Approximációs algoritmusok

Ebben a részben az egyik legegyszerűbb modellel foglalkozunk. A modellben m darab azonos $M_1, \ldots, M_m$ gépünk van , a $j_1, \ldots, j_n$ munkákhoz pedig csak egyetlen paraméter tartozik, a p_i, $i = 1, \ldots, n$ végrehajtási idő. Cél egy olyan ütemezés megkonstruálása, amelyre a befejezési idők maximuma minimális. ($Pm||C_{\max}$ probléma.)

A probléma egy gép esetén triviális. Bármely olyan ütemezésre, amelyben a gép folyamatosan dolgozik (nem várunk a munkák elvégzése közben) a $\max\{C_j : 1 \leq j \leq n\}$ érték megegyezik a végrehajtási idők összegével. Továbbá az is nyilvánvaló, hogy amennyiben minden munkát elvégzünk, akkor nem fejezhetjük be a munkákat ezen időpont előtt. Ha a gépek száma több, mint 1, akkor a feladat lényegesen nehezebb. Igazolást nyert, hogy $n \geq 2$ esetén a probléma NP-nehéz. Tehát, mint az NP-nehéz problémák esetében általában, erre a feladatosztályra is fontos közelítő algoritmusok kidolgozása.

Approximációs (illetve közelítő) algoritmuson olyan polinomiális idejű eljárásokat értünk, amelyek az optimális megoldásnál bizonyos értelemben véve nem sokkal rosszabb megoldást adnak. Az approximációs algoritmusok hatékonyságát az approximációs hányadossal (szokás legrosszabb eset korlát analízisnek is nevezni) jellemezzük.

Minimumot meghatározó problémák esetén egy A algoritmus C-*approximációs algoritmus*, ha $A(\sigma) \leq C \cdot OPT(\sigma)$ teljesül minden σ input esetén, ahol $A(\sigma)$ a célfüggvény értéke a σ inputon az A algoritmus által szolgáltatott megoldásra, és $OPT(\sigma)$ pedig a σ inputon a feladat optimális célfüggvényértéke. (A C számot szokás az algoritmus *legrosszabb eset korlátjának* is nevezni.) Az A algoritmus approximációs hányadosa a legkisebb olyan C szám, amelyre A C-approximációs algoritmus. Ebben az esetben azt mondjuk a C legrosszabb eset korlát *éles*.

Az alábbiakban két egyszerű közelítő algoritmussal ismerkedünk meg. Az eljárások ismertetése előtt fontos megjegyeznünk, hogy az algoritmusoknak csak az egyes gépekhez kell hozzárendelni az egyes munkákat, ezt követően az egyes gépeken a maximális befejezési idő minden ütemezésre, amelyben nincsennek üres időtartamok megegyezik a géphez rendelt munkák végrehajtási idejeinek összegével. Gépek és munkák egy tetszőleges halmazára azt az értéket, amit úgy kapunk, hogy a munkák végrehajtási idejeinek összegét elosztjuk a gépek számával a munkáknak az adott géphalmazra vonatkozó töltésének nevezzük.

Az első algoritmus, amelyet *LISTA algoritmusnak* nevezünk a következőképpen működik.

LISTA algoritmus ([1])

Előkészítő rész. A j_1 munkát rendeljük az M_1 géphez, továbbá legyen $r := 1$.

Iterációs rész (r-edik iteráció). Ha $r = n$, akkor vége az eljárásnak. Ellenkező esetben a j_{r+1} munkát rendeljük ahhoz a géphez, amelyre a gépen levő töltés minimális. Ha több ilyen gép is van, akkor válasszuk ezek közül a legkisebb indexűt. Növeljük r értékét 1-gyel, és folytassuk az eljárást a következő iterációval.

Az algoritmus az optimálishoz közeli eredményt eredményez, amint azt az alábbi tétel mutatja.

