

# Gyártási műveletek illeszkedésén alapuló heurisztikus ütemező módszer alkalmazása szalagrendszerű gyártásban

Farkas Gábor Ákos, Villányi Balázs János, Martinek Péter

Budapesti Műszaki és Gazdaságtudományi Egyetem  
Elektronikai Technológia Tanszék, 1111 Budapest, Egry József u. 18.  
E-mail: martinek@ett.bme.hu

**Tartalmi kivonat.** A gyártásütemezés feladata a múlt század közepe óta foglalkoztatja a kutatókat. A magas költségű gyártósorok és berendezések optimális kihasználása és a megrendelések minél gyorsabb kiszolgálása elengedhetetlenné teszi ezen módszerek használatát. Jelen cikkben a gyártástervezés egy speciális területével, a szalagrendszerű gyártással foglalkozunk. Egymást követő gyártási műveletek átfutási idejének összehasonlításával egy egyszerű illeszkedés alapú heurisztikus megközelítést mutatunk be. Módszerünk épít továbbá az utazó ügynök probléma egy korszerű megoldására is. Az ütemezés optimalizálásánál célfüggvénynek a teljes átfutási idő minimalizálását választottuk. Megoldásunk hatékonyságát annak működési paramétereinek finomhangolásával is növeltük illetve részletesen vizsgáltuk az egyes paraméterek végeredményre gyakorolt hatását. Megközelítésünk jóságát a szakirodalomban elterjedt mintafeladatok megoldásával bizonyítottuk.

**Kulcsszavak:** gyártásütemezés; szalagrendszerű gyártás; átfutási idő minimalizálása; heurisztikus megközelítés

## 1. BEVEZETÉS

A termelés rendszerszerű, módszeres megtervezése az első ipari forradalom óta foglalkoztatja a termelésirányítókat. Ennek érdekében szükséges volt munkafolyamatokat definiálni, standardizálni; a szabályozottságot a termelési rendszerben megteremteni. Ezt a feladatot ma már termeléstervezésként (és -irányításként) azonosítjuk. A Ford-féle tömeggyártási rendszer létrejötté óta a gyártási folyamatok komplexitása jelentősen nőtt, így a termeléstervezéshez szükséges munkaelemzési és folyamatszabályozási feladatokat egyre összetettebb adathalmazon kell elvégezni [1]. Ennek megfelelően, a termeléstervezés és -irányítás is már egyre inkább a digitalizációról és az ipari információtechnológiai megoldásokról szólnak [2].

A termeléstervezés a vállalati tervezés egy szintje, melynek feladata a gyártási terv elkészítése. Ehhez figyelembe veszi a termelőrendszer paramétereit: például az üzemek valós napi kapacitásait és a termelési körülményeket. A tervezés eredménye a gyártási rendelések, amelyek elemi műveleteit még ütemezni kell konkrét gyártóberendezésre. A gyártásütemezés során meghatározzuk az elemi műveletek optimális sorrendjét, a szükséges alapanyagok és munkaerő biztosítása mellett. A gyártási rendelések tehát

komplex gyártási folyamatokat írnak le, különböző gyártó berendezésekre és időtartamokra. Az előbb ismertetett ütemezés során pedig pontosan definiáljuk az egyes munkafázisok erőforrás-szükségletét és gépfoglalási idejét.

Következésképpen, a gyártásütemező algoritmusok kezelnek elemi műveleteket, gyártóberendezéseket és időintervallumokat. A cél elsősorban a megvalósíthatóság vizsgálata, másodsorban pedig a célfüggvény által meghatározott költség – például teljes átfutási idő – minimalizálása.

A gyártási elrendezéseknek alapvető hatása van az ütemezési problémák formalizálására. Két alapvető elrendezési véglet különböztetünk meg. A műhelyrendszerű gyártás (job-shop) esetében a hangsúly a terméken és az azt előállító üzemen van, e köré szerveződik a gyártás [3]. Így a hasonló műveletek gépei egy helyre, azonos műhelyekbe kerülnek. Végbemegy a gyártási folyamatok műhelyek szerinti specializációja, mely ezen gyártási rendszer jellegzetessége. A gyártási folyamat egyes műveletei tehát más-más műhelyben kerülnek elvégzésre. A másik elrendezési véglet a szalagrendszerű gyártás (flow-shop), ahol a hangsúly a termelési folyamaton és annak hatékonyságán van [4]; a termék és termelő

berendezés ennek van alávetve. Jellemzően a gépek, illetve a munkafolyamatok a gyártási folyamat sorrendjében, egymáshoz közel helyezkednek el. További általános jellemzőjük, hogy a gépek kapacitása kiegyenlített, azaz közöttük ütemazonosság áll fenn. Sajátos megvalósulási formája a futószalagos gyártási rendszer. A két szélsőséges gyártási elrendezés között több közbülső forma is ismeretes [2].

