Algoritmuselmélet

Mit tegyünk, ha nehéz a probléma?

Katona Gyula Y.

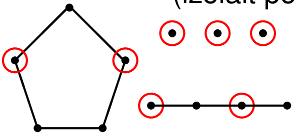
Számítástudományi és Információelméleti Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Elágazás és korlátozás

Mit tegyünk ha kiderül, hogy a megoldandó probléma NP-teljes?
Legtöbbször van c^n-es algoritmus, de nem mindegy mekkora c.
Bontsunk esetekre, azt alesetekre, ... $\Longrightarrow$ fa
Értékeljük az eseteket $\Longrightarrow$ bizonyos irányokban nem kell továbbmenni. $\Longrightarrow$ (korlátozó heurisztika)
Pl. sakkállások
Feladat: Keressünk maximális méretű független ponthalmazt egy adott G gráfban.
NP-teljes
Minden részhalmazt végignézünk $\Longrightarrow$ $O(2^n)$ lépés

Jobb algoritmus

Észrevétel: *Ha G-ben minden pont foka legfeljebb kettő, akkor a feladat lineáris időben megoldható $\Longrightarrow$ G izolált pontok, utak és körök diszjunkt uniója. $\Longrightarrow$ komponensenként minden „második" pontot bevesszük a halmazba.*



MF(G)

1. Ha G-ben minden pont foka ≤ 2, akkor MF(G) az előbbi eljárás által adott maximális független halmaz, és a munkát befejeztük.
2. Legyen $x \in G$, $fok(x) \geq 3$.
   $S_1 := \mathrm{MF}(G \setminus \{x\})$
   $S_2 := \{x\} \cup \mathrm{MF}(G \setminus \{x \text{ és szomszédai}\})$.
3. Legyen S az S_1 és S_2 közül a nagyobb méretű, illetve akármelyik, ha $|S_1| = |S_2|$.
4. MF(G) := S.

Legyen $T(v)$ az MF(G)-n ($|V(G)| \leq v$) belüli MF hívások maximális száma, beleértve MF(G)-t magát is.

Tétel

Van olyan c állandó, hogy $T(v) \leq c\gamma^v$, tetszőleges v természetes számra, ahol γ a $\gamma^4 - \gamma^3 - 1 = 0$ egyenlet pozitív gyöke ($\gamma \approx 1,381$).

Bizonyítás.

Legyen $t(v) := T(v) + 1. \implies T(v) = t(v) - 1$

$T(v) \leq T(v-1) + T(v-4) + 1$, *ha* $v > 4. \implies$

$t(v) \leq t(v-1) + t(v-4)$, *ha* $v > 4$.

Indukcióval: $t(v) \leq c\gamma^v$

$v < 5$*-re elég nagy c-vel* $\checkmark$

$\implies$ *Ezután, ha* $v \geq 5$*, indukciós feltevésből:*

$$\begin{aligned} t(v) &\leq t(v-1) + t(v-4) \leq c\gamma^{v-1} + c\gamma^{v-4} = \\ &= c\gamma^{v-4}(\gamma^3 + 1) = c\gamma^{v-4}\gamma^4 = c\gamma^v. \end{aligned}$$

Összköltség: $O(v^d T(v)) = O(v^d \gamma^v) = O(1,381^v)$.

3-színezés keresése

Feladat: Adott G, keressünk egy 3-színezést.
NP-teljes
Minden lehetséges színezést végignézünk $\Longrightarrow$ $O(3^n)$ lépés
Ötlet: Bizonyos csúcsokat kiszínezünk pirosra, a többiről polinom időben el tudjuk dönteni, hogy kiszínezhetők-e kékkel és sárgával.
Összköltség: $O(2^n n^c)$.

Jobb modellezés

Lehet, hogy a probléma NP-teljes, ha az input bármilyen gráf lehet, viszont van polinomiális algoritmus, ha az input csak valamilyen speciális gráf lehet. Pontosabb modell felállításával kiderülhet, hogy ez a helyzet.

Példák:

- Van-e Hamilton-kör fában?
- Van-e Hamilton-kör soros-párhuzamos gráfban?
- Van-e Hamilton-kör ha a „favastagság” konstans?
- Maximális klikk keresése intervallum gráfban.
- Kromatikus szám meghatározása páros gráfban, vagy ha a maximális fokszám ≤ 3.
- Kromatikus szám meghatározása síkbarajzolható gráfban.
- Maximális független ponthalmaz keresése páros gráfban. (König+Gallai)
- Élkromatikus szám meghatározása páros gráfban.

Közelítő algoritmusok

Hátha nem szükséges pontos megoldás, elég az optimumtól nem túl messze levő is, ha az polinom időben kiszámolható.

