Bonyolultságelmélet

Thursday 20th October, 2016, 21:24

Néhány további **NP**-teljes közismert probléma

Számos további **közismert** probléma is **NP**-teljes, mint például:

- Adott egy, az ETERNITY II szabályainak megfelelő csempehalmaz. Kirakható-e a csempékből szabályosan egy négyzet?
- Adott az AKNAKERESŐ játék egy részlegesen kitöltött táblája. Be lehet-e fejezni a kitöltést, hogy konzisztens legyen?
- Adott a SUPER MARIO játék egy pályája. Végig lehet-e rajta jutni?
- Adott a FLOOD-IT játék egy pályája. Megoldható-e?
- Adott egy részlegesen kitöltött SUDOKU tábla. Konzisztens-e?
- Adott a TETRIS játékban elemek egy érkezési sorrendje és egy K szám.
  - El lehet-e tüntetni K sort?
  - Le lehet-e az első K elemet pakolni anélkül, hogy betelne a tábla?
  - Lehet-e K-szor eltüntetni egyszerre négy sort?
-

NP-teljes problémák halmazokra és számokra

Az Egész Értékű Programozás egy másik speciális esete:

Hátizsák

Adott: n elem mindegyikének w_i súlya és c_i értéke, valamint W és C.
Kérdés: Kiválasztható-e ismétlés nélkül néhány elem úgy, hogy összértékük $\geq C$ és összsúlyuk $\leq W$?

Tétel

A Hátizsák probléma **NP**-teljes.

Ötlet

A Részletösszeg problémában az a_i értékekből rendre készítsünk egy tárgyat $w_i = c_i = a_i$ súllyal és értékkel, továbbá legyen $C = W = K$, a célszám.

Hátizsák

Algoritmus a Hátizsák eldöntésére

A következő rekurzív összefüggéssel számítható $T[i,w]$, ami a legfeljebb az első i tárgy felhasználásával egy w kapacitású hátizsákkal elérhető maximális haszon:

$$T[i,w] = \begin{cases} 0 & \text{ha } i = 0 \\ T[i-1,w] & i > 0, w < w_i \\ \max\{T[i-1,w], c_i + T[i-1,w-w_i]\} & i > 0, w \geq w_i \end{cases}$$

Ez ad egy $\mathcal{O}(n \cdot W)$ időigényű algoritmust, ami **nem** polinom, mert az input mérete $n(\log w_i + \log c_i) + \log W$.

Felfoghatjuk úgy is, hogy az algoritmusunk akkor polinomidejű, ha az inputban a súlyokat unárisan reprezentáljuk, vagy ha a súlyok a tárgyak számának egy polinomjával korlátozhatóak.

Ez vezet el az algoritmikus bonyolultságelméletben a **pszeudopolinomialitás** fogalmához.

Pszeudopolinomiális algoritmusok, erős/gyenge $\mathbf{NP}$-teljesség

Pszeudopolinomiális algoritmusok

Legyen A egy probléma. Egy A-t eldöntő algoritmus **pszeudopolinomiális**, ha tetszőleges $(a_1, \dots a_m)$ inputon a futásideje $\mathcal{O}\Big(\big(\sum_{1\leq i\leq m} a_i\big)^k\Big)$, valamely konstans k-ra.
(Emlékezzünk vissza: az input **mérete** $\sum_{1\leq i\leq m}\mathbf{log} a_i$ volt!
Ha egy $\mathbf{NP}$-teljes probléma eldönthető pszeudopolinomiális algoritmussal, úgy **gyengén $\mathbf{NP}$-teljesnek**, ha pedig unáris változata is $\mathbf{NP}$-teljes, akkor **erősen $\mathbf{NP}$-teljesnek** nevezzük.

Átfogalmazás

Ha egy erősen $\mathbf{NP}$-teljes problémára van pszeudopolinomiális algoritmus, akkor $\mathbf{P} = \mathbf{NP}$.

Pszeudopolinomiális algoritmusok, erős **NP**-teljesség

Példa

- A HÁTIZSÁK probléma gyengén **NP**-teljes.
- A TSP probléma viszont erősen **NP**-teljes.
- Pozitív egészek faktorizálására van pszeudopolinomiális algoritmus, de nem ismert rá polinomidejű.

