

Válasz dr. Balogh János „*Nehéz, kombinatorikus optimalizálási feladatok számítógéppel segített megoldása*” című doktori (PhD) disszertációmhoz megküldött bírálatára

1. kérdés/megjegyzés: A négy példaosztályból véleményem szerint kettőben impresszívek a futtatási eredmények. A másik kettőben is lehet, hogy az, de ezt nem tudjuk, amikor nem ismert OPT, azaz alsó és felső korlátok nem vágnak egybe, „messzire vannak” egymástól. Elképzelhetőnek tartom azt is, hogy az elért értékek az optimális, vagy ahhoz nagyon közeli értékek, csak az alsó korlátok nem elég jók (ezeket a [68] és [69] cikkekbeli alsó korlátokat használja a szerző). Így nem merült-e föl a kérdés, vagy nem foglalkozott-e azzal, hogy: Nem lehetne jobb korlátokat megadni?

1. kérdésre/megjegyzésre adott válasz: Magam is igyekeztem ezeknél erősebb alsó korlátot előállítani és áttekintettem az erre vonatkozó irodalmat is, azonban nem találtam és nem is tudtam jobb korlátokat előállítani. Ennek az oka az, hogy az egyes munkák végrehajtási idejei nagyon változatosak, közöttük kihasználható szabályszerűségeket nem fedeztem fel. Ezek a gépi idők egyenletes eloszlás szerint, véletlenszerűen kerültek generálásra. A két alsó korlát informatív-nak és jól alkalmazhatónak tekinthető. Egyrészt, mindkét alsó korlát könnyen és gyorsan számítható, másrészt megállapítható, hogy az alkalmazott két alsó korlát erős korlátnak tekinthető. Ezt bizonyítja az a megfigyelés, hogy a QLM algoritmus által talált megoldás sok esetben megegyezik az LB_1 és LB_2 alsó korlátok maximumával, és sok esetben ez a maximum érték erősebb alsó korlát volt, mint a CPLEX által kiszámított alsó korlát. Ezen okok miatt alkalmaztam ezt a két korlátot. Az LB_1 alsó korlát segítségével a problémában a megelőzési relációk által leírt láncokat vizsgálom. Ez az alsó korlát azt adja meg, hogy ha az egyes láncokban szereplő tevékenységekhez a végrehajtási idő szerint leggyorsabb gépet rendeljük, mekkora a végrehajtásban legtovább tartó lánc átfutási ideje. Az LB_2 esetében pedig minden tevékenységhez azt a gépet rendeljük, amely a tevékenységet a leggyorsabban hajtja végre, ezeknek a minimumoknak vesszük az összegét és elosztjuk a gépek számával.

A *Class #1* osztályhoz generált 10 feladat esetében a QLM algoritmus a 10 futásból 10 alkalommal találta meg az optimális megoldást. Mind a CPLEX, mind a QLM algoritmus esetén a futás gyors, kevesebb, mint 1 másodperc alatt ment végbe.

A 1. táblázatban látható, hogy minden feladat esetében sikerült megtalálni az optimális megoldást, amelyet a CPLEX is megerősített. A CPLEX által kiszámított alsó és felső korlátok minden feladatnál megegyeznek. A QLM esetében számított

	$n = 14, m = 8, NC = 5$									
Feladatok:	1	2	3	4	5	6	7	8	9	10
LB_1	9	5	9	8	8	4	9	17	7	10
LB_2	3	3	3	3	3	2	3	5	3	3
CPX LB	9	5	9	8	8	4	9	17	7	10
CPX UB	9	5	9	8	8	4	9	17	7	10
QLM	9	5	9	8	8	4	9	17	7	10
QLM* (/10)	10	10	10	10	10	10	10	10	10	10

1. táblázat. A bővített feladatok *Class #1* osztályának eredményei

korlátok közül az LB_1 korlát értékei minden oszlopban megegyeznek a CPLEX korlátaival. Mind a QLM, mind a CPLEX átlagosan 1 másodpercen belüli futási idővel dolgozott.