Jelen cikkben szalagrendszerű gyártás ütemezésére mutatunk be egy új megközelítésmódszert. Az ütemezési probléma a következőképpen általánosítható: legyen adott  $n$  darab munkafolyamat, amelyek egyenként  $m$  darab műveletből állnak, és ezeket  $m$  darab gép között kell optimálisan szétosztani térben-időben – azaz ütemezni – a gépek legjobb kihasználtsága mellett.

Az ütemezési problémák megoldásának sok módszere és csoportja ismert. Megoldási megközelítések azonban alapvetően háromféle. Egyfelől léteznek egzakt matematikai optimalizáción alapuló megoldások, amelyek célfüggvények analitikus megoldásával globálisan optimális megoldást találnak [5]. Az ezzel járó sokszor jelentős futási idő azonban kizárja ezen eljárások valós problémákra történő alkalmazhatóságát. Egy másik ütemezési megközelítés a becslő eljárás, amely az előbb említett futásidő csökkenésével kecsegtet, de az optimalitás rovására. Van azonban egy mellékelt vállalás ezekhez a megoldásokhoz, amely mégiscsak kellő alapot adnak ezen módszerek alkalmazásához: az optimumhoz képest behatárolt mértékben térnek el (approximatívák). És végül az ütemezési megközelítések gyakorlatban leginkább használható alternatíváját a heurisztikán alapuló megoldások nyújtják, amelyek futásideje rövid és lokálisan optimális megoldást eredményeznek [6]. Oly népes ezen megközelítési csoport halmaza, hogy alkategóriákba soroljuk őket. Ismerünk döntési mechanizmusokat használó heurisztikákat, ezek a konstruált heurisztikák. Ilyen például a DR algoritmus. Emellett még vannak fejlődő heurisztikák is, amelyek lokális keresésen alapszanak és jellemzően változtatják döntési logikájukat az ütemezés során. Ide sorolhatjuk például a szimulált lehűtésen [6], és a genetikus

algoritmusokon alapuló ütemezőket [7]. Jelen cikk tárgya egy javasolt új karakterisztika alapján működő heurisztikus ütemező. Módszerünk alapja az egymást követő munkafolyamatok illeszkedésének vizsgálata. Ez alapján történik az optimális végrehajtási sorrend meghatározása, az ütemezési feladatok gépekhez rendelése és az optimális foglalási időintervallumok meghatározása. Az ütemező algoritmus fejlesztésénél különböző működési és paraméteres finomhangolásokat is végeztünk. Az eredményeket az irodalomban széles körben elterjedt Taillard-féle [16] ütemezési problémákon validáltuk a tesztek során.

A cikk felépítése a következő. Ez az első szakasz betekintést nyújt a gyártásütemezés problémájába, hátterébe. A második szakasz ismerteti a kapcsolódó munkákat. Ezt követi az általunk vizsgált ütemezési feladat és a javasolt módszer részletes ismertetése. A mérési eredmények a cikk végén, az összefoglalás és kitekintés előtt kerülnek ismertetésre.

## 2. IRODALMI ÖSSZEFOGLALÓ

A szalagrendszerű gyártás ütemezésével sok munkában foglalkoznak. Rajendran és társai két heurisztikus megközelítést mutatnak a probléma megoldására, mely az új munkák egymás után történő beillesztésével közel optimális megoldást talál [8]. A [9]-ben egy genetikusan alapuló heurisztikus megoldást mutatnak be a teljes átfutási idő hatékony minimalizálására szalagrendszerű gyártásnál. Li és Wand pedig már több célfüggvényt is támogat a genetikusan alapú ütemező megközelítéssel [10]. Dai és társai az átfutási idő minimalizálása mellett az energiafogyasztást is próbálják minimalizálni az ütemezésnél. Módszerük genetikusan alapú és szimulált lehűtésre épül [11].

Léteznek továbbá más gépi tanulási módszeren alapuló megközelítések is. Például a [12]-ben egy mesterséges neurális hálózaton alapuló heurisztikus megoldást láthatunk a szalagrendszerű gyártásütemezés problémájára. Linn és társai egy speciális, úgynevezett hibrid szalagrendszerű problémát ismertetnek, ahol egy-egy művelet elvégzésére párhuzamosan több gép is beállítható. Továbbá gyakran előfordul, hogy az egyes megmunkálási fázisok gépei egyformák, vagy

univerzálisak. Az ütemezési megoldások közül a két, három illetve  $n$ -lépcsős megközelítéseket mutatják be részletesen [13]. A [14] pedig egészen sokféle megközelítést közöl a hibrid szalagrendszerű ütemezési feladat megoldására teljes osztályozási rendszer bemutatásával.