Közelítés additív konstanssal: $OPT - c \leq APPR \leq OPT + c$

Ilyen ritkán van.

Tétel

NP*-teljes annak eldöntése, hogy az élkromatikus szám* $\leq k$ *teljesül-e.*

Tétel (Vizing)

$\Delta(G) \leq \chi_e(G) \leq \Delta(G) + 1.$

Tehát $\Delta(G)$ egy jó közelítés, ami gyorsan kiszámolható.

Átlagosan gyors algoritmusok

- Lineáris programozás megoldására a szimplex módszer átlagosan gyors.
- Gyorsrendezés.
- Hashelés.
- Annak eldöntése, hogy G színezhető-e 3 színnel átlagosan $O(1)$ lépés.

Közelítő algoritmusok

C-KÖZÚT

Bemenet: G

Kérdés: Van-e legalább $v(G) - c$ hosszú út G-ben?

Tétel

C-KÖZÚT $\in$ NP-*teljes.*

Bizonyítás.

C-KÖZÚT $\in$ NP, tanú egy út. $\surd$

H-ÚT $\prec$ C-KÖZÚT: $G \Longrightarrow G' = G + c$ db izolált pont.

Ha G-ben van Hamilton-út, ez G'-ben egy $v(G') - c$ hosszú út. $\surd$

Ha G'-ben van egy $v(G') - c$ hosszú út, akkor az G minden pontját tartalmazza, tehát Hamilton-út. $\surd$ $\square$

Közelítő algoritmusok

Közelítés multiplikatív konstanssal: $\frac{1}{c}OPT \leq APPR \leq c \cdot OPT$

Ilyen sokszor van. Keressünk 1-hez minél közelebbi konstanst.

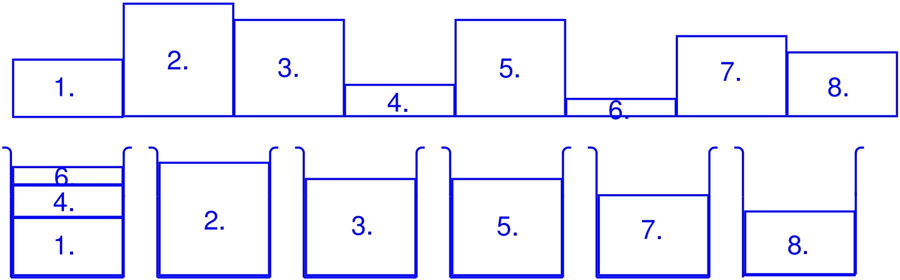
Közelítő algoritmusok

Ládapakolás: Adottak az $s_1, \ldots, s_m$ (racionális) súlyok, $0 \leq s_i \leq 1$. A cél a súlyok elhelyezése minél kevesebb 1 súlykapacitású ládába.

NP-nehéz, a PARTÍCIÓ probléma visszavezethető rá.

Ládapakolás

FF-módszer (*first fit*): Vegyünk először üres ládákat, és számozzuk meg őket az $1, 2, \dots, m$ egészekkel.
Tegyük fel, hogy az $s_1, \dots, s_{i-1}$ súlyokat már elhelyeztük. Ekkor s_i kerüljön az első (legkisebb sorszámú) olyan ládába, amelybe még befér.



First Fit

Tétel

Jelölje a Ládapakolás probléma egy I inputjára $OPT(I)$ az optimális (minimálisan elegendő), $FF(I)$ pedig az FF-módszer által eredményezett ládaszámot. A probléma tetszőleges I inputjára teljesül, hogy $FF(I) \le 2OPT(I) + 1$.

Bizonyítás.

$\left\lceil \sum_{i=1}^{m} s_i \right\rceil \le OPT(I)$
$FF(I) \le \left\lceil 2\sum_{i=1}^{m} s_i \right\rceil + 1 \Longleftarrow$ nincs két olyan láda, amely nincs félig kitöltve.
Felhasználjuk, hogy $\lceil 2x \rceil \le 2\lceil x \rceil$:

$$FF(I) \le \left\lceil 2\sum_{i=1}^{m} s_i \right\rceil + 1 \le 2\left\lceil \sum_{i=1}^{m} s_i \right\rceil + 1 \le 2OPT(I) + 1.$$

$\square$

First Fit

Tétel (D. S. Johnson és munkatársai, 1976)

