Torlódás vezérlés

Feladat: Találjuk meg minimális költséggel a maximális felhasználható sávszélességet az A és B pontok között! A próbák után csak azt az információt kapjuk meg, hogy túlléptük -e a sávszélességet vagy nem. Egy adott t időpontban, u_t jelöli a felhasználható sávszélességet, amit A használhat, és x_t az A által küldött mennyiség.

A statikus modell: A használható sávszélesség egy u konstans.

1. Az enyhe célfüggvény:

$$G_\alpha(x,u) = \begin{cases} u-x & \text{ha } x \le u, \\ \alpha(x-u) & \text{ha } x > u. \end{cases}$$

2. A szigorú célfüggvény:

$$S(x,u) = \begin{cases} u-x & \text{ha } x \le u, \\ u & \text{ha } x > u. \end{cases}$$

Az algoritmus teljes költsége

$$\textstyle\sum_t c(x_t,u).$$

A vizsgált költség

$$KOLTSEG = \max_u \textstyle\sum_t c(x_t,u)$$

Egy algoritmust megadhatunk az $[i,j]$ intervallumok halmazán értelmezett olyan függvényként, amelyre $i < A(i,j) \le j$.

Feltesszük, hogy van egy ismert $n \ge u$ korlátunk.

BIN Algoritmus: $\mathrm{B}(i,j) = \lceil \frac{j+i}{2} \rceil$

$KOLTSEG_{BIN} = \frac{n \log_2(n)}{2} + O(n)$

TCP Algoritmus: $TCP(1,j) = \lceil j/2 \rceil$, és $TCP(i,j) = (i+1)$ ha $i > 1$.

$KOLTSEG_{TCP} = \frac{n^2}{8} + O(n)$

SHRINK algoritmus:

1. Fázis: csökkentsük a lehetséges intervallumot $[2^{t-1}+1, j]$ formára, ahol $j \le 2^t$. Azaz, ha $j > 2^t$, akkor $SHRINK(2^{t-1}, j) = 2^l + 1$, ahol l maximális.

Ha $2^{t-1}+1 \le i < j \le 2^t$, és m a legnagyobb egész, amelyre $j - i < 2^t/2^{2^m}$, akkor

$SHRINK(i,j) = i + \max(1, 2^t/2^{2^{m+1}})$.
Ezt a módszert használva a 2^{t-2^i} méretű intervallum $2^{t-2^{i+1}}$ méretre csökkenthető $O(2^t)$ költséggel.

Tehát $KOLTSEG_{SHRINK} = O(n \log\log n)$.

Dinamikus modell:

- Ekkor a megengedett sávszélesség nem konstans, hanem egy u_t sorozat, amelyet egy ellenfél generál.

- Ebben a modellben mindig elérhető, hogy az optimális költség 0 legyen, míg az algoritmus költsége pozitív, így a kompetitív analízis használásához áttérünk a nyereség vizsgálatára.

- Amennyiben az algoritmus t-edik probálkozása $\{x_t\}$, akkor az algoritmus nyeresége

$$g(x_t, u_t) = \begin{cases} x_t, & \text{ha } x_t \leq u_t, \\ 0 & \text{ha } x_t > u_t. \end{cases}$$

A teljes nyereség:

$$A_j(\{u_i\}) = \sum_{t=1}^{j} g(x_t, u_t).$$

Az offline teljes nyereség:

$$OFF_j(\{u_i\}) = \sum_{t=1}^{j} u_t.$$

Az algoritmust r kompetitívnek nevezzük, ha van olyan b, konstans amelyre

$$rA_j(\{u_i\}) \geq OFF_j(\{u_i\}) + b,$$

minden j-re és $\{u_i\}$ sorozatra.

Az ellenfél korlátozásai: Amennyiben az ellenfél teljesen szabadon választhatja meg az u értékeket az algoritmus nem lehet kompetitív, így korlátoznunk kell a lehetséges sorozatokat (a valós alkalmazásokban sem változhat tetszőlegesen a sávszélesség). A továbbiakban feltesszük, hogy adott egy $[a, b]$ intervallum a pozitív félegyenesen, és az $u_i \in [a, b]$ feltételnek minden i esetén teljesülnie kell.

Determinisztikus eset: Amennyiben az algoritmus nem $x_t = a$-t választja az ellenfél egy $u_t < x_t$ értéket ad meg, így az algoritmus nyeresége 0. Tehát az optimális algoritmusnak mindig $x_t = a$ értéket kell választania, így az ellenfél $u_t = b$-t választ, és az optimális algoritmus b/a-kompetitív.

Véletlenített algoritmusok

Def: Egy maximalizálási probléma esetén egy A véletlenített algoritmusnak egy C szám legrosszabb eset korlátja (véletlenített online algoritmus C-kompetitív), ha $C \cdot E(A(I)) \geq OFF(I)$ minden I inputra.

Példa Legyen a problémára két algoritmus A, B, két input I, J. A nyereségek $A(I) = B(J) = 5$, $A(J) = B(I) = 1$. Ekkor mind A, mind B kompetitív hányadosa 5. A véletlenített algoritmusnak, amely $1/2$ valószínűséggel A és $1/2$ valószínűséggel B a kompetitív hányadosa 3.

Tétel: Van olyan véletlenített algoritmus, amely kompetitív hányadosa $1 + \ln(b/a)$.

Biz: Legyen $x_t = a$, $1/r$ valószínűséggel és $x \in (a, b]$ az $\frac{1}{rx}$ sűrűségfüggvény alapján, ahol $r = 1 + \ln(b/a)$.

Ha az ellenfélre $u_t = y$, akkor a várható offline nyereség y. A várható online nyereség pedig

$$N = a/r + \int_a^y \frac{x}{rx} dx = y/r.$$

IRODALOM

[1] Karp, Koutsoupias, Papadimitrou, Shenker, Algorithmic problems in congestion control, FOCS 2000,
http://www.cs.berkeley.edu/~christos/congestion.ps