Alkalmazott Matematikai Lapok **39** (2022), 45–58.
DOI: 10.37070/AML.2022.39.1.02

PÁROS MUNKÁK ÜTEMEZÉSE KORAI ÉS KÉSŐI BEFEJEZÉS BÜNTETÉSÉVEL – EGY BONYOLULTSÁGELMÉLETI EREDMÉNY

BÉKÉSI JÓZSEF, DÓSA GYÖRGY, GALAMBOS GÁBOR

Az egy művelettel rendelkező, egygépes ütemezési feladatokat széles körben elemezték. Kimutatták, hogy azok a feladatok, amelyekben a célfüggvényt a munkák elvárt befejezési idejéhez viszonyított korai ill. késői ütemezés költsége határozza meg (ET-feladat) bonyolultságelméleti szempontból sok esetben másképpen viselkednek, mint a „klasszikus" – a legkésőbbi befejezési időt minimalizáló – feladatok. Wan és Yuan [20]-ban bebizonyította, hogy általános esetben az ET-feladat erősen $\mathcal{NP}$-nehéz.

Az egy művelettel rendelkező, egygépes ET-feladat felfogható a kétműveletes probléma (CTP-feladat) speciális eseteként. Ebben az esetben a második művelet hossza 0, és a várakozási idő is 0. Így a páros munkák ütemezéséhez tartozó minden olyan ET-feladat is erősen $\mathcal{NP}$-nehéz, amelyben az egyes tevékenységekhez tartozó második művelet hossza konstans, és megegyezik a két művelet közötti várakozási idővel. Ebben a cikkben a korábbiaknál egyszerűbb bizonyítást adunk arra, hogy az ilyen feladat $\mathcal{NP}$-nehéz.

1. Bevezetés

A cikk páros munkák ütemezésével (*CTP feladat*) foglalkozik. A feladatot ebben a formában először Sherali és Smith definiálta a [17] cikkben: adott kétfázisú munkák egy $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ halmaza. Minden J_i munka két műveletből áll, amelyeknek végrehajtási ideje rendre a_i és b_i. Az ütemezést egy gépen kell végrehajtani úgy, hogy a műveletek nem megszakíthatók. Az általunk vizsgált feladatban a második művelet kezdési ideje és az első művelet befejezési ideje között pontosan L_i várakozási időnek kell eltelni. Egy munka akkor van befejezve, ha mindkét műveletét elvégeztük. A futtatáshoz használt gép bármely időintervallumban maximum egy műveletet képes végrehajtani. A feladat az, hogy ütemezzük úgy a munkákat az adott gépen, hogy egy előre adott célfüggvény szélsőértékét megkapjuk.

Az elmúlt évtizedekben többnyire olyan feladatokat vizsgáltak, amelyeknek a célfüggvénye a maximális végrehajtási idő minimalizálása volt. Ezt a problémát először Shapiro vizsgálta [16]-ban, később Orman és Potts [14]-ben elemezte a problémát bonyolultságelméleti szempontból. A vizsgált feladatokat egy irányított gráfban ábrázolták, ahol a gráf csúcspontjaiban a speciális esetek helyezkednek el, és irányított él mutat egy csúcsból egy másik csúcsba akkor, ha végpont a kiinduló ponthoz tartozó speciális eset egyfajta általánosításával kapható meg. Ebben a bonyolultsági gráfban az erősen $\mathcal{NP}$-nehéz osztályba sorolt esetek jól elkülönülnek a polinomiális időben megoldható problémáktól. Az ún. identikus munkák – $1|(a, L, b)|C_{\max}$ – ütemezésének algoritmikus bonyolultsága nyitva maradt. Ennek a feladatnak a megoldására Ahr és mtsai [3]-ban javasoltak egy olyan dinamikus programozásra alapuló gráf modellt a megoldásra, amelynek bonyolultsága $O(nr^{2L})$, ahol $r = \sqrt[a-1]{a}$. Később, Baptiste megmutatta, hogy a probléma megoldására adható $O(\log n)$ bonyolultságú algoritmus, ha a, L és b rögzített (lásd [5]). Ismereteink szerint az identikus műveletek ütemezésének a bonyolultsága továbbra is nyitott maradt. Az $1|(a_i, L_i, b_i)|C_{\max}$ feladat közelítő megoldására több algoritmust is elemeztek (lásd pl. [2], [13]). Yu és mtsai [19]-ben bebizonyították, hogy az $1|(1, L_j, 1)|C_{\max}$ feladat erősen $\mathcal{NP}$-nehéz. Erre a feladatra adtak egy 7/4-es approximációs algoritmust [1]-ben és [6]-ban. Ha a munkák ütemezésének sorrendje kötött, akkor a feladat bonyolultsága nem minden esetben egyszerűsödik (lásd pl. [15]). (Egy kötött sorrendű ütemezésben ha a J_i job megelőzi a J_j jobot, akkor mind az a_i, mind a b_i művelet az a_j ill. b_j művelet előtt kerül végrehajtásra.) Erre a speciális esetre Blazewicz és mtsai [7]-ben az $1|(a_j, L_j, b_j), \text{fjs}|C_{\max}$ jelölést használták, ahol az „fjs" a második mezőben mutatja a fixed-job-sequence feltételt. Ugyanebben a cikkben a szerzők vizsgálták az $1|(1, L, 1), prec|C_{\max}$ feladatot, amelyben a munkák sorrendjére kötött precedencia szabályok vonatkoznak, és bebizonyították, hogy már ebben a nagyon speciális esetben is a feladat erősen $\mathcal{NP}$-teljes marad, de az $L = 2$ esetre egy $O(n)$ idejű algoritmust fejlesztettek. A fixed-job-sequence feladatot Hwang és Lin is vizsgálta [11]-ben, és – más speciális esetek mellett – polinomiális idejű algoritmust adtak a $1|(a_j, p, p), prec|C_{\max}$ feladatra is.

