5. Gyakorlat

F1 Adjunk dinamikus programozási megoldást arra a három gépes feladatra egész ütemezési idők mellett, ahol a célfüggvény a gépekhez rendelt töltésekből álló vektor l_p normája.

megold: Miként a többi feladatban itt is a dinamikus programozási eljárásban $F_i(x,y)$ $(x, y \leq \sum p_i)$ azt jelöli, mi az a minimális töltés, amelyet elérhetünk az első i munka ütemezése után a harmadik gépen ha az első két gépen a töltés legfeljebb x és y. Az F függvény értéke ∞, ha valamelyik változó negatív. Az $F_1(x,y)$ értéke 0, ha az első munkát tudjuk ütemezni a gépeken, azaz ha $p_1 \leq x$ vagy $p_1 \leq y$ egyébként a függvény a p_1 értéket veszi fel. A többi értéket a következő rekurzió alapján kaphatjuk meg, az esetek attól függnek melyik gépen ütemezzük a munkákat.

$$F_{i+1}(x,y) = \min\{F_i(x - p_{i+1}, y), F_i(x, y - p_{i+1}), F_i(x,y) + p_{i+1}\}$$

Kiszámolva F_n értékeit olyan elemet kell keresni, amelyre a koordinátákból és a függvényértékből álló vektor l_p normája minimális. Ha megjegyezzük az optimális döntéseket, akkor az optimális ütemezés visszafejthető.

F2 Vegyük azt a modellt, ahol két független gép van, és a munkákat vissza lehet utasítani, a célfüggvény a töltések maximumának és a teljes büntetés összegének minimalizálása.

megold: A dinamikus programozási eljárásban $F_i(x,y)$ $(x, y \leq \sum p_i)$ azt jelöli, mi az a minimális teljes büntetés, amelyet elérhetünk az első i munka ütemezése után, ha az első két gépen a töltés legfeljebb x és y. Az F függvény értéke ∞, ha valamelyik változó negatív. Az $F_1(x,y)$ értéke 0, ha az első munkát tudjuk ütemezni a gépeken, azaz ha $p_{11} \leq x$ vagy $p_{12} \leq y$ egyébként a függvény a b_1 értéket veszi fel. A többi értéket a következő rekurzió alapján kaphatjuk meg, az esetek attól függnek melyik gépen ütemezzük vagy visszautasítjuk a munkákat.

$$F_{i+1}(x,y) = \min\{F_i(x - p_{i+1,1}, y), F_i(x, y - p_{i+1,2}), F_i(x,y) + b_{i+1}\}$$

Kiszámolva F_n értékeit olyan elemet kell keresni, amelyre a koordináták maximumához hozzáadva a függvényértéket minimális értéket kapunk. Ez az érték adja meg az optimális célfüggvényt. Ha minden lépésben megjegyezzük, az optimális döntést (miként a fenti példában a zárójelekben), akkor a megoldás is visszakereshető.

P1 Tekintsük az általános párhuzamos gépek esetét, ahol 2 darab gép van. Vegyük a következő $I = ((1,2), (1,2), (1,3), (1,3))$ munkasorozatot, ahol a munkák a végrehajtási idővektorok által vannak megadva. Hajtsuk végre a mohó algoritmust.

megold: Ekkor az első munka az 1 időpontban fejezhető be az M_1 gépen és a 2 időpontban a második gépen, így M_1-hez rendeljük. A következő munka a

2 időpontra fejezhető be mindkét gépen, így M_1 kapja. A következő munka a 3 időpontra fejezhető be mindkét gépen, így ismét M_1 kapja. Végül az utolsó munka a 4 időpontra fejezhető be az M_1 gépen és a 3 időpontra fejezhető be az M_2 gépen, így az M_2 géphez rendeljük.

P2 Tekintsük a hasonló párhuzamos gépek esetét, ahol 3 darab gép van és a sebességek $s_1 = s_2 = 1$, $s_2 = 3$. Vegyük a következő $I = (2, 1, 1, 3, 2)$ munkasorozatot, ahol a munkák a végrehajtási idők által vannak megadva. Hajtsuk végre a mohó algoritmust.

megold: Ekkor az első munka 2/3-kor fejezhető be az M_3 gépen és az 1 időpontban a többi gépen, így M_3-hoz rendeljük. A következő munka az 1 időpontra fejezhető be az összes gépen, így M_3 kapja, mivel ott a legkisebb a megmunkálási idő. A következő munka az 1 időpontra fejezhető be M_1-en és M_2-n, és a 4/3 időpontban az M_3 gépen, így M_1 kapja. A negyedik munka M_1-en a 4, M_2-n a 3 időpontban fejeződne be, az M_3-on a 2 időpontban, így M_3-ra kerül. Végül az utolsó munka M_1-en 3, M_2-n a 2 időpontban fejeződne be, az M_3-on 8/3, így M_2-re kerül.

P3 Tekintsük két párhuzamos gép esetét az online idő modellben. Legyen a munkák sorozata a következő $I = (1, 0), (1/2, 0), (1/2, 0), (1, 3/2), (1, 3/2), (2, 2)$, ahol a munkákat a (végrehajtási idő, érkezési idő) párral adtuk meg. hajtsuk végre az Intervallum algoritmust.

megold: Az első iterációs részben az első három munkát ütemezzük, egy optimális ütemezés az első munkát az első géphez, a második és harmadik munkákat a második géphez rendeli, és az 1 időpontban befejezi a munkák végrehajtását. Utána, mivel nem érkezett új munka és nincs vége a sorozatnak várunk, egészen a 3/2 időpontig. Ekkor egy új iterációs rész kezdődik, az algoritmus ütemezi a negyedik és ötödik munkákat az első és második gépen, a munkákat az 5/2 időpontra fejezi be. Ekkor elkezdődik a harmadik iterációs rész, és az algoritmus ütemezi a hatodik munkát valamely gépen, az $[5/2, 9/2]$ intervallumra.

F3. Igazoljuk, hogy ha az intervallum algoritmusban ez egyes fázisokban az optimális NP-nehéz probléma megoldása helyett egy c-approximációs algoritmust használunk, akkor az algoritmus $2c$-versenyképes.

megold: Az algoritmus elemzése során használt $T_{VE} \leq OPT$ illetve $T_K + T_V \leq OPT$ becslések helyett használhatjuk a $T_{VE} \leq c \cdot OPT$ és $T_K + T_V \leq c \cdot OPT$ becsléseket és az állítás adódik.