v_i) = 2|E|, \text{ ahol } |E| \text{ az élek száma;}$$

2. a páratlan fokú csúcsok száma páros.

1.2. Feladat. Egy utat Euler-útnak nevezünk, ha a gráf minden élén pontosan egyszer halad át. A zárt Euler-utat Euler-körnek hívjuk. Bizonyítsuk be, hogy

1. Egy gráfban van Euler-út akkor és csak akkor, ha összefüggő és legfeljebb két páratlan fokú pontja van.

2. Egy gráfban van Euler-út akkor és csak akkor, ha összefüggő és minden foksám páros.¹

1.3. Feladat. Tegyük fel, hogy egy gráfban minden foksám 4-gyel osztható. Két játékos, legyenek Építő és Romboló, felváltva vesz el (töröl vagy foglal el) éleket magának a gráfból. Mutassuk meg, hogy az Építő minden csúcsnál a foksám legalább 1/4-ét meg tudja szerezni.

1.4. Feladat. Tegyük fel, hogy piros és kék élek vannak egy G gráfban, minden pontban megegyezik a számuk. Ekkor van olyan Euler kör (összefüggőség esetén) amelyben a kék és piros élek váltják egymást.

¹Ebből rögtön kapjuk a megoldást a Königsbergi problémára is.

1.2. Fák

Egy G gráf fa ha összefüggő, de bármely e élét elhagyva a maradék $G \setminus e$ gráf már nem összefüggő.

1.1. Tétel. Egy n pontú G gráfra az alábbi tulajdonságok ekvivalensek:

- (i) G fa (azaz összefüggő és körmentes).
- (ii) G összefüggő, és bármely élt eltávolítva G -ből a maradék gráf már nem összefüggő.
- (iii) G körmentes, de egy tetszőleges éllel bővítve az $E(G)$ élhalmazt, már keletkezik kör.
- (iv) G körmentes, és $n - 1$ éle van.
- (v) G összefüggő, és $n - 1$ éle van.

1.5. Feladat. Bizonyítsuk be az 1.1 tételt.

1.6. Feladat. Mutassuk meg, hogy egy fában minden x, y pontpár esetén pontosan egy $x - y$ út van.

Egy fa egyfokú pontjait a fa *leveleinek* nevezzük. A későbbiek szempontjából fontos lesz a következő lemma.

1.1. Lemma. Minden T fára, ha $|V(T)| \geq 2$, akkor van legalább két elsőfokú pontja.

A fákra minimális összefüggő gráfokként gondolva a következő fogalom bevezetése kézenfekvő. Egy T részgráf G -ben *feszítőfa*, ha T fa és $V(T) = V(G)$.

Ha egy T fa esetén megadunk egy $r \in V(T)$ kitüntetett csúcsot akkor *gyökeres fának* nevezzük. Ez az r csúcs egyértelműen meghatározza a T fa következő irányítását: minden (x, y) él kezdőpontja legyen az r -hez közelebbi csúcs x és y közül.

1.3. Gráfrepresentációk

A gráfokat lerajzolásukon² túl különböző módokon reprezentálhatjuk, illetve tárolhatjuk a számítógépen. Fontos szempont a tárolásnál, hogy az adott adatszerkezet hogyan viselkedik a gráfelméleti

²Fontos megjegyezni, hogy egy „ügyes” lerajzolás nagyon sokat tud segíteni gráfok vizsgálatánál. Erre látni fogunk példákat a gráfalapú adatbányászati módszerek tárgyalásánál.

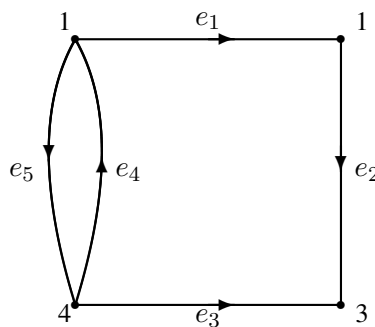
algoritmusok komplexitása szempontjából. Az alábbiakban néhány lehetséges reprezentációt mutatunk be, úgy mint a szomszédsági és pont-él illeszkedési mátrix, az éllista és szomszédsági lista. Csupán említés szintjén térünk ki az egyes reprezentációk használatának előnyeire és hátrányaira. Végül definiáljuk egy gráf Laplace-mátrixát és kimondjuk Kirchoff tételét egy gráf feszítőfáinak számáról.

1.3.1. Egyszerű reprezentációk

A gráfokat nagyon elegánsan reprezentálhatjuk a lerajzolásukon kívül különböző mátrixok segítségével is. Számunkra a legfontosabbak az *incidencia* vagy *illeszkedési mátrix* és az *adjacencia* vagy *szomszédsági mátrix*. Egy irányított gráf incidencia mátrixa a következőképpen konstruálható: G minden $x \in V(G)$ pontjához tartozik a mátrix egy sora, és minden $e \in E(G)$ éléhez egy oszlopa. Az x -hez tartozó sor és e -hez tartozó oszlop kereszteződésébe $+1$ kerül, ha e az x -be befutó, -1 , ha e az x -ből kifutó él, míg 0 , ha e nem illeszkedik x -re. Az egyszerűség kedvéért az adjacencia mátrixot irányítatlan, többszörös éleket és hurokéleket nem tartalmazó gráfra definiáljuk. Ez egy négyzetes mátrix $|V(G)|$ sorral (és oszloppal), ahol a sorokat és oszlopokat egy rögzített sorrendben rendeljük a gráf pontjaihoz. Egy sor és oszlop találkozásában 1 van, ha a megfelelő pontok között van él, 0 különben.³

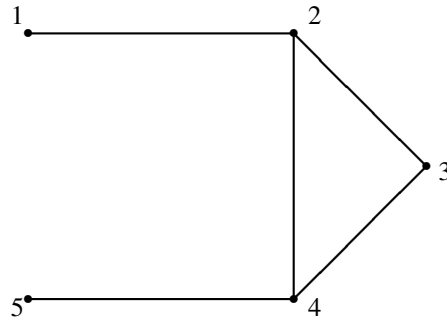
Példák:

	e_1	e_2	e_3	e_4	e_5
1	-1	0	0	1	-1
2	1	-1	0	0	0
3	0	1	1	0	0
4	0	0	-1	-1	1



³Látható, hogy irányítatlan gráfok szomszédsági mátrixa szimmetrikus. Megjegyezzük továbbá, hogy irányított gráfok szomszédsági mátrixának egy lehetséges megadása, ha a mátrix adott sor/oszlop eleme 1 akkor és csak akkor, ha a sort reprezentáló csúcsból vezet irányított él az oszlopot reprezentáló pontba.

	1	2	3	4	5
1	0	1	0	0	0
2	1	0	1	1	0
3	0	1	0	1	0
4	0	1	1	0	1
5	0	0	0	1	0

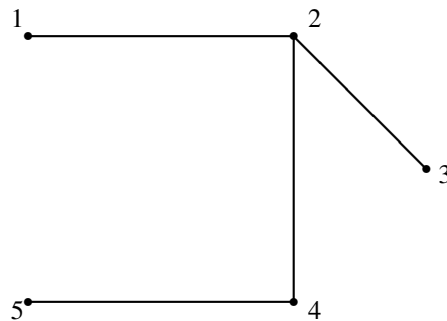


Tárolás szempontjából fontos fogalom a gráf, illetve szomszédsági mátrix *sűrűsége*, ami irányítatlan esetben $\rho = 2|E|/|V|^2$, irányított gráf esetén pedig $\rho = |E|/|V|^2$. Tekintsünk egy n pontú fát, melynek $|E| = n - 1$ éle van. Ha ezt a szomszédsági mátrixával reprezentálnánk, akkor n^2 bitet kellene letárolnunk, amit egy fa esetében túlzásnak érezhetünk. A következő, kompaktabb reprezentációk megadása természetesnek tűnik.

Egy G gráfot leíró *szomszédsági lista* a gráf minden $x \in V(G)$ pontjához felsorolja, hogy kik x szomszédai. Egy $|E|$ élet tartalmazó G (irányítatlan) gráf éllistája pedig egy egyszerű lista, mely azokat a pontpárokat tartalmazza soraiban, melyek közt van él.⁴

Példa:

Csúcs	Szomszédok
1	2
2	3, 4
3	2
4	2, 5
5	4



Éllista: $\{(1, 2), (2, 3), (2, 4), (4, 5)\}$

⁴Irányított esetben egy sor első eleme lehet az él kezdő-, második eleme pedig az él végpontja. Súlyozott gráfok is egyszerűen megadható így egy olyan lista segítségével mely hármassokat tartalmaz.

Az egyes reprezentációk és gráfelméleti algoritmusok hatékony implementációjának kapcsolata nem térünk itt ki, de az alábbi táblázatokban röviden összefoglaljuk az egyes reprezentációk tárigényét (ha a gráfnak n pontja és m éle van), illetve néhány előnyüket és hátrányukat.

Tárigény összehasonlítás

Reprezentáció	Tárigény	# Nem nullák	Sűrűség	Megjegyzés
Incidencia mátrix	$O(mn)$	$O(2m)$	$2/n$	Nem hatékony
Adjacencia mátrix	$O(n^2)$	$O(m)$	m/n^2	Hatékony sűrű gráfok esetén
Szomszédsági lista	$O(m + n)$	0	1	Hatékony ritka gráfok esetén
Éllista	$O(m)$	0	1	Hatékony ritka gráfok esetén

Előnyök és hátrányok

Reprezentáció	Előnyök	Hátrányok
Incidencia mátrix	Szép elméleti tulajdonságok	Gyakorlatban nem hatékony
Adjacencia mátrix	Élkeresés $O(1)$ időben	Szomszédok listája $O(n)$ időben
Szomszédsági lista	Szomszédok listája $O(d(v))$ időben	élkeresés $O(d(v))$ időben
Éllista	Egyszerű, kis tárigény	élkeresés $O(m)$ időben

1.7. Feladat. Határozzuk meg egy gráf háromszögeinek számát a szomszédsági mátrixának segítségével.

1.8. Feladat. Határozzuk meg egy gráf bármely két pontpárja közti k kosszú utak számát a szomszédsági mátrixának segítségével.

1.3.2. Gráf Laplace-mátrixa és a Kirchoff-tétel

Végül a gráfok egy rendkívül hasznos reprezentációját mutatjuk be, melynek motivációja elsősorban az *algebrai gráfelméletből* és a *gráfok spektrál elméletéből* ered. Ezeket itt nem tárgyaljuk, csak egy szép és fontos eredményt, Kirchoff tételét ismertetjük.

Legyen G egy egyszerű (azaz hurokél, párhuzamos él mentes és összefüggő) n csúcsú gráf, melynek szomszédsági mátrixa A . Ekkor G Laplace-mátrixa $L = D - A$, ahol D egy diagonális mátrix mely a csúcsok fokszámát tartalmazza; vagyis $D_{ii} = d(v_i) = \sum_{j=1}^n A_{ij}$ és $D_{ij} = 0$, ha

$i \neq j$.⁵ Néhány fontos tulajdonságát emeljük ki a Laplace-mátrixnak, bizonyítás nélkül.

1.1. Állítás. Legyenek $\lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_{n-1}$ az L mátrix sajátértékei. A következők teljesülnek:

- (i) L szimmetrikus,
- (ii) L pozitív szemidefinit (azaz $\lambda_i \geq 0, \forall i$),
- (iii) L minden sor- és oszlopösszege 0,
- (iv) $\lambda_0 = 0$, mivel $v_0 = (1, 1, \dots, 1)$ esetén $Lv_0 = 0$ teljesül. Ebből az is következik, hogy L szinguláris;
- (v) L $\lambda_0 = 0$ sajátértékének algebrai multiplicitása a G gráf összefüggő komponenseinek számával egyenlő.

Végül kimondjuk Kirchoff tételét egy gráf feszítőfáinak számáról.

1.2. Tétel (Kirchoff). Legyen G egy n pontú összefüggő gráf és $\lambda_1, \lambda_2, \dots, \lambda_{n-1}$ a Laplace-mátrixának nem nulla sajátértékei. Ekkor G feszítőfáinak száma

$$t(G) = \frac{1}{n} \lambda_1 \lambda_2 \dots \lambda_{n-1}.$$

Megjegyzés: Láttuk, hogy L mátrix szinguláris, ugyanakkor bármely j -re ($j = 1, 2, \dots, n$) L j -edik sorát és oszlopát elhagyva a kapott $\hat{L}$ $(n-1) \times (n-1)$ -es mátrix nonszinguláris, sőt, $\det(\hat{L}) = \lambda_1 \lambda_2 \dots \lambda_{n-1}$. Ez pedig azt jelenti, hogy a feszítőfák száma polinom időben kiszámítható. Megjegyezzük továbbá, hogy a tétel a Cayley-tétel általánosítása, mely szerint az n pontú teljes gráf feszítőfáinak száma n^{n-2} .

1.9. Feladat. Egy $n \times n$ mátrix minden oszlopa különböző. Mutassuk meg, hogy a mátrixból elhagyható egy sora úgy, hogy ez a tulajdonság megmaradjon.

1.10. Feladat. Bizonyítsuk be az 1.1 Állítás pontjait.

1.11. Feladat. Kirchoff tételét használva bizonyítsuk be Cayley-tételét, miszerint az n pontú teljes gráf feszítőfáinak száma n^{n-2} . Mutassuk meg ezt a Prüfer kód segítségével is.

⁵A Laplace-mátrix természetesen definiálható irányított, illetve súlyozott gráfok esetén is, itt ezekre az esetekre külön nem térünk ki.

2. fejezet

Gráfbejárások és alkalmazásaik

Legyen adott egy $G = (V, E)$ gráf (irányítatlan vagy irányított, továbbá legyen $|V| = n$ és $|E| = m$) és egy kijelölt pontja (amit *gyökérnek* nevezünk) $r \in V(G)$. Arra vagyunk kíváncsiak, hogy minden pont elérhető-e r -ből a gráf éleit használó úton. Szeretnénk megadni az r -ből elérhető pontokat, továbbá megadni az útvonalat is. Könnyen meggondolhatjuk, hogy ezen utak halmaza egy r gyökerű fa.

A fejezetben két alapvető algoritmus, a szélességi- és mélységi bejárás néhány alkalmazását ismertetjük. Ezek alapos ismeretét korábbi kurzusok alapján feltételezzük.

Szélességi- és mélységi bejárás

Az általános bejárési algoritmus a kijelölt r pontból indul, ez lesz egy fa gyökere és bekerül egy L listába. Az algoritmus minden iterációban kiválaszt egy pontot L -ből, kiválasztja egy gráfbeli szomszédját (irányított esetben egy kimenő élen vett szomszédot), ha ilyen létezik, és beteszi L -be. Ez addig folytatódik, amíg minden pont vizsgálatra nem kerül. A szomszéd kiválasztásának mechanizmusa teszi a különbséget az egyes bejárési algoritmusok között. Itt feltételezzük a két alapvető módszer, azaz a *szélességi bejárás* (Breadth First Search) amely az ún. *first-in-first-out* (FIFO) stratégiát alkalmazza a szomszéd kiválasztására, illetve a *mélységi bejárás* (Depth First Search) ami az ún. *last-in-first-out* (LIFO) stratégiával operál, ismeretét. Ezek részletes ismertetése helyett alkalmazásokat mutatunk be. Megjegyezzük, hogy mindkét algoritmus futási ideje $O(n + m)$.

2.1. Néhány egyszerű alkalmazás

Szélességi-, illetve mélységi bejárást alkalmazva egy gráf számos tulajdonsága egyszerűen tesztelhető, erre mutatunk néhány példát.

Egy n pontú G irányított gráf erősen összefüggősége $O(mn)$ időben megtehető. Ehhez indítsunk minden pontból egy bejárást (bármelyiket). Ha van olyan pont, ahonnan indulva a bejárás nem érinti a gráf minden pontját, akkor a gráf nem erősen összefüggő, különben igen. Sőt, ennél ügyesebbek is lehetünk és $O(n + m)$ algoritmust is megadhatunk. Ehhez indítsunk egy tetszőleges $v \in V(G)$ pontból egy bejárást és nézzük meg, bejártuk-e az összes pontot. Ha nem, akkor kész vagyunk, ha igen, akkor fordítsuk meg a gráf összes élének irányát, és indítsunk egy bejárást v -ből. Annak igazolását, hogyha a második bejárást is eléri az összes pontot, akkor G erősen összefüggő az olvasóra bízunk. Irányítatlan gráfokra az összefüggőség eldöntése még egyszerűbb, elég az előző tesztet elvégezni egyetlen, tetszőleges pontból indítva az algoritmust.

Robert Tarjan adott egy algoritmust (1972), mely egy irányított gráf erősen összefüggő komponenseit adja meg mélységi bejárással. Egy tetszőleges pontból mélységi bejárást indítunk. Amikor ez leáll (nincs olyan pont, aminek a ki-szomszédait nem jártuk még be), akkor a bejárt pontok egy erősen összefüggő komponenst alkotnak. Ezután egy tetszőleges nem bejárt pontot választva és abból újabb mélységi bejárást indítva folytatjuk az eljárást, egészen addig, amíg van nem bejárt pont.

Azt mondjuk, hogy egy gráf *körmentes* (vagy *aciklikus*), ha nem tartalmaz kört. A körmentesség ellenőrzése $O(n + m)$ időben ellenőrizhető. Ehhez indítsunk szélességi bejárást egy tetszőleges pontból. Ha az algoritmus során találunk olyan élt, ami egy már bejárt ponthoz kapcsolódik, akkor a gráf tartalmaz kört.

2.1. Feladat. *Mutassuk meg, hogy a szélességi keresés általában „gyorsabban” dönti el egy gráf körmentességét (azaz ha van kör, azt hamarabb találja meg), mint a mélységi keresés.*

Egy $G = (V, E)$ gráf *páros* (vagy *kétrészes*), ha $V(G)$ két diszjunkt halmazra bontható úgy, hogy minden él egyik végpontja az egyik halmazban, másik végpontja a másik halmazban van. Az, hogy egy gráf páros-e $O(n + m)$ időben ellenőrizhető. Ehhez fel kell használnunk az alábbi lemmát.

2.1. Lemma. *Egy gráf páros gráf akkor és csak akkor, ha nem tartalmaz páratlan hosszú kört.*

Ennek ismeretében, egy tetszőleges pontból mélységi bejárást indítva azt kapjuk, hogy két azonos szinten¹ lévő pont között van él, akkor a gráf tartalmaz páratlan kört, vagyis nem kétrészes. A lemma bizonyítását az olvasóra bízuk.

2.2. Feladat. Egy G gráf előjelezett, ha az élei ± 1 -el címkézettek. Az A, B egy stabil felbontása $V(G)$ -nek, ha A és B belsejében futó élek $+1$, a köztük levők pedig -1 -el címkézettek. Mutassuk meg, hogy akkor és csak akkor van stabil felbontás, ha minden körben páros sok -1 van.

2.3. Feladat. Vezessük vissza a 2.1 Lemmát a 2.2 Feladatra.

2.4. Feladat. Feladat: Legyen az $m \times n$ -es A mátrix egy G gráf él-csúcs illeszkedési mátrixa $GF(2)$ felett, $\vec{1}$ pedig a csupa egyesekből vektor. Mutassuk meg, hogy G páros akkor és csak akkor, ha az $Ax = \vec{1}$ megoldható $GF(2)$ -ben.

2.2. Robbins tétele

Érdekes kérdés, ha adott egy G irányítatlan, összefüggő gráf, akkor irányíthatók-e az élei úgy, hogy a kapott irányított gráf erősen összefüggő legyen. A kérdést gyakorlatiasan úgy is feltehetnénk, hogy egy város összes utcáját egyirányúvá tehetjük-e úgy, hogy a város bármely két „pontja” között megmaradjon az elérhetőség. A probléma vizsgálatához az összefüggőség fogalmát kell általánosítanunk.

Azt mondjuk, hogy egy gráf kétszeresen élösszefüggő, ha összefüggő, és bármely élét elhagyva továbbra is összefüggő marad. Egy $e \in E(G)$ élt elvágó élnek vagy hídnak nevezünk, ha $G \setminus e$ gráfnak több komponense van, mint G -nek. Azaz egy gráf kétszeresen élösszefüggő, ha nincs benne elvágó él. Az eredeti kérdésünkre a választ a következő tétel adja.

2.1. Tétel (Robbins). Egy G összefüggő gráf akkor és csak akkor irányítható erősen összefüggő gráffá, ha nincs benne elvágó él.

Bizonyítás. Ha van elvágó él, akkor azt bárhogyan irányítva a gráf nem lesz erősen összefüggő. Ha nincs benne elvágó él, akkor alkalmazzunk mélységi bejárást a gráf egy tetszőleges pontjából indítva. A bejárás által kapott feszítőfa éleit irányítsuk úgy, hogy a gyökérből indulva minden pont elérhető legyen. A gráf többi élét irányítsuk úgy, hogy a bejárás során később bejárt pontból a korábban bejárt pont felé mutasson. Ellenőrizhető, hogy így kapott irányítással a gráfunk erősen összefüggő lesz. $\square$

¹Vagyis a gyökértől vett távolságuk a bejárás során keletkező fában azonos.

2.3. Földrajzi játék, Slither

A normál földrajzi játékban² olyan földrajzi fogalmakat mondanak felváltva a játékosok, amelyek az előző szó utolsó betűjével kezdődnek. Kétszer nem használható egy szó, és aki nem tudja folytatni, az veszít. Ennek kézenfekvő absztrakt változatában egy G irányított gráf pontjain lépegetünk felváltva, adott v_0 pontból indulva, az élek mentén. Egy pont csak egyszer látogatható, és aki nem tud lépni, az veszít. Az általános eset nagyon nehéz, egy speciális játékra való visszavezetése megtalálható például P.A. Játékelmélet jegyzetben [12].

A földrajzi játék egy változata az ún. *általánosított Slither*. Itt az első játékos kezdőlépése a G egy v_0 pontjának a kiválasztása, majd a második folytatja a játékot v_0 -ból. Ennek a játéknak a speciális esetei is érdekesek, lásd [10] 8. fejezet 8. és 9. probléma. A következő tétel (iii) pontja egy újabb alkalmazását mutatja a gráfbejárásnak. A tétel többi pontjának bizonyítása megtalálható például P.A. Játékelmélet jegyzetében.

2.2. Tétel. *A következő állítások igazak.*

- (i) *Tegyük fel, hogy G körmentes. Ekkor az első játékos nyer, és megadhatók azok a pontok melyekkel kezdve nyerhet.*
- (ii) *Tegyük fel, hogy van olyan $v_0 \in V(G)$, melyre $d_-(v_0) = 0$, azaz v_0 -ba nem fut él. Ekkor az első játékosnak van nyerő stratégiája.*
- (iii) *Egy tetszőleges G irányított gráfra második játékos nyer akkor és csak akkor, ha G minden erősen összefüggő komponensében játszva is nyerne.*
- (iv) *Egy szimmetrikus G gráf³ esetén az első játékos pontosan akkor nyer, ha a G gráfban nincs teljes párosítás.⁴*

Bizonyítás. (iii) Használjuk a tényt, hogy G ponthalmaza felbomlik erősen összefüggő komponensek uniójára és a C_i, C_j ($i \neq j$) komponensek között az élek egyirányban húzódnak.⁵ Egy komponensben vesztes vagy az egész játszma vesztesét jelenti, vagy kényszerrel egy másik komponensbe elsőként történő belépésre. Ha van olyan komponens, ahonnan játszva az első játékos nyer,

²Nálunk inkább az „ország, város, fiú, lány” néven ismert.

³ G szimmetrikus, ha az $(x, y) \in E(G) \Rightarrow (y, x) \in E(G)$.

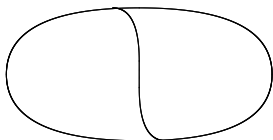
⁴Egy párosítás teljes, ha a gráf minden pontját lefedi.

⁵Az erősen összefüggő komponenseket egy pontra és a köztük húzódó éleket egy élre összehúzza keletkezik a kondenzált gráf, $C(G)$. Ez körmentes.

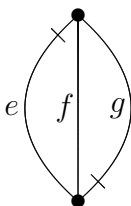
akkor vegye a $C(G)$ -ben egy levélhez (1 befokú, 0 kifokú pont) legközelebb eső ilyen C_i komponenst. Játssza ezen a C_i -hez tartozó nyerő stratégiát. A második játékos vagy veszít a C_i -ben, vagy átlép egy olyan C_j komponensbe, ahol már az második játékosnak van nyerő stratégiája; ezt viszont az első játékos ellopja tőle. Ha bármely C_i komponensben nyer a második játékos, akkor persze nyer G -ben, mindig az aktuális C_i -beli nyerő stratégiát követve. $\square$

2.4. Train sets

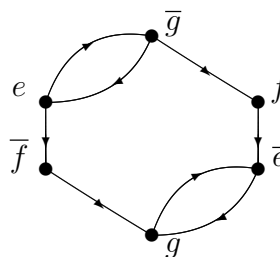
A *train set* vagy *vonathalmaz* fogalma a topológiából ered, de jól értelmezhető a terepasztalokon megvalósítható játékokra is. Két sín nulla fokos szögben találkozhat, és a vonat egyik irányból érkezve mindkettőre rátérhet, míg a másik irányból mindkét sínről a kitüntetettre vezet az út. Kérdés lehet, mely része járható be a rendszernek?



Egy train set.



Kitüntetett gráf.



Fázisgráf.

Az ábra baloldalán egy konkrét pálya látható. Középen ennek a pályának egy reprezentációja *kitüntetett (designated) gráf* formában. Ebben a sínek találkozása egy csomópont, amelyiknek az egyik élét a ponthoz közel megjelöljük egy kis vonallal. Mégpedig azt az élt, amelyen elhagyhatjuk a pontot, ha a két másiktól érkeztünk.⁶

Könnyen látható, hogy míg a kitüntetett gráf hasznos a probléma leírásában, nem támogatja megfelelően az algoritmusokat. Ehhez egy másik gráfot származtatunk, ez a *fázisgráf*, lásd az ábra jobboldala. A kitüntetett gráf éleit irányítjuk (jelen esetben mindet felfelé) és az adott élen haladó vonatról nem pusztán azt rögzítjük, hol van, hanem azt is, hogy *melyik irányba* megy. Ennek megfelelően a fázisgráf pontjai a kitüntetett gráf éleinek normál és ellentétes irányításainak felelnek meg (e ill. $\bar{e}$); ezek a fázisok. A fázisok közti lehetséges átmeneteknek felelnek meg a fázisgráf élei.

⁶Egy (x, y) élen lehet két jelölés is, ha az x -ben és y -ban is kitüntetett.

A fázisgráf $C_1, \dots, C_\ell$ erősen összefüggő komponensei felelnek meg azon állapotoknak, melyek egymásból kölcsönösen elérhetőek. Ha egy C_i komponenst egyszer elhagyunk, a benne lévő állapotok nem ismétlődnek. Ha a C_i komponensből nem vezet ki él, akkor a benne lévő állapotok ismétlődnek a továbbiakban.

2.5. Feladat. *Mutassuk meg, hogy egy n találkozóhelyből álló vonathalmaz kitüntetett gráfja egy n pontú, $3n/2$ élű 3-reguláris gráf. A fázisgráf $3n$ pontú, melyek közül n db 2-kifokú, 1-befokú, n db 1-kifokú, 2-befokú és n db 1-kifokú, 1-befokú.*

2.6. Feladat. *Mekkora a legkisebb vonathalmaz, amelynek minden szakasza minden irányban tetszőlegesen sokszor bejárható?*

2.7. Feladat. *Hogyan kellene megváltoztatnunk a fogalmainkat, ha beépíthetünk fordítókat, vagy a sínek "x" alakban is találkozhatnak?*

2.8. Feladat. *Tegyük fel, hogy a váltókban érmedobással döntünk a továbbhaladásról. Milyen Markov láncot rendelhetünk ehhez és milyen paramétereket számíthatunk ki?*

2.5. 2-SAT

A 2-SAT a Boole-féle kielégíthetőségi problémakör egy speciális esete. Egy olyan *Boole formula*⁷, kielégíthetőségi problémája, mely olyan *klózok* konjunkciója (logikai ÉS), ahol minden klóz két változó vagy negált változó diszjunkciója (logikai VAGY).

Példa:

$$(x_1 \vee x_5) \wedge (\neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_2) \wedge (x_4 \vee x_6)$$

A feladat az, hogy adjuk meg az $x_1, \dots, x_6$ változók egy olyan értékadását, melyre a formula az 1 (IGAZ) értéket számolja ki. Aspvall, Plass és Tarjan 1979-ben egy egyszerű megoldást adott a problémára. A formulához egy ún. *implikációs gráfot* definiáltak, ami egy irányított gráf, melynek pontjai a formulában szereplő változók, illetve a változók negáltja. Két pont között van irányított él

⁷Olyan formula, melyben minden változó 0 vagy 1 értéket vehet fel, melyeket az ÉS ($\vee$), VAGY ($\wedge$) és NEM ($\neg$) operátorok kötnek össze.

pontosan akkor, ha a két reprezentált változó egy klózban szerepel, az irányítás pedig a diszjunkció *implikatív formára* való átírása szerinti implikáció iránya. Például

$$(x_1 \vee \neg x_2) \equiv (\neg x_1 \Rightarrow \neg x_2) \equiv (x_2 \Rightarrow x_1)$$

ekvivalenciák a gráf $(\neg x_1, \neg x_2)$ és (x_2, x_1) irányított éleit adják. Az eredeti feladat megoldása a következő. Határozzuk meg az implikációs gráf erősen összefüggő komponenseit (pl. Tarjan algoritmusával). Ha van olyan komponens, ahol egy változó és negáltja egyszerre szerepel, akkor a formula kielégíthetetlen. Egyszerű megfontolással adódik, hogy egy erősen összefüggő komponensen belül minden változó értéke azonos kell, hogy legyen. Ezt az értéket például az implikációs gráf kondenzált gráfja segítségével adhatjuk meg (ennek megfontolását az olvasóra bizzuk).

2.9. Feladat. *Döntsük el, hogy a példában szereplő formula kielégíthető-e. Ha igen, adjunk is meg egy IGAZ értékadást.*

2.6. Expander feltétel és hosszú utak

Azt mondjuk, hogy a $G = (V, E)$ gráf (k, α) -*expander* ha minden $X \subset V$, $|X| \leq k$ esetén $N(X) \geq \alpha|X|$. A fogalom intuitívan azt mondja, hogy a pontok „nem túl nagy” részalmazainak szomszédsága „nagy”.

2.1. Állítás. *Legyen $G = (V, E)$ gráf olyan, hogy minden $|X| \subseteq V(G)$, $|X| = k$ esetén $N(X) \geq 2|X|$. Ekkor létezik legalább $2k$ hosszú út G -ben.*

Bizonyítás. Indítsunk tetszőleges pontból mélységi bejárást G -ben, ahol a már bejárt (lezárt) pontok halmaza legyen S , a nem látogatott pontok halmaza T , $U = V \setminus (S \cup T)$ pedig a „bejárási alatti/nem lezárt” pontok halmaza. Vegyük azt a pillanatot, amikor a már bejárt pontok száma éppen k . A mélységi bejárás algoritmus garantálja, hogy ennek a k pontnak nem lehet szomszédja egyetlen T -beli pont sem, vagyis $N(S) \subseteq U$. Viszont ekkor $|U| \geq 2k$, és azt is tudjuk, hogy U pontjai mindig egy utat határoznak meg G -ben. $\square$

2.2. Állítás. *Legyen $G = (V, E)$ egy n pontú összefüggő gráf olyan, hogy létezik él bármely $X, Y \subseteq V$ két k elemű diszjunkt pontalmazt között. Ekkor G -ben van legalább $n - 2k + 1$ hosszú út és $n - 4k + 4$ hosszú kör.*

Bizonyítás. Futtassuk a mélységi bejárás algoritmust tetszőleges pontból indulva. Tekintsük azt a pillanatot, amikor $|S| = |T|$, azaz a már bejárt (lezárt) pontok száma megegyezik a még nem látogatott pontok számával (gondoljuk meg, hogy ilyen pillanat van). Mivel a gráfban az aktuális S és T pontjai között nincs él (azaz diszjunktak), ezért a feltétel miatt a méretük legfeljebb $k - 1$. Ekkor tudjuk, hogy $U = V \setminus (S \cup T)$ bejárás alatt lévő pontok száma legalább $n - 2k + 2$. Mivel U pontjai egy utat határoznak meg, ezért a keresett út megvan. A körhöz vegyük ennek az útnak az első k és utolsó k pontját. Mivel a feltételünk miatt ezek között kell, hogy legyen él, ezért megvan a keresett kör is. $\square$

Megjegyzés. Egy gráfban hosszú utak létezésének igazolására más technikák is léteznek. Az egyik ilyen Pósa Lajos nevéhez fűződik 1976-ból, aki véletlen gráfokban való *Hamilton-körök*⁸ keresése során dolgozta ki módszerét.

2.7. Laplace tétel és alkalmazásai

Determinánsokat sokféleképpen kiszámíthatunk, a legritkábban kifejtési tételek segítségével. Különösen rossz választásnak tűnik az ún. *általánosított Laplace tétel* használata:

2.3. Tétel (általánosított Laplace kifejtési tétel). Egy $n \times n$ -es A mátrixra

$$\det(A) = \sum_{|I|=k, |J|=\ell} (-1)^{\sum I + \sum J} \det(A^{I,J}) \det(A^{\bar{I},\bar{J}}),$$

ahol $I, J \subset \{1, \dots, n\}$, $\bar{I} = \{1, \dots, n\} \setminus I$, $A^{I,J}$ mátrix az A mátrix I -vel indexelt soraiból és J -vel indexelt oszlopaiból áll, $\sum I = \sum_{i \in I} i$.

A formula művelet igénye $O(n!)$, ami magyarázza a ritka használatot. Speciális esetben viszont nagyon kézenfekvő. Az A mátrix *általánosított diagonális*, ha a főátlójában az $A_1, \dots, A_m$ négyzetes mátrixok állnak, a főátló alatt zérók, míg felette tetszőleges elemek.

2.10. Feladat. Mutassuk meg, hogy egy A , $n \times n$ -es általánosított diagonális mátrixra igaz, hogy $\det(A) = \prod_{i=1}^m \det(A_i)$.

2.11. Feladat. Írjunk fel egy 6×6 általánosított diagonális A mátrix, ahol A_1, A_2, A_3 rendre 2×2 mátrixok és a főátló felett csupa nem zéró elem áll. Számítsuk ki $\det(A)$ -t a 2.10 feladat formulájával.

⁸Hamilton-kör a gráf egy olyan köre, mely a gráf minden csúcsán áthalad.

2.12. Feladat. Tegyük fel, hogy az $n \times n$ -es A mátrix nem általánosított diagonális, de sok zéró eleme van. Hogyan rendezhetnénk át a sorait/oszlopait úgy, hogy a kapott mátrix a lehető legnagyobb mértékben általánosított diagonális legyen?

2.13. Feladat. Rendeljünk egy $n \times n$ -es A mátrixhoz egy irányított $\vec{G}_A$ gráfot úgy, hogy $ij \in E$ ha $a_{ij} \neq 0$. Minek felelnek meg a $\vec{G}_A$ gráf erősen összefüggő komponensei?

3. fejezet

Legrövidebb utak és alkalmazásai

Az egyik leggyakrabban felvetődő gráf optimalizálási probléma a legrövidebb utak keresése. Tegyük fel, hogy adott a G irányított gráf, amelynek éleihez súlyokat rendelünk. A hagyománynak megfelelően a (v, w) él súlyát (hosszát) $\ell(v, w)$ -vel jelöljük, míg egy p útét, ami nem más, mint a p -ben lévő élek súlyának összege, $\ell(p)$ -vel. Feltéhetjük azt is, hogy $(v, v) \in E(G)$ minden $v \in V(G)$ -re, és $\ell(v, v) = 0$. A következő két feladatot fogalmazzuk meg:

- (a) Rögzítsük most G két pontját, s -et és t -t és keressük meg azt a p utat, mely s -ből t -be vezet és $\ell(p)$ minimális.
- (b) Általánosabban kereshetjük a legkisebb súlyú (legrövidebb) utakat s és minden $v \neq s$ pont között.

