AZ IGÉNYSZERINTI TÖMEGGYÁRTÁS TERMELÉSPROGRAMOZÁSÁNAK MODELLEZÉSE ÉS HEURISZTIKUS MEGOLDÁSA

*Kulcsár Gyula**

ABSTRACT: The paper discusses the shop floor scheduling problem of customized mass production, which can be described as combining make to stock and make to order type production at the same time. The focus is set to an extended flexible flow shop scheduling model which supports alternative routes, unrelated parallel machines, sequence dependent setup times, product dependent production rates and limited machine availabilities. The paper outlines a new approach to solve the problem. A computer framework based on a new data model and heuristic algorithms are developed to determine feasible schedule for customized mass production according to various objectives. The paper presents the details of the scheduling model and outlines the basic steps of the proposed method.

1. BEVEZETÉS

Napjainkban a termelő nagyvállalatok egy része közvetlenül a vevők igényeit próbálja meg kiszolgálni (pl.: háztartási gépek, lámpák gyártása). A versenyképesség növelése érdekében szükséges, hogy minél jobban alkalmazkodjanak a piaci körülmények gyors változásaihoz. Ennek érdekében a gyártási hatékonyságot és a szállítóképességet kell javítani [12], [13]. Az előbbit az erőforrások minél jobb kihasználásával, az utóbbit pedig, előzetes becslésen alapuló raktárra történő gyártással tudják növelni. A vállalatok sikeressége nagymértékben a megrendelők igényeinek magas szintű kielégítésén múlik, ennek egyik fontos feltétele a megrendelésekhez tartozó munkák ütemezésének minél hatékonyabb megoldása [8].

Egy általánosan megfogalmazott ütemezési feladat megoldása az erőforrások allokálásának és a munkák időbeli végrehajtásának megtervezését jelenti, annak érdekében, hogy a kitűzött célok megvalósuljanak. Ezek a célok sokfélék lehetnek, és időről időre változhatnak követve a megfogalmazott elvárásokat.

A folyamat közeli ütemezés a MES (Manufacturing Execution System) szintű, rövid távú valósidejű tervezési feladat megoldását követeli meg [2]. Ezen a szinten az ütemezési feladat azt jelenti, hogy ismert és pontosan meghatározott (technológiai, anyag stb.) korlátozások figyelembe vételével, a gyártásra kiadott, ismert belső rendeléseken alapuló munkák elvégzéséhez gyártási erőforrások (gépek, eszközök, stb.) allokálását és munkák indítási időpontjának sorozatát kell megtervezni úgy, hogy a definiált korlátozások teljesüljenek, és a vállalat magasabb szintjén megfogalmazott célok megvalósuljanak.

2. ISMERT ÜTEMEZÉSI MODELLEK

A diszkrét gyártási folyamatokban a termékek elkészítése sorozatokban, kötegekben történik. A sorozatok nagysága nagyon változó lehet, egyetlen terméktől (pl.: nagy értékű alkatrészek vagy teljes egyedi gépek gyártása) több ezer, sőt több millió darab termékig is terjedhet (pl.: egyszerű komponensek, alkatrészek tömeggyártása esetén) [15].

A diszkrét gyártási folyamatokban a gyártás rendszerint elkülönült gépeken zajlik. Attól függően, hogy az egyes műveletek a gyártó (szerelő) műhely gépein hogyan kerülnek végrehajtásra, megkülönböztethetünk soros illetve nem soros elrendezésű gyártórendszereket [14]. Tömeggyártás és igény szerinti tömeggyártás során a termékek gyártási folyamata az első kategóriába tartozik [7].

A soros elrendezésű gyártási struktúrához kapcsolódó alapmodellben (Flow Shop) adott számú különböző gép a technológiai sorrendnek megfelelően egymás után van elhelyezve. A modellt erősen befolyásolja a gépek közötti munkadarab tárolás lehetősége. Ha a gépek közötti átmeneti tárolók kapacitás nulla, akkor szinkronizált gyártó (szerelő) rendszerről van szó. Véges kapacitású bufferek esetén a gyártás asszinkron jellegű [11].

A termelés során munkakötegek végrehajtásáról kell gondoskodni. Ha megengedett hogy az egyes gépeken a munkák sorrendje eltérő legyen, akkor előzéses, ellenkező esetben előzésmentes modellről van szó. Ezekről a modellekről és megoldási módszereikről számos munka nyújt részletes áttekintést (pl.: [4], [5], [10], [16], [20]).

Az alapmodellnek párhuzamos gépekkel történő kibővítéseként ismert a rugalmas egyutas (Flexible Flow Shop) modell. Ebben a modellben összetett munkahelyek vannak definiálva. Minden egyes munkahelyen adott számú egymással teljesen egyenértékű gép található. A gépek száma munkahelyenként eltérő lehet. Az egymást követő munkahelyek között korlátlan méretű átmeneti tároló helyek kialakítását tételezhetjük fel. A munkadarabokat minden munkahelyen, annak egy kiválasztott gépén kell megmunkálni. Ebben a modellben tehát megjelenik a gépválasztás feladata is, ugyanakkor továbbra is fontos szerepet játszik a munkák gépenkénti sorrendjének meghatározása.

**A Hatvany József Informatikai Tudományok Doktori Iskola volt hallgatója, egyetemi tanársegéd,*
Miskolci Egyetem, Informatikai Intézet, Alkalmazott Informatikai Tanszék. email: kulcsar@ait.iit.uni-miskolc.hu
Témavezető: Dr. Erdélyi Ferenc CSc, tudományos főmunkatárs, a Magyar Tudományos Akadémia Termelésinformatikai Kutatóhelye (PIERT).

A szakirodalomban számos cikk különböző mélységekben foglalkozik a rugalmas Flow Shop ütemezési modellel (pl.: [1], [3], [5], [9], [17], [18], [19]). A témakörhöz tartozó cikkek többsége célfüggvényként a rendeléscsoport legkésőbbi befejezési idejét (makespan) használja. Sokkal kevesebb eredmény ismert határidős gyártás (make to order) esetén a késésekkel kapcsolatos célfüggvények minimalizálására.

A jelenleg ismert többgépes modellek nem tudják figyelembe venni egyszerre az igény szerinti tömeggyártás jellegzetességeit: több művelet együttes végrehajtására képes gépeket, technológiai útvonal alternatívákat, gépek változó rendelkezésre állási időintervallumait, eltérő termelési sebességeket és sorrendfüggő átállási időket, így szükség van ezeknek a modelleknek a kiterjesztésére, továbbfejlesztésére.

3. A VIZSGÁLT GYÁRTÁSI FOLYAMAT

Az igény szerinti tömeggyártást folytató vállalatok különböző típusú termékeket állítanak elő. Jellemzően, adott egy rendelés állomány, amelyet a külső rendelések és az előrejelzések figyelembe vételével a vállalat termeléstervezés szintjén definiálnak. Minden egyes rendelés meghatározott típusú, adott darabszámú egyforma termék, adott határidőre történő legyártását igényli. A műhelyszintű irányításban a logisztikai egységet a paletta jelenti. Minden egyes palettára előre megadott darabszámú, azonos terméket lehet elhelyezni. Egy rendelés paletták halmazának is tekinthető, ahol a paletták számát a rendelt mennyiség és a paletták termékfüggő mérete együttesen határozza meg.

Az általunk vizsgált gyártási struktúra gyártó-szerelő jellegű, automatizált gépcsoportokból (gépekből és/vagy gépsorokból) áll. A termékek előállításához legfeljebb négy, vagy annál kevesebb technológiai lépést kell kötött sorrendben végrehajtani.

A technológiai lépések operációk sorozatából tevődhetnek össze. Mivel a technológiai lépések nem megszakíthatók, ezért ezek tekinthetők az ütemezés során a legkisebb allokációs egységeknek. Az egyes technológiai lépések elvégzéséhez – a megfelelő gépeken túlmenően – meghatározott anyagoknak, komponenseknek is rendelkezésre kell állni adott mennyiségben. A gyártás rugalmas jellegéből következik, hogy egy adott típusú termék különféle anyagok, komponensek, gépek és útvonalak használatával is előállítható.

