2. Gyakorlat

P1. Oldjuk meg a következő $1||\sum w_j C_j$ feladatot, ahol a munkák a (p_j, w_j) párokkal vannak megadva: $j_1 = (3,2), j_2 = (4,1), j_3 = (1,1), j_4 = (2,3)$.

Megold: A feladatot a WSPT elv oldja meg. Az egységenkénti fontosság (w/p) szerinti sorrend j_4, j_3, j_1, j_2, így ez lesz a gépen a munkák sorrendje. j_4 a $(0,2)$ j_3 a $(2,3)$ j_1 a $(3,6)$ és j_2 a $(6,10)$ időintervallumban lesz ütemezve.

P2. Oldjuk meg a következő $3||\sum C_j$ feladatot, ahol a munkák a (p_j) értékkel vannak megadva: $j_1 = 1, j_2 = 2, j_3 = 1, j_4 = 2, j_5 = 4$.

Megold: A feladatot a MSPT elv oldja meg. Az m osztója n feltétel teljesüléséhez, kell egy 0 végrehajtási idejű j_6 munka. Utána a végrehajtási idő szerinti sorrend $j_6, j_1, j_3, j_2, j_4, j_5$, így a fiktív j_6 és j_2 kerül az első gépre utóbbi a $(0,2)$ időintervallumban, a második gépre j_1 a $(0,1)$ és j_4 az $(1,3)$ időintevallumokban végül j_3 és j_5 a $(0,1)$ és $(1,5)$ időintervallumokban a harmadik gépen lesz ütemezve.

F1. Definiáljuk a MWSPT elvet $Pm||\sum w_j C_j$ feladatra és igazoljuk, hogy nem minden esetben optimális.

Megold: Miként a MSPT elvnél, a módosított WSPT elvnél is feltesszük, hogy a gépek száma osztója a munkák számának (ez elérhető fiktiv munkák felvételével). Utána rendezzük a munkákat az egységenkénti fontosság szerint csökkenő sorrendbe. Az első m munka lesz a gépeken elsőként végrehajtva, utána a második m munka másodikként az egyes gépeken, és így tovább az utolsó m munka lesz az n/m-edik munka a gépeken.

Az algoritmus nem ad mindig optimális megoldást. Legyen 2 gép és 3 munka, a munkák a (p_j, w_j) párokkal vannak megadva $j_1 = (1,1), j_2 = (1,1), j_3 = (3, 3-\varepsilon)$, ahol ε egy kicsi pozitív szám. Ekkor egy fiktív 0 végrehajtási idejű munkát felvéve, a munkák sorrendje j_4, j_1, j_2, j_3. Így a fiktív j_4 és j_2 kerül az első gépre utóbbi a $(0,1)$ időintervallumban, a második gépre j_1 a $(0,1)$ és j_3 az $(1,4)$ időintervallumokban. Az ütemezés költsége $1+1+4(3-\varepsilon) = 14-4\varepsilon$. Az optimális megoldás j_1-et és j_2-t ütemezi az első gépen, j_3-at a második gépen, és a költsége $12-3\varepsilon$.

P3. Mi az optimális megoldási elv az általános $1||h_{max}$ probléma esetén, ha $h_j = T_j$, illetve ha $h_j = T_j + L_j$.

Megold: Az algoritmus minden iterációs lépésben olyan munkát ütemez hátulra, amelyre a h_j függvény a befejezési időre minimális lesz. A függvényérték mindkét esetben arra a munkára lesz minimális, amelyre a határidő maximális. Tehát azok a munkák kerülnek hátra akiknek maximális a határideje, így az optimális algoritmus a minimális határidő elve.

P4. Vegyük az $1||h_{max}$ problémát, ahol $h_1(x) = x^2$, $h_2(x) = \min\{6, x\}$, $h_3(x) = 2x$, továbbá legyenek a munkák a végrehajtási idővel megadva $p_1 = 2, p_2 = 4, p_3 = 4$. Határozzuk meg az általános algoritmus alapján az optimális megoldást.

Megold: Utolsónak azt a munkát ütemezzük, amelyre $h_i(p_1 + p_2 + p_3) = h_i(10)$ minimális, $h_1(10) = 100, h_2(10) = 6, h_3(10) = 20$ tehát utolsónak a j_2 munkát ütemezzük. Ezt követően a maradékból azt a munkát ütemezzük, amelyre $h_i(p_1 + p_3) = h_i(6)$ minimális, $h_1(6) = 36, h_3(6) = 12$ tehát utolsó előttiként a j_3 munkát ütemezzük. Végül elsőként a j_1 munkát.

F2. Adjuk meg az általános $1||h_{max}$ algoritmusnak egy olyan kiterjesztését, ami abban ez esetben is megoldja a feladatot, ha vannak precedencia feltételek.

Megold: Az alábbiakban bemutatott algoritmus hátulról, az utolsónak végrehajtott munka felől építi fel az ütemezést. J mindig a már elhelyezett munkák halmaza, J^c azon munkák halmaza, amelyeket még el kell helyeznünk (a J halmaz komplementere) és J' azon J^C-beli munkák halmaza, amelyek az adott lépésben ütemezhetők, azaz azon munkáké, amelyekre az összes precedencia feltételek szerinti rákövetkező munkát már elhelyeztük az ütemezés végén.

Algoritmus

- *1. Lépés:* Legyen $J = \emptyset$, $J^C = \{1, \ldots, n\}$, és J' azon munkák halmaza, amelyekre nincs rákövetkező munka a precedencia gráfban.
- *2. Lépés:* Legyen $j^* \in J'$ olyan munka, amelyre

  $$h_{j^*}(\sum_{j \in J^C} p_j) = \min_{i \in J'}(h_i(\sum_{j \in J^C} p_j)).$$

  Egészítsük ki J-t j^*-al, töröljük j^*-ot a J^C halmazból. Legyen J' az új halmaza az ütemezhető munkáknak.
- *3. Lépés:* Ha $J^C = \emptyset$ az algoritmus végetért, egyébként folytassuk a 2. lépéssel.