Meglepő módon az összes olyan ismert algoritmus, ami megoldja (a)-t, (b)-t is megoldja egyben. Mielőtt rátérnénk a megoldásra, vizsgáljuk meg mikor létezik ez. Ha az s -ből elérhető egy C negatív súlyú kör, azaz $\ell(C) < 0$, és C -ből elérhető t , akkor az a feladatnak nem lehet megoldása. (C -n tetszőlegesen sokszor „körbemegetünk”, így az út súlya tetszőlegesen kicsi lehet.) Másrészt, ha nincs ilyen kör, akkor lehetséges megoldásaink halmazáról feltehetjük, hogy véges, így - amennyiben az nem üres - van optimum.

3.1. Bellman algoritmus

A (b) feladat megoldására Bellman 1956-ban publikált iteratív módszerét az ún. *dinamikus programozást* használjuk. Az eljárás leírásához szükségünk van néhány jelölésre. Egy $v \in V(G)$ pontra

jelentse $\ell(v)$ az s -ből v -be vezető legrövidebb út hosszát, míg $\ell_k(v)$ az s -ből v -be a *legfeljebb k élt tartalmazó* legrövidebb út hosszát. Amennyiben a fenti utak nem léteznek, $\ell(v)$ vagy $\ell_k(v)$ értéke végtelen (jelben ∞). Használjuk még a $p : V(G) \rightarrow V(G) \cup \{\emptyset\}$ függvényt a megfelelő utak nyilvántartására.

1. Kezdetben legyen $\bar{\ell}_0(s) := 0, p_0(s) := \emptyset, \bar{\ell}_0(v) := \infty, p_0(v) := \emptyset$, ha $v \neq s$.
2. A k -adik lépésben legyen $\bar{\ell}_k(v) := \min\{\bar{\ell}_{k-1}(w) + \ell(w, v) : (w, v) \in E(G)\}$, $p_{k-1}(v)$ értéke pedig változzon arra a w -re, amely a minimumot megvalósítja, ha $\bar{\ell}_k(v) < \bar{\ell}_{k-1}(v)$.

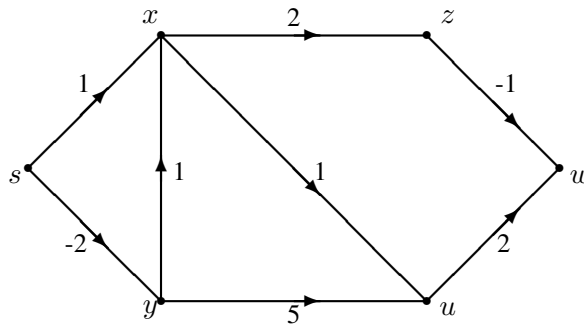
3.1. Tétel. *A fenti algoritmus helyesen számolja ki a ℓ_k függvényt, azaz $\bar{\ell}_k \equiv \ell_k$. Ha G -ben nincs negatív súlyú kör, akkor $\ell_{n-1}(v) = \ell(v)$ minden $v \in V(G)$ -re, ahol $n = |V(G)|$, a gráf pontjainak száma. Továbbá ebben az esetben egy legrövidebb $s - v$ út pontjai fordított sorrendben $v, p(v), p^2(v), \dots, p^i(v) = s$, ha $\ell(v) < \infty$, ahol $p = p_{n-1}$.*

Bizonyítás. Az első állítást k szerinti indukcióval láthatjuk be. A kezdőértékekre, azaz $k = 0$ -ra az állítás nyilvánvaló. Tegyük fel, hogy egy adott v -re a legrövidebb legfeljebb $k + 1$ élből álló $s - v$ út éppen $(s, \dots, w, v)$. Az indukciós feltevésből következik, hogy létezik $\bar{\ell}_k(w) = \ell_k(w)$ hosszúságú, legfeljebb k élből álló $s - w$ út, így $\ell_{k+1}(v) \geq \bar{\ell}_k(w) + \ell(w, v) \geq \bar{\ell}_{k+1}(v)$. Mivel $\ell_{k+1}(v) \leq \bar{\ell}_{k+1}(v)$, adódik az állítás.

Ha G -ben nincsenek negatív körök, akkor bármely séta átalakítható úttá úgy, hogy a hossza nem növekszik. Ez viszont azt jelenti, hogy $\ell_{n-1} \equiv \ell_{n-1+i}$ bármely $i \in \mathbb{N}$ esetén és így $\ell_{n-1} \equiv \ell$.

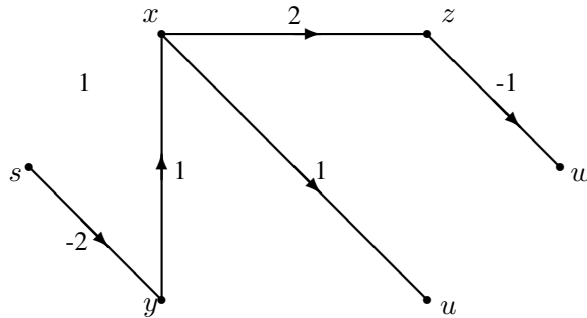
A p szerepét szintén a legrövidebb utak élszáma szerinti indukcióval láthatjuk be. (Ugyanis a $(p(v), p^2(v), \dots, p^i(v) = s)$ legrövidebb $s - p(v)$ út lesz a feltétel szerint. Ekkor a $(p(v), v)$ él hozzávételével legrövidebb $s - v$ utat kapunk.) $\square$

Példa: Legyen G az ábrán látható gráf; ekkor az algoritmus lépései az alábbi táblázatban foglalhatók össze:



	0		1		2		3		4		5	
	ℓ_0	p	ℓ_1	p	ℓ_2	p	ℓ_3	p	ℓ_4	p	ℓ_5	p
s	0	$\emptyset$	0	$\emptyset$	0	$\emptyset$	0	$\emptyset$	0	$\emptyset$	0	$\emptyset$
x	∞	$\emptyset$	1	s	-1	y	-1	y	-1	y	-1	y
y	∞	$\emptyset$	-2	s	-2	s	-2	s	-2	s	-2	s
z	∞	$\emptyset$	∞	$\emptyset$	3	x	1	x	1	x	1	x
u	∞	$\emptyset$	∞	$\emptyset$	2	x	0	x	0	x	0	x
w	∞	$\emptyset$	∞	$\emptyset$	∞	$\emptyset$	2	z	0	z	0	z

A táblázatot oszloponként töltjük ki, a $\ell_0(s) = 0$, $\ell_0(v) = \infty$ minden s -től különböző v -re, míg $p = \emptyset$. Ezek után a ℓ_k és p_k függvények értékeit rekurzióval kapjuk meg $k \geq 1$ -re. A táblázat utolsó oszlopából nemcsak a legrövidebb $s - v$ utak ($v \in G$) hosszai olvashatók ki, hanem meghatározhatók ezek az utak. Ha csak a legrövidebb utak által használt éleket ábrázoljuk, ezek a p függvény értékei, akkor az ún. *legrövidebb utak fáját* kapjuk. Ez, ha s -ből elérhető bármely pont, egy *gyökeres fa*, s gyökérponttal, ami azt jelenti, hogy $d_-(s) = 0$ és $d_-(v) = 1$, ha $s \neq v \in V(G)$.



Ebből könnyen kiolvashatók a legrövidebb $s - v$ utak és hosszai:

1. $s - x$ út: (s, y, x) , hossza $d(x) = -1$
2. $s - y$ út: (s, y) , hossza $d(y) = -2$
3. $s - u$ út: (s, y, x, u) , hossza $d(u) = 0$
4. $s - z$ út: (s, y, x, z) , hossza $d(z) = 1$
5. $s - w$ út: (s, y, x, z, w) hossza $d(w) = 0$

Megjegyzések:

1. Az eljárás alkalmazható abban az esetben is, ha G -ben van negatív kör. Ekkor egy C negatív kör létezését a $v = p^i(v)$ jelzi valamely $v \in V(G)$ -re és $i \in \mathbb{N}$ -re, továbbá a p segítségével meghatározható C . Szintén van mód annak eldöntésére, hogy egy adott v pont rajta van-e egy negatív sétán. Vegyünk hozzá a gráfhoz egy új q pontot, és a (q, v) , (v, q) éleket nulla súllyal. Ekkor világos, hogy $\ell_n(q) < 0$ akkor és csak akkor, ha v rajta van egy negatív sétán.
2. Egy irányítatlan G gráf összefüggősége is tesztelhető; vegyük fel az irányítatlan (i, j) helyére az irányított (i, j) , (j, i) éleket 1-1 súllyal. Ha minden $s - v$ legrövidebb út hossza véges, akkor G összefüggő, ha $\ell(v) = \infty$, akkor s és v különböző komponensekben vannak.
3. Az algoritmus működése azon múlt, minden $v \in V(G)$ -re van olyan $w \in V(G)$, hogy $\ell_k(v) = \ell_{k-1}(w) + \ell(w, v)$. Ez durván azt jelenti, hogy az optimális struktúránk között kapcsolat van, a „nagyobb” felépíthető a „kisebből”. Ilyen jellegű kapcsolat gyakran kihasználható egy hatékony algoritmus felépítésére a fentihez hasonló módon. Ezt a módszert *dinamikus programozásnak* nevezzük.

A legrövidebb utakkal meglepően sok probléma modellezhető, illetve fontos része más algoritmusoknak. Az utóbbit később látni fogjuk. Itt három, nem kézenfekvő alkalmazásra térünk ki.

3.2. Topologikus rendezés

Legyen G egy irányított gráf. A gráf *topologikus rendezése* a gráf csúcsainak egy lineáris rendezése, ahol minden (u, v) irányított él esetén u megelőzi v -t a rendezésben. Erre példa, amikor a gráf pontjai feladatokat reprezentálnak az élek pedig a feladatok elvégzési sorrendjéből adódó korlátozásokat mutatják. Ebben az esetben a topologikus rendezés a feladatok elvégzésének egy lehetséges sorrendjét adja meg. A következő feladat egy szükséges és elegendő feltételt mond a rendezés létezéséhez.

3.1. Feladat. *Egy irányított gráf csúcsainak akkor és csak akkor van topologikus rendezése, ha a gráf aciklikus (azaz irányított körmentes).*

Az egyik első algoritmust topologikus rendezésre Kahn adta 1962-ben. Az eljárás a következő:

1. Legyen L egy (kezdetben üres) lista, amely a rendezett csúcsokat tartalmazza, S pedig az összes olyan csúcs halmaza, melyek befoka 0.

2. Amíg S nem üres, vegyünk ki egy u elemét és tegyük L végére:

(i) Minden v -re, amelyre $e = (u, v)$ él, töröljük e -t a gráfból;

(ii) Ha v -nek nincs több bemenő éle, akkor tegyük S -be.

3. Ha a gráf az eljárás végén tartalmaz élt, akkor van legalább egy irányított köre, azaz nincs topologikus rendezése. Különben L lista a topologikus sorrendet adja meg.

Ha a gráf aciklikus, akkor egy (nem feltétel egyértelmű) megoldást az L lista adja. Ha a gráf tartalmaz irányított kört, akkor a topologikus rendezés nem lehetséges.

3.2. Feladat. *Igazoljuk, hogy Kahn algoritmusa helyesen határozza meg a topologikus sorrendet (ha az létezik).*

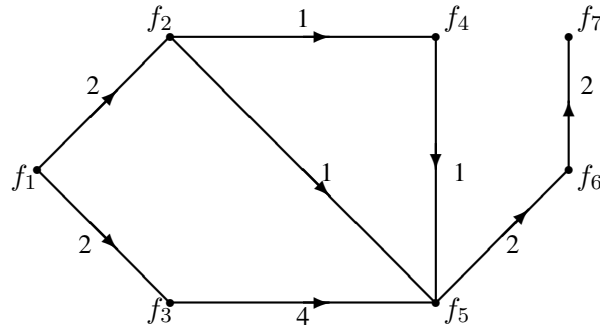
3.3. Feladat. *Adjunk meg algoritmust topologikus rendezésre, mely a mélységi keresés módszerét használja.*

3.4. Feladat. *Adjunk meg topologikus rendezésen alapuló gyors algoritmust, mely súlyozott, irányított aciklikus gráfban határozza meg az összes pont közti legrövidebb utakat.*

3.3. Kritikus út (ütemezés)

Tegyük fel, hogy adott egy projekt, amely több, egymástól nem független résztevékenységből áll. Tudjuk a résztevékenységek végrehajtásához szükséges időt és azt, hogy egy adott tevékenység elkezdésének mely más tevékenységek befejezettsége feltétele. (Pl. egy építkezésnél csak az alapozás után lehet falazni.) A célunk annak meghatározása, hogy minimálisan mennyi idő alatt teljesíthető a projekt, ha *nincs* egyéb korlátozás. A javasolt modellt és a megoldást egy példán illusztráljuk.

Példa: Jelöljük a részfeladatokat $f_1, \dots, f_6$ -tal, az elvégzéshez szükséges időket $t_1, \dots, t_6$ -tal és ha az f_j elkezdésének az f_i befejezettsége feltétele, azt $f_i < f_j$ -vel. Legyen $t_1 = 2, t_2 = 1, t_3 = 4, t_4 = 1, t_5 = 2, t_6 = 2$ és a feltételek: $f_1 < f_2, f_1 < f_3, f_2 < f_4, f_2 < f_5, f_3 < f_5, f_4 < f_5, f_5 < f_6$. ábrázoljuk ezt egy irányított gráf segítségével, ahol a f_i pontot rendeljük az f_i feladathoz, (f_i, f_j) él a gráfban, ha $f_i < f_j$, és ekkor $l(f_i, f_j) = t_i$. Vegyünk fel egy fiktív feladatot, f_7 -et, és legyen (f_i, f_7) él a gráfban, ha f_i nem feltétele más feladatnak.



Vegyünk egy irányított utat, az $(f_1, f_2, f_5, f_6, f_7)$ -et a gráfból. Ennek hossza $t_1 + t_2 + t_5 + t_6$ és mivel az irányítás szerint az f_1, f_2, f_5 és f_6 tevékenységek csak ebben a sorrendben és egymás után hajthatók végre, az út hossza alsó korlátja a szükséges minimális időnek. Mi több ez a minimális idő egybeesik egy ilyen típusú alsó korláttal, mert a modellünkben fellépő gráfok körmentesek. Egy G körmentes irányított, súlyozott gráf *leghosszabb útját* nevezzük *kritikus útnak*.

3.2. Tétel. *Egy projekt végrehajtásához szükséges minimális idő egyenlő a kritikus út hosszával.*

Bizonyítás. Teljes indukcióval a kritikus út élszáma szerint. □

A példánkban könnyen látható: a $(f_1, f_3, f_5, f_6, f_7)$ kritikus út, melynek hossza $t_1 + t_3 + t_5 + t_6 = 10$. Így a tételből következően a projekt időigénye pontosan tíz egység.

Egy tetszőleges irányított gráfban a leghosszabb út megtalálása rendkívül nehéz feladat. Esetünkben a helyzet jóval kedvezőbb: vegyük az élekhez rendelt súlyok -1 -szeresét, és keressük meg a *legrövidebb utat*. Mivel a gráfunk körmentes, így nincs negatív kör, azaz használható Bellman algoritmus és a kapott legrövidebb út a leghosszabb lesz az eredeti gráfban.

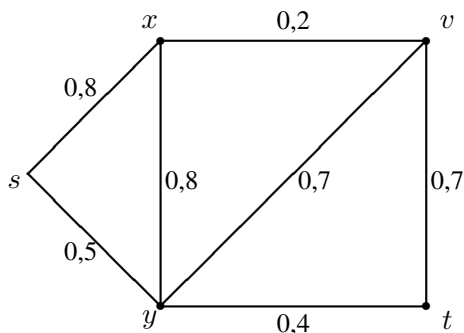
Megjegyzés: A módszer alkalmazásának egyik legszebb példája az Apolló program volt. Ebben ezernél több alvállalkozó által végzett, igen nagy számú részfeladatot sikerült összehangolni mígnem 1969-ben az ember a Holdra léphetett.

3.5. Feladat. *Egy munkásnak A, B, C, D, E, F, G, H munkákat kell végrehajtania. A munkák közötti ' $X < Y$ ' reláció azt jelenti, hogy X -et hamarabb kell elvégezni mint Y -t. Most az $A < H$, $B < H$, $D < E$, $D < F$, $E < C$, $E < G$, $F < C$ és $G < A$ feltételek adóttak. Adjunk meg egy lehetséges végrehajtási sorrendet. Mi van akkor, ha $H < E$ reláció is fennáll?*

3.4. Maximális valószínűségű út

Tegyük fel, hogy adott egy irányított gráf, amelynek az élein egymástól függetlenül, ismert valószínűségekkel lehet áthaladni. Kijelölve egy s és t pontot, szeretnénk meghatározni azt az utat, amelyen legnagyobb valószínűséggel eljutunk s -ből t -be.

Példa:



Ha rögzítünk néhány élt, akkor annak valószínűsége, hogy mindegyik járható, a függetlenség miatt a hozzájuk rendelt valószínűségek *szorzata*. Jelen esetben a legnagyobb valószínűséggel nyitva levő $s - t$ út az (s, x, y, v, t) , a járhatóság valószínűsége $0,8 \times 0,8 \times 0,7 \times 0,7 = 0,3156$. Általában a következőt tehetjük:¹ cseréljük ki minden e élre az élhez tartozó $p(e)$ valószínűséget $-\ln p(e)$ -re, és keressük a legrövidebb $s - t$ utat ezen súlyozás mellett. Legyenek R és Q két különböző $s - t$ út élhalmazai; ekkor az utak járhatóságának valószínűségei rendre

$$\prod_{e \in R} p(e) \text{ és } \prod_{e \in Q} p(e),$$

míg a transzformáció mellett a hosszuk

$$-\sum_{e \in R} \ln p(e) \text{ és } -\sum_{e \in Q} \ln p(e).$$

$$\text{Ha } -\sum_{e \in R} \ln p(e) < -\sum_{e \in Q} \ln p(e), \text{ akkor } \sum_{e \in R} \ln p(e) > \sum_{e \in Q} \ln p(e).$$

Ebből

$$\ln\left(\prod_{e \in R} p(e)\right) > \ln\left(\prod_{e \in Q} p(e)\right),$$

¹Valóban, ez az ötlet nagyon általános, a szorzatok kiszámításának összeadásra való visszavezetése a logaritmus függvény bevezetésének egyik fő motivációja.

és a logaritmus függvény monotonitása miatt kapjuk a

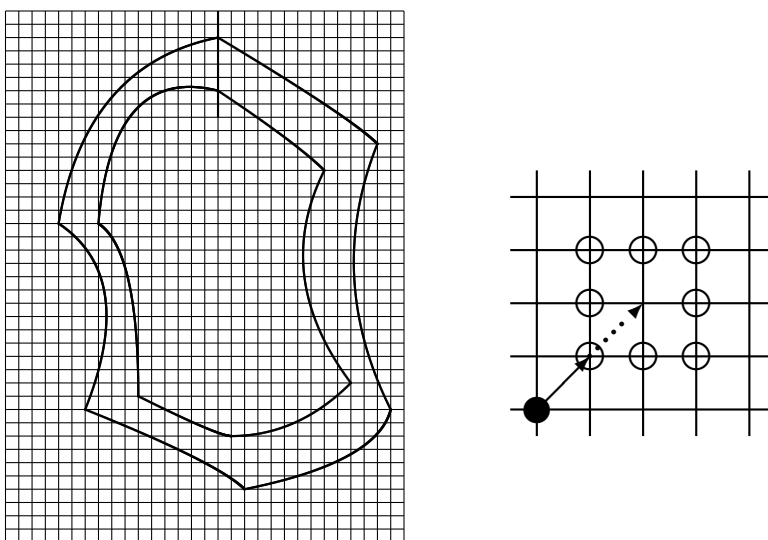
$$\prod_{e \in R} p(e) > \prod_{e \in Q} p(e)$$

egyenlőtlenséget, ami bizonyítja állításunkat.

3.5. Fázisterek

A fázisterek alapgondolata az, hogy a vizsgálni kívánt rendszer *teljes* állapotát egyetlen pont segítségével írjuk le.² Itt egy közismert, de mégsem triviális példán illusztráljuk a fázistereken alapuló modellezést.

Példa (Autóverseny): Az ún. *autóverseny* játékot egy kockás lapon játszhatjuk, ahol az autó egy lépését egy nyíl reprezentálja. (A nyíl az autó egységnyi idő alatt megtett útja. Mivel a gyorsulás nem tetszőlegesen nagy, ezért lépésenként csak $\alpha_1 e_1 + \alpha_2 e_2$ -vel változtatható, ahol e_1, e_2 a sík mérőleges egységvektorai $\alpha_1, \alpha_2 \in \{-1, 0, 1\}$.) Adott továbbá egy pálya, amit az autó nem hagyhat el, egy rajt- és célvonal, ahonnan indul, illetve ahova érkeznie kell a lehető legkevesebb lépés alatt.



²Ez rendkívül hasznosnak bizonyult a fizikában, ahol például egy n tömegpontból álló rendszer mozgását $6n$ koordináta (pontonként 3 tér és 3 sebesség) írja le. Ez felfogható az $\mathbf{R}^{6n}$ egyetlen pontjaként, míg a rendszer időbeli változása egy térgörbéként. Többnyire ez a hozzáállás segít a Markov láncokkal való modellezésben is: a lánc egy-egy állapota a rendszer teljes leírását kódolja.

A jobboldali ábrán látható, hogyan lehet meghatározni a következő lehetséges lépést: megismétljük az előző lépés vektorát (a szaggatott vektor) és a végponthoz tartozó rácspontba, valamint ennek szomszédaiba léphetünk. A keletkező út *nem* gráfút, hiszen akárhová nem léphetünk; ez a sebességtől is függ. Készíthetünk azonban egy *fázistér gráfot*, amelynek pontjai mind az autó helyét, mind a sebességét kódolják és a pontok között akkor van él, ha az átmenet egyik állapotból a másikba egy lépés alatt megtörténhet. Egy (a, b) síkbeli pontpárnak megfelelt egy gráfpont, ahol a pontbeli állapot azt jelzi, hogy az autó a b pontban van és oda az a -ból lépett. Ekkor az állapotgráfnak olyan (b, c) pontjaira lehet átlépni (van él), amelyekre a bc vektor az ab vektortól bármely koordinátájában legfeljebb eggyel tér el. Súlyozzunk minden élt eggyel, akkor az állapotgráf leg-rövidebb útja a rajt és cél között a minimális lépésszámú útvonalnak felel majd meg.

Megjegyzések:

1. Ha a pálya n pontból áll, az állapotgráf n^2 pontból fog. A példánkban $n \leq 100$, így az állapotgráfnak tízezernél kevesebb pontja van.
2. Az eljárással modellezhetőek egyéb feltételek: olajfolt, amin nem lehet lassítani, homokágy, ami lassít stb.

3.6. Feladat. *Adjuk meg a probléma modelljét, ha a versenypályán bizonyos helyeken olajfoltok, máshol pedig homokágy van.*

3.7. Feladat (Alcuin). *Kétszemélyes csónakon szállítsunk át a folyón egy kecskét, káposztát és egy farkast, úgy, hogy a kecske nem maradhat felügyelet nélkül sem a farkassal sem a káposztával.*

3.8. Feladat. *Egy folyó hídja két embert bír el egyszerre, és zseblámpát kell használni, amiből csak egy van. 4 embert kell átjuttatnunk egyik oldalról a másikra, akik egymagukban 1, 2, 5 illetve 10 percet igényelnének. Ha együtt mennek, a lassabbik idejét futják. Mennyi idő alatt végezhetnek?*

4. fejezet

Totálisan unimoduláris mátrixok

Látni fogjuk, hogy számos gráf probléma megfogalmazható lineáris programozási (LP), illetve egészértékű programozási (IP) feladatként. Az egészértékű programozás fő nehézsége abban rejlik, hogy a lehetséges megoldásokból álló poliédernek esetleg nem egész koordinátájú csúcspontjai vannak. Ha valamiképpen előre tudnánk, hogy csak egész csúcspontok vannak, akkor a folyamatos relaxáció bármely optimális bázismegoldása automatikusan az egészértékű probléma optimális megoldása lenne. Mint azt A. J. Hoffman és S. B. Kruskal 1956-ban megmutatta az egészértékű problémák egy elméleti és gyakorlati szempontból egyaránt lényeges osztályára teljesül a feltétel. Eredményük központi jelentőségű a következő fejezetben tárgyalandó ún. *hálózati* vagy *network* problémák vizsgálatában.

A fejezet megértéséhez szükséges lineáris programozás elméletének alapvető ismeretét (úgy mint a bázismegoldás, szimplex algoritmus, LP és konvex geometria közti kapcsolat) feltételezzük. További olvasnivaló a témában pl. Alexander Schrijver könyvének [15] 19-21. fejezetei.

4.1. TU mátrixok

Az alábbi fogalomra lesz szükségünk: Egy A mátrix *totálisan unimoduláris*, ha bármely négyzetes részmátrixának a determinánsa 1 , -1 vagy 0 . (Vagyis speciálisan A elemei csak a ± 1 és a 0 számok lehetnek.)

4.1. Tétel (Hoffman-Kruskal). *Ha A totálisan unimoduláris, b és c egész vektorok, akkor a $\max c^T x$ feltéve $Ax \leq b$ lineáris programozási feladat (LP) bázismegoldásai, (köztük az optimális bázismegoldásai) egész értékűek. Másképpen a $P = \{x : Ax \leq b\}$ poliéder csúcsai egész koordinátájúak.*

Bizonyítás. Ha adott egy B bázis, a hozzátartozó x_B bázismegoldás szótára:

$$x_B = A_B^{-1}b - A_B^{-1}A_Nx_N$$

$$z = c_B A_B^{-1}b + (c_N - c_B A_B^{-1}A_N)x_N$$

(A módosított szimplex algoritmus leírásánál használtuk ezt a formulát; A_B az A mátrix B bázis által kijelölt oszlopaiból áll. Konvencionálisan az A_B -t B -vel jelölik, a félreértés elkerülése miatt használjuk az A_B formát.)

Az A_B^{-1} mátrix elemeit lineáris algebrában megismert *Cramer szabály* szerint számolhatjuk ki. Ha $(A_B^{-1})_{i,j}$ a mátrix i, j -edik eleme, ${}^{ij}A_B$ az a mátrix, amit A_B -ből a i -edik sor és j -edik oszlop eltörlésével kapunk, $\det(A_B)$ pedig a mátrix determinánsa, akkor

$$(A_B^{-1})_{i,j} = \frac{(-1)^{i+j} \det({}^{ij}A_B)}{\det(A_B)}.$$

Mivel A totálisan unimoduláris, és $\det(A_B) \neq 0$, így $\det(A_B) = \pm 1$. Természetesen a $\det({}^{ij}A_B^{-1})$ mátrix *minden* eleme egész. Mivel $x_B = A_B^{-1}b$, egy egész mátrix és egész vektor szorzata, x_B is egész. □

Megjegyzés: Természetesen ha a szimplex módszerrel oldjuk meg a problémát, a szótár többi eleme $(A_B^{-1}A_Nx_N)$, ill. $z = c_B A_B^{-1}b + (c_N - c_B A_B^{-1}A_N)x_N$ is egész számokból áll, így kerekí tési hibák sem lépnek fel.

Nem nyilvánvaló viszont, hogyan dönthetjük el egy adott mátrix totálisan unimoduláris-e vagy sem. A totálisan unimoduláris mátrixok karakterizálhatók, mi több, gyors algoritmus is megadható ennek a tulajdonságnak az eldöntésére. Sajnos az erre vonatkozó tételek és algoritmusok ismeretése meghaladja lehetőségeinket. (Megjegyezzük azonban, hogy a fent említett karakterizáció általánosításképpen 1981-ben Paul Seymour bebizonyította a kombinatorika egyik legmélyebb tételét az ún. *reguláris matroidok* dekompozíciós tételét.)

A következő tétel központi jelentőségű az LP-ként felírt hálózati problémák megoldása kapcsán.

4.2. Tétel. Egy irányított gráf A illeszkedési mátrixa totálisan unimoduláris.

Bizonyítás. Használjunk indukciót a részdeterminánsok mérete szerint. Az A mátrix 1×1 -es részmátrixai ± 1 -ek és 0 -ák, így $k = 1$ -re igaz a feltétel. Ha adott egy B $(k + 1) \times (k + 1)$ -es részmátrix, akkor két esetre bomlik a bizonyítás. Ha a B mátrixnak van olyan oszlopa, amelynek legfeljebb egy nem nulla eleme van, akkor determinánsát ezen oszlop szerinti kifejtéssel kapjuk. Az indukciós feltétel miatt ez 0 vagy ± 1 . Ha B minden oszlopa pontosan két nem nulla elemet tartalmaz (azaz $+1$ -et és -1 -et), akkor B sorainak összege a zérus vektor, vagyis B sorai lineárisan függőek. Ekkor $\det(B) = 0$, és ezzel igazoltuk a tételt. $\square$

Megjegyzés: Az előző tételünkkel kombinálva leszűrhetjük, hogy minden az illeszkedési mátrix segítségével definiált egészértékű programozási feladat egy LP feladattá egyszerűsödik. Ez megkönnyíti a feladat megoldását (tulajdonképpen *polinom idejűvé* teszi), másrészt a dualitási tételt teljes erejében kihasználhatóvá teszi. Ennek pontos tárgyalása messzire vezetne, így a későbbiekben csupán illusztrálni tudjuk ezt egy-egy példával.

Egy másik alkalmazás a *maximális párosítás* megtalálása páros gráfokban. Ehhez a következő tételre van szükségünk.

4.3. Tétel. *Egy páros gráf incidencia mátrixa totálisan unimoduláris.*

Bizonyítás. A 4.2 tételhez hasonlóan a részdeterminánsok mérete szerinti mérete szerinti teljes indukcióval belátható. $\square$

Mivel a maximális párosítás felírható a következő egészértékű programozási feladatként

$$\begin{aligned} \max \quad & \sum_{e \in E} x_e \\ & \sum_{e \sim u} x_e \leq 1 \quad (\forall u \in V) \\ & x_e \in \{0, 1\} \end{aligned}$$

ahol $e \sim u$ az illeszkedést jelöli. A problémát kompaktabb formában írva és relaxálva a $\max \mathbf{1}^T x$, feltéve $Ax \leq \mathbf{1}$, $x \geq 0$ feladathoz jutunk, ahol A a páros gráfunk illeszkedési mátrixa. Mivel ez totálisan unimoduláris, ezért az LP megoldása egészértékű.

4.2. TU mátrixok karakterizációi

Általánosságban sajnos (egyelőre) nincs egyszerű eljárás ami eldönti egy mátrixról, hogy totálisan unimoduláris-e. Ugyanakkor léteznek könnyen tesztelhető szükséges (de nem elegendő) feltételek, továbbá léteznek karakterizációs tételek is, ezek közül itt kettőt említünk meg.

4.4. Tétel (Camion, 1965). Egy $0, \pm 1$ elemekből álló A mátrix akkor és csak akkor totálisan unimoduláris, ha az elemek összege 4-gyel osztható minden olyan négyzetes részmátrixában, melynek minden minden sora és minden oszlopa páros.¹

4.5. Tétel (Ghouila-Houri, 1962). Egy $m \times n$ -es A egész mátrix totálisan unimoduláris, ha minden $R \subseteq \{1, 2, \dots, m\}$ halmaz két diszjunkt R_1, R_2 halmazra bontható úgy, hogy $\sum_{i \in R_1} a_{ij} - \sum_{i \in R_2} a_{ij} \in \{0, \pm 1\}$ ($j = 1, 2, \dots, n$).

Végül egy könnyen ellenőrizhető elegendő feltételt mondunk ki.

4.6. Tétel. Egy $0, \pm 1$ elemekből álló A mátrix totálisan unimoduláris, ha egyetlen oszlopában sem tartalmaz egynél több $+1$ -est és egy egynél több -1 -est.

Megjegyzés. Az utolsó állításból rögtön következik a 4.2 tétel.

4.1. Feladat. Bizonyítsuk be, ha egy A mátrix totálisan unimoduláris, akkor bármely sorát vagy oszlopát -1 -gyel szorozva továbbra is totálisan unimoduláris lesz.

4.2. Feladat. Bizonyítsuk be, hogyha egy A mátrix totálisan unimoduláris, akkor A^T is totálisan unimoduláris.

4.3. Feladat. Egy gráf él-út illeszkedési mátrixa² totálisan unimoduláris.

4.4. Feladat. Legyen A $0, \pm 1$ elemekből álló mátrix olyan, hogy minden oszlopban az első 1 -es után (fentről-lefelé) már minden elem 1 -es. Bizonyítsuk be, hogy A totálisan unimoduláris.

4.5. Feladat. Adjunk meg egy olyan egész csúcspontú politópot ($\{x : Ax \leq b, x \geq 0\}$), ahol A 3×3 -as, A és b elemei egészek, de A nem teljesen unimoduláris. Tudunk olyan példát, ahol A elemei csak $0, \pm 1$. És olyat, hogy csak $0, +1$?

¹Az ilyen mátrixokat szokás Euler-féle mátrixnak is nevezni.

²A gráf minden éléhez tartozik a mátrix egy sora, és a gráf minden útjához a mátrix egy oszlopa. Egy sor-oszlop találkozásnál 1 -es van, ha a megfelelő él benne van a megfelelő útban, 0 különben.

5. fejezet

Hálózati problémák

Az ún. hálózati vagy *network* problémák rendkívül elterjedtek az operációkutatásban. Ennek három fő oka van:

1. Módfelett alkalmasak gyakorlati problémák modellezésére
2. Hatékony algoritmusok dolgozhatók ki megoldásukra
3. Matematikai szempontból nagyon elegáns elméletük van.

Ennek megfelelően a 40-es évektől kezdve számos modellel foglalkoztak a témakör kutatói, amiből itt csak néhányat és érintőlegesen tárgyalhatunk.

5.1. Transshipment probléma

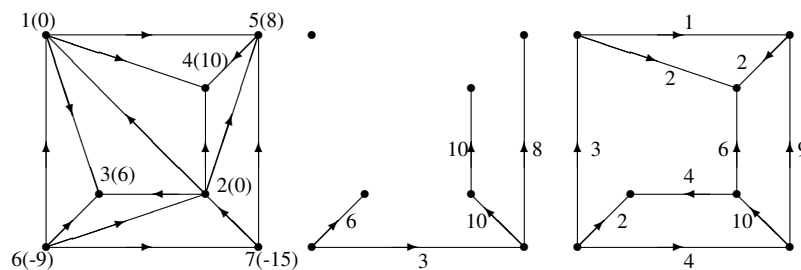
Az egyik legfontosabb a *transshipment probléma*.¹ Adott egy G irányított gráf, melynek éleihez valós számokat (*költségeket*) rendelünk. Számokat rendelünk a gráf pontjaihoz is. Ha ez a szám pozitív, akkor a pontot *nyelőnek*, ha negatív *forrásnak*, ha nulla *belső pontnak* nevezzük. Feltehetjük, hogy a pontokhoz rendelt számok összege nulla. (Ha nem, egy új pont felvételével ilyen formára hozhatjuk G -t.)

Tegyük fel, hogy valamilyen árut kell szállítanunk a forrásokból a nyelőkbe a gráf élei mentén úgy, hogy bármely forrásban a *kínálat*, illetve nyelőben a *kereslet* a hozzárendelt a hozzárendelt

¹Ezt szállítási problémának is szokták magyaráítani. Mi itt inkább *szállítmányozási problémaként* emlegetjük, hogy ne keveredjen össze egy speciális esettel, a Hitchcock-féle szállítási feladattal.

sám abszolút értéke (a belső pontokban nulla). Egy él mentén a szállítási költsége az élhez rendelt szám és a szállított áru mennyiségének szorzata, míg az összköltség a szállítási költségek összege.

Példa: A pontok mellé írt szám a pont neve, zárójelben a kereslet/kínálat. A két megoldásban csak a szállításba bekapcsolódó éleket ábrázoltuk.