Az ütemezési modellben szereplő gépek – amelyek valójában különböző gyártási képességekkel rendelkező gépsorok – a terméktől függő gyártási sebességekkel, a munkák sorrendjétől függő átállási időkkel, rendelkezésre állási időintervallumokkal és egy adott termékcsoportra érvényes technológiai lépések egymást követő sorozatával jellemezhetők.

A gyártás során egy adott gép egyszerre csak egy munkán dolgozhat, és egy adott munka egyszerre csak egy géphez lehet hozzárendelve. A gépek közötti tároló helyek mérete nem korlátozott.

Az ütemterv elkészítésekor azt is figyelembe kell venni, hogy az ütemezési időhorizonton a műhely bizonyos gépei már korábbi, még be nem fejezett, nem módosítható feladatokkal terheltek, így az utolsó megerősített és érvényben lévő ütemterv következményei hatással vannak az új ütemtervre.

4. KITERJESZTETT ÜTEMEZÉSI MODELL

Az ütemezési feladatok formális leírásának eszközeként a szakirodalomban az α|β|γ formalizmus használata a legismertebb [1]. A szimbólumok jelentése a következő:

- α: erőforrás környezetet leíró paraméter lista,
- β: korlátozásokat, végrehajtási jellemzőket leíró paraméter lista,
- γ: célfüggvényeket kitűző paraméter lista.

A listákban szereplő paraméterekre számos javaslat van az irodalomban.

A vizsgált ütemezési feladat osztályát (Extended Flexible Flow Shop, EFFS) a következő szimbolikus formában írhatjuk le:

$$M, Q, F, Set_{i,j,m}, Cal, 4 \left| R_i, D_i \right| f \qquad (1)$$

Az egyes szimbólumok jelentése a következő:

$\alpha = [\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6]$

α_1 : Az erőforrás jellege. *M*: többcélú párhuzamos gépek megengedettek.

α_2 : Az alternatív gépek jellege. *Q*: eltérő sebességű (intenzitású) gépek.

α_3 : A műveleti sorrend előírása. *F*: a műveletek (technológiai lépések) sorrendje kötött.

α_4 : A gépek átállítására vonatkozó előírás.

$Set_{i,j,m}$: munkák sorrendjétől és géptől függő időadatok.

α_5 : Erőforrásokra vonatkozó speciális előírások.

Cal: Gépekre előírt rendelkezésre állási időintervallumok. A gépek csak ezeken belül működhetnek.

α_6 : A munkahelyek maximális száma.

4: a független technológiai lépések száma négy.

$\beta = [\beta_1, \beta_2]$.

β_1 : A munkák indíthatóságára vonatkozó előírás.

R_i: munkánként előírt legkorábbi indítási időpont.

β_2 : A munkák befejezésére vonatkozó előírás.

D_i: munkánként előírt legkésőbbi befejezési időpont.

γ : A kijelölt célfüggvény, amelynek a minimumát keressük.

Az ütemtervek jóságát, minőségét célfüggvények megfogalmazásával és azok kiértékelésével lehet számszerűsíteni. Valós környezetben nagyon sokféle termelési cél megfogalmazható [6]. Ismeretes, hogy a termelési modellek széles körében az ismert termelési háromszög állapotváltozói: a lekötött készletszint, az erőforrás-kihasználás és a szállítókészség szükséges és elegendő mértékben meghatározza a termelési folyamat minősítését [14].

A szállítókészség javítására irányuló célfüggvényeket tovább bonthatjuk:

- határidőhöz kapcsolódó és
- határidőhöz nem kapcsolódó

célfüggvényekre.

A határidőhöz kapcsolódó célfüggvények megfogalmazásához figyelembe kell venni a következőket:

Adottak a J_i $(i=1,\ldots,n)$ munkák. Minden egyes J_i munkához tartozik D_i határidő (a legkésőbbi időpont amikorra a munkának el kell készülnie) és R_i legkorábbi indítási időpont (legkorábbi időpont amikor a munka végrehajtása elkezdődhet). A munka tényleges indítási időpontját S_i –vel, a tényleges befejezési időpontját C_i -vel jelölve definiálható minden munkához a:

Határidőtől való eltérés: $L_i = C_i - D_i$. (2)

Késés: $T_i = \max(0, L_i)$. (3)

Korai befejezés: $E_i = \max(0, -L_i)$. (4)

Ezeket G_i függvényeknek tekintve, a leggyakrabban alkalmazott célfüggvények kifejezhetők a következő módon:

Legnagyobb érték: $\gamma = \max_i(G_i)$. (5)

Összeg: $\gamma = \sum_i G_i$. (6)

Átlag: $\gamma = \dfrac{\sum_i G_i}{n}$. (7)

Továbbá határidőhöz kapcsolódó fontos célfüggvény még a:

Késő munkák száma: $\gamma = |\{i \mid T_i > 0\}|$. (8)

A munkák fontosságának kifejezésére w_i súlyokat lehet a munkákhoz hozzárendelni. A célfüggvényekben ezek figyelembe vehetők például a következő módon:

Súlyozott maximum: $\gamma = \max_i(w_i G_i)$. (9)

Súlyozott összeg: $\gamma = \sum_i w_i G_i$. (10)

Súlyozott átlag: $\gamma = \dfrac{\sum_i w_i G_i}{n}$. (11)

A leggyakrabban használt nem határidőhöz kapcsolódó célfüggvények a következők:

Legkésőbbi befejezési időpont:

$$\gamma = \max_i(C_i). \quad (12)$$

Összes munka átfutási ideje:

$$\gamma = \max_i(C_i) - \min_i(R_i). \quad (13)$$

Minden J_i munkára kiszámítható a saját átfutási ideje:

Átfutási idő: $F_i = (C_i - R_i)$. (14)

Ezt az F_i átfutási időt behelyettesítve a (5.) (6.) (7.) illetve a (9.) (10.) (11.) kifejezésekbe újabb nem határidőhöz kapcsolódó célfüggvényeket kapunk.

A feladatosztály leírása jól mutatja az ütemezési feladatok sokszínűségét és erős modell függését.

5. AZ ÜTEMEZÉSI FELADAT HEURISZTIKUS MEGOLDÁSA

Az ismertetett ütemezési feladat kombinatorikus tulajdonságai miatt az NP teljes feladatosztályba tartozik. Ezért az elméleti globális optimum keresése helyett, olyan megoldási módszerekkel foglalkozunk, amelyek nagy méretű feladatok (pl. 500 megrendelés, 3000 munka, 120 gép) esetén is elfogadható időn belül megvalósítható ütemtervet állítanak elő, úgy hogy a választott célfüggvény értéke minél közelebb legyen az optimumhoz.

5.1 Megoldási koncepció

A lehető legtágabb értelmezés szerint minden fizikai palettát önálló absztrakt munkának tekintünk. A feladat megoldása során ilyen munkák kerülnek ütemezésre. Nem használunk előre rögzített gyártási sorozatnagyságot. A gyártási sorozatnagyságot az azonos beállítással készülő munkák száma adja meg. A rendelési sorozatnagyság ennél nagyobb vagy kisebb is lehet. Az ütemezés során az egyes gépeken kialakuló gyártási sorozatnagyság az egy munkaköteg, amely két egymást követő beállítási időintervallum közötti azonos termékre vonatkozó rendelt munkák sorozata. A köteg nagysága tehát dinamikusan változhat, megengedve a rendelési sorozat bontását és egyesítését.

A modellben feltételezzük, hogy az ütemezés előtt minden munkához hozzárendelhető egy korlátozás, legkorábbi kezdési időpont, amely a kezdő technológiai lépés indíthatóságának időbeli korlátozására vezeti vissza a gyártási-szerelési komponensek rendelkezésre állását.

