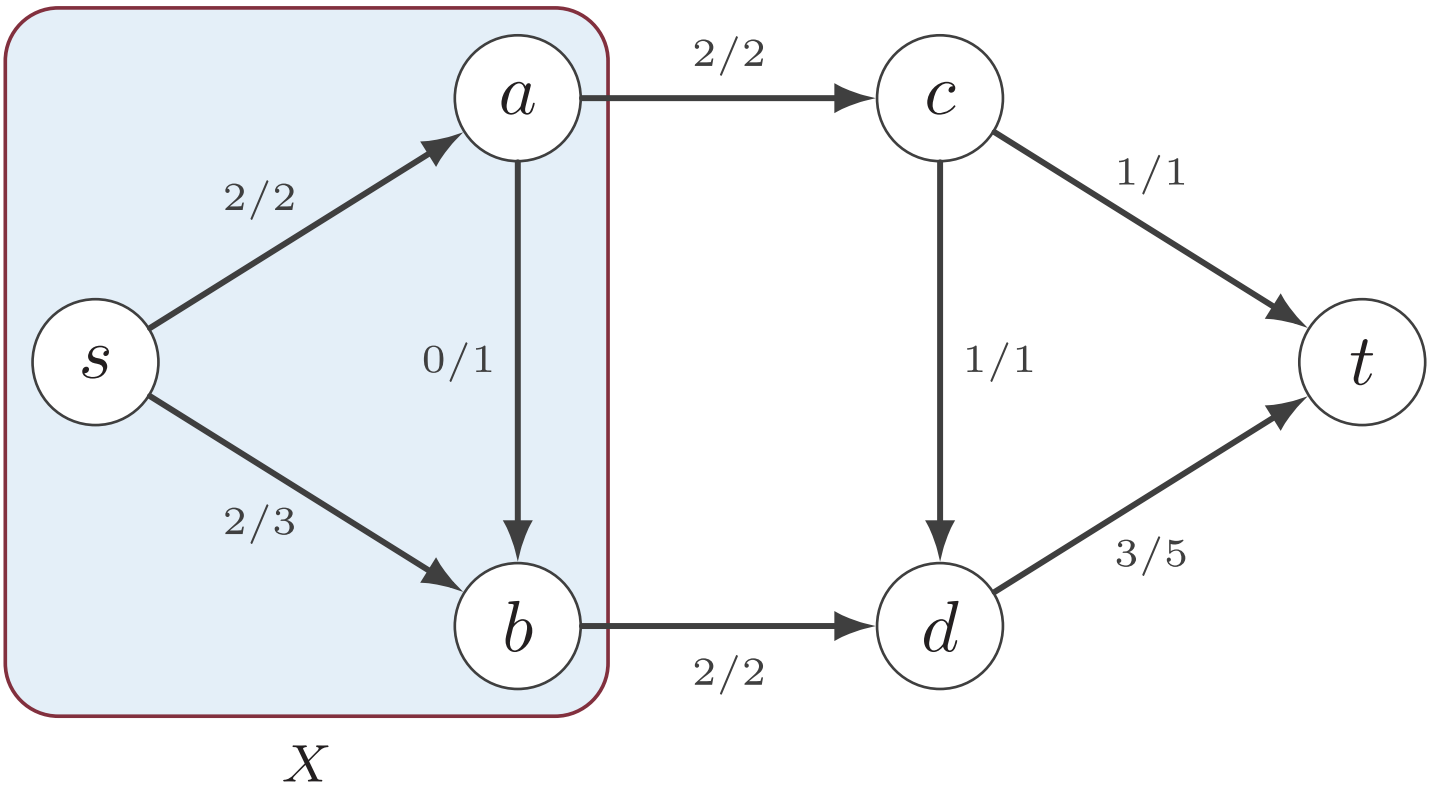
Jordán Tibor

KOMBINATORIKUS OPTIMALIZÁLÁS



KOMBINATORIKUS OPTIMALIZÁLÁS

Jordán Tibor[1]

Jegyzet az ELTE I. éves Matematika BSc-szakos hallgatóinak
2026. október 6.

[1] ELTE TTK, Matematikai Intézet, Operációkutatási Tanszék. email: tibor.jordan (at) ttk.elte.hu.
Készült az ELTE jegyzetpályázatának támogatásával.

A kötet megjelenését az Eötvös Loránd Tudományegyetem Oktatási és Képzési Tanácsának tankönyv- és jegyzettámogatási pályázata támogatta.

© Jordán Tibor, 2026

ISBN (pdf) 978-615-134-042-4



ELTE | EÖTVÖS KIADÓ
Az ELTE hivatalos könyvkiadója



ELTE | TTK
TERMÉSZETTUDOMÁNYI KAR

eltebook.hu

Felelős kiadó: az ELTE Természettudományi Kar dékánja
Projektvezető: Egresi Nóra
Korrektor: Horváth-Márjánovics Diána
Műszaki szerkesztő: Balogh Katalin

Tartalomjegyzék

1. Előszó **5**

2. Alapfogalmak **7**

3. Gráfok bejárása **9**
3.1. Elérhető pontok . . . 9
3.2. BFS és DFS . . . 10
3.3. Elvágó pontok keresése DFS-algoritmussal . . . 11
3.4. A 2-SAT probléma . . . 13
3.5. Minimális élszámú 2-élösszefüggő feszítőgráf . . . 14
3.6. Feladatok . . . 15

4. Irányítások **17**
4.1. Erősen összefüggő irányítások . . . 17
4.2. Feladatok . . . 19

5. Legrövidebb utak **21**
5.1. Dijsktra algoritmusa . . . 21
5.2. A Bellman-Ford-algoritmus . . . 22
5.2.1. Egy alkalmazás . . . 23
5.3. A PERT módszer . . . 24
5.4. Feladatok . . . 24

6. Minimális feszítőfák **27**
6.1. Mohó erdők . . . 27
6.2. Megbízható hálózatok . . . 28
6.3. Feladatok . . . 28

7. Közelítő algoritmusok **29**
7.1. Minimális lefogó ponthalmaz . . . 29
7.2. A Steiner-fa feladat . . . 30
7.3. Feladatok . . . 32

8. Párosítások és élszínezések **33**
8.1. Páros gráfok élszínezései . . . 33
8.2. Az órarendkészítési feladat . . . 34
8.3. Feladatok . . . 34

9. Hálózati folyamok **37**
9.1. Folyamok és vágások . . . 37
9.2. A növelő utak módszere . . . 38
9.2.1. Skálázási technika . . . 40
9.2.2. Legrövidebb növelő utak . . . 41
9.3. Folyamok alkalmazásai . . . 42
9.3.1. A szállítási feladat . . . 42
9.3.2. Projekt kiválasztás . . . 43
9.3.3. Nyerhet-e a csapatunk? . . . 44
9.3.4. Egy kombinatorikus alkalmazás . . . 44

10. Gráfok összefüggősége **47**
10.1. Menger tétele . . . 47
10.2. Folyamekvivalens fák . . . 49
10.3. Gráfok fülfelbontásai . . . 49
10.4. Feladatok . . . 51

11. Poliéderes kombinatorika **55**
11.1. Az LP feladat . . . 55
11.2. Diszkrét optimalizálás és LP . . . 57
11.3. Az IP feladat . . . 58
11.4. Poliéderes kombinatorika . . . 59
11.5. A párosítás politóp . . . 60
11.6. Minimális súlyú lefogó ponthalmaz . . . 62
11.7. Feladatok . . . 64

12. Dinamikus programozás **67**
12.1. A részösszeg probléma . . . 67
12.2. A hátizsák feladat . . . 68
12.3. Minimális háromszögelés . . . 68
12.4. Mátrixok szorzása . . . 69
12.5. Legrövidebb utak minden pontpárra . . . 69
12.6. Legnagyobb konvex részhalmaz . . . 70
12.7. Feladatok . . . 71

13. Összefoglalás **73**

Irodalomjegyzék **75**

1. fejezet

Előszó

Ezen jegyzet célja a kombinatorikus optimalizálás alapvető fogalmainak, eredményeinek, algoritmusainak és módszereinek bemutatása. A jegyzet elsősorban az ELTE TTK matematika alapszakos hallgatói számára készült és lefedi az első tanév tavaszán tartott *Kombinatorikus optimalizálás* előadás anyagát. Feltételezzük, hogy az olvasó elvégezte az ezt megelőző őszi félévben szereplő *Kombinatorika* tárgyat. Ezzel együtt a jegyzet előismeretek nélkül is feldolgozható.

Az alapképzés további részében több tárgy is használja és továbbfejleszti az itt bemutatott ismereteket, így például az *Operációkutatás*, az *Ütemezéselmélet*, az *Algoritmusok tervezése*, valamint az *Optimalizálás a gyakorlatban* órák.

Bérczi-Kovács Erika lektor segítő megjegyzéseit ezúton is köszönöm. A tárgy eddigi gyakorlatvezetői, Bérczi-Kovács Erika, Gehér Panna, Pap Gyula, Schwarcz Tamás és Tóthmérész Lilla számos értékes megjegyzéssel és feladattal járultak hozzá a jegyzet elkészítéséhez.

Budapest, 2026. október 6.

2. fejezet

Alapfogalmak

A *Kombinatorika* órán bevezetett alábbi fogalmakat ismertnek tekintjük: gráf, pont, él, irányított gráf (röviden digráf), fa, levél, kör, út, séta, szomszéd, fokszám, összefüggő gráf, komponens.

Ebben a rövid fejezetben néhány további, a jegyzetben sokszor előforduló alapfogalmat definiálunk. Az angol változatot is feltüntetjük, zárójelben.

Szomszédsági mátrix (Adjacency matrix) Legyen a $G = (V, E)$ gráf ponthalmaza $\{v_1, v_2, \ldots, v_n\}$. A G szomszédsági mátrixa az az $n \times n$ méretű $A(G)$ négyzetes mátrix, amelyben az i-edik sor j-edik eleme pontosan akkor 1, ha $v_iv_j \in E$, különben 0. A $D = (V, A)$ digráf esetén ezen a helyen $v_iv_j \in A$ esetén -1, $v_jv_i \in A$ esetén $+1$ szerepel.

Éllista (Edge list) A gráf éllistáján először a v_1 szomszédainak az indexei szerepelnek növekvő sorban, majd a v_2 szomszédai, és így tovább az összes pontra, egymástól egy-egy karakterrel elválasztva. (Pl. egy háromszög esetén ez 23|13|12.)

Elérhető pontok (egy adott pontból) (Reachable vertices from a given vertex) Az u pontból a v elérhető a G gráfban (ill. a D digráfban), ha vezet u-ból v-be (irányított) út.

Erősen összefüggő digráf (Strongly connected digraph) A $D = (V, A)$ digráf erősen összefüggő, ha minden pontjából minden másik pont elérhető.

Digráf alapgráfja (Underlying graph of a digraph) A $D = (V, A)$ digráf alapgráfja az a $G = (V, E)$ irányítatlan gráf, amelyet a D-ből úgy kapunk, hogy minden $uv \in A$ irányított élt egy uv irányítatlan éllel helyettesítünk.

Fenyő, fenyves, gyökér (Arborescence, branching, root) A $D = (V, A)$ digráf fenyő, ha az alapgráfja egy fa, és létezik egy olyan $s \in V$ pont, melyből minden más pont elérhető D-ben irányított úton. Ez az (egyértelmű) s pont a fenyő gyökere. A fenyves pontdiszjunkt fenyők uniója.

Kifok, befok (Out-degree, in-degree) A $D = (V, A)$ digráfban egy $X \subseteq V$ halmaz kifoka (jelölése $d^+(X)$, néha $\delta(X)$) azon élek száma, amelyek töve X-ben van, feje $V - X$-ben. A befok (jelölése $d^-(X)$, néha $\rho(X)$) definíciója hasonló, megegyezik a $V - X$ kifokával. Egypontú halmaz (azaz egy $v \in V$ pont) esetén röviden $d^+(v)$ és $d^-(v)$ jelöli a v kifokát és befokát.

Vegyes gráf (Mixed graph) A $D = (V, E \cup A)$ vegyes gráfban az élhalmaz irányítatlan (E) és irányított (A) éleket is tartalmazhat.

Egyirányú vágás (One-way cut) A $D = (V, E \cup A)$ vegyes gráfban az $\emptyset \neq X \subsetneq V$ ponthalmaz egyirányú vágás, ha minden olyan él, amelynek egyik vége X-ben, a másik $V - X$-ben van, egy X-be belépő irányított él.

Elvágó él (Cut-edge) A G összefüggő gráfban az e él elvágó, ha $G - e$ nem összefüggő.

Elvágó pont (Cut-vertex) A G összefüggő gráfban a v pont elvágó, ha $G - v$ nem összefüggő.

Távolság (Distance) A G gráfban az u és v pontok távolsága az u-ból v-be menő utak hosszainak minimuma, ahol az út hossza a benne levő élek száma. Ha nincs ilyen út, akkor a távolság $+\infty$. Digráfra a definíció hasonló, irányított utakkal.

Páros gráf (Bipartite graph) A $G = (V, E)$ gráf páros, ha a ponthalmaza két diszjunkt részre osztható úgy, hogy a gráf összes éle ezen részek között menjen. Ha a kettéosztás $V = A \cup B$, akkor a gráfot gyakran $G = (A, B; E)$ alakban adjuk meg.

Lefogó ponthalmaz (Vertex cover) A $G = (V, E)$ gráfban a $T \subseteq V$ ponthalmaz lefogó, ha a G minden élének legalább az egyik végpontja T-ben van.

k-élösszefüggő gráf (k-edge-connected graph) A $G = (V, E)$ gráf k-élösszefüggő, ha bármely $F \subseteq E$, $|F| \leq k-1$ élhalmazra $G-F$ összefüggő, azaz k-nál kevesebb él törlésével nem szüntethető meg a gráf összefüggősége. ($k = 1$ esetben ez éppen a jól ismert összefüggőségnek felel meg.)

k-pontösszefüggő gráf (k-vertex-connected graph) A $G = (V, E)$ gráf k-pontösszefüggő, ha $|V| \geq k + 1$ és bármely $S \subseteq V$, $|S| \leq k - 1$ ponthalmazra $G - S$ összefüggő, azaz k-nál kevesebb pont törlésével nem szüntethető meg a gráf összefüggősége. ($k = 1$ esetben ez éppen a jól ismert összefüggőségnek felel meg.)

Párosítás (Matching) Páronként pontdiszjunkt élek halmaza.

Euler-séta (Euler tour, Eulerian walk) Egy olyan séta a G gráfban (vagy a D digráfban), amely minden élet pontosan egyszer tartalmaz.

Feszítő részgráf (Spanning subgraph) A $G = (V, E)$ gráf olyan részgráfja, melynek ponthalmaza V.

Feszített részgráf (Induced subgraph) A $G = (V, E)$ gráfban egy $X \subseteq V$ ponthalmaz által feszített részgráf az a részgráf, melynek ponthalmaza X, élhalmaza a G összes olyan éle, amelynek mindkét végpontja X-ben van. Jele $G[X]$.

Megjegyzés a jelölésekkel kapcsolatban: az egyszerűség kedvéért a halmazokra vonatkozó különbség és unió műveleteket gyakran kivonás és összeadás műveletekkel jelöljük. Ennek megfelelően például a $G = (V, E)$ gráf, $X \subseteq V$ és $s \in V$ esetén $V - X$ jelöli az X ponthalmaz komplementerét V-ben és $X + s$ jelöli az X ponthalmaz és az s pont unióját. Hasonlóan, $G - S$ az X ponthalmaz törlésével kapott gráfot jelenti, valamint $G + e$ és $G - e$ jelöli azokat a gráfokat, amelyeket a G-ből kapunk az e él hozzáadásával, illetve törlésével.

3. fejezet

Gráfok bejárása

3.1. Elérhető pontok

Legyen a $D = (V, A)$ irányított gráfban $s \in V$ egy kijelölt pont. Jelölje $S \subseteq V$ az s-ből irányított úton elérhető pontok halmazát. Hogyan lehet S-et algoritmikusan megtalálni?

Erre a következő módszert vezetjük be, amely a pontok címkézését használja. A D minden $v \in V$ pontjához két címke tartozzon, egy E- és egy A-címke: az előbbi tartalma vagy **elért**, vagy **nem elért**, az utóbbi tartalma vagy **átvizsgált**, vagy **nem átvizsgált**. Kezdetben az s pont E-címkéje **elért**, minden más ponté **nem elért**, valamint az összes pont A-címkéje **nem átvizsgált**.

Az algoritmus a $D = (V, A)$, $s \in V$ inputtal és ezzel a kezdeti címkézéssel indul, majd az alábbi általános lépést (nevezzük kereső szubrutinnak) ismétli, amíg le nem áll:

Kereső szubrutin: Válasszunk egy olyan $u \in V$ pontot, amely **elért** és **nem átvizsgált**. Ha ilyen pont nincs, az algoritmus leáll. Ha van olyan uv éle a digráfnak, amelyre a v pont **nem elért**, akkor a v E-címkéjét **elért**-re változtatjuk. Ha nincs ilyen él, akkor az u pont A-címkéjét **átvizsgált**-ra változtatjuk.

Mivel minden pontra igaz, hogy mindkét címkéje legfeljebb egyszer változik (az s pont E-címkéje pedig egyszer sem), ezért a Kereső szubrutint legfeljebb $2|V| - 1$-szer hajtjuk végre, mielőtt az algoritmus leáll. A szubrutin végrehajtása (az u, majd a v pont megkeresése, a címkék változtatása) a D pontszámával arányos elemi lépést (címke leolvasás, annak ellenőrzése, hogy két pont között van-e él) igényel a szomszédsági mátrix vagy az éllista ismeretében. A futási időt (azaz a fenti algoritmus által végzett elemi lépések számát) ezért felülről becsülhetjük $|V|^2$, vagy kicsit ügyesebb számolással $|V| + |A|$ konstansszorosával.

3.1.1. Lemma. *Az algoritmus leállásakor az elérhető pontok S halmaza megegyezik azon pontok halmazával, melyek E-címkéje* **elért**.

Bizonyítás: Az egyik irány (minden **elért** pont S-ben van) a szubrutin végrehajtásának számára vonatkozó indukcióval adódik: kezdetben minden **elért** pont (csak az s ilyen) S-ben van. Ha a szubrutin egy v pontot **elért**-nek címkéz, akkor vezet v-be egy uv él egy olyan u pontból, amely már **elért**, azaz indukció miatt S-beli. Ekkor v is S-ben van.

A másik irányt (a végén minden S-beli E-címkéje **elért**) indirekt látjuk be. Ha van olyan $v \in S$, amely a leálláskor **nem elért**, akkor tekintsünk egy P utat s és v között D-ben. Ezen van olyan él, mondjuk $u'v'$, amely kilép az **elért** pontok halmazából. (Figyeljük meg, hogy az

algoritmus leállásakor az **elért** pontok halmaza megegyezik az **átvizsgált** pontok halmazával). Ez azonban azt jelentené, hogy az algoritmus hibázott, amikor az u' pont A-címkéjét **átvizsgált**-ra változtatta. •

Gyakran nemcsak azt szeretnénk eldönteni, hogy egy adott $t \in V$ pont benne van-e S-ben, hanem egy utat is keresünk s és t között. Ehhez segít, ha egy harmadik, F-címkét is bevezetünk, és a Kereső szubrutint kiegészítjük egy további lépéssel. Az F-címke minden v pontra vagy üres, vagy egy v-be vezető él van benne. Kezdetben minden F-címke üres.

Az algoritmus futása során minden olyan v pontra, amelynek E-címkéje **elért**-re változik (mert egy u **elért** pontból vezet bele él), ezzel egy időben kerüljön be az F-címkéjébe az uv él.

Megfigyelhetjük, hogy ezen élek halmaza (a futás során végig) egy fenyőt alkot az (addig) **elért** pontokon. Az így kapott fenyő az algoritmus leállásakor minden $t \in S$-beli pontra eltárol egy s-ből odavezető P utat, melyet könnyen megkaphatunk a t-ből "visszafelé haladva", az aktuális pont F-címkéjében levő élen továbblépve.

3.1.2. Lemma. *Az algoritmus leállásakor a pontok F-címkéjében levő élek egy s gyökerű fenyőt alkotnak, melynek ponthalmaza S.* •

A fenti algoritmus irányítatlan gráfokra is alkalmazható a természetes módosításokkal. Ekkor az s-ből elérhető pontok S halmaza annak a C összefüggő komponensnek a ponthalmaza lesz, amely az s pontot tartalmazza, az F-címkékben levő élek pedig a C egy feszítőfáját alkotják.

3.2. BFS és DFS

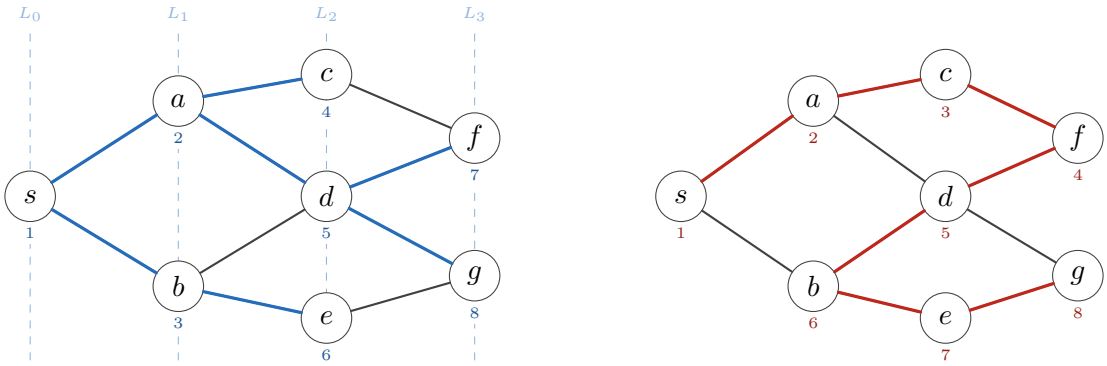
Az elérhető pontokat meghatározó algoritmus futása során a Kereső szubrutin tetszőleges **elért** és **nem átvizsgált** u pontot választhat. Továbbá az algoritmus természetes módon meghatározza az (S-beli) pontok **elérési sorrendjét** annak megfelelően, hogy hányadik szubrutin futtatás során lett az E-címkéjük **elért**.

Egy lehetséges stratégia ennek alapján azt az u pontot kiválasztani (az **elért** és **nem átvizsgált** pontok közül), amelyik a legkorábban vált **elért**-té, azaz az elérési sorrendben (ezen pontok közül) legelöl van. Ha a Kereső szubrutint ezzel a megszorítással egészítjük ki, akkor **szélességi keresésről** beszélünk (az angol breadth first search elnevezés nyomán röviden BFS). A BFS például alkalmas arra, hogy a pontok s-től való távolságát meghatározza. Ennek érdekében a futás során minden $v \in V$ pontra fenntartunk egy $d(v)$ értéket is. Kezdetben az s-re legyen ez az érték 0, minden más pontra $+\infty$. Amikor az algoritmus során egy v pontot egy uv él mentén valamely u pontból először elérünk és E-címkéje **elért** lesz, akkor a $d(v)$ értéket $d(u) + 1$-re változtatjuk.

Nem nehéz megmutatni, hogy az algoritmus során minden, már elért v pontra a $d(v)$ értéke a v pont s-től való távolságával egyezik meg.

3.2.1. Feladat. *Igazoljuk, hogy a BFS-algoritmus helyesen határozza meg az s-től való távolságot, azaz a gráf minden v pontjára az algoritmus leállásakor a $d(v)$ értéke megegyezik a legrövidebb út hosszával s és v között.*

Egy másik lehetséges stratégia az, amikor az algoritmus azt az u pontot választja ki, amelyik a legkésőbb lett **elért**. Ebben az esetben az eljárást **mélységi keresésnek nevezzük** (depth first search: DFS).



3.1. ábra. Egy s gyökerű BFS és DFS feszítőfa ugyanabban a gráfban. A pontok melletti számok az elérési sorrendet adják meg. A BFS fa esetén az s-től számított távolság szerinti szinteket is jelöljük szaggatott vonalakkal.

Ha BFS vagy DFS keresésről van szó, akkor a keresés során kapott s gyökerű fenyőt (digráf esetén), illetve fát (gráf esetén) BFS (vagy DFS) fenyőnek, illetve BFS (vagy DFS) fának nevezzük.

Gyakran gyökeres fákat vizsgálunk: ekkor a fa egyik kitüntetett pontja a gyökér. Keresési fák, pl. BFS vagy DFS fa esetén a gyökérpont eleve adott, de más esetben is hasznos lehet egy gyökeret kijelölni. Egy s gyökerű T fa esetén a fa minden v pontjára definiálhatjuk a v **gyerekeit** (azaz a v azon szomszédait T-ben, amelyek nincsenek rajta az egyértelmű s-v úton), valamint a v **gyökerű részfát** (amely tartalmazza v-t, valamint a T azon pontjait, amelyek a v-ből az egyik gyerekén áthaladó úton érhetők el T-ben).

Legyen a G gráfban a T egy s gyökerű feszítőfa. A gráf egy xy nem-fa (azaz nem a T-hez tartozó) éle **keresztél**, ha a T fában az x és y közötti egyértelmű út s-hez (a fában) legközelebbi pontja különbözik x-től és y-tól. Hasznos megfigyelés az alábbi:

3.2.2. Lemma. *Legyen T a G összefüggőgráf s gyökerű DFS fája. Ekkor nincs G-ben keresztél.* •

3.3. Elvágó pontok keresése DFS-algoritmussal

A G összefüggő gráf valamely v pontja **elvágó pont**, ha $G - v$ nem összefüggő. G kétszeresen összefüggő, vagy röviden **2-összefüggő**, ha legalább 3 pontja van és nincs elvágó pontja. Ha a gráf minden v pontjára egy (pl. DFS vagy BFS) kereséssel ellenőrizzük, hogy $G - v$ összefüggő-e, megkapjuk az összes elvágó pontot, és így eldönthetjük azt is, hogy G 2-összefüggő-e vagy sem. Van ennél hatékonyabb módszer is, melyhez mindössze kétszer kell a gráfot bejárni.

