Štetje rešitev klasičnega problema nahrbtnika

Kratko poročilo pri predmetu Finančni praktikum

Kristina Vatovec in Žan Mikola

December 2020

1 Opis problema

Podanih je n elementov z celoštevilskimi težami $w_1, ..., w_n$ in kapaciteto C, ki je prav tako celo število Privzemimo štetje klasičnega problema nahrbtnika. S pomočjo algoritma, ki temelji na dinamičnem programiranju želimo poiskati oceno za število rešitev problema znotraj relativne napake $1 \pm \epsilon$ v polinomskem času n in $1/\epsilon$.

Preden nadaljujemo, omenimo naslednji izrek, ki je bistvenega pomena pri samem problemu.

Izrek 1. *Podane so teže $w_1, ..., w_n$ in kapaciteta C pri problemu nahrbtnika. Naj bo Z število rešitev problema. Obstaja deterministični algoritem, ki za vsak $\epsilon \in [0,1]$ vrne $Z^{'}$ za katerega velja $Z \leq Z^{'} \leq Z(1+\epsilon)$.*

Poglejmo si še funkcijo $T : \{0, \ldots, n\} \times \{0, \ldots, s\} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$, ki je definirana v spodnjem algoritmu.

Vhod: Celaštevila $w_1, ..., w_n$, C in $\epsilon > 0$.

1. Postavimo $T[0,0] = 0$ and $T[o,j] = \infty$ za $j > 0$.
2. Postavimo $Q = (1 + \epsilon/(n+1))$ in $s = \lceil n log_Q 2 \rceil$.
3. Za $i = 1 \to n$, za $j = 0 \to s$, postavim

$$T[i,j] = \ min_{\alpha \in [0,1]} max \begin{cases} T[i-1, \lceil j + log_Q \alpha \rceil], \\ T[i-1, \lceil j + log_Q (1-\alpha) \rceil] + w_i, \end{cases}$$

   kjer po dogovoru velja $T[i-1, k] = 0$ za $k < 0$.
4. Naj bo
$$j^{'} := max\{j : T[n,j] \leq C\}.$$
5. Izhod $Z^{'} := Q^{j^{'}+1}$

Pri iskanju minimuna funkcije T v odvisnost od parametra $\alpha \in [0,1]$, je v resnici dovolj gledati le α, ki ustrezajo vrednostim diskretne množice S. Za $j \in \{0, 1, \ldots, s\}$, je množiča $S = S_1 \cup S_2$, kjer $S_1 = \{Q^{-j}, \ldots, Q^0\}$ in $S_2 = \{1 - Q^0, \ldots, 1 - Q^{-j}\}$.

Izkaže se, da izhodni podatek $Z^{'}$ zadošča zgoraj napisanem izreku, kar pa je tisto, kar si želimo pri algoritmu za naš problem.

V tem algoritmu smo uporabili funkcijo T, vendar na prvi pogled ni povsem jasno kaj pomeni vrednost, ki jo vrne. Fukcija je T je torej aproksimacijska funkcija funkcije τ. Funkcija $\tau : \{0, \ldots, n\} \times \mathbb{R}_{\geq 0} \to \mathbb{R} \cup \{\pm\infty\}$, kjer $\tau(i,a)$ vrne najmanjši C, Pri katerem obstaja vsaj a rešitev problema nahrbtnika z težami $\omega_1 \dots \omega_n$ in kapaciteto C. Fukcijo τ ne znamo efektivno izračunati, zato definiramo fukcijo T.

Torej točno število rešitev problema nahrbtnika je

$$Z = max\{a : \tau(n, a) \leq C\}$$

2 Načrt dela

Naloga, ki sva si jo zastavila v prihodnje je implementacija zgornjega algoritma. Algoritem bova napisala v programskem jeziku Python. Ko bova algoritem pripravila, sledi eksperimentalni del. Poskušala, bova ugotoviti ali pri različnih vhodnih podatkih prihaja do sprememb v časovni zahtevnosti.

Pri samem programiranju bo največja težava napisati algoritem, ki bo v uglednem času ocenil število rešitev za večje število elementov, ki lahko damo v nahrbtnik. Sama rešitev $Z^{'}$, največkrat ne bo vračala naravnih števil, vendar bomo lahko zaradi enakosti v Izreku 1 lahko ocenili dano rešitev.

Ko bova končala z algoritmom, ga bova morala še testirati. V ta namen bova naredila generator podatkov, ki bo izbral naključno število elemntov in podal nijhove teže, ter izbral naključno težo nahrbtnika. Pričakujeva, da bo za velike n program počasen, saj je njegova računska zahtevnost polinomska.