Az ütemezés alapegységének nem a technológiai lépést, hanem a végrehajtási lépést tekintjük. Bevezetjük a végrehajtási lépés fogalmát, amely alatt a technológiai lépések olyan sorozatát kell érteni, amely gépváltás nélkül elvégezhető. Egy multifunkcionális gépsorhoz tartozó végrehajtási lépés több technológiai lépést tartalmaz, míg egy hagyományos géphez tartozó csak egyet.

A végrehajtási útvonal (műveleti sorrendterv) fogalmát a végrehajtási lépések olyan sorozataként definiáljuk, amelyben az előforduló technológiai lépések csak egyszeresen vannak lefedve, azaz a végrehajtási lépések közös része üres halmaz.

A végrehajtási lépések típusai alapján gépcsoportokat (géptípus osztályokat) definiálunk. A gépcsoportokon belül különböző darabszámú és gyártási képességű gépek lehetnek. Mivel a gépek nem minden terméktípust képesek kezelni, így a gépcsoportok terméktípusonként (vagy terméktípus csoportonként) eltérőek.

Egy ütemterv elkészítésekor minden egyes munkához:

1. hozzárendelünk egy megfelelő végrehajtási útvonalat,
2. hozzárendelünk egy megfelelő gépet a kiválasztott útvonal minden egyes végrehajtási lépésének megfelelő gépcsoportból,
3. meghatározzuk minden hozzárendelt gépen a végrehajtási sorban elfoglalt pozícióját,
4. végül definiáljuk minden hozzárendelt gépen a kezdési időpontját és kiszámítjuk az előzőek ismeretében az időadatait.

A munkák időadatainak számítására és az ütemtervek kiértékelésére gyors szimulációs eljárást használunk. A szimuláció figyelembe veszi az egyes gépek rendelkezésre állási időintervallumait, az egyes gépeken az adott munkák sorrendje által meghatározott átállítási időket, a munka-gép összerendelések alapján számítható megmunkálási időket. A munka aktuális besorolása (korlátozások: időablak, sorrend, erőforrás, útvonal) alapján ismertté válik az egyes munkák gépenkénti indítási időpontja és a befejezési időpontja. Ezek felhasználásával végül a célfüggvények kiértékelésére kerül sor.

A változó célokhoz való rugalmas alkalmazkodás úgy valósul meg, hogy előre definiálunk célfüggvényeket és ezek közül az ütemezés indításakor kiválasztható az aktuális célfüggvény.

5.2 Adatmodell

A kidolgozott modell egyszerűsített vázának bemutatására a számítógépi programozásban általánosan alkalmazott tömbök és struktúrák jelölésrendszerét használjuk.

A modellben indexszelt tömböket használunk az adatelérés gyorsítása érdekében. Ezekben a tömbökben a gyártási környezetre jellemző entitások (pl.: megrendelések, munkák, gépek stb.) indexekkel vannak helyettesítve. Ezek nem negatív egészszámok. Az indexek hivatkoznak az egyes objektumok vektorokban elfoglalt pozíciójára. Ezáltal lehetővé válik az, hogy egy adott tömb adott elemére történő hivatkozásban a tömb indexe ugyanannak vagy egy másik tömbnek az egyik értéke legyen. Az összetartozó értékeket a bázisvektorokban tárolt alapadatokra történő hivatkozások rendszere tartja össze. Ennek előnye főként az, hogy egy tetszőleges elemből kiindulva – az indexelési szabályok betartásával – minden egyes vele kapcsolatban álló elem közvetlenül, keresés nélkül elérhető.

A tömbök elemeire történő hivatkozás általános formalizmusa a következő:

TÖMB_NÉV[INDEX]
TÖMB_NÉV[SOR_INDEX][OSZLOP_INDEX]
TÖMB_NÉV[DIM_1_INDEX][DIM_2_INDEX] [...] [DIM_N_INDEX]

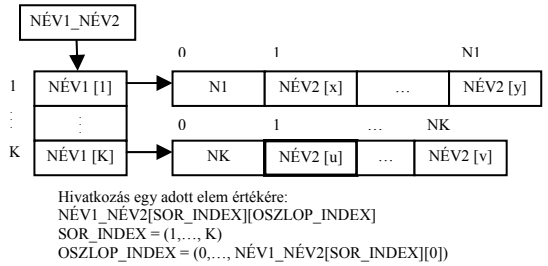
Olyan esetekben, amikor a tömb elemei mezőkből felépülő struktúrák (összetett adatszerkezetek), az egyes mezőkre a pont operátor és a mezőnév együttes használatával lehet hivatkozni:

TÖMB_NÉV[INDEX].MEZŐ_NÉV
TÖMB_NÉV[SOR_INDEX][OSZLOP_INDEX].MEZŐ_NÉV
TÖMB_NÉV[DIM_1_INDEX] [...] [DIM_N_INDEX] .MEZŐ_NÉV

A struktúrák mezői között tömbök is szerepelhetnek:

TÖMB_NÉV[INDEX_1].MEZŐ_NÉV[INDEX_2] [INDEX_3]

Egy soronként eltérő elemszámú kétdimenziós tömb általános szerkezetét az 1. ábra szemlélteti. Ilyen szerkezeteket használunk két különböző (*NÉV1 és NÉV2*) tömb elemeinek összerendelésére, ahol a *NÉV1* az alap tömböt, *NÉV2* pedig a hozzákapcsolt tömböt jelenti.



1. ábra: Kapcsolótömb általános szerkezete

A bemutatott formalizmus használatával az ütemezési modell entitásai a következőképpen írhatók le:

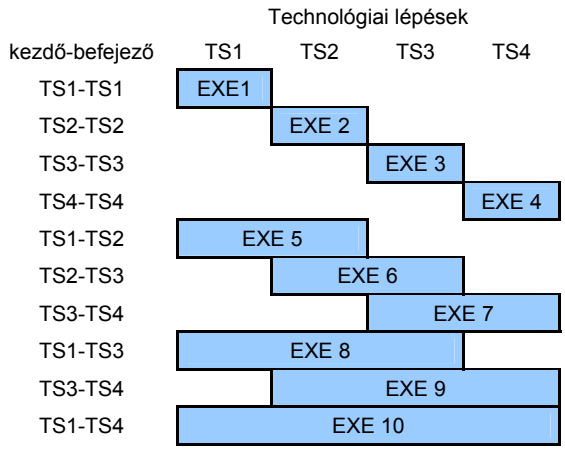
A gyártható terméktípusok p $(p = 1,\ldots, N_P)$. A termékek alap attribútumai a következők:

P[p].BOM a termék darabjegyzék-azonosítója. A lehetséges komponens beépülési kombinációk adottak egy anyagjegyzékben, amelyben *ÉS* logikai operátorok továbbá *VAGY* logikai operátorok kapcsolják össze a komponenseket.

P[p].EXE a termék gyártási folyamatának azonosítója. A termékek legyártásához kötött sorrendű technológiai lépéseket ts $(ts = 1,\ldots, 4)$ kell végrehajtani. A különböző termékek előállításához a teljes technológiai lépéssorozatot vagy annak egy összefüggő részsorozatát kell végrehajtani (2. ábra).

P[p].S a termék gyártásakor előírt gépbeállítás-típus azonosítója. A különböző termékek előállításához a gépeket megfelelő módon elő kell készíteni a munkák végrehajtása érdekében. Ez az azonosító az egyes beállítási operációkat és az előírt paraméterértékeket azonosítja.

P[p].Q legkisebb gyártható darabszám (paletta mérete). A gyártórendszerben paletták mozoghatnak. A paletták mérete előre megadott, amely termékfüggő darabszámot jelent.



2. ábra: Gyártási folyamat azonosítók

A rendelés állomány belső rendeléseket *o* $(o = 1,\ldots, N_O)$ tartalmaz. A rendelések alap attribútumai a következők:

O[o].P a gyártandó terméktípus azonosítója.