Tétel. *A LISTA algoritmus approximációs hányadosa* $2 - 1/m$*, ahol* m *a gépek száma.*

Bizonyítás: Elsőként igazoljuk, hogy LISTA $2-1/m$ approximációs algoritmus. Legyen $\sigma = \{j_1, \ldots, j_n\}$ tetszőleges munkasorozat rendre $p_1, \ldots, p_n$ végrehajtási időkkel. Tekintsük a LISTA algoritmus által kapott ütemezést. Legyen j_l az a munka, amely legkésőbb fejeződik be. Vizsgáljuk ezen munka S_l kezdési idejét. Mivel egyetlen gép sem kezdte el a munkát ütemezni S_l előtt, ezért minden gép szünet nélkül dolgozott az S_l időpontig. Ebből azt kapjuk, hogy

$$S_l \leq \frac{1}{m} \sum_{\substack{j=1 \\ j \neq l}}^{n} p_j = \frac{1}{m}(\sum_{j=1}^{n} p_j - p_l) = \frac{1}{m}(\sum_{j=1}^{n} p_j) - \frac{1}{m} p_l.$$

Következésképp

$$LISTA(\sigma) = S_l + p_l \leq \frac{1}{m}(\sum_{j=1}^{n} p_j) + \frac{m-1}{m} p_l.$$

Másrészt az optimális ütemezésben is végre kell hajtani az összes munkát, így $OPT(\sigma) \geq \frac{1}{m}(\sum_{j=1}^{n} p_j)$. Továbbá a p_l munkát is végre kell hajtani valamely gépen, így $OPT(\sigma) \geq p_l$. Ezen becslések alapján egyből adódik, hogy

$$LISTA(\sigma) \leq (1 + \frac{m-1}{m})OPT(\sigma),$$

amivel bizonyítottuk, hogy LISTA $2 - 1/m$ approximációs algoritmus.

Most igazoljuk, hogy a kapott korlát éles. Vegyünk $m(m-1)$ darab munkát $1/m$ végrehajtási idővel, majd egy munkát 1 végrehajtási idővel. Ekkor a LISTA algoritmus az első $m(m-1)$ munkát egyenletesen elosztja a gépek között, majd az utolsó munkát az M_1 gépen ütemezi. Tehát a maximális befejezési idő $1 + (m-1)/m$ lesz. Egy optimális ütemezés pedig a rövid munkákat egyenletesen osztja szét az első $m-1$ gép között, majd az utolsó munkát az m-edik géphez rendeli, és a maximális befejezési ideje 1 lesz. Tehát ebben az esetben az algoritmus által kapott megoldás és az optimális megoldás célfüggvényértékeinek hányadosa $2 - 1/m$, amivel igazoltuk az állításunkat. $\square$

Amint azt láthattuk a korlát élességének igazolásánál, a lista algoritmus legnagyobb hibája akkor jön elő, amikor sok rövid munka után egy hosszú munkát kell végrehajtani. Ezt a problémát jobban kezeli a következő eljárás.

LPT Algoritmus ([2])

- *1. lépés.* Rendezzük sorba a munkákat csökkenő végrehajtási idő szerint.
- 2. lépés. A munkák kapott listáján hajtsuk végre a LISTA algoritmust.

Az első rendezési fázis valóban javítja az eljárás hatékonyságát, amint azt a következő állítás mutatja.

Tétel. *Az LPT $4/3 - 1/(3m)$-approximációs algoritmus.*

Bizonyítás: Az állítást indirekt igazoljuk. Ehhez tegyük fel, hogy léteznek olyan ellenpéldák, amelyekre az optimális ütemezés és az LPT algoritmus által kapott ütemezés költségeinek hányadosa nagyobb mint $4/3 - 1/(3m)$. Ezen ellenpéldák közül van olyan, amely minimális számú munkát tartalmaz. Tekintsünk egy ilyen ellenpéldát! Jelölje n a munkák számát, és σ a munkák listáját.

