

GYÁRTÁSI SOROZATNAGYSÁGOK OPTIMALIZÁLÁSA TÖBBLÉPCSŐS, TÖBBTERMÉKES, TÖBBGÉPES TERMELÉSI RENDSZEREKBE

KATONA RAJMUND¹

Összefoglalás:

Korunkban, mikor a fogyasztási cikkek kereslete stagnálni látszik egyre nehezebb utat találni a növekedés általi profitnövekedésnek. Már sokan belátták, hogy az operatív folyamatok fejlesztéséből (mint a termelés) adódó költségcsökkentés és minőségi növekedés által óriási versenyelőnyre tehetnek szert a gyártó cégek és a szolgáltatók, kereskedők egyaránt. Dolgozatomban a termelési tétel nagyságok optimalizálásának témáját job shop rendszerekben járom körül (mint a termelési modellek, a szimuláció és a metaheurisztikus optimalizációs eszközök) valamint egy szimulációs modellt alkotok ennek megoldására, a szoftvercsomagban található optimalizáló modul alkalmazásával.

Kulcsszavak:

termelésütemezés, szimuláció, optimalizálás, kötegelt gyártás, job shop termelési modell

Summary:

In our age, when consumer demand seems to be stagnating it is more difficult to find a path of growth. Many firms have realized that the development of operational processes (such as production) to reduce the cost and increase quality is results huge competitive advantage for manufacturers and service providers, dealers. In this paper I am going to exhibit the topic of the optimization of the lot sizes in job shops (such as manufacturing models, simulation and metaheuristic optimization) and I am going to create a simulation model to solve the optimization by using the optimization module provided by the software package.

Keywords:

manufacturing scheduling, simulation, optimisation, batch production, job shop manufacturing model

¹ logisztikai menedzsment MA szakos hallgató,
Széchenyi István Egyetem,
katona.rajmund@gmail.com

Bevezetés

Korunk mérnöki tervezési gyakorlatában egyre fontosabbá válik, hogy minél rövidebb idő alatt a lehető legjobb eredményt lehessen elérni. A termelésitervezési rendszereket számos tényező befolyásolja, melyek kombinációja számos lehetőséget és problémát okoz. Ezzel párhuzamosan nő a termékek és azok változatainak száma és nőnek a költségcsökkentés iránti igények is, ami tovább nehezíti a termelési folyamatok tervezését.

Dolgozatomban rá fogok világítani a kistételes termelés előnyeire a nagytételes-készletező munkamódszerrel szemben és létrehozok egy szimulációs modellt, ami ennek megvalósítását nem csak, hogy támogatja, de még az igényekhez és lehetőségekhez mért optimális ütemezési paraméterek kiszámítására is képes. A téma jelentősége abban rejlik, hogy a kifejlesztendő modell számos termelésorientált vállalatnál alkalmazható lehet, ahol képes a tervezési idők redukálására és segíthet hatékonyabbá tenni a termelési folyamatokat, valamint különböző ütemezési stratégiák vizsgálatára is alkalmazható.

A termelésben használatos modellek

A termelési rendszerek számos tényező alapján jellemezhetőek, mint például a gépek, vagy erőforrások száma (valamint azok jellemzői és konfigurációja), az automatizáltság szintje, az anyagmozgató rendszer típusa, és még sorolhatnám. Ezen jellemzők különbsége nagyszámú különböző tervezési és ütemezési modell létezésére ad okot. Egy termelési modellben a termékek feldolgozási, gyártási folyamatára *feladatként* (job), és az ezt megvalósító elemekre általában *gépként* (machine) hivatkoznak. Egy gyártási folyamatban egy feladat lehet egy egyszerű művelet, vagy műveletek összessége, amiket különféle gépeken kell végrehajtani. (Pinedo, 1995)

A termelési modelleket öt különböző osztályba sorolhatjuk (Pinedo, 1995) szerint:

- Projekttervező és -ütemező modellek (the project planning and scheduling models)
- Egygépes, párhuzamos gépes és job shop modellek (single machine, parallel machine and job shop models)
- Automatizált anyagmozgatású termelési rendszerekre koncentrááló modellek (models focuses on production systems with automated material handling)
- Gyártási tétel nagyság ütemező modellek (*lot scheduling models*)
- Ellátási lánc tervező és ütemező modellek (supply chain planning and scheduling models)

Dolgozatom témájához a job shop; valamint a gyártási tétel nagyság ütemező modellek kapcsolódnak.