Egészen a közelmúltig más célfüggvénnyel rendelkező *CTP* feladatot bonyolultságelméleti szempontból nem vizsgáltak. A közelmúltban Chen és Zhang [8]-ban átfogóan elemezte azt a *CTP* feladatot, amelyben a célfüggvény a végrehajtási idők összegének a minimalizálása, $1|(a_i, L_i, b_i)|\sum C_i$. A [8] cikkben a szerzők feltérképezték a teljes bonyolultsági gráfot, és minden esetben megadták a vizsgált esetek algoritmikus bonyolultságát. Több más speciális eset mellett a $1|(a_i, p, p)|\sum C_i$ problémáról megmutatták, hogy az polinomiális időben megoldható.

Ebben a cikkben a következő feladatot fogjuk vizsgálni. Adott egy *CTP* feladat egzakt várakozási idővel. A J_i munkához adva van egy d_i előírt befejezési idő. Jelöljük a J_i munka σ ütemezésen belüli befejezési idejét $C_i(\sigma)$-val, és legyen a munkához tartozó korai ill. késői befejezés költsége $E_i(\sigma) = \max\{0, d_i - C_i(\sigma)\}$ és $T_i(\sigma) = \max\{0, C_i(\sigma) - d_i\}$. Ekkor a $\mathcal{J}$ példa költsége a σ ütemezésben

$$cost(\sigma, \mathcal{J}) = \sum_{i=1}^{n} \big(E_i(\sigma) + T_i(\sigma)\big).$$

A feladatot *ET-feladatnak* nevezzük.

Az ET-feladat gyakorlati alkalmazása az ún. „just-in-time" gyártással függ össze. Ennek során a korai és a késedelmes gyártás egyaránt kiemelten van kezelve, mert mindkettő költséggel jár akkor, ha a gyártás az elvárt rendelkezésre állási – szállítási, lejárati, befejezési – időnél korábban vagy későbben történik. Az első esetben raktározási költségek merülnek fel, az utóbbi esetnél pedig kötbér fizetési kötelezettség léphet fel. Ezért ideális esetben egy jó ütemezés akkor történik, ha az áru (munka) akkorra készül el (akkor lesz befejezve), amikor annak éppen rendelkezésre kell állnia (az eredményre szükség van).

A „klasszikus" – egy művelettel rendelkező – egygépes korai-, ill. késői ütemezési feladatokra Baker és Scudder [4]-ben adott egy áttekintést. Az első publikált eredmények általában a legkorábbi gyártás ill. a legkésőbbi elkészülés minimalizálását tűzték ki célul. Sidney [18]-ban adott egy hatékony algoritmust az optimális ütemezésre. Lakshminarayan és mtsai [12]-ben egy $O(n \log n)$ idejű javítást adtak az optimális megoldásra. Garey és mtsai [9]-ben bebizonyították, hogy az $1||\sum(E_j + T_j)$ feladat az általános értelemben $\mathcal{NP}$-nehéz. Hall [10] azt látta be, hogy a feladat $\mathcal{NP}$-nehéz marad akkor is, ha az összes munkának azonos a lejárati ideje. Wan és Yuan [20]-ban bebizonyította, hogy általános esetben a $1||\sum(E_j + T_j)$ feladat erősen $\mathcal{NP}$-nehéz.

Ha a *CTP* feladatot vizsgáljuk, akkor az látható, hogy ha minden munka közös elvárt befejezési idővel rendelkezik, és bármely $\mathcal{J}$ példa esetén a közös elvárt befejezési idő értékére $d(\mathcal{J}) = 0$, akkor az ET-feladat megegyezik a [8]-ban vizsgált problémával. Ha a munkákhoz egymástól független előírt befejezési időt rendelhetünk, akkor a speciális feladatok bonyolultságelméleti szempontból másképpen viselkedhetnek. Azt fentebb láttuk, hogy az (a_j, p, p) paraméterekkel rendelkező feladat különböző célfüggvények esetén polinomiális megoldással rendelkeznek.

Az egy művelettel rendelkező, egygépes ET-feladat felfogható a kétműveletes probléma speciális eseteként. Ebben az esetben a második művelet hossza 0, és a várakozási idő is 0, azaz a feladat a $1|(a_j, 0, 0)|\sum(E_j + T_j)$ jelöléssel írható le. Így az $1|(a_j, p, p)|\sum(E_j + T_j)$ *CTP*-feladat is erősen $\mathcal{NP}$-nehéz. Ebben a cikkben az általánosabb feladat – $1|(a_j, p, p)|\sum(E_j + T_j)$ – bonyolultságára adunk egy új, egyszerűbb bizonyítást. Azt mutatjuk meg, hogy a *CTP* probléma esetén az ET-feladat algoritmikus bonyolultsága ekvivalens a – közismerten erősen $\mathcal{NP}$-nehéz – 3-partícionálási feladat komplexitásával. A bizonyításban alapvetően a [20]-ban megadott technikát követjük, azonban a konstrukció során definiált ún. „rövid" munkák hosszának eltérő választásával el tudtuk érni, hogy az optimális megoldáshoz tartozó összes eltérés kisebb, és egyszerűbben meghatározható, továbbá a konstrukcióban szereplő konstansok – M_p és M_s – értéke jelentősen csökkenthető, ezáltal a bizonyítás áttekinthetősége is javul.