Mivel a tekintett ellenpéldában a munkák száma minimális, ezért az utolsónak ütemezett munka befejezési ideje megegyezik a maximális befejezési idővel. Amennyiben ez a tulajdonság nem teljesülne, akkor arra a σ' inputra, amelyet a tekintett ellenpéldából az utolsó munkát elhagyva kapnánk $LPT(\sigma') = LPT(\sigma)$ és $OPT(\sigma) \geq OPT(\sigma')$ teljesülne. Következésképp egy olyan ellenpéldát kapnánk, amely kevesebb munkát tartalmaz mint σ, ami ellentmond σ minimalitásának.

A fentiek alapján azt kapjuk, hogy az utolsó munka kezdési ideje $LPT(\sigma) - p_n$. Másrészt eddig az időpontig az összes gép dolgozott, így

$$LPT(\sigma) - p_n \leq \frac{\sum_{i=1}^{n-1} p_i}{m}.$$

Másrészt az optimális ütemezésben is végre kell hajtani hajtani az összes munkát, így $OPT(\sigma) \geq \frac{1}{m}(\sum_{j=1}^{n} p_j)$.

Következésképp a következő egyenlőtlenségek teljesülnek a σ ellenpédára.

$$\frac{4}{3} - \frac{1}{3m} < \frac{LPT(\sigma)}{OPT(\sigma)} \leq \frac{p_n + \sum_{i=1}^{n-1} p_i/m}{OPT(\sigma)} =$$

$$\frac{p_n(1-1/m)}{OPT(\sigma)} + \frac{\sum_{i=1}^{n} p_i/m}{OPT(\sigma)} \leq \frac{p_n(1-1/m)}{OPT(\sigma)} + 1.$$

A fenti egyenlőtlenség jobb és baloldalát véve a következő korlát adódik $OPT(\sigma)$ értékére:

$$OPT(\sigma) < 3p_n.$$

Mivel p_n a minimális végrehajtási idő, ezért a fenti egyenlőtlenség azt jelenti, hogy az optimális ütemezésben minden gép legfeljebb két munkát tartalmaz, azaz $n \leq 2m$. Másrészt ebben az esetben az ütemezési probléma a következő feladattá redukálódik. Rendezzünk $2m$ számot párokba úgy, hogy a párokba tartozó számok összegeinek maximuma minimális legyen. Ezen feladat megoldása az, ha az i-edik legnagyobb számot az i-edik legkisebb számmal párosítjuk, amely pontosan az LPT ütemezési eljárásnak felel meg. Következésképp azt kaptuk, hogy az LPT eljárás a σ ellenpéldán az optimális ütemezést adja, azaz ellentmondáshoz jutottunk, amivel a tétel állítását igazoltuk. Most igazoljuk, hogy a korlát éles. Legyen m a gépek száma és vegyük azt a σ_m munkasorozatot, amely $2m+1$ munkából áll, amelyekből 3 darab munkának a végrehajtási ideje m, továbbá az $i = m+1, \ldots, 2m-1$

értékekre rendre 2 darab munka van, amelynek a végrehajtási ideje i. Ezen sorozatra az optimális megoldás a 3 darab m hosszú munkát egy gépen ütemezi, a többieket párosítja $m-1$ párba úgy, hogy minden párban $3m$ legyen az összeg. Így $OPT(\sigma_m) = 3m$. Másrészt LPT az első $2m$ munkát párosítja a gépeken úgy, hogy mindenhol $3m-1$ lesz a töltés, majd az utolsó m hosszú munkát az első gépen ütemezi. Tehát $LPT(\sigma_m) = 4m - 1$, amivel a korlát élességét igazoltuk.

$\square$

Hivatkozások

[1] R. L. Graham , Bounds for certain multiprocessor anomalies, *Bell System Technical Journal* **45** 1966, 1563–1581.

[2] R. L. Graham, Bounds on Multiprocessing Timing Anomalies, *SIAM Journal on Applied Mathematics* **17**, 1969, 263–269.