A job shop modell

Többműveletes műhelyekben a feladatoknak gyakran különböző útvonalakon kell végighaladniuk. Az olyan környezetre, mely a flow shop általánosítása job shop-ként hivatkozik az angol szakirodalom. A magyar nyelvű szakirodalom az ilyen rendszerekre műhelyrendszerű termeléseként hivatkozik. A flow shop tulajdonképpen egy olyan job shop, ahol minden egyes feladat ugyanabban a sorrendben érinti a gépeket.

A legegyszerűbb job shop modellekben a feladatok legfeljebb egyszer érintenek meghatározott gépeket a feldolgozási folyamat során, de elképzelhető az is, hogy egy feladat ugyanazt a gépet többször is érinti. A job shop általánosítása a rugalmas job shop, ahol ez egyes munkaállomásokon belül több ugyanolyan funkciójú gép található párhuzamosan. Kombinatorikai szempontból a recirkulációt (megengedünk hurkokat) megengedő rugalmas job shop a legkomplexebb gyártókörnyezet. Ez leginkább a félvezetők gyártására jellemző. (Pinedo, 1995)

A gyártási tétel nagyság ütemező modellek

A gyártási tétel nagyság ütemező modelleket közép és hosszú távú termelési tervezéshez használják. A többi tárgyalt modellhez képest itt a termelés és az igények folyamatosak. Mikor a gép egyik termék gyártásáról egy másikéra áll át, akkor átállási költség (Changeover

Cost) lép fel. Az általános cél ilyen modellek esetén, hogy minimalizáljuk a teljes készletezési és átállási költségeket. Az ilyen modellek a feldolgozóiparra jellemzőek, mint például a vegyipar. 0

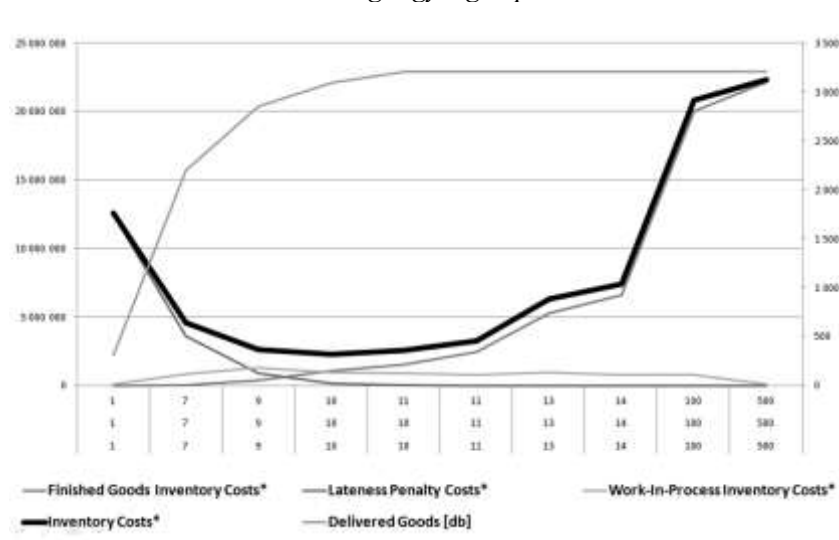
Problémaleírás

A nagytételes-készletező (batch-and-que) munkamódszer a tapasztalatok szerint mindig hosszú várakozással jár, mivel ez azt vonja maga után, hogy a termék csak vesztegel, amíg a részleg/gép át nem vált arra a tevékenységre, amire a terméknek szüksége van.

A legtöbben úgy gondolják, hogy a termelésnek ez a leghatékonyabb módszere, mivel így minden gép folyamatosan dolgozik, és a kihasználtságukat csak kis mértékben rontják az átállási idők. Valójában ez a gyakorlat egyáltalán nem hatékony, de mivel ennek negatív hatásai a vállalat más területein jelentkeznek sokan képtelenek belátni. Ezen problémák kiküszöbölésére már 50 évvel ezelőtt kifejlesztettek egy sokkal hatékonyabb módszert. Ez a kistételes (small-lot) termelés, ami számos előnnyel jár, mint például a késztermék készletezési költségek csökkenése, a keresletingadozásra való reagáló képesség növekedése mellett. (Womack, Jones, 2009)

1. ábra.

