2021. november 23.

Közelítő megoldás a hátizsák feladatra

Tétel

Tetszőleges $\varepsilon > 0$ valós számhoz létezik olyan polinomiális algoritmus, amely által visszaadott megoldás az optimális megoldás legalább $\frac{1}{1+\varepsilon}$-szorosa

Bizonyítás

Feltehetjük, hogy minden tárgy súlya $\leq W$ (a nagyobbak figyelmen kívül hagyhatók, hiszen triviakkor nem lehetnek benne semmilyen megengedett megoldásban)

Legyen ε olyan pozitív szám, amelynek reciproka egész (elég ilyenekre szorítkozni nyilván)

Legyen $b = \frac{\varepsilon}{2n} v^*$ $\quad [\, v^* = \max_i v_i \,]$

Definiáljunk egy új hátizsákfeladatot:
a súlyok és a hátizsák kapacitása változatlan, csak az értékek újak, v_i helyett legyen az új érték $\tilde{v}_i = \lceil v_i / b \rceil b$ $\quad (1 \leq i \leq n)$

Ez nem olyan nagy változás: $v_i \le \tilde{v}_i \le v_i + b$

Még valami fontos itt: a $\tilde{v}_i$ értékek mind b egész számú többszörösei.

Definiálunk egy harmadik hátizsák feladatot is: a súlyok és a hátizsák kapacitása még mindig ugyanaz, csak az értékek változnak ismét, itt t_i értéke $\hat{v}_i = \tilde{v}_i / b$

Vegyük észre, hogy a harmadik hátizsák feladat optimális megoldása ugyanaz (a tárgyhalmaz), mint a másodiknál.

Futassuk le a második DP algoritmusunkat a harmadik hátizsák feladatra.

Mennyi a költség? Ehhez szükség van $\max_i \hat{v}_i$-re

Mennyi $\max_i \hat{v}_i$? Vegyük észre, hogy ha

$\max_i v_i = v_j$, akkor $\hat{v}_j = \max_i \hat{v}_i$.

Igen ám, de $\hat{v}_j = \lceil v_j / b \rceil = \lceil v^* / b \rceil =$

$\lceil v^* / \frac{\varepsilon}{2n} v^* \rceil = \lceil \frac{2n}{\varepsilon} \rceil = \frac{2n}{\varepsilon}$

Így a költség $O(n^2 \frac{2n}{\varepsilon}) = O(n^3 \varepsilon^{-1})$

ami polinomiális, ha ε rögzített.

Legyen I a DP algoritmus által visszaadott opt. megoldás tárgyainak indexhalmaza.

Nézzük, hogyan viszonyul $\sum_{i \in I} v_i$ az eredeti hátizsák feladat optimális megoldásához.

Ne felejtsük, I megengedett megoldás, hisz a súlyok és a hátizsák kapacitása ugyanaz.

Legyen I^* egy tetszőleges megengedett megoldása az eredeti hátizsák feladatnak (ez persze lehet az optimális is).

Azt fogjuk belátni, hogy $\sum_{i \in I^*} v_i \leq (1+\varepsilon) \sum_{i \in I} v_i$

(ez elég, az optimális I^* pont azt adja ebből, amit szeretnénk)

Hogyan megy ez?

A DP algoritmus által adott megoldás optimális a $\tilde{v}_i$ értékekkel:

$$\sum_{i \in I^*} \tilde{v}_i \leq \sum_{i \in I} \tilde{v}_i$$

Ennélfogva

$$\sum_{i\in I^*} v_i \le \sum_{i\in I^*} \tilde{v}_i \le \sum_{i\in I} \tilde{v}_i \le \sum_{i\in I} (v_i + b) \le \sum_{i\in I} b + \sum_{i\in I} v_i \le nb + \sum_{i\in I} v_i$$

Legyen ismét $v_j = v^* = \max\limits_i v_i$, ekkor

$v_j = \dfrac{2nb}{\varepsilon}$ egyrészt $\qquad \left[b = \dfrac{\varepsilon}{2n} v^* = \dfrac{\varepsilon}{2n} v_j \right]$,

$v_j = \tilde{v}_j$ másrészt $\qquad \left[\tilde{v}_j = \left\lceil \dfrac{v_j}{b} \right\rceil b = \left\lceil \dfrac{v_j}{\frac{\varepsilon}{2n} v_j} \right\rceil \dfrac{\varepsilon}{2n} v_j \right.$

$$\left. = \left\lceil \frac{2n}{\varepsilon} \right\rceil \cdot \frac{\varepsilon}{2n} v_j = \frac{\cancel{2n}}{\cancel{\varepsilon}} \cdot \frac{\cancel{\varepsilon}}{\cancel{2n}} v_j \right]$$

Feltételünk szerint $w_i \leq W$ $(1 \leq i \leq n)$, így

$$\sum_{i \in I} \tilde{v}_i \geq \tilde{v}_j = \frac{2nb}{\varepsilon}$$

(a j-edik tárgy egymaga egy megengedett megoldás, értéke nyilván nem nagyobb, mint az opt. megoldásban az összérték)

Tekintsük újra az egyenlőtlenség láncot, abból is ezt a részt:

$$\sum_{i \in I} \tilde{v}_i \leq nb + \sum_{i \in I} v_i$$

Átrendezve

$$\sum_{i\in I} v_i \geq \left(\sum_{i\in I} \tilde{v}_i\right) - nb,$$

így

$$\sum_{i\in I} v_i \geq \frac{2nb}{\varepsilon} - nb = \left(\frac{2}{\varepsilon} - 1\right) nb = (2-\varepsilon)\frac{nb}{\varepsilon}$$

$$\varepsilon \sum_{i\in I} v_i \geq (2-\varepsilon) nb \geq nb$$

hiszen ε egy pozitív egész reciproka, így nem nagyobb, mint 1

Innen, ismét az egyenlőtlenség láncból

$$\sum_{i\in I^*} v_i \leq \left(\sum_{i\in I} v_i\right) + nb \leq \sum_{i\in I} v_i + \varepsilon \sum_{i\in I} v_i = (1+\varepsilon)\sum_{i\in I} v_i$$

amit bizonyítani akartunk.

Mohó algoritmusok

Alkalmanként a dinamikus programozás megoldása egy feladatnak olyan, mintha ágyúval lőnénk verébre akkor, amikor azt mondjuk:

„az optimális megoldás ezen lehetőségek valamelyike, vegyük számba az összeset és válasszuk a legkedvezőbbet"

Sokszor érezhetjük, hogy melyik a legkedvezőbb!

Persze ezt be is kell látni. Tipikusan az optimális részstruktúra tulajdonság belátásán kívül itt azt is meg kell mutatni, hogy van olyan optimális megoldás, amely "tartalmazza" a mohó választást.

A mohó algoritmus fogalmát nem könnyű megragadni (igazából nincs általános "definíció")

Mohó algoritmust könnyű javasolni, a kihívás a helyesség bizonyítása.

Ez azért is kihívás, mert többnyire egy "mohó algoritmus" nem ad optimális megoldást. Ilyenkor se feltétlenül kell kidobni a mohó algoritmust, mert lehet, hogy egy jó közelítő megoldást ad (pl. NP-nehéz feladatoknál ez elég tipikus)