Legyen T egy rögzített DFS bejárás fája a $G = (V, E)$ összefüggő gráfban, az r gyökérrel. Azt mondjuk, hogy az u pont **korábbi** a v pontnál, ha a bejárás során előbb vált elértté. Jelölje ezt $u \prec v$. Ha korábbi, vagy ugyanaz, akkor a $u \preceq v$ jelölést használjuk. A T fa u pontban gyökerező részfáját T_u-val jelöljük (tehát minden $v \in V(T_u)$ pontra $u \preceq v$). Tekintsük azokat a pontokat, amelyekbe vezet G-beli nem-fa él a T_u részfából: legyen

$$X_u = \{x \in V - V(T_u) : \text{ létezik } xy \in E - E(T), \text{ melyre } y \in V(T_u)\}$$

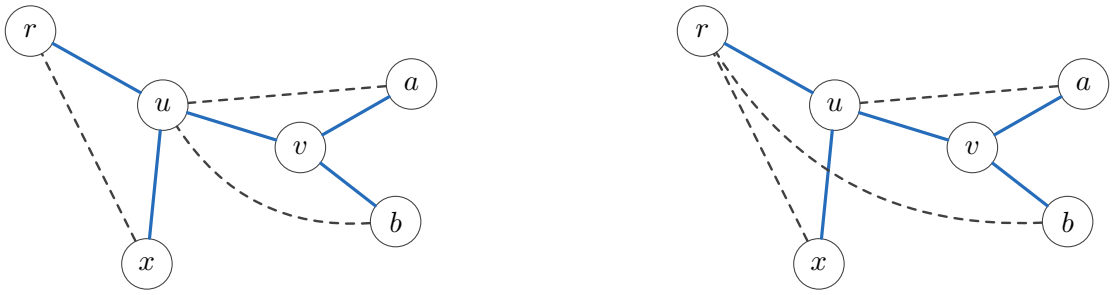
Az alábbi lemma a 3.2.2 Lemmából következik.

3.3.1. Lemma. *Az X_u minden pontja a T-beli r-u úton van.* •

Legyen $f(u)$ az X_u azon pontja, amely a T-beli r-u úton legközelebb van r-hez (ha X_u nemüres), illetve legyen $f(u) = u$, ha X_u üres (azaz T_u-ból nem lép ki nem-fa él). Az algoritmus az $f(u)$ függvényt használja. Helyessége a következő lemmából adódik.

3.3.2. Lemma. *Legyen T a G összefüggő gráf r gyökerű DFS fája. Ekkor*
(i) az r pont pontosan akkor elvágó pont, ha egynél több gyereke van T-ben,
(ii) egy $u \neq r$ pont pontosan akkor elvágó pont, ha van olyan v gyereke T-ben, melyre $u \preceq f(v)$.

Bizonyítás: Az (i) tulajdonság világos. A (ii) tulajdonság igazolásához először tegyük fel, hogy v az u gyereke és $u \preceq f(v)$. Ekkor az f definíciója miatt (és mivel G-ben nincs keresztél T-re nézve) a T_v ponthalmazából kifele csak az u ponthoz megy él G-ben. Mivel u nem a gyökér, így elvágó pont. A fordított irány indoklásához tegyük fel, hogy u elvágó, de minden v gyerekéhez $f(v) \prec u$. Belátjuk, hogy ekkor r-ből minden (u-tól különböző) ponthoz vezet olyan út a gráfban, mely nem használja az u pontot. A T_u-n kívüli pontokra ez világos, ilyen út magában a T fában is van. Legyen most w a T_u egy pontja, $w \neq u$. Ez u valamely v gyerekére benne van T_v-ben. Mivel $f(v) \prec u$, ezért létezik egy olyan yx nem-fa él, melyre y benne van T_v-ben, x pedig nincs benne T_u-ban. Így a T-beli $r - x$ utat és az yx élet használva az u érintése nélkül eljuthatunk T_v-be, majd csak T_v éleit használva w-be. Tehát $G - u$-ban r-ből minden pont elérhető, ami ellentmond annak, hogy u elvágó pont. •



3.2. ábra. A gráfokban egy r gyökerű DFS fát rögzítünk, a szaggatott élek a nem-fa élek. Az első ábrán u elvágó pont: itt $u \preceq f(v)$. A másodikon u nem elvágó: ebben az esetben $r = f(v) \prec u$.

Az algoritmus az $f(u)$ értékeket a következő módon számítja ki: egy DFS kereséssel felépít egy T DFS fát egy tetszőleges r gyökérpontból, majd a levelektől visszafelé haladva minden u pontra meghatározza az $f(u)$ értéket oly módon, hogy

(i) ha u levél T-ben, akkor $f(u)$ a legkorábbi olyan pont legyen, melybe megy u-ból nem-fa él (illetve legyen $f(u) = u$, ha ilyen él nincs),
(ii) ha u nem levél, akkor $f(u)$ a legkorábbi pont legyen az

$$S = \{u\} \cup \{w : uw \text{ nem-fa él}\} \cup \{f(v) : v \text{ az } u \text{ gyereke}\}$$

halmazból. Figyeljük meg, hogy így valóban a helyes $f(u)$ értékek adódnak. A DFS bejárás $O(|V| + |E|)$ lépést igényel, továbbá, mivel az f értékek számítása során a második fázisban minden élen legfeljebb kétszer járunk (miért?), az egész eljárásra is érvényes marad az $O(|V|+|E|)$ lépésszámbecslés.

A kapott $f(u)$ értékek alapján (a 3.3.2 Lemma segítségével) az elvágó pontokat azonosítani tudjuk.

Megjegyezzük, hogy ismert olyan módszer, amely az $f(u)$ értékeket már a DFS keresés során kiszámítja, így a második fázisra ekkor nincs is szükség.

3.4. A 2-SAT probléma

Az erősen összefüggő (röviden eöf) komponenseket lineáris időben megtaláló algoritmust szubrutinként használva hatékony eljárást kapunk speciális **konjunktív normálformák** kielégíthetőségének eldöntésére.

Tekintsünk egy olyan Φ logikai kifejezést, melyben az $x_1, ..., x_n$ változókból (és ezek $\bar{x}_1, ..., \bar{x}_n$ negáltjaiból) képzett zárójeles kifejezéseket kapcsolunk össze *és* műveletekkel, és amelyben minden zárójelben két (esetleg negált) változó szerepel *vagy* művelettel összekapcsolva. (Például $(x_i \vee x_j) \wedge (x_l \vee \bar{x}_j) \wedge (\bar{x}_k \vee x_i)$.) Mikor létezik olyan értékadás (azaz az x_i változókhoz 'igaz' vagy 'hamis' értékek hozzárendelése), melyre a Φ kifejezés igaz értéket ad? Ez a 2-SAT feladat, melyben a kettes a zárójelekben szereplő változók maximális számára utal. A 3-SAT feladat a klasszikus NP-teljes feladatok egyike. A 2-SAT feladatra hatékony algoritmust szolgáltat az alábbi módszer.

Készítsük el a $D(\Phi) = (V, E)$ irányított gráfot úgy, hogy legyen $V = \{x_1, ..., x_n, \bar{x}_1, ..., \bar{x}_n\}$, és pontosan akkor menjen él az a-ból b-be, ha a $(\bar{a} \vee b)$ zárójel szerepel a Φ kifejezésben. (Ekkor persze $\bar{b}$-ből is megy él $\bar{a}$-ba.) A kielégíthetőség és a segédgráf kapcsolatát a következő állítás adja meg. Egy D digráf tartalmazásra maximális erősen összefüggő részgráfját a D **erősen összefüggő komponensének nevezzük**. Az erősen összefüggő komponensek ponthalmazai particionálják a V ponthalmazt (lásd még a fejezet végén a *Feladatokat*).

3.4.1. Tétel. *Φ pontosan akkor kielégíthető, ha $D(\Phi)$ egyetlen erősen összefüggő komponense sem tartalmazza egyszerre valamely x_i változót és annak $\bar{x}_i$ negáltját.*

Bizonyítás: Tegyük fel, hogy Φ kielégíthető és tekintsünk egy jó értékadást. Ekkor ha valamely a pont 'igaz' értéket vesz fel, akkor a segédgráf konstrukciója miatt minden olyan pont 'igaz' értéket kell felvegyen, amely a-ból $D(\Phi)$-ben irányított úton elérhető. Így minden eöf komponensre teljesül, hogy benne vagy az összes pont értéke 'igaz', vagy az összes 'hamis'. Ez esetben persze $D(\Phi)$ egyetlen erősen összefüggő komponense sem tartalmazhatja egyszerre valamely x_i változót és annak negáltját.

Fordítva, tegyük fel, hogy minden x_i-re x_i és $\bar{x}_i$ különböző eöf komponensekben vannak. Ekkor az $x_i \to \bar{x}_i$ és az $\bar{x}_i \to x_i$ irányított utak közül legalább az egyik nem létezik. Ha egyik sincs, akkor kössük őket össze valamilyen irányban egy új éllel. Ez megőrzi a feltételt, és így ilyen lépésekkel végül $D(\Phi)$-nek olyan 'kibővítéséhez' jutunk, melyben minden $x_i, \bar{x}_i$ pár közt pontosan az egyik irányban vezet út.

Nézzük azt az értékadást, melyben x_i pontosan akkor igaz, ha $\bar{x}_i$-ből vezet út x_i-be. Ez jó értékadás lesz: ha pl. szerepel az $(x_i \vee x_j)$ zárójel, de x_i és x_j is 'hamis', akkor $D(\Phi)$ definíciója és az értékadás miatt lenne irányított út x_i és $\bar{x}_i$ (és ugyanígy x_j és $\bar{x}_j$) között mindkét irányban, ami ellentmond a feltevésünknek. A többi eset is hasonlóan igazolható. •

3.4.2. Feladat. *Adjunk algoritmust az alábbi feladatra: adottak a $G_i = (V, E_i)$, $i = 1, 2$ gráfok ugyanazon a V ponthalmazon. Döntsük el, hogy van-e olyan $V = C_1 \cup C_2$ kétrészes partíciója V-nek, melyre C_i teljes gráfot feszít G_i-ben, minden $i = 1, 2$-re.*

3.5. Minimális élszámú 2-élösszefüggő feszítőgráf

A DFS keresés (és az általa adott DFS fa) speciális tulajdonságai olyan esetekben is hasznosnak bizonyulnak, amikor a cél nem elsősorban a hatékonyság növelése, hanem egy bizonyos tulajdonságokkal rendelkező részgráf létezésének igazolása és annak megtalálása. Tekintsük a következő problémát: adott egy $G = (V, E)$ 2-élösszefüggő gráf, keressünk benne minimális élszámú 2-élösszefüggő feszítőgráfot. Figyeljük meg, hogy a minimum értéke pontosan akkor $|V|$, ha van G-ben Hamilton-kör. A feladat tehát NP-nehéz, így polinomidejű, optimális megoldást adó algoritmust nem várhatunk. Azonban olyan algoritmus, amely gyorsan talál egy, az optimumnál nem sokkal rosszabb megengedett megoldást, létezik. A következő algoritmus olyan 2-élösszefüggő feszítőgráfot talál, melyben az optimumnál legfeljebb $\frac{3}{2}$-szer több él van.

Algoritmus

Hajtsunk végre egy mélységi keresést (DFS) a gráfban egy tetszőleges r pontból kiindulva, és közben jelöljük ki azt az F élhalmazt, amelyet a megoldásunk tartalmazni fog. A kiinduláskor $F = \emptyset$. A keresés során minden olyan él, mely a felépítendő T mélységi fa éle lesz, bekerül F-be. Ezen kívül minden olyan pillanatban, amikor a keresés egy v pontból visszalép a T valamely uv éle mentén (azaz amikor v átvizsgálttá válik és a következő vizsgált pont az u lesz, amely esetben az uv él a T_v részfába lépő egyetlen él, és ekkor a T-nek a v pontnál gyökerező T_v részfáját már teljesen bejártuk), ellenőrizzük, hogy az uv él elvágó él-e az eddig kijelölt F élhalmaz által feszített gráfban. Ha igen, adjunk egy olyan T_v-ből kilépő élt az F-hez, mely nincs T-ben, és a T_v-n kívüli végpontját a keresés legelőször érte el (azaz a T_v-ből az $f(v)$-be menő élt). A mélységi keresés végén kapott F élhalmaz által feszített részgráf, azaz a G gráf (V, F) feszítőgráfja legyen az output.

Könnyű igazolni, hogy az algoritmus által adott F élhalmaz 2-élösszefüggő gráfot feszít a V ponthalmazon. Valamivel bonyolultabb annak igazolása, hogy az optimum (jelölje OPT) legalább kétszer akkora, mint a végső F élhalmazban levő nem-fa élek száma. Nevezzük az $X, Y \subset V$ halmazokat **élfüggetlen**nek G-ben, ha nincs olyan él, mely mindkettőből kilép. Figyeljük meg, hogy ha $X_1, ..., X_t$ páronként élfüggetlen halmazok G-ben, akkor minden megengedett megoldás legalább $2t$ élet tartalmaz.

3.5.1. Lemma. *Legyen T_u és T_v két olyan részfa T-ben, melyre az algoritmus egy T_u-ból $f(u)$-ba és egy T_v-ből $f(v)$-be menő nem-fa élet is a megoldásba választott. Ekkor T_u és T_v ponthalmazai élfüggetlenek G-ben.*

Bizonyítás: Ha u és v az r gyökerű T fa különböző ágán vannak, azaz egyik sem elődje a másiknak, akkor a lemma következik abból, hogy DFS fában nincs keresztél. Ha u elődje v-nek, akkor az algoritmus először a T_v-ből $f(v)$-be menő nem-fa élet választja. Ezután a T_u-ból az $f(u)$-ba menő élt csak akkor vesz a megoldásba, ha T_u és T_v ponthalmazai élfüggetlenek. •

A 3.5.1 Lemma és az azt megelőző megfigyelés miatt az optimum legalább annyi, mint az algoritmus által beválasztott nem-fa élek számának kétszerese, azaz $2|F - E(T)|$. Ezt felhasználva, és megfigyelve, hogy az optimum legalább $|V|$, hiszen minden megengedett megoldásban minden pont foka legalább kettő, a következő becsléshez jutunk:

$$|F| = |F - E(T)| + |E(T)| \leq OPT/2 + |V| - 1 \leq \frac{3}{2}OPT.$$

Tehát a kívánt közelítési hányados következik.

Példa A következő példa segítségével igazolni tudjuk, hogy a fenti algoritmus által adott megoldás és OPT hányadosa akármilyen közel kerülhet $\frac{3}{2}$-hez. Legyen a G gráf a $K_{n,n}$ teljes páros gráf és az egyik színosztályára illesztett K_n teljes gráf uniója. Tegyük fel, hogy a DFS bejárás először a K_n pontjait járja be (ez feltehető – miért?). Ekkor az algoritmus által adott megoldásban $3n-1$ él lesz (ellenőrizzük!). Ugyanakkor $OPT = 2n$, hiszen a $2n$ pontú G gráfban van Hamilton kör.

3.6. Feladatok

3.6.1. Feladat. *Tegyük fel, hogy $D = (V, A)$ digráf alapgráfja összefüggő és minden pontjának ugyanannyi a befoka, mint a kifoka. Legyen s egy tetszőlegesen kijelölt pont és legyen F egy s-gyökerű feszítő be-fenyő (avagy fordított fenyő). Tekintsük a következő eljárást, amely egy sétát épít az s-ből kiindulva úgy, hogy minden u pontnál tetszés szerint halad tovább egy még nem használt uv élen, csak arra ügyelve, hogy $u \neq s$ esetén az u-ból kilépő egyetlen F-beli élt csak akkor használhatja, ha más lehetőség nincs. Az eljárás akkor ér véget, amikor egy olyan ponthoz ér, amelynek minden kimenő élét már felhasználta. Igazoljuk, hogy ez a végső pont éppen az s pont lesz, valamint a felépített séta a D minden élét tartalmazza (azaz egy irányított Euler-séta lesz).*

3.6.2. Feladat. *A D digráf u és v pontjai akkor álljanak relációban, ha van $u \to v$ és $v \to u$ út is D-ben.*

(a) Igazoljuk, hogy ez ekvivalencia reláció!

(b) Mutassuk meg, hogy minden X ekvivalencia osztályra a $D[X]$ digráf, azaz az X ponthalmaz által a D-ben feszített részdigráf, erősen összefüggő!

3.6.3. Feladat. *Igazoljuk, hogy az előző feladatban definiált $D[X]$ digráfok megegyeznek a D erősen összefüggő komponenseivel.*

3.6.4. Feladat. *Mutassuk meg, hogy ha egy D digráfban a fenti feladatban definiált ekvivalencia osztályok mindegyikét egy-egy pontra összehúzzuk, akkor egy aciklikus digráfot kapunk.*

3.6.5. Feladat. *Adjunk polinomiális algoritmust annak eldöntésére, hogy egy adott G gráf páros gráf vagy sem.*

3.6.6. Feladat. *Tegyük fel, hogy a D digráfban minden v pontra a v kifoka megegyezik a befokával és D gyengén összefüggő. Az Euler-séta létezésére vonatkozó tétel használata nélkül igazoljuk, hogy (a) a D minden ponthalmazára teljesül, hogy a kifoka megegyezik a befokával, és (b) D erősen összefüggő.*

3.6.7. Feladat. *Bizonyítsuk be, hogy minden erősen összefüggő digráf Euler digráffá tehető úgy, hogy bizonyos éleit megtöbbszörözzük!*

3.6.8. Feladat. *Bizonyítsuk be, hogy minden összefüggő gráf Euler gráffá tehető úgy, hogy bizonyos éleit megtöbbszörözzük!*

3.6.9. Feladat. *Bizonyítsuk be, hogy összefüggő, egyszerű irányítatlan gráfban, amelyben minden pont foka legalább három, létezik egy olyan húrmentes C kör, amelynek pontjait elhagyva összefüggő gráfot kapunk.*

4. fejezet

Irányítások

4.1. Erősen összefüggő irányítások

A $D = (V, A)$ digráf **erősen összefüggő**, ha minden $u, v \in V$ pontpárra van D-ben irányított út u-ból v-be (és így v-ből u-ba is).

4.1.1. Lemma. *A $D = (V, A)$ digráfban pontosan akkor van út u-ból v-be, ha minden $u \in X \subseteq V - \{v\}$ ponthalmazból lép ki él.* •

Vegyes gráfban lehetnek irányított és irányítatlan élek is. A $G = (V, E \cup A)$ vegyes gráfban a ponthalmaz egy X valódi részhalmaza **egyirányú vágás**, ha minden olyan (irányított vagy irányítatlan) él, amelynek pontosan az egyik vége van X-ben, az az X-be belépő irányított él.

4.1.2. Lemma. *A $D = (V, A)$ digráf pontosan akkor erősen összefüggő, ha nincs benne egyirányú vágás.* •

A $G = (V, E)$ gráf **irányítása** egy olyan $D = (V, A)$ digráf, amely a G-ből úgy áll elő, hogy minden $uv \in E$ élét egy irányított éllel helyettesítjük (amelynek vagy u a töve és v a feje, vagy fordítva).

A célunk azon $G = (V, E)$ gráfok jellemzése, melyekre van G-nek erősen összefüggő irányítása. A G összefüggő gráfban az $e \in E$ él **elvágó**, ha $G - e$ nem összefüggő. Világos, hogy ha G-nek van erősen összefüggő irányítása, akkor G összefüggő kell legyen. Kis meggondolás mutatja, hogy G-ben nem lehet elvágó él. Robbins következő tétele azt mutatja meg, hogy ez az egyszerű szükséges feltétel elégséges is.

4.1.3. Tétel. *A $G = (V, E)$ gráfnak pontosan akkor van erősen összefüggő irányítása, ha G összefüggő és nincs elvágó éle.*

Bizonyítás: A szükségesség világos. Az elégségesség igazolásához azt fogjuk megmutatni, hogy ha G összefüggő és nincs elvágó éle, akkor van olyan D irányítása, amelyben nincs egyirányú vágás. Láttuk, hogy ez esetben D erősen összefüggő.

Rendezzük sorba az éleket tetszőlegesen, majd ebben a sorrendben haladva irányítsuk őket úgy, hogy ne keletkezzen egyirányú vágás, amíg csak lehet. Ha minden élt tudtunk így irányítani, akkor kész vagyunk. Kezdetben (azaz G-ben) nincs egyirányú vágás (miért?).

Tekintsük a soron következő $e = uv$ élt. Meg fogjuk mutatni, hogy az uv élt tudjuk úgy irányítani, hogy továbbra se legyen egyirányú vágás.

Figyeljük meg, hogy az uv él irányítása u-ból v-be pontosan akkor rossz, azaz pontosan akkor hoz létre egyirányú vágást, ha van olyan $X \subseteq V$, $v \in X$, amelyre az egyetlen irányítatlan él X és $V - X$ között az uv, és minden irányított él, amelynek pontosan egy végpontja van X-ben, az belép X-be. Hasonlóan, a v-ből u-ba irányítás pontosan akkor rossz, ha van olyan $Y \subseteq V$, $u \in Y$, amelyre az egyetlen irányítatlan él Y és $V - Y$ között az uv, és minden irányított él, amelynek pontosan egy végpontja van Y-ban, az belép Y-ba. Feltehetjük, hogy ilyen X és Y halmazok léteznek, ellenkező esetben kész vagyunk.

Ekkor az $X \cap Y$ halmazból semmilyen él sem léphet ki, hiszen egy ilyen él vagy az X-ből, vagy az Y-ból (vagy mindkettőből) kilépne. Mivel az aktuális (vegyes) gráfban nincs egyirányú vágás, ez azt jelenti, hogy $X \cap Y = \emptyset$. Ugyanígy, az $X \cup Y$ halmazból sem léphet ki semmilyen él. Ezért $X \cup Y = V$. Mindezek alapján az X, Y egy olyan kettéosztása a V-nek, amelyre a két rész között pontosan egy él vezet: az uv. Ez ellentmondás, hiszen ekkor uv elvágó él lenne G-ben. •

Figyeljük meg, hogy a bizonyítás algoritmikus, és elvezet egy polinomiális algoritmushoz, amely megtalálja a kívánt irányítást (ha létezik). Ehhez két megfigyelésre van szükség: a soron következő uv él irányítása u-ból v-be pontosan akkor jó, ha ezen irányítás mellett van a vegyes gráfban v-ből u-ba olyan (vegyesen) irányított út, amely az irányítatlan éleket tetszőleges irányban használhatja. Ez pedig a kereső algoritmussal eldönthető, ha azt abban a digráfban futtatjuk, amelyben minden irányítatlan élt két (ellentétesen) irányított éllel helyettesítünk.

Vegyes gráfok irányítása során csak az irányítatlan éleket irányítjuk. A fenti bizonyítás gondolatmenete vegyes gráfokra is kiterjeszthető és a következő általánosabb eredményre vezet. A vegyes gráf **alapgráfja** az a gráf, amelyben minden uv irányított élt az uv éllel helyettesítünk.

4.1.4. Tétel. *A $G = (V, E \cup A)$ vegyes gráfnak pontosan akkor van erősen összefüggő irányítása, ha nincs benne egyirányú vágás és az alapgráfjában nincs elvágó él.*

Bizonyítás: A feltételek szükségessége világos. Tegyük most fel, hogy a G-ben nincs egyirányú vágás és a G'-vel jelölt alapgráfjában nincs elvágó él. Rendezzük sorba a G' éleit úgy, hogy a G-ben irányított éleket vesszük előre és irányítsuk ezeket a G-beli irányításuknak megfelelően. Mivel a kapott vegyes gráfban (ami éppen a G) nincs egyirányú vágás és G'-ben nincs elvágó él, az előző bizonyítást követve azt kapjuk, hogy a többi él irányítható úgy, hogy a kapott digráf erősen összefüggő legyen. •

4.2. Feladatok

4.2.1. Feladat. *A G összefüggő gráfban minden pont foka páros és legalább 4. Igazoljuk, hogy*

a) G-nek van erősen összefüggő irányítása,

b) G-ben van olyan F lefogó élhalmaz, melyre $G - F$ összefüggő! (Egy F élhalmaz lefogó, ha minden pontra illeszkedik F-beli él.)

4.2.2. Feladat. *Jellemezzük azokat a $G = (V, E)$, $s \in V$ párokat, melyekre van G-nek olyan irányítása, amelyben s-ből minden pont elérhető.*

4.2.3. Feladat. *Jellemezzük azokat az összefüggő G gráfokat, amelyeknek minden irányításában legfeljebb egy olyan pont van, ahonnan minden csúcs elérhető irányított úton!*

4.2.4. Feladat. *Jellemezzük azokat a $G = (V, E)$ gráfokat, amelyeknek van olyan irányítása, amelyben minden pont befoka pontosan egy!*

4.2.5. Feladat. *Mutassuk meg, hogy minden gráfnak van olyan irányítása, amelyben minden pontra a befok és kifok legfeljebb eggyel tér el egymástól!*

5. fejezet

Legrövidebb utak

5.1. Dijsktra algoritmusa

Legyen $D = (V, A)$ egy digráf és $\ell : A \to R$ egy függvény. Egy $e \in A$ élre az $\ell(e)$ érték az e **hossza**. Egy P út (vagy általában egy P séta, vagy még általánosabban, egy P részgráf) esetén a P **hossza** az

$$\ell(P) = \sum_{e \in A(P)} \ell(e)$$

érték, azaz a P-ben levő élek hosszainak összege. A D-ben az $s, t \in V$ pontok **távolsága** (az ℓ-re nézve) az s-ből t-be vezető utak hosszainak minimuma. Jelölése $dist(s, t)$. Ha nincs ilyen út, akkor legyen $dist(s, t) = \infty$.

Dijkstra algoritmusa adott D, nem-negatív (!) ℓ és $s \in V$ esetén minden $t \in V$ pontra kiszámolja a $dist(s, t)$ távolságot, illetve kis plusz munkával meghatároz egy legrövidebb utat s-ből t-be.

Az algoritmus minden $v \in V$ pontra fenntart egy $f(v)$ számot, valamint fenntart egy $U \subseteq V$ ponthalmazt. Kezdéskor $f(s) = 0$, valamint minden más v pontra $f(v) = \infty$ és $U = V$. Az algoritmus az alábbi általános lépést ismétli mindaddig, amíg végül $U = \emptyset$ teljesül. Ekkor leáll.

(D) Legyen $u \in U$ egy olyan pont, amelyre $f(u)$ minimális. Minden olyan $uv \in A$ élre, amelyre $v \in U$ és $f(v) > f(u) + \ell(uv)$, legyen $f(v) := f(u) + \ell(uv)$. Legyen $U := U - \{u\}$.

5.1.1. Tétel. *Az algoritmus leállásakor $f(v) = dist(s, v)$ minden $v \in V$ pontra.*

Bizonyítás: Legyen $dist(v) = dist(s, v)$ minden $v \in V$-re. Az iterációk számára vonatkozó indukcióval könnyen látszik, hogy minden v pontra végig teljesülni fog, hogy van olyan séta s-ből v-be, amelynek hossza legfeljebb $f(v)$. Mivel ℓ nem-negatív, ezért ilyen út is van. Azaz $dist(v) \leq f(v)$ mindig fennáll.