A transshipment probléma felírható LP feladatként, ahol A a G gráf incidencia mátrixa, az x vektor x_{ij} komponense az (i, j) élen szállított áru mennyisége, c_{ij} az egységnyi szállítás költsége, a b vektor pedig az egyes pontokban jelentkező kereslet/kínálat.

$$\begin{aligned}
 \min \quad & \sum_{(i,j) \in E(G)} c_{ij} x_{ij} = c^T x \\
 (*) \quad & Ax = b \\
 & x \geq 0
 \end{aligned}$$

A példánkban az A mátrix a következő:

	1,3	1,4	1,5	2,1	2,3	2,4	2,5	5,4	6,1	6,2	6,3	6,7	7,2	7,5
1	-1	-1	-1	1	0	0	0	0	1	0	0	0	0	0
2	0	0	0	-1	-1	-1	-1	0	0	1	0	0	1	0
3	1	0	0	0	1	0	0	0	0	0	1	0	0	0
4	0	1	0	0	0	1	0	1	0	0	0	0	0	1
5	0	0	1	0	0	0	1	-1	0	0	0	0	0	1
6	0	0	0	0	0	0	0	0	-1	-1	-1	-1	0	0
7	0	0	0	0	0	0	0	0	0	0	0	1	-1	1

továbbá

$$x_T = [x_{13}, x_{14}, x_{15}, x_{21}, x_{23}, x_{24}, x_{25}, x_{54}, x_{61}, x_{62}, x_{63}, x_{67}, x_{72}, x_{75}]$$

és

$$b_T = [0, 0, 6, 10, 8, -9, -15].$$

Vegyük észre, hogy a (*) probléma mátrixából törölhetjük az utolsó (de tulajdonképpen bármely) sort és a b vektor utolsó koordinátáját a lineáris függőség miatt.

A transshipment probléma megoldására különböző lehetőségek adódnak. Ha a G gráf, irányítástól eltekintve összefüggő, akkor a (*) LP egy tetszőleges bázis megoldásának a bázisban lévő változókhoz tartozó élei egy fát alkotnak. Ezeknek a fának a megfelelő transzformációival fogalmazható meg az ún. *network simplex algoritmus*. Ez a jól ismert simplex algoritmus egy, a gyakorlatban rendkívül hatékony változata. Más eljárások ennél is direktebb módon használják ki a feladat speciális szerkezetét, és, legalábbis elméleti vonatkozásban, felülmúlják a network simplex algoritmust. Egy egyszerűbb feladat, a *maximális folyam probléma* vizsgálatában betekintést adunk ezekbe a módszerekbe. Itt megelégszünk az ún. *integralitás tétel* bizonyításával.

5.1. Tétel (integralitás). *Tegyük fel, hogy a fenti transshipment probléma b vektorának minden koordinátája egész szám. Ha a feladatnak van lehetséges megoldása, akkor van egész megoldása is. Ha van optimális megoldása, akkor van optimális egész megoldása is.*

Bizonyítás. A lehetséges megoldás létezéséből következik (az LP alaptételéből), hogy van lehetséges bázis megoldás. Korábbi tételeink szerint az A mátrix totálisan unimoduláris, és így a probléma bármelyik lehetséges bázis megoldása egész. Ismét az LP alaptételét használva, ha van optimális megoldás, akkor van optimális bázismegoldás, amely az előző pont miatt szükségképpen egész. □

A transshipment probléma formalizmusa és az integralitás tétel az operációkutatás és a kombinatorika szerencsés keveréke. Valójában tömördek korábbi probléma/modell közös általánosítása. Mi a fordított utat követjük: az általános probléma speciális eseteiként mutatunk be néhányat ezek közül. Először azonban lássunk egy nem triviális alkalmazást.

5.2. Az étkeztető (caterer) probléma

A modell eredetileg egy *ütemezési probléma*, amellyel a légierő repülőgép motorjainak karbantartását tervezték. Később a megalkotói (Jacobs, Gaddum, Hoffman, Sokolowsky és Prager) a demilitarizált *caterer (étkeztetési) probléma* néven publikálták.

Adott egy étkeztető, akinek n napon át tiszta szalvétára van szüksége, a j -edik napon egy előre ismert d_j darabra. Ezt részint új szalvéták vásárlásával (a cent darabja), részint a használtak mosásával fedezheti. Az utóbbiban választhat a gyors (q nap alatt, darabonként b centért) és a lassú (p

nap alatt, c centért) ajánlott szolgáltatások között. Természetesen $p > q$ és $a > b > c$, a minimális költség pedig a cél.

Példa: Legyen egy étkeztető feladatban $n=10$, $d_1=50$, $d_2=60$, $d_3=80$, $d_4=70$, $d_5=50$, $d_6=60$, $d_7=90$, $d_8=80$, $d_9=50$, $d_{10}=100$; míg $p=4$, $q=2$, $a=200$, $b=75$, $c=25$.

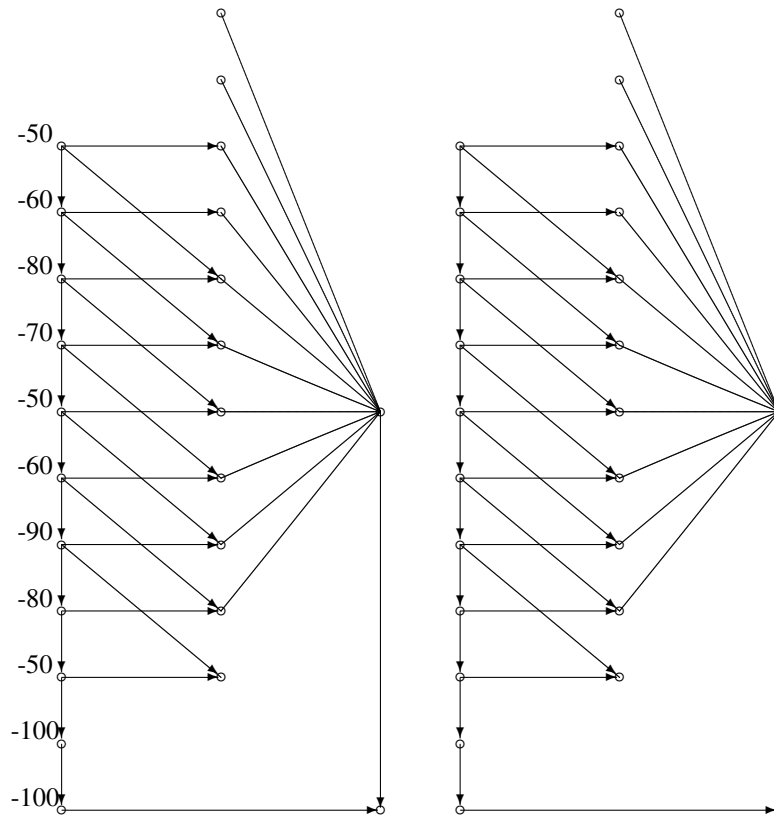
Az alábbi táblázatban összefoglalunk egy lehetséges és egy optimális (a zárójelben lévő számok) stratégiát. Az utóbbi egy a feladathoz rendelt transshipment probléma optimális bázis megoldása. A feladat grájában a hozzátartozó fát szintén ábrázoljuk majd.

Nap	Mosott	Vásárolt	Gyors mosás	Lassú mosás	Tárolás
1	0 (0)	50 (50)	50 (0)	0 (50)	0 (0)
2	0 (0)	60 (60)	60 (0)	0 (60)	0 (0)
3	50 (0)	30 (80)	50 (0)	30 (80)	0 (0)
4	60 (0)	10 (70)	60 (0)	0 (70)	10 (0)
5	50 (50)	0 (0)	60 (10)	0 (0)	0 (40)
6	60 (60)	0 (0)	60 (10)	0 (90)	0 (0)
7	90 (90)	0 (0)	50 (50)	0 (0)	40 (40)
8	60 (80)	20 (0)	100 (10)	0 (0)	20 (110)
9	50 (50)	0 (0)	0 (0)	0 (0)	70 (160)
10	100 (100)	0 (0)	0 (0)	0 (0)	170 (260)

A transshipment modell és az optimális megoldás fája a következő: Az baloldali ábra értelmezése a következő. A baloldali pontokban, mint tárolókban jelenik meg a naponként elhasznált szalvéta. Ezekből a forrásokból tovább küldhetők gyors mosással a két nappal későbbi felhasználásra (vízszintes élek, b költség), lassú mosással (ferde élek, c költség), vagy tárolhatók a következő napig (függőleges élek, 0 költség). A j -dik napon igényelt d_j darab szalvéta beszerezhető a $j - q$ tárolóból gyors, a $j - p$ -ből lassú mosatással, vagy a boltból a költségű élen. Célszerű feltenni, hogy a boltban elegendő szalvéta van, és akár az egész szükséglet fedezhető vásárlás útján. Az utolsó napon a tárolóban lévő szalvétákat egy (képzelt) gyűjtőbe irányítjuk zero költség árán csakúgy, mint az n -edik nap után a boltban maradtakat. (Ez utóbbi szintén zero költséggel tehető meg.) Ezek segítségével az étkeztetési problémát átvitük egy transshipment problémára.

A jobboldali ábra értelmezése: Az optimális bázis megoldás bázisában lévő változóknak megfelelő éleket ábrázoltuk. Az élekre írt számok a változók értékei. Vegyük észre, hogy némelyik bázisváltozó nulla értéket kap, azaz a megoldás degenerált. Ez a hálózati problémák megoldásánál

igen gyakori jelenség. Az integralitás tétel garantálja, hogy van egész optimális megoldás, és így a kapott ütemezési terv kerekítések nélkül alkalmazható.



5.3. Legrövidebb út probléma

A legrövidebb utak keresésére már adtunk kielégítő algoritmust. Most az új modellünk erejét illusztráljuk. Tegyük fel, hogy egy irányított, súlyozott G gráfban egy legrövidebb $s - t$ utat keresünk. Rendeljük s -hez -1 , míg t -hez 1 keresletet. Tekintsük a súlyokat költségeknek és vegyük az így adódó transshipment probléma egy egész optimumát, ha az létezik. Ha G -ben nincs negatív kör, könnyen látható, hogy az élekhez rendelt változók csak 0 és 1 értéket vesznek majd fel, és az 1 értékhez tartozó éleken eljuthatunk s -ből t -be. (Parciálisan a megfordítás is igaz: a legrövidebb utak keresése fontos *részfeladat* a transshipment probléma megoldására adott némely algoritmusban.)

5.4. Szállítási (transportation) feladat

A problémát 1941-ben F. I. Hitchcock tanulmányozta, így néha *Hitchcock-féle szállítási feladat*-ként is emlegetik:

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ & \sum_{j=1}^n x_{ij} = r_i, \quad (i = 1, 2, \dots, m) \\ & \sum_{i=1}^m x_{ij} = s_j, \quad (j = 1, 2, \dots, n) \\ & x_{ij} \geq 0 \end{aligned}$$

Könnyen látható, hogy ez egy kettő kromatikus számú gráfban (vagy más szóval *páros gráfon*) definiált transshipment probléma. Definiáljuk G úgy, hogy $V(G) = U \cup V$, ahol $U = \{u_1, u_2, \dots, u_m\}$ a források, $V = \{v_1, v_2, \dots, v_n\}$ a nyelők halmaza, és $E(G) = \{u_i v_j : u_i \in U, v_j \in V\}$. Legyen továbbá a kínálat r_i az u_i pontnál ($i = 1, \dots, m$), a kereslet pedig s_j a v_j pontnál ($j = 1, \dots, n$). A szállítási feladat ezek után megoldható az integralitás tétel alapján. A szállítási feladatnak a nyilvánvalóan látszó felhasználhatóságon kívül sok mély alkalmazása van, melyekre itt nem térhetünk ki.

5.5. Hozzárendelési (assignment) probléma

Egy példán keresztül vezetjük be a feladatot. Tegyük fel, hogy van 5 ember, 5 különböző feladat, és egy számmal leírhatjuk az i -edik ember teljesítményét a j -edik feladaton ($0 \leq i, j \leq 5$). Adjunk minden embernek pontosan egy feladatot úgy, hogy a teljesítmények összege a lehető legnagyobb legyen. A teljesítmények leírhatók egy mátrix-szal:

	1	2	3	4	5
1	7	5	4	4	5
2	7	9	7	9	4
3	4	6	5	8	5
4	5	4	5	7	4
5	4	5	5	8	9

A feladatra egész értékű programozási modellt célszerű felállítani. Legyen $x_{ij} = 1$, ha az i -edik ember a j -edik feladatot kapja, $x_{ij} = 0$ különben. Ezzel a feltételek $\sum_{j=1}^5 x_{ij} = 1$ ($i = 1, \dots, 5$) (minden ember pontosan egy feladatot kap) és $\sum_{i=1}^5 x_{ij} = 1$ ($j=1, \dots, 5$) (minden feladatot pontosan egy ember végez el), a célfüggvény pedig a mátrix együtthatókból áll: $\sum_i \sum_j c_{ij} x_{ij}$. Általában a hozzárendelési probléma:

$$\begin{aligned} \max \quad & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ (*) \quad & \sum_{j=1}^n x_{ij} = 1 \quad (i = 1, \dots, n) \\ & \sum_{i=1}^n x_{ij} = 1 \quad (j = 1, \dots, n) \\ & x_{ij} \in \{0, 1\} \quad (1 \leq i, j \leq n). \end{aligned}$$

Vegyük észre, hogy a (*) problémának *mindig* van pontosan $n!$ lehetséges, és ebből következően legalább egy optimális megoldása is, és az integralitási tétel miatt az alábbi szállítási feladat optimális bázismegoldása a (*)-nak is optimális megoldása:

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^n (-c_{ij}) x_{ij} \\ & \sum_{j=1}^n x_{ij} = 1 \quad (i = 1, \dots, n) \\ & \sum_{i=1}^n x_{ij} = 1 \quad (j = 1, \dots, n) \\ & x_{ij} \leq 0 \quad (1 \leq i, j \leq n). \end{aligned}$$

Megjegyezzük, hogy a (*) feladatnak 1950-ben P. R. Halmos és H. Vaughn egy meglehetősen abszurd „alkalmazását” publikálta. Ebben n házasodni kívánó férfi, illetve nő egymáshoz rendelését írják le olyan módon, hogy a c_{ij} szám a j -edik nő által az i -edik férfinak adott pontszám (n a legjobb esetben, 1 a legrosszabban), míg az „összboldogság” arányos $\sum_i \sum_j c_{ij}$ -vel. Ez számunkra

csak annyiban érdekes, hogy a későbbiekben egy valamivel reálisabb feltétel mellett megvizsgáljuk és megoldást adunk az ún. *stabil házassági problémára*. A probléma és a rá vonatkozó integralitás tétel jól használható a *páros gráfok párosítási problémáinak* tanulmányozásában. Egy G gráf *páros*, ha $V(G) = A \cup B$, ahol $A \cap B = \emptyset$, és minden él A és B között húzódik (jelben $\chi(G) \leq 2$). Könnyen belátható, hogy egy G páros gráf maximális M párosítása az alábbi LP megoldásával megkapható:

$$\begin{aligned} \max \quad & \sum_{j \in B} \sum_{i \in A} x_{ij} \\ & \sum_{j: (i,j) \in E(G)} x_{ij} \leq 1 \quad (i \in A) \\ & \sum_{i: (i,j) \in E(G)} x_{ij} \leq 1 \quad (j \in B) \\ & x_{ij} \leq 0 \quad (i \in A, j \in B), \end{aligned}$$

ahol M azon (i, j) élekből áll, melyekre $x_{ij}^* = 1$ az optimális megoldásban. Hasonló könnyedséggel kaphatjuk meg König Dénes híres tételét:

5.2. Tétel. *Tegyük fel, hogy G olyan páros gráf, melyre $|A| = |B| = n$ és minden $v \in V(G)$ pont fokszáma $d(v) = k \geq 1$. Ekkor G -ben van teljes párosítás.*

Bizonyítás. Készítsünk egy transshipment problémát a G gráfhoz úgy, hogy A pontjaihoz -1 -et, B pontjaihoz 1 -et rendelünk és az éleket A -ból B -be irányítjuk. (Az él költségével nem törődünk.) A keletkező problémának *van* lehetséges x megoldása: egyszerűen legyen $x_e = 1/k$ minden e élre. Így az integralitás tétel miatt van x^* lehetséges egész megoldás, amelyben $x_e^* \in \{0, 1\}$ minden e élre. Valójában A minden pontjából pontosan egy él vezet ki, melyekhez tartozó változó értéke 1 , és ezek az élek egy M teljes párosítást adnak. $\square$

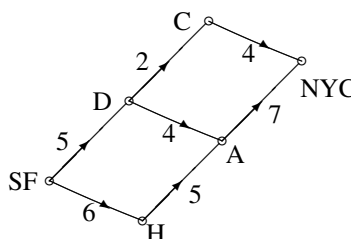
5.6. Maximális folyam

A maximális folyam probléma felfogható egy olyan transshipment problémaként, amelyben a változók értékeire felső korlátokat kötünk ki. Ezt a kapcsolatot felhasználva elméleti következtetéseket vonhatunk le, illetve a feladatot megoldó algoritmusokhoz juthatunk. A terület fejlődése nem

ezt az irányt követte és mi sem fogjuk ezt tenni, ugyanis közvetlen, a simplex algoritmusra nem hivatkozó eszközökkel tárgyalható a probléma. Mi több, az adódó bizonyítások egyszerűbbek, az algoritmusok hatékonyabbak, mintha általános elméleten keresztül érnénk el céljainkat.

Példa: Kezdjük egy klasszikus példával, amelyben San Franciscoból New Yorkba igyekszik egy repülőtársaság maximális számú embert utaztatni. Útközben több átszállási lehetőség van és az egyes városok között szállítható utasok számát egy táblázatban foglaljuk össze.

Honnan	Hová	Helyek száma
San Francisco	Denver	5
San Francisco	Houston	6
Denver	Atlanta	4
Denver	Chicago	2
Houston	Atlanta	5
Atlanta	New York	7
Chicago	New York	4



A táblázatból egy irányított gráfot írhatunk fel; az élek súlyozása az elszállítható utasok maximális száma a két pontnak megfelelő város között. A következőket szeretnénk formalizálni egy szállítási tervben:

1. Ha (v, w) úton k ember megy az ugyanaz, mint a (w, v) úton $-k$ ember.
2. A egy úton küldött emberek száma nem haladhatja meg az út korlátját.
3. SF és NYC kivételével a beérkező és távozó utasok száma egyenlő, azaz az 1. pont miatt az összeg nulla.

Legyen G egy irányított gráf két kitüntetett ponttal, s (forrás) és t (nyelő). Jelölje $c(v, w)$ a (v, w) él kapacitását, azaz az élhez tartozó változó maximális értékét, továbbá legyen $c(v, w) = 0$, ha (v, w) nem él G -ben. Az $f : V(G) \times V(G) \rightarrow R$ függvény *folyam*, ha az alábbi három feltételnek eleget tesz:

- (1) *Ferde szimmetria.* Minden v, w pontra $f(v, w) = -f(w, v)$. Ha $f(v, w) > 0$, akkor azt mondjuk, *van* folyam v -ből w -be.

(2) *Kapacitás korlát.* Minden v, w pontra $f(v, w) \leq c(v, w)$. Ha (v, w) olyan, hogy $f(v, w) = c(v, w)$, azt mondjuk a folyam *telíti* (v, w) -t.

(3) *A folyam megmaradás elve.* Minden s és t -től különböző v pontra $\sum_{w \in V(G)} f(v, w) = 0$.

A folyam *értéke*, $|f| := \sum_{w \in V(G)} f(s, w)$, azaz a forrásból kifolyó mennyiség. A *maximális folyam probléma* a maximális értékű folyam megkeresése. Vizsgálataink központi fogalma a *vágás*. Egy $X, \bar{X}$ vágás a $V(G)$ pontthalmaz olyan partíciója, hogy $s \in X$ és $t \in \bar{X} = V(G) \setminus X$. Az $X, \bar{X}$ vágás *kapacitása* $c(X, \bar{X}) := \sum_{v \in X, w \in \bar{X}} c(v, w)$. Egy minimális kapacitású vágást *minimális vágásnak* nevezünk. Ha f egy folyam, $X, \bar{X}$ egy vágás, akkor a *vágáson átmenő folyam* $f(X, \bar{X}) := \sum_{v \in X, w \in \bar{X}} f(v, w)$.

5.1. Lemma. *Bármely f folyam és $X, \bar{X}$ vágás esetén a vágáson átmenő folyam egyenlő a folyam értékével.*

Bizonyítás.

$$f(X, \bar{X}) = \sum_{v \in X, w \in \bar{X}} f(v, w) = \sum_{v \in X, w \in V(G)} f(v, w) - \sum_{v \in X, w \in X} f(v, w) = |f|,$$

mert $\sum_{v \in X, w \in V(G)} f(v, w) = |f|$ a folyam megmaradás, $\sum_{v \in X, w \in X} f(v, w)$ pedig nulla a ferde szimmetria törvénye miatt. $\square$

A lemma szemléletes jelentése az, hogy a folyam értéke egyenlő az X -et elhagyó *nettó* mennyiséggel. Mivel $f(v, w) \leq c(v, w)$,

$$|f| = \sum_{v \in X, w \in X} f(v, w) \leq \sum_{v \in X, w \in \bar{X}} c(v, w)$$

minden f folyam és $X, \bar{X}$ vágás esetén. Így a maximális folyam értéke nem haladja meg a minimális vágás értékét. A meglepő az, hogy ez a két érték egyenlő. Ez igazából az LP dualitás tétel egy speciális esete, melyet néhány fogalom bevezetésével közvetlenül is beláthatunk.

A *maradék kapacitás* egy f folyamra v és w pontok között $r(v, w) := c(v, w) - f(v, w)$. Szemléletesen $r(v, w)$ egységgel növelhetjük a folyamot v -ből w -be (ha ugyanennyivel csökkentjük $f(w, v)$ értékét). Az R *maradék gráf* egy folyamra G pontjaiból és azon (v, w) párokból, mint élekből áll, amelyekre $r(v, w) > 0$: lehetnek olyan élei, melyek *nincsenek* G -ben. Egy p *növelő út* az f folyamra egy s -ből t -be vezető út R -ben. A p út *maradék kapacitása*, $r(p)$ a p -ben lévő élek

maradék kapacitásainak minimuma. A p növelő út szerepe nyilvánvaló: a p élei mentén $r(p)$ -vel növelve az f folyamot, a folyam értéke $r(p)$ -vel növelhető. (Ha $f(v, w)$ -t változtatjuk, változik $f(w, v)$ is!)

5.2. Lemma. *Legyen f egy tetszőleges, f^* pedig egy maximális folyam G -n. Ha R az f -hez tartozó maradék gráf, akkor R -en a maximális folyam értéke $|f^*| - |f|$.*

Bizonyítás. Legyen f' tetszőleges folyam R -en és definiáljuk az $f + f'$ folyamot az $(f + f')(v, w) := f(v, w) + f'(v, w)$ egyenlőséggel. Ekkor $f + f'$ folyam G -en, értéke $|f| + |f'| \leq |f^*|$, amiből $|f'| \leq |f^*| - |f|$ adódik. A hasonlóan definiált $f^* - f$, $(f^* - f)(v, w) := f^*(v, w) - f(v, w)$ folyam R -en, értéke $|f^*| - |f|$, azaz az előzőek miatt maximális folyam R -en. $\square$

5.3. Tétel (maximális folyam - minimális vágás). *A következő feltételek ekvivalensek:*

- (i) *Az f folyam maximális.*
- (ii) *Nincs növelő út f -re.*
- (iii) *Valamely $X, \bar{X}$ vágásra az $|f| = c(X, \bar{X})$.*

Bizonyítás. (i) $\Rightarrow$ (ii) Ha van egy p növelő út f -re, akkor növelhetjük a folyam értékét a p -ben lévő éleken megváltoztatva a folyamot. (ii) $\Rightarrow$ (iii) Tegyük fel, hogy f -re nincs növelő út. Legyen X azon pontok halmaza, melyek elérhetők s -ből az R éleit használva, és legyen $\bar{X} = V(G) \setminus X$. ekkor $X, \bar{X}$ egy vágás ($s \in X, t \in \bar{X}$), és

$$|f| = \sum_{v \in X, w \in \bar{X}} f(v, w) = \sum_{v \in X, w \in \bar{X}} c(v, w) = c(X, \bar{X}),$$

mert $v \in X, w \in \bar{X}$ -ből következik, hogy (v, w) nem éle R -nek, azaz $f(v, w) = c(v, w)$. (iii) $\Rightarrow$ (i) Mivel $|f| \leq c(X, \bar{X})$ bármely f folyam és $X, \bar{X}$ vágás esetén, az $|f| = c(X, \bar{X})$ egyenlőségből következik, hogy f maximális folyam és $X, \bar{X}$ minimális vágás. $\square$

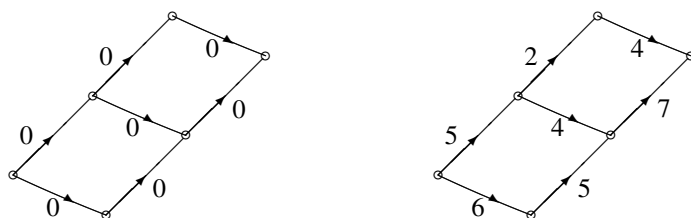
Az előző tétel az alapja Ford és Fulkerson ún. *növelő út módszerének*: Induljunk ki a zérus folyamból ($f(v, w) = 0$ minden (v, w) élre), majd ismételjük meg a *növelő lépést* amíg olyan folyamhoz jutunk, amiben nincs növelő út. *Növelő lépés*: Találjunk egy p növelő utat a pillanatnyi f folyamra. Növeljük f értékét $r(p)$ egységgel a p -ben lévő éleken.

5.4. Tétel (integralitás). *Ha egy folyam problémában a kapacitások egészek, akkor van egy egész értékű maximális folyam.*

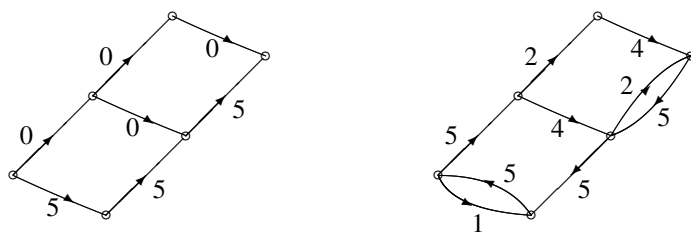
Bizonyítás. Tegyük fel, hogy a kapacitások egészek. Ekkor a növelő út módszer legalább egy egységgel növeli a folyam értékét minden iterációban, így f^* maximális folyamot legfeljebb $|f^*|$ lépésben megkapja. Masrészt a kezdeti egész értékeket minden lépésben egész számmal változtattuk, így $f^*(v, w)$ egész minden (v, w) élre. $\square$

A példánkban az alábbi lépésekkel kaphatunk maximális folyamot. Baloldalon a folyam értékeket, mellette a hozzátartozó maradék gráfot a növelő úttal együtt ábrázoljuk.

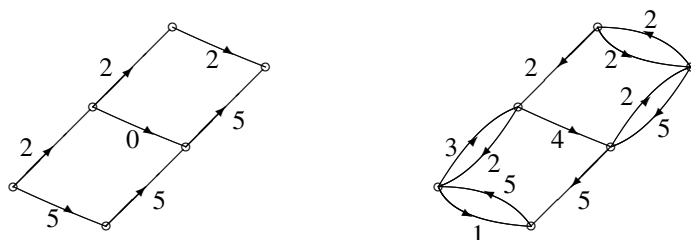
Inicializálás. A növelő út: SF-H-A-NYC, $r(p) = 5$



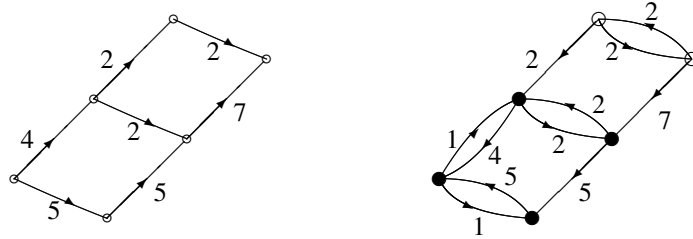
1. lépés. A növelő út: SF-D-C-NYC, $r(p) = 2$



2. lépés. A növelő út: SF-D-A-NYC, $r(p) = 2$

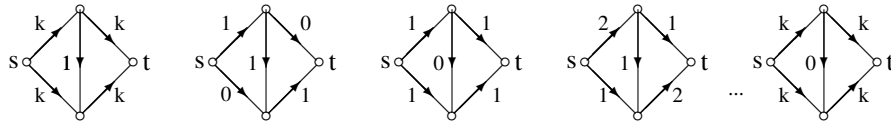


3. lépés. Nincs növelő út, $|f^*| = 9$, X , $\bar{X}$ a teli ill. üres pontok az ábrán.



Megjegyzés: A növelő út módszer nem szükségképpen konvergál, és egész számok esetén is lehet nagyon hosszú. Edmonds és Karp viszont megmutatta 1969-ben, hogy a *legkevesebb élből* álló növelő út választásával az algoritmus cnm^2 műveletet igényel, ahol c konstans, n a pontok, m pedig az élek száma G -ben.

Példa: Az első rajzon a kapacitások, a többin a közelítések, ha a növelő út áthalad a függőleges élen. (Csak a folyamat ábrázoltuk, a maradékgráfot nem.) Ekkor algoritmus lépésszáma k . Ha a legrövidebb növelő utakat választjuk, akkor két lépésben eljutunk az optimumhoz.



5.1. Feladat (baseball elimination). Adott egy bajnokság, ahol minden mérkőzésen van egy nyertes és egy vesztes (azaz nincs döntetlen). Tegyük fel, hogy az i csapat w_i meccset nyert eddig és még r_i mérkőzése van hátra; továbbá az i és j csapatok között még r_{ij} mérkőzés lesz a bajnokság hátralévő részében. Legyen $w_{\max} = w_i + r_i$ a maximális győzelmek száma, melyet i csapat a szezon során elérhet. Adjuk meg, mely csapatnak van a legtöbb győzelemre esélye a szezonban. [Ötlet: vegyük azt a páros gráfot mely egyik osztályában a csapatokat, másikban a lejátszandó mérkőzéseket reprezentáljuk, továbbá egy forrás és egy nyelő pontot. Az éleket és élsúlyokat ügyes módon válasszuk meg.]

5.7. A külszíni fejtés probléma

A maximális folyam probléma egy váratlan és mély alkalmazása fűződik Balinsky, Rhys és Picard nevéhez, amit *külszíni fejtés* (vagy *open pit*) problémának hívnak. Tegyük fel, hogy egy adott

területen valamely mélységig felosztottuk a talajt, és ismerjük egy-egy részben lévő ásvány értékét, illetve az adott rész kitermelési költségét. Az egyszerűség kedvéért téglatest alakú darabokkal számolunk és az i -edik darab értéke w_i a benne levő ásvány értékének és a kitermelési költség különbsége.

Feladatunk azon P halmaz megtalálása, amelyre a $\sum_{i \in P} w_i$ maximális. A nehézség abban rejlik, hogy P nem lehet akármilyen halmaz; meg kell felelnie egy lehetséges kitermelésnek. Ez alatt azt értjük, ha egy téglát kitermelünk, akkor nemcsak a felette lévő téglát, de még annak szomszédait is ki kell termelnünk, még ha negatív is az értékük.²

Példa: Az alábbi ábra egy adott földterület esetén mutatja az ásványok értékének és kitermelési költségének különbségét a föld egyes darabkáin.

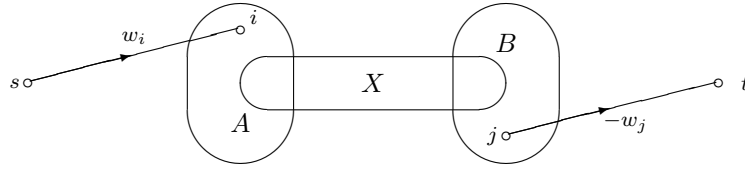
	-1	-1	-1	-1	-1	-1	-1	-1	-1	
	-2	-1	1	5	3	1	-5	4	-1	-1
	1	1	0	10	5	1	1	1	1	0

Általában ha az i -edik elem hozzávétele a P halmazhoz kikényszeríti a j -edik elem hozzávételét azt egy (i, j) irányított éllel jelöljük majd a G gráfban, melynek a pontjai az elemek. Legyen ennek a gráfnak egy $X \subset V(G)$ ponthalmaza *zárt*, ha $i \in X$ és $(i, j) \in E(G)$ esetén $j \in X$. Újrafogalmazva a problémát egy olyan X zárt részhalmazát keressük a pontoknak, hogy a $\sum_{i \in X} w_i$ maximális.

Ezt a következő eljárással visszavezethetjük egy minimális vágás, vagy ami ezzel ekvivalens, egy maximális folyam problémára. Osszuk fel G pontjait két részre; $i \in A$, ha $w_i \geq 0$ illetve $i \in B$, ha $w_i < 0$. Vegyünk hozzá G -hez két új pontot, egy s *forrást* és egy t *nyelőt*, valamint egy (s, i) élt minden $i \in A$ -ra, és egy (j, t) élt minden $j \in B$ -re. legyen az új élek kapacitása $c(s, i) := w_i$ minden $i \in A$ -ra, $c(j, t) := -w_j$ minden $j \in B$ -re, míg G eredeti éleinek kapacitása végtelen.

Könnyen látható, hogy egy X ponthalmaz akkor és csak akkor zárt, ha az $X \cup \{s\}$, $\overline{X \cup \{t\}}$ vágás kapacitása véges.

²Ez bányatechnikai előírás, ugyanis bizonyos szögknél meredekebb lejtőt nem szabad létrehozni külszíni fejtésnél.



Továbbá ha az előbbi vágás kapacitása véges, akkor az éppen

$$\begin{aligned} c(X \cup \{s\}, \overline{X \cup \{t\}}) &= \sum_{i \in A \setminus X} c(s, i) + \sum_{i \in B \cap X} c(i, t) = \\ &= \sum_{i \in A \setminus X} w_i + \sum_{i \in B \cap X} (-w_i) = \sum_{i \in A} w_i - \sum_{i \in X} w_i. \end{aligned}$$

Mivel a $\sum_{i \in A} w_i$ konstans, a fenti vágás kapacitásának minimalizálása a $\sum_{i \in X} w_i$ maximalizálását jelenti. Azaz a minimális vágás megtalálása egyben megadja a keresett maximális súlyú zárt pontthalmazt.

5.2. Feladat. Definiáljuk az Általánosított Cirkulációs problémát (alsó-felső kapacitások és költség egy-egy élre), minimális összköltség a cél.

1. Mondjuk ki (bizonyítsuk) egy integralitás tételt.
2. Vezessük vissza a transshipment és maxflow problémákat erre.
3. Keressünk gyakorlati példákat ÁCP-re.

5.3. Feladat. Azt mondjuk, hogy két (e, e') irányított él anti-párhuzamos, ha végpontjaik ugyanazok, de irányuk különböző. Mutassuk meg, hogy bármely folyamhálózat esetén létezik olyan maximális folyam, hogy e és e' legalább egyikén nem megy át folyam.

5.4. Feladat. Tegyük fel, hogy egy folyamhálózatban több forrás és több nyelő van. Mutassuk meg, hogyan lehet a Ford-Fulkerson algoritmust használni ebben az esetben. Írjuk fel az általános LP modellt is.

5.5. Feladat. Tegyük fel, hogy egy folyamhálózatban minden él kapacitása páros. Mutassuk meg, hogy ekkor a maximális folyam is páros.

5.6. Feladat. Adott egy irányított gráf és két csúcsa s és t . Adjunk meg algoritmust, ami az éldiszjunkt $s - t$ utak számát adja meg.

5.7. Feladat. Adott egy irányított gráf és két csúcsa s és t . Adjunk meg algoritmust, ami megadja minimálisan hány élet kell törölni, hogy t ne legyen elérhető s -ből.

5.8. Feladat. Adott egy irányított gráf és két csúcsa s és t . Adjunk algoritmust, ami megadja a csúcs-diszjunkt $s - t$ utakat.

6. fejezet

Intervallumgráfok és rokonaik

Kezdjük a vizsgálatainkat néhány, könnyen láthatóan rokonságban álló, de nem nyilvánvalóan gráfelméleti problémával. A témában jobban elmélyedni vágyó olvasó számára pl. a [4] könyvet ajánljuk.

6.1. Probléma. (Hűtőszekrény) Egy laboratóriumban el kell tárolni n mintát oly módon, hogy az i -edik t_i hőmérséklet felett és T_i alatt tartsuk, $1 \leq i \leq n$. Kérdés, minimálisan hány hűtőszekrényre van szükség ehhez, ha egy-egy hűtőben egy rögzített hőmérsékletet állíthatunk be?

6.2. Probléma. (Számlla) Folyamatosan kapunk számlákat, amelyeket be kell fizetniünk a lejárat idejük előtt. Egy alkalommal több számlát is rendezhetünk, és minimalizálni kívánjuk az alkalmak számát. Az i -esik számlát t_i időpontban kapjuk, és T_i a befizetés határideje.

