Online algoritmusok

Algoritmusok és bonyolultságuk

Horváth Bálint

2018. március 30.

Motiváció

- Gyakran előfordul, hogy a bemenetet csak részenként ismerjük meg: a döntésünket a már megkapott információ alapján, a további adatok ismerete nélkül kell meghoznunk.
- Ezekben az esetekben *online problémáról* beszélünk.
- Első eredmények: 1970-es évek.

Fogalmak

Definíció

***Offline algoritmusok**: a teljes bemenet előre adott.*

Definíció

***Online algoritmusok:** a bemenet csak részekben lesz ismert.*

Az online algoritmusok hatékonyságának vizsgálatára két alapvető módszer ismert.

- Átlagos eset elemzése
- Versenyképességi elemzés

Átlagos eset elemzése

- Valószínűségi eloszlást használunk a bemenetek terén.
- A célfüggvénynek az erre az eloszlásra vonatkozó várható értékét vizsgáljuk.
- Hátrány: általában nincs információnk erről az eloszlásról.

Versenyképességi elemzés [1]

- Elterjedtebb: mi is ezt fogjuk használni.
- Legrosszabb-eset alapú megközelítés.
- Az online algoritmus által kapott megoldás célfüggvényértékét hasonlítjuk össze az optimális offline célfüggvényértékkel.

Versenyképességi elemzés [2]

- Egy tetszőleges ALG online algoritmusra az I bemeneten felvett célfüggvényértéket $ALG(I)$-vel fogjuk jelölni.
- Az I bemeneten felvett optimális offline célfüggvényértéket $OPT(I)$-vel jelöljük.
- Ezekkel a jelölésekkel leírhatjuk a C-versenyképesség definícióját.

Versenyképességi elemzés [3]

Definíció

Egy minimalizálási feladatnál az ALG algoritmus ***C-versenyképes****, ha* $ALG(I) \leq C \cdot OPT(I)$ *tetszőleges* I *bemenet esetén.*

Definíció

Egy minimalizálási feladatnál az ALG algoritmus ***enyhén C-versenyképes****, ha létezik olyan* B *konstans, hogy* $ALG(I) \leq C \cdot OPT(I) + B$ *tetszőleges* I *bemenet esetén.*

Definíció

Egy algoritmus ***enyhe versenyképességi hányadosa*** *az a* C *szám, amelyre az algoritmus enyhén* C*-versenyképes.*

Versenyképességi elemzés [4]

A definíció hasonlóan értelmezhető maximalizálási problémák esetén is.

Definíció

Egy maximalizálási feladatnál az ALG algoritmus ***C-versenyképes****, ha* $ALG(I) \geq C \cdot OPT(I)$ *tetszőleges* I *bemenet esetén.*

Definíció

Egy maximalizálási feladatnál az ALG algoritmus ***enyhén C-versenyképes****, ha létezik olyan* B *konstans, hogy* $ALG(I) \geq C \cdot OPT(I) + B$ *tetszőleges* I *bemenet esetén.*

Minimalizálandó függvény esetén $C \geq 1$, míg maximalizálandó feladatnál $C \leq 1$.

Síbérlési feladat [1]

- Elmegyünk a téli szünetben síelni és minden nap síelünk, ha az idő megengedi. De hogy az idő alkalmas lesz-e a következő napon, illetve napokon a síelésre, azt nem tudjuk előre.
- Adott egy pár síléc, amit vagy bérelhetünk 1 egységért, vagy megvásárolhatunk B egységért, ahol $B > 1$ egész.
- Feladat: mikor vásároljuk meg a sílécet?
- Online: amikor egy napon el kell dönteni, béreljük vagy vásároljuk a sít, akkor nincs információnk arról, hogy a következő napokon is síelünk-e.

Síbérlési feladat - V algoritmus [2]

- A bemenet egyetlen pozitív egész, ami megadja, hogy hány napig síelünk.
- Egy online algoritmus csak annyit tehet, hogy valahány napon keresztül bérli a sílécet, és ha addig nem hagytuk abba a síelést, akkor vásárol egyet.
- Azt az algoritmust, amely $V-1$ napig béreli a sít, aztán a V-edik napon megvásárolja, V algoritmusnak nevezzük.

Tétel

A V algoritmus $V = B$ esetén $(2 - \frac{1}{B})$-versenyképes.

