4. Gyakorlat

P1. Vegyük az $P3||C_{max}$ problémát, ahol a munkák végrehajtási időkkel megadva $j_1 = 1, j_2 = 1, j_3 = 2, j_4 = 2$. Oldjuk meg a feladatot dinamikus programozási eljárással.

Megold: 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 $F_1(x, y)$ értéke 0, ha az első munka kisebb, mint $\max\{x, y\}$ és p_1 egyébként. Így ebben az esetben $F_1(0, 0) = 1$ és $F_1(x, y) = 0$ a többi esetben. A további értékek az $F_{i+1}(x, y) = \min\{F_i(x - p_{i+1}, y), F_i(x, y - p_{i+1}), F_i(x, y) + p_{i+1}\}$ rekurzióval határozhatóak meg. A rekurzióban a döntés, hogy melyik gépen ütemezzük az $i + 1$-edik munkát. Az értékeket a következő mátrixok tartalmazzák, a zárójelben mindenhol megjegyeztük, milyen érték mellett kaptuk a minimumot, hogy az optimális ütemezést megtalálhassuk visszafejtéssel. Ha több választás is minimális értéket ad, a legelső gépet választjuk közülük.

$$F_2 = \begin{pmatrix} 2(3) & 1(2) & 0(2) & 0(2) & 0(2) & 0(2) & 0(2) \\ 1(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) \\ 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) \\ 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) \\ 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) \\ 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) \\ 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) \end{pmatrix}$$

$$F_3 = \begin{pmatrix} 4(3) & 3(3) & 2(2) & 1(2) & 0(2) & 0(2) & 0(2) \\ 3(3) & 2(3) & 1(2) & 0(2) & 0(2) & 0(2) & 0(2) \\ 2(1) & 1(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) \\ 1(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) \\ 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) \\ 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) \\ 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) \end{pmatrix}$$

$$F_4 = \begin{pmatrix} 6(3) & 5(3) & 4(2) & 3(2) & 2(2) & 1(2) & 0(2) \\ 5(3) & 4(3) & 3(2) & 2(2) & 1(2) & 0(2) & 0(2) \\ 4(1) & 3(1) & 2(1) & 1(1) & 0(1) & 0(1) & 0(1) \\ 3(1) & 2(1) & 1(1) & 0(1) & 0(1) & 0(1) & 0(1) \\ 2(1) & 1(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) \\ 1(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) \\ 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) & 0(1) \end{pmatrix}$$

Az utolsó mátrixban kell olyan értéket keresni, amelyre a maximuma a koordinátáknak és a függvényértéknek minimális. Egy ilyen érték van az $F_4(2, 2)$. A hozzá tartozó megoldás az utolsó munkát a zárójelben látható első gépen ütemezi. A maradék probléma $F_3(0, 2)$-ben látható, itt az optimális döntés a harmadik gépen ütemezi p_3-at. A következő részfeladat $F_2(0, 2)$, itt a zárójel alapján tudjuk, hogy p_2-t a második gépen kell ütemezni, végül $F_1(0, 1)$ megoldásánál szintén a második gépen a p_1 munkát.

F1. Adjunk dinamikus programozási megoldást a $R3||C_{max}$ feladatra egész ütemezési idők mellett.

Megold: 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 $p_1/v_1 \leq x$ vagy $p_1/v_2 \leq y$ egyébként a függvény a p_1/v_3 értéket veszi fel. A többi értéket a következő rekurzió alapján kaphatjuk meg, az esetek attól függnek melyik gépen ütemezzük a munkákat.

$$F_{i+1}(x,y) = \min\{F_i(x - p_{i+1}/v_1, y), F_i(x, y - p_{i+1}/v_2), F_i(x,y) + p_{i+1}/v_3\}$$

Kiszámolva F_n értékeit olyan elemet kell keresni, amelyre a koordináták és a függvényérték maximuma minimális. 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.

F2. Vegyük azt az ütemezési feladatot, ahol a munkákat vissza lehet utasítani. Két gép van, a munkáknak van egy végrehajtási ideje és egy büntetés, amit akkor kapunk ha visszautasítjuk. A cél az elfogadott munkák ütemezésében a maximális befejezési idő és a visszautasított munkákért kapott 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 $p_1 \leq x$ vagy $p_1 \leq y$, 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.

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

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.