3-Partícionálási Feladat (3PP). Adva van $3u + 1$ pozitív egész szám: $\alpha_1, \ldots, \alpha_{3u}$ és B, amelyekre érvényes, hogy $\sum_{j=1}^{3u} \alpha_j = uB$ és $\frac{B}{4} < \alpha_j < \frac{B}{2}$,

$1 \leq j \leq 3u$. A kérdés az, hogy létezik-e az $A = \{1, 2, \ldots, 3u\}$ index halmaznak olyan $A_1, A_2, \ldots, A_u$ diszjunkt halmazokra történő felbontása, amelyekre $|A_i| = 3$, és $\sum_{j \in A_i} \alpha_j = B$, ahol $1 \leq i \leq u$.

Az általánosság megszorítása nélkül feltehetjük, hogy egy 3PP feladatban $\alpha_j \geq 3$ minden j-re.

2. A $1|(a_j, p, p)|\sum(E_j + T_j)$ feladat bonyolultsága

Legyen adott egy 3PP feladat az $\alpha_1, \ldots, a_{3u}, B$ konstansokkal. Definiáljuk az

$$M_p = 12u^2 \qquad \text{és} \qquad M_s = M_p B u^2 = 12Bu^4.$$

konstansokat, és konstruáljuk meg a következő $1|(a_j, p, p)|\sum(E_j + T_j)$ ütemezési feladatot, amelyben a munkák halmazát jelölje $\mathcal{J}$. A munkáknak két típusát különböztetjük meg: lesz $3u$ darab particionáló és uM_s rövid (short) munka. Ekkor $\mathcal{J} = \{\mathcal{P}, \mathcal{S}\}$.

- Legyen $\mathcal{P} = \{P_1, P_2, \ldots, P_{3u}\}$ a particionáló munkák halmaza. Ekkor a P_i particionáló munka két műveletének végrehajtási ideje és a várakozási ideje legyen

  $$a_i = M_p \alpha_i - 2 \qquad b_i = 1, \qquad L_i = 1.$$

  Így a P_i munka az $(M_p \alpha_i - 2, 1, 1)$ hármassal írható le. Ezért a P_i munka futásideje $M_p \alpha_i$. Legyen továbbá a munkák közös elvárt befejezési ideje $d(P_i) = 0$.

- Az uM_s darab *rövid* munkát osszuk u blokkba. Jelölje a rövid munkák halmazát $\mathcal{S}$. Ekkor $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_u\}$, ahol $\mathcal{S}_i$ az i. blokk, amely M_s darab elemet tartalmaz, azaz $\mathcal{S}_i = \{S_{i,1}, \ldots, S_{i,M_s}\}$. A rövid munkák műveleteinek végrehajtási ideje legyen $a_i = 2$, $b_i = 1$, és a műveletek közötti várakozási idő $L_i = 1$ minden $S_{i,k} \in \mathcal{S}_i$. Az $S_{i,k}$ munka a $(2, 1, 1)$ hármassal írható le. A definícióból adódik, hogy egy rövid munka végrehajtási ideje 4. Minden rövid munkának eltérő elvárt befejezési ideje van: a $\mathcal{S}_i$ blokkba tartozó k. munkához $(1 \leq k \leq M_s)$ a

  $$d(S_{i,k}) = iM_p B + 4(i-1)M_s + 4k$$

  előírt befejezési idő tartozik.

Ez a konstrukció pszeudo-polinomiális időben elkészíthető, és polinomiális idejűvé válik, ha unáris kódolást alkalmazunk (lásd [20]). A konstrukciónak van néhány egyszerű következménye:

2.1. Következmény. *A munkákon belüli várakozási idő nem használható fel egyetlen művelet számára sem, azaz a munkák nem ütemezhetők „átfedett” módon.*

2.2. Következmény. *Egy particionáló munka végrehajtási ideje* $M_p\alpha_i > \frac{M_pB}{4}$.

A W_0 küszöbérték legyen

$$W_0 = 6M_su(u-1) + \frac{3}{2}u(u+1)M_pB. \tag{1}$$

Azt fogjuk megmutatni, hogy a 3PP feladatnak akkor és csak akkor van megoldása, ha a hozzá tartozó ütemezési feladatnak van olyan lehetséges megoldása, amelyre $cost(\sigma) \leq W_0$.

2.1. Segédtétel. *Ha a 3PP feladatnak van megoldása, akkor a hozzá tartozó ütemezési feladatnak van olyan* σ *lehetséges megoldása, amelyre* $cost(\sigma) \leq W_0$.

Bizonyítás. Ha a 3PP-nek van megoldása, akkor az $A = \{1, 2, \dots, 3u\}$ indexhalmaznak létezik egy olyan $A_1, A_2, \dots, A_u$ diszjunkt halmazokra történő partíciója, amelyekre $|A_i| = 3$ és $\sum_{j\in A_i} \alpha_j = B$ minden i-re, ahol $1 \leq i \leq u$. Legyen a konstruált ütemezési feladatban a particionáló munkák halmaza a következő:

$$\mathcal{P}_{A_i} = \{P_j : j \in A_i\}.$$

Készítsük el azt a σ ütemezést, amelyben a munkákat a 0 időponttól kezdjük el ütemezni holtidő nélkül a

$$\mathcal{P}_{A_1}, \mathcal{S}_1, \mathcal{P}_{A_2}, \mathcal{S}_2, \dots, \mathcal{P}_{A_u}, \mathcal{S}_u$$

sorrendben. A $\mathcal{P}_{A_i}$ halmazban elhelyezkedő egyetlen munkának sem lehet nagyobb a befejezési ideje, mint $4(i-1)M_s + iM_pB$. Egy halmazon belül 3 ilyen munka van, ezért

$$cost(\mathcal{P}_{A_i}, \sigma) \leq 12(i-1)M_s + 3iM_pB. \tag{2}$$

Mivel most a $\mathcal{S}_i$ halmazba eső munkák összköltsége $cost(\mathcal{S}_i, \sigma) = 0$, ha $1 \leq i \leq u$, ezért

$$\begin{aligned} cost(\mathcal{J}, \sigma) &\leq \sum_{i=1}^{u} \Big\{ cost(\mathcal{P}_{A_i}, \sigma) + cost(\mathcal{S}_i, \sigma) \Big\} \\ &\leq \sum_{i=1}^{u} \Big\{ 12(i-1)M_s + 3iM_pB \Big\} \\ &= 6M_su(u-1) + \frac{3}{2}u(u+1)M_pB \\ &= W_0. \end{aligned}$$

□

A következő lemma előtt szükségünk van néhány egyszerű állítás bizonyítására.