O[o].NP a megrendelt darabszám.

O[o].CET az előírt legkésőbbi befejezési időpont.

Minden egyes *o* rendelést munkákra bontunk, ahol a munkák számát a rendelt mennyiség és a paletták termékfüggő mérete együttesen határozza meg. A rendelésekhez további származtatott attribútumok rendelhetők:

O[o].CST a legkorábbi kezdési időpont (a komponensek rendelkezésre állása alapján).

O[o].NJ a rendeléshez tartozó munkák száma.

O[o].FJ a rendeléshez tartozó legkisebb sorszámú munka azonosítója.

Minden palettát önálló munkának tekintünk, ezek az *i* $(i = 1,\ldots, N_J)$ munkák kerülnek ütemezésre. A munkák legfontosabb attribútuma:

J[i].OID a megrendelés azonosítója.

A munkákhoz kapcsolódó egyéb adatok hivatkozással elérhetők.

A műhelyben lévő gépek (gépsorok) *m* $(m=1,\ldots, N_M)$ a következő tulajdonságokkal jellemezhetők:

M[m].PR[p] $(p= 1,\ldots, N_P)$ termékfüggő gyártási sebességek (időegység alatt megmunkálható *p* típusú termékek száma az *m* gépen).

M[m].ST[p1][p2] $(p1 = 1,\ldots, N_P;\ p2 = 1,\ldots, N_P)$ terméksorrendtől függő átállási idők (az *m* gép átállításának ideje *p1* terméktípus gyártásáról *p2* terméktípus gyártására).

M[m].NCAL az *m* géphez tartozó rendelkezésre állási időintervallumok aktuális száma.

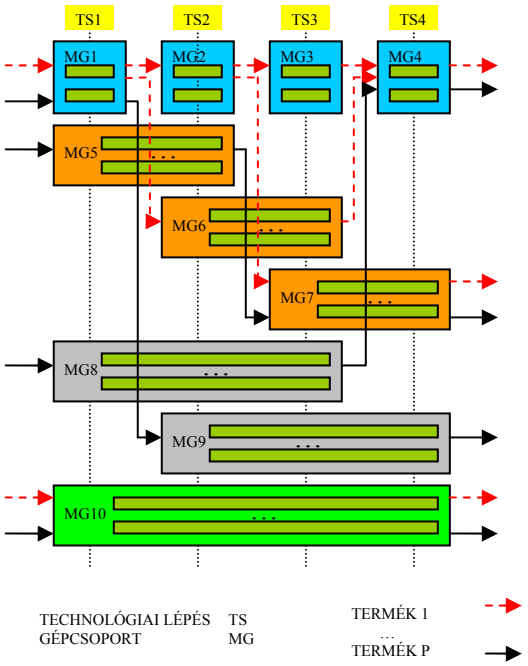
M[m].CAL[ci] $(ci = 1,\ldots,M[m].NCAL)$ az *m* géphez tartozó rendelkezésre állási időintervallumok sorozata, ahol:

M[m].CAL[ci].ST a *ci*-edik intervallum kezdete,

M[m].CAL[ci].ET a *ci*-edik intervallum vége.

A gépek csak a megadott időintervallumaikon belül dolgozhatnak.

M[m].ES az *m* gép műveletvégző képességét kifejező végrehajtási lépés típusa. A gépek között vannak olyanok, amelyek csak egy, és vannak olyanok is, amelyek több egymást követő technológiai lépést képesek elvégezni (3. ábra).



3. ábra: Kiterjesztett rugalmas Flow Shop modell

Az összes technológiai lépést tartalmazó sorozat olyan összefüggő részsorozatát, amely gépváltás nélkül végrehajtható – mint önálló egységet – végrehajtási lépésként *(es = 1,…, 10)* kezeljük. Négy technológiai lépés esetén tíz végrehajtási lépés különböztethető meg.

A végrehajtási lépések alapján gépcsoportokat *mg* *(mg = 1,…, 10)* definiálunk (3. ábra). A csoportokon belül különböző darabszámú és gyártási képességű gépek lehetnek. Ezeket az összerendeléseket egy kapcsolótömbben *MG_M* rögzítjük (1.ábra: *NÉV1 = MG, NÉV2 = M*).

Mivel a gépek nem minden terméktípust képesek kezelni (az *m* gép *p* terméktípusra vonatkozó gyártási sebessége nulla is lehet: *M[m].PR[p] = 0*), így a gépcsoportok terméktípusonként eltérő összetételűek lehetnek. Ezeket a csoportosításokat a *P_MG_M[p][mg][am]* kapcsolótömb írja le. Ahol:

p =1,…, N_P; a termékek,

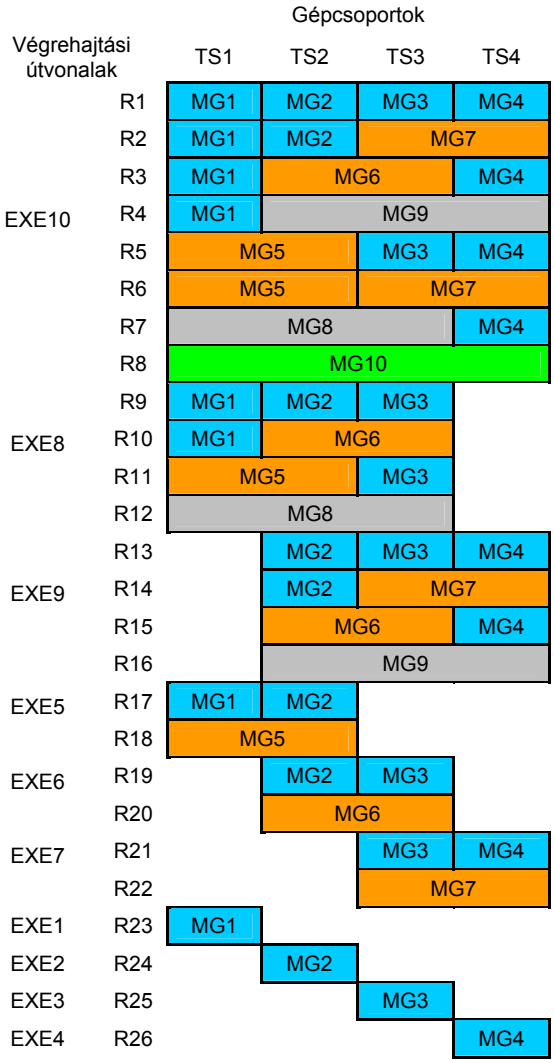
mg =1,…, 10; a gépcsoportok,

am = 1,…, P_MG_M[p][mg][0]; a *p* termék szempontjából az *mg* gépcsoporthoz tartozó gépek.

A munkákat végrehajtási útvonalakon mozgatjuk. Egy útvonal a végrehajtási lépéseket realizáló gépcsoportok olyan sorozatát fogja egységbe, amelyben az előforduló technológiai lépések csak egyszeresen vannak lefedve, azaz a végrehajtási lépések közös része üres halmaz (4. ábra).

Minden végrehajtási útvonalat *r (r = 1,…,26)* a hozzá tartozó gépcsoportokkal együtt az *R_MG* kapcsolótömbben definiálunk (3. ábra: *NÉV1 = R, NÉV2 = MG*).

A különböző *p* típusú termékek és azok gyártása során megengedett *r* végrehajtási útvonalak összerendelése egy *P_R* kapcsolótömbben leírható (3. ábra: *NÉV1 = P, NÉV2 = R*) figyelembe véve a gépek képességeit és a termékek gyártási folyamatának végrehajtás-típusait.



4. ábra: Végrehajtási útvonalak

Az elkészítendő új ütemterv szempontjából a gépeknek már meglévő terhelését (amelyeket az utolsó érvényben lévő ütemterv határoz meg) az *M_ENGAGED[m]* $(m = 1,\ldots, N_M)$ tömbben tároljuk. Ebből kiolvasható az egyes gépek felszabadulásának időpontja. Ez a megoldás az ütemtervek egymáshoz csatolására alkalmas (a beütemezett munkák zárolásának kérdését a cikkben nem részletezzük).