Meg fogjuk mutatni, az iterációk számára vonatkozó indukcióval, hogy minden lépésben a kiválasztott u pontra érvényes az $f(u) = dist(u)$ egyenlőség. Mivel minden s-ből elérhető pontot egyszer kiválasztunk és $f(u)$ nem csökkenhet $dist(u)$ alá, ebből a tétel következik. Az első iterációban $u = s$, amelyre $f(s) = 0 = dist(s)$ teljesül. Tegyük fel, indirekt, hogy $f(u) > dist(u)$. Legyen P egy legrövidebb út s-ből u-ba. Tekintsük a $P = (s = v_0, v_1, \ldots, v_k = u)$ út első élét, legyen ez a $v_{i-1}v_i$ él, amely belép U-ba. Ekkor $v_{i-1} \in V - U$, azaz a v_{i-1} pont egy korábbi iterációban volt kiválasztott pont. Ez a tény és az algoritmus szabálya mutatja az első egyenlőtlenség

érvényességét:

$$f(v_i) \leq f(v_{i-1}) + \ell(v_{i-1}v_i) = dist(v_{i-1}) + \ell(v_{i-1}v_i) = dist(v_i).$$

Az első egyenlőség az indukciós feltevésből következik (amely szerint a korábban kiválasztott v_{i-1} pontra az f-érték megegyezik a távolságával s-től), míg a második egyenlőség azért teljesül, mert a P út s-ből v_{i-1}-be vezető részútja legrövidebb út. (Itt kihasználtuk, hogy ℓ nem-negatív!). Tehát, ismét használva a nem-negativitást és az indirekt feltevést, azt kapjuk, hogy

$$f(v_i) \leq dist(v_i) \leq dist(u) < f(u).$$

Ez azonban ellentmond az u minimális választásának. •

Példa Tekintsük azt a digráfot, melynek pontjai s, u, v, élei (az élek hosszával együtt) $\ell(su) = 1, \ell(sv) = \ell(vu) = 0, \ell(uv) = -2$. Itt a legrövidebb s-u út kezdőszelete v-ig nem legrövidebb s-v út.

A Dijkstra-algoritmus futási ideje $O(|V|^2)$.

A D-ben – az adott s gyökérhez és ℓ hosszfüggvényhez – egy s gyökerű F fenyőt akkor nevezünk **legrövidebb utak fenyőjének**, ha minden irányított út F-ben egy D-beli legrövidebb út a végpontjai között. Első látásra nem világos, hogy ilyen fenyő mindig létezik. A Dijkstra-algoritmus elemzése alapján ez is következik: ilyen F-et kapunk, ha minden v ponthoz feljegyezzük azt az élt, amely mentén utoljára változtattuk meg az $f(v)$ értéket, és az algoritmus leállásakor vesszük ezen élek unióját.

5.2. A Bellman-Ford-algoritmus

Ebben a szakaszban azt vizsgáljuk meg, hogy mi a helyzet akkor, ha a $D = (V, A)$ digráf $\ell : A \to \mathbb{R}$ élhossz függvénye negatív értékeket is felvehet. Látni fogjuk, hogy hasznos lesz legrövidebb sétákat keresni, és ezek segítségével legrövidebb utakat is találhatunk majd.

Ha semmilyen megszorítást sem teszünk az ℓ függvényre, akkor előfordulhat, hogy két pont között, melyek között van út, nem létezik legrövidebb séta. Továbbá a legrövidebb út keresésének problémája általánosítja az irányított Hamilton út feladatot (gondoljuk meg!), ezért nem várható, hogy hatékony megoldó algoritmust találjunk anélkül, hogy valamilyen megszorítást tennénk ℓ-re.

Azt mondjuk, hogy ℓ **konzervatív**, ha a D minden C irányított körére $\ell(C) \geq 0$, azaz a C élhosszainak összege nem-negatív. Konzervatív ℓ esetén a feladatunk hatékonyan megoldható. Első lépésként a következő egyszerű tulajdonságot igazoljuk.

5.2.1. Lemma. *Ha ℓ konzervatív, akkor minden olyan $s, t \in V$ párra, melyek közt vezet séta, vezet legrövidebb séta is. Továbbá, a legrövidebb séták között van s-t út.*

Bizonyítás: Mivel van s-t séta, van s-t út is. Emiatt van legrövidebb s-t út, hiszen ezen utak száma D-ben véges. Legyen P egy legrövidebb s-t út, hossza $\ell(P)$.

Megmutatjuk, hogy P az összes s-t séta között egy legrövidebb s-t séta. Indirekt tegyük fel, hogy van egy rövidebb Q séta s-ből t-be. Erre tehát $\ell(Q) < \ell(P)$ teljesül. Tegyük fel, hogy Q élszáma a lehető legkisebb a P-nél rövidebb séták közül. A Q tartalmaz egy C kört, hiszen nem lehet s-t út. Legyen $Q' = Q - C$. Mivel ℓ konzervatív, $\ell(C) \geq 0$, ezért $\ell(Q') = \ell(Q) - \ell(C) \leq \ell(Q) < \ell(P)$.

Ez esetben Q' egy kevesebb élű, P-nél rövidebb séta s-ből t-be, ami ellentmond a Q választásának. •

A lemmából következik, hogy D két pontja között a legfeljebb $|V|-1$ élű séták minimális hossza megegyezik a legrövidebb út hosszával. Erre épül az alábbi Bellman-Ford-algoritmus. A Dijkstra-algoritmushoz hasonlóan itt is egy adott $s \in V$ pontból egyidejűleg az összes többi pontba kiszámoljuk a legrövidebb séta (út) hosszát.

Input: a $D = (V, A)$, $s \in V$, valamint az $\ell : A \to \mathbb{R}$ konzervatív élhossz függvény.

Legyen $n = |V|$. Az algoritmus az $f_0, f_1, \ldots, f_n : V \to \mathbb{R} \cup \{\infty\}$ függvényeket számolja ki az alábbi módszerrel. Legyen $f_0(s) = 0$ és $f_0(v) = \infty$ minden $v \neq s$ pontra. Ezután:

Ha az $f_0, \ldots, f_k$ függvényeket már kiszámoltuk valamely $k < n$-re, akkor minden $v \in V$ pontra legyen

$$f_{k+1}(v) = \min\{f_k(v), \min_{uv \in A}(f_k(u) + \ell(uv))\}. \tag{5.1}$$

5.2.2. Tétel. *Ha ℓ konzervatív, akkor $f_{n-1}(v)$ megegyezik a legrövidebb s-v út hosszával minden $v \in V$-re. Ha nincs út s-ből v-be, akkor $f_{n-1}(v) = \infty$.*

Bizonyítás: A bizonyítás következik az alábbi megfigyelésből, melyet k-ra vonatkozó indukcióval igazolhatunk (5.1) alapján: minden $k = 0, 1, \ldots, n-1$-re és minden $v \in V$-re teljesül az

$$f_k(v) = \min\{\ell(P) : P \text{ egy legfeljebb } k \text{ élt tartalmazó } s\text{-}v \text{ séta}\}$$

egyenlőség. •

Amennyiben egy legrövidebb útra is kíváncsiak vagyunk, az algoritmus futása során tartsunk fenn egy $g : V \to V$ függvényt, melyet menet közben minden v-re $g(v) = u$-nak definiálunk, valahányszor (5.1)-ben az éppen meghatározandó $f_{k+1}(v)$-re $f_{k+1}(v) = f_k(u) + \ell(uv) < f_k(v)$ áll. Ekkor az algoritmus leállásakor a $v, g(v), g(g(v)), \ldots, s$ pontsorozat egy legrövidebb s-v út pontjait adja meg (fordított sorrendben).

Az algoritmus futási idejére az alábbi becslést kapjuk, ha megfigyeljük, hogy $|V|$-szer iteráljuk a (5.1) rekurziót, és minden iterációban minden $uv \in A$ élt egyszer vizsgálunk.

5.2.3. Tétel. *Adott $D = (V, A)$ digráf, $s, t \in V$ és az ℓ konzervatív élhossz függvény esetén egy legrövidebb s-t út $O(|V||A|)$ időben megkapható.*

5.2.1. Egy alkalmazás

A hátizsák feladatban adott térfogatú és értékű tárgyakat (illetve ezek egy részét) pakolunk egy adott kapacitású hátizsákba úgy, hogy a bepakolt tárgyak kapacitásainak összege ne haladja meg a hátizsák kapacitását, és a bepakolt tárgyak értékeinek összege legyen a lehető legnagyobb.

Ha n a tárgyak száma, a_i és c_i jelöli az i-edik tárgy térfogatát és értékét ($1 \leq i \leq n$), valamint b a hátizsák kapacitása, akkor ez az optimalizálási feladat felírható n változóval így:

határozzuk meg az $x_1, x_2, \ldots, x_n$ számokat úgy, hogy

$$x_1, x_2, \ldots, x_n \in \{0, 1\},$$

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b,$$

valamint erre nézve

$$c_1x_1 + c_2x_2 + \cdots + c_nx_n \text{ maximális.}$$

Ezt a feladatot átfogalmazhatjuk egy legrövidebb út feladatra egy olyan digráfban, melyben lehetnek negatív élhosszak, de nincsenek irányított körök. (Lásd a *Feladatok* között.) A Bellman-Ford-algoritmus tehát alkalmazható. A futási idő azonban, az általános esetben, nem lesz polinomiális! Kis kapacitásokra ugyanakkor hatékony lehet.

5.3. A PERT módszer

Nagyobb (sok kisebb feladatból álló) projektek tervezésekor gyakran merül fel, hogy mikor és milyen sorrendben célszerű a feladatokat elvégezni, ha a projekt minél gyorsabb befejezése a cél, de közben figyelni kell arra is, hogy bizonyos feladatpárok között megelőzési feltételek is adottak: az egyik csak azután kezdhető meg, ha a másikkal már végeztek. Adottak tehát a feladatok elvégzési idejei (az i feladat esetén jelölje ezt t_i) és a megelőzési feltételek. Ez utóbbiak természetes módon reprezentálhatók egy $D = (V, A)$ digráffal, amelyben a pontok a feladatoknak felelnek meg, az élek pedig a feltételeknek: az ij él mutatja, hogy a j feladat csak az i befejezése után kezdhető el. Cél a feltételeket teljesítő olyan időbeosztás, amelyben az utolsó munka befejezése a leghamarabb történik.

Ennek megoldása érdekében adjunk két további pontot D-hez, s-et és t-t, továbbá vezessünk élt s-ből minden más pontba, és minden más pontból t-be. Ennek a két extra feladatnak legyen 0 az elvégzési ideje. Az élek hosszait definiáljuk úgy, hogy az ij él hossza legyen t_i. A kapott aciklikus digráfot az élhosszakkal jelölje D'.

5.3.1. Lemma. *A projekt elvégzésének legrövidebb ideje megegyezik a D'-beli leghosszabb s-t út hosszával.*

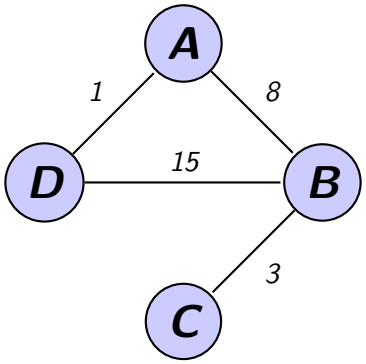
Bizonyítás: Az világos, hogy a leghosszabb s-t út hossza alsó korlát a projekt elvégzési idejére. Jelölje l_i a leghosszabb s-i út hosszát D'-ben. Figyeljük meg, hogy ha minden i feladatot az l_i időpontban kezdünk, az megfelelő beosztást ad, hiszen ha j az i után kell legyen, akkor $l_j \geq l_i + t_i$ a leghosszabb út definíciója miatt. (Ne felejtsük el, hogy a D' aciklikus.) Tehát t elkezdhető l_t időben, ami épp azt jelenti, hogy minden feladat addigra elkészül. •

A D'-beli leghosszabb s-t út megkapható a Bellman-Ford-algoritmussal, hiszen ez ekvivalens egy legrövidebb út kereséssel, miután minden él hosszát megszoroztuk -1-gyel. Mivel D' aciklikus, a Bellman-Ford-algoritmus tetszőleges hosszak esetén működik.

5.4. Feladatok

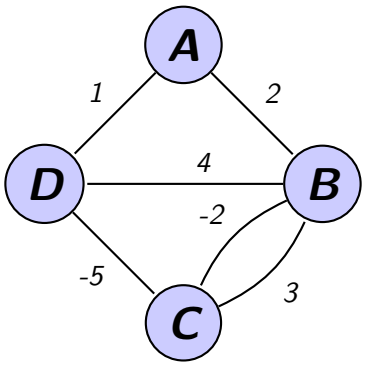
5.4.1. Feladat. *Lássuk be, hogy (a) nem-negatív élköltségű gráfban (b) konzervatív élköltségű digráfban a távolságok kielégítik a háromszög-egyenlőtlenséget! (c) Mi a helyzet általános költségfüggvényre?*

5.4.2. Feladat. *Dijkstra algoritmusa nem-negatív élköltségű digráfban egy gyökérpontból minden pontba kiszámolja a legolcsóbb utat. Ismételjük át a Dijkstra-algoritmust a lenti példán, a D gyökérpontból indulva! Milyen értékeket tárolunk az algoritmus során?*



5.4.3. Feladat. *Módosítsuk a példában szereplő élköltségeket úgy, hogy szerepeljenek negatív élköltségek is, és az algoritmus rossz eredményt adjon!*

5.4.4. Feladat. *Határozzuk meg a Bellman-Ford-algoritmussal az A pontból induló legrövidebb utakat a következő alapgráffal és élhosszakkal rendelkező digráfban, melyben az élek irányításai:* $A \to D, A \to B, B \to D, C \to D, B \to C$ $(3), C \to B$ (-2).



5.4.5. Feladat. *Mi történik ha a Bellman-Ford-algoritmust nem konzervatív gráfra próbáljuk futtatni?*

a) Rajzoljunk egy gráfot nem konzervatív költségfüggvénnyel (lehetőleg minél kisebb gráfot) és próbáljuk ki!

b) Ez alapján találjuk ki, mi történik az általános esetben.

5.4.6. Feladat. *Adott egy digráf és az élein egy tetszőleges költségfüggvény. Adjunk polinomiális algoritmust minimális összhosszú* pontosan k *élből álló séta kiszámítására!*

5.4.7. Feladat. *Legyen* $D = (V, A)$ *egy digráf,* $\ell \colon A \to \mathbb{R}_+$ *egy nemnegatív élhosszfüggvény. Igazoljuk, hogy a legrövidebb* $s \to t$ *út hossza megegyezik a* $\phi(t) - \phi(s)$ *értékek maximumával, ahol a maximum azokra a* $\phi \colon V \to \mathbb{R}$ *függvényekre tekintendő, amelyekre* $\phi(w) - \phi(v) \leq \ell(vw)$ *minden* $vw \in A$ *élre.*

5.4.8. Feladat. *Mutassuk meg, hogy aciklikus digráfban lehet polinomiális algoritmussal leghosszabb utat keresni.*
a) Hogyan használhatjuk erre a problémára a Bellman-Ford algoritmust?
b) Adjunk egy másik algoritmust is, ami az aciklikus gráf egy topologikus sorrendjét használja: azaz $s = v_1, v_2, \ldots, v_n$ *topologikus sorrend esetén, ha ismert a leghosszabb* $s - v_i$*-út hossza minden* $i \leq K$ *esetén (jelölje ezt* $h(s, v_i)$*), akkor hogyan lehet gyorsan kiszámítani* $h(s, v_{K+1})$*-et?*

5.4.9. Feladat. *Egy számsorozatban keressünk leghosszabb monoton növő részsorozatot. Vezessük vissza ezt a problémát aciklikus digráfban leghosszabb út keresésre! Azaz konstruáljunk a számsorozathoz egy aciklikus digráfot, hogy a leghosszabb út a digráfban a leghosszabb monoton növő részsorozatnak feleljen meg.*

5.4.10. Feladat. *Fogalmazzuk át a hátizsák feladatot egy legrövidebb út feladatra. (Ötlet: készítsünk egy* D *digráfot, amelyben minden* (i, q) *párnak megfelel egy pont, ahol* $0 \leq i \leq n + 1$ *és* $0 \leq q \leq b$ *egészek. Vezessen egy él* $(i - 1, x)$*-ből* (i, y)*-ba, ha*
(i) $y = x$ *és* $i \leq n$*,*
(ii) $y = x + a_i$ *és* $i \leq n$*, vagy*
(iii) $i = n + 1$ *és* $y = b$*.*
A megfelelő élhosszak a fenti három esetben legyenek 0*,* $-c_i$ *és* 0*.)*

6. fejezet

Minimális feszítőfák

Ebben a fejezetben áttekintjük a korábbi tanulmányokból talán már ismert minimális költségű (vagy, ha az élekhez rendelt számokat költség helyett hossznak hívjuk, akkor minimális összhosszúságú) feszítőfa feladatot és az ezt megoldó ún. mohó algoritmust, valamint mutatunk egy alkalmazást. Ez az algoritmus a jegyzet további részében többször is felbukkan majd bonyolultabb algoritmusok szubrutinjaként (lásd pl. Steiner-fa feladat, folyamekvivalens fák).

6.1. Mohó erdők

Legyen $G = (V, E)$ egy összefüggő gráf, $\ell : E \to \mathbb{R}$ egy hosszfüggvény az éleken. Célunk egy olyan T feszítőfa keresése, melyre

$$\ell(T) = \sum_{e \in E(T)} \ell(e)$$

minimális. (Egy ilyen feszítőfát röviden **minimális feszítőfának** nevezünk.)

Két olyan algoritmust mutatunk, amely egy mohó stratégiával megtalál egy optimális megoldást. Mindkettő helyességének igazolása az alábbi lemmán alapul.

Az F erdő **mohó**, ha van olyan T minimális feszítőfa, amely tartalmazza az éleit. (Az egyszerűség kedvéért az F-et és T-t néha azonosítjuk az élhalmazaikkal.)

6.1.1. Lemma. *Legyen (V, F) mohó erdő, legyen U az egyik komponense és legyen $e \in \delta(U)$. Ha e egy legrövidebb él $\delta(U)$-ban, akkor $(V, F \cup \{e\})$ mohó erdő.*

Bizonyítás: Legyen $e = uv$ és T olyan minimális feszítőfa, amely tartalmazza F-et. Ha $e \in T$, akkor kész vagyunk. Egyébként legyen P az egyértelmű u-v út T-ben. A P úton van legalább egy olyan f él, amelyre $f \in \delta(U)$. A $P \cup \{e\}$ egy kör, ami tartalmazza f-et, ezért $T' = T - f + e$ egy feszítőfa. Feltevésünk szerint $\ell(e) \leq \ell(f)$, így $\ell(T') \leq \ell(T)$. Ekkor T' is minimális feszítőfa. Mivel $F \cup \{e\}$ része T'-nek, $F \cup \{e\}$ mohó erdő. •

Mohó algoritmus (1. változat) Az algoritmus egy F fát épít a gráfban egy tetszőleges egypontú fával indulva. Ha még nincs F-ben minden pont, akkor F-et a $\delta(V(F))$ egyik legrövidebb élével bővíti.

Mohó algoritmus (2. változat) Az algoritmus egy F erdőt épít a gráfban a V ponthalmazon, az él nélküli erdővel indulva. Ha F még nem alkot összefüggő gráfot a V-n, akkor egy legrövidebb olyan élet ad F-hez, amelynek hozzáadása nem hoz létre kört.

Mindkét algoritmus egy minimális feszítőfával áll le. Ez következik az 6.1.1 Lemmából. Hasznos megfigyelés, hogy a fenti módszer maximális feszítőfa keresésére is használható.

6.2. Megbízható hálózatok

Ebben a szakaszban a maximális feszítőfa algoritmus egy alkalmazását mutatjuk be egy olyan feladatban, amelyben a cél egy megbízható hálózat keresése.

Tegyük fel, hogy a $G = (V, E)$ összefüggőgráfban az élek **erőssége** (avagy ellenálló képessége) eltérő lehet. Ezen erősségeket egy $f : E \to \mathbb{R}_+$ függvény adja meg. Egy P út **megbízhatóságát** az úton levő élek erősségeinek minimuma definiálja. Ez megadja minden $u, v \in V$ pontpár $m_G(u, v)$-vel jelölt **megbízhatóságát**, melyet úgy definiálunk, hogy ez legyen az u és v között vezető utak maximális megbízhatósága.

Legyen T egy maximális erősségű feszítőfa G-ben, azaz olyan, amelyre $\sum_{e \in E(T)} f(e)$ a lehető legnagyobb. Egy ilyen T feszítőfa a mohó algoritmussal gyorsan kiszámolható. Az érdekes tulajdonság az, hogy a T-ben minden pontpár megbízhatósága pontosan ugyanannyi, mint G-ben, azaz

$$r_T(u, v) = r_G(u, v)$$

minden $u, v \in V$-re teljesül. Ennek igazolását feladatként tűzzük ki.

6.3. Feladatok

6.3.1. Feladat. *Igazoljuk a fenti $r_T(u, v) = r_G(u, v)$ egyenlőséget a gráf pontpárjaira!*

6.3.2. Feladat. *Mutassuk meg, hogy ha minden költség különböző, akkor a minimális költségű feszítőfa egyértelmű.*

6.3.3. Feladat. *Tegyük fel, hogy két költségfüggvény adott az éleken: c_1, c_2. Adjunk algoritmust olyan feszítőfa megkeresésére, amely a c_1-re nézve minimális költségű és ezen belül c_2-re nézve minimális költségű. Hogyan általánosítható az eljárás több költségfüggvényre?*

6.3.4. Feladat. *Mutassuk meg, hogy egy $G = (V, E)$ összefüggő gráf egy F feszítőfája akkor és csak akkor minimális költségű, ha a fa bármely e élének a költsége nem nagyobb, mint az $F - e$ két komponense között vezető bármely él költsége. Továbbá F akkor és csak akkor minimális költségű, ha bármely $uv \in E - E(F)$ él költsége legalább akkora, mint a fában az u és v között vezető egyértelmű út bármely élének a költsége.*

7. fejezet

Közelítő algoritmusok

Gyakran előfordul, hogy egy optimalizálási feladatról kiderül, hogy olyan algoritmust, amely tetszőleges input esetén polinomiális időben optimális megoldást talál, nem várhatunk (ennek tipikus esete, amikor a feladatról kimutatható, hogy egy NP-nehéz feladat visszavezethető rá, így ő is NP-nehéz).

A feladat megoldása során valamelyik követelményből engednünk kell. Megszoríthatjuk a lehetséges inputok halmazát (pl. csak páros gráfokat, vagy nem-negatív élsúlyokat tekintünk egy gráf optimalizálási feladatban), eltekinthetünk a polinomiális futási időtől (pl. megelégedhetünk ún. pszeudo-polinomiális algoritmussal), vagy engedhetünk az optimalitásból.

Ebben a fejezetben ezt az utóbbi engedményt tesszük: bevezetjük a közelítő algoritmus fogalmát. Legyen a P egy minimalizálási feladat, ahol egy olyan x megengedett megoldást keresünk, amelyre $f(x)$ minimális, ahol f a célfüggvény. Azt mondjuk, hogy az $\mathcal{A}$ algoritmus **közelítő algoritmus**, vagy **α-közelítő algoritmus** (valamely $\alpha \geq 1$ számra) a P feladatra, ha

(i) a P minden inputjára megengedett megoldást talál, vagy kimutatja, hogy ilyen nincs,
(ii) futási ideje polinomiális,
(iii) az algoritmus által adott x megoldásra teljesül, hogy

$$\frac{f(x)}{OPT} \leq \alpha,$$

ahol $OPT = f(x_{opt})$ a feladat optimuma, azaz egy x_{opt} optimális megoldásra a célfüggvény értéke.

Maximalizálási feladatoknál a definíció hasonló, ott $\frac{f(x)}{OPT} \geq \alpha$, valamely $0 < \alpha \leq 1$-re.

7.1. Minimális lefogó ponthalmaz

Bevezető példaként tekintsük a minimális méretű lefogó ponthalmaz feladatot. Ennek inputja egy $G = (V, E)$ gráf, megengedett megoldások a $T \subseteq V$ lefogó ponthalmazok, a célfüggvény az $f(T) = |T|$. Ez a feladat NP-nehéz.

Az $\mathcal{A}$ algoritmus a következő: keressünk G-ben egy nem bővíthető $M \subseteq E$ párosítást. Legyen az output az M által fedett pontok T halmaza.

7.1.1. Tétel. *Az $\mathcal{A}$ algoritmus 2-közelítő a minimális méretű lefogó ponthalmaz feladatra.*

Bizonyítás: Egy nem bővíthető M párosítás egyszerűen előállítható: diszjunkt éleket választunk egymás után. Ha van olyan él, amelynek egyik kiválasztottal sincs közös pontja, akkor azt hozzávesszük. Ha nincs, akkor az élhalmazunk egy nem bővíthető párosítás. A lépésszám könnyen láthatóan polinomiális.

Legyen T az M végpontjainak halmaza. Ekkor T egy lefogó ponthalmaz. Mivel minden megengedett megoldás (lefogó ponthalmaz) az M minden élét lefedi, és ezek az élek diszjunktak, $|M| \leq OPT$. Emiatt

$$|T| = 2|M| \leq 2OPT,$$

azaz az algoritmus 2-közelítő. •

7.2. A Steiner-fa feladat

A Steiner-fa feladat a következő: adott egy $G = (V, E)$ gráf, egy $c : E \to \mathbb{R}_+$ nem-negatív költségfüggvény, és a ponthalmaz egy $V = T \cup S$ partíciója két részre. A T a terminálok halmaza, S-ben vannak a Steiner-pontok. A cél egy olyan minimális költségű fa keresése G-ben, amely minden terminált tartalmaz (valamint a Steine-pontok egy tetszőleges részhalmazát).

Vegyük észre, hogy a $T = V$ és $|T| = 2$ eseteket meg tudjuk oldani polinomidőben. Általában a feladat NP-nehéz.

Pontosan akkor van megengedett megoldás, ha az összes terminál a G ugyanazon komponensében van. A további komponensekben levő Steiner-pontok érdektelenek. Feltehetjük tehát, hogy G összefüggő.

Azt mondjuk, hogy a c **metrikus**, ha G teljes gráf és minden u, v, w ponthármasra

$$c(u, v) \leq c(u, w) + c(w, v).$$

A Steiner-fa feladat azon speciális esete, amelyben c metrikus, a metrikus Steiner-fa feladat. Megmutatjuk, hogy ez a megszorítás nem lényeges.

7.2.1. Tétel. *A Steiner-fa feladat visszavezethető a metrikus Steiner-fa feladatra úgy, hogy a visszavezetés megőrzi a közelítési hányadost.*

Bizonyítás: A Steiner-fa feladat egy I példányát polinom időben átalakítjuk a metrikus Steiner-fa feladat egy I' példányává. Legyen az I a $G = (V, E)$, c, $V = T \cup S$. Legyen G' egy teljes gráf a V ponthalmazon. A G'-ben egy uv él c'-beli költsége legyen

$$c'(uv) = dist^c_G(u, v),$$

azaz egy legrövidebb út hossza G-ben, c-re nézve. A V felosztása ugyanaz a G és G' gráfokban.