A *Class #2* feladatai esetében már változóak az eredmények. A CPLEX mindössze egy esetben tudta megerősíteni az optimális megoldást, méghozzá a 6-os számú esetben. Ha a feladatok közül például az 1-es számút tekintjük, akkor látható, hogy a 10 futásból a QLM 6 alkalommal számolt 9-es értékű átfutási időt. Ugyanezen feladat esetében az $LB_1 = 8$ és $LB_2 = 7$. Ebből látható, hogy az optimális megoldás legfeljebb 9 és legalább 8.

	$n = 28, m = 7, NC = 8$									
Feladatok:	1	2	3	4	5	6	7	8	9	10
LB_1	8	6	5	8	7	7	6	6	5	9
LB_2	7	7	7	8	9	6	7	6	7	9
CPLEX LB	8	6	5	8	7	7	6	6	5	9
CPLEX UB	9	8	8	9	12	7	7	8	7	11
QLM	9	8	8	9	12	7	7	8	7	11
QLM* (/10)	6	10	10	3	6	2	6	7	9	6

2. táblázat. A bővített feladatok *Class #2* osztályának eredményei

A CPLEX is ugyanezeket az értékeket számolta, azaz az alsó korlát 8, a felső korlát pedig 9 volt. A CPLEX által számított alsó korlátok a 2-es, 3-as, 5-ös, 7-es és 9-es feladatok esetében rosszabbak, mint a $\max(LB_1, LB_2)$. A 6-os számú feladat esetében a CPLEX alsó korlátja és felső korlátja megegyeznek, így itt sikerült megerősíteni az optimális megoldást, amelyet a QLM is megtalált. A 7-es és 9-es feladatoknál látható, hogy az LB_1 és LB_2 értékei megegyeznek a CPLEX alsó és felső korlátjával, továbbá a QLM által számított eredmények mindkét esetben egyenlők az LB_2 értékével, így a megoldás optimális, azonban ezt a CPLEX nem erősítette meg. A CPLEX átlagosan 12 másodperc alatt megtalálja az általa legjobb megoldást, de ezen utána nem tud már javítani. A 8 órás időkeretben nem képes a megoldás optimalitását verifikálni (kivéve az említett 6-odik input esetén.)

A *Class #3* feladataihoz tartozó eredmények a 3. táblázatban láthatók. Megálapíthatóak a következők:

- a CPLEX alsó korlátja mindig az LB_1 -el egyezik meg, de ennél lényegesen jobb az LB_2 érték, érdekes, hogy ezt a CPLEX nem találta meg,

	$n = 27, m = 4, NC = 1$									
Feladatok:	1	2	3	4	5	6	7	8	9	10
LB_1	5	8	5	7	6	8	5	8	6	6
LB_2	16	18	15	18	16	19	17	18	19	16
CPLEX LB	5	8	5	7	6	8	5	8	6	6
CPLEX UB	18	20	18	19	16	20	17	19	21	17
QLM	18	20	17	19	16	20	17	19	21	17
QLM* (/10)	10	9	7	8	2	10	1	8	9	10

3. táblázat. A bővített feladatok *Class #3* osztályának eredményei

- a QLM által adott megoldás és az LB_2 értékei között két alkalommal 0, négy alkalommal 1 és ugyancsak négy alkalommal 2 a különbség,
- a CPLEX soha nem ad jobb megoldást, mint a QLM, azonban egy feladat esetén a QLM jobb megoldást ad (3. feladat).

A CPLEX itt sem képes garantáltan optimális megoldást találni 8 órás futásidő alatt.