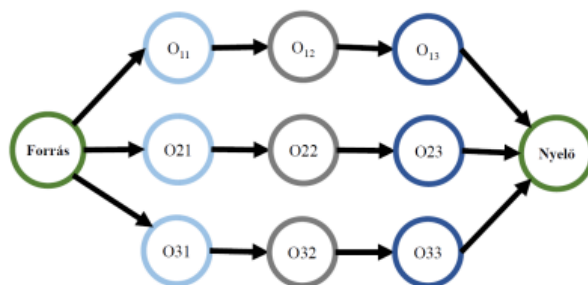
### 3. AZ ÜTEMEZÉSI PROBLÉMÁK OSZTÁLYOZÁSA

Ebben a szakaszban röviden ismertetjük a gyártásütemezési legfontosabb jellemzőit, osztályait, amely egyben betekintést ad megoldásunk alkalmazási környezetébe is.

Az ütemezési probléma általános esetben tartalmaz gyártási tevékenységeket (job), azokat felépítő műveleteket (operation), valamint gyártóberendezéseket (machine). Célunk a tevékenységek műveleteinek szétosztása a gyártóberendezések között úgy, hogy az átfutási idő minimális legyen. Ez előbbit különböző megkötések módosíthatják, mint a határidőtartás, a késedelem minimalizálása, műveleti függőségek betartása stb. Egy gyártási ütemezés megfelelő, ha nincs átlapolódás a műveletek időintervallumai között. Ezen felül optimális is, ha az adott probléma jellemzőjét – jelen esetben például az átfutási időt – minimalizáltuk. További probléma specifikus jellemző lehet az átlagos késedelem mértéke, a maximális késés mértéke, a késedelmes feladatok száma, a gyártóberendezések állásideje, a műveletközi átállások időigénye stb.

Az ütemezési problémákat jellemzően a következő elemek felhasználásával modellezzük. A modell tartalmaz egyrészt gyártási tevékenységeket, amelyek felhasználnak olyan erőforrásokat, mint például a megmunkálást végző gép munkavégző képessége. Adott továbbá egy célfüggvény – például a teljes átfutási idő –, amit minimalizálni szeretnénk, tehát amihez viszonyítjuk az optimumot. Az ütemezés elsődleges szemléltető eszköze a Gantt-diagram, amely kifejezetten ütemezett tevékenységek időintervallumainak relációit hivatott szemléltetni. Ezen felül, az ütemezések függőségeit is modellezhetjük, például gráfok segítségével. Ezek általában súlyozott, irányított, összefüggő, hurok nélküli gráfok, melyeknek nincs izolált csomópontja. Jellemző rájuk továbbá, hogy egy

forrás- és egy nyelőcsomópontjuk van. Az 1. ábrán három gyártási feladatot látunk, melyek külön-külön is három egymásra épülő műveletből állnak, ahol  $O_{nm}$  az  $n$ . feladat  $m$ . művelete.



1. ábra. Az ütemezés irányított gráfja

Ahogy a bevezetőben taglaltuk, két alapvető gyártási elrendezést ismerünk, melyek közül jelen cikkben a szalagrendszerű gyártással (flow-shop) foglalkozunk. A szalagrendszerű gyártás ütemezési problémája a következő: legyen adott  $n$  darab gyártási tevékenység (job),  $m$  darab gép és  $m$  darab művelet. A műveletek a gépeknek megfelelően definiáltak, sorrendjük adott, végrehajtásuk nem félbeszakítható. Célunk olyan munkafolyamatsorrend meghatározása, mellyel a teljes ütemterv átfutási ideje minimális. Ez az általános forma – amely az imént felsorolt megkötéseket tartalmazza – NP-nehez, ezért megoldása nem triviális, sokszor közelítő megoldások alkalmazását indokolja. Ahogy azt a bevezetőben kifejtettük, a megoldóknak a következő főbb csoportjai ismeretesek: egzakt matematikai optimalizáció, becslő eljárás, közelítő módszerek. A megközelítésünk szempontjából nem nélkülözhető továbbá a szalagrendszerű gyártás célfüggvényeinek ismerete. Általános esetben a teljes átfutási idő minimalizálására törekszünk, tehát minél rövidebb idő alatt igyekszünk a gyártási tevékenységekkel végezni. Ha a határidőre nem tudnánk végezni az összes előírt feladattal, akkor a késés időben kifejezett mértékét próbáljuk minimalizálni. További fontos célfüggvény lehet a kapacitáskihasználás is: az adott erőforrások névleges feladatvégző képességét szeretnénk a lehető legjobb mértékben megközelíteni.