A költségek és a darabszám alakulása a kötegnagyságok párhuzamos növekedésének függvényében



Forrás: Saját szerkesztés (az adatok az elvégzett szimulációk eredményeiből származnak)

Az 1. ábráról leolvasható, hogy a költségek viselkedésében azt figyelhetjük meg, hogy a tétel nagyságok növekedésével a megrendelés késve szállításának költsége csökken, viszont a késztermékek raktározási költsége növekszik. A legyártott darabszám is növekszik a tétel nagysággal, mivel nagyobb tétel nagyság esetén az átállási idők összege egyre csökken, viszont a növekedés megáll, ha az összes megrendelt terméket szállították. Az ez után legyártott termékek már csak a késztermékek raktározási költségét növelik.

A célunk hogy meghatározzuk a termelési tétel nagyságok azon kombinációját, amely minimális összköltséget eredményez, ehhez egy modellt hozok létre melyben egy job shop modell működése folyamatos termékáram esetén vizsgálható és segítségével megállapítható az egyes feladatok optimális kötegnagysága (batch/lot size). E modell összevonja a job shop és a gyártási tétel nagyság ütemező modellek tulajdonságait. „Ilyen komplex rendszerek tervezési feladatainak elvégzésére létezik egy lehetséges eljárás. A modellezés és dinamikus szimuláció alkalmas a legtöbb kérdés megválaszolására, és képes leképezni az adott termelési rendszer időbeli viselkedését. Ez a technika a diszkrét, eseményvezérelt szimuláció.” (Jósvai, Perger, 2010 p. 3.)

A szimulációs modell

„A szimulációs modell alkalmazásával képesek vagyunk mesterségesen létrehozni azoknak a – döntéseknek, beavatkozásoknak, eseményeknek a hatására bekövetkező – állapotoknak a sorozatát, amelyek leírják az illető rendszer vagy a rendszer néhány komponensének viselkedését egy bizonyos időintervallumban., (Szily, 1994 p. 148.)

A modell tervezési eszköz, amely számos termelés tervezési kérdés megválaszolására alkalmas. Létrehozása és felépítése a fiktív és generált paraméterek alapján történt, fő célként szem előtt tartva a rugalmasságot, annak érdekében, hogy valós rendszerekre könnyedén adaptálható legyen. A létrehozott modell elsősorban a termelési tétel nagyságok optimalizálására készült, de számos egyéb termelésütemezési problémára szolgálhat megoldással.

A szimulációs környezet

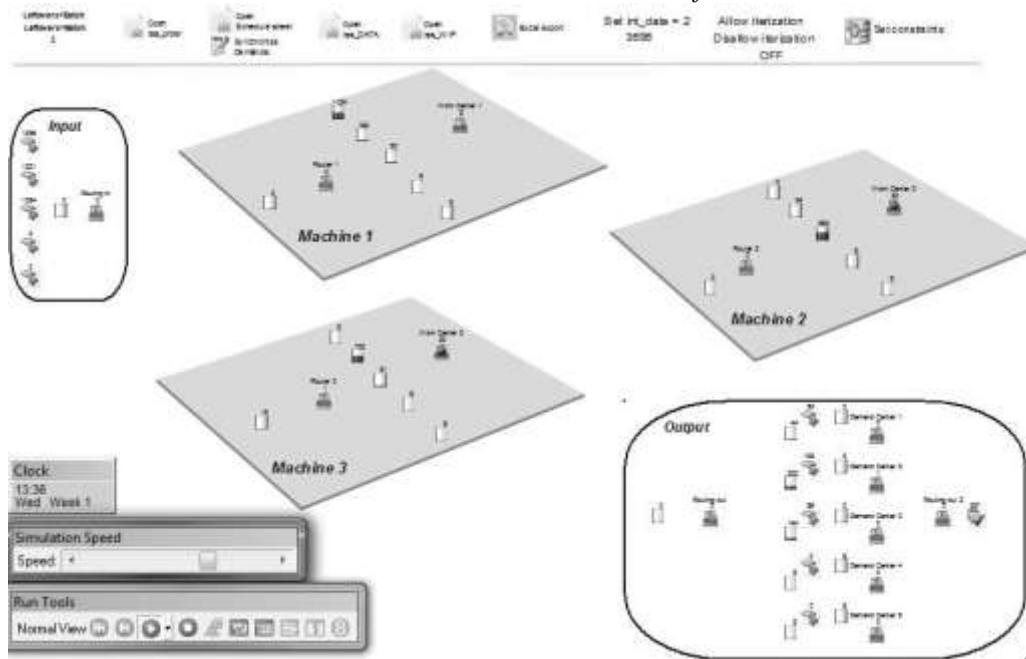
A szoftver piacon legelterjedtebbnek mondható szimulációs programok a Witness, az Enterprise Dynamics Developer, a Technomatix Plant Simulation és a Simul8. Ezen szoftverek számos alkalmazási terület folyamatainak tervezését, modellezését, optimalizálását és elemzését lehetővé tevő vizuális interaktív szimulációs programok, melyek különféle (logisztikai, termelési, közlekedési stb.) rendszerek számítógépes vizsgálatára alkalmas döntéstámogató eszközök. Segítségükkel logisztikai tervezési folyamatok széles skálája oldható meg. A négy rendszer működési filozófiája megegyezik, azok hasonló számítástechnikai és matematikai alapokon nyugszanak. (Prezenszki, 1999) A szimulációs modellt, Simul8 szimulációs szoftver 2009-es verziójával alkottam meg.