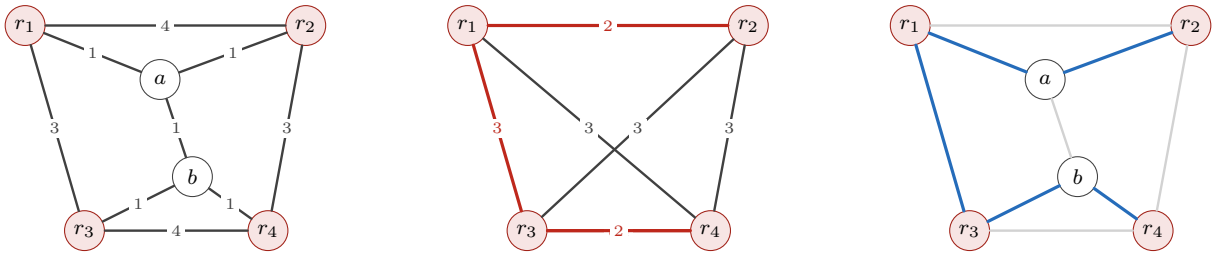
Minden $uv \in E$ élre $c'(uv) \leq c(uv)$, ezért az I' egy optimális megoldásának költsége nem nagyobb, mint az I egy optimális megoldásának költsége. Tömören $OPT(I') \leq OPT(I)$.

Tekintsünk egy F' Steiner-fát az I' inputhoz. Az F' minden élének költsége megfelel egy G-beli út hosszának. Az F' minden élére vegyük ezeknek az utaknak az éleit, és vizsgáljuk az ezen élek uniója által alkotott részgráfot G-ben. Ez a részgráf összefüggő, valamint minden terminált tartalmaz. A költsége legfeljebb annyi, mint az F' c'-költsége. Ugyan tartalmazhat köröket, de tartalmazni fog egy F Steiner-fát G-ben, melyre $c(F) \leq c'(F')$. Azaz $OPT(I) \leq OPT(I')$. Ez mutatja, hogy az átalakítás nem változtatja meg az optimum értékét. Az is következik, hogy a

metrikus változatra adott közelítés hányadosa az eredeti feladatra is érvényes. •

Célunk a metrikus Steiner-fa feladatra egy 2-közelítő algoritmus kidolgozása. Legyen $G = (V, E)$ teljes gráf, c, $V = (T \cup S)$ az input, ahol c metrikus. Jelölje OPT az optimum értékét.

Az algoritmus egyszerű: keressünk egy minimális költségű F_m feszítőfát a terminálok által feszített $G[T]$ részgráfban a c költségekre. Ilyen fa polinomidőben kapható a mohó algoritmussal. Természetesen F_m egy Steiner-fa, azaz megengedett megoldás. Előfordulhat, hogy nem optimális (keressünk példát!), azonban a következő egyenlőtlenség teljesül (amelyből a kívánt közelítési hányadost kapjuk).



7.1. ábra. A bal oldalon a Steiner-fa feladat inputja látható: a hatpontú gráfban a terminálok halmaza $T = \{r_1, r_2, r_3, r_4\}$, a c élköltségeket az élekre írt számok jelölik. Középen a teljes gráf a T halmazon, amelyben a súlyok a G-beli távolságok c-re nézve, valamint egy minimális súlyú F feszítőfa. A jobb oldali ábrán az F éleinek megfelelő utak uniója G-ben. Ez az algoritmus által megtalált Steiner-fa.

A bizonyításban használni fogjuk a gráfelmélet egy alapvető eredményét: ha a gráf összefüggő és minden pontjának foka páros, akkor van benne olyan zárt séta, amely a gráf minden élét pontosan egyszer tartalmazza (Euler tétele Euler-sétákról).

7.2.2. Tétel.

$$c(F_m) \leq 2OPT$$

Bizonyítás: Legyen F egy optimális megoldás. Duplázzuk meg az éleit. Ekkor olyan F_{dupla} gráfot kapunk, amely összefüggő, és amelyben minden pont foka páros. Benne van minden terminál, valamint költsége $2c(F)$.

Keressünk ebben a gráfban egy Euler-sétát valamely $t \in T$ pontból indulva (például az éleket egy DFS bejárás szerint követve). Ezután az Euler-sétából élek és pontok elhagyásával készítsünk egy C kört a T pontjain. (A t-ből indulva kövessük a sétát. Minden Steiner-pontot és korábban már érintett terminált vágjunk le, azaz hagyjuk el a sétából és helyette illesszünk be egy új élt, amely közvetlenül a sétán utána következő pontba vezet. A gráf teljes, tehát ez megtehető.)

Mivel c metrikus, ezek a levágások nem növelik a költséget. A C egy élét elhagyva egy F' fát (utat) kapunk a terminálokon, ami megengedett megoldás lesz. A fenti megfontolások alapján

$$c(F_m) \leq c(F') \leq c(C) \leq c(F_{dupla}) = 2c(F) = 2OPT.$$

•

A fentiek alapján a Steiner-fa feladatra adott 2-közelítő algoritmusunk lépései:

(i) Minden pontpárra számoljuk ki a legrövidebb út hosszát (Dijkstra-algoritmussal),

(ii) Keressünk minimális feszítő fát a T halmazon,

(iii) Határozzuk meg a kapott fa éleinek megfelelő részgráfot G-ben. Élek elhagyásával redukáljuk Steiner-fa megoldássá.

Ha c metrikus, akkor az (i) és (iii) lépésekre nincs szükség.

Kérdés, hogy van-e olyan éles példa, ami mutatja, hogy az algoritmusunk hányadosa nem javítható. Legyen a G-ben teljes gráfban n terminál és egy s Steiner-pont. Az s-re illeszkedő élek költsége legyen 1, minden más elre 2. Ez metrikus. Ekkor a T halmazon vett minimális feszítőfa költsége $2(n-1)$, míg $OPT = n$.

7.3. Feladatok

7.3.1. Feladat. *Adjunk polinomiális algoritmust a Steiner-fa feladatra abban az esetben, amikor* $|T| = 3$.

7.3.2. Feladat. *Adjunk polinomiális algoritmust a Steiner-fa feladatra abban az esetben, amikor* $|V - T| = 3$. *(Mit tudunk mondani, ha* $|V - T|$ *egy fix konstans?)*

7.3.3. Feladat. *Tekintsük a következő optimalizálási feladatot: adott a* $D = (V, A)$ *digráf. Keressünk egy maximális élszámú aciklikus részgráfot* D*-ben. Adjunk erre a feladatra* $\frac{1}{2}$*-közelítő algoritmust!*

7.3.4. Feladat. *Legyen a* $G = (V, E)$ *gráf élein* $c\colon E \to \mathbb{R}_+$ *egy nem-negatív súlyfüggvény. Keressünk a* V *ponthalmaznak egy olyan* $V = V_1 \cup V_2$ *partícióját, amelyre a* V_1 *és* V_2 *között futó élek összsúlya maximális. Adjunk erre a maximalizálási feladatra* $\frac{1}{2}$*-közelítő algoritmust!*

7.3.5. Feladat. *Igazoljuk, hogy a következő algoritmus 2-közelítő a minimális méretű lefogó ponthalmaz feladatra (egy* G *összefüggő gráfban): keressünk egy* T *DFS-feszítőfát* G*-ben. Legyen az output a* T *fa azon pontjainak halmaza, amelyek nem levelek.*

7.3.6. Feladat. *Egy* G *gráfban maximális méretű független ponthalmazt szeretnénk találni. Mutassuk meg, hogy a mohó algoritmus nem* α*-közelítő semmilyen* $\alpha > 0$ *valós számra.*

7.3.7. Feladat. *Legyen* $G = (V, E)$ *gráf,* $S \subset V$, $R \subset V$ *a csúcsok két nemüres, egymástól diszjunkt részhalmaza,* c *pedig nemnegatív élsúlyfüggvény. Legyen* F *egy minimális költségű részgráfja* G*-nek, melyre teljesül, hogy minden* $u \in R$ *esetén létezik* u*-ból* S*-be vezető út.*

a) Ha $S \cup R = V$, *akkor* F *polinom időben megtalálható.*

b) Általános esetben adjunk 2*-közelítő algoritmust!*

7.3.8. Feladat. *Adottak az* $S = \{1, 2, \ldots, n\}$ *alaphalmaznak* $X_1, X_2, \ldots, X_m$ *részhalmazai. A lehető legkevesebb* X_i *halmaz segítségével szeretnénk lefedni* S*-et. Mutassuk meg, hogy a mohó algoritmus (ami minden lépésben azt az* X_i *halmazt választja, ami a legtöbb még fedetlen elemet fedi) nem* α*-közelítő semmilyen* $\alpha > 0$ *valós számra.*

8. fejezet

Párosítások és élszínezések

Feltételezzük, hogy az olvasó ismeri Kőnig, Frobenius, Hall és Ore páros gráfok párosításaira vonatkozó klasszikus tételeit, valamint a javító utas algoritmust maximális párosítás keresésére páros gráfban. Ebben a fejezetben az ezekhez szorosan kapcsolódó élszínezésekkel foglalkozunk.

8.1. Páros gráfok élszínezései

A $G = (V, E)$ gráf egy **élszínezése** az E élhalmaz egy $\{M_1, M_2, \ldots, M_p\}$ partíciója, melyben minden M_i egy párosítás. Mindegyik M_i-t azonosíthatunk egy színnel. Illeszkedő élek színe ekkor különböző. A G **élszínezési száma**, jelölje $\chi'(G)$, a legkisebb olyan p, amelyre van G-nek p elemű élszínezése. Jelölje $\Delta(G)$ a G gráf maximális fokszámát. Világos, hogy

$$\chi'(G) \geq \Delta(G).$$

A K_3 gráf mutatja, hogy itt állhat szigorú egyenlőtlenség. Páros gráfokra azonban ez nem fordulhat elő, amint azt Kőnig alábbi tétele igazolja.

8.1.1. Tétel. *Bármely $G = (A, B; E)$ páros gráfra*

$$\chi'(G) = \Delta(G),$$

azaz az élszínezési szám megegyezik a maximális fokszámmal.

Bizonyítás: Figyeljük meg, hogy $\Delta(G) \leq 2$ esetén a tétel könnyen belátható, hiszen ekkor a G minden komponense egy út, vagy páros kör.

Vegyünk $\Delta(G)$ párosítást G-ben, amelyek uniója a lehető legtöbb élt tartalmazza. Ha G minden éle benne van az unióban, akkor ezek a párosítások egy élszínezést alkotnak és készen vagyunk. Ha nem, akkor valamely $e = uw$ él egyikben sincs benne. Legalább egy szín, mondjuk a piros, nem illeszkedik u-ra. Hasonlóan, legalább egy szín, mondjuk a kék, nem illeszkedik w-re. (Persze piros≠kék, ellenkező esetben az e él kiszínezhető lenne pirossal, ami ellentmondana a maximalitásnak.)

Tekintsük a G azon H részgráfját, amely a piros és kék élekből áll, plusz az e élből. Ekkor $\Delta(H) = 2$, ezért $\chi'(H) = \Delta(H) = 2$. A H éleit ki tudjuk színezni pirossal és kékkel. A H éleit így átszínezve olyan párosításokat kapunk, amelyek uniója több élt tartalmaz (hiszen már e is benne van), ami ellentmondás. •

A fenti bizonyítás algoritmikus, polinomiális algoritmust ad egy páros gráf optimális élszínezésének meghatározására.

8.2. Az órarendkészítési feladat

Az élszínezések látványos alkalmazása az órarendkészítési feladat. Tegyük fel, hogy egy adott napra szeretnénk megtervezni egy órarendet egy iskolában. A tanítás reggel 8-től kezdődik, minden óra (szünettel együtt) 60 perc. Az iskolában n osztály van és m tanár. Adott az aznap tartandó órák listája, mondjuk egy n-szer m-es M mátrix formában, ahol $M[i,j]$, azaz az i-edik sor j-edik eleme 1, ha a j-edik tanárnak van aznap órája az i-edik osztállyal, különben 0. Cél olyan órarendet készíteni, amelyben a tanítás aznap a lehető leghamarabb fejeződik be.

A megoldáshoz készítsünk egy $G = (V, E)$ páros gráfot, amelyben az egyik színosztály az osztályoknak, a másik a tanároknak felel meg, továbbá pontosan akkor van él az i és j pontok között, ha $M[i,j] = 1$. Figyeljük meg, hogy G egy élszínezése k színnel megfelel egy olyan órarendnek, amelyben k óra alatt befejeződik a tanítás, és fordítva: minden egyes párosítás megfelel egy egyórás időintervallum tanár–osztály hozzárendelésének. Ez alapján Kőnig tétele és a megfelelő élszínező algoritmus megoldja a feladatot. Az optimum a G maximális fokszáma.

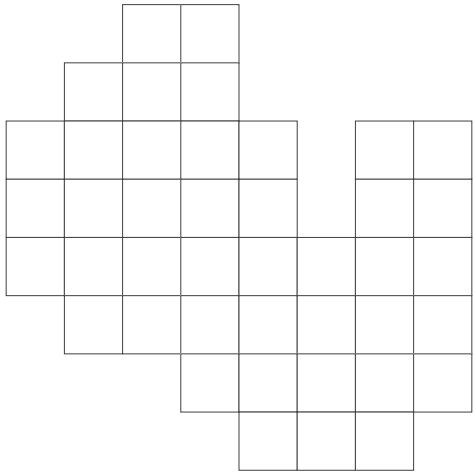
Ez a módszer akkor is alkalmazható, ha egy tanár egy adott osztályt aznap többször is tanít (például más tárgyakra). Legyen k az optimum. Azt is feltehetjük, hogy az optimális megoldásunkban bármely két színosztály (avagy párosítás) mérete legfeljebb eggyel tér el. Ekkor minden színosztály mérete $\lfloor \frac{|E|}{k} \rfloor$ vagy $\lceil \frac{|E|}{k} \rceil$, azaz az órarend megvalósításához $\lceil \frac{|E|}{k} \rceil$ tanteremre van szükség, ami a lehető legkevesebb.

8.3. Feladatok

8.3.1. Feladat. *A G páros gráfban pontosan egy teljes párosítás van. Mutassuk meg, hogy van G-ben elsőfokú pont!*

8.3.2. Feladat. *Bizonyítsuk be, hogy ha egy páros gráfbeli ponthalmaz fedhető egy párosítással, akkor fedhető maximális párosítással is!*

8.3.3. Feladat. *Kiparkettázható az alábbi ábrán látható alakzat dominókkal (2 szomszédos négyzetből álló téglalapokkal)?*



8.1. ábra

8.3.4. Feladat. *Tegyük fel, hogy a G páros gráf élei kiszínezhetők k színnel. Mutassuk meg, hogy van olyan színezés, k színnel, amelyben bármely két színosztály mérete legfeljebb eggyel tér el.*

9. fejezet

Hálózati folyamok

Ez a fejezet a kombinatorikus optimalizálás egyik központi problémájával foglalkozik: hálózati folyamokkal és az ezekhez kapcsolódó hatékony algoritmusokkal. Azt is kijelenthetjük, hogy a jegyzetben vizsgált fogalmak és algoritmusok közül ezeknek van a legszélesebb alkalmazási területe.

9.1. Folyamok és vágások

Egy $D = (V, A)$ irányított gráf és az élein értelmezett valamely $x : A \to \mathbb{R}$ függvény és $S \subseteq V$ ponthalmaz esetén legyen

$$d_x^-(S) = \sum(x(uv) : uv \in A, uv \text{ belép } S\text{-be}),$$

valamint legyen

$$d_x^+(S) = d_x^-(V - S).$$

Tekintsünk egy $D = (V, A)$ digráfot, melyben adott egy kitüntetett $s \in V$ **forrás** és egy $t \in V$ **nyelő** pont. ($s \neq t$.) A D-ben egy **folyam** egy olyan $x : A \to \mathbb{R}$ függvény, amelyre

(i) $x(uv) \geq 0$ minden $uv \in A$ élre, és

(ii) minden $v \in V - \{s, t\}$-re

$$d_x^-(v) = d_x^+(v)$$

teljesül.

A (ii)-ben szereplő egyenlőség a **megmaradási szabály** (conservation rule). Az $x(uv)$ a folyam **értéke** az $uv \in A$ élen. Az x folyam **egészértékű**, ha minden $uv \in A$ élen a folyam értéke egész szám.

Tegyük fel, hogy a D élein adott egy $g : A \to \mathbb{R}_+$ nemnegatív **kapacitás függvény**. Ekkor azt mondjuk, hogy x **megengedett folyam** (g-re nézve), ha $x \leq g$, azaz minden uv élre $x(uv) \leq g(uv)$. Egy s-et tartalmazó, de t-t nem tartalmazó $S \subset V$ halmazt nevezzünk $s\bar{t}$**-halmaznak**, vagy s-t**-vágásnak**, vagy néha csak röviden **vágásnak**. Egy S **vágás kapacitása** a $d_g^+(S)$ érték.

Az x folyam **nagysága** a $val(x) = d_x^+(s) - d_x^-(s)$ szám, a forrásból való nettó kiáramlás.

9.1.1. Lemma. *Minden x megengedett folyamra és minden S s-t-vágásra*

$$val(x) \leq d_g^+(S).$$

Bizonyítás:

$$\begin{aligned} val(x) &= d_x^+(s) - d_x^-(s) \\ &= d_x^+(s) - d_x^-(s) + \sum_{v \in S-\{s\}} (d_x^+(v) - d_x^-(v)) \\ &= \sum_{v \in S} (d_x^+(v) - d_x^-(v)) = d_x^+(S) - d_x^-(S) \\ &\leq d_x^+(S) \leq d_g^+(S), \end{aligned}$$

ahol az utolsó két egyenlőtlenség az (i) és (ii) tulajdonságokból következik. •

A lemma bizonyítása azt is mutatja, hogy tetszőleges S $s\bar{t}$-halmaz és x folyam esetén

$$d_x^+(S) - d_x^-(S) = d_x^+(s) - d_x^-(s),$$

azaz minden S $s\bar{t}$-halmazra a $d_x^+(S) - d_x^-(S)$ értékkel definiált nettó kiáramlás független S választásától és megegyezik a folyam nagyságával.

Az is megállapítható, hogy egy x folyam maximális nagyságú, ha létezik egy olyan S $s\bar{t}$-halmaz, amelyre teljesülnek az alábbi **optimalitási feltételek**.

(a) $x(a) = g(a)$ minden a élre, amely kilép S-ből, és
(b) $x(a) = 0$ minden a élre, amely belép S-be.

A D, s, t, g együttese egy **hálózat**. Az optimalizálási feladat, amelyet vizsgálni fogunk az, hogy adott hálózatban hogyan találhatunk maximális nagyságú megengedett folyamot.

A következő, Fordtól és Fulkersontól származó tételt fogjuk igazolni.

9.1.2. Tétel. *Akkor és csak akkor létezik k nagyságú megengedett folyam valamely $k \geq 0$ számra, ha minden S $s\bar{t}$-halmazra $d_g^+(S) \geq k$. Ha e feltétel teljesül, g egészértékű, k pedig egész, úgy a folyam is választható egészértékűnek.*

A tételt néha az alábbi ekvivalens alakban említik, melyet gyakran MFMC tételnek neveznek. (Az MFMC elnevezés az angol Max-flow Min-cut rövidítése.)

9.1.3. Tétel. *A megengedett st-folyamok maximális nagysága egyenlő a $d_g^+(S)$ értékek minimumával, ahol a minimum az összes S $s\bar{t}$-halmazra megy. Ha g egészértékű, úgy a maximum egészértékű folyamon is felvétetik.*

Az MFMC tételt először racionális g kapacitás függvényre igazoljuk a Ford–Fulkerson növelő utas algoritmus segítségével.

9.2. A növelő utak módszere

A tételt először csak egészértékű (illetve racionális) g kapacitás függvény esetén igazoljuk, majd tetszőleges g-re is. A bizonyítás algoritmikus.

A Fordtól és Fulkersontól származó algoritmus tetszőleges x megengedett st-folyamból indul ki (például $x \equiv 0$), és azt iteratívan javítja, mindaddig, amíg egy olyan x, S párt talál, amelyek teljesítik az optimalitási feltételeket.

Adott x megengedett folyamhoz készítsünk el egy $D_x = (V, A_x)$ digráfot a következő módon. Egy uv él A_x-hez tartozik, ha vagy

(i) $uv \in A$ és $x(uv) < g(uv)$,

és ekkor ezen élét D_x-nek **előre** élnek hívjuk, vagy

(ii) $vu \in A$ és $x(vu) > 0$,

és ekkor uv neve **hátra** él. Jelölje S az s-ből D_x-ben irányított úton elérhető pontok halmazát.

1. eset $t \notin S$, azaz t nem érhető el s-ből. Mivel D_x semelyik éle sem lép ki S-ből, ezért D-ben minden S-ből kilépő él telített (azaz $x(uv) = g(uv)$) és minden S-be belépő uv élen $x(uv) = 0$. Vagyis az (a) és (b) optimalitási feltételek teljesülnek: az adott x folyam nagysága egyenlő $d_g^+(S)$-sel. Az algoritmus ekkor leáll, az x folyam nagysága maximális.

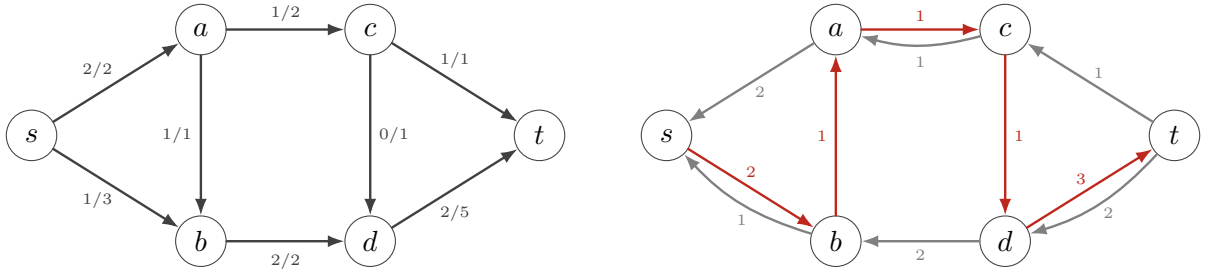
2. eset $t \in S$, azaz t elérhető s-ből. Legyen P tetszőleges s-ből t-be vezető irányított út D_x-ben.

Legyen $\Delta_1 = \min\{g(uv) - x(uv) : uv$ előre éle P-nek$\}$ és $\Delta_2 = \min\{x(v,u) : uv$ hátra éle P-nek$\}$. Legyen $\Delta = \min\{\Delta_1, \Delta_2\}$. Ekkor Δ pozitív. Nevezzük P egy élét **kritikusnak**, ha Δ ezen az élen vétetik fel.

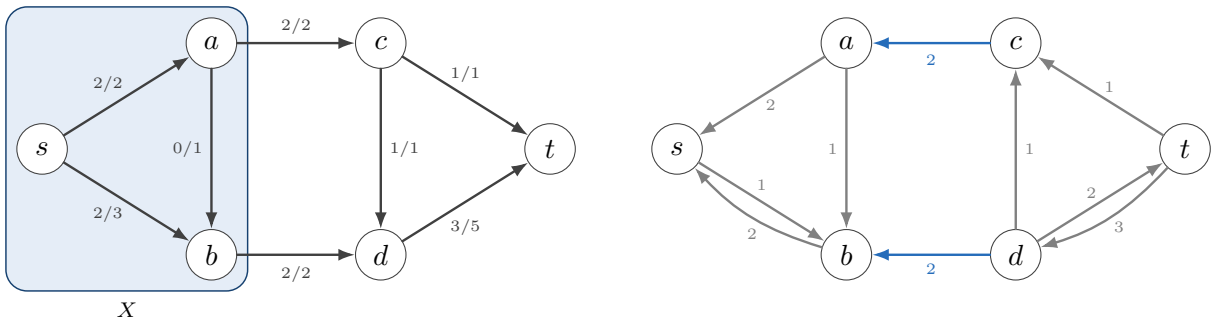
Módosítsuk x-t a következőképp. Ha uv előre éle P-nek, úgy a D uv élén növeljük $x(uv)$-t Δ-val. Ha uv hátra éle P-nek, úgy a D vu élén csökkentsük $x(vu)$-t Δ-val. Könnyen látható, hogy a módosított x' megengedett folyam lesz, amelynek nagysága Δ-val nagyobb, mint x-é. Figyeljük meg, hogy ha g egészértékű, akkor $\Delta \geq 1$, így a 2. eset csak véges sokszor fordulhat elő, vagyis véges sok növelés után az 1. eset következik be, amikor az algoritmus leáll. Tehát egész kapacitások esetén az MFMC tétel bizonyítást nyert.

Amennyiben g racionális, a nevezők legkisebb közös többszörösével a kapacitásokat végigszorozva visszajutunk az egész kapacitású esethez.

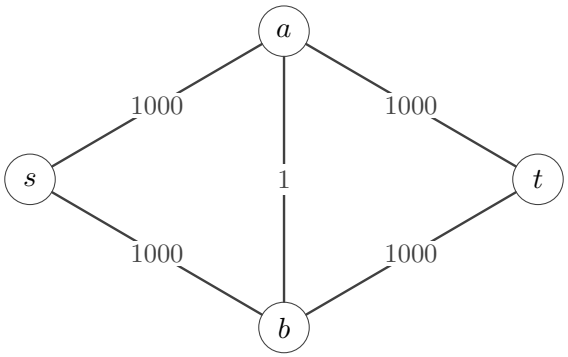
Megjegyzendő, hogy ha g irracionális, akkor a fenti eljárás nem feltétlenül ér véget véges sok lépésben, sőt, a folyamnagyságok konvergálhatnak olyan számhoz is, amelynél van nagyobb folyam. Másik hátrány, hogy még egész kapacitások esetén is az iterációk száma arányos lehet az előforduló legnagyobb kapacitás nagyságával. Így az algoritmus lépésszáma az input méretének exponenciális függvényével arányos, azaz nem polinomiális. (Egy szám nagysága a jegyei számának exponenciális függvénye.)



9.1. ábra. Egy x hálózati folyam és a D_x segédgráf. Az első ábrán az éleken a folyam értéke és az él kapacitása szerepel. A második ábrán az élek és a rajtuk levő számok az él lehetséges felhasználási irányát mutatják D_x-ben és a növelés vagy csökkentés maximumát az élen. A piros élek egy javító utat jelölnek.