#### 3.1. Az illeszkedés alapú ütemezés

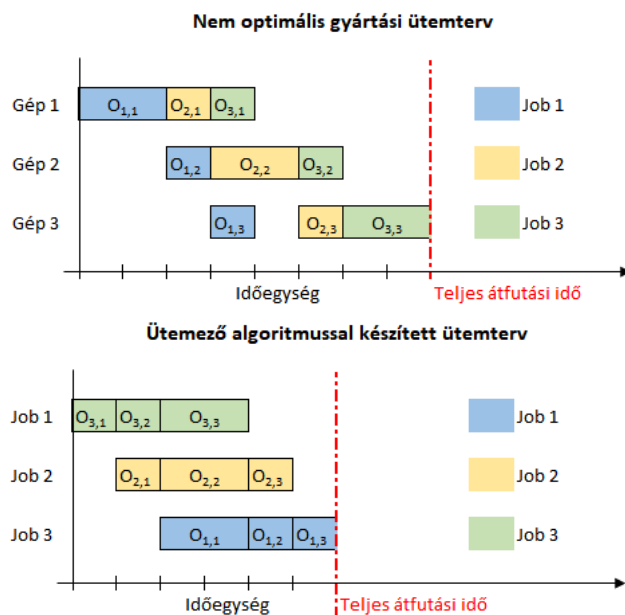
A rövid problémáismertetést követően, most definiáljuk a gyártásütemezés műveleti illeszkedésének fogalmát. Ehhez először is rávilágítunk szükségességére, majd néhány

kiegészítő fogalom segítségével definiáljuk azt és módszert javasolunk az így definiált ütemezési problémák megoldására.

Vegyük az átfutási idő szempontjából talán legkritikusabb faktort: az állásidőt. Legyen  $T_f(O_{j,i})$  a  $j$ . gyártási tevékenység  $i$ . műveletének befejezési ideje, míg  $T_s(O_{j,i+1})$  a  $j$ . gyártási tevékenység  $i+1$ . műveletének kezdési ideje. Ekkor az állásidő a következőképpen adható meg:

$$t_d = T_s(O_{j,i+1}) - T_f(O_{j,i}) \quad (1)$$

A 2. ábrán látható két ütemezés közül az elsőnél állásidő fordul elő, mely optimális ütemezéssel kiküszöbölhető és így csökken a teljes átfutási idő.

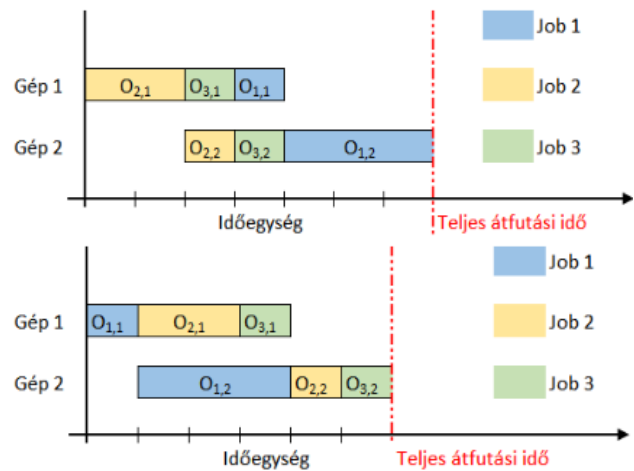


2. ábra. Állásidő kiküszöbölése optimális ütemezéssel

Sajnos az állásidő definiálása és figyelembe vétele még önmagában nem garancia az optimalitásra. Vegyük például azt az esetpárost, amelyet 3. ábra szemléltet. Jól látható, hogy bár egyik esetben sincs a műveletek között állásidő, az első esetben nagyobb a teljes átfutási idő. Az itt bemutatott esetekben tehát az állásidő segítségével nem tudunk differenciálni, mégis a célfüggvény értékében eltérés van. Az ilyen esetek megkülönböztetésére alkalmas a műveleti illeszkedés. Ez ugyanis azt fejezi ki, hogy az egyes műveletek milyen hatással vannak más műveletek műveleti (végrehajtási) idejére. Jelölje  $pt(O_{j1,i})$  a  $j_1$  tevékenység  $i$ . műveleti idejét. A műveleti

illeszkedés kiindulási alapja az egymást követő tevékenységek egymást követő műveleteinek végrehajtási idejében jelentkező különbség:

$$F = pt(O_{j1,i}) - pt(O_{j2,i+1}) \quad (2)$$



3. ábra. Illeszkedésből adódó eltérés különböző ütemezéseknél

Ez idáig analógiát mutat az előzőekben definiált állásidővel, két különbséggel. Egyrészt itt a műveleti időket vizsgáljuk, másrészt két különböző tevékenységen belül. Annak érdekében, hogy az illeszkedés számszerű értéke minél alkalmasabb legyen ütemezési problémák leírására, az előbb bemutatott kiindulási alapot még adekvát módon súlyozzuk:

$$\begin{aligned}
 F &= \left( pt(O_{j1,i}) - pt(O_{j2,i+1}) \right)^{c_2} && \text{ha} \\
 pt(O_{j1,i}) - pt(O_{j2,i+1}) &\geq 0 && \text{vagy} \\
 F &= c_1 \left| pt(O_{j1,i+1}) - pt(O_{j2,i+1}) \right|^{c_2 c_3} && \text{ha} \\
 pt(O_{j1,i}) - pt(O_{j2,i+1}) &< 0
 \end{aligned} \quad (3)$$

A fenti képlet azért tartalmazza külön esetként a negatív különbséget, mert ugyanis ekkor áll elő az a kifejezetten kerülendő helyzet, hogy  $j_2$  tevékenység  $i+1$ . művelete várakozni kényszerül, azaz állásidő keletkezik. Mivel ez súlyosan rontja az átfutási időt, ezért ezt az esetet különösen drasztikusan szükséges büntetni: erre hivatottak a  $c_1$  és  $c_3$  faktorok. A  $c_2$  faktor pedig egyszerűen átskálázza az illeszkedést, így lehetővé válik a kisebb illeszkedési eltéréseket is kellő hangsúllyal kezelni. Összességében minél kisebb az  $F$  érték, annál jobb az illeszkedés, tehát az ütemezés szempontjából a minél alacsonyabb kumulált  $F$  érték az előnyös.