Az elkészített ütemtervet – annak érdekében, hogy minden részlete indexelhető legyen – három tömbben tároljuk. Ezek a következők:

1. A *J_A* kapcsolótömb mutatja a munkákhoz hozzárendelt útvonalakat az érintett konkrét gépekkel együtt a következő módon: *J_A[i][am]*.

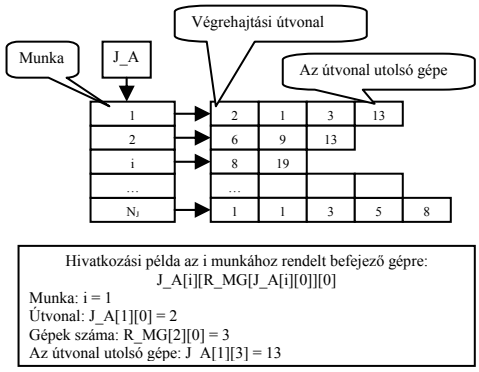
Ahol:

i jelenti a munkát $(i = 1,\ldots, N_J)$,

J_A[i][0] jelenti az *i* munkához hozzárendelt útvonalat,

R_MG[J_A[i][0]][0] jelenti az útvonal bejárása során érintett gépek számát,

J_A[i][am] $(am=1,\ldots, R_MG[J_A[i][0]][0])$ jelenti az *i* munka *am*-edik végrehajtási lépéséhez rendelt gépet.



5. ábra: A J_A kapcsolótömb felépítése

2. Az *M_WLOAD* kapcsolótömb mutatja a munkák sorrendjét az egyes gépeken a következő módon: *M_WLOAD[m][ai]*.

Ahol:

m $(m= 1,\ldots,N_M)$ jelenti a gépet,

M_WLOAD[m][0] jelenti a munkák számát az *m* gépen,

M_WLOAD[m][ai] $(ai = 1,\ldots, M_WLOAD[m][0])$ jelenti az *m* gép munkasorozatának *ai*-edik munkáját (3. ábra: *NÉV1 = M, NÉV2 = J*).

3. Az *M_STET* tömb tárolja az *M_WLOAD* tömb munkáinak időadatait a következő módon:

M_STET[m][ai].ST jelenti a kezdési időpontot,

M_STET[m][ai].SetT jelenti az átállás időtartamát,

M_STET[m][ai].ProcT jelenti a műveletvégzés időtartamát,

M_STET[m][ai].ET jelenti a befejezési időpontot.

A bemutatott adatmodell fontos tulajdonsága, hogy input adatai feltölthetők a termelésinformatikai rendszerekben jellemzően tárolt alapadatokból.

5.3 Szimulációs modell

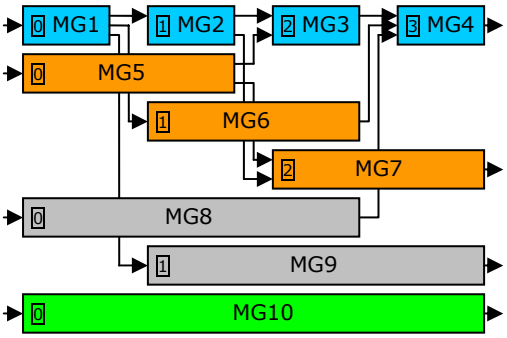
A gyártási folyamatot egy szabályalapú számítási eljárással szimuláljuk, amelynek során az összerendelt munka-gép párosok időadatait értékeljük ki (*M_STET*). A szimuláció során feltételezzük, hogy a gépek között korlátlan méretű átmeneti tárolók állnak rendelkezésre és a gépek közötti tranzakciós idők elhanyagolhatók. (A szimuláció továbbfejlesztésével könnyedén elérhető a különböző átmeneti tárolási stratégiák használata és tranzakciós idők definiálásával a gépek közötti anyagmozgatási idők figyelembe vétele is.)

A számítási folyamat a *J_A* kapcsolótömb és az *M_WLOAD* kapcsolótömb adataiból indul ki.

A számítási folyamat végrehajtásához definiáljuk a számítási lépések olyan sorrendjét, amely figyelembe veszi a következőket:

- Egy adott *i* munkának egy adott *m* közbenső géphez tartozó időadatainak számításához szükséges – többek között – az *i* munka előző géphez tartozó befejezési időpontjának ismerete és az *m* gép előző munkájának befejezési időpontja is.
- A gyártási környezet tartalmaz útvonal csatlakozásokat és szétválásokat (csomópontok), ezért egy adott *m* gépre több azonos típusú és különböző típusú gépről is érkezhetnek munkák.

Ezeknek megfelelően a gépcsoportok között sorrendi korlátozásokat fogalmazunk meg. A korlátozások leírhatók egy egyszerű (körútmentes) irányított gráffal, amelyben a csúcspontok jelentik a gépcsoportokat és az irányított élek mutatják a kötelező sorrendet.



6. ábra: Gépcsoportok kapcsolódásai

Minden egyes gépcsoporthoz hozzárendeljük a saját „befokát" (6. ábra). A befok az adott gépcsoportba érkező élek megengedett legnagyobb számát jelenti. A gépcsoportok keresett sorrendjét úgy kapjuk meg, hogy a befokok növekvő sorrendjének megfelelően rendezzük a gépcsoportokat. Az azonos befokú gépcsoportok között a további sorrendet a gépcsoportokra jellemző végrehajtási lépések határozzák meg úgy, hogy a több technológiai lépés megvalósítására képes gépcsoport lesz a magasabb prioritású.

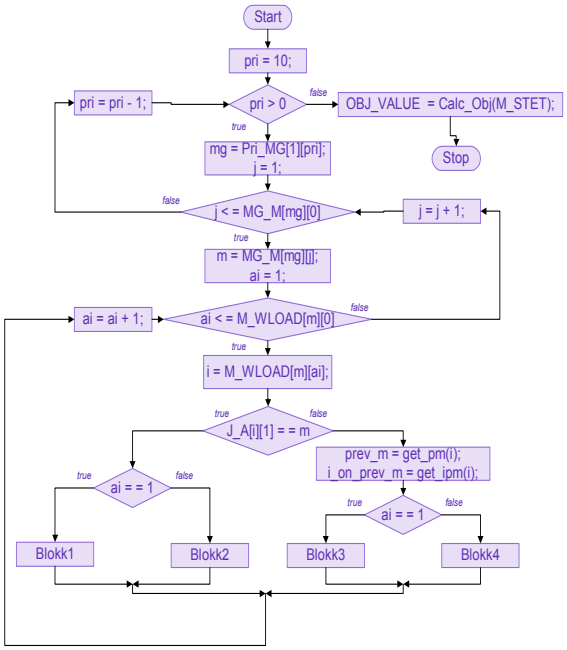
A gépcsoportokhoz rendelt prioritásokat a *PRI_MG* tömb tartalmazza a következő formában:

Prioritás: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
Gépcsoport: {4, 3, 7, 2, 6, 9, 1, 5, 8, 10}

A szimuláció fontosabb lépései a 7. ábrán láthatók. A számítási folyamatban a prioritások nem növekvő sorrendjében vesszük sorra a gépcsoportokat. Az adott gépcsoporthoz tartozó összes gépen (párhuzamos gépek) az *M_WLOAD* tömb által meghatározott sorrendben kiszámításra kerülnek a munkákhoz tartozó időadatok *(ST, SetT, ProcT, ET)*. Az eredmények bekerülnek az *M_STET* tömbbe.