A síbérlési feladat - Versenyképesség bizonyítása [3]

Bizonyítás. A feladat I bemenete a napok száma, ameddig síelünk. Két eset lehetséges I függvényében:

- Ha $I < B$, azaz kevesebb napot síelünk, mint amennyi a sílécek ára, akkor mind az online algoritmusnak, mind pedig az optimális offline algoritmusnak a költsége I, tehát $\frac{B(I)}{OPT(I)} = 1$.
- Ha $I \geq B$, azaz legalább annyi napot síelünk, mint amennyi a sílécek ára, akkor $OPT(I) = B$, míg $B(I) = B - 1 + B = 2B - 1$, tehát $\frac{B(I)}{OPT(I)} = \frac{2B-1}{B} = 2 - \frac{1}{B}$.

Mivel mindkét esetben teljesül, hogy a vizsgált hányados legfeljebb $2 - \frac{1}{B}$, az állítást beláttuk. □

A síbérlési feladat -Versenyképesség [4]

Tétel

Nem létezik olyan online síbérlési algoritmus, melynek a versenyképességi hányadosa kisebb, mint $2 - \frac{1}{B}$.

A síbérlési feladat - Versenyképesség bizonyítása [5]

Bizonyítás. A V-algoritmusra mutatunk egy rossz bemenetet. Legyen a feladat bemenete V napnyi síelés, ekkor a költség $V - 1 + B$. Hasonlóan, két esetünk van V értékétől függően:

- Ha $V < B$, akkor az optimális költség V. Tehát a hányados: $\frac{V-1+B}{V} = 1 + \frac{B-1}{V} \geq 2 > 2 - \frac{1}{B}$.
- Ha $V \geq B$, akkor az optimális költség B. A hányadosunk: $\frac{V-1+B}{B} \geq \frac{2B-1}{B}$.

□

A síbérlési feladat általánosan [1]

- Tegyük fel, hogy van egy panziónk, ahol a szállóvendégek részére biztosítjuk a szükséges felszerelést.
- Egyszerre legfeljebb K vendég szállhat meg, azaz legfeljebb K darab sífelszerelésre van szükség.
- Kezdetben egy síléc sem áll rendelkezésre.
- A panzió kölcsönözhet egy napra egy pár sílécet 1 egységért, vagy megvásárolhat egyet B egységért.
- A kölcsönzés naponta történik, bármelyik nap vásárolhatnak bármennyit.

A síbérlési feladat általánosan [2]

- Az nyilvánvaló, hogy nem érdemes K-nál több sílécet vásárolni.
- Kérdés: mikor vásárolja meg a panzió a K darab sílécet?
- Ha soha nem vásárolnánk meg a K darabot, akkor ha a sílécek iránti igények sorozata elég hosszú, és mindig elég sok sílécet igényelnek, az algoritmus nem lehet konstans versenyképes.

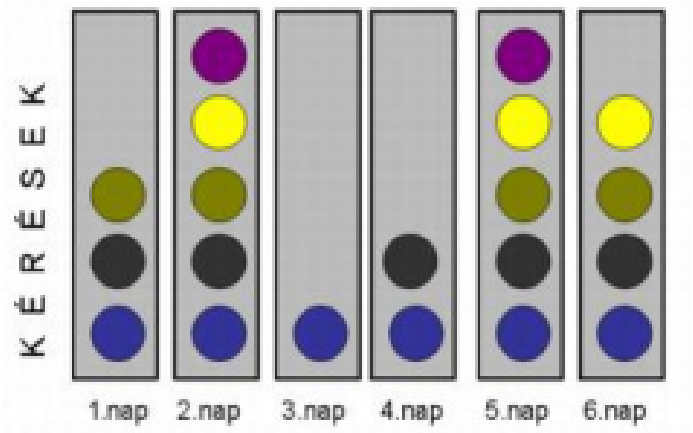
A síbérlési feladat általánosan [3]

A következő, általánosított algoritmus $2 - \frac{1}{B}$-versenyképes, hasonlóan, mint $K = 1$ esetén:

Minden $1 \leq k \leq K$ esetén vásároljuk meg a k-adik sílécet azon a napon, amely napon B-szer érkezett már legalább k darab sílécre igény.

A síbérlési feladat általánosítása [4]

ábra: Az ábra egy hat napos időszakot szemléltet. Minden nap igényelnek valahány sílécet, esetünkben $K = 5$. Az első nap 3, a második nap 5, a harmadik napon egy sífelszerelésre érkezik igény és így tovább.



Memóriakezelési probléma [1]

- A számítógépek gyorsmemóriájának a kezelését modellezik.
- Adott lapok egy halmaza, ebből származó lapok egy sorozata a bemenet.
- k lap kapacitású gyorsmemóriát kell kezelni.
- Ha az aktuálisan igényelt lap nincs a memóriában, akkor az algoritmusnak ezt a lapot be kell tennie és ha már nincs hely, valamely másik lapot ki kell raknia.