6.3. Probléma. (Felügyelet) Egy írásbeli vizsgáztatásnál n csoporthoz kell felügyelőt beosztanunk, szaktudásra nincs szükség, de két helyen egy időben nem lehet a felügyelő egyszerre. Az i -edik csoport t_i időben kezd és T_i -ben fejezi be a vizsgát. Minimálisan hány felügyelőre lesz szükség?

6.1. Intervallumgráfok

Mindhárom feladathoz társítható egy $\mathcal{I} = \{I_1, \dots, I_n\}$ intervallum rendszer, ahol az $I_i = [t_i, T_i] \in \mathbb{R}$. A hűtőszekrény illetve a számla problémákban olyan $K = \{x_1, \dots, x_k\} \subset \mathbb{R}$ halmaz lehet a megoldás, amelyre minden $i \in \{1, \dots, n\}$ esetén $K \cap I_i \neq \emptyset$.¹ Az első esetben K elemei az egyes

¹A K ún. lefedő halmaz $\mathcal{I}$ -re.

hűtők beállított hőmérsékletei és az i -edik minta a $K \cap I_i$ -ből vett hőmérsékletű gépbe kerülhet. A második esetben a K halmaz befizetési időpontokat jelöl, amely az előzőhöz hasonlóan rendelhető számlákhoz. Az optimális megoldásban k minimális.

A harmadik feladat megoldásában számozni (*színezn*i) akarjuk az intervallumokat, formálisan egy $f : \mathcal{I} \rightarrow \{1, \dots, \ell\}$ függvényt keresünk, amelyre $f(I_i) \neq f(I_j)$, ha $I_i \cap I_j \neq \emptyset$. Az optimális megoldásban ℓ minimális.

Az 1. és 2. problémákat nagyon egyszerű *mohó* algoritmussal kezelhetjük. Legyen x_1 legkisebb eleme a $\{T_i\}_{i=1}^n$ halmaznak, azaz a *legbaloldalibb jobb végpont*.² Vegyük az x_1 -gyel metsződő intervallumokat, azaz $\mathcal{I}^1 \subset \mathcal{I}$ és $I_i \in \mathcal{I}^1$ akkor és csak akkor, ha $x_1 \in I_i$ és a lefedett intervallumokat kidobhatjuk. A x_r számot, $2 \leq r \leq k$ esetén a mohó algoritmus meghívásával nyerhetjük a maradék intervallumokra, azaz a $\mathcal{I} \setminus \bigcup_{i=1}^{r-1} \mathcal{I}^i$ rendszerre. Másrésztől, az x_s -eket definiáló intervallumaink, azaz amely miatt x_s bekerült K -ba, diszjunktak, és a lefedésükhöz szükség van minimum k pontra.

A 3. probléma megoldásához célszerű átszámozni az intervallumokat a jobb végpontjuk nagysága szerint, azaz feltehető, hogy $T_1 \geq T_2 \geq \dots \geq T_n$. Az intervallumokat ebben a sorrendben színezzük mohó módon. Kapja az I_i a legkisebb pozitív egész számot, amelyet nem adtunk korábban a I_i -vel metsződő egyik intervallumnak sem. Miért optimális ez a színezés? Tegyük fel, hogy az I_i intervallum az ℓ színt kapta. Ez azt jelenti, hogy az I_i -vel metsződő, már színezett intervallumok közt van $1, 2, \dots, \ell - 1$ színű is. Ezen intervallumok mindegyike tartalmazza a T_i pontot, tehát bármely színezést választanánk is, legalább ℓ színre lenne szükség.

Megjegyzések. Mielőtt gráfokon foglalmaznánk meg az előbbi struktúrákat, szögezzük le, ismét egy szép alkalmazását láttuk a mohó megközelítésnek. A számla problémával kapcsolatban természetes módon felvetődik az *on-line* és *off-line* problémák kérdése. Az *on-line* arra utal, korlátozott az információnk, adott pillanatban kell egy döntést hozni és a már meghozott döntések nem módosíthatóak. Általában ennek ára van, nem érjük el a feladat teljes ismeretében (*off-line* eset) kapható optimumot. Egyfajta árat viszont meg kell fizetnünk a megoldás *instabil* voltában.³ Ha késünk egy befizetéssel (vagy valamelyik hűtőnk nem pontosan tartja az előírt hőmérsékletet) akkor összeomlik a megoldás.

6.1. Feladat. *Mutassuk meg, hogy G intervallumgráf akkor és csak akkor, ha csupa nyílt interval-*

²Egy tipikus bizonyítás jó eséllyel ennek a pontnak a vételével kezdődik.

³Nagyon általános elv, hogy minél jobban optimalizált egy rendszer, annál törékenyebb. A műszaki, gazdasági konstrukciók problémái vagy a biológiai organizmusok túlélése, kihalása is sok példát ad erre.

lummal is reprezentálható.

6.2. Feladat. Mutassuk meg, ha G intervallumgráf akkor van olyan reprezentációja, amelyben az összes intervallum végpont egész szám.

6.3. Feladat. Egy G valódi intervallumgráf, ha a reprezentációjának bármely két, $I_i \neq I_j$ intervallumára teljesül az $I_i \not\subset I_j$. Lássuk be, hogy nem minden intervallumgráf valódi intervallumgráf.

6.4. Feladat. Egy G intervallumgráf időgráf (vagy egység intervallumgráf), ha van olyan reprezentációja, amelyben az összes intervallum egységnyi hosszú. Mutassuk meg, hogy G pontosan akkor időgráf, ha valódi intervallumgráf.

6.2. Metszetgráfok

Kézenfekvő egy $\mathcal{I}$ intervallum rendszerhez egy G egyszerű gráfot rendelni az alábbi módon. $V(G) := \{I_1, \dots, I_n\}$ míg $(I_i, I_j) \in E(G)$ akkor és csak akkor ha $I_i \cap I_j \neq \emptyset$ és $i \neq j$.⁴

Másrészt egy G gráf intervallumgráf, ha van olyan $\mathcal{I}$ rendszer, amely G gráfot reprezentálja. Az intervallumgráfok osztályát $\mathbf{I}$ -vel jelöljük. Általában is érdemes bevezetni az alábbi:

A $H_1, \dots, H_n$ halmazrendszer metszetgráfja az a G gráf, melyre $V(G) = \{H_1, \dots, H_n\}$ és $(H_i, H_j) \in E(G)$ ha $H_i \cap H_j \neq \emptyset$ és $i \neq j$. Fordítva pedig a $H_1, \dots, H_n$ rendszer G gráf egy metszet reprezentációja.

Könnyen látható, hogy bármely egyszerű gráfnak van metszet reprezentációja, így teljes általánosságban a fogalom semmitmondó. Az intervallumgráf egy szerencsés fogalom, amely képes összekötni a folytonos és diszkrét matematika néhány jelenségét. A modell sikere további általánosításokat is sugallt, ahol a halmazok körívek, intervallumok uniói a valós egyenesen vagy épp szorzatai az $\mathbb{R}^k$ -ban, konvex alakzatok stb. Megjegyezzük, hogy az ún. kisvilág gráfok vizsgálatához és adatbányászati modellek készítésében is igen hasznos fogalom a metszetgráf. A feladatok közt kitérünk ezek némelyikére.

A hűtőszekrény illetve a számla problémák gráf feladattá változtatásához szükség van egy észrevételre, tkp. az egy-dimenziós *Helly tétel*: Ha véges, zárt intervallumok egy $\{I_i\}_{i=1}^k$ rendszere

⁴Az intervallumokhoz rendelt gráfok vizsgálatát Hajós György kezdeményezte 1957-ben, majd 1959-ben Seymour Benzer, Nobel-díjas biológus vetett fel ezekkel kapcsolatos kérdéseket.

páronként metsző, akkor $\cap_{i=1}^k H_i \neq \emptyset$.⁵ A bizonyítás egyszerű, vegyük a legbaloldalibb jobb végpontot, az benne van az összes intervallumban.

Mindig igaz, ha x közös pontja egy reprezentációban a $\{H_i\}_{i \in \Gamma}$ halmazoknak, akkor a metszetgráfban a $\{H_i\}_{i \in \Gamma}$ halmazokhoz tartozó pontok klikket alkotnak. Egy G intervallumgráf esetén a Helly tétel miatt ez megfordítható; egy klikk a G -ben egy közös pontot jelent a reprezentációban. Azaz a reprezentáció egy lefoglaló pontthalmaza megfeleltethető a G intervallumgráf egy klikkfedésének és viszont. Speciálisan az algoritmusunk azt is bizonyítja, hogy $\alpha(G) = k(G)$, ha G intervallumgráf.

Még egyszerűbb a dolgunk a 3. probléma értelmezésében. Diszjunkt intervallumok a reprezentációban független pontthalmazt indukálnak a gráfban (és viszont). Mivel a megfelelő módon végrehajtott mohó algoritmus annyi független halmazra bontja $V(G)$ -t, ami a maximuma az egy pontban metsződő intervallumok számának, kapjuk az $\omega(G) = \chi(G)$ egyenlőséget.

6.2.1. Benzer probléma

Az 1950-es évekre általánosan elfogadottá vált az a nézet, hogy az élőlények felépítéséhez szükséges információ a lineáris szerkezetű DNS molekulában tárolódik. Az egyes tulajdonságokat kódoló elemi egység neve gén. Felmerült, hogy egy-egy gén vajon milyen részhalmaza a DNS molekulának? Benzer hipotézise szerint egy gén a DNS molekula egy intervallumának feleltethető meg, megengedve a gének átfedését. Közvetlenül ez nem volt eldönthető, így Benzer egy eredeti ötlettel próbálkozott.

Vett egy vírust (T4 bakteriofág), ennek azonosította néhány tulajdonságát és vizsgálta a mutációit. Feltette, hogy a pontmutációk történtek, és két mérhető tulajdonság változása azt jelenti, az őket kódoló gének közös részében történt változás. Reményei szerint a sok mutáció feltárja az összes kapcsolatot az gének közt, azaz bármely két gén közös részén keletkezik majd felismerhető mutáció.

Ha a hipotézis igaz, a közösen változó tulajdonságokat összekötve intervallumgráf keletkezik. Ahogy a jó játékosnak, a jó kutatónak is szerencséje van, Benzer intervallumgráfot kapott, ami megerősítette a hipotézisét.⁶

⁵Általában is igaz: Ha $\mathbb{R}^n$ -ben adott véges sok véges, zárt, korlátos konvex halmaz úgy, hogy közülük bármely $n + 1$ metszete nem üres, akkor az összes metszete sem üres.

⁶A magasabbrendű élő szervezetek génjei tipikusan diszjunkt intervallumok, a fenti kísérlet üres gráfot adna rájuk. A DNS szerkezetének feltárását feldarabolással, a darabok feltárásával, majd összeillesztésekkel végzik, ahol az intervallumokra vonatkozó ismeretek kamatoztathatók.

Tanulságok. Látható, hogy alkalmazástól függően speciális gráfok, gráfosztályok bukkannak fel. A számunkra fontos problémák ezen gráfok egy-egy gráfparaméterével vannak kapcsolatban. Az elsődleges célunk a *hatékony algoritmusok* kezdeményezése.

Ebben a folyamatban hasznos lehet a gráfok leírása *lokális* tulajdonságaikkal, esetleges tiltott részgráfokkal. Szintén cél a *globális* szerkezet meghatározása; ez mélyebb megértéshez vezet és algoritmikus vonatkozásai is lehetnek.

A megismerés része az esetleges reprezentációk, felbontások vagy különböző gráfosztályok egymáshoz viszonyított kapcsolatainak feltárása. Ezek segíthetnek a modellek készítésében, kiértékelésében illetve új gondolatok felvetésében.

6.5. Feladat. *Mutassuk meg, hogy bármely n -pontú G gráfra van olyan $\mathcal{H}_G = \{H_1, \dots, H_n\}$ halmazrendszer aminek a metszetgráfja G -vel izomorf.*

6.6. Feladat. *Lássuk be, hogy C_k nem intervallumgráf, ha $k \geq 4$. Mutassunk olyan fát, amely nem intervallumgráf.*

6.7. Feladat. *Egy G gráf óragráf, ha a metszetgráfja egy kör ívhalmazának. Mit modellezhetünk óragráfokkal?*

6.8. Feladat. *Egy moziban öt néző van és elképzelhető, hogy egyszer, de nem többször, elalszanak a film közben. Tudjuk, hogy bármely két nézőhöz van olyan időpont, amikor ők ketten biztosan ébren vannak. Mutassuk meg, van olyan időpont, amikor legalább hárman is ébren vannak.*

6.3. Háromszögezett gráfok

A háromszögezett gráfok fogalma és a rájuk vonatkozó első tételek Hajnal András és Surányi János (1958) valamint Claude Berge (1960) munkáiban találhatók.

Egy G gráf *háromszögezett*⁷, ha nem tartalmaz feszített C_n -et, $n \geq 4$ esetén. Mielőtt algoritmusokat keresnénk háromszögezett gráfokra, a szerkezetükről kell információt szerezni. Ebben néhány új fogalom segít.

Egy G gráfban a nem összekötött x, y pontokra az $S \subset V(G)$ halmaz x, y -szeparátor, ha x és y különböző komponensekben van a $G \setminus S$ gráfban. S minimális, ha a valódi része nem x, y -szeparátorok. Egy S *minimális szeparátor*, ha minimális x, y -szeparátor valamely nem összekötött x és y pontokra.

⁷Előfordulnak még a húrozott vagy merevkörű elnevezések is. A háromszögezett gráfok osztályának jele: T .

Egy $x \in V(G)$ *szimpliciális pont* G gráfban, ha az $N(x)$ pontok klikket feszítenek. Az $x_1, x_2, \dots, x_n$ *perfekt eliminációs séma*, röviden PES, G gráfra, ha az x_i szimpliciális pont az $x_i, x_{i+1}, \dots, x_n$ pontok által feszített részgráfban minden $i = 1, \dots, n$ esetén.

6.1. Tétel. *Egy G háromszögezett gráf minden S minimális szeparátora klikket feszít.*

Bizonyítás. Tegyük fel, hogy G háromszögezett és S minimális x, y -szeparátor. Legyen C_x (C_y) az x -et (y -t) tartalmazó komponens $G \setminus S$ feszített részgráfban. S minden pontjából vezet él C_x -be és C_y , különben nem lenne minimális x, y -szeparátor.

Ha $u, v \in S$ és $u \neq v$, akkor van egy legrövidebb, az u és v pontokat leszámítva C_x -ben (illetve C_y -ban) őket összekötő P_x (illetve P_y) út. Vegyük a $P_x \cup P_y$ kört; ez legalább négy hosszúságú és a minimális utak és az S szeparátor miatt az egyedüli potenciális húrja az u, v .

Megfordítva, ha bármely minimális szeparátor klikk, két eset lehet. Amennyiben G a teljes gráf, kész vagyunk. Ha $x, y \in V(G)$, $x \neq y$ és $(x, y) \notin E(G)$, akkor van egy S , amely minimális x, y -szeparátor. Indukcióval kapjuk bármely C komponensre a $G \setminus S$ -re, hogy a $C \cup S$ halmazon feszített részgráf háromszögezett. Így ha egy kör teljes egészében egy $C \cup S$ halmazban marad, akkor azért van húrja, ha két pontot felhasznál S -ből, akkor szintén van húrja. $\square$

6.2. Tétel. *Egy G gráf háromszögezett akkor és csak akkor, ha minden feszített részgrájában van szimpliciális pont.*

Bizonyítás. Ha G -ben van egy C_k kör, $k \geq 4$, akkor ennek nincs szimpliciális pontja. A másik irány kicsit komplikáltabb. Feltehetjük, hogy G nem a teljes gráf. Vegyünk egy olyan $x \in V(G)$ pontot, amelyre a $G \setminus N(x)$ legnagyobb komponense, nevezzük C -nek, a lehető legnagyobb. Legyen $S \subset N(x)$ amelyekből vezet él C -be.

Minden $z \in V(G) \setminus (C \cup S)$ -re $(z, s) \in E(G)$ minden $s \in S$. Ha $z = x$, akkor azért, mert $S \subset N(x)$. Ha $z \in V(G) \setminus (C \cup S \cup \{x\})$, és $s \in S$ úgy, hogy $(z, s) \notin E(G)$, akkor a $C \cup \{s\}$ benne van $G \setminus N(z)$ egy komponensében. Ez a komponens viszont nagyobb lenne, mint C , ellentmondás.

Vegyük most a $G' := G \setminus (S \cup C)$ gráfot, és használjuk az indukciót egy z szimpliciális pont létezéséhez. Legyen $N'(z) := N_{G'}(z)$, amely tehát klikk. A fentiek miatt $N(z) = N'(z) \cup S$. Mivel S maga is egy klikk, illetve minden él létezik S és $N'(z)$ között, $N(z)$ is klikk, z pedig szimpliciális pont. $\square$

6.1. Következmény. *Egy G gráf háromszögezett akkor és csak akkor, ha van perfekt eliminációs sémája.*

Bizonyítás. Ha G háromszögezett, akkor alkalmazhatjuk a 6.2 tételt. Ha az $x_1, x_2, \dots, x_n$ perfekt eliminációs séma, akkor annak egy tetszőleges $x_{i_1}, x_{i_2}, \dots$ részsorozata is az. $\square$

A perfekt eliminációs séma központi adatstruktúra a háromszögezett gráfok vizsgálatában. Könnyen látható $O(nm)$ algoritmus az előállítására; keressünk egy x_1 szimpliciális pontot, majd, $i = 1, \dots, n - 2$ -re egy x_{i+1} egy szimpliciális pontot $G \setminus \{x_1, \dots, x_i\}$ -ben. Vegyük észre, hogy az utolsó pont már külön ellenőrzés nélkül felvehető, hisz üres feltétel vonatkozik a szomszédságára. Úgy is fogalmazhatnánk, *visszafele* is felépíthető egy PES. Ezt kihasználva Leuker (1974), majd Rose és Tarjan (1975) lineáris idejű algoritmust adtak egy PES-ra.

Az általuk kifejlesztett ún. *lexikografikus széltében keresés* az alábbi: Kétféle címkét használunk, $\sigma(i)$ jelenti a PES i -edik elemét míg G minden pontja kap egy halmazcímkét, ami kezdetben $\emptyset$.⁸

Ezután $i = n$ -től visszafele $i = 1$ -ig a következőket hajtjuk végre:

$\sigma(i) := x$, ahol x még nincs PES-ben és x halmazcímkéje maximális⁹, valamint x összes, a PES-be még be nem került szomszédjának halmazcímkéjéhez hozzáadjuk i -t.

6.3.1. A $\omega(G)$, $\chi(G)$, $\alpha(G)$ és $k(G)$ kiszámítása háromszögezett G -re

Feltehetjük, hogy G összefüggő, a komponenseken megoldott problémák összefűzhetőek. Határozzunk meg először egy $\{x_1, \dots, x_n\}$ PES-t G -re.

$\omega(G)$: Vegyük észre, hogy x_1 egy maximális klikknek eleme, ami belőle és a szomszédaiból áll.¹⁰ Rekurzióval lépünk $G \setminus x_1$ -re, $\omega(G) := \max\{|N(x_1)| + 1, \omega(G \setminus x_1)\}$.

$\chi(G)$: A mohó algoritmussal színezzük G -t a PES sorrendjében *visszafele*. A k szín használata egy legalább k -méretű klikk esetén történik, így a mohó algoritmus $\omega(G)$ színnél többet nem igényel (kevesebbel meg nem lehet). Vegyük észre speciálisan $\omega(G) = \chi(G)$ a háromszögezett G -re.

$\alpha(G)$: Az $N(x_1) \cup \{x_1\}$ klikkből legfeljebb egy pontot tartalmazhat egy maximum méretű független halmaz, legyen ez x_1 . Rekurzióval megkaphatunk egy maximális méretű U' független halmazt $G \setminus (N(x_1) \cup \{x_1\})$ -ben. A $U := U' \cup \{x_1\}$ maximum méretű független halmaz G -ben, így $\alpha(G) = |U|$.

⁸Véges $\mathbb{N}$ -beli halmazokat lexikografikusan rendezve először a legnagyobb elemeket hasonlítjuk össze, ha ezek egyenlőek, a 2. legnagyobb elemet s.í.t. Ha A valódi része B -nek, akkor A a nagyobb.

⁹Egyenlőség esetén a maximális halmazcímkéjűek közül tetszőlegesen választunk.

¹⁰Hasonlóan kapható, hogy a maximális klikkek száma G -ben legfeljebb n .

$k(G)$: Az x_1 -et fedő maximális klikk $Q_1 := N(x_1) \cup \{x_1\}$, így rekurzióval a G lehető legjobb klikkfedése Q_1 uniója a $G \setminus (N(x_1) \cup \{x_1\})$ legjobb klikkfedésével. Indukcióval az utóbbi mérete $\alpha(G \setminus (N(x_1) \cup \{x_1\}))$, ami egy $\alpha(G)$ méretű klikkfedést ad G -re. Speciálisan $\alpha(G) = k(G)$ a háromszögezett G -re.

6.3.2. A PES mátrixokra

A lineáris algebra alapfeladata az $Ax = b$ egyenletrendszer megoldása. Sok esetben az A mátrix szimmetrikus, azaz $A = A^t$. Ekkor hozzárendelhető az $n \times n$ -es A -hoz a G_A gráf a következőképpen: $V(G_A) := \{1, \dots, n\}$, $(i, j) \in E(G_A)$, ha $a_{i,j} \neq 0$.

Hasznos megérteni, hogyan változik a mátrixhoz rendelt gráf, ha Gauss eliminációt végzünk. Ha az A mátrix egy $a_{i,i}$ elemét generáló elemnek választva végrehajtjuk az elimináció egy lépését, akkor az i -edik oszlop zéró elemei esetleg nem zéróvá válnak. Ez azzal ekvivalens, hogy az új mátrix (törölve a i -edik oszlopot és sort) gráfja része annak gráfnak, amit G_A -ból az i -edik pont törlésével, és az N_i -ben az összes él behúzásával kapunk. Ezzel beláttuk az alábbi tételt:

6.3. Tétel. *Egy A mátrixra végrehajtható a Gauss elimináció új nem zéró elemek létrehozása nélkül ha a G_A gráf háromszögezett.*

6.9. Feladat. *Mutassuk meg, hogy G gráf háromszögezett akkor és csak akkor, ha $L(G), G$ élgráfja szintén az.*

6.10. Feladat. *Mutassuk meg, hogy a zsaru nyer a zsaru-rabló játékban a G gráfon, ha G háromszögezett.*

6.11. Feladat. *Legyen σ a G háromszögezett gráf egy PES-e, és irányítsuk G éleit, úgy ha $(x, y) \in E(G)$ és x megelőzi y -t a PES-ben, akkor x -ből y -ba megy él, ez $\vec{G}$. Mutassuk meg, hogy (1) $\vec{G}$ irányított kör mentes gráf (2) $\vec{G}$ bármely topologikus rendezése PES G -n.*

6.12. Feladat. *Javasoljunk algoritmusokat, melyek beágyaznak egy tetszőleges H gráfot egy G háromszögezett gráfba úgy, hogy a $e(G) - e(H)$ maradjon kicsi.*¹¹

¹¹ A minimális érték megtalálása NP-nehéz.

6.3.3. Háromszögezett gráfok reprezentációi

A háromszögezett gráfok az intervallumgráfok általánosításai. Kézenfekvő a kérdés, van-e hasonló metszetgráf reprezentációjuk? Bár az intervallumokhoz hasonló reprezentáció bizonyítása lehetséges közvetlenül is, a későbbi általánosítások kedvéért megéri tenni egy látszólagos kitérőt.

Legyen G egy gráf, $(T, \mathcal{S})$ pedig egy pár, ahol T egy fa $\mathcal{S}$ elemei $V(G)$ részhalmazai, melyek rendre T pontjainak felelnek meg. Azaz ha i pont T -ben, akkor és csak akkor, ha $S_i \in \mathcal{S}$. A $(T, \mathcal{S})$ pár *klikkfá* G -re, ha

- (a) $\mathcal{S}$ a maximális klikkek halmaza G -ben,
- (b) minden $x \in V(G)$ esetén ha $x \in S_i \cap S_j$ akkor $x \in S_\ell$ ha ℓ eleme az i -t és j -t összekötő útnak T -ben.

Azaz G egy klikkfájának pontjai G maximális klikkjeit reprezentálják oly módon, hogy egy $x \in V(G)$ pontot tartalmazó maximális klikkek halmaza T egy részfája.

6.4. Tétel. *Egy gráf háromszögezett akkor és csak akkor, ha van klikkfája.*

Bizonyítás. Tegyük fel, hogy a $(T, \mathcal{S})$ a G gráf egy klikkfája. Legyen $\ell \in T$ egy levél T -ben, és S_ℓ az ℓ ponthoz tartozó maximális klikk G -ben. Ekkor van egy olyan $z \in S_\ell$ pont, mely szimpliális G -ben. Vegyük ℓ egyetlen T -beli szomszédját, p -t. S_ℓ és S_p maximális klikkek G -ben, így $S_\ell \setminus S_p \neq \emptyset$, azaz van egy $z \in S_\ell \setminus S_p$. A z -t tartalmazó maximális klikkek részfat alkotnak T -ben, de ez csak a S_ℓ pont maga lehet, így z szimpliális G -ben. Innen indukcióval jön tétel csak akkor irányja.

A másik irányban feltesszük, hogy G háromszögezett és nem a teljes gráf (mely eset triviális). Vegyünk egy lehető legkisebb méretű S minimális szeparátort. Legyenek $C_1, \dots, C_t$ a $G \setminus S$ komponensei, $t \geq 2$, hisz S minimális szeparátor. Ekkor S minden pontjának van szomszédja minden komponensben. Tegyük fel, hogy egy $s \in S$ pontnak nincs pl. C_1 -ben szomszédja. Ekkor viszont $S' := S \setminus s$ egy x, y -szeparátor tetszőleges $x \in C_1$ illetve $y \in C_2 \cup \dots \cup C_t$ esetén, ami ellentmond S definíciójának.

A G gráf klikkfáját a $C_i \cup S$ részgráfokban rekurzióval nyert T_i klikk fákba rakhatjuk össze. Vegyük azt a M_i maximális klikket $C_i \cup S$ amely tartalmazza S -et. Könnyen látható, hogy $M_i \setminus S \neq \emptyset$. (Legyen $x_i \in C_i$ az S -ben legtöbb szomszédal rendelkező pont. Ha $S \not\subset N(x_i)$, azaz $s \in S \setminus N(x_i)$, akkor van olyan y_i , mely C_i -ben minimális távra van x_i -től és $(s, y_i) \in E(G)$).

Ekkor a háromszögezethez miatt $(s', y_i) \in E(G)$ minden $s' \in N(x_i) \cap S$ pontra. Vegyük észre, hogy M_i egyértelműen meghatározott.)

Ezek után a T fát úgy képezzük, hogy $i = 2, \dots, t$ -re a T_i fák M_i -nek megfelelő pontját összekötjük a T_1 fa M_1 -nek megfelelő pontjával. A T fa G klikkfája, ehhez az (a) és (b) feltételek teljesülését kell ellenőriznünk. Egy klikknek nem lehet eleme C_i -ben és C_j -ben $i \neq j$ egyszerre, mert S szeparátor, amiből jön az (a) feltétel. Ha $x \in C_i$, akkor minden x -et tartalmazó klikk G_i -ben van, így az indukciós feltétel szerint ezen klikkek részfat alkotnak T_i -ben, így T -ben is. Ha $x \in S$, akkor az x pontosan az M_i , $i = 1, \dots, t$ maximális klikk eleme G -ben. A T felépítése miatt ezeknek a klikkeknek éppen egy csillag T -ben (az M_1 -nek megfelelő pont a középpontja, a többi M_i -nek megfelelő pontok a levelei). $\square$

Egy G gráf metszetgráf reprezentációja T fareprezentáció, ha G egy x pontjához a T egy T_x részfája tartozik. Továbbá az $x \neq y$ esetén $T_x \cap T_y \neq \emptyset$ pontosan akkor, ha $(x, y) \in E(G)$.

6.5. Tétel. *Egy gráf háromszögezt akkor és csak akkor, ha van egy fareprezentációja valamely T fával.*

Bizonyítás. Ha G háromszögezt, akkor a 6.4 tétel miatt van egy (T, S) klikkfája. Rendeljük az $x \in V(G)$ ponthoz T_x részfat, ami a x -et tartalmazó maximális klikkekhez tartozó pontokon feszül T -ben. (Ez a klikkfa (b) tulajdonsága miatt összefüggő.)

Ha $(x, y) \in E(G)$, akkor van olyan Q maximális klikk G -ben amelynek része. Így a Q -hoz rendelt $q \in V(T)$ -re kapjuk, hogy $q \in T_x \cap T_y \neq \emptyset$. Másrészt ha egy q -ra $q \in T_x \cap T_y$, akkor a q által reprezentált G -beli klikk tartalmazza az $x, y \in V(G)$ pontokat, azaz $(x, y) \in E(G)$.

A másik irányhoz tegyük fel, hogy egy T fa $\{T_x\}_{x \in V(G)}$ részfái metszet reprezentációja G -nek. Törölhetünk T -ben bármely olyan ℓ levelet, amelyik (1) nem része egy T_x -nek sem, vagy (2) az összes ℓ -et tartalmazó T_x tartalmazza p -t, ℓ egyetlen szomszédját T -ben. Így vagy $|V(T)| = 1$ és egy teljes gráf G , vagy van olyan T_x részfa, amely éppen egy ℓ levélből áll. Így viszont minden T_y részfa, amelyre $T_x \cap T_y \neq \emptyset$ kapjuk, hogy $\ell \in T_y$, azaz x szomszédai G -ben klikket alkotnak. Azaz az $x \in V(G)$ pont szimpliciális G -ben. Megismételve az eljárást $T \setminus \{\ell\}$ -re indukcióval a G perfekt eliminációs sémáját kapjuk, így a 6.1 következményt alkalmazva kész vagyunk. $\square$

6.13. Feladat. *Ha a G gráfnak van egy T fareprezentációja, akkor van olyan T' is, ahol a részfák egyike sem valódi része a másinak.*

6.14. Feladat. *Egy háromszögezt G gráfnak van olyan T fareprezentációja, amire $v(T) \leq v(G)$.*

6.3.4. Splitgráfok

A háromszögezett gráfok egy önmagában is érdekes részosztálya az ún. *splitgráfok*. Egy G gráf *splitgráf*, ha $V(G)$ felbontható A és B halmazokra úgy, hogy A klikk B pedig független halmaz G -ben.¹²

Egy karakterizációját a splitgráfoknak Földes István és Hammer Péter adta 1977-ben:

6.6. Tétel. *Egy G gráfra az alábbiak ekvivalensek:*

1. G split gráf
2. G és $\overline{G}$ háromszögezett gráfok
3. G gráf C_4 , $2K_2$ és C_5 -mentes.

Bizonyítás. (1) $\Rightarrow$ (2) Tegyük fel, hogy G split a (A, B) felbontással. Ekkor bármely feszített C kör legfeljebb két pontja lehet A -ban. Ha $|A \cap C| = 1$, C csak a C_3 lehet. Ha $|A \cap C| = 2$, a közös pontok szomszédosak C -ben, így C újra csak a C_3 lehet. Mivel $\overline{G}$ split gráf a (B, A) felbontással, $\overline{G}$ szintén háromszögezett.

(2) $\Rightarrow$ (3) Nyilvánvaló.

(3) $\Rightarrow$ (1) Legyen K olyan maximális klikk G -ben, amelyre a $G \setminus K$ részgráf élszáma minimális. Belátjuk, hogy $G \setminus K$ -ban nincs él, így a $A := K$, $B := V(G) \setminus K$ egy split felbontás. Tegyük fel, hogy (x, y) él a $G \setminus K$ részgráfban. Ekkor egyetlen $z \in G \setminus K$ pont sem lehet szomszédja K összes pontjának (különben $K' := K \cup \{z\}$, és K nem maximális). Az sem lehet, hogy x és y csak egyetlen (közös) $z \in K$ ponttal ne legyenek szomszédok, hisz a $K'' := (K \setminus \{z\}) \cup \{x, y\}$ egy a K -nál nagyobb klikk lenne. Azaz léteznek $u, v \in K$ különböző pontok, melyekre $(x, u) \notin E(G)$ és $(y, v) \notin E(G)$. Az (x, v) és (y, u) élek közül *pontosan* az egyik létezik, ha mindkettő lenne C_4 -gyel, ha egyik sem, akkor $2K_2$ -vel izomorf gráfot feszítene az $\{x, y, u, v\}$ halmaz. Feltehető, hogy $(x, v) \notin E(G)$ és $(y, u) \in E(G)$. Minden $w \in K \setminus \{u, v\}$ -re ha (y, w) és (x, w) egyike sem él, akkor az $\{x, y, v, w\}$ pontokon $2K_2$, míg ha $(y, w) \notin E(G)$ és $(x, w) \in E(G)$, akkor C_4 feszül. Azaz y valójában össze kell legyen kötve minden ponttal a $K \setminus \{v\}$ -ből, és így a $K' := (K \setminus \{v\}) \cup y$ maximális klikk G -ben.

K választása miatt a $G \setminus K'$ részgráfnak nem lehet kevesebb éle, mint a $G \setminus K$ -nak, így kell lennie olyan $t \in G \setminus (K \cup \{y\})$ pontnak, amelyre $(t, v) \in E(G)$ de $(t, y) \notin E(G)$. A $(t, x) \in E(G)$,

¹²A jelük **S**.

különben a $\{t, x, y, v\}$ egy $2K_2$ -t feszítene, míg a $(t, u) \notin E(G)$, mert az meg C_4 -et eredményezne a $\{t, x, y, u\}$ halmazon. Így viszont a $\{t, x, y, u, v\}$ halmazon lenne egy C_5 , azaz a $(x, y) \in E(G)$ ellentmondásra vezet. $\square$

6.15. Feladat. Egy G gráfban van két különböző, nem szükségképpen diszjunkt C_5 . Ekkor G tartalmaz $2K_2$ vagy C_4 feszített részgráfokat is.

6.16. Feladat. Egy G gráf $2K_2$ és C_4 -mentes. Mi mondhatunk a szerkezetéről?

6.3.5. Threshold gráfok

A splitgráfok egy nagyon tanulságos speciális esetét tanulmányozta Václáv Chvátal és Hammer Péter 1973-ban.¹³

Egy G gráf *threshold* (vagy *küszöb*) gráf, ha a pontjaihoz rendelhetők $x_1, \dots, x_n$ nem negatív valós számok és egy $t \in \mathbb{R}$ úgy, hogy $(i, j) \in E(G)$ pontosan, ha $x_i + x_j \leq t$.

6.7. Tétel. Egy G gráfra az alábbiak ekvivalensek:

1. G threshold gráf
2. G gráf $C_4, 2K_2$ és P_4 -mentes
3. G splitgráf és $\overline{N}(1) \supseteq \dots \supseteq \overline{N}(n)$.

Bizonyítás. (1) $\Rightarrow$ (2) Ha a (2) feltétel valamelyik gráfja része G -nek, feltehetjük, hogy az első négy ponton; az ezekhez rendelt számok $x_1 \leq x_2 \leq x_3 \leq x_4$. Az (1) szerint $\overline{N}(1) \supseteq \dots \supseteq \overline{N}(4)$, ami nem teljesül $C_4, 2K_2$ és P_4 gráfok egyikére sem.

(2) $\Rightarrow$ (3) A $C_4, 2K_2$ és P_4 -mentesség a 6.6 tétel szerint G splitgráf. Tegyük fel, hogy $|\overline{N}(i) \cap \overline{N}(j)| < \min\{|\overline{N}(i)|, |\overline{N}(j)|\}$, azaz van olyan k, ℓ , hogy $k \in \overline{N}(i) \setminus \overline{N}(j)$ és $\ell \in \overline{N}(j) \setminus \overline{N}(i)$. Ha a maradék lehetséges élek, (i, j) és (k, ℓ) közül egyik sem létezik, akkor $2K_2$, ha pontosan az egyik, akkor P_4 , ha mindkettő, akkor pedig C_4 feszül az $\{i, j, k, \ell\}$ ponthalmazon.

