EP, közelítés

1. Tegyük fel, hogy van egy eljárásunk, ami egy tetszőleges n csúcsú gráfról polinom időben megmondja, hogy az kiszínezhető-e 3 színnel. Hogyan lehet ezt felhasználva polinom időben megtalálni egy adott G gráfnak egy 3 színnel színezését (ha van ilyen egyáltalán)?

   Megoldás: Az ötlet hasonló az előzőhöz, úgy akarjuk módosítani a gráfot, hogy a végére már csak egy megoldás maradjon és az könnyen kiolvasható legyen a kapott gráfból.

   Kezdjük most is azzal, hogy az eljárásnak beadjuk a G gráfot. Ha az az eredmény, hogy nem lehet 3 színnel kiszínezni, akkor készen vagyunk.

   Ellenkező esetben húzzunk be egy új élet és az így kapott G' gráfot adjuk az eljárásnak. Ha ez is jó, akkor folytassuk a $G = G'$ gráffal, különben meg G-vel. Ha ezt minden hiányzó élre megcsináljuk, akkor a végeredményül kapott gráf még mindig kiszínezhető 3 színnel, viszont egyetlen további él sem húzható be úgy, hogy ez a tulajdonság megmaradjon. Azaz a gráf pontjai 3 független halmazra oszthatók, amik között minden él be van húzva. Tehát ha azokat színezzük egyformára, amik között nincs él, egy jó színezést kapunk.

   A módszer során az eljárást $O(n^2)$-szer hívtuk (n a gráf csúcsainak száma) és ugyanennyiszer kell egy élet hozzáadnunk, így tehát a feltevés miatt egy polinomiális algoritmust kaptunk.

   Egy másik megoldás: A csúcsokat egymás után megpróbáljuk úgy kiszínezni, hogy folytatható legyen az egész gráfra. De az eljárásnak nem adhatunk be egy részben kiszínezett gráfot, ezért G mellé vegyünk fel 3 új csúcsot, legyenek ezek a, b és c, és ezt a három csúcsot kössük össze egymással. Most ha G egy v csúcsát összekötjük a-val és b-vel, és az így kapott gráf kiszínezhető 3 színnel, akkor v színe a c színével megegyező.

   Ezek után a módszer a következő lehet: először az eljárást futtatjuk azon a G' gráfon, ami G-ből és a 3 új csúcson levő K_3-ból áll. Ha ez nem színezhető ki 3 színnel, akkor G sem, és készen vagyunk. Utána sorban minden v csúcsra csináljuk a következőt: Először összekötjük a-val és b-vel, és ezt adjuk az eljárásnak. Ha az a válasz, hogy igen, akkor ez lesz az új G'. Különben v-t összekötjük a-val és c-vel, ha most az a válasz, hogy ez a gráf 3 színnel színezhető, akkor ez az új G', egyébként meg az, amikor v a b-vel és c-vel van összekötve. (Ha a másik kettő nem, akkor ez a harmadik biztos kiszínezhető 3 színnel.) Végül G minden csúcsa össze lesz kötve az a, b, c csúcsokból kettővel, azok lesznek egyforma színűek, amelyek ugyanazzal a kettővel vannak összekötve.

   Lépésszám: El kell készíteni a G' gráfot, ehhez mindig új éleket adunk, ami polinom lépés, $(2n + 1)$-szer hívjuk meg az eljárást, ami összesen $O(n^{c+1})$, tehát polinom lépés.

2. Tegyük fel, hogy van egy eljárásunk, ami ha egy $G = (V, E)$ gráfot és egy $k \leq |V|$ pozitív egészet kap bemenetként, akkor polinom időben megmondja, hogy van-e G-ben k független pont.

   (a) Hogyan lehet ezt felhasználva polinom időben meghatározni egy adott H gráfban a legnagyobb független ponthalmaz méretét?

   (b) Hogyan lehet megtalálni egy ilyen legnagyobb ponthalmazt polinom időben?

   Megoldás: (a) Az eljárást hívjuk meg a $(H, 1)$, $(H, 2)$, ... párokra. Ha a k-adik az utolsó, amire a válasz igen, akkor $\alpha(H) = k$. Nyilván legfeljebb annyi hívás lesz, amennyi csúcsa van H-nak, tehát ez összesen polinom sok lépés. (A megfelelő értéket bináris kereséssel is meghatározhatjuk, akkor a lépésszám is kevesebb lesz.)

   (b) Most már tudjuk, hogy k méretű független halmazt keresünk. Élek behúzogatásával eljuthatunk egy olyan G-t tartalmazó gráfhoz, amiben egy k méretű független ponthalmaz kivételével már minden él be van húzva (ahogy a 3 színnel színezhetőség 1. megoldásában). Ehhez összesen kevesebb, mint n^2-szer kell hívni az eljárást n pontú gráfokra, ami polinomiális lépésszám.