A gyengén **NP**-teljes problémák mindazon példányai gyakorlatilag megoldhatónak tekinthetőek, melyekben az inputon érkező számok **értéke** korlátozható az input **méretének** egy **polinomfüggvényével**.
(Pl. a HÁTIZSÁK vagy a PARTÍCIÓ sok ,,kisméretű" tárgy esetén.)

PARTÍCIÓ

- **Input**: $a_1, \ldots, a_n$ pozitív egészek.
- **Kérdés**: Van-e $I \subseteq \{1, \ldots, n\}$, melyre $\sum_{i \in I} a_i = \frac{\sum_{1 \leq i \leq n} a_i}{2}$?

A probléma (gyengén) **NP**-teljes.

Relaxáció

A HÁTIZSÁK probléma egészértékű programozási feladatként felírva:

$$\sum_{i=1}^{n} w_i x_i \leq W$$

$$\sum_{i=1}^{n} c_i x_i \geq C$$

$$0 \leq x_i \leq 1$$

Egy ILP feladat **LP-relaxációját** kapjuk, ha az egészek helyett a valósak körében oldjuk meg. (Azaz elhagyjuk az „x_i egész" feltételt.)

Mivel $\mathrm{LP} \in \mathbf{P}$, a relaxált feladat hatékonyan megoldható.

TÖREDÉKES HÁTIZSÁK

- Mint a HÁTIZSÁK, de a tárgyak törhetők: az i-edik tárgy x_i-ed része, $0 \leq x_i \leq 1$ egy $c_i x_i$ értékű, $w_i x_i$ súlyú tárgy.
- Erre a feladatba a hatékony **mohó** algoritmus optimális.
- Kérdés, hogy mennyire jól ,,közelíti'' az eredeti (függvény)problémát?

A következőkben olyan **optimalizálási** problémákra próbálunk adni hatékony **közelítő** algoritmusokat, melyeknek eldöntési változata $\mathbf{NP}$-nehéz.

Az f függvény minimalizálási/maximalizálási probléma, ha $f(I) = \arg\min_{s\in S(I)} c(s)$ / $\arg\max_{s\in S(I)} c(s)$ alakú, ahol $S(I)$ az I inputhoz tartozó **lehetséges megoldások** (nemüres) halmaza, c pedig minden megoldáshoz egy valós költséget/értéket rendelő függvény.

Ha f egy optimalizálási probléma, $\alpha > 0$ egy konstans, A pedig egy olyan polinomidejű algoritmus, melyre $A(I) \in S(I)$ minden I inputra úgy, hogy

$$\forall I: \ \max\left\{\frac{c(A(I))}{c(f(I))}, \frac{c(f(I))}{c(A(I))}\right\} \ \leq \ \alpha,$$

akkor A egy $\boldsymbol{\alpha}$**-approximáló algoritmus** $\boldsymbol{f}$**-re**.

Emlékeztető

Input: $G = (V, E)$ gráf, $K > 0$ egész.
Output: van-e olyan, legfeljebb K méretű X csúcshalmaz G-ben, melyre igaz, hogy G minden élének legalább az egyik végpontja X-beli?

Optimalizálási változat

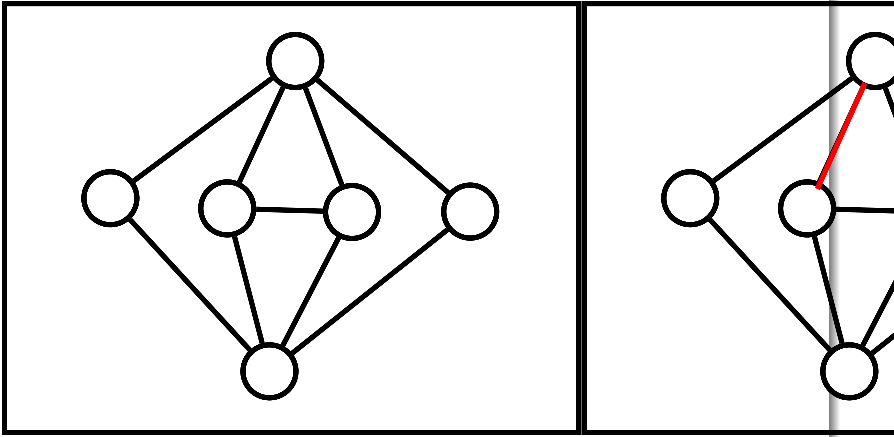
Input: $G = (V, E)$ gráf.
Output: mekkora a **legkisebb** lefogó csúcshalmaz G-ben?
(megoldás: egy lefogó csúcshalmaz; költsége: a mérete)