2.1. Állítás.

$$\frac{M_p M_s}{2} > 2M_s(3u^2 - 3u + 2) > W_0. \tag{3}$$

Bizonyítás. A bal oldali egyenlőtlenség teljesüléséhez elegendő azt belátni, hogy $M_p > 4(3u^2 - 3u + 2) = 12u^2 - 12u + 8$, ami M_p definíciójából következik.

Vizsgáljuk most a második egyenlőtlenséget. Itt – elvégezve az egyszerűsítéseket – annak kell teljesülni, hogy $8M_s > 3u(u+1)M_pB$. Helyettesítsük be M_s definícióját, és osszunk M_pB-vel. Ekkor azt kapjuk, hogy $8u^2 > 3(u+1)$.

□

Egy lehetséges ütemezést *megfelelőnek* fogunk nevezni, ha $cost(\sigma) \leq W_0$. Egy σ megfelelő ütemezés minimális, ha bármely σ' megfelelő ütemezésre $cost(\sigma) \leq cost(\sigma')$.

2.2. Segédtétel. *Legyen σ egy megfelelő ütemezés. Ekkor az ütemezéshez tartozó 3PP-nek van megoldása.*

Bizonyítás. A bizonyítás első részében azt vizsgáljuk, hogy milyen szabályok teljesülnek a particionáló és a rövid munkák ütemezése során.

2.2. Állítás. *Legyen S_i és S_j két olyan rövid munka, amelyek elvárt befejezési idejére $d_i < d_j$. Ekkor van olyan minimálisan megfelelő ütemezés, amelyben az S_i munka előbb van ütemezve, mint a S_j munka, azaz $s(S_i) < s(S_j)$.*

2.1. Megjegyzés. A 2.2. Állítás megfogalmazásában eltekintettünk a dupla alsó indextől, mert az állítás minden i, j párra igaz függetlenül attól, hogy azonos blokkon belül helyezkednek el, vagy sem.

Bizonyítás. Legyen σ egy minimálisan megfelelő ütemezés. Tegyük fel, hogy az állítás nem igaz, és a σ ütemezésben $d_i < d_j$, de $s(S_i) > s(S_j)$.

Legyen $C_i(\sigma)$ és $C_j(\sigma)$ a két munka befejezési időpontja. Ekkor a σ ütemezésben

$$cost(S_i, S_j, \sigma) = |C_i(\sigma) - d_i| + |C_j(\sigma) - d_j|.$$

Azt tudjuk, hogy $C_j(\sigma) < C_i(\sigma)$ és $d_i < d_j$. Legyen σ' egy olyan ütemezés, amelyben a S_i és S_j munkákat megcseréljük. Ekkor azt kapjuk, hogy

$$\begin{aligned}
cost(S_i, S_j, \sigma') &= |C_i(\sigma') - d_i| + |C_j(\sigma') - d_j| = |C_j(\sigma) - d_i| + |C_i(\sigma) - d_j| \\
&= \max\{C_j(\sigma), d_i\} - \min\{C_j(\sigma), d_i\} \\
&+ \max\{C_i(\sigma), d_j\} - \min\{C_i(\sigma), d_j\} \\
&\leq \max\{C_j(\sigma), d_j\} - \min\{C_i(\sigma), d_i\} \\
&+ \max\{C_i(\sigma), d_i\} - \min\{C_j(\sigma), d_j\} \\
&= |C_i(\sigma) - d_i| + |C_j(\sigma) - d_j| = cost(S_i, S_j, \sigma).
\end{aligned}$$

Így σ' is minimális megfelelő ütemezés.

$\square$

A fenti Állításnak az a következménye, hogy van olyan σ minimálisan megfelelő ütemezés, amelyben a rövid munkák az elvárt befejezési idejük szerint vannak elhelyezve. Osszuk a rövid munkákat blokkokba. Egy blokkba akkor kerül az j. és az $(j+1)$. rövid munka, ha $d(S_{j+1}) = d(S_j) + 4$. Vizsgáljuk ezek után az $\mathcal{S}_i$ – rövid munkákat tartalmazó – blokkot.

2.3. Állítás. *Van olyan minimálisan megfelelő σ ütemezés, amelyben particionáló munka nincs a rövid munkákat tartalmazó blokkon belül ütemezve.*

Bizonyítás. Legyen σ egy minimálisan megfelelő ütemezés. Először vizsgáljuk meg σ-t. Ha abban találunk olyan munkákat, ahol egy particionáló munka közvetlenül megelőz egy rövid munkát, akkor vizsgáljuk meg, hogy ezek cseréje után változik-e az ütemezés költsége? Ha a költség nem változik, akkor cseréljük meg a két munkát. Ezek után σ-ban nem marad olyan pár, amelyekre egy rövid munka közvetlenül követ egy particionáló munkát, és a munkák cseréje után az ütemezés költsége nem változik.