3. Írja le egészértékű programozási feladatként az alábbi problémák optimalizálási változatát!

   (a) MAXKLIKK

   (b) MAXPÁROSÍTÁS

   (c) Minimális lefogó ponthalmaz keresése egyszerű, irányítatlan gráfban

Megoldás:

(a) Az adott G gráfban maximális klikket keresünk, amibe a megfelelő csúcsokat kell beválasztani. Ezért a változók feleljenek meg a csúcsoknak.

A feltételek: $x_i \geq 0$, $x_i \leq 1$ minden $1 \leq i \leq |E(G)|$ esetén és $x_i + x_j \leq 1$ minden $\{i, j\} \notin E(G)$.

A célfüggvény: $\max \sum_i x_i$

Ez jó: Tekintsünk egy teljes részgráfot G-ben. Ekkor ha $x_i = 1$ a kiválasztott csúcsokra, egyébként meg $x_i = 0$, akkor ezek a változók teljesítik az egyenlőtlenségeket, mert két nem összekötött pont közül legfeljebb az egyik tartozik a teljes részgráfhoz.

Másrészt ha az x_i-k az egyenlőtlenségeket teljesítő egész számok, akkor mindegyik változó értéke 1 vagy 0, Azokat a csúcsokat kiválasztva, melyekre $x_i = 1$ egy teljes részgráfot kapunk, mert olyan pontpárból, amelyek nincsenek összekötve, legfeljebb az egyik lehet benne.

A célfüggvény pont azt írja le, hogy a maximális méretűt keressük.

(b) Az adott G gráfban maximálisan sok független élet keresünk, azaz most az élek beletartozásáról kell dönteni. Ezért a változók feleljenek meg az éleknek.

A feltételek: $x_i \geq 0$, $x_i \leq 1$ minden $1 \leq i \leq n$ esetén és $x_i + x_j \leq 1$ ha az f_i és f_j éleknek van közös csúcsa.

A célfüggvény: $\max \sum_i x_i$

Ez jó: Tekintsünk egy párosítást. Ekkor ha a párosítás éleire $x_i = 1$, egyébként meg $x_i = 0$, akkor ezek a változók teljesítik az egyenlőtlenségeket. Másrészt ha az x_i-k az egyenlőtlenségeket teljesítő egész számok, akkor mindegyik változó értéke 1 vagy 0, és azokat az éleket kiválasztva, melyekre, $x_i = 1$ egy párosítást kapunk, mert minden csúcsra csak egy olyan él illeszkedhet, amire $x_i = 1$.

A célfüggvény pont azt írja le, hogy a maximális méretűt keressük.

(c) Ez egy minimalizálási feladat, de a mi EP változatunk maximalizálás. Egy lehetséges megoldás, hogy észrevesszük, hogy egy minimális lefogó rendszer komplementere maximális független ponthalmaz. Ezért a változók feleljenek meg a csúcsoknak.

A feltételek: $x_i \geq 0$, $x_i \leq 1$ minden $1 \leq i \leq n$ esetén és $x_i + x_j \leq 1$ ha az $\{i, j\} \in E(G)$.

A célfüggvény: $\max \sum_i x_i$

Ez jó: Tekintsük egy lefogó pontrendszert. Ekkor, ha a benne levő csúcsokra $x_i = 0$, egyébként meg $x_i = 1$, akkor ezek a változók teljesítik az egyenlőtlenségeket. Másrészt ha az x_i-k az egyenlőtlenségeket teljesítő egész számok, akkor mindegyik változó értéke 1 vagy 0, és azokat a csúcsokat kiválasztva, melyekre, $x_i = 0$ egy lefogó rendszert kapunk, mert minden élnek legalább az egyik vége benne van.

A célfüggvény azt írja le, hogy a komplementere maximális, azaz a ponthalmazunk minimális.

4. Adott egy $G = (V,E)$ irányítatlan, egyszerű gráf, melyben minden $v \in V$ csúcshoz tartozik egy c_v súly, ami tetszőleges egész szám lehet. A G gráf legfeljebb 2000 elemű független csúcshalmazai közül keressünk olyat, melyben a csúcsok súlyainak összege maximális. Fogalmazza meg a kérdést egészértékű programozási feladatként! (A feladatot nem kell megoldani, csak átfogalmazni!)

   Megoldás: Az előző (e) feladatban már tulajdonképpen felírtuk a megfelelő feltételeket, csak azzal kell kiegészíteni, hogy kizárólag a legfeljebb 2000 csúcsú független halmazok érdekesek.

   Minden v csúcshoz tartozik egy x_v változó.

   A feltételek: $x_v \geq 0$, $x_v \leq 1$ minden $v \in V(G)$ esetén, $x_v + x_w \leq 1$ ha az $\{v, w\} \in E(G)$ és $\sum_v x_v \leq 2000$

   A célfüggvény: $\max \sum_v c_v \cdot x_v$

   Ez jó: Az előzőek szerint az egyenlőtlenségek egész megoldásai megfelelnek az olyan független ponthalmazoknak, melyek mérete nem nagyobb, mint 2000.