A számításban meghatározó szerepet kapnak a műveletkezdési időpontok. Az *i* munka kezdési időpontját a hozzárendelt *m* gépen a következő értékek együttesen határozzák meg:

- a legkorábbi kezdési időpontja az *i* munkának *(O[J[i].OID].CST)*,
- a befejezési időpontja az *i* munkának az előző gépen *(M_STET[prev_m][i_on_prev_m].ET)*,
- a befejezési időpontja az előző munkának az *m* gépen *(M_STET[m][ai-1].ET)*,
- az *i* munkához tartozó átállási idő az *m* gépen *(M[m].ST[pprev][O[J[i].OID].P])*
- az *m* gép felszabadulásának időpontja az előző ütemterv terhelése alól *(M_ENGAGED[m])*, és
- az *m* gépnek a rendelkezésre állási időintervallumai *M[m].CAL[ci] (ci = 1,..., M[m].NCAL)*.



7. ábra: A szimuláció folyamatábrája

A szimuláció alapvetően kétféle működési módot támogat:

1. Független átállások megengedettek: Ez azt jelenti, hogy egy adott munka adott gépre érkezése előtt elkezdhető a gép átállítása az adott munka végrehajtásához (pl.: 8. ábra e2, e3).
2. Független átállások nem megengedettek: Ez azt jelenti, hogy a gépek átállítása csak akkor kezdhető el, amikor az adott munka már a gépre megérkezett.

Mindkét működési mód esetében az *m* gépen az *i* munka *ai* sorszámának és az *m* gépnek az *i* munka útvonalához tartozó sorszámának figyelembe vételével négy esetet különböztetünk meg (7. ábra: Blokk1, Blokk2, Blokk3, Blokk4).

A független átállások engedélyezése (1. működési mód) esetén ezek a következők:

Blokk1: az *i* munka útvonalának első gépe az *m*, és az *m* gép első munkája az *i*.

```
o = J[i].OID;
pact = O[o].P;
M_STET[m][ai].SetT  = max_all_p( M[m].ST[p][pact] );
M_STET[m][ai].ST    = max( O[o].CST, M_ENGAGED[m] );
M_STET[m][ai].ProcT= P[pact].Q / M[m].PR[pact];
M_STET[m][ai].ET    = M_STET[m][ai].ST +
                      M_STET[m][ai].SetT +
                      M_STET[m][ai].ProcT;
Load_STET_to_CAL( M_STET[m][ai].ST,
                  M_STET[m][ai].ET, 0, m);
```

Blokk2: az i munka útvonalának első gépe m, az m gépnek egy közbenső munkája az i.

```
o = J[i].OID;
pact = O[o].P;
pprev = O[J[M_WLOAD[m][ai-1]].OID].P
M_STET[m][ai].SetT   = M[m].ST[pprev] [pact];
M_STET[m][ai].ST     = max( M_STET[m][ai-1].ET,
                            O[o].CST );
M_STET[m][ai].ProcT = P[pact].Q / M[m].PR[pact]);
M_STET[m][ai].ET     = M_STET[m][ai].ST +
                       M_STET[m][ai].SetT +
                       M_STET[m][ai].ProcT;
Load_STET_to_CAL( M_STET[m][ai].ST,
                  M_STET[m][ai].ET,
                  M_ STET [m][ai-1].ET, m);
```

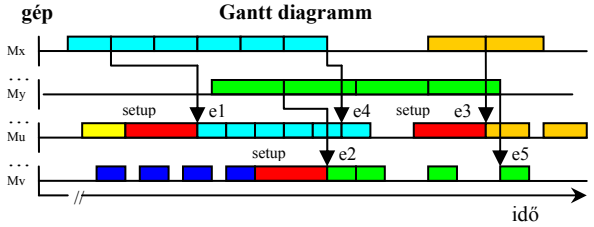
Blokk3: az i munka útvonalának egy közbenső gépe az m, az m gépen az első munka az i.

```
o = J[i].OID;
pact = O[o].P;
M_STET[m][ai].SetT   = max_all_p( M[m].ST[p][pact] );
M_STET[m][ai].ST     = max(
                 M_STET[prev_m][i_on_prev_m].ET –
                 M_STET[m][ai].SetT, M_ENGAGED[m] );
M_STET[m][ai].ProcT = P[pact].Q / M[m].PR[pact];
M_STET[m][ai].ET     = M_STET[m][ai].ST +
                       M_STET[m][ai].SetT +
                       M_STET[m][ai].ProcT;
Load_STET_to_CAL( M_STET[m][ai].ST, M_STET[m][ai].ET,
0, m);
```

Blokk4: az i munka útvonalának egy közbenső gépe az m, az m gépnek egy közbenső munkája az i.

```
o = J[i].OID;
pact = O[o].P;
pprev = O[J[M_WLOAD[m][ai-1]].OID].P
M_STET[m][ai].SetT = M[m].ST[pprev] [pact];
if ( M_STET[prev_m][i_on_prev_m].ET <=
                                M_STET[m][ai -1].ET)
   M_STET[m][ai].ST = M_STET[m][ai -1].ET;
else
   M_STET[m][ai].ST = max(
                 M_STET[prev_m][i_on_prev_m].ET –
                 M_STET[m][ai].SetT,
                 M_STET[m][ai -1].ET);
M_STET[m][ai].ProcT    = P[pact].Q / M[m].PR[pact]);
M_STET[m][ai].ET       = M_STET[m][ai].ST +
                         M_STET[m][ai].SetT +
                         M_STET[m][ai].ProcT;
Load_STET_to_CAL( M_STET[m][ai].ST,
                  M_STET[m][ai].ET,
                  M_ STET [m][ai-1].ET, m);
```

A Blokk4 működését bemutató példák a 8. ábrán láthatók. A vázolt Gantt diagramon az *e1, e2, e3, e4, e5* szimbólumokkal jelölt események a munkák átvitelét jelentik a hozzárendelt előző gépről az aktuális gépre.



8. ábra: A Blokk4 működése

A kiemelt példákon jól megfigyelhető, hogy az

- *e1* esetben az adott gép előző munkájának befejezési ideje a meghatározó, ezután következhet a gép átállítása és a munka elindítása.
- *e2* esetben az előző munka befejezési időpontja kisebb, mint az aktuális munka befejezési időpontja az előző gépen, így ennek a különbségnek az értékével korábban kezdődhet a gép átállítása a munka tényleges megérkezése előtt.
- *e3* esetben az előző munka befejezési időpontja jóval kisebb, mint az aktuális munka befejezési időpontja az előző gépen, ezért a gép átállítása a teljes átállítási idővel korábbra tehető, így az befejeződik a munka megérkezésének időpontjáig.
- *e4* esetben az előző munka befejezési időpontja a meghatározó, továbbá azonos terméktípusú munkák követik egymást ezért nincs szükség átállításra,
- *e5* esetben sem szükséges átállítás és a munka előző gépről érkezése dominál.

A *Load_STET_to_CAL* eljárás végzi az *i* munka kezdési és befejezési időpontjainak az *m* gép rendelkezésre állási időintervallumaihoz való illesztését, figyelembe véve a gép utolsó terhelésének befejezési időpontját. Ennek során, az *i* munka által képviselt fix méretű terhelést (időablakot) az *m* gép első (vagyis legkorábban kezdődő) megfelelő méretű szabad időintervallumára balra illesztve helyezi el.

A szimuláció lefutásának eredményei az egyes munkák időadatai, amelyek az *M_STET* tömbben találhatók. Ezek felhasználásával az ütemtervet jellemző célfüggvények (4. fejezet) értékei számíthatók, amelyek az *OBJ_VALUE* tömbben érhetők el.

5.4 Heurisztikus ütemező algoritmus

Többféle heurisztikus megoldási módszert fejlesztettünk ki. A végrehajtási útvonal és gépek munkákhoz való rendelésének, valamint a munkák gépenkénti sorrendjének meghatározása alapján két csoportba sorolhatók az algoritmusok.