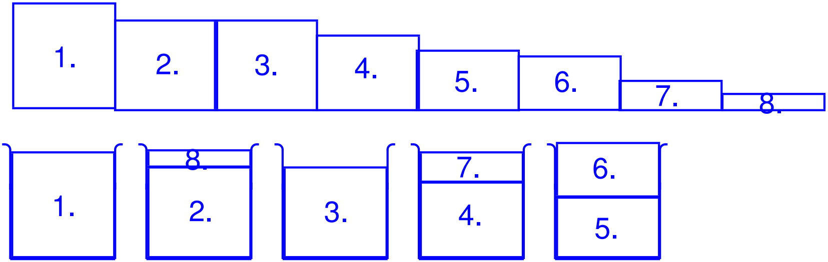
A probléma tetszőleges I inputjára teljesül, hogy $FF(I) \leq \lceil 1.7 OPT(I) \rceil$*. Továbbá vannak tetszőlegesen nagy méretű I inputok, melyekre* $FF(I) \geq 1.7(OPT(I) - 1)$*.*

Tétel (Dósa, Sgall, 2013)

A probléma tetszőleges I inputjára teljesül, hogy $FF(I) \leq \lfloor 1.7 OPT(I) \rfloor$*. Továbbá vannak tetszőlegesen nagy méretű I inputok, melyekre* $FF(I) = \lfloor 1.7 OPT(I) \rfloor$*.*

First Fit Decreasing

FFD-módszer (first fit decreasing): először rendezzük a súlyokat nem növő sorrendbe, utána alkalmazzuk az *FF*-módszert.



Tétel (D. S. Johnson, 1973)

Tetszőleges I inputra teljesül, hogy $FFD(I) \leq \frac{11}{9} OPT(I) + 4$, és tetszőlegesen nagy méretű I inputok vannak, melyekre $FFD(I) \geq \frac{11}{9} OPT(I)$. ($\frac{11}{9} = 1.222\ldots$)

Jobb approximáció

Tétel (W. Fernandez de la Vega, G. S. Lueker)

Tetszőleges $\varepsilon > 0$*-hoz van olyan P lineáris algoritmus, amire* $P(I) \leq (1+\varepsilon)OPT(I) + 1$.

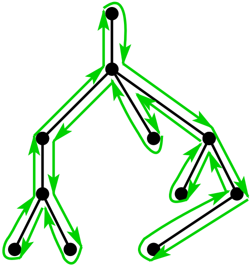
Futásideje: $O(n) + 2^{2^{O((1/\varepsilon)\log(1/\varepsilon))}}$

Euklideszi utazó ügynök probléma

Az n pontú K_n teljes gráf élein adott a nemnegatív értékű d súlyfüggvény. Erre teljesül a háromszög-egyenlőtlenség: tetszőleges különböző u, v, w csúcsokra érvényes a $d(u, w) \leq d(u, v) + d(v, w)$ egyenlőtlenség (az euklideszi feltétel).

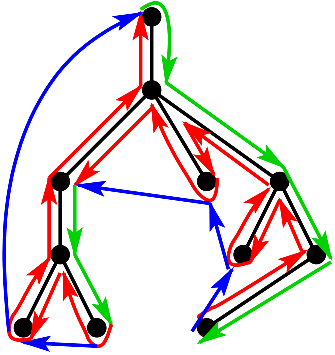
A cél egy minimális összsúlyú Hamilton-kör keresése.

Keresünk egy minimális összsúlyú feszítőfát (pl. Kruskal), megkettőzzük az éleit és „körbejárjuk” egy Euler-körsétával.



A minimális feszítőfa összsúlya legyen s $\implies$ Euler-séta hossza $2s$.

Ez nem Hamilton-kör $\Longrightarrow$ levágjuk a fölösleges részeket, közben rövidítünk is.



Ha az optimális Hamilton-körből elhagyunk egy élet $\Longrightarrow$ egy legalább s súlyú feszítőfát kapunk.

A módszer legfeljebb 2-szer akkora utat ad, mint az optimális.

Véletlent használó algoritmusok

Bemenő adatként adott (binárisan) egy m páratlan egész; szeretnénk eldönteni, hogy m prímszám-e.

Fermat-teszt (m)

1. Válasszunk egy véletlen a egészet az $[1, m)$ intervallumból.
2. Ha $\text{lnko}(a, m) \neq 1$, akkor a válasz „m összetett”.
3. Ha $a^{m-1} \equiv 1 \pmod{m}$, akkor a válasz „$m$ valószínűleg prím”, különben a válasz „m összetett”.

2. Euklideszi algoritmussal gyorsan végrehajtható.
3. Gyors hatványozással gyorsan végrehajtható.

Ha azt kapjuk, hogy „m összetett” $\Longrightarrow$ ez biztos igaz.
Pl.: $m = 21 = 7 \cdot 3$ és $a = 2 \Longrightarrow a$ az m Fermat-tanúja, hiszen $2^{20} \equiv 4 \pmod{21}$.