Memóriakezelési probléma [2]

- Ha egy igényelt lap nincs a memóriában, azt **hibának** nevezzük.
- Cél: hibák minimalizálása.
- Online probléma, hiszen az algoritmusnak a lap elhelyezéséhez szükséges döntést (azaz, hogy mit távolítson el a gyorsmemóriából) a további igények ismerete nélkül kell meghoznia.

Memóriakezelési probléma [3]

Tétel

Nincs olyan online algoritmus a lapozási problémára, melynek a versenyképességi hányadosa kisebb lenne, mint k, ahol k a gyorsmemória kapacitása.

Bélyegző algoritmus [1]

Az eljárás egyes, korábban kért lapok megjelölésére bélyegeket használ. Kezdetben egyetlen lap sincs megjelölve. Egy bejövő kérés esetén a következő lépéseket hajtja végre:

1. Ha a kért lap a memóriában van és ez a lap még jelöletlen, megjelöljük.
2. Ha a kért lap nincs a memóriában, és nincs már jelöletlen lap a memóriában, az összes jelölést töröljük.
3. Ezután veszünk egy jelöletlen lapot a memóriából (ilyet biztosan találunk az előző lépés miatt), a kért lapot ennek a helyére rakjuk és bejelöljük.

Az algoritmus működését nagyban befolyásolja, hogy milyen szabály alapján választjuk ki a törlendő lapot.

Bélyegző algoritmus [2]

Jelöljük a Bélyegző algoritmust egyszerűen B-vel.

Tétel

A B algoritmus enyhe versenyképességi hányadosa k.

LRU (least recently used) algoritmus

- Gyakorlatban elterjedt módszer.
- Minden esetben azt a lapot töröljük a memóriából, amelyet a legrégebben használtunk.
- A bélyegző algoritmusok családjába tartozik.
- Hasonlít a jól ismert FIFO eljáráshoz, mely szintén azt a lapot rakja ki a memóriából, mely a legrégebben került be.

Véletlenített online algoritmusok [1]

- Az algoritmus véletlen döntéseket is hoz, ezen döntésektől kimenetelétől függ az algoritmus futása.
- A célfüggvény várható értékét vizsgáljuk.
- Az $A(I)$ valószínűségi változó várható értékét vizsgáljuk, melyet összehasonlítunk az optimális megoldás $OPT(I)$ értékével.

Véletlenített online algoritmusok [2]

Példa

Tegyük fel, hogy van két doboz: A és B, amelyek egyike 1000 Ft-ot tartalmaz, a másik üres. 500 Ft-ért választhatunk egy dobozt, aminek a tartalmát megkapjuk. Két determinisztikus algoritmus használható: vagy A-t, vagy B-t választjuk, mindkét esetben az algoritmus költsége a legrosszabb esetben 500.
Azonban, ha egy olyan véletlenített algoritmust használunk, mely $\frac{1}{2}$ valószínűséggel választja mindkét ládát, akkor az algoritmus költségének várható értéke

$$\frac{1}{2} \cdot 500 + \frac{1}{2} \cdot (-500) = 0$$

Véletlenített algoritmus a síbérlési feladatra

- A sílécek vásárlási ára továbbra is B egység, a bérlés ára napi 1 egység.
- A véletlen R algoritmus $\frac{1}{2}$ valószínűséggel a $\frac{3B}{4}$ időpontig vár és utána vásárol.
- $\frac{1}{2}$ valószínűséggel a B időpontig vár és utána vásárol.

Tétel

Az R algoritmus $\frac{15}{8}$-versenyképes.

Véletlenített bélyegző algoritmus

1. Ha a kért lap a memóriában van és ez a lap még jelöletlen, megjelöljük.
2. Ha a kért lap nincs a memóriában, és nincs már jelöletlen lap a memóriában, az összes jelölést töröljük.
3. Ezután veszünk egy jelöletlen lapot **egyenletes eloszlással** a memóriából (ilyet biztosan találunk az előző lépés miatt), a kért lapot ennek a helyére rakjuk és bejelöljük.

Tétel

Ha a bemenetet az algoritmus véletlen döntéseinek eloszlása ismeretében, de a döntések kimenetének ismerete nélkül adjuk meg, akkor a fenti algoritmus $2H_k$-versenyképes, ahol $H_k = \sum_{i=1}^{k} \frac{1}{i}$.

Köszönöm a figyelmet!