9.2. ábra. Az első ábrán az az x' folyam látható, amelyet az előző ábrán látható folyamból kapunk az ott jelölt növelő úton javítva egy egységnyit. Az X halmaz egy minimális vágást alkot a hálózatban. Ez a halmaz megegyezik a második ábrán látható új $D_{x'}$ segédgráfban az s-ből elérhető pontok halmazával.



9.3. ábra. A példa alapgráfja a kapacitásokkal.

Példa: Legyen a $D = (V, A)$ digráfban $V = \{s, t, a, b\}$, $A = \{sa, sb, ab, at, bt\}$, valamint a kapacitás minden élen 1000, kivéve az ab élt, amelyen 1. Ekkor előfordulhat, hogy a 2000 nagyságú maximális folyam eléréséhez a folyam növelő szubrutint 2000-szer kell futtatnunk (ha a növelő utakat kedvezőtlenül választjuk).

9.2.1. Skálázási technika

Az alábbiakban bemutatunk egy ügyes fogást, amelynek segítségével bizonyos eljárásokat polinomiális futásidejűvé lehet tenni. Használatát a maximális folyam problémán szemléltetjük, mert ott igen egyszerű, de számos alkalommal bonyolultabb körülmények között is használható.

Tételezzük fel, hogy a kapacitások egész számok és kettes számrendszerben vannak megadva. A legnagyobb kapacitás álljon M jegyből. Azaz minden kapacitásra gondolhatunk úgy, mint egy M hosszú 0–1-sorozat, melynek első néhány jegye lehet 0. Összesen M darab folyam problémát fogunk megoldani, mindegyikben a megelőzően megkapott maximális folyamot használjuk kiindulási folyamként. Jelölje g_i azt a kapacitás függvényt, amely úgy áll elő, hogy minden élen az eredeti kapacitásnak (balról) az első i jegyét tekintjük, míg a többit eltöröljük. Tegyük fel, hogy a g_i kapacitás függvényre nézve már meghatároztuk az x_i maximális folyamot. Ekkor $2x_i$ megengedett folyam a g_{i+1}-re nézve. A $2x_i$-ből kiindulva alkalmazzuk a fent leírt növelő út módszert a g_{i+1} kapacitás függvényre vonatkozólag.

Miután minden e élre $g_{i+1}(e)$ értéke vagy $2g_i(e)$ vagy $2g_i(e)+1$, legfeljebb élszámnyi növelés után megkapjuk az x_{i+1} maximális folyamot (a g_{i+1}-re nézve). Összesen tehát legfeljebb $M|A|$ növelés segítségével megkonstruáltunk egy eredeti kapacitásokra vonatkozó maximális folyamot.

9.2.2. Legrövidebb növelő utak

A fenti eljárás hátránya, hogy csak egész (és így racionális) kapacitásokra működik. Ezen nehézség leküzdésére Edmonds és Karp azt javasolták, hogy minden iterációban a legkisebb élszámú növelő utat válasszuk. Ez az egyszerű megszorítás lehetővé teszi, hogy a Ford–Fulkerson algoritmus bonyolultságát $|V|$ és $|A|$ polinomjával korlátozzuk, függetlenül a kapacitások nagyságától. (Ezt persze úgy értve, hogy a számokkal végzett alapműveleteket egyetlen lépésnek tekintjük.)

9.2.1. Tétel. *Ha a Ford–Fulkerson féle növelő utas algoritmusban mindig a legrövidebb növelő utat használjuk, akkor az eljárás tetszőleges g kapacitás függvény esetén legfeljebb $O(|V||A|)$ növelés után véget ér.*

Bizonyítás: Jelölje $\sigma_x(v)$ a v távolságát D_x-ben s-től. (Ha nincs egyáltalán s-ből v-be út, akkor $\sigma_x(v)=\infty$). Legyen P egy legrövidebb út D_x-ben s-ből t-be. Ekkor P mindegyik uv élére, $\sigma_x(v)=\sigma_x(u)+1$.

9.2.2. Lemma. *Amikor P mentén végrehajtunk egy növelést, a $\sigma_x(v)$ érték semmilyen v-re sem csökken.*

Bizonyítás: Nézzük meg, milyen hatással van a növelés a D_x segédgráfra. Mivel a folyamot D-nek csak olyan élein változtattuk, melyek P éleinek felelnek meg, D_x a P éleinél változhat. A D'_x lehetséges új élei csak P bizonyos éleinek megfordítottjai lehetnek. Emellett a P kritikus élei (ahol Δ felvétetik) eltűnnek D_x-ből. A v pont s-től való távolsága csak akkor csökkenhetne, ha legalább egy olyan uw élt adnánk a segédgráfhoz, melyekre $\sigma_x(w)>\sigma_x(u)+1$. Mivel ilyen új él nem lehet, a lemma állítása következik. •

A növelések sorozatát fázisokra bontjuk. Egy fázis során $\sigma_x(t)$ ugyanaz marad. A lemma szerint legfeljebb $|V|-1$ fázis lehetséges.

9.2.3. Lemma. *Egy fázison belül legfeljebb $|A|$ növelésre kerülhet sor.*

Bizonyítás: Tekintsünk egy fázist. Jelölje $\sigma(v)$ a v pont távolságát s-től az fázis kezdetén az aktuális segédgráfban. Nevezzünk egy uv élt **szorosnak**, ha $\sigma(v)=\sigma(u)+1$. A fázis során csupán szoros éleket használunk. Tudjuk, hogy egy növelés legalább egy szoros élt eltüntet az aktuális segédgráfból és nem hoz be új szoros élt. Mivel a segédgráfnak legfeljebb $|A|$ darab szoros éle van, a lemma következik. •

Mindezeket összevetve kapjuk, hogy legfeljebb $|V||A|$ növelésre van szükségünk. •

Ennek alapján megállapíthatjuk, hogy az algoritmus Edmonds–Karp változatának lépésszáma $O(|V||A|^2)$, hiszen egyetlen növelés (azaz egy s-ből t-be vezető út megkeresése) $O(|A|)$ lépést igényel.

Miután a fenti algoritmus futása során a rendelkezésre álló legrövidebb növelő utak közül bármelyiket választhatjuk, a végül kapott maximális folyam függ ezen választásoktól. A végső S azonban nem!

9.2.4. Feladat. (a) *A végül kapott minimális vágás független az algoritmus futásától.* (b) *Ha X és Y minimalizálja az $d_g^+(Z)$ értéket az összes $s\bar{t}$-halmazra, akkor $X \cap Y$ és $X \cup Y$ is minimalizáló $s\bar{t}$-halmazok.* (c) *A minimalizáló halmazok metszete S.*

9.2.5. Feladat. *Adott e élre hogyan lehet eldönteni, hogy* (a) *létezik-e olyan maximális folyam, amely telíti e-t,* (b) *minden maximális folyam telíti e-t.*

9.2.6. Feladat. *Algoritmikusan határozzuk meg az összes (x, y) rendezett pontpárt, amelyre létezik olyan X halmaz, hogy $s, x \in X, y \in V - X$ és X minimális vágást határoz meg.*

9.2.7. Feladat. *Adott két kapacitás függvény esetén algoritmikusan döntsük el, hogy létezik-e olyan S $s\bar{t}$-halmaz, amely mindkét kapacitás függvényre nézve minimális vágást határoz meg.*

9.2.8. Feladat. *Adott g_1 és g_2 kapacitás függvények esetén keressünk olyan g_1-re nézve minimális $s\bar{t}$-vágást, amely a g_2-re nézve a lehető legkisebb.*

9.2.9. Feladat. *Digráfban keressünk két diszjunkt halmazt, melyek egyike $s\bar{t}$-halmaz, másika $t\bar{s}$-halmaz, úgy, hogy befok összegük minimális.*

9.2.10. Feladat. *A D digráfban adott s, t, valamint $k \geq 1$ kapacitás függvény, $g_1, g_2, \ldots, g_k$. Adjunk polinomiális algoritmust, amely eldönti, hogy van-e olyan S vágás, amely mindegyik kapacitás függvényre nézve minimális vágás.*

Az x folyamot **út-folyamnak** nevezzük, ha x csak egy s-ből t-be vezető út mentén pozitív. Az x **kör-folyam**, ha egy irányított kör mentén pozitív.

9.2.11. Feladat. *Igazoljuk, hogy minden nem-negatív folyam kör- és út-folyamok összege.*

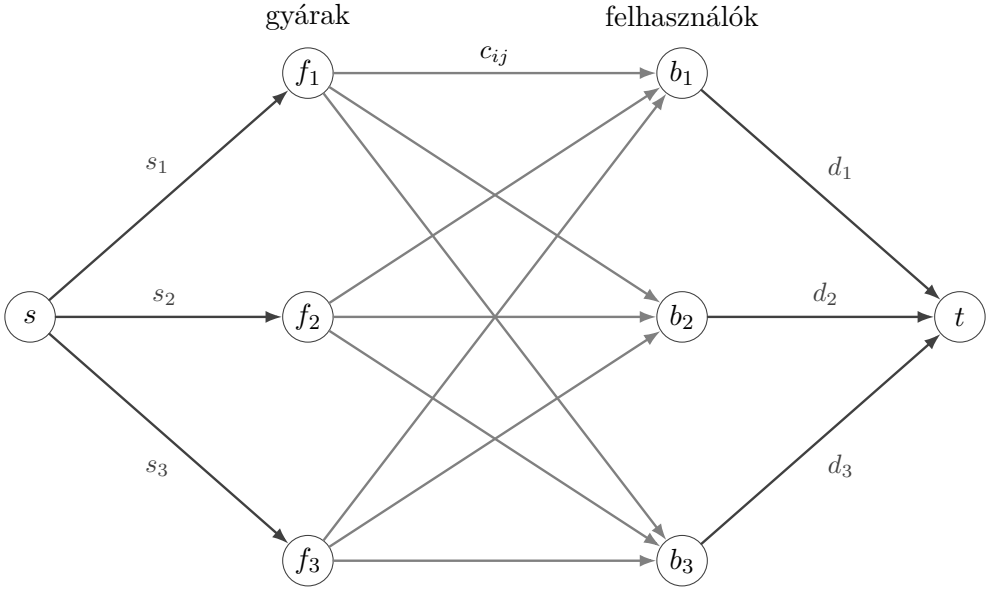
9.3. Folyamok alkalmazásai

9.3.1. A szállítási feladat

Ugyanazt a terméket állítja elő m gyár, amelyek n felhasználónak szállítanak. Az i gyár hetente s_i tonnát tud termelni. A j felhasználónak d_j tonnára van szüksége minden héten. Az i gyárból a j felhasználónak hetente legfeljebb c_{ij} tonna szállítható. Minden igényt ki tudnak elégíteni a gyárak?

Tekintsük azt a hálózatot, melyben minden gyárnak és minden felhasználónak megfelel egy-egy pont ($f_1, ..., f_m$ és $b_1, ..., b_n$), valamint tartalmaz egy további s forrást, amely minden f_i-vel össze van kötve, egy t nyelőt, amelybe minden b_j-ből megy él, továbbá minden f_i-ből megy él minden b_j-be.

Az s-re illeszkedő élek kapacitása s_i, a t-be menő éleké d_j, az $f_i b_j$ éleken pedig c_{ij}.



9.4. ábra. A szállítási feladat hálózata.

Ekkor pontosan akkor teljesíthető minden igény, ha van egy $\sum_{j=1}^{n} d_j$ nagyságú megengedett folyam s-ből t-be. Ennél nagyobb folyam nem lehet, így a maximális folyam algoritmus megoldja a feladatot. Ha van ekkora nagyságú x folyam, akkor az x értékei a megfelelő éleken megadják, hogy melyik gyár mennyit gyártson, és ebből melyik felhasználónak mennyit szállítson.

9.3.2. Projekt kiválasztás

Tekintsük a következő optimalizálási feladatot: adott n projekt, valamint mindegyikhez a hozzá tartozó p_i, $1 \leq i \leq n$, profit. A profitok valós számok: negatív profitra tekinthetünk úgy, mint egy beruházás. Ezenkívül adott néhány olyan feltétel, valamely i, j párokra, melyek szerint egy i projekt csak akkor választható ki, ha a j-t is kiválasztjuk. A cél a projektek egy olyan részhalmazát kiválasztani, amelyekre a profitjaik összege maximális, és amely teljesíti a plusz feltételeket. Ez utóbbi esetben a részhalmaz **megengedett**.

A feladatot hálózati folyamokkal oldhatjuk meg. Legyen $D = (V \cup \{s, t\}, A)$ digráf, melyben a V pontjai a projektek, s a forrás, t a nyelő. Minden „az i projekt csak akkor választható ki, ha a j-t is kiválasztjuk” feltételhez betesszük az ij élt. Figyeljük meg, hogy az $X \subseteq V$ pontosan akkor felel meg megengedett projekthalmaznak, ha kifoka nulla (avagy **nyelő halmaz**). Ezen felül minden si élt behúzunk, amelyre $p_i \geq 0$, és minden it élt, amelyre $p_i < 0$, $1 \leq i \leq n$.

Kapacitásokat is definiálunk. A projektek közti ij élekre a $c(ij)$ kapacitás $+\infty$ (vagy egy kellően nagy szám, mondjuk a profitok abszolút értékeinek összege), egy si él $c(si)$ kapacitása p_i, egy it élen a $c(it)$ kapacitás $-p_i$. Ezzel egy hálózatot definiáltunk. Figyeljük meg, hogy a minimális vágáskapacitás véges (pl. $\{s\}$ egy véges st-vágás) és ha egy vágásból kilép egy olyan él, amely valamely feltételnek felel meg, akkor az biztos nem minimális vágás.

A döntő észrevétel az, hogy ha U egy minimális vágás, akkor $U - \{s\}$ egy maximális profitú megengedett projekthalmaz. Ennek a belátásához vizsgáljunk meg egy véges kapacitású U vágást. Legyen $Q = \sum_{i:\ p_i \geq 0} p_i$ a nemnegatív profitok összege. Ekkor a $c(U)$ kapacitásra teljesül, hogy

$$\sum_{a \in \delta^+(U)} c(a) = \sum_{i \in V-U,\ p_i \geq 0} p_i + \sum_{i \in U-\{s\},\ p_i < 0} (-p_i) = Q - \sum_{i \in U-\{s\},\ p_i \geq 0} p_i - \sum_{i \in U-\{s\},\ p_i < 0} p_i =$$

$$= Q - \sum_{i \in U - \{s\}} p_i.$$

Vegyük észre, hogy amit levonunk Q-ból (ami egy fix szám), az az $U - \{s\}$-beli projektek össz profitja. Ezért a minimális kapacitású vágás egy maximális profitú, azaz optimális megoldásnak felel meg.

9.3.3. Nyerhet-e a csapatunk?

Képzeljünk el egy olyan bajnokságot, ahol n csapat indul, minden hétvégén van forduló, ahol a csapatokból alkotott párok játszanak egymás ellen. A győztes egy pontot kap, a vesztes nullát. Döntetlen nincs. A szezon végén az a csapat nyer, amelynek a legtöbb pontja van. Több nyertes is lehet.

A kedvenc csapatunk, nevezzük Z-nek, indul a bajnokságban. A szezon közben, amikor már több forduló lement, és előttünk van a tabella pillanatnyi állása, azt a kérdést tesszük fel magunknak, hogy van-e még esélye a csapatunknak a végső győzelemre, azaz elképzelhető-e olyan kimenetele a további fordulóknak, amely után az egyik nyertes Z. Ismerjük az összes forduló menetrendjét.

Nézzük például az alábbi egyszerűsített példát. Négy csapat indul, A, B, C és Z, amelyek pontszáma $10, 9, 9, 8$. A hátralevő meccsek: $A-B, A-C, B-C, B-Z, C-Z$. Elsőre azt gondolhatnánk, hogy a csapatunk még nyerhet, hiszen még két meccse van, melyeket ha megnyer, 10 pontja lesz és ezzel az élre kerülhet. De nem ez a helyzet. Mivel a másik három csapatnak most összesen 28 pontja van és még 3 meccset játszanak egymással, azaz 31 pontjuk lesz. Valamelyiküknek tehát mindenképp legalább 11, azaz Z nem nyerhet.

9.3.1. Feladat. *Dolgozzunk ki egy hatékony algoritmust, amely eldönti, nyerhet-e a csapatunk. (Segítség: feltehetjük, hogy Z minden további meccsét megnyeri. Készítsünk egy hálózatot, melyben az összes többi csapatnak megfelel egy pont, valamint minden olyan párnak, akik még játszanak egymás ellen. Vegyünk fel még egy extra forrás és nyelő pontot. Az élek és a kapacitások alkalmas megválasztásával érjük el, hogy pontosan akkor legyen a hálózatban olyan folyam, amely minden forrásból kilépő élt telít, ha Z-nek van esélye nyerni.*

Ha Z nem nyerhet, hasznos lehet erre egy közérthető bizonyítékot találni. Legyen az X csapat pillanatnyi pontszáma w_X, a hátralevő X, Y közötti meccsek száma $g_{X,Y}$, és legyen a Z által elérhető maximális pontszám m. (Azaz $m = w_Z + \sum_X g_{Z,X}$.)

9.3.2. Feladat. *Mutassuk meg, hogy Z pontosan akkor nem nyerhet, ha van a csapatoknak egy T halmaza (melyben nincs benne Z), amelyre*

$$\sum_{X \in \mathcal{T}} w_X + \sum_{\{X,Y\} \subseteq \mathcal{T}} g_{X,Y} > m|\mathcal{T}|.$$

9.3.4. Egy kombinatorikus alkalmazás

Ismert, hogy az MFMC tételből könnyen levezethető Menger tételének az irányított él- és pontváltozata is. Hasznos megfogalmazni ezeknek az alábbi közös általánosítását.

9.3.3. Tétel. *Legyen $D = (V, A)$ digráf és legyenek k, l pozitív egészek. Akkor és csak akkor létezik D-ben kl éldiszjunkt út s-ből t-be úgy, hogy minden csúcson legfeljebb l darab út halad keresztül, ha bárhogy kihagyva egy $X \subseteq V - \{s, t\}$ halmazt $(0 \leq |X| \leq k-1)$, a maradékban minden $s\bar{t}$-halmazból legalább $(k - |X|)l$ él kilép.*

Bizonyítás: Ha létezik a kívánt útrendszer, akkor ennek legfeljebb $|X|l$ tagja használhat X-beli pontot, tehát a maradék $(k-|X|)l$ darab út $D-X$-ben halad, és ezért $D-X$ minden $s\bar{t}$-halmazából legalább $(k-|X|)l$ él kilép, vagyis a feltétel szükséges.

Az elegendőség igazolásához feltehetjük, hogy s-be nem lép be él és t-ből nem lép ki. Használjuk a szokásos pontszéthúzós technikát, azaz minden $v \in V$ csúcsot helyettesítsünk két új csúccsal, melyeket jelölje v' és v''. Vezessünk l párhuzamos élt v'-ből v''-be, töröljük s'-t és t''-t, végül minden $uv \in E$ élre vezessünk egy élt u''-ből v'-be. Az így keletkező $D' = (V', E')$ digráfban legyen a g kapacitás függvény azonosan 1. A tétel feltételéből adódik, hogy minden $S' \subseteq V'$ $s''\bar{t}'$-halmazra $d_g^+(S) \geq kl$ a D'-ben. Ezért az MFMC tétel miatt létezik kl nagyságú folyam, amely egészértékű. Ez a folyam felbomlik körfolyamokra és kl útfolyamra, amely az eredeti D digráfban a konstrukció miatt kl élidegen útnak felel meg úgy, hogy minden csúcson legfeljebb l út halad keresztül. •

Figyeljük meg, hogy az $l = 1$ speciális esetben az irányított Menger tétel pontidegen változatát kapjuk k darab belsőleg pontdiszjunkt st út létezéséről, míg a $k = 1$ esetben az éldiszjunkt változatot l darab éldiszjunkt st út létezéséről. (Mivel ilyenkor összesen l élidegen utat keresünk, az a kívánság, hogy minden csúcson legfeljebb csak l út haladhat keresztül automatikusan teljesül.)

9.3.4. Tétel. *Egy $D = (V, A)$ digráf élei akkor és csak akkor színezhetők ki l színnel, hogy mindegyik színben létezzék k belsőleg pontdiszjunkt út s-ből t-be, ha bárhogyan kihagyva egy $X \subseteq V - \{s, t\}$ halmazt $(0 \leq |X| \leq k-1)$, a maradékban minden $s\bar{t}$-halmazból legalább $(k-|X|)l$ él kilép.*

Bizonyítás: Tekintsük az 9.3.3 tétel által biztosított kl élidegen utat. Feltehető, hogy a digráfban nincs is más él, mert azokat kitörölhetjük a feltétel megsértése nélkül. Ekkor minden $v \in V - \{s, t\}$ pont befoka megegyezik a kifokával (és legfeljebb l), továbbá s-ből kl él lép ki és t-be kl él lép be.

Készítsük el a következő páros gráfot. Minden $v \in V - \{s, t\}$ csúcsot helyettesítsünk két új csúccsal, melyeket jelölje v' és v''. Az s pontot helyettesítsük az $s'_1, \ldots, s'_k$ új pontokkal, míg t-t a $t''_1, \ldots, t''_k$ pontokkal. Legyen $l - \varrho(v) = l - \delta(v)$ párhuzamos él v' és v'' között minden v-re, valamint minden uv élre adjunk a gráfhoz egy $u'v''$ élt. Végül az s-ből kifutó kl darab élt osszuk k darab l-es csoportba tetszőlegesen, majd az i-dik csoportban lévő l darab sv él mindegyikére kössük össze s_i-t v''-vel. Hasonlóan, a t-be befutó kl élt osszuk k darab l-es csoportba tetszőlegesen, majd az i-dik csoportban lévő l darab ut él mindegyikére kössük össze u'-t és t_i-t. A keletkező páros gráf minden csúcsa l-ed fokú lesz. Így Kőnig élszínezési tétele miatt ennek élhalmaza felbontható l darab teljes párosításra. Könnyen ellenőrizhető, hogy egy teljes párosítás éleinek megfelelő élhalmaz az eredeti digráfban k belsőleg pontdiszjunkt utat definiál s-ből t-be. •

10. fejezet

Gráfok összefüggősége

10.1. Menger tétele

Az x folyam **útfolyam**, ha a pozitív értékű élek egy s-t utat alkotnak. Ekkor a folyam értéke az út minden élén ugyanakkora.

Egyszerű megfigyelés, hogy folyamok összege (ahol minden élre az összegben a folyam értéke a folyamok értékeinek összege) is folyam. Megmutatjuk, hogy minden folyam előáll útfolyamok összegeként.

10.1.1. Lemma. *Legyen x folyam a D, s, t, g hálózatban. Ekkor x felírható*

$$x = x_1 + x_2 + \dots x_t + x_0$$

alakban, ahol x_i $(1 \leq i \leq t)$ útfolyam, x_0 egy nulla nagyságú folyam, valamint $val(x) = \sum_1^t val(x_i)$. Továbbá, ha x egészértékű, akkor minden x_i választható egészértékűnek.

Bizonyítás: Legyen $D = (V, A)$. Indukció az élek számára. Az $|A| = 1$ eset világos. Ha $val(x) = 0$, akkor $x = x_0$ választással készen vagyunk. Pozitív folyamnagyság esetén van olyan P s-t út, melynek minden élén a folyam értéke pozitív, hiszen bármely S vágásra a $d_x^+(S) - d_x^-(S) = val(x)$ nettó kiáramlás pozitív, tehát kell legyen S-ből kilépő pozitív él. Legyen $\alpha_1 = \min\{x(e) : e \in P\}$ és legyen x_1 egy α_1 nagyságú útfolyam. Ekkor $x' = x - x_1$ is folyam, mégpedig a $D' = (V, A - e)$ hálózatban, ahol e egy olyan él, amelyen α_1 felvétetik. Az indukciós feltevést használva x'-nek van jó felbontása. Ehhez az x_1 folyamot hozzávéve az x kívánt felbontását kapjuk. •

Azt is igazoltuk, hogy olyan felbontás is van, amelyre $t \leq |A|$.

Az MFMC tételből könnyen levezethető Menger klasszikus tétele. Ez a tétel, melyet négy változatban is megfogalmazunk, a gráfok magasabb összefüggőségére vonatkozó eredmények nagy részében alapvető szerepet játszik.

Egy F élhalmaz **lefog** minden s-t utat egy digráfban, ha bármely P s-t útnak legalább egy éle F-ben van. Ezzel ekvivalens, hogy az F élhalmaz kitörlése után nincs út s-ből t-be.

10.1.2. Tétel. *Legyen a $D = (V, A)$ digráfban s, t egy kijelölt pontpár. Az éldiszjunkt s-t utak maximális száma egyenlő az s-t utakat lefogó élek minimális számával.*

Bizonyítás: A max $\leq$ min irány abból következik, hogy egy lefogó élhalmazon minden s-t út áthalad.

Legyen k a lefogó élek minimális száma. Tekintsük a D, s, t, g hálózatot, ahol $g \equiv 1$. Ebben a minimális vágás kapacitása legalább k, hiszen minden vágás kilépő élei lefogó élhalmazt alkotnak. Az MFMC tétel miatt van (legalább) k nagyságú x folyam. A Lemma miatt x felírható $x = x_1 + x_2 + \ldots x_t + x_0$ alakban, ahol x_i egész útfolyam. A kapacitások választása miatt minden útfolyam nagysága 1, az utak éldiszjunktak, és $t = val(x) \geq k$. Azaz van D-ben (legalább) k éldiszjunkt s-t út. •

10.1.3. Tétel. *Legyen a $D = (V, A)$ digráfban s, t egy kijelölt pontpár, $st \notin A$. A pontdiszjunkt s-t utak maximális száma egyenlő az s-t utakat lefogó $V - \{s, t\}$-beli ponthalmazok minimális méretével.*

Bizonyítás: A tétel a pontszéthúzási módszerrel igazolható: minden $v \notin \{s, t\}$ pontot helyettesítsünk egy v', v'' pontpárral, valamint egy $v'v''$ éllel. Ezután a v-be lépő élek lépjenek v'-be, a v-ből kilépők induljanak v''-ből. Az így módosított digráfra alkalmazzuk az előző tételt. •

A Menger-tétel irányítatlan gráfokra is érvényes, az alábbi formában.

10.1.4. Tétel. *Legyen a $G = (V, E)$ gráfban s, t egy kijelölt pontpár. Az éldiszjunkt s-t utak maximális száma egyenlő az s-t utakat lefogó élek minimális számával.*

Bizonyítás: A gráf minden uv élét helyettesítsük egy uv, vu irányított élpárral, majd alkalmazzunk a digráfokra vonatkozó tételt. •

A következő változat is a fenti módszerekkel igazolható.