(3) $\Rightarrow$ (1) Ha $x_1, \dots, x_n$ és t a G -t reprezentáló súlyok, akkor $ax_1 + r, \dots, ax_n + r$ és $at + r/2$ is az, ha $r \geq 0, a > 0$. Tegyük fel, hogy $x_2 < \dots < x_n$ és t értékeket már meghatároztuk továbbá $(1, i) \in E(G)$, de $(1, i+1) \notin E(G)$. Legyen $x_1 \in (t - x_i, t - x_{i+1})$ intervallumban, majd szükség szerint skálázzuk át az $x_1, \dots, x_n$ és t értékeket. $\square$

¹³Ekkor persze a splitgráfokat nem ismerték még.

6.17. Feladat. Egy T fa mikor lesz gráf *threshold* gráf?

6.18. Feladat. Mutassuk meg, hogy egy G gráf *threshold* gráf akkor és csak akkor, ha minden H feszített részgráffára $\chi(H) + k(H) \geq v(H) + 1$.

6.19. Feladat. Egy G gráf $2K_2$, C_4 , P_4 és K_3 -mentes. Mi mondhatunk a szerkezetéről?

6.4. Összehasonlítási gráfok

Egy $\vec{G}$ gráf irányított gráf irányítása *tranzitív*, ha az xy és yz irányított élek jelenlétéből következik, hogy xz is irányított él. Egy G gráf *összehasonlítási gráf*¹⁴ ha van tranzitív irányítása.

Egy összehasonlítási gráf felfogható, mint egy véges $(X, \leq)$ parciálisan rendezett halmaz, ún. *poset*, egy lehetséges reprezentációja. Míg szokásos Hasse diagramm csak azokat az xz éleket tartalmazza, amelyekre nincs olyan $y \notin \{x, z\}$, hogy $x \leq y \leq z$, itt jelen van az összes xy , amelyre $x \leq y$.

6.8. Tétel. Ha G egy összehasonlítási gráf, akkor $\omega(G) = \chi(G)$.

Bizonyítás. Az $\omega(G) \leq \chi(G)$ miatt elég egy $\omega(G)$ szint felhasználó jó színezését megadni G -nek. Egy $x \in V(G)$ pontra legyen $f(x)$ az x -ből induló legnagyobb irányított út pontjainak a száma. Ha $(x, y) \in V(G)$, akkor xy vagy yx irányított él $\vec{G}$ -ben. Ha xy , akkor $f(x) \geq f(y) + 1$, azaz az f jó színezés. Másrészt ha $f(x) = k$, akkor x benne van egy k méretű klikkben, azaz $\omega(G) \geq \max_{x \in V(G)} f(x) \geq \chi(G)$. $\square$

Egy $(X, \leq)$ parciálisan rendezett halmazban egy lineárisan rendezett $L \subset X$ részhalmaz *láncc*, egy páronként nem összehasonlítható elemekből álló $A \subset X$ részhalmaz pedig *antilánc*.

Ha antilánccok uniójára bontunk egy parciálisan rendezett halmazt, akkor egy lánc elemeinek csupa különböző antiláncba kell tartoznia. Mivel az antilánc független halmaz, egy lánc pedig klikk a poset-hez tartozó gráfban, a 6.8 tétel Leon Mirsky egy tételét adja:

6.2. Következmény (Mirsky 1971). Egy $(X, \leq)$ véges, parciálisan rendezett halmaz felbontható annyi antilánccra, mint a legnagyobb lánc mérete.

Később belátjuk, hogy $\alpha(G) = k(G)$, más szóval $\omega(\overline{G}) = \chi(\overline{G})$. Ennek következménye Robert Dilworth egy klasszikus eredménye.

¹⁴Eredetiben comparability graphs, így az osztály jele C .

6.3. Következmény (Dilworth 1950). Egy $(X, \leq)$ véges, parciálisan rendezett halmaz felbontható annyi láncra, mint a legnagyobb antilánc mérete.

Az összehasonlítási gráfok leírhatók tiltott részgráfok segítségével

6.9. Tétel (Gilmore-Hoffman 1964, Ghouilla-Houri 1962). A G gráf összehasonlítási gráf akkor és csak akkor, ha bármely $(x_1, x_2), (x_2, x_3), \dots, (x_k, x_1) \in E(G)$ esetén, ha (x_1, x_{k-1}) , illetve (x_{i-1}, x_{i+1}) ($i = 2, \dots, k-1$) egyike sem él G gráfban, akkor k páros.

6.20. Feladat. G gráfra $V(G) := \{x_i, y_i : i = 1, 2, 3\}$, az x_1, x_2, x_3 pontok háromszöget alkotnak, ezen kívül $(x_i, y_i) \in E(G)$, $i = 1, 2, 3$. Összehasonlítási gráf G ?

6.4.1. Permutációgráfok

Egy π permutáció bijekció a $\{1, \dots, n\}$ -ből önmagára, $\pi(i)$ jelöli az i képét. A π permutációhoz hozzárendeljük a $G[\pi] = (V, E)$ gráfot, ahol $V := \{1, \dots, n\}$ és $E := \{(i, j) \mid (i-j)(\pi(i)-\pi(j)) < 0\}$.

A G gráf permutációgráf, ha van olyan π permutáció, amelyre G izomorf a $G[\pi]$ gráffal.

A permutációgráfoknak van természetes metszetgráf reprezentációja, ezt permutáció diagramnak nevezik. Vegyünk két párhuzamos egyenest és az egyiken az $1, 2, \dots, n$ pontokat jelöljük be ebben a sorrendben, míg a másikon a $\pi(1), \pi(2), \dots, \pi(n)$ sorrendben. Az i ponthoz tartozó halmaz a $[i, \pi(i)]$ síkbeli szakasz.

6.10. Tétel (Pnuelli, Lempel, Evan 1971). A G gráf permutációgráf akkor és csak akkor, ha G és $\overline{G}$ összehasonlítási gráfok.

Bizonyítás. Tegyük fel, hogy G permutációgráf így van egy permutáció diagramja. Először belátjuk, hogy $\overline{G}$ összehasonlítási gráf. Ha $(i, j) \notin E(G)$, akkor a permutáció diagramban nem metsző szakaszok tartoznak hozzájuk. A $\overline{G}$ gráfban irányítsuk az (i, j) élt ij -nek, ha $i < j$. Ez tranzitív irányítás lesz $\overline{G}$ -ben, azaz $\overline{G}$ összehasonlítási gráf. Könnyen látható, hogy egy permutációgráf komplementere is permutációgráf, így G maga is összehasonlítási gráf.

Ha G és $\overline{G}$ összehasonlítási gráf, elég megadnunk egy permutáció diagramját G gráfnak, abból következik, hogy G permutációgráf is egyben. Legyen O_1 illetve O_2 a G illetve $\overline{G}$ éleinek tranzitív rendezése, O_1^{-1} illetve O_2^{-1} pedig ezen rendezések megfordításai. Mind az $O_1 \cup O_2$ mind az $O_1^{-1} \cup O_2$ rendezések irányított kör mentesek (lásd feladat), így a $\{1, 2, \dots, n\}$ alaphalmaz lineáris

rendezéseit adják, legyen ezek π és ρ . Vegyünk fel egy permutáció diagramot úgy, hogy az egyik egyenesen π szerint, a másikon pedig a ρ szerint ábrázoljuk a pontokat. $\square$

6.21. Feladat. Egy G gráf permutációgráf akkor és csak akkor, ha $\overline{G}$ gráf is az.

6.22. Feladat. Lássuk be, hogy a 6.10 tételben említett $O_1 \cup O_2$ és $O_1^{-1} \cup O_2$ rendezések irányított kör mentesek.

6.23. Feladat. Lássuk be, hogy a 6.10 tételben említett π, ρ sorrendekkel G egy permutáció diagramját kapjuk.

6.24. Feladat. Lássuk be, $\max\{\omega(G), \alpha(G)\} \geq \sqrt{n}$, ha G egy n -pontú permutációgráf. (Erdős-Szekeres, 1935)

6.25. Feladat. Mutassuk meg, ha G gráf P_4 -mentes, akkor G és $\overline{G}$ gráf is összehasonlítási gráf.

6.4.2. P_4 -mentes gráfok

A 6.25 feladat szerint a feszített P_4 mentes gráfok osztálya része az összehasonlítási gráfok, sőt a permutációgráfok osztályának is. A feszített P_4 -ek szerkezet egy gráfon belül rengeteg információt hordoz¹⁵ így önmagában is érdekes, mikor egyáltalán nincs P_4 egy gráfban. A feszített P_4 mentes gráfokat nevezik kográfnak (cograph) is.

6.11. Tétel. A G gráf kográf akkor és csak akkor, ha minden H feszített részgráffjára G -nek az alábbiak közül pontosan egy igaz:

1. H egy pontból áll,
2. H nem összefüggő,
3. $\overline{H}$ nem összefüggő.

Bizonyítás. Egyrészt ha G kográf, akkor $\overline{G}$, és minden feszített részgráffjuk is az, elég látni, hogy G -re igaz az állítás. Ha G pontszáma kisebb, mint négy, G csak kográf lehet és ezekre könnyen ellenőrizhető az állítás. Feltehető, hogy G legalább négy pontú kográf, és minden H feszített részgráffjára H vagy $\overline{H}$ nem összefüggő.

¹⁵Lásd később az ún. félerős perfekt gráf tételt.

Azaz bármely $x \in V(G)$ pontra $G \setminus x$ vagy $\overline{G} \setminus x$ nem összefüggő. Az első esetben a $G \setminus x$ komponensei $C_1, \dots, C_t$, $t > 1$. Ha vannak olyan $u, v \in C_i$ pontok, hogy $(u, x) \in E(G)$, $(v, x) \in E(G)$ és $w \in C_j$, $(w, x) \in E(G)$, $i \neq j$, akkor a $\{w, x, u, v\}$ halmaz P_4 -et feszít.

Csak az marad, hogy minden komponens minden pontja össze lenne kötve x -szel, amiből következik a $\overline{G}$ nem összefüggősége. A másik eset, mikor $\overline{G} \setminus x$ nem összefüggő, teljesen hasonlóan megy.

Végül ha G gráf nem kográf, akkor van benne egy P_4 feszített részgráf, amire nem teljesül egyik pont sem. $\square$

A G kográf dekompozíciós fája¹⁶ egy (T, f) pár, ahol T egy bináris fa, f pedig egy bijekció T levelei és G pontjai közt. A belső pontokat $\otimes$ vagy $\oplus$ jelekkel címkézzük. Legyenek $V_t(\ell)$ illetve $V_t(r)$ a t belső pontból induló bal illetve jobb részfa leveleinek megfelelő pontok. Ha t címkéje $\otimes$, akkor minden él létezik $V_t(\ell)$ és $V_t(r)$ között, míg ha a címke $\oplus$, akkor egy sem.

A 6.11 tétel alapján könnyen megkaphatunk egy dekompozíciós fát egy G kográfra. Ha G nem összefüggő, akkor a komponenseit $\oplus$ címkéjű pontokkal felfűzhetjük, és rekurziót használunk. Ha G összefüggő, akkor $\overline{G}$ nem összefüggő, és $\overline{G}$ komponensei szerint kapcsolhatjuk össze a részfákat $\otimes$ címkéjű pontokat felvéve.

6.26. Feladat. Adjunk egy algoritmust, amely a G kográf dekompozíciós fáját felhasználva határozza meg $\omega(G)$.

6.27. Feladat (Sumner). Egy G gráf P_5 és K_3 -mentes. Mi mondhatunk a szerkezetéről? Mi lehet $\chi(G)$ értéke?

6.5. Perfekt gráfok

Sok példát láttunk arra, hogy a $\omega = \chi$ egyenlőség igaz lehet számos fontos gráfra, és ez igen hasznos algoritmikus szempontból. Természetes hát önmagában is vizsgálni ezt a kérdést:

Egy G gráf *perfekt* [Berge, 1961], ha G minden feszített H részgráfjára teljesül $\omega(H) = \chi(H)$.

Berge megfogalmazott néhány sejtés a perfekt gráfokra vonatkozóan, melyek közül a legismertebbek az ún. gyenge és erős perfekt gráf sejtések.¹⁷

¹⁶Másnéven *cotree*, melyre nem találtunk jó magyar elnevezést.

¹⁷Weak Perfect Graph Conjecture, röviden WPGC illetve Strong Perfect Graph Conjecture, SPGC.

6.1. Sejtés (WPGC). *Ha G gráf perfekt, akkor $\overline{G}$ is perfekt.*

6.2. Sejtés (SPGC). *G gráf perfekt akkor és csak akkor, ha nem tartalmaz feszített feszített részgráfként C_{2k+1} és $\overline{C}_{2k+1}$ gráfokat, ahol $k \geq 2$.*

Egy évtizeddel később Lovász László bebizonyította az első állítást, majd őt követve Ray Fulkerson egy másik technikával szintén eljutott ide. Mi a Lovász által adott ekvivalens formát és Gasparian 1996-ban publikált ötletét mutatjuk be:

6.12. Tétel (Lovász 1972). *A G gráf perfekt akkor és csak akkor, ha G minden feszített H részgrájára $\omega(H)\alpha(H) \geq v(H)$. Speciálisan, G perfekt akkor és csak akkor, ha $\overline{G}$ is az.*

Bizonyítás. Minden G gráfra igaz, hogy $\chi(G)\chi(\overline{G}) \geq v(G)$. Ha G perfekt, akkor minden H feszített részgrájára $\chi(H) = \omega(H)$ illetve $\chi(\overline{H}) = \alpha(H)$, amiből $\omega(H)\alpha(H) \geq v(H)$.

Ha egy gráf nem perfekt, akkor van minimális nem perfekt feszített részgráfja. Legyen G egy minimális nem perfekt gráf, elég belátnunk, hogy $v(G) = \omega(G)\alpha(G) + 1$. (Azaz a perfektség tagadása tagadja az $v(G) \leq \omega(G)\alpha(G)$ egyenlőtlenséget.) A rövidség kedvéért legyen $\omega := \omega(G)$, $\alpha := \alpha(G)$, $n := v(G)$. Minden S független halmazra és $x \in V(G)$ pontra az alábbiak igazak:

1. $\omega(G \setminus S) = \omega$
2. $\chi(G - v) = \omega = \omega(G - v)$
3. Ha $X_1, \dots, X_\omega$ a $G - v$ egy ω színezésének színosztályai, és K egy ω méretű klikk G -ben, akkor két eset lehetséges:
 - (i) $v \notin K$ és $K \cap X_i \neq \emptyset$ minden i -re, vagy
 - (ii) $v \in K$ és $K \cap X_i = \emptyset$ pontosan egy i -re.

Legyen S_0 egy α méretű független halmaz G -ben, $S_1, \dots, S_{\alpha\omega}$ rendre a színosztályok egy-egy ω színezésében a $G - s$ -nek, ahol s befutja S_0 -t. (Azaz $S_1, \dots, S_\omega$ a $G - s_1$ -hez, $S_{\omega+1}, \dots, S_{2\omega}$ a $G - s_2$ -höz és így tovább színosztályok.) Legyen továbbá $Q_1, \dots, Q_{\alpha\omega}$ rendre ω méretű klikkek a $G \setminus S_0$ -ból, mely az (1) által definiált.

Szükségünk lesz két $(\alpha\omega + 1) \times (\alpha\omega + 1)$ mátrixra, J minden eleme 1, míg I a szokásos egységmátrix. Legyen A az a $(\alpha\omega + 1) \times n$ mátrix, amelynek az i -edik sora az S_i halmaz incidencia vektora, míg B egy $n \times (\alpha\omega + 1)$ mátrix, melynek az i -edik oszlopa a Q_i halmaz incidencia vektora.

Ha K egy tetszőleges ω méretű klikk, akkor pontosan egy S_i -től diszjunkt, ugyanis

(i) $K \cap S_0 = \emptyset$ és $K \cap S_i \neq \emptyset, i \geq 1$ a (3) miatt, vagy

(ii) $K \cap S_0 \neq \emptyset$ és $K \cap S_i = \emptyset$, pontosan egy i -re.

A $Q_i \cap S_i = \emptyset$ és $Q_i \cap S_j \neq \emptyset$, ha $i \neq j$ az $AB = J - I$ mátrix egyenletben foglalható össze.

A $G - x$ gráf ω független halmazra bontható, azaz $n \leq \omega\alpha + 1$. A fordított egyenlőtlenség az $AB = J - I$ -ből a rangszám tétel miatt $n \geq rk(A) \geq rk(J - I) = \alpha\omega + 1$ jön. $\square$

Az ember azt hihetné a fentiek alapján, olyan sokat tudunk a minimális nem perfekt gráfokról, hogy nem lehet túl nehéz belátni, csak a C_{2k+1} ill. $\overline{C}_{2k+1}, k > 1$ lehetnek ezek. Sajnos ez messze nem így van, az erős perfekt gráf sejtés bizonyítása nem egyszerű és nem is ezt az utat követte.

Egy G gráf *Berge gráf*, ha G nem tartalmaz feszített C_{2k+1} vagy $\overline{C}_{2k+1}, k > 1$ gráfokat.

Egy $Y \subset V(G)$ *antikomponens*, ha Y komponens $\overline{G}$ -ben. Hasonlóan az antiél (antiút) élt (utat) jelent a komplementer gráfban. A $V(G)$ egy (X, Y) partíciója (X, Y üres is lehet) *jó partíció*, ha

(i) $G[X]$ minden komponense és $G[Y]$ minden antikomponense legfeljebb kételemű

(ii) ha C_X komponens $G[X]$ -ben, C_Y antikomponens $G[Y]$ -ban, és $v \in C_X \cup C_Y$, akkor legfeljebb egy él és egy antiél létezik C_X és C_Y között, amely v -re illeszkedik.

Egy G gráf *duplázott gráf*, ha van jó partíciója.

Egy G gráf *alapgráf*, ha

1. G páros gráf,
2. $\overline{G}$ páros gráf,
3. G egy páros gráf élgráfja,
4. G egy páros gráf élgráfjának komplementere,
5. G duplázott gráf.

Belátható, hogy az alapgráfok perfektek, lásd a feladatokban.

6.13. Tétel (Chudnovsky, Robertson, Seymour, Thomas 2006). *Egy G Berge gráf felépíthető alapgráfokból olyan kapcsolásokkal, melyek perfekt gráfokat perfekt gráfokba visznek. Speciálisan, G perfekt akkor és csak akkor, ha G Berge gráf.*

Az 6.13 tétel első bizonyítása meglehetősen nehéz, és a későbbi egyszerűsített formája is meghaladja a rendelkezésünkre álló keretet. Megjegyezzük, hogy a felhasznált eszközökkel belátható az is, hogy hatékony (polinom időben) tesztelhető, hogy egy G gráf perfekt-e, illetve ha az, akkor megadható például $\omega(G)$.

6.28. Feladat. *Lássuk be, ha G páros gráf, akkor $\alpha(G) = k(G)$. Más szóval G páros gráf komplementere perfekt.*

6.29. Feladat. *Jelölje $\chi'(G)$ a G gráf kromatikus indexét, azaz $\chi'(G) := \chi(L(G))$. Lássuk be, ha G páros gráf, akkor $\chi'(G) = \Delta(G)$, ahol $\Delta(G)$ a maximális fokszám G -ben. Mutassuk meg, hogy ekkor $L(G)$ perfekt.*

6.30. Feladat. *Jelölje $\nu(G)$ a maximum párosítás míg $\tau(G)$ a minimális számát a pontokat fedő éleknek egy G gráfban. Lássuk be, hogy egy K_3 -mentes G gráfra $\nu(G) = k(G) + \nu(G) = \alpha(G) + \tau(G)$.*

6.31. Feladat. *Lássuk be a 6.30 feladatot használva, hogy egy G páros gráf élgráfjának komplementere perfekt.*

7. fejezet

Exponenciális algoritmusok

Tudjuk, hogy ha egy algoritmus futási idejére exponenciális (alsó) korlát van, akkor általános (vagy legrosszabb) esetben reménytelen lehet az adott feladat megoldása. Ugyanakkor jelentős különbség lehet egy-egy algoritmus legrosszabb-eset futási ideje és a valós példákon való futási ideje között. Gyakorlati szempontból érdemes fontos gráfalgoritmusok további fejlesztése, ugyanis tapasztalati tény, hogy a gyorsabb számítógép nem képes annyit javítani az algoritmusok futási idején, mint magának az algoritmusnak a javítása.¹ Az egzakt megoldást adó algoritmusok mellett másik megközelítés az approximációs algoritmusok fejlesztése exponenciális feladatokra. A fejezetben csak néhány fontos példát mutatunk be.

7.1. Hálózati megbízhatóság

Egy korábban tárgyalt modellhez hasonlóan adott egy irányítatlan G gráf az s és t kitüntetett pontokkal, továbbá G éleihez valószínűségeket rendelünk. Értelmezésünk szerint egy élhez rendelt szám az él járhatóságának (tkp. *létezésének*) a valószínűsége, és az élek létezései független események. Láttuk, hogy ekkor például a legnagyobb valószínűséggel létező $s - t$ út keresése a legrövidebb út problémára vezethető vissza. Sokkal nehezebb probléma kiszámolni azt, milyen valószínűséggel juthatunk el s -ből t -be. Ez olyannyira nehéz, hogy hatékony algoritmus nem ismert rá ez idő szerint, mi több, nem is remélt. Ez annál inkább szomorú, mert a probléma kezelhetősége lenne az előfeltétele olyan döntések vizsgálatához, hogy melyik él valószínűségét érdemes növelni és mennyivel, vagy éppen mit okozna a hiánya. Nem túl nagy feladatok azonban megoldhatók

¹Erre egy érdekes példa: egy 2017-es IP megoldó sokkal gyorsabban fut egy 1980-as évekbeli szuperszámítógépen, mint egy 1980-as megoldó a mostani legjobb gépeken.

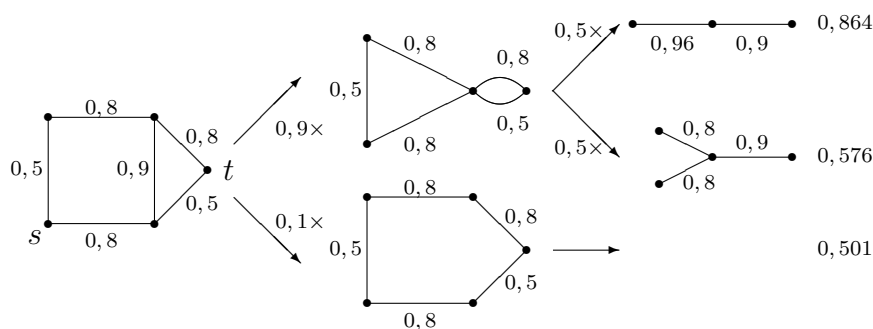
a lehetséges esetek megfelelő csoportosításával. A gondolat egy rekurzív algoritmus, amely az adatok számának függvényében *exponenciális* időt igényel.

Legyen G a kiindulási gráf és válasszunk ki egy $e \in E(G)$ élt, melynek valószínűségét jelöljük $p(e)$ -vel. Jelölje továbbá $G \setminus e$ és G/e rendre azokat a gráfokat amelyekből elhagytuk az e élt, illetve ahol az e él két végpontját egybeejtjük, azaz *összehúzzuk* e -t. Ekkor $Pr(G)$ -vel jelölve az $s - t$ elérhetőség valószínűségét

$$Pr(G) = (1 - p(e))Pr(G \setminus e) + p(e)Pr(G/e),$$

és így az eredeti feladatot két kisebb problémára vezettük vissza.

Példa:



Vegyük észre, hogy egy párhuzamos élpár, melyeknek valószínűsége p_1 és p_2 , helyettesíthető egy $p_1 + p_2 - p_1p_2$ valószínűségű éllel. Az alsó ág kiszámolása hasonlóan megy, illetve ha olyan részgráfra jutunk, amit már kiszámoltunk, az felhasználható. Így $Pr(G) = 0,72 \times 0,9 + 0,501 \times 0,1 = 0,6981$. Összehasonlítva ezt a legnagyobb valószínűségű úttal látható, hogy míg az csak 0,516 valószínűségű, az összefüggőségre ennél sokkal nagyobb az esély.

7.2. Approximáció gráfproblémákra

Számos fontos, széles körben alkalmazható optimalizálási probléma NP -nehéz, vagyis lényegében nem várhatjuk hogy optimális megoldást találjunk reális ($\sim$ polinomiális) futási időben. Az *approximációs algoritmusok* ennek a problémának a feloldására próbálnak választ adni a megoldás legalább valamilyen pontosságú közelítésével. Kulcskérdés, hogy mekkora lehet a közelítés (azaz approximáció) pontossága.

Azt mondjuk, hogy $\mathcal{A}$ algoritmus egy P minimalizálási probléma δ -approximációja, ha P minden I inputja esetén, melyre az optimális megoldás $OPT(I)$, olyan $\mathcal{A}(I)$ megoldás ad, amire

$$\mathcal{A}(I) \leq \delta \cdot OPT(I).$$

Természetesen a definícióban $\delta \geq 1$, és minél közelebb van 1-hez annál jobb az algoritmus. Maximizálási probléma esetén

$$OPT(I) \geq \mathcal{A}(I) \geq \delta \cdot OPT(I),$$

$0 \leq \delta \leq 1$, a követelmény. Itt most csak néhány gráfelméleti problémára mutatunk approximációs algoritmusokat, az érdeklődő olvasónak például a [7, 6, 17] műveket ajánljuk.

7.2.1. Pontlefedés

Adott $G = (V, E)$ gráf és $c : V \rightarrow \mathbb{R}_+$ pontsúlyozás. Keressük azt az $S \subseteq V$ pontthalmazt, amely lefogja G összes élét (azaz minden él illeszkedik valamely S -beli pontra) és a $c(S) = \sum_{v \in S} c(v)$ súlyösszeg minimális. A feladat általánosságban NP -nehéz, ami indokolja approximációs algoritmus keresését.

A feladat speciális esete minimális méretű lefogó halmaz keresése, azaz amikor $c \equiv 1$. Ebben az esetben a maximális párosítás elemszáma ($|M|$) alsó korlátot ad az optimumra, ugyanis a párosítás egyik pontját lefogva lefogtuk a párosításban szereplő éleket. Ugyanakkor $2|M|$ felső korlát, ugyanis ha az M -beli élek végpontjai nem fogtak le minden élt, akkor növelhető lenne a párosítás. Azaz a maximális párosítás egy 2-approximációs algoritmus a pontlefedésre, ha nincs súlyozás. Könnyen látható, hogy a legrosszabb esetre példa a $K_{n,n}$ teljes páros gráf. Ekkor az algoritmus $2n$ -et ad, ugyanakkor n -ponttal fedhető a gráf.

Nézzük most az általános esetet. Ehhez először vegyük azt a súlyfüggvényt ami a csúcs fokszámával arányos, azaz $c(v) = \lambda d(v)$, minden $v \in V$. Ekkor ha U egy optimális lefogó halmaz c -szerint, akkor

$$c(V) = \sum_{v \in V} c(v) = \lambda \sum_{v \in V} d(v) = 2\lambda |E| \leq 2c(U),$$

mivel

$$\sum_{v \in V} d(v) \geq \sum_{v \in U} d(v) \geq |E|,$$

ugyanis U lefogja az összes élt.

Legyen $\lambda := \min_{v \in V} [c(v)/d(v)]$, és $t(v) = \lambda d(v)$ (ekkor $c(v) \geq t(v)$ és létezik olyan v pont, amelyre $c(v) = t(v)$). Ekkor a következő (ún. szintező) algoritmus adható meg.

1. Legyen W azon $v \in V$ pontok halmaza, amelyre $c(v) = t(v)$. Ezeket vegyük be a lefogó pontthalmazba.

2. A már bevett és 0-fokú pontokat vegyül ki a gráfból, és legyen $c(v) := c(v) - t(v)$ a megmaradó gráfon.
3. Ismételjük az eljárást, amíg van nem 0 fokú pont.

Nem túl nehezen bizonyítható, hogy ez az eljárás 2-approximáció a súlyozott pontlefedés problémára.

7.1. Feladat. Írjuk fel a pontlefedés problémát 0 – 1 egészértékű programozási feladatként. Mutassuk meg, hogy a relaxált feladatnak van olyan megoldása, ahol minden változó értéke 0, 1/2 vagy 1. Adjunk ez alapján 2-approximációt a problémára.

7.2. Feladat. Adott $\mathcal{U}$ n elemű alaphalmaz és $\mathcal{S} = \{S_1, \dots, S_m\}$ részhalmazai, úgy, hogy $\bigcup_{i=1}^m S_i = \mathcal{U}$. A feladat az, hogy találjuk meg $\mathcal{S}$ legkisebb részhalmazát úgy, hogy a benne levő halmazok uniója egyelő $\mathcal{U}$ -val. Adjunk meg approximációs megoldást (pl. mohó algoritmus, IP felírás majd relaxáció).

7.2.2. Az utazó ügynök probléma

Az egyik legismertebb NP -nehéz gráf probléma az *utazó ügynök (traveling salesman) probléma*. Adott egy G súlyozott gráf, a feladat az, hogy egy adott pontból indulva látogassuk meg a gráf összes pontját, mielőtt visszaérnénk a kiindulási helyünkre, úgy, hogy a megtett út hossza minimális. Feltesszük, hogy érvényesül a háromszög egyenlőtlenség, azaz a súlyokra (pontok közti távolságokra) teljesül, hogy $\ell_{ij} + \ell_{jk} \geq \ell_{ik} \forall (i, j, k)$.² A feladatra viszonylag könnyen kapunk egy 2-approximációs algoritmust a minimális feszítőfa segítségével. Emlékezzünk vissza, hogy egy összefüggő gráfban pontosan akkor van Euler-út, ha minden foksám páros. Ebben az esetben az Euler-út gyorsan (élszámban lineáris időben) megadható. Az approximációs algoritmus a következő:

1. Konstruáljuk meg a gráf minimális feszítőfáját.
2. Duplázzuk meg a feszítőfa éleit (így minden foksám páros lesz) és adjuk meg az Euler-utat.
3. „Rövidítéseket” használva, ahol lehet, adjunk meg egy valódi utat a gráfon, ami minden pontot egyszer látogat meg.

²Természetesen ezzel a feltevessel is NP -nehéz marad a probléma.

Mivel teljesül a háromszög egyenlőtlenség, így a rövidített út biztosan nem hosszabb, mint az Euler-út, aminek a hossza nem több, mint kétszer a minimális feszítőfa összhossza (súlya). Mivel az optimális út egy feszítőfa és egy plusz él, adódik a kívánt közelítés.

Egy jobb, $3/2$ -es közelítést Christofides adott 1978-ban. Az algoritmus azt használja ki, hogy ha a minimális feszítőfa egyes pontjainak fokszáma páros, akkor azoknál a pontoknál az élek duplázása (az előző algoritmus 2-es pontja szerint) felesleges. Christofides algoritmus csak a minimálisan szükséges plusz élet adja a feszítőfához.

1. Konstruáljuk meg a gráf minimális feszítőfáját. Legyen a páratlan fokú pontok halmaza V^{odd} .
2. Keressük meg a minimális súlyú teljes párosítást³ a V^{odd} által feszített gráfon, legyen ez M^* .
3. Adjuk M^* éleit a feszítőfa éleihez, ezzel minden pont foka páros lesz a kapott gráfban. Adjuk meg az Euler-utat ebben a gráfban.
4. „Rövidítéseket” használva, ahol lehet, adjunk meg egy valódi utat a gráfon, ami minden pontot egyszer látogat meg.

Tudjuk, hogy a minimális feszítőfa súlya alsó korlát az optimumra. Figyeljük meg, hogy az optimális út megszorítható V^{odd} pontjaira úgy, hogy hossza nem nő. Tudjuk, hogy V^{odd} páros számú csúcsot tartalmaz, és egy páros hosszú kör (V^{odd} pontjain) két teljes párosításra bontható, legyenek ezek M_1, M_2 . Így $|M_1|, |M_2| \geq |M^*|$. Ebből adódik, hogy az optimális út hossza nagyobb egyenlő mint $|M_1| + |M_2| \geq 2|M^*|$. A feszítőfa és $|M^*|$ éleinek összsúlya adja a $3/2$ -es közelítést.

7.2.3. Maximális klikk probléma

Egy G gráf $K \subseteq V$ csúcshalmaza *klikk*, ha K minden csúcspárja össze van kötve. Számos alkalmazásban felmerül, hogy megkeressük a gráf legnagyobb (vagy legnagyobb súlyú) klikkjét. A probléma NP -nehéz, sőt, még az $n^{0.99}$ -nél jobb approximáció elérése is NP -teljes probléma, ami elég lehangoló eredmény. Számos fontos alkalmazás miatt ugyanakkor klikkek keresése nagy gráfokban ma is aktívan kutatott téma. Itt egy nem triviális approximációs algoritmust mutatunk a feladatra.

³Olyan párosítás, mely a gráf összes pontját lefedi és a benne lévő élek súlyösszege minimális.

1. A gráf n pontját osszuk n/k darab k méretű részre. Minden halmazra nézzük meg, hogy nem-e klikk.
2. Adjuk meg a legnagyobb klikket (K) amit találtunk. Fix k -ra az algoritmus $O(\frac{n}{k} \cdot 2^k)$ idő alatt megtehető (2^k részgráfját ellenőrizzük minden k elemű ponthalmaznak). A $k = \log n$ választással az algoritmus polinom idejű lesz.

Az algoritmus működése azon áll, hogy egy klikk minden részhalmaza is klikk. Ha K^* a maximális klikk a gráfban, akkor a skatulyaelv miatt van olyan halmaz az n/k közül, amelyben $|K^*|/(n/k) = (k/n)|K^*|$ eleme van K^* -nak. Az algoritmus garantálja, hogy megadott $|K| \geq (k/n)|K^*|$, vagyis egy $O(n/\log n)$ approximációt kapunk.

7.3. Feladat. Egy gráf csúcsainak jó színezése egy olyan színezés, ahol bármely két szomszédos csúcs különböző színű. A feladat a gráf jó kiszínezése minimális számú színnel. Mutassuk meg, hogy a maximális klikk mérete alsó korlátot ad erre. Mutassuk meg, hogy a mohó színezés nem több mint $\Delta(G) + 1$ színt használ, ahol $\Delta(G)$ a maximális fokszám G -ben.

7.4. Feladat. Adjunk approximációs algoritmust a maximális független halmaz problémára a maximális klikkre vonatkozó algoritmus felhasználásával.

7.3. A maximális vágás probléma

A maximális vágás egy újabb NP -nehéz feladat.⁴ Egy $G = (V, E)$ súlyozott gráf ($c : E \rightarrow \mathbb{R}_+$) vágását és minimális vágását a szállítási probléma kapcsán már definiáltuk. A maximális vágás olyan $X, \bar{X} = V - X$ vágás amelyre $c(X, \bar{X}) = \sum_{i \in X, j \in \bar{X}} c_{ij}$ maximális. Láttuk, hogy a minimális vágás feladat polinom megoldható, talán emiatt némileg meglepő, hogy a maximalizálási feladat NP -nehéz.

A következő két egyszerű algoritmus a problémára Erdős Pál nevéhez fűződik.

7.3.1. Maximális vágás mohó módszerrel

Az első kézenfekvő megközelítés a mohó algoritmus használata. Vegyük a maximális súlyú $e = (x, y)$ élt és legyen $x \in X$ és $y \in \bar{X}$. A maradék csúcsokat sorban vizsgálva tegyük a vizsgált

⁴Itt nem tárgyaljuk, de érdekes módon síkgráfok esetén polinomiális algoritmus adható.

csúcsot oda, ahol nagyobb növelést érünk el c -n. Könnyű meggondolni, hogy

$$c(X, \overline{X}) \geq \frac{1}{2} \sum c(e) \geq \frac{1}{2} c(e).$$

7.3.2. Maximális vágás véletlen módszerrel

Elsőként egy egyszerű, polinom futási idejű véletlen algoritmust adunk meg, ami 2-approximációt ad a problémára. Itt feltesszük, hogy $c \equiv 1$, azaz a maximális vágás a partíciók közti élek számát maximalizálja. Minden egyes v csúcshoz dobjunk fel egy szabályos érmét; ha a dobás fej, legyen $v \in X$, különben legyen $v \in \overline{X}$. Legyen S a vágásban szereplő élek száma. Minden $e \in E(G)$ élhez legyen $\xi_e = 1$, ha e a vágásban van és $\xi_e = 0$ különben. Ekkor $S = \sum_{e \in E(G)} \xi_e$. A várható érték linearitása miatt

$$\mathbb{E}(S) = \mathbb{E}\left[\sum_{e \in E(G)} \xi_e\right] = \sum_{e \in E(G)} \mathbb{E}[\xi_e] = \sum_{e \in E(G)} P(\xi_e = 1).$$

Gondoljuk meg, hogy $P(\xi_e = 1) = 1/2$, amiből $\mathbb{E}(S) = |E(G)|/2$. Mivel a vágás maximális élszáma $|E(G)|$, adódik a 2-approximáció.