(Az  $F = 0,00$  érték a tökéletes illeszkedést reprezentálja.)

Az ütemezési probléma megoldásához az illeszkedésből gyártó berendezéseken allokált műveleti idő intervallumokat kell előállítanunk. Ez azonban triviális feladattá válik, ha észre vesszük, hogy a műveleti illeszkedés által egy végrehajtási sorrend is adottá válik a tevékenységek műveletei között. Ezután már nincs más dolgunk, mint az így kialakult végrehajtási sorrendnek megfelelően ütemezni a gyártási tevékenységek műveleteit a gépekre.

Az optimalizáláshoz a gyártási tevékenységek illeszkedését is vizsgálunk kell műveleteik illeszkedésén keresztül. Ehhez egy olyan eljárást kell alkalmaznunk, amely  $j_1$  tevékenység  $\{1,2, \dots, m-1\}$  műveleteit összehasonlítja  $j_2$  tevékenység  $\{2,3, \dots, m\}$  műveleteivel, amennyiben  $j_1$  a  $j_2$  előtt közvetlenül végrehajtott gyártási tevékenység. Erre esetünkben az a legjobb megközelítés, ha a műveletek egész sorozatait hasonlítjuk össze, hiszen a szalagrendszerű gyártásban a műveleti sorrend kötött. Valamennyi művelet költséges egyenkénti összehasonlítása helyett azonban dolgozhatunk közelítő megoldásokkal melyek terméksorozatokat hasonlítanak össze egyben. Ez természetesen befolyásolja a végeredmény pontosságát, melyet részletesen bemutatunk lentebb az eredmények leírásánál.

Ezután az ütemezési problémához definiáljunk egy  $n \times n$  műveleti illeszkedés-mátrixot, amely  $F(i,j)$  eleme az  $i$ . tevékenység illeszkedését tartalmazza a  $j$ . tevékenységhez (4).

$$F_{\text{cost}} = \begin{pmatrix} \infty & \dots & F_{1,n} \\ \vdots & \ddots & \vdots \\ F_{n,1} & \dots & \infty \end{pmatrix} \quad (4)$$

Vegyük észre, hogy  $F(i,j) \neq F(j,i)$ , tehát a műveleti illeszkedés nem kommutatív. Az  $F$  érték jellegénél fogva az átlókba tetszőlegesen nagy értéket tehetünk, hiszen egy gyártási tevékenységet sem követheti saját magát. Ezzel a mátrixszal egy teljes gráfot definiálhatunk adekvát élsúlyozással. Az illeszkedés alapú ütemezési probléma pedig átalakult egy súlyozott gráfbejárás problémára, ahol minden csomópont meglátogatásával jelöljük ki a

végrehajtási sorrendet, melynek jóságát pedig a bejárt élek súlya adja meg. Ennek minimalizálását már könnyen elvégezhetjük, hiszen a jól ismert utazó ügynök problémáról van szó.

Az utazó ügynök problémára használt algoritmus kiválasztása során elsődleges szempont volt, hogy egy olyan algoritmust használjunk, amely a kutatásban eddig használt módszerekhez jól illeszkedik. Választásunk a Nicos Christofides által publikált Christofides heurisztikus algoritmusra esett [15].

Az algoritmus futási ideje gyors –  $O[n^2 \log_2(n)]$  – és garantálja, hogy az algoritmus által talált célérték maximum másfélszerese az optimális célértéknek.

#### 4. EREDMÉNYEK

A közölt futási eredmények egy Intel (R) Core (TM) i5-2410M típusú / 2.30 GHz processzoron, 4 GB RAM-on, 5400 rpm fordulatszámú háttértáron 64-bites Windows 10 operációs rendszeren, kerültek kiszámításra. A  $c_1$  változtatásával azokat az eseteket emeljük ki, ahol a követő munkafolyamatnak nagy valószínűséggel várakoznia kell. Teszteléshez az E. Taillard [16] által közölt példafeladatokat használtuk, azok bizonyított és ismert minimális megoldási értékeivel. A  $c_1$  paraméter hatását a végeredményre az 1. táblázat szemlélteti. Az első sor  $c_1$  beállított értékét, míg a 2. sor a tesztesetknél megtalált szakirodalomból ismert optimális értékek számát mutatja adott  $c_1$  beállítás mellett.