10.1.5. Tétel. *Legyen a $G = (V, E)$ gráfban s, t egy kijelölt pontpár, $st \notin E$. A pontdiszjunkt s-t utak maximális száma egyenlő az s-t utakat lefogó $V - \{s, t\}$-beli ponthalmazok minimális méretével.*

Egy G gráfban (vagy D digráfban) az s és t pontok között vezető éldiszjunkt utak maximális száma a **lokális élösszefüggőség**, jele $\lambda_G(u, v)$. A fenti megfontolások alapján ez a szám egy alkalmas hálózatban egy maximális folyam kiszámolásával megkapható.

A G gráf k-élösszefüggő, ha minden $F \subseteq E$, $|F| \leq k - 1$ esetén $G - F$ összefüggő. A G élösszefüggőségi száma, $\lambda(G)$, a legnagyobb olyan k, amelyre G k-élösszefüggő. Azaz a minimális szeparáló élhalmaz mérete.

Az alábbi három lemma mindegyike könnyen megkapható Menger megfelelő tételeit használva.

10.1.6. Lemma. $\lambda(G) = \min_{u,v \in V} \lambda_G(u, v) = \min_{\emptyset \subsetneq X \subsetneq V} d(X)$ •

Digráfokra a definíció hasonló.

10.1.7. Lemma. $\lambda(D) = \min_{u,v \in V} \lambda_D(u, v) = \min_{\emptyset \subsetneq X \subsetneq V} d^-(X)$ •

A G gráf k-pontösszefüggő, ha $|V(G)| \geq k + 1$ és minden $S \subseteq V$, $|S| \leq k - 1$ esetén $G - S$ összefüggő. A G pontösszefüggőségi száma a legnagyobb k, amelyre G k-pontösszefüggő. Jele $\kappa(G)$.

10.1.8. Lemma. *Legyenek u, v, w a G gráf különböző pontjai. Ekkor*

$$\lambda_G(u, v) \geq \min\{\lambda_G(u, w), \lambda_G(w, v)\}.$$

•

10.2. Folyamekvivalens fák

Ebben a szakaszban a lokális élösszefüggőség általánosabb, kapacitásos változatával foglalkozunk. A $G = (V, E)$ gráf és $c : E \to \mathbb{R}_+$ élkapacitások esetén jelölje $\lambda_G^c(u, v)$ a gráfban a maximális folyam nagyságát u-ból v-be a c kapacitások mellett. Ahogy láttuk, ez nem más, mint a G-ben az u és v pontokat szeparáló vágások kapacitásainak minimuma. Ha $c \equiv 1$, akkor visszakapjuk a korábban már bevezetett $\lambda_G(u, v)$ értéket.

Legyen $G = (V, E)$ egy $c : E \to \mathbb{R}_+$ élkapacitásokkal ellátott gráf. Azt mondjuk, hogy a $T = (V, F)$ fa a $c' : F \to \mathbb{R}_+$ élkapacitásokkal **folyam-ekvivalens fa** (a G, c párhoz), ha minden $u, v \in V$ párra

$$\lambda_T^{c'}(u, v) = \lambda_G^c(u, v).$$

Egyszerű megfigyelés, hogy $\lambda_T^{c'}(u, v)$ nem más, mint a T-beli egyértelmű uv-út mentén a c' kapacitások minimuma.

Tekintsük például a $K_{3,3}$ teljes páros gráfot az egységnyi $c \equiv 1$ kapacitásokkal. Mivel minden pontpárra a maximális folyam értéke ebben a gráfban pontosan 3, ezért bármely fa a V alaphalmazon a $c' \equiv 3$ kapacitásokkal folyam-ekvivalens fa lesz a $K_{3,3}, c$ párhoz.

10.2.1. Tétel. *Minden G, c-hez létezik folyam-ekvivalens fa.*

Bizonyítás: Tekintsük a V alaphalmazon vett teljes gráfot a $c'(u, v) = \lambda_G^c(u, v)$ élsúlyokkal. Legyen T ebben a gráfban maximális súlyú feszítőfa. Azt állítjuk, hogy ez a T fa a c' kapacitásokkal folyam-ekvivalens G, c-hez.

Ennek igazolásához legyen $s, t \in V$ egy tetszőleges pontpár, és legyen a T-beli P st-út mentén a minimális c' érték m. Mivel T maximális súlyú c'-re, $\lambda_G^c(s, t) = c'(s, t) \leq m$ (ellenkező esetben a T-hez az st élt hozzávéve és a P út mentén egy m súlyú élt kidobva nagyobb súlyú feszítőfát kapnánk). Legyen X egy minimális st-vágás G-ben. Ekkor a P valamely xy éle kilép X-ből, így annak c' értékére $m \leq c'(xy) = \lambda_G^c(x, y) \leq \lambda_G^c(s, t)$ teljesül. A fentiek alapján $m = \lambda_G^c(s, t)$, ahogy állítottuk. •

Következménye a 10.2.1 Tételnek, hogy egy gráfban legfeljebb $n - 1$ különböző folyamérték lehet. (Mutassunk olyat, ahol pont ennyi van.)

10.3. Gráfok fülfelbontásai

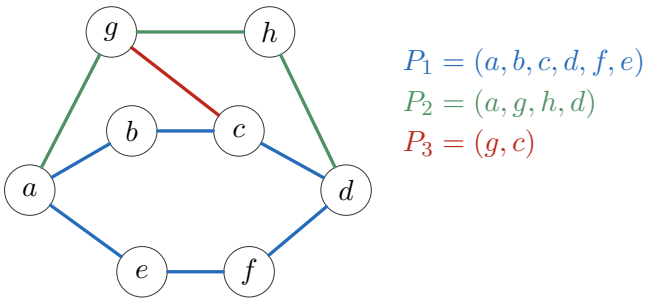
A $G = (V, E)$ gráf **fülfelbontása** G-beli utak egy olyan $(P_1, P_2, \ldots, P_t)$ sorozata, amelyben az első út zárt (azaz P_1 a gráf egy köre), és amely teljesíti az alábbiakat:

(i) $E(P_i) \cap E(P_j) = \emptyset$ minden $1 \leq i < j \leq t$-re, azaz utak éldiszjunktak,

(ii) $\bigcup_{i=1}^t E(P_i) = E$, azaz az utak lefedik a G élhalmazát,

(iii) $V(P_j)$ és $\cup_{i=1}^{j-1} V(P_i)$ metszete pontosan a P_j két (esetleg egybeeső) végpontja.

A fülfelbontásban szereplő utakat **füleknek** nevezzük. A kétpontú fülek elnevezése **triviális fül**, a P_1 a **kezdőkör**. Ha minden fül, kivéve az elsőt, nyílt (azaz a végpontjai különbözők), akkor a fülfelbontás **nyílt**.



10.1. ábra. Az ábrán látható gráf egy lehetséges fülfelbontása: $P_1 = (a, b, c, d, f, e)$, $P_2 = (a, g, h, d)$, $P_3 = (g, c)$.

Ez az egyszerű fogalom számos bizonyításban és algoritmusban fontos szerepet kap. Első kérdés az, hogy mely G gráfoknak van egyáltalán fülfelbontása. Az rögtön látszik, hogy G összefüggő kell legyen, és tartalmaznia kell legalább egy kört. Ennél többet is mondhatunk.

10.3.1. Tétel. *A $G = (V, E)$ gráfnak pontosan akkor van fülfelbontása, ha 2-élösszefüggő.*

Bizonyítás: Egy fülfelbontásban a kezdőkör 2-élösszefüggő, és minden további fül hozzávétele megőrzi az összefüggőséget és azt a tulajdonságot, hogy minden él rajta van körön. Tehát ha G-nek van fülfelbontása, akkor 2-élösszefüggő.

Fordítva, tegyük fel, hogy G 2-élösszefüggő. Ekkor, mivel G minden pontjának foka legalább kettő, van G-ben egy C kör. Tekintsünk egy olyan maximális H részgráfot G-ben, amelynek van a $P_1 = C$ kezdőkörrel induló fülfelbontása. Ilyen részgráf van, pl. C ilyen. Ha $H = G$, készen vagyunk. Ha nem ez a helyzet, de $V(H) = V$, akkor az $E - E(H)$-beli éleket triviális fülekként hozzá tudjuk venni a H fülfelbontásához, ami ellentmond H maximalitásának. A fennmaradó eset az, amelyben $V - V(H)$ nemüres. A G összefüggő, ezért van él $V(H)$ és $V - V(H)$ között, legyen egy ilyen él uv, ahol $u \in V(H)$. A 2-élösszefüggőség miatt $G - uv$-ben van egy P út v-ből $V(H)$-ba. Ekkor a H fülfelbontásához a $P + uv$ utat hozzávéve egy fülfelbontást kapunk, ami ismét ellentmond a H maximális választásának. •

Figyeljük meg, hogy ha van fülfelbontás, akkor annak kezdőkörét tetszőlegesen választhatjuk.

A következő tétel bizonyítása nagyon hasonló.

10.3.2. Tétel. *A $G = (V, E)$ gráfnak pontosan akkor van nyílt fülfelbontása, ha 2-pontösszefüggő.*

Egy gráfnak számtalan különböző fülfelbontása lehet. Hasznos megfigyelés, hogy a fülek száma mindig ugyanannyi.

10.3.3. Tétel. *Legyen G 2-élösszefüggő gráf. Ekkor G minden $(P_1, P_2, ..., P_t)$ fülfelbontására*

$$t = |E| - |V| + 1.$$

A fülfelbontások több látványos alkalmazását a fejezet végén szereplő feladatokban mutatjuk be. Itt egy elméleti eredményre adott rövid bizonyítással illusztráljuk a módszert.

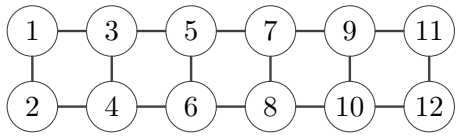
A G 2-pontösszefüggő gráfra akkor mondjuk, hogy **kritikusan 2-pontösszefüggő**, ha bármely pontját elhagyva már nem az. Ilyen például egy kör, vagy egy olyan gráf, amit egy 2-pontösszefüggő gráfból kapunk úgy, hogy minden élét felosztjuk két ponttal.

10.3.4. Tétel. *Minden kritikusan 2-pontösszefüggő G gráfban van másodfokú pont. Ha G-nek legalább négy pontja van, akkor a másodfokú pontjainak száma legalább négy.*

Bizonyítás: Tekintsük a G egy olyan $(P_1, P_2, ..., P_t)$ nyílt fülfelbontását, amelyben az utolsó nem-triviális fül, mondjuk P_j, a lehető legrövidebb. Ha $j = 1$, akkor a gráf a P_1 kezdőkörből áll és annak néhány átlójából, amelyek triviális fülek. Ha egy átló sincs, akkor G maga egy kör, amelyre a tétel állítása nyilvánvaló. Ha van, akkor a P_1 kört az egyik átló mentén kettévágva tudnánk olyan fülfelbontást találni, amelyben az utolsó nem-triviális fül rövidebb, mint P_1. Ez ellentmondás.

Feltehetjük tehát, hogy $P_j = (v_1, v_2, ..., v_r)$ egy olyan út, amelyre $v_1 \neq v_r$ és amelynek legalább egy belső pontja van. Tegyük fel, hogy pontosan egy belső pontja van, a v_2. Ekkor $G - v_2$ 2-pontösszefüggő, hiszen ennek a gráfnak a P_j választása miatt a $(P_1, P_2, ..., P_{j-1})$ egy fülfelbontása. Ez azonban ellentmond annak, hogy a G kritikusan 2-pontösszefüggő.

Ez alapján legalább két belső pontja van. Azt állítjuk, hogy ezekre nem illeszkedik más él, csak a P_j-beliek. Ha nem ez lenne a helyzet, akkor egy ilyen él segítségével a P_j fület helyettesíthetnénk két rövidebb nem-triviális füllel, ellentmondva a fülfelbontás választásának. Mindezek alapján a P_j belső pontjai másodfokú pontok G-ben. További két másodfokú pontot kapunk, ha a gondolatmenetet egy olyan fülfelbontásra ismételjük meg, amelynek kezdőköre tartalmazza a P_j utat. •



10.2. ábra. Az ábrán látható kritikusan 2-pontösszefüggő gráf mutatja, hogy a tételben szereplő alsó korlát a másodfokú pontok számára éles. Hasonló példa (n pontú "létra gráf") minden $n \geq 4$ páros számra létezik.

10.4. Feladatok

10.4.1. Feladat. *Adjuk meg az alábbi gráfok él-, illetve pontösszefüggőségi számát: K_n, $K_{n,m}$, C_n.*

10.4.2. Feladat. *Legfeljebb mekkora lehet két fa uniójának pontösszefüggősége?*

10.4.3. Feladat. *Igazoljuk, hogy ha egy n pontú egyszerű G gráfban minden fokszám legalább $(n + k - 2)/2$, akkor G k-összefüggő!*

10.4.4. Feladat. *Mutassuk meg, hogy egy háromszorosan összefüggő gráfban létezik páros hosszú kör!*

10.4.5. Feladat. *Legyen G egy 2023 pontú, 32-összefüggő gráf. Igazoljuk, hogy G bármely pontpárja közt a legrövidebb út hossza legfeljebb 64.*

10.4.6. Feladat. *Legyen $r(u,v)$ szimmetrikus, nemnegatív, egészértékű függvény a V alaphalmaz elempárjain. Igazoljuk, hogy akkor és csak akkor létezik olyan $G=(V,E)$ gráf, amelyre $\lambda(u,v;G)=r(u,v)$ minden $\{u,v\}$ pontpárra, ha*

$$r(u_1,u_k)\geq \min\{r(u_1,u_2),r(u_2,u_3),\ldots,r(u_{k-1},u_k)\}$$

fennáll minden, a V különböző elemeiből készített $u_1,u_2,\ldots,u_k$ sorozatra. (Útmutatás: legyen F egy maximális súlyú feszítőfa a V ponthalmazú teljes gráfban az r súlyozásra nézve. Helyettesítsük ennek mindegyik uv élét $r(u,v)$ párhuzamos éllel. Ez jó lesz G-nek.)

10.4.7. Feladat. *Egy $G=(V,E)$ irányítatlan gráfban minden fokszám páros. Lássuk be, hogy ekkor bármely különböző u,v pontokra a $\lambda(u,v)$ lokális élösszefüggőségi szám páros!*

10.4.8. Feladat. *Tegyük fel, hogy G egy k-élösszefüggő gráf. Válasszuk ki k (nem feltétlenül független) élét, és csípjük őket össze egy új csúccsal (lásd ábra). Mutassuk meg, hogy az így keletkezett (nem feltétlenül egyszerű) gráf is k-élösszefüggő.*



10.4.9. Feladat. *Határozzuk meg, hogy maximálisan hány éle lehet egy 6 csúcsú egyszerű gráfnak, ha élösszefüggőségi száma 2.*

10.4.10. Feladat. *Fülfelbontások segítségével adjunk új bizonyítást arra, hogy egy G irányítatlan gráfnak pontosan akkor van erősen összefüggő irányítása, ha G kétszeresen élösszefüggő.*

10.4.11. Feladat. *Legyen $G=(V,E)$ egy olyan 2-élösszefüggő irányítatlan gráf, amelyből tetszőleges élet törölve a kapott gráf már nem 2-élösszefüggő. Bizonyítsuk be, hogy G-nek van másodfokú pontja (sőt, legalább kettő ilyen pontja van, ha G legalább kétpontú)!*

10.4.12. Feladat. *Mutassuk meg, hogy egy összefüggő gráf pontosan akkor 2-élösszefüggő (illetve 2-pontösszefüggő), ha bármely éle (bármely két pontja) rajta van egy körön.*

10.4.13. Feladat. *Legyen e a legalább négypontú G 2-pontösszefüggő gráf egy éle. Mutassuk meg, hogy a $G-e$ és G/e gráfok közül legalább az egyik 2-pontösszefüggő.*

Legyen a $G=(V,E)$ gráfban s,t két kitüntetett pont. Ekkor a pontjainak $v_1,v_2,...,v_n$ sorrendje **s-t-számozás**, ha

(i) $v_1=s$, $v_n=t$;

(ii) minden v_i-nek ($i\notin\{1,n\}$) van olyan v_j szomszédja, amelyre $i<j$, és olyan v_k is, amelyre $i>k$.

10.4.14. Feladat. *Mutassuk meg, hogy minden 2-pontösszefüggő gráfnak van s-t számozása, bármely s,t pontpárra. (Igaz az állítás megfordítása is?)*

Legyen a $G=(V,E)$ gráfban $r\in V$ egy kijelölt gyökérpont. A G T_1,T_2 feszítőfái **függetlenek** (**élfüggetlenek**), ha minden v-re a két fában levő két v-r út belsőleg pont-diszjunkt (éldiszjunkt).

10.4.15. Feladat. *Mutassuk meg, hogy G pontosan akkor 2-pontösszefüggő(illetve 2-élösszefüggő), ha bármely gyökérre van két független (élfüggetlen) feszítőfája.*

10.4.16. Feladat. *Legyen $T \subseteq V$ egy páros ponthalmaz $G = (V, E)$ 2-pontösszefüggő gráfban. Igazoljuk, hogy van a pontoknak olyan $\{V_1, V_2\}$ kettéosztása, amelyre $G[V_1]$ és $G[V_2]$ is összefüggő, és $|V_1 \cap T| = |V_2 \cap T|$.*

10.4.17. Feladat. *Igazoljuk, hogy egy $n \geq 4$ pontú minimálisan 2-pontösszefüggő gráf élszáma legfeljebb $2(n-2)$, és pontosan akkor áll egyenlőség, ha $G = K_{2,n-2}$.*

10.4.18. Feladat. *Mutassuk meg, hogy egy G n-pontú 2-pontösszefüggő gráfban legfeljebb $\frac{n(n-3)}{2}$ elvágó pontpár van. Jellemezzük azokat a gráfokat, amelyekre egyenlőség teljesül.*

11. fejezet

Poliéderes kombinatorika

11.1. Az LP feladat

Ebben a fejezetben központi szerepet kapnak az

$$f(x_1, x_2, ..., x_n) = c_1x_1 + c_2x_2 + \dots c_nx_n$$

alakú **lineáris függvények**, melyekben minden x_j változó ($1 \leq j \leq n$) egy c_j valós együtthatóval szerepel. Egy ilyen függvényt gyakran a $\sum_{j=1}^{n} c_jx_j$ tömör alakban adunk meg. A **lineáris egyenlet** $f(x_1, x_2, ..., x_n) = b$ alakú, ahol f egy lineáris függvény, b egy valós szám. **Lineáris egyenlőtlenséget** kapunk, ha $=$ b helyett $\leq$ b vagy $\geq$ b szerepel. A lineáris egyenleteket és lineáris egyenlőtlenségeket együttesen **lineáris feltételeknek** nevezzük.

Egy **Lineáris Programozási**, vagy röviden **LP** feladatban véges sok lineáris feltétel adott n változóra (x_j, $1 \leq j \leq n$) vonatkozóan, valamint egy lineáris függvény, melyet **célfüggvénynek** nevezünk. A cél egy olyan megoldást (avagy behelyettesítést, értékadást) találni, amely minden feltételt teljesít (ekkor **megengedett megoldásnak** mondjuk), és erre nézve a célfüggvény értékét maximalizálja. Egy ilyen megoldást az LP feladat **optimális megoldásának** nevezzük.

Egy tipikus LP feladat (melyben n változó és m lineáris feltétel szerepel) tömör alakja:

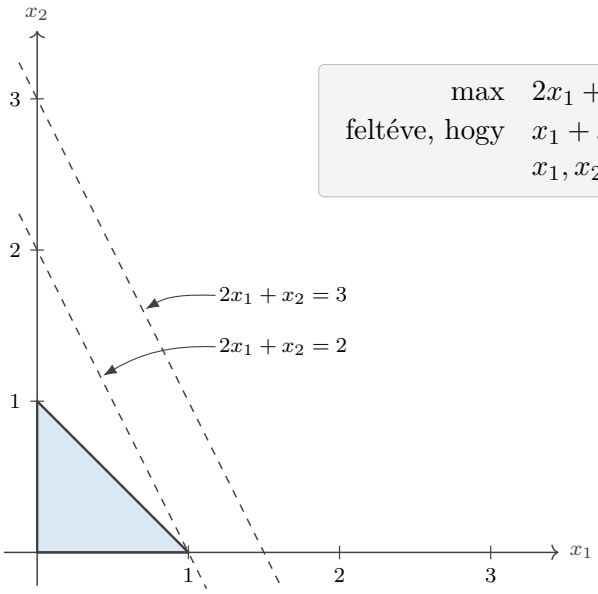
$$\max \sum_{j=1}^{n} c_jx_j$$

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i \quad (i = 1, ..., m)$$

Az i-edik feltételben $a_{ij} \in \mathbb{R}$ az x_j változó együtthatója. Gyakran hasznos az

$$x_j \geq 0 \quad (j = 1, ..., n)$$

alakú lineáris feltételeket különválasztani, ha vannak ilyenek. Ezek a **nem-negativitási feltételek**. Egy ilyen alakú LP feladatra (maximalizálás, lineáris egyenlőtlenségek, nem-negativitási feltételek) azt mondjuk, hogy **standard alakban** van. Általában egy LP feladatot könnyen át lehet alakítani egy vele ekvivalens standard alakú LP feladattá. Például, ha egy lineáris függvény minimalizálása a cél, akkor maximalizálhatjuk a -1-szeresét. Egy egyenlőség helyettesíthető két egyenlőtlenséggel, és így tovább. A példáink nagy része standard alakú lesz, kivéve, hogy majdnem mindig minimalizálási feladatokat tekintünk majd.



11.1. ábra. Az első példában szereplő LP feladat. A megengedett megoldások halmaza (szürke háromszög), valamint az $(1, 0)$ pont optimalitását illusztráló egyenesek (hipersíkok).

Hasznos az LP feladatok geometriai megközelítése. Vizsgáljuk meg, milyen objektumok bukkannak fel egy LP feladatban. Tekintsük az $\mathbb{R}^n$ n-dimenziós valós teret, ahol n a változók száma. Minden változónak feleltessük meg az egyik koordinátatengelyt. Ekkor $\mathbb{R}^n$ minden $p = (p_1, p_2, ..., p_n)$ pontja (amely a koordinátáiból álló rendezett valós szám n-es) megfelel egy értékadásnak: $x_j = p_j$, $1 \leq j \leq n$. A p pont **megengedett**, ha a neki megfelelő értékadás minden lineáris feltételt kielégít. A p pontban a célfüggvény **értéke** a $c_1p_1 + c_2p_2... + c_np_n$ szám. Célunk tehát egy olyan $p \in \mathbb{R}^n$ megengedett pontot találni, amelyre a célfüggvény értéke maximális.

Azok a pontok, amelyek egy adott lineáris egyenlőséget kielégítenek, egy **hipersíkot** alkotnak $\mathbb{R}^n$-ben. Egy lineáris egyenlőtlenséget kielégítő pontok a megfelelő hipersík egyik oldalán helyezkednek el, azaz egy **félteret** alkotnak. A megengedett megoldások tehát véges sok féltér metszetében helyezkednek el. Jegyezzük meg, hogy egy hipersík maga is két féltér metszete. Egy olyan ponthalmazt, amely véges sok féltér metszete, **poliédernek** nevezünk. Mivel minden féltér **konvex**, ezért minden poliéder is az.

A megengedett megoldások halmaza lehet üres. Ekkor az LP feladatnak nincs megoldása. Ha nem üres, akkor vagy **korlátos** (azaz benne van egy kellően nagy $\mathbb{R}^n$-beli gömbben), vagy **nem korlátos**. Ha korlátos, akkor – nem nehéz megmutatni, hogy – létezik legalább egy optimális megoldás. Ha nem korlátos, akkor lehetnek optimális megoldások, de az is előfordulhat, hogy a megengedett megoldások halmazán a célfüggvény értéke tetszőlegesen nagy lehet. Ebben az esetben nincs optimális megoldás. A P poliéder egy pontja a P **csúcsa**, ha nem írható fel két másik P-beli pont **konvex kombinációjaként**.

Példa Egy LP feladat, melyet a fenti ábrán szemléltetünk:

$$\max \ 2x_1 + x_2$$

$$x_1 + x_2 \leq 1$$

$$x_1, x_2 \geq 0$$

Példa Készítsünk hasonló ábrát és oldjuk meg az alábbi LP feladatot:

$$\max \ x_1 + x_2$$

$$x_1 + 2x_2 \leq 5$$

$$2x_1 + x_2 \leq 4$$

$$x_1, x_2 \geq 0$$

A következő geometriai alaptételeket itt nem bizonyítjuk:

11.1.1. Tétel. *Legyen P korlátos poliéder $\mathbb{R}^n$-ben. Ekkor*
(a) P-nek véges sok csúcsa van, valamint
(b) P a csúcsainak konvex burka.

Véges sok $\mathbb{R}^n$-beli pont konvex burkát **politópnak** nevezzük. Az előző tétel alapján minden korlátos poliéder egy politóp. Ennek a megfordítása is igaz:

11.1.2. Tétel. *Minden politóp korlátos poliéder.*

Az LP feladatokkal kapcsolatban a következő tulajdonság alapvető.

11.1.3. Tétel. *Ha egy LP feladat megengedett megoldásainak halmaza egy P korlátos poliéder, akkor van olyan optimális megoldás, amely a P egy csúcsának felel meg.*

Algoritmikus szempontból kiemelten fontos eredmény, hogy az LP feladatok megoldhatók polinomidőben. Jegyezzük meg, hogy egy LP feladat inputjának mérete arányos az n, m számokkal (azaz a változók és a feltételek számával), valamint az ezekben és a célfüggvényben szereplő összes a_{ij}, c_j, és b_i szám logaritmusainak összegével.

11.2. Diszkrét optimalizálás és LP

Tekintsük a következő, meglehetősen általános optimalizálási feladatot: legyen S egy véges halmaz, $c : S \to \mathbb{R}$ egy valós értékű súlyfüggvény az S elemein, valamint legyen $\mathcal{F}$ az S részhalmazainak egy nemüres családja. Keressünk egy olyan $X \in \mathcal{F}$ halmazt, melyre az általa tartalmazott elemek súlyainak összege maximális.

Például lehet S egy G gráf élhalmaza, az $\mathcal{F}$ pedig tartalmazhatja a G-beli párosítások élhalmazait. Ekkor a jól ismert maximális súlyú párosítás feladatot kapjuk. Ha G összefüggő és $\mathcal{F}$ a G feszítőfáinak élhalmazaiból áll, akkor egy maximális súlyú feszítőfa keresése a cél.