Az első csoportba tartoznak a felépítő jellegű heurisztikák. Ezek – a korábban ütemezett és nem módosítható hozzárendelésektől eltekintve – üres ütemtervből kiindulva, a kiválasztott soron következő ütemezendő munkához tartozó útvonalra, gépre és sorrendre vonatkozó döntéseket heurisztikus szabályok és szimulációs kiértékelés alapján hozzák meg.

A második csoportba tartoznak az iteratív javításon alapuló heurisztikák. Ezeknél a módszereknél egy kezdeti megvalósítható ütemtervből kiindulva megengedett módosítások és szimulációs kiértékelések iterálásával alakul ki a választott célfüggvénynek jobban megfelelő ütemterv.

A továbbiakban egy lokális keresési technikán alapuló algoritmuson keresztül mutatjuk be az

ütemezési feladat megoldásának menetét, felhasználva az 5.2 és az 5.3 fejezetben leírtakat.

Heurisztikus kereső eljárás tabulistával:

```
Sch_alg
{
 So = inicializal( );
 Sm = So;
 TS = 0;
for( l = 1; l <= L ; l++ )
{ for ( k = 1; k < K; k++ )
  { S = Sm;
    i = 1 + random( NJ );
    S = kivesz( i );
    S = random_beszur( i );
    v = random( V );
    for ( p = 1; p < v; p++ )
     {
      m = 1 + random( NM );
      h = random( H );
      S = random_permutal( h, m );
     }
   if ( Nem_eleme_a_tabulistanak( S ) )
     { TL_elejere_beszur( S );
       if ( TL_eleminek_szama > E )
        {  TL_utolso_elemet_torli( );
        }
       f = szimulacio( S );
       if ( k == 1 )
        {   fk = f; Sk = S;
        }
      else
        {   if ( fk > f )
                { fk = f; Sk = S;
                }
        }
     }
 }//k
 Sm = Sk;
 if ( l == 1 )
   { fl = fk; Sl = Sk;
   }
 else
   { if ( fk < fl )
       {  fl = fk; Sl = Sk;
       }
   }
}//l
return Sl;
}
```

A modellben szereplő korlátozások közül a munkákhoz rendelt befejezési határidőket puha (megsérthető) korlátozásnak tekintjük, a célfüggvényekben vesszük figyelembe azokat. Minden más korlátozást kemény (nem megsérthető korlátozásnak) tekintünk.

A keresési folyamat során egy kezdeti ütemtervből kiindulva megengedett módosítások ismételt végrehajtásával alakul ki a végső ütemterv. A kiindulási ütemterv elemei a megfelelő tömbökből egyenletes valószínűséggel kiválasztott értékekkel vannak inicializálva.

A módosító operátor alapvetően kétféle változtatást idézhet elő az ütemtervben:

1. Egy kiválasztott munkát kiemel az ütemtervből, majd a hozzárendelt útvonalat és gépeket megváltoztatva újra beszúrja azt az ütemtervbe.
2. Egy kiválasztott gépen a munkák sorrendjét megváltoztatja.

Egy közbenső lépés során az algoritmus bizonyos számú (K) kiterjesztett ütemtervet készít a módosító operátor alkalmazásával. Ha egy kiterjesztett ütemterv szerepel a tabulistán, akkor azt nem értékeli ki, figyelmen kívül hagyja. Ha egy kiterjesztett ütemterv nem szerepel a tabulistán, akkor felkerül a listára és a legkorábban felvett listaelem törlődik. A szimulációs kiértékelést követően, ha a célfüggvénynek a kiterjesztett ütemtervre vonatkozó értéke jobb, mint az adott kiterjesztés legjobb értéke, akkor megjegyzésre kerül.

A kiterjesztés legjobb ütemterve lesz a következő lépés kiterjesztésének a kiindulási bázisa. Továbbá, ha ez az érték jobb, mint a keresés során megtalált legjobb érték akkor ez kerül megjegyzésre.

Az algoritmus leírása során használt azonosítók jelentése a következő:

Paraméterek:

NJ: A munkák száma.

NM: A gépek száma.

E: A Tabulista maximális mérete.

L: A lépések száma.

K: Egy lépéssel elérhető szomszédos ütemtervek száma.

V: Egy adott módosítás során a munkasorrendek változtatásának maximális száma.

H: A permutáció ciklusának maximális hossza.

Változók:

S: egy megvalósítható ütemterv (megoldás). Az *M_WLOAD* és a *J_A* kapcsolótömbök együttesét jelenti.

So: Egy kezdeti megoldás, amelynek elemei a megfelelő tömbökből egyenletes valószínűséggel kiválasztott értékekkel vannak inicializálva.

Sm: Az aktuális lépés kiindulási megoldása, (elmentett megoldás).

Sk: A legjobb megoldása az aktuális kiterjesztésnek.

Sl: A lépések során megtalált legjobb megoldás.

TL: Tabulista, amely tiltott, korábban már kiértékelt megoldásokat tartalmaz időrendi sorrendben.

f: A célfüggvény *S* megoldásra vonatkozó értéke.

fk: A célfüggvény *Sk* megoldásra vonatkozó értéke.

fl: A célfüggvény *Sl* megoldásra vonatkozó értéke.

i: Egy kiválasztott munka.

m: Egy kiválasztott gép.

l: A lépésekhez tartozó ciklusváltozó.

k: A kiterjesztésekhez tartozó ciklusváltozó.

v: Az aktuális permutációk száma.

p: A permutációkhoz tartozó ciklusváltozó.

h: A permutáció ciklusának aktuális hossza.

Az ismertetett alapalgoritmus további – jelen cikkben nem részletezett – bővítő elemeket is tartalmaz. Ezek lehetővé teszik az algoritmus leállási feltételének összetettebb megfogalmazását (pl.: a legjobb érték adott számú lépésben nem javul, a futási idő elér egy megadott korlátot). Továbbá, lehetőséget biztosítanak a keresési paraméterek futás közben történő változtatására (*E*, *K*, *V*, *H*, *L*), ezáltal különböző vezérlési stratégiák alkalmazását teszik lehetővé.

6. FUTÁSI EREDMÉNYEK

A bemutatott adatmodellt és algoritmusokat Borland C++ Builder 5 fejlesztőrendszerben implementáltuk.

Az elvégzett tesztelések közben azt tapasztaltuk, hogy a futási idő alakulását alapvetően befolyásolja a feladat mérete. Minél több munka és gép szerepel a feladatban annál több időbe telik egyetlen ütemterv kiértékelése. Ennek az időtartamnak és a feladat megoldására rendelkezésre álló időtartamnak az ismeretében könnyen megbecsülhető a keresés során kiértékelhető ütemtervek maximális száma.

A futási idő szempontjából további fontos befolyásoló tényező a feladat nehézségi foka. Egy feladat nehézségi foka annál magasabb, minél nehezebb az összes kemény korlátozást kielégítve javítani a célfüggvény értékét. Erre próbafuttatással kaphatunk becslést.

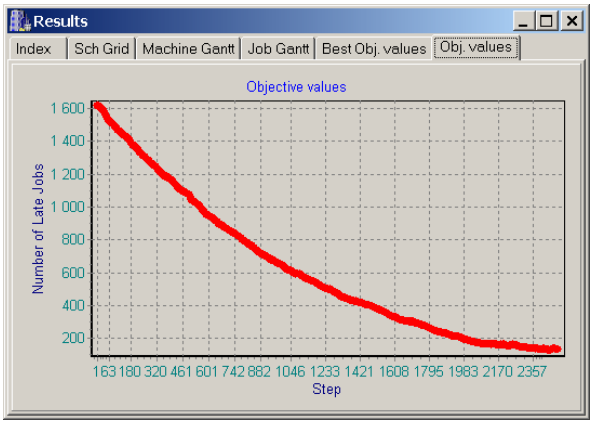
Ezeket a becsléseket kiindulásként felhasználva, valamint a keresési paraméterek futás közben történő megfelelő szabályozásával alakítható ki a hatékony keresési stratégia.