1. táblázat. A  $c_1$  paraméter hangolása

$c_1$	0,5	1,5	2,5	3,5	4,5
N	73 db	67 db	62 db	66 db	66 db

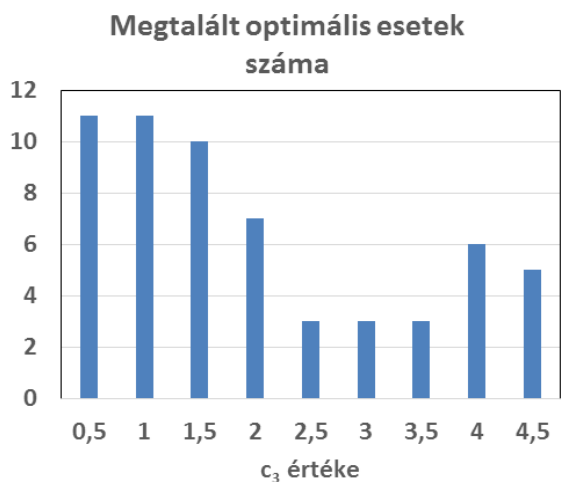
A  $c_2$  paraméter hangolása során a "kiseltérés-nagyeltérés" fogalmat vizsgáltuk, azaz hogy érdemes-e a nagyobb eltéréseket jóval nagyobb mértékben büntetni az illeszkedésszámításnál, illetve esetleg elhanyagolni a nagyon kis eltéréseket. Az eredményeket az 2. táblázat összegezi, mely második sorában megmutatja, hogy összesen hány esetben találtuk meg az irodalomban is közölt minimális átfutási idejű ütemterveket a paraméter változtatásának függvényében. Minél nagyobb a megtalált optimális esetek száma, annál hatékonyabban működik a módszerünk.

**2. táblázat.** A  $c_2$  paraméter hangolása

$c_2$	0,5	1	2
Megtalált optimális esetek száma	2 db	20 db	8 db

Mivel a  $c_2 = 1,00$  kitevőérték alkotta legtöbbször a legjobb ütemterveket, ezért úgy tűnik, hogy a jó és rossz illeszkedések egyértelmű elkülönítése már legtöbbször megtörténik a paraméter változtatása nélkül is (tehát annak elhagyásával). Megfigyelhető ugyanakkor, hogy néhány szélsőséges esetben érdemes lehet  $c_2$  értékét növelni.

A  $c_3$  paraméter vizsgálata során azt tapasztaltuk, hogy csak bizonyos esetekben figyelhető meg ezen büntetés jótékony hatása, mivel a  $c_3 = 0,5; 1$  és  $1,5$  értékek esetén közel azonos számban értük el a legjobb átfutási idejű ütemterveket. A paraméter finomhangolása során a legjobb átfutási idejű ütemtervek tipikusan akkor készültek el, amikor  $c_3 = 0,5; 1$  vagy  $1,5$  értéket választottuk. Ezen értékek között nem tapasztaltunk jelentős különbségeket, viszont a  $c_3$  érték további növelésével csökken a megtalált minimális átfutási idejű ütemtervek száma. Az eredményeket a 4. ábra szemlélteti.



**4. ábra.** A  $c_3$  értékének hatása a működésre

A 3. táblázatban módszerünk hatékonysága látható a szakirodalomban közölt legjobb eredményekkel összevetve. A táblázat fejléc értékei a következők: E. Taillard [16] által közölt példafeladatok sorszáma (Id), az irodalomban publikált és bizonyított minimális átfutási idejű értékek az egyes példafeladatokon (LM), és az általunk talált legjobb eredmények hányadosa (E). Megfigyelhető hogy az optimálisan elérhető célértéktől minden esetben kevesebb, mint 20%-kal

tértünk el az egyébként egyszerű és gyorsan meghatározható illeszkedésen alapuló megközelítésünkkel. Összesítve átlagosan 8%-al hosszabb átfutási idejű ütemtervek készültek.

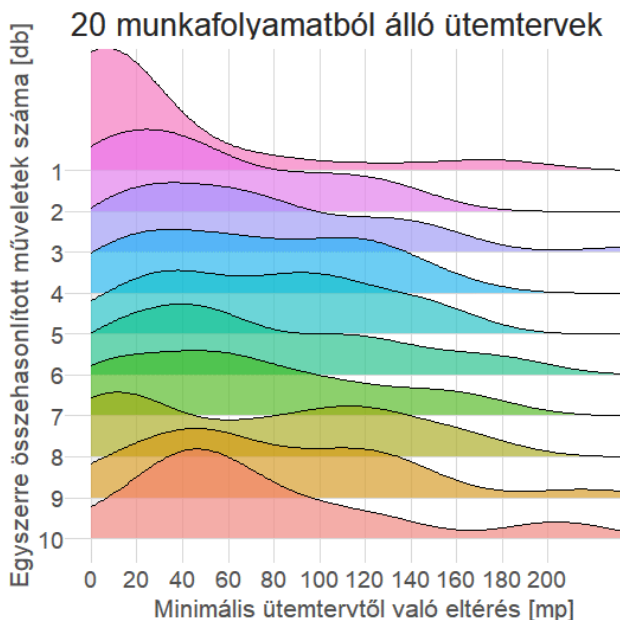
**3. táblázat.** Eredményeink összehasonlítása, az irodalomban közölt optimummal