Megmutatjuk, hogy minden ilyen típusú problémára LP feladatként is tekinthetünk. Ennek igazolásához legyen $m = |S|$, és rögzítsünk egy bijekciót az m koordinátatengely és az S elemei között. Ekkor az S minden X részhalmazához definiálhatjuk a χ^X **karakterisztikus vektorát**: ebben a vektorban (avagy szám m-esben), a j-edik koordináta 1, ha $j \in X$, és 0 különben. Ez tehát egy olyan $\mathbb{R}^n$-beli pont, melynek minden koordinátája 0 vagy 1. Az $\mathcal{F}$ halmazhoz tartozó

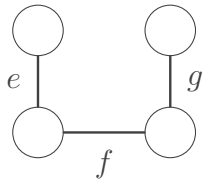
karakterisztikus vektorok tehát véges sok $\mathbb{R}^n$-beli pontot határoznak meg. Vegyük észre, hogy egy X részhalmazban az elemek súlyainak összege éppen $f(\chi^X)$, ahol

$$f(x_1, x_2, ..., x_m) = c_1x_1 + c_2x_2 + \dots c_mx_m$$

a c súlyok által meghatározott lineáris függvény. Azaz a célunk az $f(\chi^X)$ maximalizálása az $\mathcal{F}$-beli halmazok karakterisztikus vektorain.

Jelölje $conv(\mathcal{F})$ az $\mathcal{F}$-ben levő halmazok karakterisztikus vektorainak **konvex burkát**. Definíció szerint $conv(\mathcal{F})$ egy politóp, így a fenti tételek alapján egy korlátos poliéder. Ebből adódóan f maximalizálása a $conv(\mathcal{F})$ halmazon egy LP feladat. Ugyanakkor azt is tudjuk, hogy az LP feladatnak van olyan optimális megoldása, amely a poliéder csúcsa. Egy ilyen megoldás pedig szükségképpen valamely $X \in \mathcal{F}$ halmaz karakterisztikus vektorának felel meg, azaz optimális megoldás az eredeti feladatunkra. Összefoglalva, a konvex burok feletti optimalizálás (az LP feladat) ekvivalens egy optimális karakterisztikus vektor megtalálásával, és így a maximális súlyú $\mathcal{F}$-beli halmaz megkeresésével.

Ha a fenti gondolatok mentén szeretnénk egy diszkrét optimalizálási feladatot LP feladatként megoldani, akkor meg kell találjunk egy olyan lineáris feltételrendszert, ami éppen a kívánt politópot határozza meg. Ez komoly kihívás, sok esetben nem is ismert ilyen leírás. Kapcsolódó kérdés, hogy hogy van megadva az optimalizálási feladat inputja (az $\mathcal{F}$ elemeinek felsorolásával, vagy esetleg implicit, például egy G gráffal, amely alapján egy halmazról el tudjuk dönteni, hogy $\mathcal{F}$-ben van-e), mekkora az input mérete, továbbá ehhez képest mekkora a feladatot leíró LP mérete, illetve van-e hatékony algoritmusunk az LP megoldására?



M	χ^M
$\emptyset$	$(0,0,0)$
$\{e\}$	$(1,0,0)$
$\{f\}$	$(0,1,0)$
$\{g\}$	$(0,0,1)$
$\{e,g\}$	$(1,0,1)$

11.2. ábra. A háromélű út, az általa tartalmazott párosítások, valamint ezek karakterisztikus vektorai.

11.3. Az IP feladat

Egy **egészértékű programozási**, vagy tömören IP (Integer Programming) feladat egy LP feladatból és véges sok további feltételből áll, mely feltételek azt írják elő, hogy bizonyos változók csak meghatározott egész értékeket vehetnek fel. Ismert, hogy IP feladatok optimális megoldásának keresése NP-nehéz probléma.

Ennek ellenére az IP feladatok fontos szerepet töltenek be a diszkrét optimalizálásban. A következő példák mutatják, hogy ilyen problémák esetén gyakran kézenfekvő egy ekvivalens ún. IP felírás. Az ezekben szereplő egyszerű egészértékűségi feltételek bevezetése, amelyek előírják, hogy minden változó csak 0 vagy 1 értéket vehet fel, azt eredményezik, hogy a megengedett megoldások karakterisztikus vektorain optimalizálunk.

Példa Az alábbi IP feladat a Hátizsák feladattal ekvivalens: megengedett megoldásai a megengedett pakolásoknak felelnek meg, optimális megoldása az optimális pakolást adja:

$$\max \sum_{i=1}^{k} c_i x_i$$

$$\sum_{i=1}^{k} a_i x_i \leq b$$

$$x_i \in \{0,1\}, \quad 1 \leq i \leq k.$$

Példa A minimális súlyú lefogó ponthalmaz feladat IP felírása az alábbi:

$$\min \sum_{v \in V} c_v x_v$$

$$x_u + x_v \geq 1, \quad uv \in E$$

$$x_v \in \{0,1\}, \quad v \in V$$

Példa Maximális súlyú feszítőfa keresése ekvivalens a következő IP feladattal:

$$\max \sum_{e \in E} c_e x_e$$

$$\sum_{e \in E} x_e = |V| - 1$$

$$\sum_{e \in \delta(S)} x_e \geq 1, \quad \emptyset \subsetneq S \subsetneq V$$

$$x_e \in \{0,1\}, \quad v \in V$$

11.4. Poliéderes kombinatorika

Tekintsünk egy $n \times n$-es M mátrixot, melynek elemei nem-negatív valós számok. Az M **α-mátrix**, ha minden sorában és oszlopában az elemek összege α. Ha az $\alpha = 1$-re teljesül, akkor a **duplán sztochasztikus mátrix** elnevezést is használjuk. Az M **permutáció mátrix**, ha egy olyan duplán sztochasztikus mátrix, melynek minden eleme 0 vagy 1.

11.4.1. Tétel. *Minden α-mátrix felírható permutáció mátrixok nem-negatív lineáris kombinációjaként. Egy ilyen felírásban az együtthatók összege α.*

Bizonyítás: Legyen a mátrix $n \times n$-es. Indukcióval bizonyítunk, egyszerre minden α-ra, a mátrixban szereplő nem-nulla elemek száma szerint. Ha ez a szám nulla, akkor a mátrix a nulla mátrix, a tétel állítása nyilvánvaló.

Tegyük fel, hogy az M α-mátrixban vannak nem-nulla elemek. Készítsük el azt a $G = (V, W; E)$ páros gráfot, melyben az egyik színosztály (pontjai $v_1, v_2, \ldots, v_n$) az M sorainak, a másik (pontjai $w_1, w_2, \ldots, w_n$) az oszlopainak felel meg, és $v_i w_j$ pontosan akkor él, ha $M[i, j] > 0$. Figyeljük meg, hogy minden $X \subseteq V$ ponthalmaznak legalább $|X|$ szomszédja van W-ben: ez abból következik, hogy az X-re illeszkedő éleknek megfelelő elemek összege $\alpha|X|$, mely élek mindegyike az X egy szomszédjára illeszkedik - ugyanakkor a szomszédok mindegyikére ebből legfeljebb α rész juthat.

A Hall-tétel alapján emiatt van G-ben egy S teljes párosítás. Az S élei a mátrix n pozitív elemének felelnek meg, minden sorból és minden oszlopból egynek. Legyen ezen n elem minimuma $\lambda > 0$. Jelölje P azt a permutáció mátrixot, amelyben pontosan ott van 1-es, ahol az S-nek megfelelő elemek vannak. Ekkor $M' = M - \lambda P$ egy $(\alpha - \lambda)$-mátrix, melyben legalább eggyel kevesebb a nem-nulla elem. Az indukciós feltevés alapján

$$M' = \lambda_1 P_1 + \lambda_2 P_2 + \cdots + \lambda_t P_t,$$

ahol a P_i-k permutáció mátrixok, a λ_i-k pedig olyan pozitív számok, melyek összege $(\alpha - \lambda)$. Az M' definíciója alapján a tétel állítása így M-re is teljesül. •

A tétel az $\alpha = 1$ esetben Birkhoff és Neumann alábbi nevezetes eredményét adja:

11.4.2. Tétel. *Minden duplán sztochasztikus mátrix felírható permutáció mátrixok konvex kombinációjaként.*

11.5. A párosítás politóp

A $G = (V, E)$ gráf egy M párosításának χ^M **incidencia vektora** az a 0-1-vektor (avagy $\mathbb{R}^{|E|}$-beli pont), amelyre

$$\chi^M(e) = 1, \text{ ha } e \in M,$$
$$\chi^M(e) = 0, \text{ ha } e \notin M.$$

Itt feltesszük, hogy adott az élek egy $e_1, e_2, ..., e_m$ sorrendje ($m = |E|$), amely sorrend meghatározza az élek és a koordinátatengelyek közötti bijekciót.

A G **párosítás politópja** a következő:

$$P_{ps}(G) = conv\{\chi^M : M \text{ párosítás } G\text{-ben}\}. \tag{11.1}$$

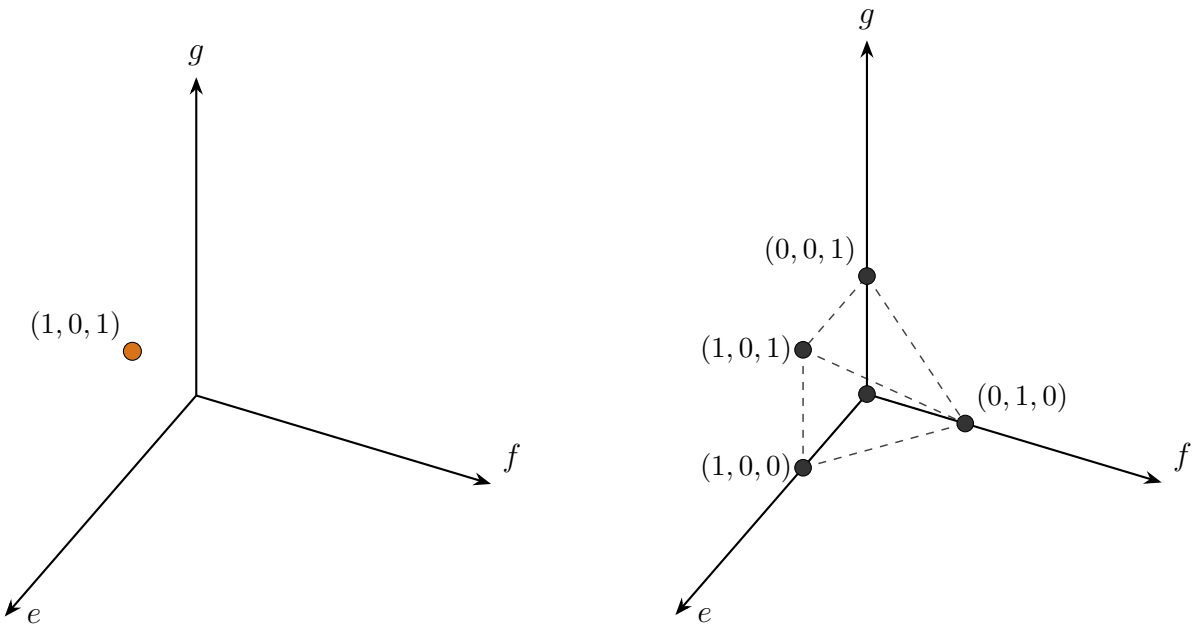
Tehát $P_{ps}(G)$ egy politóp $\mathbb{R}^{|E|}$-ben, amely a fenti tétel alapján egy poliéder, félterek metszete, amelyeket lineáris egyenlőtlenségek definiálnak. Ezen egyenlőtlenségek meghatározása a célunk, legalábbis abban a speciális esetben, amikor G páros gráf. Ehhez érdemes először a G **teljes párosítás politópját** vizsgálni, amely a következő:

$$P_{tps}(G) = conv\{\chi^M : M \text{ teljes párosítás } G\text{-ben}\}. \tag{11.2}$$

Példa Legyen G a három élű út, melynek élei e_1, e_2, e_3. A G párosításainak karakterisztikus vektorai:

$$(0,0,0), (1,0,0), (0,1,0), (0,0,1), (1,0,1).$$

Teljes párosításcsak egy van, ennek az $(1, 0, 1)$ felel meg. Ennek megfelelően $P_{ps}(G)$ egy gúla $\mathbb{R}^3$-ban, öt csúccsal, $P_{tps}(G)$ pedig egyetlen pont.



P_{tps}: egyetlen pont P_{ps}: gúla (5 csúcs)

11.3. ábra. A három élű út teljes párosítás poliédere és párosítás poliédere

Páros gráfok esetén a teljes párosítás politóp leírása a következő.

11.5.1. Tétel. *Legyen $G = (A, B; E)$ páros gráf. Ekkor $P_{tps}(G)$ azon $x \in \mathbb{R}^{|E|}$-beli pontok halmaza, melyekre*

$$\sum_{e:\ e \text{ illeszkedik } v\text{-re}} x_e = 1, \text{ minden } v \in V\text{-re, és,} \tag{11.3}$$

$$x_e \geq 0, \text{minden } e \in E\text{-re.} \tag{11.4}$$

Bizonyítás: Legyen P az a poliéder, amelyet a (11.3) és (11.4) egyenlőtlenségek definiálnak. Mivel minden M teljes párosítás esetén χ^M kielégíti az egyenlőtlenségeket, $P_{tps}(G)$ minden csúcsa benne van P-ben. Mivel P konvex, ebből $P_{tps}(G) \subseteq P$ következik.

A másik irányú tartalmazás igazolásához először figyeljük meg, hogy $P = \emptyset$ esetén P és $P_{tps}(G)$ is üres, így a kívánt egyenlőség teljesül. Tegyük fel, hogy P nem-üres és legyen $x \in P$ egy tetszőleges pontja. Jelölje a G színosztályait A, B, ahol $A = \{a_1, a_2, \ldots, a_n\}$ és $B = \{b_1, b_2, \ldots b_m\}$. Vegyük észre, hogy (11.3) miatt

$$|A| = \sum_{i=1}^{n} \sum_{j:a_ib_j \in E}^{m} x_{a_ib_j} = \sum_{j=1}^{m} \sum_{i:a_ib_j \in E}^{n} x_{a_ib_j} = |B|,$$

ezért $|A| = |B| = n$. Az x egyértelműen megfeleltethető egy n-szer n-es D_x nem-negatív mátrixnak, amelyben

$$D_x[i, j] = x_{a_ib_j}, \text{ ha } a_ib_j \in E,$$

és

$$D_x[i,j] = 0, \text{ ha } a_i b_j \notin E.$$

A D_x mátrix (11.3) és (11.4) miatt duplán sztochasztikus. A Birkhoff–Neumann-tétel alapján ezért felírható

$$D_x = \lambda_1 P_1 + \lambda_2 P_2 + \cdots + \lambda_t P_t,$$

alakban, ahol a P_i permutáció mátrix és $\lambda_i \geq 0$, minden $1 \leq i \leq t$-re, valamint $\sum_{i=1}^{t} \lambda_i = 1$. Figyeljük meg, hogy egy P_i permutáció mátrixot a fenti módon definiáló $\mathbb{R}^{|E|}$-beli pont a G gráf valamely M teljes párosításának karakterisztikus vektora. Így ez a felírás pontosan azt mondja, hogy x felírható G-beli teljes párosítások karakterisztikus vektorainak konvex kombinációjaként, azaz $x \in P_{tps}(G)$. Tehát $P \subseteq P_{tps}(G)$. •

A tétel egyik következménye, hogy a P **egész poliéder**, azaz minden csúcsának minden koordinátája egész szám. További következmény, hogy egy minimális súlyú teljes párosítás megkapható egy LP feladat megoldásaként. (Figyeljük meg, hogy a P-t definiáló egyenlőtlenségek száma polinomiális a G méretében.) Erre a feladatra kombinatorikus algoritmus is ismert, az ún. magyar módszer.

Az előbbi tétel következménye:

11.5.2. Tétel. *Legyen G páros gráf. Ekkor $P_{ps}(G)$ azon $x \in \mathbb{R}^{|E|}$-beli pontok halmaza, melyekre*

$$\sum_{e:\ e \text{ illeszkedik } v\text{-re}} x_e \leq 1, \text{ minden } v \in V\text{-re, és,} \tag{11.5}$$

$$x_e \geq 0, \text{minden } e \in E\text{-re.} \tag{11.6}$$

A (11.5) és (11.6) által definiált P poliéderre $P_{ps}(G) \subseteq P$ akkor is teljesül, ha G nem páros gráf. Előfordulhat azonban, hogy a tartalmazás valódi: legyen például $G = K_3$. Ekkor az $x = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ pont benne van P-ben, de nincs benne a $P_{ps}(G)$ politópban.

11.6. Minimális súlyú lefogó ponthalmaz

Egy $G = (V, E)$ gráfban a $T \subseteq V$ ponthalmaz **lefogó ponthalmaz**, ha G minden élének legalább az egyik vége T-ben van. Adott G gráfban minimális méretű lefogó ponthalmaz keresése jól ismert NP-nehéz feladat. Ebben a fejezetben az ennél általánosabb minimális súlyú lefogó ponthalmaz feladatra mutatunk közelítő algoritmust.

Ennek inputja egy $G = (V, E)$ gráf, továbbá a pontjain egy $c : V \to \mathbb{R}_+$ nem-negatív súlyozás. (Az, hogy c nem-negatív, nem lényeges megszorítás – miért?) A közelítő algoritmusunk a feladat egy egészértékű programozási (IP) feladatként való felírására és annak LP relaxációjára épül.

Tekintsük az alábbi IP feladatot:

$$\min \sum_{v \in V} c_v x_v$$

$$x_u + x_v \geq 1, \quad uv \in E$$

$$x_v \in \{0, 1\}, \quad v \in V.$$

Ennek megengedett megoldásai a lefogó ponthalmazoknak, optimális megoldásai a minimális súlyú lefogó ponthalmazoknak felelnek meg. A feladat LP relaxációja, amelyben az egészértékűségi feltételeket nem-negativitási feltételekkel helyettesítjük, a következő:

$$\min \sum_{v \in V} c_v x_v$$
$$x_u + x_v \geq 1, \quad uv \in E$$
$$x_v \geq 0, \quad v \in V.$$

Egy konkrét inputra a fenti IP és LP feladatok optimumait jelölje IP_{OPT} és LP_{OPT}. Egyszerű, de fontos megfigyelés, hogy

$$\mathrm{LP}_{OPT} \leq \mathrm{IP}_{OPT}.$$

Ez lehet szigorú egyenlőtlenség: legyen például $G = K_3$ és $c \equiv 1$. (Keressünk optimális LP megoldást!)

11.6.1. Feladat. *Legyen $G = K_n$ és $c \equiv 1$. Határozzuk meg az LP_{OPT} és IP_{OPT} értékeket!*

Idézzük fel, hogy egy LP feladat egy csúcsmegoldása (vagy extrém pont megoldása) egy olyan megengedett megoldás, amely nem írható fel két másik megengedett megoldás konvex kombinációjaként. Az LP egy **félegész megoldása** egy olyan megengedett megoldás, melyben minden változó értéke 0, 1, vagy $\frac{1}{2}$.

11.6.2. Lemma. *Tekintsük a $G = (V, E)$ gráfhoz és $c : V \to \mathbb{R}_+$ függvényhez tartozó fenti LP feladatot. Legyen $x = \{x_v : v \in V\}$ az LP egy olyan megengedett megoldása, amely nem félegész. Ekkor x felírható két másik megengedett megoldás konvex kombinációjaként.*

Bizonyítás: Feltehetjük, hogy az x_v érték sehol sem nagyobb, mint 1, hiszen ellenkező esetben az x értékét a x_v-n egy kis ϵ-nal csökkentve (illetve növelve) szintén megengedett megoldásokat kapunk, melyek átlaga (avagy konvex kombinációja $\frac{1}{2}$ együtthatókkal) éppen az x. Ez esetben a lemma állítása rögtön következik.
Tekintsük azon $v \in V$ pontokat, amelyekre x_v nem félegész. Osszuk ezeket két részre: legyen

$$V_- = \left\{v : 0 < x_v < \frac{1}{2}\right\} \quad \text{és} \quad V_+ = \left\{v : \frac{1}{2} < x_v < 1\right\}$$

ez a felosztás. Adott $\varepsilon > 0$ esetén az x megengedett megoldásból kiindulva definiáljuk az y és z függvényeket a gráf ponthalmazán (avagy $\mathbb{R}^{|V|}$-beli pontokat) a következőképp: legyen $y_v = x_v + \epsilon$, ha $v \in V_+$, legyen $y_v = x_v - \epsilon$, ha $v \in V_-$, és legyen $y_v = x_v$ egyébként. Legyen $z_v = x_v - \epsilon$, ha $v \in V_+$, legyen $z_v = x_v + \epsilon$, ha $v \in V_-$ és legyen $z_v = x_v$ egyébként. Feltevésünk szerint $V_+ \cup V_- \neq \emptyset$, emiatt x nem egyezik meg sem y-nal, sem z-vel. Továbbá az x ezek konvex kombinációja, hiszen $x = \frac{1}{2}(y + z)$. Megmutatjuk, hogy kellően kis ϵ-ra y és z is az LP megengedett megoldásának felel meg, amiből a lemma következik.

Világos, hogy kellően kis ϵ esetén y és z minden koordinátája nem-negatív. Nézzük az éleknek megfelelő feltételeket. Tegyük fel, hogy $x_u + x_v > 1$ valamely $uv \in E$ élre. Ha ϵ kellően kicsi, akkor ez a feltétel nem sérül. Végül nézzük azt az esetet, amikor $x_u + x_v = 1$. Ekkor az x_u, x_v-re lényegében három lehetőség van: mindkettő $\frac{1}{2}$, egyik 1 és a másik 0, valamint az egyik V_+-ban van, a másik V_--ban van. Mindegyik esetben, bármely ϵ-ra teljesülni fog, hogy $x_u + x_v = y_u + y_v = z_u + z_v$.
•

A lemma alapján kimondhatjuk:

11.6.3. Tétel. *A fenti LP minden csúcsmegoldása félegész.*

Ez a tulajdonság természetes módon egy 2-közelítő algoritmushoz vezet a minimális súlyú lefogó ponthalmaz feladatra.

(1) Keressünk egy $x = \{x_v : v \in V\}$ optimális csúcsmegoldást az LP feladatra.

(2) Legyen az output a

$$T = \left\{ v \in V : x_v \geq \frac{1}{2} \right\}.$$

Ekkor tehát a T-ben a gráf azon pontjai lesznek, melyek x-értéke $\frac{1}{2}$ vagy 1. A T lefogó ponthalmaz, hiszen minden $uv \in E$ élre vagy u-hoz, vagy v-hez (vagy mindkettőhöz) az x LP megoldás legalább $\frac{1}{2}$-et rendel. Továbbá teljesül, hogy

$$\sum_{v \in T} c_v \leq 2 \sum_{v \in V} c_v x_v = 2LP_{opt} \leq 2IP_{opt},$$

hiszen az x optimális LP megoldás, valamint – ahogy az korábban megfigyeltük – a feladat LP relaxációjának optimuma alsó becslés a minimális súlyú lefogó ponthalmaz súlyára. Ez igazolja, hogy a fenti algoritmus valóban 2-közelítő.

11.7. Feladatok

11.7.1. Feladat. *Limonádét és teát árulunk. Egy liter limonádé elkészítéséhez szükségünk van 8 kiskanál citromlére, 3 kiskanál cukorra és egy citromra. Egy liter tea elkészítéséhez pedig szükségünk van 4 kiskanál citromlére és 6 kiskanál cukorra. Egy liter limonádét 3 forintért, egy liter teát 5 forintért tudunk eladni. Rendelkezésünkre áll 4 citrom, 40 kiskanál citromlé és 42 kiskanál cukor. Hány liter limonádét, illetve teát készítsünk, hogy maximalizáljuk a bevételünket?*
Írjuk fel a feladatot lineáris programozási feladatként, majd grafikus módszerrel oldjuk meg a feladatot!

11.7.2. Feladat. *Adott s, t valós számokra tekintsük a következő LP feladatot:*

$$\max x_1 + x_2 : \qquad sx_1 + tx_2 \leq 1, \quad x_1, x_2 \geq 0.$$

Rajzoljuk le a megengedett megoldások halmazát $s = 1$, $t = 1$, illetve $s = 1$, $t = -1$ esetén. Milyen s és t esetén teljesül, hogy

(a) nincs megengedett megoldás?

(b) a megengedett megoldások halmaza nem korlátos és a célfüggvény sem korlátos?

(c) a megengedett megoldások halmaza nem korlátos, de a célfüggvény korlátos?

(d) a megengedett megoldások halmaza korlátos, de nem üres (és akkor a célfüggvény korlátos)?

11.7.3. Feladat. *Igazoljuk, hogy bármely G gráfra a (11.5) és (11.6) által definiált P poliéderre teljesül, hogy $P_{ps}(G) \subseteq P$.*

11.7.4. Feladat. *Legyen $G = K_3$. Mutassuk meg, hogy az $x = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ pont nincs benne a $P_{ps}(G)$ politópban.*

11.7.5. Feladat. *Adjunk minél jobb felső becslést cx-re az alábbi standard alakú LP-feladatban! Optimális megoldása az $x^* = (\frac{2}{7}; 0; \frac{3}{7}; 0)$ vektor a feladatnak?*

$$\begin{aligned} 2x_1 - x_2 + x_3 - x_4 &= 1 \\ -x_1 - 4x_2 + 3x_3 + x_4 &= 1 \\ x_1, x_2, x_3, x_4 &\geq 0 \\ \max 4x_1 - 20x_2 + 9x_3 - 5x_4 & \end{aligned}$$

11.7.6. Feladat. *Tegyük fel, hogy a*

$$\max \left\{ \sum_{j=1}^{n} c_j x_j : \qquad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \ (i = 1, 2, \ldots, m), \quad x_j \geq 0 \ (j = 1, 2, \ldots, n) \right\}$$

LP-feladat nem korlátos. Következik-e ebből, hogy valamely k-ra a

$$\max \left\{ x_k : \qquad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \ (i = 1, 2, \ldots, m), \quad x_j \geq 0 \ (j = 1, 2, \ldots, n) \right\}$$

LP-feladat nem korlátos?

11.7.7. Feladat. *Adott egy $D = (V, A)$ irányított gráf rögzített s és t csúcsokkal és $g \colon A \to \mathbb{R}_+$ kapacitásfüggvénnyel. Írjuk fel LP-feladatként azt a kérdést, hogy van-e a gráfban 5 nagyságú folyam s-ből t-be!*

11.7.8. Feladat. *Adott egy $G = (V, E)$ irányítatlan gráf. Írjunk fel olyan IP-feladatot, melynek optimuma a G gráf kromatikus számát adja!*

11.7.9. Feladat. *Adott a $G = (V, E)$ gráf. Írjunk fel olyan IP-feladatot, melynek optimuma a G gráf minimális méretű lefogó ponthalmazának méretét adja!*

11.7.10. Feladat. *Írjuk fel egészértékű programozási (IP) feladatként az alábbi feladatot: Adottak $a_1, a_2, \ldots, a_n$ nagyságú súlyok. Ezekből szeretnénk legfeljebb k db-ot kiválasztani úgy, hogy az összsúly minél kevesebb, de legalább m legyen.*

11.7.11. Feladat. *Írjuk fel egészértékű programozási (IP) feladatként az alábbi feladatot! Adott egy $G = (V, E)$ gráf, és egy $c \colon V \to \mathbb{R}_+$ súlyozás. Válasszunk ki egy minimális súlyú ponthalmazt, hogy G minden háromszögének legalább az egyik csúcsa ki legyen választva!*

11.7.12. Feladat. *Adott egy $G = (V, E)$ irányítatlan gráf és egy $c \colon E \to \mathbb{R}_+$ súlyfüggvény. Írjuk fel IP-feladatként a minimális súlyú Hamilton-kör keresését G-ben úgy, hogy*

a) a feltételek száma véges, de tetszőleges;

b) a feltételek száma legyen az input méretében polinomiális.

