

Félig online modellek tanácsadói komplexitással

A tanácsadói komplexitás annyiban tér el a tisztán online algoritmustól, hogy itt rendelkezésünkre áll egy információforrás, amely az adott probléma teljes inputját ismeri, és tetszőleges számítási kapacitással rendelkezik, azaz NP-nehéz feladatokat is meg tud oldani.

Attól függően, hogy mennyi információt kapunk javíthatunk a versenyképességen. Szokás azt vizsgálni mennyi információ kell ahhoz, hogy optimális megoldásunk legyen, de azt is vizsgálják kevesebb információ mennyivel csökkenti a lehetséges versenyképességi hányadost.

Például síbérletnél 1 bit elég az 1-versenyképességhez, ami megmondja vásárol-e az optimum vagy bérel.

Ütemezés két gépre

Egy munkát α -nagynak nevezünk, ha a mérete nagyobb α -szor az optimális költségnél, egyébként a munka α -kicsi. A lista algoritmus elemzéséhez hasonlóan látható, hogy ha minden munka α kicsi, akkor a Lista algoritmus $1 + \alpha(1 - 1/m)$ -versenyképes.

Észrevétel $n - 1$ bit tanács mellett könnyen megtaláljuk az optimális megoldást. Az első munkát tetszőleges gépre rakhatjuk, a többire megkapjuk a helyét tanácsként.

Tétel Legalább $n - 2$ tanácsadói bit szükséges ahhoz, hogy egy online algoritmus garantáltan az optimumot adja.

Vegyünk egy inputot, ahol a munkák $1, 2, 4, \dots, 2^{n-2}, d$, ahol d úgy van megválasztva, hogy az első $n - 1$ munka valamely kiosztása mellett egyenlővé teszi a gépek töltését. Az első $n - 1$ munka minden szimmetriától eltekintett különböző kiosztására más a d értéke, és minden d -hez csak az adott kiosztás ad optimális megoldást. Összesen 2^{n-2} különböző ütemezése van az első munkáknak és a tanácsadónak meg kell különböztetni ezeket, különben nem tudunk optimális megoldást találni.

Algoritmus logaritmus sok bittel Ha nincs $2/3$ -nagy munka, akkor használjuk a lista algoritmust és $4/3$ -versenyképesek vagyunk. Ha van $2/3$ -nagy munka, akkor a tanácsadó megadja az indexét, azt a munkát rakjuk az egyik gépre a többit a másikra és $4/3$ -versenyképesek vagyunk.

Ládapakolás

Ládapakolásnál ha minden elemre megadjuk melyik ládába kell rakni, akkor van optimális megoldásunk, ehhez $O(n \log OPT)$ bit kell. Az is igazolható, hogy nagyságrendileg ennyi bit szükséges is (az alsó korlát $\Omega(n - 2OPT) \log OPT$ bit).

A következő algoritmus kevesebb bit segítségével javít az ismert legjobb felső korláton.

Az elemeket méret szerint csoportosítjuk kicsi, közepes és nagy kategóriákba amik a $(0, 1/2]$, $(1/2, 2/3]$ $(2/3, 1]$ intervallumokba eső elemek. A tanácsadó a közepes elemek K számát adja meg, ami legfeljebb $O(\log n)$ bit. Ezen információ alapján a következőképpen pakolunk.

Az algoritmus kijelöl K kritikus ládát és mindegyikben lefoglal $2/3$ virtuális töltést. A többi láda virtuális töltése 0. Ha nagy elem jön, új ládába rakja, az új láda töltése (és egyben virtuális töltése) az elem mérete lesz. Ha közepes elem jön berakjuk valamelyik közepes elemnek foglalt ládába a lefoglalt virtuális töltés által biztosított helyre, a virtuális töltést a ténylegesre változtatjuk. A kicsi elemeket az FF stratégia alapján pakoljuk a virtuális töltést figyelembe véve és aktualizálva.

Tétel Az algoritmus aszimptotikusan $3/2$ -versenyképes.

Gráfszínezés

Az online esetben a gráf csúcsait egyenként kapjuk meg és mindig azt tudjuk meg, hogy az új csúcs az eddigi csúcsok közül kikkel van összekötve. A feladat igen nehéz, már fák esetén sincs jobb, mint $O(\log n)$ -versenyképes algoritmus. Ez binomiális fákkal igazolható. Az alábbiakban páros gráfokkal foglalkozunk.

Könnyen látható, hogy $n - 2$ bit elég az optimális megoldás eléréséhez. Az első csúcs tetszőleges színt kaphat, az utolsó csúcsnál minden információ adott, a többiekre kell megkapnunk az infót.

Valójában, ha tudjuk, hogy páros gráfot kapunk, akkor csak az olyan új csúcsok esetén kell segítséget kapnunk, amik izoláltak. Ha egy csúcs valamely régebbi csúcshoz hozzá van kötve, akkor tudjuk melyik színosztályba esik. Ennek az észrevételnek a következménye, hogy utakat tudunk $n/2$ tanácsadói komplexitással is optimálisan színezni.

Igazolható az is, hogy általában páros gráfokra nincs olyan algoritmus, ami $n - 3$ bitnél kevesebb tanácsadói információval garantálni tudja az optimalitást.