Tegyük fel, hogy az állítás nem igaz, és a σ ütemezésben a $\mathcal{S}_i$ halmaz két egymás utáni munkája között vannak particionáló munkák. Legyen $S_{i,j}$ és $S_{i,j+1}$ a két egymás utáni munka. Ekkor a két munka elvárt befejezési idejére igaz lesz, hogy $d(S_{i,j+1}) = d(S_{i,j}) + 4$. Tegyük fel, hogy a két rövid munka között van legalább egy particionáló munka. Legyenek ezek $P_{k_1}, \dots, P_{k_r}$. Jelölje s és t az ebből a halmazból legkésőbb ütemezett particionáló munka (P_{k_r}) kezdési és befejezési időpontját. Legyen továbbá $d = d(S_{i,j}) + 2$.

A. Eset. $d \le (s+t)/2$. Cseréljük fel P_{k_r} és $S_{i,j+1}$ ütemezését. Ekkor a P_{k_r} ütemezési költsége 4-gyel növekszik. Azt tudjuk, hogy az ütemezés előtt a $S_{i,j+1}$ rövid munka későbbre van ütemezve, mint a hozzá tartozó elvárt befejezési idő.

A.1. Eset. Ha $s + 4 \ge d(S_{i,j+1})$, akkor a csere után is késni fog az $S_{i,j+1}$ rövid munka. A csökkenés nagyobb lesz, mint 4, mert bármely particionáló munka hossza lényegesen nagyobb, mint a rövid munkák hossza. Ez pedig ellentmond annak, hogy σ minimálisan megfelelő ütemezés.

A.2. Eset. Tegyük fel, hogy $s + 4 < d(S_{i,j+1})$. Ekkor a $S_{i,j+1}$ rövid munka a csere után nem késik, hanem korábban fejeződik be, mint a hozzá tartozó elvárt befejezési idő. A csere előtt a késés $t+4-d(S_{i,j+1}) = t+4-(d+2) = t-d+2$ volt. A csere után a munka ütemezésének a költsége $d(S_{i,j+1})-(s+4) = d+2-(s+4) = d-s-2$ lesz. Így az $S_{i,j+1}$ rövid munka átütemezése során a költség csökkenése legalább

$$t - d + 2 - (d - s - 2) = t + s - 2d + 4 \ge 4.$$

Ha a bal oldal nagyobb, mint 4, akkor ismét ellentmondásra jutunk. Viszont egyenlőség nem állhat, mert – a bizonyítás elején történt csere miatt ha egyenlőség lett volna, akkor a két munkát már korábban megcseréltük volna.

B. ESET. Tegyük fel, hogy $d > (s+t)/2$. Ekkor a legkésőbben ütemezett particionáló munka helyett tekintsük a legkorábban ütemezett particionáló munkát. (Ha $r = 1$, akkor ezt a munkát választjuk.) Ez P_{k_1}. Legyen ennek a kezdési és befejezési ideje s' és t'. Világos, hogy $d > (s'+t')/2$.

Cseréljük most meg $S_{i,j}$ és P_{k_1} ütemezését. Ekkor a particionáló munka ütemezésének a költsége 4 egységgel csökkenni fog, és a rövid munka kezdési időpontja eltolódik $t'-4$-be. Azt tudjuk, hogy a csere előtt a rövid munka korábban fejeződik be, mint a hozzá tartozó elvárt befejezési idő.

B.1. ESET. $t' \leq d(S_{i,j})$. Ekkor a rövid munka továbbra is korábban lesz ütemezve, és munka ütemezésének költsége legalább annyival csökken, mint a particionáló munka hossza, ami lényegesen nagyobb, mint 4. Így a csere után az ütemezés költsége csökkenni fog, ami ellentmond annak, hogy σ minimálisan megfelelő volt.

B.2. ESET. $t' > d(S_{i,j})$. Ekkor a csere után a rövid munka késni fog. A csere előtt a munka ütemezésének a költsége legalább $d(S_{i,j}) - s' = d - 2 - s'$ volt, és a csere után a költség $t' - d(S_{i,j}) = t' - d + 2$. Így a rövid munka ütemezési költségének változása

$$(t' - d + 2) - (d - 2 - s') = (t' + s') - 2d + 4 < 4$$

lesz. Itt – az utolsó egyenlőtlenséghez – kihasználtuk, hogy $d > (s'+t')/2$. Így ebben az esetben is ellentmondásra jutunk, mert a csere után az ütemezés költsége csökken.

□

Definiáljuk most az A_1-t mint azon j indexeknek a halmazát, amelyekre a P_j particionáló munkák $\mathcal{S}_1$-t megelőzik, és legyen A_i $(2 \leq i \leq u+1)$ azon j indexek halmaza, amelyekre a P_j particionáló munkák a $\mathcal{S}_{i-1}$ és $\mathcal{S}_i$ között lettek ütemezve egy σ minimálisan megfelelő ütemezésben.

Legyen $\mathcal{P}_{A_i} = \{P_j : j \in A_i\}$, $1 \leq i \leq u+1$. Ekkor σ-ban a munkákat a következő sorrendeben ütemezzük:

$$\mathcal{P}_{A_1},\ \mathcal{S}_1,\ \mathcal{P}_{A_2},\ \mathcal{S}_2,\ \ldots,\ \mathcal{P}_{A_u},\ \mathcal{S}_u,\ \mathcal{P}_{A_{u+1}}.$$

A $\mathcal{P}_{A_{u+1}}$ halmazra azért van szükségünk, mert nem tudjuk, hogy a σ ütemezés particionáló munkákkal vagy rövid munkákkal fejeződik be.

Nevezzünk egy ütemezést *ideálisnak*, ha minden $\mathcal{P}_{A_i}$ $(1 \leq i \leq u)$ halmazban elhelyezett munkák összhossza pontosan M_pB.