A másik oldalról a következő negatív eredmény ismert.

7.1. Tétel (Håstad). *Ha létezik polinomiális algoritmus, ami kiszámol egy $(X, \overline{X})$ vágást mely az optimumot $16/17$ -nél jobban közelíti, akkor $P = NP$.*

7.3.3. Reprezentáció R^n -ben

A maximális vágás probléma megfogalmazható

$$\max \frac{1}{2} \sum_{i < j} c_{ij} (1 - y_i y_j)$$

kvadratikus egészértékű programozási feladat formájában az $y_i \in \{-1, 1\}$ minden $i \in V$ feltétel mentén. Legyen ugyanis $X = \{i \in V(G) : y_i = 1\}$, ekkor $c(X, \overline{X})$ pontosan az előbb megadott célfüggvény. Ilyen feladatokat megoldani természetesen NP -nehéz. A feladatot relaxálva y_i -ket x_i magas dimenziós (Euklideszi) térbeli egységvektorokra cseréljük le, és a következő feladatot kapjuk:

$$\max \frac{1}{2} \sum_{i < j} c_{ij} (1 - x_i \cdot x_j),$$

feltéve, hogy $x_i \in \mathbb{S}^n$ minden $i \in V$. Itt $x_i \cdot x_j$ a vektorok skalárszorzatát jelenti, $\mathbb{S}^n$ pedig az $n + 1$ -dimenziós egységgömb felszíne. Így egy *szemidefinit programozási feladatot* kapunk, ami konvex programozási feladat, vagyis polinom időben megoldható.⁵

A megoldás mögötti intuíció az, hogy azon vektoroknak, melyek a vágás különböző oldalára eső pontokhoz tartoznak, „távol” kell lenniük egymástól ($\mathbb{S}^n$ -en). Az algoritmus menete a következő. Először megoldjuk a relaxált feladatot, ezzel megkapjuk $x_1, x_2, \dots, x_n \in \mathbb{S}^n$ vektorokat. Ezután véletlenszerűen veszünk egy $r \in \mathbb{S}^n$ vektort. Legyen $X := \{x_i : x_i \cdot r \geq 0\}$ és $\bar{X} := \{x_i : x_i \cdot r < 0\}$.

7.1. Lemma. $\mathbb{P}(\text{sign}(x_i \cdot r) \neq \text{sign}(x_j \cdot r)) = \frac{1}{\pi} \arccos(x_i \cdot x_j)$ minden $i \neq j$ esetén.⁶

Bizonyítás. Gondoljuk meg, hogy elég ha x_i, x_j és r vektorokat a síkon tekintjük (vegyük a síkot, melyet x_i és x_j vektorok feszítenek, vetítsük le ide r -et, és normalizáljuk, hogy egység hosszú legyen; figyeljük meg, hogy csak r iránya érdekes). Szintén egyszerű meggondolni, hogy ez az eljárás ekvivalens azzal, hogy választunk egy egységvektort véletlenül az x_i és x_j által meghatározott egységkörön. Ekkor $\text{sign}(x_i \cdot r) \neq \text{sign}(x_j \cdot r)$ pontosan akkor teljesül, ha r az x_i -re és x_j -re merőleges egyenesek által közrefogott részbe (két körcikk) esik. Ha x_i és x_j által bezárt szög τ , akkor ennek a valószínűsége $2\tau/2\pi$. Mivel x_i és x_j egységvektorok, ezért $x_i \cdot x_j = \cos(\tau)$. Ezzel, behelyettesítés után adódik az állítás. $\square$

7.2. Tétel. Legyen C véletlen változó annak vágásnak kapacitása melyet az algoritmus adott. Ekkor $\mathbb{E}(C) = \frac{1}{\pi} \sum_{i < j} c_{ij} \cdot \arccos(x_i \cdot x_j)$

Bizonyítás. Legyen $\xi_{ij} = 1$, ha (i, j) él benne van a vágásban és $\xi_{ij} = 0$ különben. Ekkor $C = \sum_{i < j} c_{ij} \xi_{ij}$. A várható érték linearitásából és a 7.1-ből egyszerűen adódik az állítás. $\square$

Legyen

$$\alpha = \min_{0 \leq \psi \leq \pi} \frac{2}{\pi} \frac{\psi}{1 - \cos \psi},$$

akkor minden $-1 \leq y \leq 1$ esetén

$$\frac{\arccos(y)}{\pi} \geq \alpha \cdot \frac{1}{2}(1 - y),$$

mely az $y = \cos(\psi)$ helyettesítés után könnyen adódik. Ezután egyszerű számolással kapjuk hogy $\alpha > 0.87856$. Ezek után gyorsan adódik a fő eredmény.

⁵A megoldás azt jelenti, hogy az optimum $(1 + \varepsilon)$ pontossággal közelíthető bármely $\varepsilon > 0$ esetén.

⁶Itt sign az előjel függvényt jelöli.

7.3. Tétel (Goemans- Williamson). *Az algoritmus egy olyan vágást ad mely, mely kapacitása várható értékben $\alpha \cdot OPT \geq 0.87756 \cdot OPT$, ahol OPT a maximális vágás kapacitása.*

Bizonyítás. Legyen a relaxált szemidefinit programozási feladat optimális megoldása $\gamma \geq OPT$.

Ekkor

$$\mathbb{E}(C) = \frac{1}{\pi} \sum_{i < j} c_{ij} \cdot \arccos(x_i \cdot x_j) \geq \sum_{i < j} c_{ij} \alpha \frac{1}{2} \cdot (1 - x_i \cdot x_j) = \alpha \gamma \geq \alpha \cdot OPT$$

□

7.5. Feladat. *Formalizáljuk a maximális vágás problémát LP feladatként. Mutassuk meg, hogy a megoldás nem ad jobb mint 2-approximációt.*

8. fejezet

Mohó algoritmusok

Mint korábban láthattuk, számos iteráción alapuló algoritmusnak van *mohó* aspektusa: az algoritmus az adott pillanatban a legkedvezőbbnek tűnő lehetőséget választja. Ez a megközelítés néha kiváló, néha katasztrofális eredménnyel jár; mindenesetre egy újonnan felmerülő problémánál célszerű szerencsét próbálni vele. Ha pontos megoldást nem is, hasznos információkat nyerhetünk általa. A mohó algoritmusok általában nagyon gyorsak és egyszerűek, így ha egy feladatot pontosan oldanak meg, akkor nemigen lehet jobbat találni náluk. A következő néhány problémában ez az eset áll fenn.

8.1. Matroidok

8.1.1. Maximális súlyú feszítőfa

Tegyük fel, hogy adott egy irányítatlan, összefüggő G gráf, melynek éleihez nem negatív súlyokat rendelünk; $w(e) \geq 0$, ha $e \in E(G)$. Célunk egy olyan $X \subseteq E(G)$ élhalmaz megadása, mely X nem tartalmaz kört és a $\sum_{e \in X} w(e)$ maximális. (A szumma jelölhető $w(X)$ -szel és ez X *súlya*.)

Megjegyzés: Mivel $w(e) \geq 0$ minden e élre, feltehető, hogy X maximális körmentes halmaz, azaz *feszítőfa* G -ben. Könnyebben motiválható lenne egy *minimális súlyú* feszítőfa keresése, ami például egy minimális költségű összefüggő kommunikációs hálózatot modellezhet. A probléma valóban így vetődött fel először (Borůvka 1926, illetve Kruskal 1956), nem nehéz viszont belátni az ekvivalenciájukat.

Az alábbi mohó algoritmus tűnik kézenfekvőnek: Rendezzük az élhalmaz $e_1, e_2, \dots, e_m$ elemeit úgy, hogy $w(e_1) \geq w(e_2) \geq \dots w(e_m)$. Legyen $X_1 = \{e_1\}$, majd az X_{i-1} halmazhoz próbáljuk hozzávenni az e_i élt; ha $X_{i-1} \cup \{e_i\}$ körmentes, akkor $X_i := X_{i-1} \cup \{e_i\}$, ellenkező esetben $X_i := X_{i-1}$. Legyen végül $X := X_m$, illetve tulajdonképpen $X := X_i$, ahol i a legkisebb szám, amelyre $|X_i| = n - 1$.

Az algoritmus helyességét sokkal általánosabb körülmények között is beláthatjuk; csupán a körmentes élhalmazok néhány tulajdonságára van szükség. Nyilvánvaló, hogy a $\emptyset$, azaz az üres halmaz körmentes, illetve ha $X \subseteq E(G)$ körmentes, akkor bármely $Y \subseteq X$ is az. Szükségünk van továbbá az alábbira állításra:

8.1. Lemma. *Ha X és Y egy G gráf körmentes élhalmazai és $|Y| < |X|$, akkor van olyan $e \in X \setminus Y$, hogy $Y \cup \{e\}$ is körmentes.*

Bizonyítás. A G gráf pontjain az Y élek által alkotott komponensek részfák. Egy-egy ilyen komponensben az X élhalmaznak sem lehet több éle, mint Y -nak, hiszen különben tartalmazna kört. Mivel $|X| > |Y|$, van olyan $e \in X$, mely az Y két komponensét köti össze. Ezzel az e éllel az $Y \cup \{e\}$ halmaz körmentes. $\square$

8.1. Feladat. *Mutassuk meg, hogy ha egy gráfban minden súly különböző, akkor a maximális (minimális) feszítőfa egyértelmű.*

8.2. Feladat. *Egy irányítatlan súlyozott gráf egy feszítőfáját üvegnyakú feszítőfának nevezzük, ha a maximális élsúlya minimális a feszítőfák között.*

1. *Igaz-e, hogy minden minimális feszítőfa üvegnyakú is?*

2. *Igaz-e, hogy minden üvegnyakú feszítőfa minimális is?*

8.3. Feladat. *Igaz-e, hogy ha egy összefüggő irányítatlan súlyozott gráfban a legnagyobb súly szigorúan nagyobb, mint a többi, akkor ez az él nem szerepelhet a minimális feszítőfában? Igazoljuk, ha van olyan kör, ami ezt az élt tartalmazza, akkor igaz az állítás.*

8.1.2. Matroidok: formális definíció

Ha adott egy E véges halmaz és az S részhalmazainak egy $\mathcal{I}$ halmaza úgy, hogy

(I_1) $\emptyset \in \mathcal{I}$;

(I_2) Ha $X \in \mathcal{I}$ és $Y \subseteq X$, akkor $Y \in \mathcal{I}$;

(I_3) Ha $X \in \mathcal{I}$ és $Y \in \mathcal{I}$; továbbá $|X| > |Y|$, akkor van olyan $x \in X \setminus Y$ úgy, hogy $Y \cup \{x\} \in \mathcal{I}$,

akkor az $M = (E, \mathcal{I})$ pár *matroid*. Az $\mathcal{I}$ -be tartozó halmazokat a matroid *független* halmazainak nevezzük.

Egy halmazrendszer, amely az (I_1) és (I_2) axiómákat kielégíti *függetlenségi rendszer*.

Példák:

1. Az előbbieket szerint egy G irányítatlan gráf $E(G)$ élhalmazának körmentes részhalmazai matroidot alkotnak.

Ez a G gráf *körmatroidja*.

2. *Vektormatroidok*. Legyen E véges sok vektorból álló halmaz, $\mathcal{I}$ pedig a lineárisan független részhalmazainak halmaza.

Erre (I_1) és (I_2) nyilvánvalóan következik, (I_3) pedig a jól ismert *Steinitz-féle kicserélési tétel*.

Megjegyzés: (I_3)-ból következik, hogy a maximális független halmazok egyforma elemszámúak. Ezek az M matroid *bázisai*, a halmazukat $\mathcal{B}$ jelöli..

Célunk egy $M = (E, \mathcal{I})$ matroid és $w : E \rightarrow \mathbb{R}^+$ esetén annak az $X \in \mathcal{I}$ halmaz megtalálása, amelyre a $\sum_{x \in X} w(x)$ maximális. (Feltehető, hogy ez a halmaz maximális, azaz bázis.) A *mohó algoritmus* a következő: Rendezzük $E = \{x_1, \dots, x_m\}$ -et úgy, hogy $w(x_i) \geq w(x_j)$, $1 \leq i < j \leq m$.

Legyen $X_1 := \{x_1\}$, $X_i := X_{i-1} \cup \{x_i\}$, ha $X_{i-1} \cup \{x_i\} \in \mathcal{I}$, $X_i := X_{i-1}$ különben. Végül pedig $X := X_m$.

8.1. Tétel. *Tetszőleges $M = (E, \mathcal{I})$ matroid és w nem negatív súlyfüggvény esetén a mohó algoritmus egy maximális súlyú független X halmazt talál.*

Bizonyítás. Tegyük fel, hogy a mohó algoritmus az $X = \{x_1, x_2, \dots, x_n\}$ bázist adta, de egy $Y = \{y_1, y_2, \dots, y_n\}$ bázisra $\sum_{i=1}^n w(x_i) < \sum_{i=1}^n w(y_i)$. Feltesszük, hogy az X és Y elemei

csökkenő súly szerint vannak rendezve, azaz $w(x_i) \geq w(x_j)$ és $w(y_i) \geq w(y_j)$, $1 \leq i < j \leq n$. Mivel $\sum w(x_i) < \sum w(y_i)$, lennie kell egy olyan legkisebb k számnak, amelyre $w(x_k) < w(y_k)$. Legyen $A = \{x_1, \dots, x_{k-1}\}$ és $B = \{y_1, \dots, y_k\}$. (Ha $k = 1$, akkor $A = \emptyset$.) Az (I_2) miatt $A, B \in \mathcal{I}$ és $|A| < |B|$, így (I_3) miatt van olyan $y_j \in B$, melyre $A \cup \{y_j\} \in \mathcal{I}$. Ekkor viszont $w(y_j) \geq w(y_k) > w(x_k)$ és így a mohó algoritmus nem az x_k , hanem az y_j elemet választotta volna. $\square$

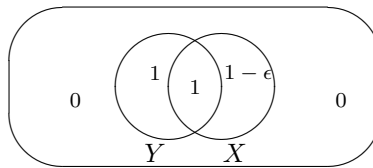
A fenti tétel részlegesen megfordítható és ez mutatja a matroidok és a mohó algoritmus szoros kapcsolatát.

8.2. Tétel (Edmonds). *Legyen E egy véges halmaz, $\mathcal{I}$ pedig S részhalmazainak egy (I_1) -et és (I_2) -öt kielégítő halmaza, amely nem matroid. Ekkor van olyan $w : E \rightarrow R^+$ súlyfüggvény, amelyre a mohó algoritmus nem találja meg $\mathcal{I}$ maximális súlyú elemét.*

Bizonyítás. Mivel az $(E, \mathcal{I})$ pár nem matroid ((I_3) nem mindig teljesül), van olyan $X, Y \in \mathcal{I}$, hogy $|X| > |Y|$, de bármely $x \in X \setminus Y$ esetén az $Y \cup \{x\} \notin \mathcal{I}$. Legyen $w(y) = 1$ minden $y \in Y$ -ra, $w(x) = 1 - \epsilon$, ha $x \in X \setminus Y$, ahol $\epsilon > 0$, és $w(z) = 0$ különben. Ekkor a mohó algoritmus az Y halmazt szolgáltatja; ennek súlya $\sum_{y \in Y} w(y) = |Y|$. Másrészt $X \in \mathcal{I}$, a súlya pedig

$$\begin{aligned} \sum_{x \in X} w(x) &= \sum_{x \in X \cap Y} w(x) + \sum_{x \in X \setminus Y} w(x) = \\ &= |X \cap Y| + (1 - \epsilon)|X \setminus Y| \geq (1 - \epsilon)|X| \geq (1 - \epsilon)(|Y| + 1). \end{aligned}$$

Ha $0 < \epsilon < 1$, akkor az $(1 - \epsilon)(|Y| + 1) > |Y|$, és így $w(X) > w(Y)$.

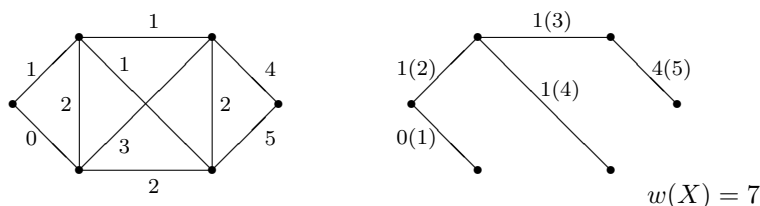


$\square$

Megjegyzés: Amennyiben minimális súlyú bázist szeretnénk találni, alkalmazhatunk egy egy transzformációt: Vegyük a súlyok -1 -szeresét, majd adjunk hozzá egy elegendően nagy konstans. ($\hat{w}(x) = -w(x) + K$, ahol $K \geq w(x)$ minden $x \in S$ -re.) Mivel a bázisok azonos elemszámúak, az új súlyok mellett maximális $X \in \mathcal{I}$ minimális az eredeti súlyfüggvénnyel. Ugyanezen transzformáció szerint módosíthatjuk a mohó algoritmust és a súlyok szerint növekvő sorrendbe állított

elemeken végrehajtva egyből a minimumot kapjuk.

Példa: Balra a súlyozott G gráf, jobbra a minimális feszítőfa; a zárójelben az adott él megoldásba kerülés ideje (lépése) látható.



Felmerülhet a kérdés, van-e egyáltalán más matroid, mint a körmatroidok. Vegyünk egy n elemű S halmazt, és legyen $\mathcal{I}$ az összes k -nál ($k \leq n$) nem nagyobb elemszámú részhalmazának halmaza. Ez az ún. $U_{n,k}$ *uniform matroid*. Könnyű belátni, hogy például az $U_{4,2}$ nem lehet semmilyen G gráf körmatroidja. (Szintén nem nehéz megmutatni, hogy az $U_{n,k}$, $0 \leq k \leq n$ felfogható, mint vektormatroid.)

Természetesen más körülmények között a mohó algoritmus sikeres lehet anélkül, hogy egy matroid struktúra garantálná az algoritmus elérését. Egy egyszerű példa erre a legrövidebb utak problémájának egy speciális esete.

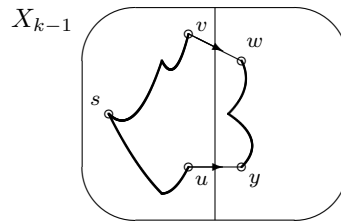
8.1.3. Dijkstra algoritmusa

Tegyük fel, hogy egy G irányított gráfban minden él $\ell(v, w)$ súlya nem negatív. A G gráfban nincs negatív súlyú kör, így bármely s és v pontokra létezik az s -ből v -be vezető utak hosszának a minimuma (ez végtelen, ha nincs $s - v$ út). Az s -ből kiinduló legrövidebb utak hosszát, illetve ezen utakat leíró feszítőfát egy egyszerű és gyors algoritmussal, $n = |V(G)|$ iterációban megkaphatjuk. Az eljárást 1956-ban publikálta Edsger Dijkstra. Az algoritmus a következő lépésekből áll:

1. lépés: legyen $X_1 := \{s\}$, $d(s) = 0$, $d(v) = \infty$ ha $v \neq s$ és $T_1 := \{s\}$, egypontú fa.
- k. lépés: tegyük fel, hogy X_{k-1} és $d(v)$ ($v \in X_{k-1}$) már definiáltak. Válasszunk egy olyan $w \in V(G) \setminus X_{k-1}$ pontot, amelyre a $d(v) + \ell(v, w)$ minimális, ahol $v \in X_{k-1}$. Legyen $X_k := X_{k-1} \cup \{w\}$, $d(w) := d(v) + \ell(v, w)$, T_k pedig az a fa, amelyet T_{k-1} -ből a w pont és a (v, w) út hozzávételével nyerünk.

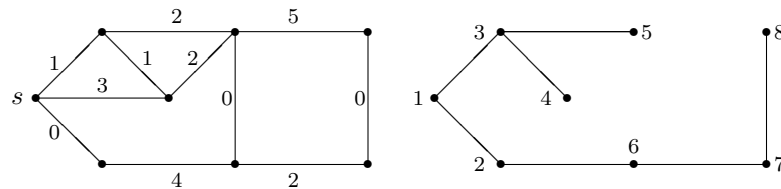
8.3. Tétel. A fentiek mellett az algoritmus n -edik lépése után $d(v)$ értéke a legrövidebb $s - v$ út hossza minden $v \in V(G)$ esetén. Továbbá T_n az s -ből kiinduló legrövidebb utakat kódoló fa.

Bizonyítás. A tételt a lépések száma szerinti teljes indukcióval látjuk be. Az indukciós feltétel szerint $v \in X_{k-1}$ esetén $d(v)$ a legrövidebb $s - v$ út hossza, és tegyük fel, hogy $w \in X_k \setminus X_{k-1}$ -re $d(w) = \min_{v \in X_{k-1}} d(v) + \ell(v, w)$ nem ezt, azaz a legrövidebb $s - w$ út hosszát adja. Ekkor a p legrövidebb $s - w$ út egy u pontból kilépve először hagyja el X_{k-1} -et. Legyen a p útban az u után következő pont y (feltehető, hogy $y \neq w$).

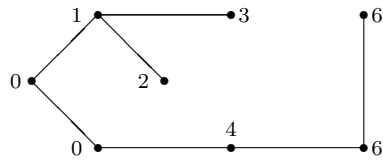


Mivel p hossza kisebb, mint $d(w)$, a p vonalán haladó $s - y$ út hossza is kisebb, mint $d(w)$. (Itt használjuk ki a súlyok nem negativitását.) Ekkor viszont az algoritmus az y pontot választotta volna a k -edik lépésben, ellentmondás. A T_n fa szerepének bizonyítása szintén indukcióval történhet; így T_{k-1} rögzíti a legrövidebb $s - v$ utakat ($v \in X_{k-1}$) és azon (v, w) él hozzávételével változtatjuk T_{k-1} -et T_k fává, amelyen keresztül a w pont $d(w)$ hosszú úton elérhető s -ből. $\square$

Példa: Baloldalon egy irányítatlan, súlyozott G gráf az s kiindulóponttal. Jobboldalon az s -ből induló legrövidebb utak fáját; a számozás azt mutatja, hányadik lépésben került be a pont a fába.

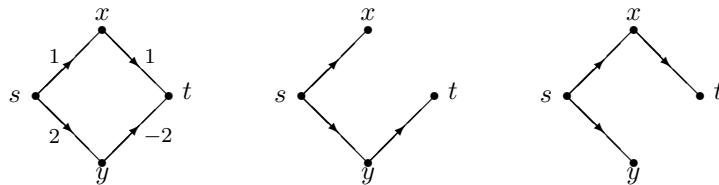


A legrövidebb utak hosszát is könnyen leolvashatjuk. A gyökérpont, azaz s önmagától vett távolsága nulla, egy x pontra pedig úgy kaphatjuk meg $d(x)$ -et, ha összeadjuk az fában az $s - x$ út élein lévő súlyokat.



Megjegyzés: Dijkstra algoritmusával jóval egyszerűbb és gyorsabb, mint a korábban ismertetett Bellman algoritmus, ugyanakkor *csak* nemnegatív súlyok esetén alkalmazható.

Példa: Baloldalon a gráf, középen a Bellmann algoritmus által adott (helyes) legrövidebb utak fája, jobboldalon pedig a Dijkstra algoritmus végrehajtásával kapott fát ábrázoltuk.



Azaz a Dijkstra módszer *nem* találja meg a legrövidebb $s - t$ utat, hisz $d(t) = 0$, nem kettő és a legrövidebb $s - t$ út nem az $\{s, x, t\}$ hanem az $\{s, y, t\}$.

8.4. Feladat. Az incidencia mátrix felhasználásával lássuk be, hogy egy G gráf körmentes élhalmazai vektormatroidot alkotnak.

8.5. Feladat. Vegyük egy G gráf olyan részhalmazait, amelyek legfeljebb egy kört tartalmaznak. Matroid lesz ez? Általában, ha az S halmaz felett adott az M_1 és M_2 matroid, vegyük azon $F \subset S$ halmazokat, amelyekre $F = A \cup B$, ahol A ill. B független M_1 -ben ill. M_2 -ben. Matroid lesz ez a halmazrendszer?

8.6. Feladat. Lássuk be, hogy az $U_{4,2}$ nem lehet semmilyen gráf körmatroidja. Van olyan matroid, amely nem vektormatroid?

8.7. Feladat. Egy $M = (E, \mathcal{I})$ matroid bázisa $\mathcal{B}$. Lássuk be, hogy

1. $\mathcal{B} \neq \emptyset$
2. ha $A, B \in \mathcal{B}$ és $a \in A \setminus B \neq \emptyset$, akkor van olyan $b \in B \setminus A$, hogy $A \setminus \{a\} \cup \{b\} \in \mathcal{B}$.

8.8. Feladat. Tegyük fel, hogy a 8.7 feladat 1. és 2. feltételeit teljesítő $\mathcal{B} \subset 2^E$ halmazrendszer adott. Legyen $F \in \mathcal{I}$ ha $F \subset A$ valamely $A \in \mathcal{B}$ bázisra. Ekkor $(E, \mathcal{I})$ matroid, amelynek éppen a $\mathcal{B}$ a bázisainak halmaza.

8.9. Feladat. Legyen $M = (M\mathcal{B})$ a 8.8 feladatnak megfelelően a bázisával megadott matroid. Legyen $\overline{\mathcal{B}} := \{E \setminus A : A \in \mathcal{B}\}$. Lássuk be, hogy $\overline{M} := (E, \overline{\mathcal{B}})$ is matroid; ez lesz az M matroid duális matroidja.

8.10. Feladat. Egy G síkgráf egy tetszőleges síkbarajzolásának duális gráfja legyen G^* . Lássuk be, hogy G körmatroidjának duális matroidja éppen G^* körmatroidja.

8.11. Feladat. Egy $M = (E, \mathcal{I})$ matroidra definiáljuk az r halmazfüggvényt, ez az ún. rangfüggvény. Ha $X \subset E$, akkor $r(X) := \max\{|F| : F \subset X, F \in \mathcal{I}\}$. Lássuk be, hogy r szubmoduláris, azaz minden $X, Y \subset E$ -re $r(X \cup Y) + r(X \cap Y) \leq r(X) + r(Y)$.

8.1.4. Szubmoduláris függvények

Legyen S véges halmaz, $f : 2^S \rightarrow \mathbb{R}$ az S részhalmazaihoz egy valós számot rendelő függvény. Azt mondjuk, hogy f szubmoduláris, ha minden $X, Y \subseteq S$ esetén

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y).$$

Ekvivalens definíció, ha minden $X, Y \subseteq S$, $X \subseteq Y$ és minden $x \in S \setminus Y$ esetén

$$f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y).^1$$

Egy szubmoduláris függvény *monoton* ha minden $X \subseteq Y$ esetén $f(X) \leq f(Y)$.

Példák.

1. Minden $f(S) = \sum_{x \in S} w(x)$ alakban adott függvény szubmoduláris. Ha $w(x) \geq 0$ minden $x \in S$, akkor monoton is.²
2. Legyen $M = (E, \mathcal{I})$, $A \subseteq S$ akkor A maximális független részhalmazai azonos méretűek. Ez a méret A rangja, $r(A)$. A rangfüggvény r szubmoduláris.

¹Közgazdászok ezt gyakran *csökkenő hozadék elvének* nevezik.

²Gondoljunk vissza a fejezet elején tárgyalt minimális súlyú feszítőfára

Természetesen adódik az optimalizálási probléma, hogy keressük meg azt az $X \subseteq S$ halmazt, amelyre f szubmoduláris függvény minimális. A feladat polinomiális időben megoldható. Például a minimális feszítőfa, vagy minimális vágás megtalálása is ennek a problémának egy speciális esete. Ugyanakkor szubmoduláris függvények maximalizálása NP -nehéz feladat. Egy kézenfekvő megközelítés monoton szubmoduláris függvények maximalizálásra a mohó stratégia. Eszerint üres halmazból kiindulva minden lépésben azt az $x \in S$ elemet adjuk hozzá a keresett $X \subseteq S$ halmazunkhoz, mely elem választásával az f függvényérték a legnagyobb mértékben nő. A feladat a $\max_{X \subseteq S} \{f(X) : |X| \leq K\}$ formában fogalmazható meg.

Mohó heurisztika szubmoduláris halmazfüggvényekre³

Legyen $X_0 = \emptyset$, $S_0 = S$ és $\rho_x(X) = f(X \cup \{x\}) - f(X)$, azaz f növekménye ha az aktuális halmazunkhoz a $x \in S$ elemet adjuk hozzá. A t -edik lépésben válasszuk azt a $x(t) \in S_{t-1}$ elemet amelyre $\rho_{x(t)}(X_{t-1}) = \max_{x \in S_{t-1}} \rho_x(X_{t-1})$. Ha több ilyen elem is van, akkor az egyiket tetszőlegesen. Legyen $\rho_{t-1} = \rho_{x(t-1)}$.

1. lépés. Ha $\rho_{t-1} \leq 0$ akkor megállunk, $|X| = K^* = t - 1 < K$. Ha $\rho_{t-1} > 0$, akkor legyen $X_t = X_{t-1} \cup \{x(t)\}$ és $S_t = S_{t-1} \setminus \{x(t)\}$ és tovább a 2. lépésre.
2. lépés. Ha $t = K$, akkor megállunk, $|X| = K^* = K$. Különben legyen $t = t+1$ és folytassuk az eljárást.

Legyen F a maximalizálási feladat optimális megoldást, F_G pedig a mohó heurisztikával kapott megoldás. Vegyük észre, hogy $F_G = f(\emptyset) + \rho_0 + \dots + \rho_{K^*-1}$, ($K^* \leq K$). Megmutatható, hogy ekkor $(F - F_G)/(F - f(\emptyset)) \leq [(K - 1)/K]^K \approx 1 - 1/e$. A következő tétel kimondja, hogy a mohó algoritmus egy elfogadhatóan jó approximációját adja az optimalizálási feladatnak.

8.4. Tétel (Nemhauser-Wosley-Ficher, 1978). *Legyen $f : 2^S \rightarrow \mathbb{R}_+$ nemnegatív szubmoduláris függvény és $\{S_t\}_{t \geq 0}$ a mohó módon választott halmazok sorozata az előző algoritmus szerint. Ekkor minden pozitív egész k és ℓ esetén*

$$f(S_\ell) \geq (1 - e^{-\ell/k}) \max_{S: |S| \leq k} f(S).$$

Speciálisan, ha $k = \ell$, akkor $f(S_k) \geq (1 - 1/e) \max_{|S| \leq k} f(S)$.

³Jóval részletesebben ld. Nemhauser, Wosley és Fisher, 1978.

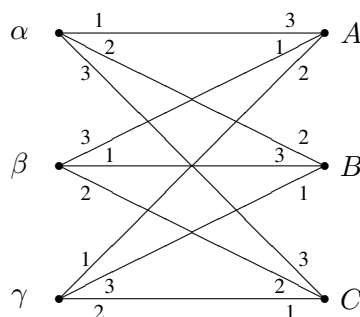
8.2. Stabil párosítások

A stabil párosítás vagy stabil házasság probléma kiváló példa mind a gyakorlat és elmélet viszonyának szemléltetésére, mind a mohó algoritmus egy újabb illusztrációjára. A problémakört eredetileg az USA-ban a 40-es évek közepén kulmináló orvos gyakornok hiány, illetve elosztási zavar motiválta. A végzős orvosok ezreit kellett a kórházak által meghirdetett helyekre beosztani; ráadásul mindkét fél (orvos vs. kórház) a saját preferenciáit igyekezett érvényesíteni. Az eredetileg alkalmazott technikák teljesen alkalmatlanná váltak 1947-re, mikor is egy radikálisan új rendszert vezettek be helyettük. Érdekes módon ennek elméleti vizsgálatát csak 1962-ben tette meg D. Gale és L. S. Shapley, s igazából ők nem tudtak a problémáról: az egyetemi felvételi rendszert illetve a házasságok stabilitását akarták modellezni.⁴

Mi az általuk vizsgált legegyszerűbb modellt ismertetjük, utalva rá, hogy igen sok általánosítás született azóta. A stabil házasság problémában adott n férfi, n nő és mindegyikük valahogyan rangsorolja az ellentétes nem tagjait; ez az illető személy *preferencia listája*. A férfiakat görög, a nőket latin betűkkel jelöljük majd. Így például akkor mondjuk, hogy az α (férfi) *jobban kedveli* vagy *preferálja* A -t B -hez képest, ha α preferencia listáján A előrébb van, mint B . A személyeket és preferenciáikat leírhatjuk (duplán) súlyozott páros gráfokkal, vagy mátrixokkal is az alábbiaknak megfelelően:

Példa:

	A	B	C
α	1, 3	2, 2	3, 1
β	3, 1	1, 3	2, 2
γ	2, 2	3, 1	1, 3



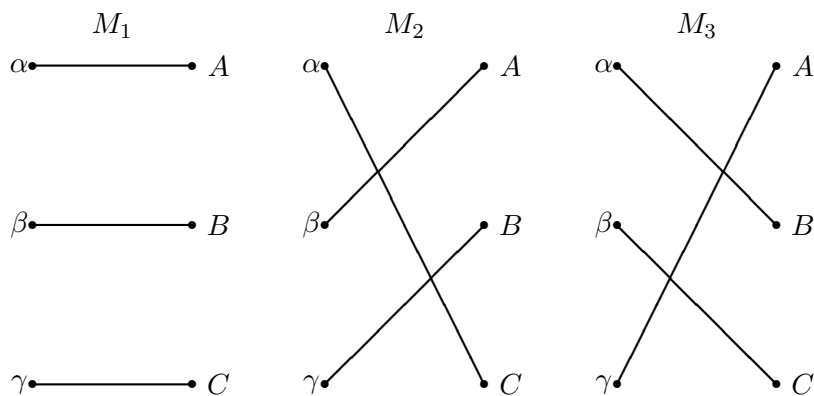
A mátrix egy elemének első koordinátája a megfelelő oszlop által reprezentált nő helyezése a sorhoz tartozó férfi ranglistáján, míg a második koordináta a fordított helyezés.

⁴Néhány éve a magyar felsőoktatási felvételi rendszere is hasonló algoritmust használ.

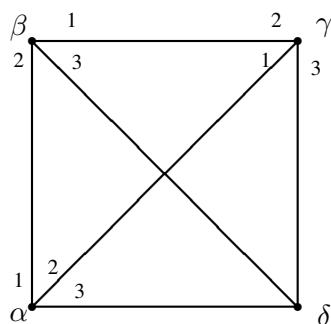
A feladat egy olyan n -elemű M párosítás megadása, amely, legalábbis valamely értelemben, elképzelhető. Gyakorlati és elméleti megfontolások alapján az alábbi definíció tűnik ésszerűnek. Egy M párosítás *instabil* ha vannak olyan α, β férfiak és A, B nők, hogy $(\alpha, A) \in M, (\beta, B) \in M$, de β preferálja A -t B -hez képest, és A preferálja β -t α -hoz képest. Egy M párosítás *stabil*, ha nem instabil.

A definíció motivációja kézenfekvő: feltehető, hogy az instabil esetben β illetve A felbontja pillanatnyi kapcsolatát, és egymással lép kapcsolatra. A célunk egy stabil M párosítás keresése lesz majd, már ha van ilyen egyáltalán. (A korábban említett Halmos-Vaughan modell globális optimumra törekedett, nem véve figyelembe a lokális érdekeket, lehetőségeket. Ezért legfeljebb kikényszeríthető, míg a fenti stabilitás szerint egy M teljes párosítás nem bomlik fel, ha magára hagyjuk a rendszert.)

Kérdés persze, van-e egyáltalán megoldás? A fenti példában három megoldás van: $M_1 = \{(\alpha, A), (\beta, B), (\gamma, C)\}$, $M_2 = \{(\alpha, C), (\beta, A), (\gamma, B)\}$ és $M_3 = \{(\alpha, B), (\beta, C), (\gamma, A)\}$.



Példa: A stabil házasság mintája alapján definiálhatjuk az ún. *szobatárs* problémát. Itt adott $2n$ ember, akiket kétszemélyes szobákba kell telepíteni és az előzőekhez hasonlóan preferenciákkal rendelkeznek. Nyilvánvaló, ha adott négy személy $(\alpha, \beta, \gamma, \delta)$ úgy, hogy α, β és γ preferencia listáján δ az utolsó, α -én β , β -én γ és γ -én α az első, akkor *nincs* stabil párosítás.