Természetesen ha az optimalizálási változatra lenne egy hatékony módszer, akkor az eldöntésire is. Mivel az eldöntési változat $\mathbf{NP}$-teljes, így az optimalizálási változatra sem ismert hatékony algoritmus.

Tekintsük a Csúcslefedés-re a következő egyszerű algoritmust:

- Legyen $X = \emptyset$.
- Amíg van él G-ben:
  - Válasszunk egy **tetszőleges** e élt.
  - Vegyük be X-be e **mindkét** végpontját.
  - Vegyük el G-ből az összes, e bármelyik végpontjára illeszkedő élt.
- Adjuk vissza X-et.

Példa



Az algoritmus által visszaadott érték 4. (Lehetne 6 is. Az optimum 3.)

Csúcslefedés

Tétel

Az előző fólián szereplő algoritmus egy 2-approximáló algoritmus a Csúcslefedésre.

Ötlet

- Az algoritmus nyilván egy lefogó csúcshalmazt ad vissza.
- Ha az X halmazba rendre az e_1, e_2, ..., e_k élek végpontjai kerültek bele, akkor $e_1, \ldots, e_k$ egy párosítás G-ben, mindegyiküknek legalább az egyik végpontját le kell fogni, vagyis a minimum legalább k.
- Az algoritmus által visszaadott eredmény pedig $2k$.

A Csúcslefedésre a mai napig **nem ismert ennél jobb** approximációs algoritmus.

TSP

Tétel

A TSP probléma nem közelíthető: **nincs** α-approximáló algoritmus, csak ha $\mathbf{P} = \mathbf{NP}$.

Ötlet

Tegyük fel, hogy van egy α-approximáló A algoritmus TSP-re. Akkor A-t felhasználva meg tudjuk oldani a Hamilton-kört a következőképpen: a $G = (V, E)$ gráfból készítsük $V \times V$-n az alábbi $D(G)$ távolságmátrixot:

$$d_{u,v} = \begin{cases} 1 & \text{ha } (u,v) \in E; \\ |V| \cdot \alpha + 1 & \text{egyébként.} \end{cases}$$

Ekkor ha van Hamilton-kör, az optimum $|V|$; ha nincs, az optimum legalább $|V| \cdot \alpha + 1$.

Ötlet befejezése

Ekkor a következő algoritmus:

Ha $A(D(G)) \leq |V| \cdot \alpha + 1$, fogadjuk el G-t, egyébként utasítsuk el

eldönti a Hamilton-kör problémát, polinomidőben.

Így várhatóan (hacsaknem $\mathbf{P} = \mathbf{NP}$) nincs approximáló algoritmus sem TSP-re.
Kereshetünk viszont olyan speciális esetet, amire van.

Metrikus TSP

A probléma

Input: egy D távolságmátrix, **ami teljesíti a háromszögegyenlőtlenséget**:

$$\forall i, j, k: \; D_{i,j} + D_{j,k} \; \geq \; D_{i,k}.$$

Output: a minimális összsúlyú körút súlya.

Bonyolultság

Az eldöntési probléma továbbra is $\mathbf{NP}$-nehéz.
(Ilyen mátrixot állítottunk elő a Hamilton-kör $\mathrm{TSP}(E)$-re való visszavezetésekor.)