2.4. ÁLLÍTÁS. *A $\mathcal{P}_{A_i}$ $(1 \leq i \leq u)$ halmazban elhelyezett particionáló munkák összhossza legfeljebb M_pB.*

Bizonyítás. Egy ideális ütemezésben legyen a j. particionáló halmazban elhelyezkedő első munka kezdési ideje s és befejezési ideje $t = s + M_pB$.

Tegyük fel, hogy az állításunk nem igaz, és van egy olyan $\mathcal{P}_{A_j}$, amelyben az ütemezett particionáló munkák végrehajtási idejének összege legalább $M_pB + M_p$. Ekkor $\mathcal{P}_{A_j}$-ben legalább 3 munka van. Mivel a $\mathcal{P}_{A_j}$-ben elhelyezkedő munkák összhossza legalább $M_pB + M_p$, ezért ezek a munkák vagy s előtt, vagy t után foglalnak le $M_p/2$ helyet. Egy particionáló munka hossza legalább $M_pB/4 > M_p/2$.

A. Eset. Tegyük fel, hogy a $\mathcal{P}_{A_j}$ halmazba tartozó első munka kezdési időpontja legfeljebb $s - M_p/2$. Ekkor $j > 1$.Vizsgáljuk meg az $\mathcal{P}_{A_j}$ halmaz előtt közvetlenül elhelyezkedő rövid munkák költségét. Ezek a munkák most korábban lettek ütemezve, és mindegyik befejezési ideje legalább $M_p/2$-vel csökkent. Így a $\mathcal{S}_{j-1}$-ben ütemezett munkák összköltsége legalább $M_pM_s/2$.

B. Eset. Tegyük fel, hogy a $\mathcal{P}_{A_j}$ halmazba tartozó utolsó munka befejezési időpontja legalább $t+M_p/2$. Vizsgáljuk meg az $\mathcal{P}_{A_j}$ halmaz után közvetlenül elhelyezkedő rövid munkák költségét. Ezek a munkák most későbben lesznek ütemezve, és mindegyik befejezési ideje legalább $M_p/2$-vel nő. Így a $\mathcal{S}_j$-ben ütemezett munkák összköltsége legalább $M_pM_s/2$.

A 2.1 Állítás miatt mindkét esetben az ütemezés összköltsége nagyobb lesz, mint W_0, ami ellentmond annak, hogy σ egy minimálisan megfelelő ütemezés.

□

2.5. Állítás. $A_{u+1} = \emptyset$.

Bizonyítás. Tegyük fel, hogy az állítás nem igaz, és $A_{u+1} \neq \emptyset$. Adjunk egy alsó becslést az ütemezésben elhelyezett particionáló munkák költségére. A 2.4 Állításból következik, hogy minden $\mathcal{P}_{A_i}$ halmazban legfeljebb 3 munka van, ahol $i = 1, 2, \ldots, u$. Ezért az első három particionáló munka összes eltérése legalább 0. A következő három particionáló munkát megelőz M_s darab 4 hosszúságú rövid munka, ezért ezek összes eltérése legalább $4M_s$, függetlenül attól, hogy melyik particionáló halmazban kerülnek elhelyezésre. Folytatva ezt a gondolatot azt kapjuk, hogy az $(u-1)$. három particionáló munkára az összes eltérés legalább $4(u-2)M_s$.

Maradt 3 nem vizsgált particionáló munka. Mivel azt feltételeztük, hogy $A_{u+1} \neq \emptyset$, ezért ezek közül kettőt biztosan megelőz $(u - 1)$ darab rövid munkákat tartalmazó blokk. Egy ilyen blokk pontosan M_s darab 4 hosszúságú munkát tartalmaz, ezért a két particionáló munka összes költsége legalább $2[4M_s(u - 1)]$. A legutoljára ütemezett particionáló munka költsége legalább $4uM_s$, mivel ezt a munkát megelőz minden rövid munka. Így – felhasználva a 2.1 Állításban bizonyított egyenlőtlenséget – kapjuk, hogy

$$\begin{aligned} cost(\mathcal{P},\sigma) &\geq 3[4M_s + 8M_s + \ldots + (u-2)M_s] + 8(u-1)M_s + 4uMs \\ &= 12M_s[1+2+\ldots+(u-2)] + 8(u-1)M_s + 4uM_s \\ &= 6u(u-1)M_s + 4M_s \\ &= 2M_s(3u^2 - 3u + 2) \\ &> W_0. \end{aligned}$$

Ez pedig ellentmondás, mert azt feltételeztük, hogy σ minimálisan megfelelő ütemezés volt.

□

A 2.4 Állításból és a 2.5 Állításból következik, hogy $\sum_{j\in A_i} a_j \leq B$ minden $1 \leq i \leq u$-ra, és $\sum_{i=1}^{u}\sum_{j\in A_i} a_j = \sum_{i=1}^{3u} a_j = B$. Ezért $\sum_{j\in A_i} a_j = uB$ minden $1 \leq i \leq u$. Így az $(A_1, A_2, \ldots, A_u)$ halmazpartíció a 3PP feladat egy lehetséges megoldása.

□

A fenti levezetés eredményeként kimondható az alábbi tétel.