Az ütemezési feladat jellemző méreteinek és az alkalmazott keresési paramétereknek a futási időre gyakorolt együttes hatása – magas nehézségi fokú feladatok esetén – az 1. táblázatban látható. (Tesztkörnyezet: Intel Pentium 4 2,8 GHz CPU, 512MB DDR333 SDRAM, Microsoft Windows XP OS.)

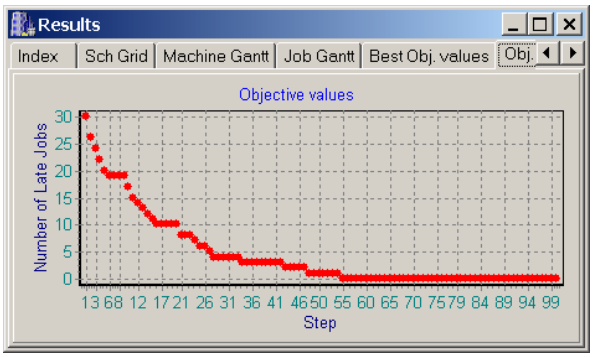
1. táblázat. **Futási idők**

	NO	*NJ*	*NM*	*L*	*K*	**Futási idő**
1	22	78	11	100	100	**2 s**
2	300	1052	51	2000	25	**1 min 9 s**
3	700	2455	151	2500	10	**2 min 35 s**

Az 1. táblázatban az egyes szimbólumok jelentése a következő: *NO* a megrendelések száma, *NJ* a munkák száma, *NM* a gépek száma, *L* a lépések száma, *K* a vizsgált szomszédos ütemtervek száma.



9. ábra: A késő munkák számának változása az 1. feladat megoldása közben



10. ábra: A késő munkák számának változása a 3. feladat megoldása közben

Ezekben a leegyszerűsített keresési feladatokban az *L* és *K* konstansok, a kiválasztott célfüggvény pedig a késő munkák száma (8).

A késő munkák számának változása a megtett lépések számának függvényében a 9. és 10. ábrákon látható.

Az eddig elvégzett vizsgálatok igazolták, hogy a javasolt megoldási módszer nagyméretű feladatok esetében is hatékonyan alkalmazható. Rugalmasan alkalmazkodik a kiválasztott célfüggvényhez és kivárható időn belül szolgáltatja az eredményt. További hatékonysági vizsgálatokat jelenleg is végzünk a kidolgozott algoritmusok értékelése és továbbfejlesztése érdekében.

7. ÖSSZEFOGLALÁS

A cikkben az igény szerinti tömeggyártás műhelyszintű ütemezési feladataival foglalkoztunk. Az erőforrások allokálásának és a munkák ütemezésének modellezésére egy új kiterjesztett rugalmas Flow Shop ütemezési modellt vezettünk be. Az alternatív útvonalak, korlátozottan rendelkezésre álló párhuzamos gépek, sorrendfüggő átállási idők, terméktől függő termelési sebességek valamint határidős munkák együttes figyelembevételével valós gyártási környezethez alkalmazkodó ütemezési koncepció kidolgozására helyeztük a hangsúlyt. Ismertettünk egy heurisztikus megoldási módszert, amely lokális keresési technikát valamint indexelt adatmodellen alapuló gyors kiértékelést kombinál.

KÖSZÖNETNYILVÁNÍTÁS

A szerző ezúton mond köszönetet a kutatási és fejlesztési munka támogatásáért a Magyar Tudományos Akadémia Termelésinformatikai Kutatóhelyének (alapítva a Miskolci Egyetem Alkalmazott Informatikai Tanszékén, Grant No. MTA – TKI 06108). Az ismertetett eredmények a „VITAL” nevű projekthez (Nemzeti Kutatási és Technológiai Hivatal, Grant No.: 2/010/2004) kapcsolódó kutatási munkák során születtek.

IRODALOM

[1] ACERO-DOMINGUEZ M. J., PATERNINA-ARBOLEDA C. D., 2004, *Scheduling Jobs on a K-Stage Flexible Flow Shop Using a TOC- (Bottleneck) Procedure*, Proceedings of the 2004 Systems and Information Engineering Design Symposium.

[2] BARKMEYER E., DENNO P., FENG S., JONES A., WALLACE E., 1999, *"NIST Response to MES Request for Information"*, NISTIR 6397, National Institute of Standards and Technology: pp. 1-124.

[3] BRUCKER P., 1998, *Scheduling Algorithms*, Springer-Verlag, Berlin.

[4] BAUER A., BOWDEN R., BROWNE J., DUGGAN J., LYONS G., 1991, *Shop Floor Control Systems – From design to implementation*, Chapman &.Hall, UK.

[5] YAMADA T., 2003, *Studies on Metaheuristics for Jobshop and Flowshop Scheduling Problems*, Ph.D Thesis, Kyoto University, Japan.

[6] KULCSÁR GY., ERDÉLYI F., 2005, *Production Goals and Heuristic Methods in Extended Flexible Flow Shop Scheduling Problems*, Forum of PhD Students, Miskolc, Hungary. pp. 104-109.

[7] KULCSÁR GY., HORNYÁK O., ERDÉLYI F., 2005, *Shop Floor Decision Supporting and MES Functions in Customized Mass Production*, Conference on Manufacturing Systems Development - Industry Expectations, Wroclaw, Poland, pp. 138 – 152.

[8] KURNAZ A., COHN Y., KOREN Y., 2005, *A Framework for Evaluating Production Policies to Improve Customer Responsiveness*, CIRP Annals, Volume 54/1, pp. 401-406.

[9] LINN R., ZHANG W., 1999, *Hybrid Flow Shop Scheduling: A Survey*, Computers and Industrial Engineering, 37, pp. 57-51.

[10] MCKAY K. N., WIERS V. C. 1999, *Unifying the Theory and Practice of Production Scheduling*, Journal of Manufacturing Systems, Vol. 18, No. 4, pp. 241-248.

[11] MCCLELLAN M., 1997, *Applying Manufacturing Systems*, St. Lucie Press, Florida.

[12] TÓTH T., ERDÉLYI F., 2004, *Research and Development (R&D) Requirements for Up-to-date Production Planning & Scheduling (PPS) Systems*, The Eleventh International Conference on Machine Design and Production, pp. 13 – 15, Antalya, Turkey.

[13] TÓTH T., ERDÉLYI F., 1997, *The Role of Optimization and Robustness in Planning and Control of Discrete Manufacturing Processes*, Proceedings of the 2nd World Congress on Intelligent Manufacturing Processes & Systems, Budapest, Hungary, pp. 205-210.

[14] TÓTH T., 2004, *Termelési rendszerek és folyamatok*, Egyetemi tankönyv, Miskolci Egyetemi Kiadó, Miskolc.

[15] TÓTH T., 1998, *Tervezési elvek, modellek és módszerek a számítógéppel integrált gyártásban*, Egyetemi tankönyv Miskolci Egyetemi Kiadó, Miskolc.

[16] TÓTH T., 1994, *Heurisztikus módszerek termelés-programozási feladatok megoldására*, Oktatási segédlet, Miskolci Egyetem, Miskolc.

[17] TAILLARD E., 1990, *Some Efficient Heuristic Methods for the Flow Shop Sequencing Problem*, European Journal of Operational Research, 47, (1) pp. 65-74.

[18] ZWEBEN M., FOX M. S., 1994, *Intelligent Scheduling*, Morgan Kaufmann Publishing, San Francisco.

[19] SAUER J., 2006, *Modeling and solving multi-site scheduling problems*, Book section in: van WEZEL W., JORNA R., MEYSTEL A., *Planning in Intelligent Systems: Aspects, Motivations and Methods*, pp. 281 - 299, Wiley.

[20] VÍZVÁRI B., 2004, *Ütemezés elmélet*, 9. fejezet az IVÁNYI A. (szerk.), *Informatikai algoritmusok*, 364-415, ELTE Eötvös Kiadó.