A példa fényében kellemes meglepetés az alábbi tétel.

8.5. Tétel (Gale-Shapley). *A stabil házasság problémának mindig van megoldása.*

Bizonyítás. Valójában egy nagyon hatékony, mohó típusú algoritmust adunk, melynek végeredménye bizonyosan stabil párosítás. Tradícionálisan, hisz ez igencsak tradícionális eljárás, a megfelelő köznap kifejezéseket használjuk a leírás során. A eljárás első lépésében minden férfi ajánlatot tesz a kedvencének. Minden nő a legjobb ajánlatot fogadja el, de ez csak annyit jelent, hogy „várakozó listára” helyezi a kérőt. A második lépésben az elutasított kérők újra ajánlatot tesznek, ezúttal a preferencia listájukon 2. helyezett hölgynek. A nők ismét a pillanatnyilag legjobb ajánlatot fogadják el; esetlegesen lecserélve a várakozó listán lévő kérőt. Hasonlóan folytatódik ez a későbbiekben is: egy elutasított (vagy egy várakozó listáról lekerült) férfi a soron következő jelölttel próbálkozik, míg a nők a lehető legjobb jelöltet tartják meg.

Legkésőbb $n^2 - 2n + 2$ lépés elteltével minden hölgy kap legalább egy kérőt, így a várakozó listáján is lesz majd valaki. (Ugyanis ha egy lépésben van olyan nő, aki nem kapott még ajánlatot, akkor lennie kell elutasításnak is ebben a lépésben, illetve egy férfi csak egyszer tesz ajánlatot egy nőnek.)

Mikor minden nő kapott ajánlatot, akkor véget vetünk az eljárásnak, és a pillanatnyi párokat véglegesnek kiáltjuk ki. Megmutatjuk, hogy az így kapott M párosítás stabil. Tegyük fel, hogy van olyan α és A , melyre $(\alpha, A) \notin M$, de α preferálja a párjához képest. Ekkor viszont α valamikor ajánlatot tett A -nak és A elutasította őt, azaz a várakozó listáján α -nál „jobb” személy volt, s ha cserélődött is azóta, csak még jobb lehet később. Így A a párját jobban kedveli, mint α -t, azaz nincs instabilitás. $\square$

Példa:

	A	B	C	D
α	1, 3	2, 3	3, 2	4, 3
β	1, 4	4, 1	3, 3	2, 2
γ	2, 2	1, 4	3, 4	4, 1
δ	4, 1	2, 2	3, 0	1, 4

Az algoritmus végrehajtását egy táblázaton követhetjük. Egy cella baloldali eleme az adott lépésben a sor által kódolt személytől ajánlatot kapó, a jobboldali elem pedig a sor ajánlatát elfogadott személy.

	1. lépés	2. lépés	3. lépés	4. lépés	5. lépés	6. lépés
α	A, A	$\emptyset$, A	$\emptyset$, A	$\emptyset$, $\emptyset$	B, $\emptyset$	C, C
β	A, $\emptyset$	D, D	$\emptyset$, D	$\emptyset$, D	$\emptyset$, D	$\emptyset$, D
γ	B, B	$\emptyset$, B	$\emptyset$, $\emptyset$	A, A	$\emptyset$, A	$\emptyset$, A
δ	D, D	$\emptyset$, $\emptyset$	B, B	$\emptyset$, B	$\emptyset$, B	$\emptyset$, B

A követhetőség kedvéért felvettük a férfiak preferencia listáit, ahol felülvonással jeleztük, ha már történt ajánlat: $\alpha(\bar{A}, \bar{B}, \bar{C}, D)$, $\beta(\bar{A}, \bar{D}, C, B)$, $\gamma(\bar{B}, \bar{A}, C, D)$, $\delta(\bar{D}, \bar{B}, C, A)$. A megoldást az utolsó oszlop jobboldaláról olvashatjuk le:

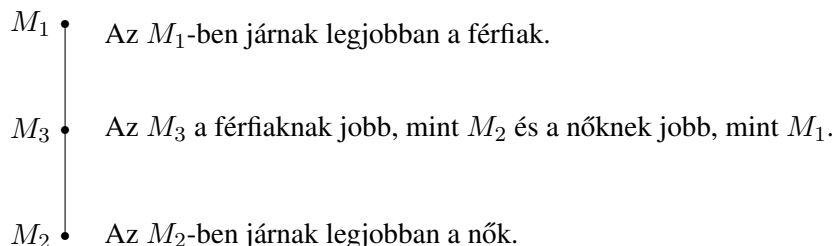
$$M = \{(\alpha, C), (\beta, D), (\gamma, A), (\delta, B)\}.$$

Felmerül a kérdés, tudunk-e valami közelebbit mondani a stabil párosítások szerkezetéről, összehasonlíthatóak-e stb. Vegyünk két stabil párosítást, M_1 -et és M_2 -őt. Az M_1 férfi szempontból jobb, mint M_2 , ha minden férfi legalább olyan jó párt kap M_1 -ben, mint M_2 -ben; jelölésben $M_1 \geq_F M_2$. Nem lehet bármely két stabil párosítást összehasonlítani, de az összes stabil párosítás, mint azt J. H. Conway megmutatta, ún. *disztributív* vagy más szóval *geometriai hálót* alkot.⁵

Speciálisan van legnagyobb és legkisebb eleme. (Ha a nők szempontjából nézzük, ugyanazt a hálót kapjuk, csak megfordítva. Azaz ami az egyik nemnek a legjobb, a másiknak a legrosszabb.) Ez utóbbit egyszerűen beláthatjuk.

⁵Egy L háló, ha van két kétváltozós művelete, $\vee$ és $\wedge$, és ezek *idempotensek* $x \vee x = x$, $x \wedge x = x$, *kommutatívak*, $x \vee y = y \vee x$, $x \wedge y = y \wedge x$, *asszociatívak*, $x \vee (y \vee z) = (x \vee y) \vee z$, $x \wedge (y \wedge z) = (x \wedge y) \wedge z$ és *elnyelők*, azaz $x \vee (x \wedge y) = x$ és $x \wedge (x \vee y) = x$ minden $x, y, z \in L$ esetén. *Disztributív* a háló, ha teljesül még az $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ egyenlőség is.

Példa: Az első példában szereplő stabil párosítások az alábbi hálót alkotják:



Egy A hölgy *lehetséges* az α férfi számára, ha van olyan M stabil párosítás, amelyre $(\alpha, A) \in M$.

8.6. Tétel. Az előző algoritmus minden férfinak a legjobb lehetséges párt adja, míg minden nőnek a legrosszabbat.

Bizonyítás. Az első állítást a lépések szerinti indukcióval látjuk be. Tegyük fel, hogy a soron következő lépésig egyetlen férfit sem utasított el olyan nő, aki lehetséges lett volna számára. Tegyük fel továbbá, hogy ebben a lépésben A elutasítja α -t. Azt állítjuk, hogy ekkor A nem lehetséges α számára. Valóban, ha A pillanatnyi párja β , akkor β preferálja A -t az összes nőhöz képest, kivéve akik korábban visszautasították. Ezek azonban, az indukciós feltétel miatt, nem lehetségesek β számára. Ha tehát létezne egy olyan M stabil párosítás, amelyre $(\alpha, A) \in M$, ebben β a párját (nevezzük B -nek) kevésbé kedvelné, mint A -t, hiszen mind A és B lehetséges β számára. Ekkor viszont az (α, A) , (β, B) instabilitást okoz, azaz A nem lehetséges α számára, s ezzel beláttuk a tétel első felét.

Legyen M^* az algoritmusunk által adott *férfi optimális* megoldása, M pedig egy tetszőleges stabil párosítás. Belátjuk, hogy bármely A nő esetén az ő M -beli párja nem rosszabb, mint az M^* -beli párja. (Igazából pontosan akkor nem rosszabb, ha ugyanaz a párja a két párosításban, és határozottan jobb, ha nem.) Ha $(\alpha, A) \in M^*$, $(\beta, A) \in M$ és $\alpha \neq \beta$, akkor $(\alpha, B) \in M$ valamely $B \neq A$ hölgyre. A tétel első fele miatt persze α preferálja A -t B -hez képest. Másrészt az M párosítás stabil, így speciálisan az (α, B) , (β, A) párok stabilitása az jelenti, hogy A preferálja β -t α -hoz képest; s pont ezt akartuk bizonyítani. $\square$

Megjegyzések: Az algoritmus eredeti felhasználásánál a kórházak tették az ajánlatokat, és azt hangoztatták (természetesen bizonyítás nélkül), hogy ez az orvosok javára válik. Számos tanulság vonható itt le, s ezekből csak az egyik, hogy nem árt meggondolni nyilvánvalónak tűnő (vagy annak beállított) állításokat, mint azt Gale és Shapley tették volt.

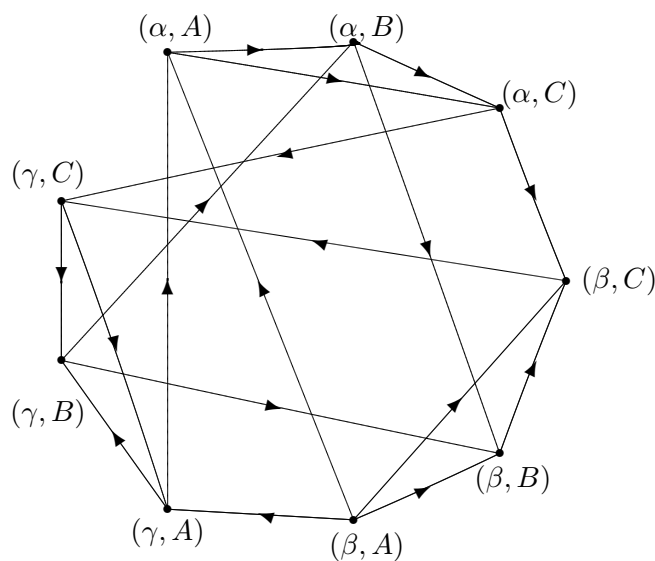
A másik, hasonlóan fontos észrevétel a döntési helyzetek illetve stratégiák buktatóira vonatkozik. A modell azt sugallja, „elébe kell menni” az eseményeknek és kishitűség nélkül megpróbálni a legjobbnak tűnő megoldásokat a „sült galambra várás” helyett.

A stabil párosítás mint mag: Korábban definiáltuk egy irányított G gráf magját; ez egy olyan $S \subset V(G)$ halmaz volt, amely független és domináló egyben. Ez a fogalom különösen fontos a játékelméletben, hisz a megoldások stabilitását fogalmazza meg matematikai formában. A stabil párosítás motiválhatja ezt a definíciót, ugyanis egy rögzített probléma stabil párosításai tulajdonképpen magok egy megfelelően alkotott gráfban.

Definiáljuk egy G gráf *vonalg ráfját*, $L(G)$ -t, a következőképpen: $L(G)$ pontjai G élei lesznek, azaz $V(L(G)) := E(G)$, és $L(G)$ két pontja, e és f , között van él, ha G -ben tekintve az e és f éleknek van közös pontja. Ha G pontjaihoz preferencia listák vannak rendelve, irányítsuk az (e, f) élt $L(G)$ -ben úgy, hogy az a preferált pontból indul és a kevésbé kedvelt pontra mutat közös pont-hoz tartozó lista szerint.

8.1. Állítás. *A fenti definíciókkal a G gráf stabil párosításai éppen az $L(G)$ gráf magjainak felelnek meg.*

Példa: Az első példa G gráfjához tartozó $L(G)$:



8.3. Shannon kapacitás

A következő gráfelméleti probléma érdekes módon az információelméletből ered. Claude Shannon, az *információelmélet* megalapozója a *zajos csatorna modell* megalkotásakor jutott egy, a gráfelmélet nyelvén is megfogalmazható szép feladathoz. Az eredeti probléma a következő. Legyen $\mathcal{X} = \{0, 1\}$ és $\mathcal{C} \subseteq \mathcal{X}^n = \mathcal{X} \times \cdots \times \mathcal{X}$ az ún. kódszavak (0-1 bitsorozatok) halmaza. Tegyük fel, hogy egy kódszót egy zajos csatornán küldve bizonyos karakterek megváltoznak (itt most 0-ról 1-re, vagy 1-ről 0-ra) és emiatt bizonyos kódszavak összetéveszthetővé válnak. Formálisan, legyen $W : \mathcal{X} \rightarrow \mathcal{Y}$ a zajos csatorna melyet a W sztochasztikus mátrix ír le, ahol $W(y|x) \geq 0$ és $\sum_{y \in \mathcal{Y}} W(y|x) = 1$. Ekkor az n -hosszú bitsorozatok esetén $W^n : \mathcal{X}^n \rightarrow \mathcal{Y}^n$, ahol $W^n(y|x) = \prod_{i=1}^n W(y_i|x_i)$. Annak a valószínűsége, hogy egy $\varphi : \mathcal{Y}^n \rightarrow \mathcal{C}$ függvénnyel visszakapjuk az eredeti kódszót hiba nélkül $W^n(y = \varphi^{-1}(x)|x)$. A hiba maximális valószínűsége

$$\text{err}(W^n, \mathcal{C}, \varphi) := \max_{x \in \mathcal{C}} (1 - W^n(\varphi^{-1}(x)|x)).$$

A kérdés az, hogy minél kisebb (pl. 0-hoz tartó) hibavalószínűség mellett mekkora lehet $\mathcal{C}$ mérete, azaz hány kódszavunk lehet maximálisan ahhoz, hogy egyértelműen tudjunk még dekódolni, ha a kód a megérkezése előtt keresztül meg egy zajos csatornán.

Legyen $R \geq 0$ az elérhető érték egy ún. diszkrét memória nélküli csatorna esetén⁶, ha létezik $\{\mathcal{C}_n, \varphi_n\}_n$ sorozat úgy, hogy

1. $\limsup_{n \rightarrow \infty} \frac{1}{n} \log |\mathcal{C}_n| \geq R$
2. $\lim_{n \rightarrow \infty} \text{err}(W^n, \mathcal{C}_n, \varphi_n) = 0$

A $S(W) = \max R$ értékét a csatorna *Shannon kapacitásának*⁷, a $S_0(W) = \sup_n \sup_{\mathcal{C}} \frac{1}{n} \log |\mathcal{C}_n| < S(W)$ értéket pedig a W csatorna *zeró-hiba kapacitásának* nevezzük.

Tekintsük az alábbi mátrixsal megadott W zajos csatornát:

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

⁶

⁷Shannon híres tétele szerint $\max_{P_X} R = I(X; Y)$, ahol $I(\cdot, \cdot)$ a két véletlen változó ún. kölcsönös információ függvénye.

Egyrészt Shannon formulából adódik, hogy $S(W) = \log \frac{5}{2}$, vagyis $S_0(W) \leq \log \frac{5}{2}$. Shannon azt sejtette, hogy $\frac{1}{2} \log \frac{5}{2} \leq S(W)$. A sejtést Lovász László bizonyította, az információelmélet eszköztárából addig távol eső technikával.

A maximális $|\mathcal{C}|$ eléréshez zéró-hiba kapacitás mellett kell, hogy $\varphi^{-1}(x') \cap \varphi^{-1}(x'') = \emptyset$ minden $x', x'' \in \mathcal{C}$ esetén. Természetesen módon adódik a következő gráf bevezetése. Legyen $G = G(W)$ az a gráf, melynek csúcshalmaza $V(G) = \mathcal{X}$, vagyis az ábécé „betűi”, továbbá $(x', x'') \in E(G)$ él pontosan akkor, ha $\varphi^{-1}(x') \cap \varphi^{-1}(x'') \neq \emptyset$, azaz a két kód összetéveszthető. Ezt általánosítva kódszavakra legyen $G^n = G(W^n)$, melynek csúcshalmaza $V = \mathcal{C} \subseteq \mathcal{X}^n$, továbbá $x' = (x'_1, \dots, x'_n)$ és $x'' = (x''_1, \dots, x''_n)$ esetén $(x', x'') \in E(G^n)$ pontosan akkor, ha létezik i index, amelyre $(x'_i, x''_i) \in E(G)$. Világos, hogy G^n -ben a maximális független halmaz mérete a nem összetéveszthető kódszavak számát adja meg. Egy G gráf Shannon kapacitását a

$$S(G) = \limsup_{n \rightarrow \infty} \sqrt[n]{\alpha(G^n)}$$

formulával definiáljuk.

Az előző példánk esetén legyen az ábécénk $C_5 = \{0, 1, 2, 3, 4\} = V(C_5)$. Az $\alpha(C_5^2) = 5$ könnyen ellenőrizhető, vagyis adódik hogy $\sqrt{5} \leq S(C_5)$.

8.12. Feladat. *Mutassuk meg, hogy minden G egyszerű gráfra a $\limsup_{n \rightarrow \infty} \sqrt[n]{\alpha(G^n)}$ határtérék létezik és véges.*

8.3.1. Gráfok Lovász-féle ortonormált reprezentációja

Tekintsük G egy klikkfedését $X_1, \dots, X_k$ klikkekkel. Legyen A mátrix a gráf csúcs-klikk incidencia mátrixa, vagyis $a_{ij} = 1$, ha $i \in X_j$, és $a_{ij} = 0$ különben ($i = 1, \dots, n; j = 1, \dots, k$). Ekkor

$$\chi(\overline{G}) = \min\{1^T x : Ax \geq 1, x \geq 0 \text{ és } x \text{ egész}\}$$

$$\alpha(G) = \max\{y^T 1 : y^T A \leq 1, y \geq 0 \text{ és } y \text{ egész}\}.$$

Az LP gyenge dualitás tételből adódik, hogy $\alpha(G) \leq \chi(\overline{G})$, ahol $\overline{G}$ a G gráf komplementer gráfja⁸, $\chi(G)$ pedig a legkevesebb számú klikk, mellyel G pontjai lefedhetőek. Sőt, meggondolható az is, hogy

$$\alpha(G) \leq S(G) \leq \chi(\overline{G}).$$

⁸ $\overline{G}$ csúcsai megegyeznek G csúcsaival, és két csúcs pontosan akkor szomszédos $\overline{G}$ -ben, ha azok nem szomszédosak G -ben.

Ebből adódik, hogy ha $\alpha(G) = \chi(\overline{G})$, akkor $S(G) = \alpha(G)$. Ez nem olyan ritkán fordul elő, például páros, sőt perfekt gráfok esetén teljesül. A legkisebb gráf, amire ez nem teljesül, a C_5 , vagyis az öt-hosszú kör: $\alpha(C_5) = 2 < 3 = \chi(\overline{C_5})$.

Lovász ötlete a következő. Rendeljünk minden $i \in V(G)$ csúcshoz egy u_i egységvektort úgy, hogy $u_i \cdot u_j = 0$, ha i és j nemszomszédos pontok. A C_5 esetén ezeket a vektorokat úgy képzeljük, mint egy esernyő bordái, továbbá legyen c egységvektor az esernyő nyele. A bordák szabályosan nyílnak, minden pillanatban végpontjaik egy síkra esnek, egy szabályos ötszög csúcsait adják. Alkalmas pozícióban a nyél és az öt borda C_5 egy ortonormált vektorrendszer reprezentációját adják. A reprezentáció értéke

$$\max_{i \in V} \frac{1}{(c \cdot u_i)^2}.$$

Legyen $\vartheta(G)$ a legkisebb érték az összes vektorreprezentáció közül. Az esernyő-reprezentáció bizonyítja, hogy $\vartheta(G) \leq \sqrt{5}$.

8.2. Lemma. $\alpha(G) \leq \vartheta(G) \leq \chi(\overline{G})$ minden G gráf esetén.

8.13. Feladat. Bizonyítsuk be 8.3 lemmát. Ötlet: $1 = c \cdot c = \sum_{i \in I} (c \cdot u_i)^2$, minden $\{u_i\}_{i \in I}$ ortonormált vektorrendszer esetén. I -t szorítsuk meg X -re, ahol X a G egy $\alpha(G)$ méretű független halmaz pontjaihoz tartozó vektorainak halmaza. A második egyenlőtlenséget a klasszikus klikkfedés vektor klikkfedésként való értelmezése adja.

Két $\mathbb{R}^n$ -beli a és b vektor tenzor szorzata

$$a \otimes b = (a_1 b_1, a_2 b_2, \dots, a_n b_n) \in \mathbb{R}^n.$$

8.14. Feladat. Mutassuk meg, hogy ha $a, c \in \mathbb{R}^n$ és $b, d \in \mathbb{R}^m$, akkor $(a \otimes b) \cdot (c \otimes d) = (a \cdot c)(b \cdot d)$.

8.3. Lemma. $S(G) \leq \vartheta(G)$ minden G gráf esetén.

Bizonyítás. Legyen $\{u_i; c\}_{i \in V(G)}$ a G gráf ortonormált vektorreprezentációja c nyéllel, melyhez $\vartheta(G)$ érték tartozik. Ekkor megadható G^ℓ egy ortonormált vektorreprezentációja, amely értéke $\vartheta(G^\ell)$ lesz: a gráf egy $(v_1, v_2, \dots, v_\ell)$ csúcsához tartozó vektor legyen $u_{v_1} \otimes u_{v_2} \otimes \dots \otimes u_{v_\ell}$, míg a nyél legyen $c \otimes c \otimes \dots \otimes c$. A részletek kidolgozásához használjuk a 8.14 feladatot állítását. Az ellenőrzés után adódik, hogy

$$\alpha(G^\ell) \leq \vartheta(G^\ell) \leq \vartheta^\ell(G).$$

□

8.1. Következmény. $S(C_5) = \vartheta(C_5) = \sqrt{5}$.

Érdekesség, hogy a C_7 Lovász féle thetafüggvény értéke az esernyőkonstrukció segítségével különösebb nehézség nélkül meghatározható, ugyanakkor $S(C_7)$ pontos értéke máig nem ismert.

9. fejezet

Síkgráfok

Egy gráf *síkra rajzolható* (például egy papírlapra) ha lerajzolható úgy, hogy bármely két éle nem metszi egymást. Egy gráf *síkgráf* ha létezik síkra rajzolása. Azonnal felmerül a kérdés, hogy van-e egyáltalán olyan gráf, ami nem ilyen. Látni fogjuk, hogy K_5 és $K_{3,3}$, azaz az 5 pontú teljes gráf, és a teljes páros gráf, melynek a két színosztályában 3-3 pont van nem síkgráfok. A fejezetben néhány alapvető eredmény mellett két rendkívül hasznos algoritmikus bizonyítási technikát, nevezetesen a $\Delta - Y$ -transzformációt és a discharging módszert, is bemutatunk.

9.1. Síkgráfok és lerajzolásaik

Síkba rajzolt gráfok esetén fontos fogalom a *tartomány*. A precíz matematikai definíciótól itt eltekintünk, helyette egy egyszerű szemléltetést adunk: a síkra rajzolt gráfot úgy fogjuk fel, mint egy térképet, akkor a tartomány egy országot jelent. A következő nagyon fontos tétel síkra rajzolt gráfok paramétereit között ad meg kapcsolatot

9.1. Tétel (Euler). *Ha adott egy véges összefüggő síkgráf, és $|V|$ a csúcsok, $|E|$ az élek, $|F|$ pedig a tartományok (beleértve a „külső”, végtelen nagy területet is) száma, akkor*

$$|V| - |E| + |F| = 2.$$

9.1. Feladat. *Bizonyítsuk be Euler tételét. 1. javaslat: használjunk teljes indukciót. 2. javaslat: használjuk a gráf duálisának fogalmát.*¹

¹Egy síkba rajzolt G gráf *duálisa* az a G^* gráf, melynek csúcsai G tartományainak felelnek meg, élei pedig G éleinek úgy, hogy egy G^* -beli él G két élszomszédos tartományának megfelelő G^* -beli pontot köt össze. A fogalom jóldefiniált, viszont a síkra rajzolás leírása nem egyértelműen definiálja azt.

9.2. Feladat. Euler tételét használva bizonyítsuk a következő állításokat. Tetszőleges egyszerű síkgráfra

1. $|E| \leq 3|V| - 6;$

2. $|E| \leq 2|V| - 4.$

A fenti egyenlőtlenségekből következik, hogy néhány gráf biztosan nem síkgráf:

9.2. Tétel. A K_5 és $K_{3,3}$ nem síkgráfok.

Egy másik következmény a következő:

9.1. Következmény. Minden egyszerű síkgráf tartalmaz legfeljebb ötfokú pontot.

A síkgráfok karakterizációjával kapcsolatos fő tételt is kimondjuk.

9.3. Tétel (Kuratowski). Egy G gráf akkor és csak akkor síkra rajzolható, ha G -nek K_5 vagy $K_{3,3}$ nem topologikus részgráfja.²

Szorosan kapcsolódik a témához a következő fogalom. Egy G gráf $cr(G)$ metszési száma azt mondja meg, hogy egy gráf legkevesebb hány él átmetszéssel rajzolható le a síkra. Gondoljuk meg, hogy $cr(K_5) = cr(K_{3,3}) = 1$.

9.3. Feladat. Határozzuk meg $K_{4,4}$ metszési számát.

9.4. Feladat (Metszési lemma). Mutassuk meg, ha G egyszerű gráf és $|E| \geq 4|V|$ akkor $cr(G) = 1/64 \cdot |E|^3 / |V|^2$.

9.5. Feladat. Mutassuk meg, hogy a Petersen-gráf metszési száma 2.

²Azaz olyan részgráfja, ahol bármely élt egy tetszőleges hosszú úttal helyettesíthetjük. Másképpen, bármelyik él „belsejében” újabb pontokat vehetünk fel.

9.2. A $\Delta - Y$ transzformáció

Tekintsünk egy *elektromos áramkört*, melyet gráffal reprezentálunk melynek minden (i, j) élén egy R_{ij} ellenállás van. Az alapvető fizikája az áramköröknek négy törvénnyel leírható: Kirchoff törvényei, Ohm törvénye és Joule törvénye. Ezeket itt nem részletezzük, de a törvények alapján nem túl nehezen megmutatható, hogy egy $K_3 = (i, j, k)$ háromszög, melyben az ellenállások R_{ij}, R_{jk}, R_{ik} , helyettesíthető egy $K_{1,3} = (\ell; i, j, k)$ ℓ középpontú csillaggal, ahol $R_{i\ell} = R_{i\ell}R_{ik}/(R_{ij}+R_{jk}+R_{ik})$. Könnyen látható, hogy ezzel az átalakítással az egyes csomópontokban az *eredő ellenállás* nem változik. Az eljárást $\Delta - Y$ transzformációnak nevezzük.

Egy G gráfosztály $\Delta - Y$ -*redukálható*, ha redukálható pontjainak egy részhalmazára $\Delta - Y$ (vagy $Y - \Delta$) transzformációkkal és „soros-párhuzamos” redukciókkal. Az utóbbi a következő lehetőségeket jelenti:

R0: Egy hurokél törlése;

R1: Egy egyfokú pont és a ráilleszkedő él törlése;

R2: *Soros redukció*: egy 2-fokú y pont és a két ráilleszkedő él, (x, y) és (y, z) törlése, majd (x, z) él hozzáadása;

R3: *Párhuzamos redukció*: Egy párhuzamos él törlése.

Mind a 4 redukciós lépés egyel csökkenti a pontok vagy élek számát a gráfban. Az eljárás végén a pontok egy olyan részhalmazát kapjuk, melyeket már nem tudunk törölni további redukciós lépésekkel és transzformációkkal. Ezeket a pontokat *terminális pontoknak* hívjuk.

9.6. Feladat. Mutassuk meg, hogy $K_{4,4}$ nem $\Delta - Y$ -redukálható.

Epifanov 1966-ban bizonyította a következő fontos tételt.

9.4. Tétel (Epifanov). Minden összefüggő síkgráf adott két terminális ponttal $\Delta - Y$ -redukálható a két terminális pontra és egy köztük lévő élre.

9.2. Következmény. Minden összefüggő síkgráf $\Delta - Y$ -redukálható egyetlen pontra.

Egy $e = (x, y)$ él *kontrakciója* az e él törlését és végpontjainak $x = y$ egy pontnak való megfeleltetését jelenti. Egy G gráf *minorja* olyan gráf, amely megkapható G -ből az éltörlés, él kontrakció és izolált pont törlése műveletek sorozatával.

Topologikus részgráf helyett Wagner minorokkal fogalmazza meg ugyanazt, mint Kuratowski:

9.5. Tétel (Wagner). *Egy G gráf akkor és csak akkor síkra rajzolható, ha G -nek K_5 vagy $K_{3,3}$ nem minorja.*

Egy *terminális minor* egy olyan gráf, amit az előző három művelet sorozataként kapunk úgy, hogy a két terminális pont közti élt nem kontraktálhatunk, illetve izolált terminális pontot már nem törölhetünk.

9.6. Tétel (Trueemper; Gitler; Archdeacon et al.). *Legyen H egy terminális minora G -nek. Ha G $\Delta - Y$ -redukálható, akkor H is $\Delta - Y$ -redukálható.*

Egy $\mathcal{G}$ gráfosztály *minorra zárt* ha $G \in \mathcal{G}$ esetén G minden H minora is $\mathcal{G}$ -ben van. Robertson és Seymour mély tétele szerint minden minorra zárt gráfosztály karakterizálható véges sok tiltott minorral (melyek nem részei az osztálynak). Az előző tétel tulajdonképpen azt mondja ki, hogy $\Delta - Y$ -redukálható gráfok osztálya minorra zárt. Epifanov tétele szerint a síkgráfok $\Delta - Y$ -redukálhatók, vagyis karakterizálhatók véges sok tiltott minorral. Ez pedig nem más, mint Wagner tétele. Általánosan ugyanakkor a $\Delta - Y$ -redukálható gráfok osztályának véges kizárt minor karakterizációja nem ismert.

9.3. A discharging módszer

A *discharging* (vagy magyarul „kisütés”) egy rendkívül hatékony algoritmikus bizonyítási módszer a gráfelméletben, továbbá széles körben alkalmazott technika síkgráfok esetén is. A modell lényege, hogy a gráf pontjaira, éleire, vagy tartományaira pozitív vagy negatív töltést helyezünk, és kiszámítjuk a gráf teljes töltöttségét (általában az Euler formula segítségével). Ezután valamilyen kisütési szabályok szerint újraosztjuk a töltéseket (ez a kisütési fázis). Ezután újraszámoljuk a teljes töltöttséget (valamilyen gráftulajdonságot kihasználva), majd a két számolás eredménye közti különbségből vonjuk le a következtetéseket. Nézzünk néhány konkrét példát.

9.3.1. Euler-formula

Látni fogjuk, hogy az Euler-formula a discharging módszer egyik alapvető eleme, ugyanakkor ennek a klasszikus eszköznek is van discharging módszerrel történő bizonyítása.³ Ha adott egy

³A bizonyítás William Thruston-tól származik

véges összefüggő síkgráf, és $|V|$ a csúcsok, $|E|$ az élek, $|F|$ pedig a tartományok (beleértve a külső, végtelen nagy területet is) száma, akkor

$$|V| - |E| + |F| = 2.$$

Rajzoljuk le úgy a gráfot a síkra, hogy egyik él sem vízszintes, vagyis van pontosan egy legalsó L és egy legfelső U csúcsa a lerajzolás szerint. Adjunk minden csúcsnak $+1$, minden élnek -1 és minden tartománynak $+1$ töltést (ld. ábra). A kisütési szabály szerint minden él és csúcs adja át a töltését egy szomszédos tartománynak balról jobbra haladva vízszintes irányban. Meggondolható, hogy minden csúcs, él és tartomány töltése így 0 lesz, kivéve L és U csúcsokat, ahol marad összesen $+2$ töltés.

9.3.2. Egyszerű alkalmazások

Az első példa tulajdonképpen nem is tartalmazza a kisütési fázist, de jól mutatja, hogyan működhet a kezdeti töltések kiosztása és hogyan lehet számolni a gráf teljes töltöttségét.

9.1. Állítás. *Minden síkgráfnak, melynek minden tartománya legalább ℓ oldalú, van olyan csúcsa melynek foka legfeljebb öt.*

Bizonyítás. Tegyük rendre egy d fokú csúcsra $6 - d$ és egy ℓ oldalú tartományra $2(3 - \ell)$ töltést. Ekkor a gráf teljes töltöttsége

$$\sum_{v \in V} (6 - d_v) + \sum_{f \in F} 2(3 - \ell_f) = 6|V| - 2|E| + 6|F| - 4|E| = 6(|V| - |E| + |F|) = 6 \cdot 2 = 12.$$

Emiatt kell legyen olyan v csúcs, amelyre $6 - d_v > 0$, azaz $d_v \leq 5$. □

9.2. Állítás. *Ha egy síkgráfban minden pont foka legalább 3 és minden tartomány legalább 3 oldalú, akkor van olyan d fokú pont mely egy ℓ oldalú tartomány csúcsa (sarka), miközben $d + \ell \leq 8$.*

Bizonyítás. Legyen egy d fokú pont töltése $4 - d$ és egy ℓ oldalú tartomány töltése $4 - \ell$. A teljes töltöttsége ekkor

$$\sum_{v \in V} (4 - d_v) + \sum_{f \in F} (4 - \ell_f) = 4|V| - 2|E| + 4|F| - 2|E| = 4 \cdot 2 = 8.$$

A kisütési fázis a következő. Minden d fokú pont a kezdeti $4 - d$ töltéséből egyenletesen átad $(4 - d)/d$ -t a szomszédos sarokpontoknak. Ekkor minden ilyen pontnak 0 lesz a töltöttsége. Minden tartományra hasonlóan járunk el. Végül minden pont, illetve tartomány töltöttsége vagy 0 vagy pozitív. Vegyünk egy pozitív töltöttségű sarokpontot. Legyen ezzel szomszédos egy d fokú pont és egy ℓ oldalú tartomány. Mivel $d \geq 3$ és $\ell \geq 3$, ezért

$$\frac{4 - d}{d} + \frac{4 - \ell}{\ell} > 0,$$

azaz

$$4\ell + 4d + 2d\ell > 0,$$

ebből pedig

$$\ell < \frac{2d}{2 - d} \leq 6, \text{ illetve } d < \frac{2\ell}{\ell - 2} \leq 6.$$

Innen $d + \ell < \ell^2/(\ell - 2)$, vagyis $d + \ell \leq \max\{8, 7, 8\}$, ha $\ell = 3, 4, 5$. □

Legyen G páros gráf V és W színosztályokkal olyan, hogy minden V -beli csúcs foka 5 és minden W -beli csúcs foka 3. Legyen $\lambda : V \cup W \rightarrow \{1, 2, \dots, 8\}$ a pontok egy színezése úgy, hogy minden V -beli pont színe 1, 2, vagy 3, és szomszédja a 4, 5, 6, 7, vagy 8 színt kapja; visszafelé, egy W -beli pont a 4, 5, 6, 7, vagy 8 színt kapja, és szomszédja az 1, 2, vagy 3 színt kapja.

9.3. Állítás. *A most definiált gráf nem létezik.*

Bizonyítás. Legyen egy d fokú pont töltése $3(4 - d)$ és egy ℓ oldalú tartomány töltése pedig $3(4 - \ell)$. A teljes töltöttség ekkor

$$\sum_{v \in V} 3(4 - d_v) + \sum_{f \in F} 3(4 - \ell_f) = 3(4|V| - 2|E| + 4|F| - 2|E|) = 12 \cdot 2 = 24.$$

Az első kisütési szabály az, hogy minden $w \in W$ pont küldjön egy egység töltést mindhárom szomszédjának. Ekkor w töltése 0 lesz. A második kisütési szabály szerint minden olyan $v \in V$ csúcs küld két egység töltést a legalább 6 oldalú szomszédos tartományoknak. Ha f egy 4 oldalú tartomány, akkor kezdeti töltése 0 és nincs olyan szabály, ami változtatna ezen. Ha $f \ell \geq 6$ oldalú, akkor a kezdeti töltése $3(4 - \ell)$ és a kisütés során legfeljebb $2(\ell/2)$ töltést kap V -beli illeszkedő csúcsokról, vagyis a végső töltése $3(4 - \ell) + \ell = 12 - 2\ell \leq 0$. Minden $v \in V$ csúcs kezdő töltése -3 , majd 5 egység töltést kap az első kisütési szabály miatt, illetve 2 egységet küld a legalább 6 oldalú szomszédos tartományokba. Azaz ha minden $v \in V$ csúcsnak lenne legalább

6 oldalú szomszédos tartománya, akkor v végső töltése 0 lenne. Ennek igazolását az olvasóra bízjuk. Látjuk, hogy a gráf kezdőtöltése 24 volt, ugyanakkor a kisütés utáni töltés nempozitív. Ez az ellentmondás igazolja az állítást. $\square$

9.7. Feladat. *Adott egy síkgráf, találjunk benne 2 szomszédos csúcsot melyek fokszámösszege minimális.*

9.8. Feladat. *Tekintsük a síkon egymást nem metsző körök unióját. Legyen G az a gráf, melynek csúcsai a körök középpontjai és két csúcs összekötött, ha a megfelelő körök érintik egymást. Mutassuk meg, hogy G síkgráf. Fordítva, mutassuk meg, hogy minden síkgráf megkapható egy ilyen geometriai konstrukcióból.*

9.9. Feladat (Fáry-tétel). *Mutassuk meg, hogy bármely egyszerű síkba rajzolható gráf lerajzolható a síkra úgy is, hogy éleit egyenes szakaszok alkotják.*

9.10. Feladat. *Mutassuk meg, hogy egy focilabda nem rakható (varrható) össze csak hatszög alakú bőrdarabokból. Mutassuk meg, hogy ha ötszögeket is használhatunk, akkor már sikerülni fog.⁴ Ekkor mennyi ötszög és mennyi hatszög kellene?*

⁴Ld. például az 1978-as argentin foci vb-re készült hivatalos labda, a Telstar.