2.1. Tétel. *Az $1|(a_j,p,p)|\sum(E_j+T_j)$ feladat erősen $\mathcal{NP}$-nehéz.*

Ha a konstrukciót úgy készítjük el, hogy a particionáló ill. rövid munkákat a $(1,1,M_p\alpha_i - 2)$ ill. $(1,1,2)$ hármasokkal írjuk le, akkor – a fenti bizonyítás lépéseit használva – belátható, hogy igaz a következő

2.2. Tétel. *Az $1|(p,p,b_j)|\sum(E_j+T_j)$ feladat erősen $\mathcal{NP}$-nehéz.*

Összefoglalás

A cikkben arra adtunk egy egyszerűbb bizonyítást, hogy az $1|(a_j,p,p)|\sum(E_j+T_j)$ *CTP* feladat algoritmikus bonyolultsága eltér azoktól az ütemezési feladatoktól, ahol más célfüggvényeket vizsgáltak. A [20]-ban közölt bizonyítást három ponton javítottuk: bizonyításunkban elegendő két korlátot vizsgálni a W_0 küszöbértékre, konstrukcióban a „rövid” munkák befejezési idejei különböznek, ezáltal az optimális megoldáshoz tartozó összes eltérés kisebb és egyszerűbben meghatározható, végül a munkákhoz tartozó első műveletek hosszát meghatározó képletben a konstansok értéke jelentősen kisebb, ezáltal a bizonyítás áttekinthetősége javul.

Ez azt jelenti, hogy a feladat bonyolultsági gráfjában máshol van a határ a $\mathcal{P}$-beli és az $\mathcal{NP}$-beli speciális feladatok között. A további kutatások szempontjából érdekes lehet megvizsgálni a $1||\sum(E_j+T_j)$ ET-feladatot a különböző speciális esetekre, és ezek alapján megkonstruálni a problémához tartozó bonyolultsági gráfot. Azt érdemes előre vetíteni, hogy már a legegyszerűbbnek látszó $1|(p,p,p)|\sum(E_j+T_j)$ feladat bonyolultságának meghatározása sem tűnik egyszerűnek, így kérdéses, hogy ennek a feladatnak van-e olyan speciális esete, amelynek algoritmikus bonyolultsága $\mathcal{P}$-ben van.

Köszönetnyilvánítás

A szerzők köszönetet mondanak a bírálóknak, akik a korábbi változat körültekintő olvasásával és konstruktív bírálatukkal jelentősen javítottak a cikk minőségén.

Békési József kutatási tevékenységét az NKFIH Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal SNN 129178 számú OTKA pályázata és az EFOP- 3.6.2-16-2017-00015 pályázata támogatta. Dósa György kutatási tevékenységét az NKFIH Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal SNN 129364 számú OTKA pályázata és az EFOP- 3.6.1-16-2016-00015 pályázata támogatta.

Hivatkozások

[1] A.A. Ageev, and A.E. Baburin, *Approximation algorithms for UET scheduling problems with exact delays*, Operations Research Letters, Vol. **35**, pp. 533–540 (2007). DOI: 10.1016/j.orl.2006.09.006

[2] A.A. Ageev, and A.V. Kononov, *Approximation algorithms for scheduling problems with exact delays*, Lecture Notes in Computer Sciences, Vol. **4368**, pp. 1–14 (2006). DOI: 10.1007/11970125_1

[3] D. Ahr, J. Békési, G. Galambos, M. Oswald, and G.Reinelt, *An exact algorithm for scheduling identical coupled tasks*, Mathematical Methods of Operations Research, Vol. **59**, pp. 193–203 (2004). DOI: 10.1007/s00186-014-0469-6

[4] K.R. Baker, G.D. Scudder, *Sequencing with earliness and tardiness: a survey*, Operations Research, Vol. **38**, pp. 22–36 (1990). DOI: 10.1287/opre.38.1.22

[5] P. Baptiste, *A note on scheduling identical coupled task in logarithmic time*, Discrete Applied Mathematics, Vol. **158**, pp. 583–587 (2010). DOI: 10.1016/j.dam.2009.10.012

[6] J. Békési, G. Galambos, M. Oswald, and G. Reinelt, *Improved analysis of an algorithm for the coupled task problem with UET jobs*, Operations Research Letters, Vol. **37**, pp. 93–96 (2009). DOI: 10.1016/j.orl.2008.11.002

[7] J. Blazewicz, K. Ecker, T. Kis, C.N. Potts, M. Tanas, and J. Whitehead, *Scheduling of coupled tasks with unit processing times*, Journal of Scheduling, Vol. **13**, pp. 453–461 (2010). DOI: 10.1007/s10951-010-0167-z

[8] B. Chen, X. Zhang, *Scheduling of Coupled tasks with Exact Delays for Minimum Total Job Completeion Time*, J. of Scheduling (2020). DOI: 10.1007/s10951-020-00668-1

[9] M.R. Garey, R.E. Tarjan, G.T. Wilfong, *One-processor scheduling with symmetric earliness and tardiness penalties*, Math.of Operations Research, Vol. **13**, pp. 330–348 (1988). DOI: 10.1287/moor.13.2.330

[10] M.N.G. Hall, W. Kubiak, S.P. Sethi, *Earliness and tardiness scheduling problems II: deviation of completion times about a restrictive common due date*, Operations Research, Vol. **39**, pp. 847–856 (1991). DOI: 10.1287/opre.39.5.847

[11] F.J. Hwang, and M.T. Lin, *Coupled-task scheduling on a single machine subject to a fixed-job-sequence*, Computer Industrial Engineering Vol. **60**, pp. 690–698 (2011). DOI: 10.1016/j.cie.2011.01.002

[12] S.V.R. Lakshminarayan, R.L. Papineau, R. Rochette, *Optimal single-machine scheduling with earliness and tardiness penalties*, Operations Research, Vol. **26**, pp. 1079–1082 (1978). DOI: 10.1287/opre.26.6.1079

[13] H. Li, and H. Zhao, *Scheduling coupled-tasks on a single machine*, In Proceedings of 2007 IEEE symposium on computational intelligence in scheduling, Honolulu, Hawaii pp. 137–142 (2007). DOI: 10.1109/SCIS.2007.367681

[14] A.J. Orman, and C.N. Potts, *On the complexity of coupled-task scheduling*, Discrete Applied Mathematics, Vol. **72** No. **1-2**, pp. 141–154 (1997). DOI: 10.1016/S0166-218X(96)00041-8