	$n = 74, m = 19, NC = 10$									
Feladatok:	1	2	3	4	5	6	7	8	9	10
LB_1	6	4	4	5	5	4	4	6	5	4
LB_2	5	5	5	5	5	5	5	5	5	5
CPLEX LB	6	4	4	5	5	4	3	5	5	4
CPLEX UB	13	9	11	9	9	15	9	9	8	10
QLM	6	5	6	6	6	5	6	6	5	5
QLM*	6	1	7	10	10	1	10	8	1	3

4. táblázat. A bővített feladatok *Class #4* osztályának eredményei

A *Class #4* feladataihoz tartozó eredmények a 4. táblázatban láthatók. Az eredményekből látható, hogy a QLM algoritmus az esetek több, mint a felében (1-es, 2-es, 6-os, 8-as, 9-es, 10-es feladatok) ugyanazt az eredményt adta, mint a $\max(LB_1, LB_2)$. A 3-as, 4-es, 5-ös és 7-es feladatok esetében a QLM eredménye csak eggyel nagyobb, mint $\max(LB_1, LB_2)$. Az 1-es, 2-es, 6-os, 8-as, 9-es és 10-es feladatok esetben a QLM eredményei egyben az optimális megoldások is. Az is látható, hogy a CPLEX számára nehezek voltak ezek a feladatok, mert nagyon magas felső korlátokat számolt ki, egyúttal egyik esetben sem tudta megerősíteni az optimális megoldást. A CPLEX átlagosan 9 másodperc alatt találja meg az általa talált legjobb megoldást, és a 8 órás időkeret nem elég annak eldöntésére, hogy ez optimális-e.

A vizsgálatok alapján elmondható, hogy a megoldások minősége nem a QLM paraméterein múlik. Ezek a feladatok felépítésükből eredően nehezek. Nem arról van szó, hogy a megoldások nem sikeresek, hanem a feladatok nehézsége folytán a CPLEX nem tudta igazolni, hogy az optimális megoldást kaptuk-e meg vagy sem.

2. kérdés/megjegyzés: A tézisfüzetben és a disszertációban is szerepelnek a feladatra megadott teszt példányokra a CPLEX-szel végzett vizsgálatok. A disszertációban a szerző a C. függelékben megadja a vizsgált modellt, amely egy kevert egészértékű lineáris programozási modell, és ezt konkrét teszt példányokon végrehajtva működése során nem csak felső, hanem alsó korlátokat is szolgáltat. A tézisfüzetben is szerencsés lett volna a „CPLEX megoldó által szolgáltatott eredmények” esetén utalni (arra) a modellre, amelyen ezeket a futtatásokat végrehajtotta a szerző. (Hiszen a CPLEX csak egy eszköz meglátásom szerint, amely más – esetleg jobb vagy rosszabb modellekre – más eredményeket adna.)

2. kérdésre/megjegyzésre adott válasz: A feladatnak az alábbi modelljét alkalmaztam. Ez a modell szerepel az [1] cikkben.

$$\text{Min } C_{\max} = \max_{j \in J} FT_j \quad (1)$$

s. t.

$$\sum_{v=1}^m \sum_{r=1}^{UB} x_{jvr} = 1, \forall j \in J \quad (2)$$

$$\sum_{j=1}^n x_{jvr} \leq 1, \forall r \in R, \forall v \in M \quad (3)$$

$$\sum_{i=1}^n x_{ivr} - \sum_{j=1}^n x_{j,v,r-1} \leq 0, \quad (4)$$

$$\forall v \in M, \forall r \in \{2, \dots, UB\}$$

$$FT_j - FT_i + L(2 - x_{jvr} - x_{i,v,r-1}) \geq p_{jv}, \quad (5)$$

$$\forall i, j \in J, i \neq j, \forall v \in M, \forall r \in \{2, \dots, UB\}$$

$$FT_j \geq \sum_{r=1}^{UB} p_{jv} x_{jvr}, \forall j \in J, \forall v \in M \quad (6)$$

$$FT_j - FT_i \geq \sum_{v=1}^m \sum_{r=1}^{UB} p_{jv} x_{jvr}, \forall i \in P_j \quad (7)$$

$$x_{jvr} \in \{0, 1\}, FT_j \geq 0, \forall j \in J, \forall v \in M, \forall r \in R$$