10. fejezet

A gráfalapú adatbányászat alapjai

Az utóbbi évtizedekben világossá vált, hogy a „valós” hálózatokat¹ leíró gráfok szerkezete lényegesen különbözik gráfelmélszkek által legtöbbet vizsgált, például perfekt gráfokétól vagy Erdős-Rényi véletlen gráfokétól. A *kisvilág gráfok* felfedezése jelentősen megváltoztatta, kibővítette a gráfelméleti kutatások irányát. Az ilyen gráfok sajátosságai többek közt a rövid átlagos úthossz, jelentős háromszögezetség, illetve általában kicsi élsűrűség². Nem csak ezek a gráfok különböznek a korábban vizsgált gráfoktól, hanem a velük kapcsolatban megfogalmazott kérdések és problémák is.

Nem könnyű feladat egy kisvilág gráf felépítéséhez szükséges információk összegyűjtése vagy éppen annak eldöntése, *hogyan* készítsünk a rendelkezésre álló adatokból gráfot. Ugyanígy, bár számos próbálkozás történt, nincs minden igénynek eleget tevő modell véletlen kisvilág gráfok generálására sem.

A valós alkalmazásokban fellépő méretek miatt időigényes algoritmusok nemigen használhatók, így jobbára meg kell elégedni egyszerűbb heurisztikákkal, melyek sokszor a fizikából kölcsönzött intuíciónak erednek, lásd, többek között, Barabási Albert László, Bollobás Béla és Mark Newman cikkei.

A fejezetet elsősorban Bartalos István és Pluhár András összefoglaló cikke [13] és az ott hivatkozott szakirodalom alapján építjük fel.

¹Ilyen például emberek ismeretségi kapcsolatok hálójá, cégek közti banki átutalások hálózat, tömegközlekedési hálózat, stb.

²Általában cn vagy $cn \log n$, ha n a csúcsok száma.

10.1. Vizualizáció

Míg Eulernek a Königsbergi hidak és az általuk összekötött szárazföldek „absztrakt lerajzolása” (és ezáltal a gráf fogalmának megalkotása) a „hídprobléma” megoldásához kellett, később a *gráfok lerajzolása*, más szóval *vizualizációja*, számos gráfelméleti kérdést vetett fel. Korábban láttuk, hogy a gráfok síkba való lerajzolhatósága nehéz gráfelméleti problémákhoz vezetett. A *metszési szám* problémával, vagyis, hogy egy gráf minimálisan hány él átmetszésével rajzolható le a síkra, többek közt olyan kiemelkedő matematikusok is foglalkoztak, mint Turán, Szemerédi, Trotter, Chvátal, vagy Beck, továbbá számos fontos alkalmazása is ismert.

A modern gráfvizualizációs eljárások célja a gráfok olyan hatékony lerajzolásának és a rajzolás algoritmusának megadása, amely a szemlélő számára bizonyos értelemben a leginformatívabb a lehetséges lerajzolások közül. Természetesen a végtelen sok lehetséges lerajzolást nem lehet összehasonlítani, de konkrét alkalmazások esetén jó heurisztikák, és ezzel együtt rajzadási algoritmusok születtek gráfok vizualizációjára. Mára a gráfok vizuális megjelenítése önálló területté fejlődött. Itt nem is célunk teljes áttekintést adni, ehelyett csak néhány technikát említünk meg.

10.1.1. Erőhatás alapú elrendezés

Az eljárás (melyet a szakirodalomban *force-directed layout*-nak neveznek) a csúcsok kezdeti, általában véletlenszerű elhelyezése után folytonosan kezdi el módosítani a pontok helyzetét valamilyen fizikai törvényszerűség szerinti analógia alapján. Ilyen lehet például a tömegvonzás, mely szerint a „közeleli” (pl. szomszédos) pontok nagyobb erővel hatnak egymásra, ezért közelebb kell, hogy kerüljenek egymáshoz a lerajzolásban, mintha „távol” lennének. A pont „tömegének” pl. a fokszáma, vagy a ponthoz rendelt más, ún. központisági mérték felel meg.

Más megközelítésben a pontok közé rugókat képzelünk el melyek az él a megnyúlással (összenyomódásával) arányos erőt fejtenek ki (Hooke-törvény). Emellett a pontok között taszító erők hatnak, például a Coulomb-törvény alapján, ekkor a pontok „elektromos töltöttsége” és a köztük lévő erőhatások alapján végzett szimuláció „állítja be” a pontok *egyensúlyi* helyzetét. Egy másik megközelítésben

Tisztán gráfelméleti megközelítésben egy (i, j) pontpár δ_{ij} „ideális” távolsága a lerajzolásban arányos a két pont gráfon vett távolságával. A lerajzolás megadása ekkor egy optimalizálási probléma, mely a pontok közti (lerajzolásbeli) Euklideszi és az „ideális” távolság különbségét minimalizálja.

A módszertan legfőbb előnyei az egyszerűség, könnyű implementálhatóság és az intuitív meg-

közelítés. Ugyanakkor a nagy futási idő (jellemzően a pontok számában köbös), illetve nagy pontszám esetén a kapott lerajzolás túlzott függése a kezdeti véletlen lerajzolástól negatívumiként említendő. Két konkrét algoritmust említve a *Fruchterman-Reingold*, illetve a *Kamada-Kawai* algoritmusok kipróbálását az olvasóra bízunk.³

10.1.2. Spektrális elrendezés

Ez a módszer a gráf sajátvektor komponenseit, mint koordinátákat használja a gráf lerajzolásához. Az alapötlet az, hogy a gráf Laplace mátrixának két legnagyobb (vagy legkisebb) sajátértéket kiszámolva a hozzájuk tartozó sajátvektorokat használjuk a pontok síkon való elhelyezéséhez. Ha a térben szeretnénk ábrázolni, akkor három sajátérték-sajátvektor párra lenne szükség. Az elhelyezés a következőképp működik: az i pont ($i = 1, 2, \dots, n$), (melyhez a Laplace mátrix i -edik sora és oszlopa tartozik) első (x -) koordinátája legyen legyen az első sajátértékhez tartozó sajátvektor i -edik eleme, míg a második (y -) koordinátája a második sajátértékhez tartozó sajátvektor i -edik eleme. Fő előnyei a módszerek a gyors kiszámíthatóság, egzaktság és egyenletes pontsűrűség, ugyanakkor bizonyos tulajdonságokat (pl. a gráf közösségszerkezete, ld. később) nem szemléltet hatékonyan.

10.1.3. Réteges elrendezés

Gyakran *Sugiyama lerajzolásnak* nevezik, melyet elsősorban irányított körmentes, vagy közel körmentes gráfok lerajzolására terveztek. A módszer szerint a gráf csucsai függőlegesen elhelyezkedő rétegekbe lesznek rendezve úgy, hogy a legtöbb él fentről lefelé van irányítva a lerajzolásban (ezt pl. a *Coffman-Graham* algoritmussal érhetjük el). Ezután az egyes rétegeken belül a pontokat úgy helyezi el, hogy a rétegek közti élek metszései száma minimális legyen.

10.2. Kisvilág gráfok

Informálisan *kisvilág gráfnak* nevezzük az olyan gráfokat, melyben a pontok fokszáma általában kicsi, ugyanakkor a legtöbb pont csupán néhány lépésből elérhető bármely másik pontból. Egy kisvilág gráf esetén jellemző, hogy a pontok közti átlagos távolság kicsi, és ugyan léteznek nagy

³Ezek elérhetőek többek közt az *iGraph* függvénykönyvtárban, további a *Gephi* és *Cytoscape* ingyenes szoftverekben.

szomszédsággal rendelkező pontok, a legtöbb pont fokszáma kicsi. Ezen felül fontos karakterisztikája, hogy ha (u, v) és (v, w) létező élek, akkor az (u, w) él meglétének a valószínűsége nagyobb, mint egy véletlen gráfban.⁴ Watts and Strogatz nevéhez fűződik az első algoritmus, ami egy n pontú gráfból, melyben minden pont foka azonos, származtat kisvilág tulajdonságú gráfot az élek bizonyos valószínűséggel történő máshová „drótozásával”.

Sokféle módon lehet véletlen gráfokat generálni, melyek megragadják a kisvilág gráfok egy-egy lényeges tulajdonságát. Ezek közül a *Preferential Attachment* (PA) és a *Vertex Copy* (VC) modelleket említjük meg. Megjegyezzük, hogy másfajta megközelítések is vannak, pl. ami a véletlen metszetgráf modellt vizsgálja.⁵

Mindkét modell rekurzívan definiált; egy már meglévő részgráfhoz vesz hozzá egy új x pontot, de az x szomszédságát másképp generálják. A PA modellben az x pont k új élet hoz, ezeket egymástól függetlenül random kötjük a régi pontokhoz; egy y -hoz a $d(y)$ fokszámmal arányos valószínűséggel. A VC modellben egy régi s pontot választunk egyenletes eloszlással, és az új x ponttal az $N(s)$ pontjait p valószínűséggel, egymástól függetlenül összekötjük.

A tapasztalatok vegyesek és többet mondanak a modellekről, mint a valós hálózatokról, szemben például a CPM vagy N^{++} algoritmusok, melyek esetén az jó alkalmazhatóság a fő motiváció.

10.1. Feladat. *Implementáljuk a PA és a VC modelleket. Az input legyen egy G_0 kezdeti gráf és az elérendő pontszám n . Ábrázoljuk is a kapott gráfokat.*

10.3. Gráfok klaszterezése

A gráfok egy hatékony vizualizációja után számos további vizsgálat egyik alapvető feltétele a gráf pontjainak klasszifikációja, csoportokba rendezése. Ez történhet osztályozással, azaz $V(G)$ -t felbontjuk $\{C_i\}_{i=1}^m$ halmazok, ún. *klaszterek* diszjunkt uniójára. A másik megközelítésben nem kívánjuk meg sem a csoportjaink diszjunkságát, sem azt, hogy együtt kiadják $V(G)$ -t. Ezeket az entitásokat szokás közösségeknek hívni; mi itt *közösség* alatt mindig ezeket értjük, míg az osztályozás elemeit klasztereknek hívjuk. Rengeteg erőfeszítés történt a klaszterek előállítására, vizsgálatára, illetve alkalmazására. Annyit megjegyeznénk, hogy a klaszterek előállítására mind ún.

⁴Pl. egy Erdős-Rényi véletlen gráf esetén minden él $0 < p < 1$ valószínűséggel létezik, ugyanakkor szociális hálózatokban megfigyelhető az „ismerősöm ismerősét nagy valószínűséggel én is ismerem” paradigma; ugyanakkor technológiai hálózatok esetén, mint pl. egy energia-ellátó hálózat nem jellemző a háromszög-képződés.

⁵A metszetgráfokra a CPM hajlamos túl nagy közösségeket adni. A lehetséges javítás erre maximálni a közösségek átmérőjét az N^{++} -hoz hasonlóan.

top down (felülről lefelé) és *bottom up* (alulról felfele) építkező algoritmusokat javasoltak. Ezzel szemben a közösségek keresésére szolgáló algoritmusok jobbára az alulról építkezést használják, azaz kisebb közösségek növelésével próbálnak megfelelő eredményhez jutni.

A klaszterezés (és így a közösségkeresés is) elméletileg megalapozhatatlan Jon Kleinberg eredménye szerint, ezért a sokszor követett pragmatikus megoldás marad: veszünk egy ésszerűnek tűnő algoritmust, az eredményét definiáljuk klasztereknek/közösségeknek és megnézzük használhatóságát.

Itt néhány tipikus közösségkereső algoritmust mutatunk be, melyek hasonló elven alapulnak. Az egyik első, ténylegesen használt algoritmus az N^{++} . A k -klick perkolációs algoritmus, a CPM, az első széles körben ismert módszer, melyet Palla és társai szintén valós feladatokra alkalmaztak. Az élek klaszterezése a harmadik - főként elméleti érdekességű. Negyedikként az ún. *címke terjesztés*⁶ módszert mutatunk be. Egy összefoglaló cikk a témában pl. [2].

10.3.1. Az N^{++} algoritmus

Ez egy generikus algoritmus egy tetszőleges $f : 2^{V(G)} \times V(G) \rightarrow \mathbb{R}$ és $c : \mathbb{N} \rightarrow \mathbb{R}$ függvénnyel, ahol $f(A, x)$ jelenti az A közösség és az x csúcs kapcsolatának erősségét. Csatoljuk x -et A -hoz, ha $f(A, x) \geq c(|A|)$. Az algoritmus első fázisa lentől felfelé építkezve megadja a közösségek $\mathcal{K}$ halmazának első közelítését.

1. Adott G, k, c , ahol k a maximális közösségméret, továbbá kezdetben legyen $\mathcal{K} := V(G)$
2. Minden $A \in \mathcal{K}$ és $x \in V(G)$ esetén ha $f(A, x) \geq c(|A|)$ akkor tegyük $A \cup \{x\}$ -t $\mathcal{K}$ -ba (ha még $|A| \leq k$)
3. Töröljük az összes olyan $A \in \mathcal{K}$ -t, amelyre $A \subset B \in \mathcal{K}$ és $A \neq B$.

Ezután a második fázisban összeolvasztjuk a majdnem azonos méretű közösségeket. Legyen C olyan gráf, amelyben $V(C) = \mathcal{K}$ és $(A, B) \in E(C)$, ha $A \cap B$ „elég nagy”. Cseréljük ilyenkor $\mathcal{K}$ -t $(\mathcal{K} \setminus \{A, B\}) \cup \{A \cup B\}$ -ra. Ezután a C elemei legyenek a közösségek. A tapasztalat az alábbi értékeket javasolja. Jelentse a nagy a 60%-át a kisebb halmaz elemszámának. Az $f(A, x)$ értéke az x és A közötti egy és kettő hosszúságú utak számától függ. Tehát hogy megkapjuk az x -et tartalmazó közösségeket, elegendő keresni a $N^{++}(x) := N(N(x))$ halmazban, azaz legfeljebb a másod szomszédok között.⁷

⁶Az irodalomban *label propagation* néven található meg.

⁷Néhány hasonló módszert sorol Fortunato összefoglaló cikke.

10.3.2. k -klikkek perkolációja

Röviden CPM módszer. Itt $k \in \mathbb{N}$ adott, mint az algoritmus paramétere. Miután megtaláltuk az összes k -klikket G -ben, tekintjük azt a Q_k gráfot, melynek csúcsai ezen klikkek és $(A, B) \in E(Q_k)$ pontosan akkor, ha $|A \cap B| = k - 1$. A közösségek Q_k összefüggő komponensei klikkjeinek egyesítései lesznek.

10.3.3. Élek klaszterezése

Klaszterezzük valamilyen módon az élek halmazát. Az egyes klaszterek éleinek végpontjai lesznek a közösségek.

Ezek a módszerek különböznek a talált közösségek típusaiban és a számítási költségeikben is. Jóllehet az élek klaszterezését könnyű végrehajtani, használata mégis jelentős hátrányokkal jár. (pl. a kapott közösségek átfedése legfeljebb egy csúcspont mélységű.) Természetesen az implementációk minősége lényeges szempont. Kisvilág gráfokon az $\mathbb{N}^{++}$ és a CPM is majdnem lineáris időben fut, ami természetes követelmény, ha valódi feladatokkal foglalkozunk.⁸

10.3.4. Általános szemlélet

Vegyük észre, hogy a három felsorolt algoritmus család végrehajtása két lépésből áll. Először egy $\mathcal{F} = (V, \mathcal{H})$ hipergráfot⁹ határoznak meg, ahol $V = V(G)$ és $\mathcal{H} \subset 2^V$. A $\mathcal{H}$ elemei lesznek a közösségek *építőkövei*. A második lépésben $\mathcal{H}$ -t alkalmas d távolságfüggvénnyel ellátva $\mathcal{M} = (\mathcal{H}, d)$ metrikus teret készítünk. Ezután valamilyen klaszterező algoritmussal $\mathcal{M}$ klasztereinek egy $\mathcal{C}$ halmazát kapjuk. Végül a keletkezett klasztereket V részhalmazaival azonosítjuk úgy, hogy egy $C_i \in \mathcal{C}$ -re $K_i := \cup_{H \in C_i} H$, ahol K_i közösség megfelel C_i klaszternek.

A fenti algoritmusoknál $\mathcal{H}$ elemei (az építőkövek) rendre kis sűrűségű részgráfok, k -klikkek illetve élhalmazok. A köztük levő kapcsolatot leíró $\mathcal{D}$ gráfban pontosan akkor van él, ha a kapcsolat szoros. Az első esetben $(K_i, K_j) \in \mathcal{D}$, ha $|K_i \cap K_j|$ elég nagy, a másodikban, ha $|K_i \cap K_j| = k - 1$, míg a harmadik esetben ez paraméter.

Megjegyzés. A fenti általános szemlélet paradigmájába bele nem illő megoldások is lehetségesek.

⁸Ez csúcsok millióit jelenti.

⁹Gakran használt neve: halmazrendszer.

-ben a nagy rangú pontok közül választ egy független halmazt; ezek lesznek a közösségek középei, majd ρ sugarú gömböket képez körülöttük. Távolság függvénynek a G természetes metrikáját használja, amely a ρ paraméter értékétől függően átfedésekhez vezet(het). Egy másik megközelítésben -ben először egy kifinomult hatás függvényt számolnak ki, amely a pontok „központosságának” mértéke. Ennek alapján nívófelületet képeznek, és a felület kiemelkedéseit azonosítják mint közösségeket.

10.3.5. Címke terjesztés

Rahhavan, Albert és Kumara egy ún. címke terjesztési eljárást adtak meg közösségkeresésre. Csak az alapötletet vázoljuk. Legyenek egy x csúcs szomszédai $x_1, x_2, \dots, x_k$, melyek címkéje az a közösség, ahová tartoznak. Az x csúcs abba a közösségbe fog tartozni (vagyis azt a címkét kapja, amelyik közösségbe a legtöbb szomszédja tartozik. Kezdetben minden csúcsnak különböző címkéje van. A címke terjesztési folyamat lehetséges implementációit itt nem tárgyaljuk.

10.2. Feladat. *Adjunk egy implementációt a címke terjesztésre. Kezeljük le azt az esetet is, amikor egy csúcs címkéje nem egyértelmű a szomszédok címkéi alapján.*

10.3.6. Az algoritmusok kiértékelése

Mivel a közösségek (vagy klaszterek) definíciói többé-kevésbé tetszőlegesek, hasznosságuk mérése is sokféle elgondolás született. Jóllehet ez alapvető kérdés, a kutatók nézőpontjai természetesen eltérőek. Az alábbiakban vázoljuk, hogyan lehet egy-egy közösség fogalom használhatóságát megállapítani. Egy *direkt módszer* közvetlenül hasonlítja össze az adódó közösségeket és a gráfról meglevő egyéb információkat, míg az *indirekt módszerek* egy modell változójaként kezelik a közösségi információt, és az előrejelzés pontosításának a mértékén mérik ennek a hasznosságát.

Először futtatni kell az algoritmusokat, meg kell kapni az eredményeket és esetleg matematikai következtetéseket levonni bizonyos gráf osztályokról. Nagyon fontos az algoritmusok sebessége. Valódi sebességüket nem könnyű összehasonlítani, mivel ez erősen függ az implementációjuktól és a tesztráfoktól (gyakorlati gráf avagy elméleti konstrukció).

Mindhárom algoritmus gyors és általában is az általános paradigma szerinti algoritmuscsalád tagjai hatalmas méretű problémák megoldására képesek.¹⁰

¹⁰A futási időkről és a megoldások jóságáról, részletesen lásd Griechisch és Pluhár (2010).

A klikk-perkolációs módszer figyelemre méltó mind elméleti, mind gyakorlati szempontból nézve. Mindazonáltal a CPM néha túl nagy közösségeket ad és a paraméterezése is rejtélyes, hiszen nem világos hogyan döntjük el, milyen értéke legyen k -nak.

Az N^{++} algoritmus meglehetősen heurisztikus, elméleti vizsgálata nem kivitelezhető. Fő előnye a sebesség, a közösségek kis átmérője és a megbízhatóság.

Az él-klaszterező módszereket még kevésbé vizsgálták. Nyilvánvaló hátrányuk, hogy az általuk kapott közösségeknek legfeljebb egy közös elemük lehet. Valódi gráfoknál ez túl szoros feltétel.

A címke terjesztési algoritmus élszámban közel lineáris futási idejű, ami lehetővé teszi nagy (több millió csúcsú) gráfok vizsgálatát.

10.3.7. Modularitás

A Newman-modularitás a G gráf és komponenseinek alábbi függvénye:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j),$$

ahol $m = |E(G)|$, A_{ij} a G adjacencia mátrixa, k_i az i -edik csúcs fokszáma, c_i a komponense és $\delta(c_i, c_j)$ a Kronecker szimbólum. A klaszterező algoritmusok alapulhatnak valamilyen matematikai vagy fizikai heurisztikán¹¹, vagy megpróbálják maximalizálni a modularitási függvényt az összes komponensek halmazán valamilyen mohó algoritmussal.

A modularitásra adott formula általánosítható közösségekre¹², ha s_{ij} -t írunk $\delta(c_i, c_j)$ helyett, ahol s_{ij} valamilyen i és j közötti hasonlósági mérték. (Jelen esetben u_i az i -edik pont valószínűségi eloszlása a közösségek fölött és $s_{ij} = \langle u_i, u_j \rangle$, de lehetne bármely $\|u_i - u_j\|$ norma is.)

Másrészt a közösségek *közvetlenül* is megkaphatók a modularitási függvény értékének maximalizálásával is. Mivel egy kvadratikusan maximalizálást kell elvégezni, ez a megközelítés csak kis gráfok esetén lehetséges, bár így is hasznos benchmark-okat ad. Egy másik út a optimum heuristikákkal való megközelítése, csakúgy, mint a klaszterezés esetén. Egy másik tanulság, hogy a klaszterek és a közösségek szerkezete nem mérhető ugyanazzal a mértékkel, ezért további súlyozást kell használni.

10.3. Feladat. Próbáljunk ki különböző klaszterező algoritmusokat különböző gráfokon.

¹¹Mint pl. edge-betweenness (EB), eigenvectors (EV), label propagation (LP), spin glass (SG), walk trap (WT)

¹²ld. pl. Népusz és tsai (2007).

10.4. Feladat. Adjunk meg heurisztikákat a Newman-féle modularitás függvény maximalizálásán alapuló klaszterezésre. Implementáljuk és hasonlítsuk össze az eredményeket különböző inputgráfok esetén.

10.4. Információ terjedés gráfokon

Az fertőzési (vagy infekciós) modellek a valódi gráfok alkalmazásának középpontjában állnak, de alkalmasat konstruálni nehéz. Fő szempontjai:

- (i) melyik modellt válasszuk,
- (ii) mik a lényeges változók,
- (iii) hogyan határozzuk meg a modell paraméterek értékét.

Itt két fertőzési modellt ismertetünk. A témában a összefoglaló munkákat és az ottani hivatkozásokat ajánljuk.

10.4.1. Független Kaszkád modell (IC)

Adott egy G élsúlyozott gráf, ahol a (v, w) élhez a $p_{v,w}$ valószínűséget társítjuk. Az infekció az alábbi módon történik.

Az első lépésben a fertőzött csúcsok F_1 halmazát tekintjük aktívnak, azaz $F_1 = A_1$. Általánosan a $w \in V(G) \setminus F_{i-1}$ csúcs $p = \prod_{v \in A_{i-1}} p_{v,w}$ valószínűséggel fertőződik meg az i -edik lépésben, és ekkor $w \in F_i$. A frissen fertőzött pontok a *rákövetkező lépésben* fertőzhetnek csupán, azaz $A_i = F_i \setminus F_{i-1}$. Ha valamely i -re $F_i = F_{i-1}$, akkor leáll a folyamat.¹³

Megjegyezzük a pontok fertőzési valószínűségének kiszámítása nehéz probléma, jobbára szimulációkon alapul.

10.5. Feladat. Implementáljuk az IC modell magját. Az input egy G gráf, minden (v, w) él $p_{v,w}$ fertőzési valószínűsége és a kezdetben aktív pontok halmaza. Az output a folyamat végén fertőzött pontok halmaza.

¹³A független kaszkádról, vagy a megalkotói alapján Domingos-Richardson modellről lásd bővebben Domingos, vagy Kempe, Kleinberg és Tardos cikkeit. Megjegyezzük, hogy a modell egy ekvivalens változatát vizsgálta korábban Granovetter.

10.4.2. Lineáris Küszöb modell (LT)

A modellben G gráf (v, w) élének súlya $b_{v,w} \in [0, 1]$, tovább minden v ponthoz tartozik egy $\theta_v \in [0, 1]$ fertőzési küszöbérték. A kezdeti aktív pontok halmaza A_1 , a fertőzés determinisztikusan történik. Az i -edik lépésben az $i - 1$ már fertőzött pontok továbbra is fertőzhetnek, továbbá egy nem fertőzött pont v megfertőződik, ha $\sum_w b_{w,v} \geq \theta_v$, ahol az összegzés az olyan (w, v) éleken történik, ahol w pont már fertőzött (azaz fertőzhet).

A θ küszöb megadása az egyes pontokhoz többféleképpen lehetséges. Alkalmazástól függően lehet egy előre adott érték (pl. minden ponthoz $1/2$), lehet véletlen valamilyen eloszlással, vagy az input része a fertőzés maximalizálási problémához. Az utóbbi esetben feladat az, hogy határozzuk meg azt a kezdetben fertőző k pontot (k fix paraméter), amely pontokból indítva a fertőzési folyamatot a lehető legtöbb pont fertőződik meg.

A modell természetes módon általánosítható. Egy v pont fertőzöttsége egy tetszőleges monoton f_v függvénye v aktív szomszédainak. Legyen $f_v \in [0, 1]$ és $f_v(\emptyset) = 0$. Ha $\theta_v \in [0, 1]$ a v ponthoz tartozó fertőzési küszöbérték, akkor v a t -edik lépésben fertőzötté válik pontosan akkor, ha $f_v(S) \geq \theta_v$, ahol S a v csúcs $t - 1$ -ben aktív szomszédainak halmaza.

Az f_v függvény *aktivációs függvénynek* nevezzük. Figyeljük meg, hogy az LT modell esetén $f_v(S) = \min(1, \sum_w b_{w,v})$.

10.6. Feladat. *Implementáljuk az LT modell magját. Az input egy G élsúlyozott gráf, minden v csúcsához egy $\theta_v \in [0, 1]$ szám és a kezdetben aktív pontok halmaza. Az output a folyamat végén fertőzött pontok halmaza.*

10.4.3. A két modell ekvivalenciája

Talán kissé meglepő, hogy az IC és LT modellek ekvivalensek és explicit módon adható meg a köztük lévő összefüggés. Legyen $p_v(u, S)$ annak a valószínűsége, hogy u megfertőzi egy v szomszédját feltéve, hogy ismerjük $u \notin S \subseteq N(v)$ -t, amely pontok már próbálták fertőzni v -t (sikertelenül). Vegyük észre, hogy az IC modell esetén $p_v(u, S) = p_{u,v}$, S -től függetlenül. Legyen

$$f_v(S) = 1 - \prod_{i=1}^r (1 - p_v(u_i, S_i)),$$

ahol $S = \{u_1, \dots, u_r\}$ és $S_i = \{u_1, \dots, u_{i-1}\}$. Ha feltesszük, hogy v megfertőződése független a szomszédok fertőzési kísérleteinek sorrendjétől, akkor $f_v(S)$ függvény jól-definiált. Megfordítva,

ha adott f_v aktivációs függvény, akkor legyen

$$p_v(u, S) = \frac{f_v(S \cup \{u\}) - f_v(S)}{1 - f_v(S)} \quad (*).$$

10.1. Lemma. *Tegyük fel, hogy $p_v(u, S)$ és $f_v(S)$ olyanok, hogy kielégítik $(*)$ -ot. Ekkor bármely $T \subseteq V$ csúcshalmaz és t időpillanat esetén annak a valószínűsége, hogy t – ben a T halmaz aktív (fertőzött és fertőző) egyenlő a $p_v(u, S)$ fertőzési valószínűségekkel definiált IC, és az $f_v(S)$ aktivációs függvénnyel megadott LT modellben.*

10.7. Feladat. *Bizonyítsuk be 10.1 lemmát.*

10.5. Gráfok $\mathbb{R}^d$ -beli reprezentációi

Végezetül egy újabb, dimenzió-redukciós eljárást mutatunk be gráfok elemzésére Schreinerman és Tucker 2009-es cikke alapján. A módszer ismerős lehet a maximális vágás probléma esetén tárgyalt $\mathbb{R}^n$ -beli reprezentációs eljárásához.

Legyen G egy súlyozott gráf. Szeretnénk a csúcsokat vektorokkal reprezentálni úgy, hogy a vektorok skalárszorzata jól közelíti a megfelelő élek súlyát. Legyen $x_1, x_2, \dots, x_n \in \mathbb{R}^d$ és tegyük fel, hogy $x_i \cdot x_j \in [0, 1]$ ha $i \neq j$. Legyen X az az $n \times d$ mátrix, melynek oszlopai x_i vektorok ($i = 1, 2, \dots, n$). Felhasználva ezeket a vektorokat, generálhatunk egy $G = (V, E)$ n pontú véletlen gráfot, ahol (i, j) élt $x_i \cdot x_j$ valószínűséggel húzunk be. Ha $\mathcal{G}_n$ az összes egyszerű gráf n ponton és X adott, akkor annak a valószínűsége, hogy a $G \in \mathcal{G}_n$ gráfot generáltuk

$$P_X(G) = \left(\prod_{i < j, (i,j) \in E} x_i \cdot x_j \right) \times \left(\prod_{i < j, (i,j) \notin E} (1 - x_i \cdot x_j) \right).$$

Általános megközelítésben x_i vektorokat választhatjuk egyenletesen random egy $\mathbb{R}^d$ feletti eloszlásból. Ekkor megmutatható, hogy több, szociális hálózatok esetén jellemző tulajdonság (mint pl. a kicsi átmérő, hatványtörvényes fokszám-eloszlás és klaszterezettség) nagy valószínűséggel megkapható ezen véletlen skalárszorzat gráfok segítségével.

Ahelyett, hogy véletlen skalárszorzat gráfokat generálunk azt vizsgáljuk, hogy adott gráf esetén mi az az X amely a „legjobban” modellezi a gráfot. Egy lehetséges megközelítés a *maximum likelihood* módszer, azaz keressük $\arg\max_X P_X(G)$ -t. Ez sajnos nagy gráfok esetén nem hatékony.

Ehelyett vegyük a G gráf A szomszédsági mátrixát és legyen

$$f_A(X) = \sum_{i \neq j} (x_i \cdot x_j - a_{ij})^2.$$

A feladat azon X megtalálása, mely esetén f_A minimális. A feladat megoldására különböző optimalizálási módszereket használhatunk.

10.5.1. Geometriai elemzési módszerek

Miután megvan a G gráf R^d -beli vektorokkal való reprezentációja lehetőség adódik a gráf ezen vektorrendszer segítségével való elemzésére. A klasszikus k -közép klaszterezés egy kiterjesztését kaphatjuk.

A klasszikus módszer szerint adott n pont, $p_1, p_2, \dots, p_n$, a síkon, vagy magasabb dimenzióban, melyeket k osztályba szeretnék sorolni a következőképp. Legyen a k osztály középpontja kezdetben tetszőleges. Ezután minden pontot hozzárendelünk a hozzá legközelebb álló középponthoz, ezek az induló klaszterek. Ezután frissítjük a klaszter középpontokat úgy, hogy a klaszter pontjainak geometriai középpontja legyen a középpont. Az eljárást addig folytatjuk amíg egy adott lépésben egyetlen pont sem vált klasztert.

Egy gráf esetén legyen a reprezentáló vektorok $x_1, x_2, \dots, x_d$.

1. Tegyük a vektorokat tetszőlegesen k osztályba, legyenek ezek $P_1, P_2, \dots, P_k$.
2. Minden P_i osztályra legyen $a_i := \sum_{x \in P_i} x$, majd legyen $a_i := a_i / \|a_i\|$ egység-hosszúra skálázva.
3. Minden x_i vektort tegyük abba az P_j osztályba, ahol $a_j \cdot x_i$ (azaz a bezárt szög) minimális ($j = 1, 2, \dots, k$).
3. Ha egy adott lépésben egyetlen vektor sem kerül új osztályba vége, különben folytassuk a 2. ponttal.

Az eljárás végén az egy osztályba tartozó vektorok által reprezentált pontok lesznek egy klaszteren, azaz egy újabb gráf klaszterező eljárást kaptunk.

10.8. Feladat. Gondolkozzunk a vektorreprezentáció további lehetséges gráf-elemzési alkalmazásain.

Irodalomjegyzék

- [1] CORMEN, T. H., LEISERSON, C. E., RIVEST, R. R., ANTAL, I., AND ANDRÁS, B. *Algoritmusok*. Műszaki Könyvkiadó, 2001.
- [2] FORTUNATO, S. Community detection in graphs. *Physics reports* 486, 3-5 (2010), 75–174.
- [3] GASPARIAN, G. Minimal imperfect graphs: a simple approach. *Combinatorica* 16, 2 (1996), 209–212.
- [4] GOLUBIC, M. C. *Algorithmic graph theory and perfect graphs*, vol. 57. Elsevier, 2004.
- [5] HAJNAL, P. *Gráfelmélet*. Polygon, 2003.
- [6] HOCHBAUM, D. S. *Approximation algorithms for NP-hard problems*. PWS Publishing Co., 1996.
- [7] JOHNSON, D. S. Approximation algorithms for combinatorial problems. *Journal of computer and system sciences* 9, 3 (1974), 256–278.
- [8] KLOKS, T. *advanced graph algorithms*. Klops, 2014.
- [9] LAWLER, E. L. *Combinatorial optimization: networks and matroids*. Courier Corporation, 2001.
- [10] LOVÁSZ, L. *Combinatorial problems and exercises*, vol. 361. American Mathematical Soc., 2007.
- [11] LOVÁSZ, L., AND GÁCS, P. *Algoritmusok*. Műszaki Könyvkiadó, 1978.
- [12] PLUHÁR, A. *Játékelmélet*. Szegedi Tudományegyetem, 2011.

- [13] PLUHÁR, A., AND BARTALOS, I. Községek és szerepük a kisvilág gráfokban. *Alkalmazott Matematikai Lapok* 29 (2012), 55–68.
- [14] RADOIČIĆ, R., AND TÓTH, G. The discharging method in combinatorial geometry and the. In *Surveys on Discrete and Computational Geometry: Twenty Years Later: AMS-IMS-SIAM Joint Summer Research Conference, June 18-22, 2006, Snowbird, Utah* (2008), vol. 453, American Mathematical Soc., p. 319.
- [15] SCHRIJVER, A. *Theory of linear and integer programming*. John Wiley & Sons, 1998.
- [16] SPINRAD, J. P. *Efficient graph representations*. American Mathematical Society, 2003.
- [17] VAZIRANI, V. V. *Approximation algorithms*. Springer Science & Business Media, 2013.