11.7.13. Feladat. *Adott egy $G = (V, E)$ irányítatlan gráf. Írjunk fel olyan egészértékű programozási (IP) feladatot, melynek optimuma a G gráfban*

(a) a maximális klikk méretét adja!

(b) a minimális lefogó élhalmaz méretét adja!

11.7.14. Feladat. *Adott egy $G = (V, E)$ irányítatlan gráf, $c : E \to \mathbb{R}$ élsúlyozással. Azt mondjuk, hogy egy élhalmaz* 2-párosítás, *ha minden csúcsra legfeljebb két él illeszkedik. Írjunk fel olyan IP feladatot, melynek optimuma a G gráf maximális súlyú 2-párosítását adja!*

11.7.15. Feladat. *Adott egy $G = (V, E)$ irányítatlan gráf. Írjunk fel olyan IP feladatot, melynek optimuma a G gráfban a csúcsok egy olyan maximális méretű U részhalmazát adja, amely előáll két független ponthalmaz uniójaként!*

11.7.16. Feladat. *Adott egy $G = (V, E)$ irányítatlan gráf. Írjunk fel olyan IP feladatot, melynek optimuma a G gráfban egy* maximális vágást *ad, azaz megadja, mennyi $d_G(X, V - X)$ maximuma V részhalmazain!*

11.7.17. Feladat. *Adott egy $G = (V, E)$ irányítatlan gráf. Írjunk fel olyan IP feladatot, melynek optimuma a G gráf kromatikus számát adja!*
(Segítség: vezessük be az x_{ik} bináris változókat ($i, k = 1, \ldots, n$), majd írjunk fel egy olyan IP feladatot, melynek optimumában x_{ik} pontosan akkor 1, ha az i-edik csúcsot a k-adik színnel színezzük.)

11.7.18. Feladat. *Adott egy $D = (V, A)$ irányított gráf, $s, t \in V$, és egy $c : A \to \mathbb{R}_+$ élfüggvény. Írjuk fel IP feladatként a legrövidebb $s - t$ út keresés problémáját!*

12. fejezet

Dinamikus programozás

A dinamikus programozás egy algoritmikus módszer optimalizálási problémák megoldására. Az alapgondolata az, hogy a megoldandó feladat optimumát, vagy optimális megoldását úgy határozza meg, hogy alkalmasan választott részfeladatok egy sorozatát oldja meg, ezek optimumait a már megoldottak optimumaiból kiszámolva, egy megfelelő rekurzív eljárással.

Nem mindenhol alkalmazható ez a módszer. Gyakran annak a felismerése sem egyszerű, hogy dinamikus programozási megközelítést érdemes használni. Példákon mutatjuk be, hogyan működik.

12.1. A részösszeg probléma

Adott n nem-negatív egész szám, $a_1, a_2, \ldots, a_n$, valamint egy további b egész. Van-e az adott számoknak olyan részhalmaza, amelyre a számok összege pontosan b? Ez a részösszeg probléma, amely az ismert NP-teljes problémák egyike.

A dinamikus programozási módszer ennek ellenére jól használható, és hatékony ún. pszeudopolinomiális algoritmust ad. Ehhez vezessük be az $f(i, b')$ értékeket minden $0 \leq i \leq n$ és $0 \leq b' \leq b$ egészre. Legyen $f(i, b') = 1$, ha a részösszeg feladatnak van megoldása az $a_1, a_2, \ldots, a_i, b'$ inputra, különben pedig legyen $f(i, b') = 0$. A feladatunk ekvivalens az $f(n, b)$ érték meghatározásával.

Az algoritmusunk alapgondolata az, hogy minden $f(i, b')$ értéket kiszámolunk, rekurzívan, a már kiszámoltakból következtetve a még ismeretlenekre, míg eljutunk a keresett $f(n, b)$-ig. Kezdetben csak a kézenfekvő extrém eseteket ismerjük: a definícióból világos, hogy $f(i, 0) = 1$ minden $0 \leq i \leq n$-re (hiszen az üres részhalmazra az összeg 0), valamint $f(0, b') = 0$ minden $b' > 0$-ra. A döntő megfigyelés az, hogy

$$f(i, b') = 1 \text{ pontosan akkor, ha vagy } f(i-1, b') = 1, \text{ vagy } f(i-1, b'-a_i) = 1.$$

A rekurziót az a megfigyelés igazolja, hogy ha van megoldás az (i, b') párra, akkor ebben vagy nincs benne az a_i (és akkor az $(i-1, b')$-re is van), vagy benne van (és akkor van megoldás az $(i-1, b'-a_i)$ párra). Ez alapján egyre növelve az i értékeket, majd minden új i-re növelve a b' értékeit ki tudjuk számolni az összes $f(i, b')$-t, $O(nb)$ lépésben.

Figyeljük meg, hogy a futási idő nem polinomiális az input méretéhez képest. Ugyanakkor korlátos, vagy kellően kis a_i értékekre gyors algoritmust kapunk.

12.2. A hátizsák feladat

A hátizsák feladat egy optimalizálási probléma, a részösszeg feladat általánosítása. Itt adott $2n$ nem-negatív egész szám, $a_1, a_2, \ldots, a_n$, $c_1, c_2, \ldots, c_n$, valamint egy további b nem-negatív egész. Minden i-re a c_i a megfelelő a_i értéke. Célunk az a_i-k közül kiválasztani néhányat úgy, hogy az összegük legfeljebb b legyen, és erre nézve az értékeik összege legyen a lehető legnagyobb. Azaz olyan $I \subseteq \{1, 2, ..., n\}$ a cél, amelyre

$$\sum_{i \in I} a_i \leq b, \text{ és}$$

$$\sum_{i \in I} c_i$$

maximális. Az algoritmusunk alapgondolata hasonló az előzőhöz. Vezessük be az $f(i, b')$ értékeket minden $0 \leq i \leq n$ és $b' \leq b$ egészre. Legyen a $f(i, b')$ a feladat optimuma az $a_1, a_2, \ldots, a_i$, $c_1, c_2, \ldots, c_i$, b' inputra, ha $b' \geq 0$, és legyen $f(i, b') = 0$ különben. Most $f(i, b') = 0$ minden olyan párra, ahol $i = 0$ vagy $b' \leq 0$. A rekurzió, amellyel dolgozhatunk, az alábbi:

$$f(i, b') = \max\{f(i-1, b') \ ; \ f(i-1, b' - a_i) + c_i\}.$$

Ezzel az összes olyan $f(i, b')$, amelyre $b' \geq 0$, kiszámolható $O(nb)$ lépésben, és megkapható a keresett $f(n, b)$.

12.3. Minimális háromszögelés

Adott egy P konvex n-szög a síkban az $(1, 2, \ldots, n)$ ciklikusan körberendezett ponthalmazon, továbbá minden ij átlójának hossza, melyet $c(i, j)$ jelöl. A feladat a P háromszögelése átlók behúzásával úgy, hogy a felhasznált átlók hosszainak összege legyen minimális.

Kényelmes lesz a c hosszfüggvényt kiterjeszteni a P éleire úgy, hogy ezek hossza legyen nulla, azaz $c(i, i+1) = 0$, $1 \leq i \leq n-1$, és $c(n, 1) = 0$.

A dinamikus programozási megközelítés alapötlete az, hogy minden $1 \leq i \leq j - 2 \leq n - 2$ párra meghatározzuk az optimumot arra a $P(i, j)$ konvex poligonra, melynek csúcsai $(i, i+1, \ldots, j)$. Jelölje a megfelelő optimumokat $f(i, j)$. A feladatunk optimuma az $f(1, n)$ érték.

Ha $j = i + 2$, akkor $P(i, j)$ egy háromszög. Ez alapján indokolt az $f(i, i + 2) = 0$ értékadás minden $1 \leq i \leq n - 2$-re. Legyen továbbá $f(i, i + 1) = 0$ minden $1 \leq i \leq n - 1$-re. Ez utóbbival az elfajuló eseteket fogjuk tudni kezelni.

A döntő megfigyelés az, hogy ha $P(i, j)$ egy optimális háromszögelésében az ij él az (i, j, k) háromszögben van, akkor a háromszögelés megszorítása a $P(i, k)$ és $P(k, j)$ poligonokra azok optimális háromszögelése kell legyen. A k csúcs pedig valahol $i + 1$ és $j - 1$ között van. A rekurzió ez alapján:

$$f(i, j) = \min_{i<k<j} \{f(i, k) + f(k, j) + c(i, k) + c(k, j)\}$$

Vegyük észre, hogy a képlet az $k = i+1$ és $k = j-1$ esetben is érvényes, hiszen az élek hosszát nullának definiáltuk.

12.4. Mátrixok szorzása

Adott m mátrix, $A_1, A_2, ..., A_m$, számoljuk ki a szorzatukat. Határozzuk meg a szorzásoknak azt a sorrendjét, amelyre az elemi szorzások száma a lehető legkisebb.

Fel kell tennünk, hogy a mátrixok méretei alkalmasak a szorzásra, azaz valamely $p_0, p_1, ..., p_m$ pozitív egészekre az A_i mátrixnak p_{i-1} sora van és p_i oszlopa, minden $1 \leq i \leq m$ indexre. Egy $a \times b$ és egy $b \times c$ méretű mátrix szorzata egy $a \times c$ méretű mátrix. A szorzat minden elemének kiszámolásához b elemi szorzásra van szükség. Az elemi szorzások teljes száma tehát abc. Legyen ez a két mátrix szorzási költsége.

A mátrixok szorzási sorrendjét egy zárójelezéssel adhatjuk meg. Legyen mondjuk $m = 4$. Ekkor három lehetőség (összesen öt van) a megfelelő szorzási költségekkel:

$(A_1A_2)(A_3A_4)$ esetén $p_0p_1p_2 + p_2p_3p_4 + p_0p_2p_4$,
$((A_1A_2)A_3)A_4$ esetén $p_0p_1p_2 + p_0p_2p_3 + p_0p_3p_4$,
$(A_1(A_2A_3))A_4$ esetén $p_1p_2p_3 + p_0p_1p_3 + p_0p_3p_4$.

Látszik, hogy nem mindegy a sorrend. Jelölje $A_{i...k}$ az $A_i...A_k$ mátrixok szorzatát és legyen $c_{i...k}$ az ehhez tartozó minimális szorzási költség. A célunk a $c_{1...m}$ meghatározása.

Tegyük fel, hogy egy optimális zárójelezés olyan, hogy az A_j-t követően bezáródik egy maximális zárójel és elkezdődik egy másik maximális. (Egy összetartozó zárójelpár maximális, ha nincs benne nagyobban.) Ekkor az $A_1, ..., A_j$ és az $A_{j+1}, ..., A_m$ részsorozatokra megszorított zárójelezés is optimális kell legyen.

Az összes $A_i, ..., A_k$ részsorozatra ($1 \leq i \leq k \leq m$) meg fogjuk határozni az optimális $c_{i...k}$ költséget. Ilyen részsorozatokból nincs túl sok: $\binom{m+1}{2}$. Az egytagúakra persze $c_{i...i} = 0$. Tegyük fel, hogy a legfeljebb t tagúakra már meghatároztuk az optimumokat. Ekkor a $t + 1$ tagúakkal a következőképp járhatunk el.

Legyen $A_i, ..., A_k$ ilyen, azaz $k = i + t$. Ekkor

$$c_{i...k} = \min\{c_{i...j} + c_{j+1...k} + p_{i-1}p_jp_k : j = i, ..., k - 1\},$$

Tehát a már kiszámoltakból gyorsan kapható a következő érték.

12.5. Legrövidebb utak minden pontpárra

Adott egy $D = (V, A)$ digráf, valamint az élein egy $c : A \to \mathbb{R}$ konzervatív (azaz negatív kör nélküli) hosszfüggvény. Célunk egy minimális hosszúságú s-t utat keresni, az összes s, t (rendezett) pontpárra. A korábban tanult Bellman–Ford-algoritmust futtatva minden $s \in V$ pontból, megkapjuk a választ $O(|V|^2|A|)$ lépésben.

Van ennél gyorsabb módszer is, amely sűrű gráfokon egy nagyságrendet javít a lépésszámon. Floyd és Warshall alábbi algoritmusa – bizonyos értelemben – egyszerre számolja ki a keresett minimumokat, nem külön mindegyik gyökérpontra.

Az algoritmus n fázisból áll, ahol $n = |V|$. Rögzítsük a pontok egy $v_1, v_2, \ldots, v_n$ sorrendjét és jelölje $V_k = \{v_1, v_2, \ldots, v_k\}$ a sorrend kezdőszeletét, minden $1 \leq k \leq n$ számra. Legyen $w_0(i, j) = c(v_iv_j)$, ha $v_iv_j \in A$, és legyen $w_0(i, j) = +\infty$ különben, minden v_i, v_j pontpárra. A k-adik fázisban minden v_i, v_j pontpárra meghatározzuk a v_i-ből v_j-be menő legrövidebb olyan út hosszát, amelynek minden belső pontja benne van a V_k halmazban. Jelölje $w_k(i, j)$ ezt a minimumot. (Figyeljük meg, hogy w_0 definíciója ezzel konzisztens: az megadja a legrövidebb belső pont nélküli út hosszát.)

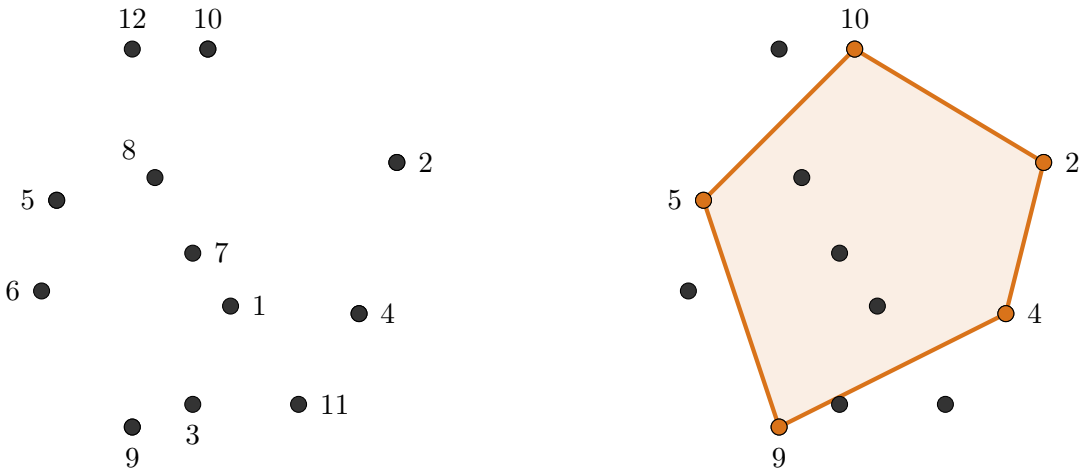
Legyen P egy minimális olyan, v_i-ből v_j-be menő út, amelynek minden belső pontja V_k-ban van. Ha P nem tartalmazza a v_k pontot, azaz minden belső pontja V_{k-1}-ben van, akkor $w_k(i,j) = w_{k-1}(i,j)$. Ha a v_k rajta van a P úton, akkor a P-nek a v_i-től v_k-ba és a v_k-ból v_j-be vezető részútjai olyanok, hogy minden belső pontjuk V_{k-1}-ben van. A P minimalitása miatt ezek a részutak legrövidebbek kell legyenek az ilyen utak közül (ne felejtsük el, hogy c konzervatív!), azaz ezek hosszai $w_{k-1}(i,k)$, illetve $w_{k-1}(k,j)$. Ezek alapján érvényes az alábbi rekurzió: minden i,j párra és minden k-ra

$$w_k(i,j) = \min\{w_{k-1}(i,j), w_{k-1}(i,k) + w_{k-1}(k,j)\}.$$

Ennek felhasználásával az összes $w_k(i,j)$ érték, és így a keresett $w_n(i,j)$ értékek is $O(|V|^3)$ lépésben kiszámolhatók.

12.6. Legnagyobb konvex részhalmaz

A síkban adott egy általános helyzetű S ponthalmaz. Válasszunk ki egy maximális méretű **konvex** részhalmazt S-ből. (Egy K ponthalmaz konvex, ha K minden pontja a K konvex burkának határán van.)



12.1. ábra. Az ábrán látható ponthalmazban az $\{5, 10, 2, 4, 9\}$ részhalmaz konvex.

Az S halmaz pontjai az (x_i, y_i) koordinátákkal adottak, és akkor tekintjük őket általános helyzetűnek, ha semelyik három pont sem esik egy egyenesre és nincs két olyan pont, melyek x koordinátája megegyezik.

Egy $a, b \in S$ pontpárra, melyre $x_a < x_b$, jelölje $f(a,b)$ az a és b közti, S-beli pontokból álló 'konkáv ívek' maximális hosszát (ahol a hossz a pontok száma mínusz egy). Ilyen konkáv ív maga az ab szakasz, továbbá minden olyan S-beli U konvex ponthalmaz ilyet alkot, melyre $a, b \in U$, valamint $x_a \leq x_v \leq x_b$ és v az ab szakasz felett van minden $v \in U$-ra. Hasonlóan definiáljuk a $\bar{f}(a,b)$ értékeket az a és b közötti 'konvex ívekre'. A megoldás egyik kulcsa az a megfigyelés, hogy az optimumra, jelölje OPT, teljesül, hogy

$$OPT = \max\{f(a,b) + \bar{f}(a,b) : a, b \in S, x_a < x_b\}.$$

A másik fontos észrevétel az, hogy az $f(a,b)$ érték kiszámolására van hatékony módszer, mely dinamikus programozási technikát használ. Ehhez vezessük be az $f(a,b,z)$ értékeket minden olyan ponthármasra, melyre $x_a \leq x_z < x_b$ és amelyre z az ab szakasz fölött van. Ha $a = z$, akkor legyen $f(a,b,z) = 1$, egyébként pedig legyen az olyan a és b közti, S-beli pontokból álló 'konkáv ívek' maximális hossza, melyek utolsó előtti pontja a z. Ezekre az értékekre teljesül, hogy

$$f(a,b,z) = \max\{f(a,z,u) : u \in S, u \; abz\text{-megengedett}\} + 1,$$

ahol egy u pont akkor abz-**megengedett**, ha nem a z pont és benne van az a-n átmenő függőleges egyenes, az a és z egyenese, illetve a b és z egyenese által meghatározott háromszögben. Ezt a rekurziót használva, az a, b pontok távolsága szerint növekvő módon az összes $f(a,b,z)$ érték megkapható. (Itt az a és b távolsága azon S-beli v pontok száma, melyekre $x_a \leq x_v < x_b$.) Ezek után az

$$f(a,b) = \max\{f(a,b,z) : z \in S, x_a \leq x_z < x_b, z \text{ az } ab \text{ szakasz fölött van}\}$$

képlet alapján az $f(a,b)$ értékek is kiszámolhatók, majd hasonlóan az összes $\bar{f}(a,b)$ érték is. Ezekből végül OPT a fenti megfigyelés alapján kiolvasható.

12.6.1. Feladat. *Terjesszük ki a fenti megoldást arra az esetre, amikor nem tesszük fel, hogy a pontok x koordinátái különbözők.*

12.7. Feladatok

12.7.1. Feladat. *Hogyan lehet alkalmazni a Floyd–Warshall-algoritmust annak eldöntésére, hogy van-e a digráfban, adott c élhosszakra nézve, negatív kör?*

12.7.2. Feladat. *Legyenek a G páros gráf színosztályai $A = \{a_1, a_2, \ldots, a_n\}$ és $B = \{b_1, b_2, \ldots, b_m\}$. Az M párosítás G-ben* **keresztezés-mentes***, ha M minden a_ib_k és a_jb_l élpárjára, amelyre $i < j$, egyúttal $k < l$ is teljesül. Adjunk polinomiális algoritmust egy maximális keresztezés-mentes párosítás megtalálására.*

13. fejezet

Összefoglalás

VIZSGATEMATIKA címszavakban

Definíció: szomszédsági mátrix, éllista, polinomiális algoritmus, input mérete

Algoritmus: digráfban adott pontból elérhető pontok megkeresése Elért/nem elért, átvizsgált/nem átvizsgált címkék. Feszítő fenyő az elért pontokon, távolság, BFS és DFS speciális esetként, BFS szinthalmazok, legrövidebb utak fenyője/fája

Lemma: leálláskor elért címkéjű = elérhető

Futási idő

DFS alkalmazásai: elvágó pontok, 2-SAT feladat, minimális méretű 2-élösszefüggő feszítőgráf

Definíció: fenyő, erősen összefüggő digráf, kifok, befok, vegyes gráf, egyirányú vágás, gráf irányítása, elvágó él, vegyes gráf

Lemma: D-ben pontosan akkor van u-v út, ha minden u-v szeparáló halmazból legalább egy él kilép

Lemma: D pontosan akkor erősen összefüggő, ha a pontok minden valódi részhalmazának kifoka legalább 1

Tétel: G-nek van erősen összefüggő irányítása akkor és csak akkor, ha összefüggő és nincs elvágó éle

Következmény: algoritmus erősen összefüggő irányítás meghatározására

Következmény: vegyes gráfnak mikor van erősen összefüggő irányítása

Algoritmus: Dijkstra-algoritmus

Legrövidebb utak fenyője, legrövidebb út meghatározása, futási idő

Definíció: séta, konzervatív élhossz függvény

Lemma: konzervatív esetben van olyan legrövidebb séta, ami út

Algoritmus: Bellman–Ford-algoritmus

Futási idő, legrövidebb út meghatározása

Alkalmazások: hátizsák feladat, projekt ütemezés (PERT módszer)

Definíció: feszítőfa, mohó erdő

Lemma: mohó erdő növelése egy éllel

Algoritmus: Mohó algoritmus (két változat)

Definíció: közelítő algoritmus, közelítési hányados, Steiner-fa, metrikus élhosszak

Algoritmus: 2-közelítő algoritmus minimális lefogó ponthalmazra

Algoritmus: 2-közelítő algoritmus a minimális Steiner-fa feladatra

Definíció: élszínezés

Kőnig élszínezési tétele

Alkalmazás: az órarendkészítési feladat

Definíció: folyam, kapacitás, vágás, hálózat, folyam nagysága

Lemma: maximális folyam nem nagyobb a minimális vágásnál

Optimalitási feltételek

Algoritmus: Ford–Fulkerson növelő utas algoritmus

Tétel: max flow min cut tétel racionális kapacitásokra, egész kapacitások esete. Skálázás

Alkalmazás: szállítási feladat, nyerhet-e a csapat, projekt kiválasztás

Futási idő, Edmonds–Karp-algoritmus.

Lemma: folyamok felbontása útfolyamokra és nulla nagyságú folyamra

Tétel: Menger tétele irányított gráfok éldiszjunkt útjairól, és a további három változat

Definíció: lokális élösszefüggőség, globális élösszefüggőség

Algoritmus: gráf globális élösszefüggőségi számának meghatározása folyamokkal

Definíció: 2-él- és 2-pont-összefüggő gráfok, fülfelbontás

Lemma: a G gráf pontosan akkor 2-él-összefüggő, ha van fülfelbontása

Definíció: folyamekvivalens fa

Tétel: folyamekvivalens fa minden (élkapacitásos) gráfhoz létezik

Definíció: lineáris függvény, lineáris egyenlőség és egyenlőtlenség, lineáris feltétel, LP feladat, standard alak, megengedett megoldás, célfüggvény, hipersík, féltér, poliéder, politóp, konvex, konvex burok, csúcs

Tételek: korlátos poliédernek véges sok csúcsa van, korlátos poliéder a csúcsainak konvex burka, korlátos esetben van optimális csúcs, korlátos poliéder = politóp (ezek bizonyítás nélkül)

Tétel: Birkhoff–Neumann-tétel duplán sztochasztikus mátrixokról

Definíció: karakterisztikus vektor

Poliéderes kombinatorika alapgondolata, diszkrét optimalizálási feladat politópja és célfüggvénye

Tétel: páros gráf teljes párosítás poliéderének leírása

Következmény: páros gráf párosítás poliéderének leírása

Definíció: IP feladat, rácspont, LP relaxáció, félegész megoldás

Konkrét problémák IP felírása (pl. lefogó ponthalmaz, maximális súlyú feszítőfa), ezek LP relaxációja

Tétel: lefogó ponthalmaz LP-hez van félegész optimális megoldás

Algoritmus: a minimális súlyú lefogó ponthalmaz feladatra 2-közelítő algoritmus LP relaxációval

Dinamikus programozás alapgondolata

Példák: részösszeg, hátizsák, háromszögelés, mátrixszorzás, legrövidebb utak minden pontpárra, legnagyobb konvex részhalmaz

Irodalomjegyzék

[1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, *Network flows: Theory, Algorithms, and Applications*, Prentice Hall, 1993.

[2] V. Chvatal: *Linear Programming*, W.H. Freeman and Company, 1983.

[3] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, A. Schrijver: *Combinatorial Optimization*, Wiley, 1998.

[4] Frank András: *Operációkutatás*, ELTE, elektronikus jegyzet.

[5] Frank András, Jordán Tibor: *Diszkrét Optimalizálás*, ELTE TTK elektronikus jegyzet, Typotex kiadó, 2013.

[6] Jordán Tibor, Recski András, Szeszlér Dávid: *Rendszeroptimalizálás*, Typotex kiadó, 2004.

[7] Király Tamás, Kis Tamás, Szegő László: *Egészértékű Programozás*, elektronikus jegyzet.

[8] J. Kleinberg, É. Tardos: *Algorithm Design*, Pearson International Edition, Addison Wesley, 2006.

[9] Alexander Schrijver: *A Course in Combinatorial Optimization*, online jegyzet.

[10] V.V. Vazirani: *Approximation Algorithms*, Springer, 2001.

[11] D.P. Williamson, D.B. Shmoys: *The Design of Approximation Algorithms*, Cambridge University Press, 2011.



ELTE | EÖTVÖS KIADÓ

Az ELTE hivatalos könyvkiadója



ELTE | TTK

TERMÉSZETTUDOMÁNYI KAR