J	munkák halmaza, $J = \{1, \dots, n\}$
M	gépek halmaza, $M = \{1, \dots, m\}$
UB	azon pozíciók maximális száma minden gépen, ahová a feladatok kerülnek, $UB = n - m + 1$
R	pozíciók halmaza, $R = \{1, \dots, UB\}$
p_{jv}	a j . munka végrehajtási ideje a v gépen
P_j	a j . munkát közvetlenül megelőző munkák halmaza
L	egy nagy pozitív szám
x_{jvr}	1 az értéke, ha a j . munka az r . pozícióban került feldolgozásra a v . gépen, különben 0
FT_j	a j . munka befejezési ideje

5. táblázat. A modell paraméterei

- (1) A célfüggvény, a teljes átfutási időt minimalizáljuk, amely nem lehet kisebb mint bármely munka befejezésének ideje.
- (2) Minden munka esetén, a munkát pontosan egy gép fogja végrehajtani, és ezen a gépen a munka az r -edik pozícióba fog kerülni, valamely r -re.
- (3) Minden v munka és r pozíció esetén, ide legfeljebb egy munka ütemezhető.
- (4) Az r -edik pozícióba csak akkor teszünk munkát, ha az $r - 1$ -edik pozícióban is van munka, egyébként nem (emiatt a munkák folyamatosan lesznek ütemezve a gépeken, tehát pl. olyan nem fordulhat elő, hogy van munka az 1. pozícióban, aztán a 2-ban nincs, de utána a 3-ban meg megint van).
- (5) Ha valamely v gép esetén valamely j munka közvetlenül az i munka utáni pozícióban van, akkor a j -edik munka befejezési ideje legalább akkora, mint az előtte levő (tehát az i munka) befejezési ideje, plusz a j -edik munka megmunkálási ideje azon a gépen.
- (6) A j -edik munka befejezési ideje legalább akkora, mint a v -edik gépen a j -edik munka végrehajtási ideje, ha a munka erre a gépre van ütemezve.
- (7) A j -edik munka befejezési ideje legalább akkora, mint bármely azt közvetlenül megelőző munka befejezési ideje, plusz a j -edik munka megmunkálási ideje.

A tézisfüzetben „*CPLEX megoldó által szolgáltatott eredmények*” esetén valóban nem utaltam az alkalmazott modellre. Az észrevétel teljesen jogos, köszönöm szépen!

A modellel kapcsolatban a következő észrevételeket teszem. Az irodalomban kétféle modell létezik: az egyik a munkák sorrendjére épül, a másik pedig a *time slot* modell, amely azt modellezi, hogy mikor és melyik munka van elvégezve melyik gépen. A 2. fejezetben megadott feladat a munkák sorrendjére épít, így egyszerű és egyértelmű volt a modellek közötti választás. Az általam alkalmazott modell arra épül, hogy a munkák valamilyen sorrendben vannak végrehajtva. Emiatt nem gondoltam arra, hogy más modelleket is kipróbáljak. De lehetett volna a *time slot*

modellt is vizsgálni, elképzelhető, hogy arra jobb eredményeket adna a CPLEX, viszont az erősen függ a végrehajtási idők nagyságától. Az alkalmazott modell viszont nem. A másik típusú modell vizsgálata további kutatás tárgya lehet. Önmagában a modellek összehasonlítása nagy feladat [2].

Szeretném megköszönni dr. Balogh János docens úrnak, hogy elvállalta és elkészítette a bírálatot. A bírálóban megfogalmazott kérdések és észrevételek nagyon hasznosak voltak számomra.

Veszprém, 2022. november 9.



.....
Ábrahám Gyula
doktorandusz
Pannon Egyetem
Informatikai Tudományok Doktori Iskola

Hivatkozások

- [1] C. Liu and S. Yang. A heuristic serial schedule algorithm for unrelated parallel machine scheduling with precedence constraints. *J. Softw.*, 6:1146–1153, 2011.
- [2] T. Hajba and Z. Horváth. Milp models for the optimization of real production lines. *Central European Journal of Operations Research*, 23(4):899–912, 2015.