Id	LM	E	Id	LM	E	Id	LM	E
1	1278	1,04	31	2724	1,03	61	5493	1,01
2	1359	1,02	32	2834	1,03	62	5268	1,02
3	1081	1,13	33	2621	1,03	63	5175	1,02
4	1293	1,08	34	2751	1,01	64	5014	1,02
5	1235	1,05	35	2863	1,02	65	5250	1,03
6	1195	1,09	36	2829	1,04	66	5135	1,01
7	1234	1,06	37	2725	1,03	67	5246	1,02
8	1206	1,10	38	2683	1,05	68	5094	1,02
9	1230	1,08	39	2552	1,05	69	5448	1,03
10	1108	1,07	40	2782	1,04	70	5322	1,03
11	1582	1,15	41	2991	1,14	71	5770	1,07
12	1659	1,11	42	2867	1,11	72	5349	1,07
13	1496	1,09	43	2839	1,15	73	5676	1,05
14	1377	1,11	44	3063	1,10	74	5781	1,05
15	1419	1,12	45	2976	1,09	75	5467	1,10
16	1397	1,13	46	3006	1,11	76	5303	1,08
17	1484	1,06	47	3093	1,09	77	5595	1,06
18	1538	1,11	48	3037	1,07	78	5617	1,07
19	1593	1,08	49	2897	1,11	79	5871	1,05
20	1591	1,10	50	3065	1,10	80	5845	1,05
21	2297	1,07	51	3850	1,13	81	6202	1,14
22	2099	1,08	52	3704	1,13	82	6183	1,15
23	2326	1,06	53	3640	1,10	83	6271	1,14
24	2223	1,06	54	3723	1,13	84	6269	1,10
25	2291	1,08	55	3611	1,17	85	6314	1,13
26	2226	1,10	56	3681	1,13	86	6364	1,12
27	2273	1,07	57	3704	1,16	87	6268	1,08
28	2200	1,05	58	3691	1,14			
29	2237	1,09	59	3743	1,14			
30	2178	1,09	60	3756	1,13			

Vizsgáltuk továbbá az  $F_{cost}$  mátrix illeszkedési értékeinek pontosságát különböző bonyolultságú számítási módszerek esetén is. Megfigyelhető, hogy a legtöbb minimális átfutási idejű ütemtervet az 1-1 összehasonlítás során kapjuk eredményül. Ugyanakkor ez a módszer jóval több számítási időt igényel, mint ha egyszerre több műveletet hasonlítanánk össze az  $F_{cost}$  mátrix illeszkedési

értékeinek meghatározásánál. Így érdemes vizsgálni mennyit romlik a várható végeredmény, ha az optimalizálás sebességét próbáljuk növelni az egyszerre vizsgált esetek számának növelésével.

Az egyszerre összehasonlított műveletek számának befolyását az 5. ábra szemlélteti. Látható hogy a legjobb megoldást az egyenkénti összehasonlítás adja. Ekkor tudtuk a legjobban megközelíteni a szakirodalomban közölt optimális minimum átfutási időket az ütemezésünkkel. Természetesen ez jelentősen több futásidőbe kerül, mintha teljes sorozatokat hasonlítanánk össze az egyszerre összehasonlított műveletek számának növelésével. Az 5. ábrán az is látható, hogy a növeléssel együtt fokozatosan romlik a végeredményben kapott ütemezések jósága is a tesztfeladatok eredményeinek átlagát tekintve.



5. ábra. Egyszerre összehasonlított műveletek vizsgálata 20 job-ból álló feladatokon

## 5. ÖSSZEFOGLALÁS

Egy egyszerű, ugyanakkor hatékony módszert mutattunk be a szalagrendszerű gyártás optimális ütemezésére jelen cikkben. Az gyártási tevékenység műveleteinek illesztésén alapuló heurisztikus megközelítésünk az utazó ügynök probléma megoldására is építve gyorsan képes jó eredményeket adni. Az elért eredményeinket a szakirodalomban ismert E. Taillard [16] által publikált tesztfeladatok bizonyított optimális megoldásaival hasonlítottuk össze. Átlagosan 8 százalékos, de maximum 20 százalékos eltéréssel

sikerült közelítenünk ezen eredményeket, bizonyítva ezzel, hogy módszerünk alkalmas a szalagrendszerű gyártásütemezés átfutási idejének gyors minimalizálására.