METRIKUS TSP: 2-approximáló algoritmus

Algoritmus

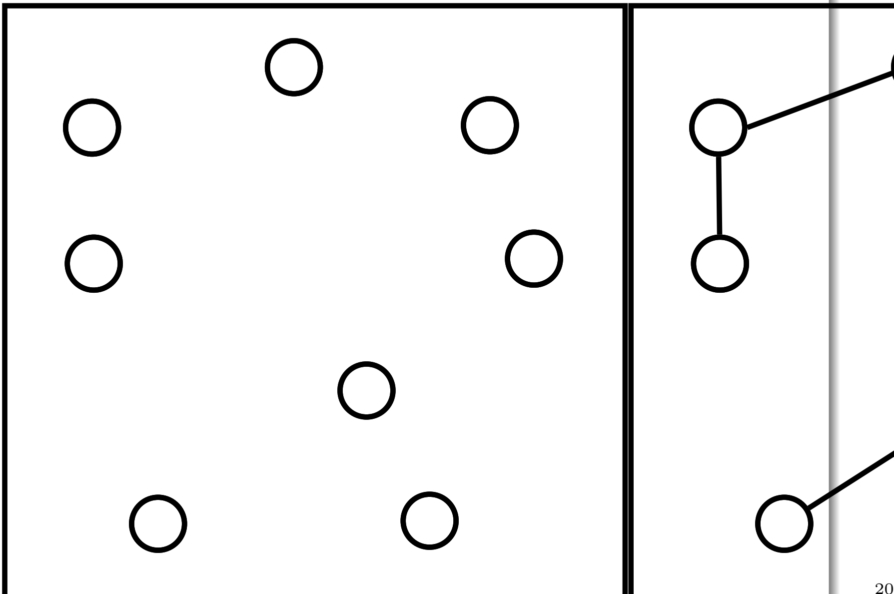
- Állítsuk elő (Prim vagy Kruskal algoritmusával, például) D egy **minimális feszítőfáját**.
- Kettőzzük meg a fa éleit.
- A kapott multigráfnak vegyük egy Euler-körvonalát.
- A körvonalból hagyjuk el a korábban már meglátogatott csúcsokat.
- Adjuk vissza az így kapott kör összsúlyát.

Közelítés

A fenti egy 2-approximáló algoritmus a METRIKUS TSP-re.

Metrikus TSP: 2-approximáló algoritmus

Példa



Metrikus TSP: 2-approximáló algoritmus

2-approximálás

Az algoritmus 2-approximáló, ez a következőkből áll össze:

- egy tényleges körút költségét adja vissza;
- bármilyen körútból elhagyva egy élt egy feszítőfát kapunk $\Rightarrow$ a minimális feszítőfa összsúlya kisebb, mint a minimális körúté;
- az élek kettőzésével kapott gráfban ill. annak az Euler-körvonalában az élek összsúlya tehát kisebb, mint **kétszer** a minimális körúté;
- a már látott pontok kihagyásával **a háromszög-egyenlőtlenség miatt** nem növekszik az összsúly.

Megjegyzés: $\frac{3}{2}$-approximálás

Az élek kettőzése helyett egy ,,ügyesebb'' módszerrel kaphatunk egy $\frac{3}{2}$-approximáló algoritmust is.

Összefoglalás

- Megismertük a **pszeudopolinomiális** algoritmusokat.
- Adtunk egy pszeudopolinomiális algoritmust a HÁTIZSÁK problémára.
- Megismertük a **gyenge** és az **erős NP-teljességet**.
- A HÁTIZSÁKról beláttuk, hogy gyengén, míg a TSP(E) erősen **NP**-teljes.
- Megismertük a gyengén **NP**-teljes PARTÍCIÓ problémát.
- Felírtuk a HÁTIZSÁK probléma LP relaxációját.
- Megismertük az α-approximáló algoritmusokat.
- Megadtunk egy 2-approximáló algoritmust a CSÚCSLEFEDÉS problémára.
- Megmutattuk, hogy a TSP probléma nem közelíthető.
- Megismertük a METRIKUS TSP problémát, megmutattuk, hogy erősen **NP**-teljes és adtunk rá egy 2-approximáló algoritmust.