[15] Y.M. Shafransky, and V.A. Strusevich, *The open shop scheduling problem with a given sequence of jobs on one machine*, Naval Research Logistics, Vol. **45**, pp. 705–731 (1998). DOI: 10.1002/(SICI)1520-6750(199810)45:7<705::AID-NAV4>3.0.CO;2-F

[16] R.D. Shapiro, *Scheduling coupled tasks*, Naval Research Logistics Quarterly, Vol. **27**, pp. 4489–498 (1980). DOI: 10.1002/nav.3800270312

[17] H.D. Sherali, and J. C. Smith, *Interleaving two-phased jobs on a single machine*, Discrete Optimization, Vol. **2**, pp. 348–351 (2005). DOI: 10.1016/j.disopt.2005.08.002

[18] J.B. Sidney, *Optimal single-machine scheduling with earliness and tardiness penalties*, Operations Research, Vol. **25**, pp. 62–69 (1977). DOI: 10.1287/opre.25.1.62

[19] W. Yu, H. Hoogeveen, and J.K. Lenstra, *Minimizing makespan in a twomachine flow shop with delays and unit-time operations is NP-hard*, Journal of Scheduling, Vol. **7**, pp. 3333–348 (2004). DOI: 10.1023/B:JOSH.0000036858.59787.c2

[20] L. Wan, J. Yuan, *Single machine scheduling to minimize the total earliness and tardiness is strongly NP-hard*, Operations Research Letters, Vol. **41**, pp. 3693–365 (2013). DOI: 10.1016/j.orl.2013.04.007

Békési József 1963-ban született. 1987-ben végzett a JATE Matematikus szakán. PhD fokozatát 1998-ban szerezte a Szegedi Tudományegyetemen és 2020-ban habilitált. Munkahelyei: Juhász Gyula Tanárképző Főiskola (1987-2000), Szegedi Tudományegyetem (2000 óta). 2008 óta főiskolai tanár. Leginkább az ütemezéselmélet és a ládapakolás különféle változataival foglalkozik. Több egyetemi jegyzet szerzője vagy társszerzője, 50-nél több tudományos publikációja van, amelyekre több mint 300 hivatkozást kapott (Google Scholar). Köztestületi tag, tagja a Magyar Operációkutatási Társaságnak (MOT), továbbá a Bolyai János Matematikai Társulatnak.

BÉKÉSI JÓZSEF
Számítógépes Algoritmusok és Mesterséges Intelligencia Tanszék,
Természettudományi és Informatikai Kar,
Szegedi Tudományegyetem,
POB 361, H-6701 Szeged
bekesi.jozsef@szte.hu

Dósa György 1963-ban született. 1987-ben végzett az ELTE Matematikus szakán, Operációkutatási szakirányon. PhD fokozatát 2009-ben szerezte a Szegedi Tudományegyetemen, és 2018-ban lett az MTA doktora. Munkahelyei: ELTE Operációkutatási Tanszék (1987-1991), Széchenyi István Egyetem (1991-94), Pannon Egyetem (1994 óta). Egyetemi tanár 2019 óta. Leginkább az ütemezéselmélet és ládapakolás különféle változataival foglalkozik. Több egyetemi jegyzet szerzője vagy társszerzője, 100-nál több tudományos publikációja van, amelyekre több mint 1300 hivatkozást kapott (Google Scholar). Köztestületi tag, tagja a Magyar Operációkutatási Társaságnak (MOT), továbbá az MTA regionális szervezetének (VEAB) is tagja, két munkabizottságban, az egyiknek titkára.

DÓSA GYÖRGY
Matematika Tanszék, Műszaki Informatikai Kar,
Pannon Egyetem,
Egyetem utca 10.,
H-8200 Veszprém,
dosagy@almos.vein.hu

Galambos Gábor 1947-ben született. 1971-ben végzett a JATE Programtervező Matematikus szakán. PhD fokozatát 1992-ben védte, 1999-ben habilitált a Debreceni Egyetemen. Az MTA Doktora címet 2017-ben kapta meg, 2000-2017 időszakban egyetemi tanár az SZTE-n. Jelenleg professzor emeritusz. Fő kutatási területe: diszkrét matematikai problémák megoldására készített közelítő algoritmusok elemzése. Több külföldi egyetemen töltött hosszabb időt, többek között Augsburgban, Rotterdamban, Bolognában, Grazban és Heidelbergben volt vendégkutató vagy vendégprofesszor. Tudományos közleményeinek a száma 79, amelyekre több, mint 1400 hivatkozást kapott (Google Scholar). Köztestületi tag, tagja a Magyar Operációkutatási Társaságnak.

GALAMBOS GÁBOR
Informatika Alkalmazásai Tanszék,
Juhász Gyula Pedagógusképző Kar,
Szegedi Tudományegyetem,
POB 361, H-6701 Szeged,
GalambosGabor@szte.hu

ON MINIMIZING TOTAL EARLINESS AND TARDINESS OF COUPLED-TASKS – A COMPLEXITY RESULT

József Békési, György Dósa, Gábor Galambos

In the paper we prove that the complexity of the *CTP* problem $1|(a_j,p,p)|\sum(E_j+T_j)$ differs from scheduling problems with other objective functions. This means that the line between the problems in P and NP runs along different special cases. According to the future research it can be interesting to investigate the problem $1||\sum(E_j+T_j)$ for different special cases, and taking into account these results one can construct the complexity graph. It is worth mentioning that to determine the complexity of the problem even in the simplest case $1|(p,p,p)|\sum(E_j+T_j)$ is not easy. So it is possible that this problem does not have any subcase with complexity in the class P.