További kutatásainkban még részletesebben szeretnénk vizsgálni az egyes paraméterek eredményre gyakorolt hatását. Vizsgálnánk továbbá más megoldásokat is az utazóügynök probléma kezelésére is a jelenleg használt Christofides-féle megközelítés helyett illetve mellett. A jövőben szeretnénk elemezni még módszerünk működését más a szakirodalomban elterjedt ütemezési feladatok végrehajtásával is.

## IRODALOMJEGYZÉK

- [1] Graves S.C., „A Review of Production Scheduling”, *Oper. Res.*, 29. évfolyam, 4. szám, 646-675 o., 1981. <https://doi.org/10.1287/opre.29.4.646>
- [2] Nahmias S., Olsen T.L., „Production and operations analysis”, *Waveland Press*, USA, 2015.
- [3] Błażewicz J., Domschke W., Pesch E., „The job shop scheduling problem: Conventional and new solution techniques”, *Eur. J. Oper. Res.*, 93. évfolyam, 1. szám, 1-33 o., 1996. [https://doi.org/10.1016/0377-2217\(95\)00362-2](https://doi.org/10.1016/0377-2217(95)00362-2)
- [4] Dudek R.A., Panwalkar S.S., Smith M.L., „The Lessons of Flowshop Scheduling Research”, *Oper. Res.*, 40. évfolyam, 1. szám, 7-13 o., 1992. <https://doi.org/10.1287/opre.40.1.7>
- [5] Graham R.L., Lawler E.L., Lenstra J.K., Kan A.H.G.R., „Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey”, *Ann. Discrete Math.*, 5. évfolyam, 287-326 o., 1987. [https://doi.org/10.1016/S0167-5060\(08\)70356-X](https://doi.org/10.1016/S0167-5060(08)70356-X)
- [6] Harjunkoski I. és tsai., „Scope for industrial applications of production scheduling models and solution methods”, *Comput. Chem. Eng.*, 62. évfolyam, 161-193 o., 2014. <https://doi.org/10.1016/j.compchemeng.2013.12.001>
- [7] Bierwirth C., Mattfeld C., „Production Scheduling and Rescheduling with Genetic Algorithms”, *Evol. Comput.*, 7. évfolyam, 1. szám, 1-17 o., 1999. <https://doi.org/10.1162/evco.1999.7.1.1>
- [8] Rajendran C., Chaudhuri D., „Heuristic algorithms for continuous flow-shop problem”, *Nav. Res. Log.*, 37. évfolyam, 5. szám, 695-705 o., 1990. [https://doi.org/10.1002/1520-6750\(199010\)37:5<695::AID-NAV3220370508>3.0.CO;2-L](https://doi.org/10.1002/1520-6750(199010)37:5<695::AID-NAV3220370508>3.0.CO;2-L)
- [9] Etiler O., Toklu B., Atak M., Wilson J., „A genetic algorithm for flow shop scheduling problems”, *J. Oper. Res. Soc.*, 55. évfolyam, 8. szám, 830-835 o., 2004. <https://doi.org/10.1057/palgrave.jors.2601766>

- [10] Li B.B., Wang L., „A Hybrid Quantum-Inspired Genetic Algorithm for Multiobjective Flow Shop Scheduling”, *IEEE T. Syst. Man Cy. B*, 37. évfolyam, 3. szám, 576-591 o., 2007.  
<https://doi.org/10.1109/TSMCB.2006.887946>
- [11] Dai M., Tang D., Giret A., Salido M.A., Li W.D., „Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm”, *Rob. Cim-Int. Manuf.*, 29. évfolyam, 5. szám, 418-429 o., 2013.  
<https://doi.org/10.1016/j.rcim.2013.04.001>
- [12] Ramanan T.R., Sridharan R., Shashikant K.S., Haq A.N., „An artificial neural network based heuristic for flow shop scheduling problems”, *J. Intell. Manuf.*, 22. évfolyam, 2. szám, 279-288 o., 2011.  
<https://doi.org/10.1007/s10845-009-0287-5>
- [13] Linn R., Zhang W., „Hybrid Flow Shop Scheduling: A survey”, *Comput. Ind. Eng.*, 37. évfolyam, 57-61 o., 1999.  
[https://doi.org/10.1016/S0360-8352\(99\)00023-6](https://doi.org/10.1016/S0360-8352(99)00023-6)
- [14] Riba I., Leisten R., Framiñanc J.M., „Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective”, *Comput. Oper. Res.*, 37. évfolyam, 8. szám, 1439-1454 o., 2010.  
<https://doi.org/10.1016/j.cor.2009.11.001>
- [15] Goodrich M., Tamassia R., „The Christofides Approximation Algorithm”, *Algorithm Design and Applications*, Wiley, 513-514 o., 2015.
- [16] Taillard E., „Benchmarks for basic scheduling problems”, *Eur. J. Oper. Res.*, 64. évfolyam, 2. szám, 278-285 o., 1993.  
[https://doi.org/10.1016/0377-2217\(93\)90182-M](https://doi.org/10.1016/0377-2217(93)90182-M)