   A célfüggvény pont a részhalmaz összsúlyát írja le.

5. A LÁDAPAKOLÁS feladatban legyenek a súlyok: $s_1 = 0{,}4$; $s_2 = 0{,}7$; $s_3 = 0{,}1$; $s_4 = 0{,}6$. Hajtsa végre erre
   (a) az FF algoritmust;
   (b) az FFD algoritmust!

(c) Hány ládát használ az optimális pakolás?

Megoldás: (a) 3 láda lesz: $0{,}4 + 0{,}1$, $0{,}7$, $0{,}6$.
(b) 2 láda lesz: $0{,}7 + 0{,}1$, $0{,}6 + 0{,}4$.
(c) Nyilván az utóbbi optimális, hiszen a súlyok összege $0{,}4 + 0{,}7 + 0{,}1 + 0{,}6 = 1{,}8 > 1$, azaz 1 láda nem elég.

6. A LÁDAPAKOLÁS feladatban legyen két súly 0,34 és négy súly 0,33 értékű. Hajtsa végre erre

   (a) az FFD algoritmust!

   (b) Hány ládát használ az optimális pakolás?

   Megoldás: (a) 3 láda lesz: $0{,}34 + 0{,}34$, $0{,}33 + 0{,}33 + 0{,}33$, $0{,}33$.
   (b) 2 láda elég, mindkettő $0{,}34 + 0{,}33 + 0{,}33$. Mivel ezek a ládák tele vannak, ez nyilván optimális.

7. Tekintsük a LÁDAPAKOLÁS problémának azt a speciális esetét, amikor minden súly 1/2 vagy 1. Igazolja, hogy ez a változat P-ben van!

   Megoldás: Az 1 méretű tárgyak 1-1 ládába kerülnek, az 1/2-eseket meg tetszőlegesen párosíthatjuk. A ládák száma: $n_1 + \lceil (n - n_1)/2 \rceil = \lceil (n + n_1)/2 \rceil$, ha n az összes, n_1 az 1 méretű tárgyak száma.

8. Egy csomagküldő szolgálatnál csupa egyforma dobozba pakolják az elküldendő árut. Céljuk az, hogy a ládákban maradó üres helyet kitöltő anyagból minél kevesebbre legyen szükség. Valaki azt javasolja, hogy alkalmazzák az FF algoritmust a pakolásra. Igaz-e, hogy ez erre a problémára is egy 2-közelítő algoritmust ad?

   Megoldás: Ez is egy minimalizálási feladat, a kimaradó helyet minimalizáljuk. Az hogy az FF itt nem 2-közelítő, látszik pl. az 1(a)-ból, ahol az FF-nél a kimaradó hely: 1,2, míg az optimálisnál 0,2.

   Megjegyzés: az FF erre a feladatra semmilyen c konstansra nem c-közelítő. Egy egyszerű példa erre, ha az optimálisnál nem marad ki hely, az FF-nél meg igen (a több ládából). Vagy pl. $0{,}5$, $0{,}5 - 2a$, $0{,}5$, $0{,}5 + a$ esetén az optimális 2 ládás megoldás $0{,}5 + 0{,}5$ és $(0{,}5 - 2a) + (0{,}5 + a)$, az FF meg $0{,}5 + 0{,}5 - 2a$, $0{,}5$, $0{,}5 + a$, azaz a kimaradó hely az optimális esetben a, az FF-nél $1 + a$. Ha az a értékét úgy választjuk, hogy $(1 + a)/a > c$, azaz $a < 1/(c - 1)$, akkor látszik, hogy ez nem c-közelítés.

9. Az UTAZÓÜGYNÖK problémára tekintsük a következő mohó algoritmust: tetszőleges csúcsból indulunk, és minden alkalommal a legkisebb olyan súlyú élen megyünk tovább, ami új csúcsba visz. Ha már minden csúcsban jártunk, akkor egy élen visszatérünk a kezdő csúcsba.

   Tetszőlegesen adott $c \geq 1$ konstansra adjon meg olyan gráfot, ami mutatja, hogy ez nem c-közelítő algoritmus!

   Megoldás: Legyen pl. a gráf a K_4, a súlyok: $s(1,2) = s(2,3) = s(3,4) = 1$, $s(1,3) = s(2,4) = 2$ és $s(1,4) = a$. Az 1 csúcsból kezdve az algoritmus az $\{1,2\}$, $\{2,3\}$, $\{3,4\}$ éleket választja ami után csak a $\{4,1\}$ jöhet, ennek súlya összesen $3 + a$. Az $\{1,2\}$, $\{2,4\}$, $\{4,3\}$, $\{3,1\}$ útvonal súlya pedig 6. Az arányuk $(3 + a)/6 > c$, ha $a > 6c - 3$.