Modellalkotás

A modellalkotás során, mint fentebb már leírtam, a rugalmasság volt a fő szempont. Sikerült egy olyan modellt alkotnom, mely diszkrét rendszerek garmadájának szimulációjára képes és mindehhez az esetek nagy részében csak minimális átalakításra lehet szükség. Meg kell, hogy jegyezzem, hogy a SIMUL8 Process plug-in építőelemeit alkalmazva a modell alkalmassá válhat folytonos rendszerek működésének szimulálására is.

Felhasználói kezelő felület készült a modellhez, hogy a szimuláció kezelését biztosítsa, és a kialakított számos funkció átláthatóvá és kezelhetővé váljék.

2. ábra.
A létrehozott modell vizuális interfésze



Forrás: Saját szerkesztés

A modell adatkommunikációja a Simul8 *spreadsheet*-jein keresztül történik. A szoftver képes folyamatos adatkommunikációt folytatni az Excellel is, viszont ez jelentősen ronthatja a modell futási idejét, ezért a modell csak az egyes futások eredményeinek exportálása során teremt azzal kapcsolatot. Bizonyos paraméter-beállítások megváltoztatását a felhasználó speciális szimulációs ismeretek nélkül is végrehajthatja, és a szimulációs eredményeket is értelmezheti. A Simul8 tartalmaz egy eszközt, mely a Visual Logic-ot kihagyva teszi lehetővé az egyes feladatok definiálását. Ez a Jobs Matrix. Segítségével megadhatjuk, hogy az egyes műveletek milyen sorrendben és melyik gépeket érintsék, valamint a műveleti és átszerelési idők értékeit vihetjük be, akár mint előre definiált valószínűségi változókat.

A Simul8 programozási nyelve a Visual Logic; speciálisan erre a programkörnyezetre fejlesztett programozási nyelv. A programnyelv a Microsoft Visual Basic-ből fejlődött ki, ezért sokban hasonlítanak. A programkód elemek a modellek dinamikus irányítására és szabályzására szolgálnak, a készített programkódok (Logic-ok) a szimuláció során kerülnek futtatásra, valahányszor meghívásra kerülnek. Fontos, hogy a programkódok a lehető legegyszerűbb módon kerüljenek megalkotására, mivel a bonyolult, nagy számításigényű programkódok komolyan megnövelhetik a szimuláció lefutási idejét.

3. ábra.

A létrehozott modell Visual Logic menüje és az egyik programkód a szerkesztőablakban



Forrás: Saját szerkesztés

A létrehozott modell tesztelése

A modell tesztelését egy 3 termékes 3 lépcsős 3 gépes rendszerben, az egyes termékek gazdaságos gyártási tétel nagyságainak meghatározásával végeztem. A modell méretének leginkább a későbbi optimalizációs eljárásra való előrelátás szabott gátat, mivel nagyobb modell optimalizálásához nem állt rendelkezésemre megfelelő számítástechnikai háttér. A modell determinisztikus a lefutási idő minimalizálása érdekében, de bármelyik paraméter azonnal valószínűségi változóra cserélhető.

Asszumpciók

A modellben az alábbi megállapítások ismerhetők el érvényesnek:

- A rendszerre a már tárgyalt Job Shop modell szabályai vonatkoznak.
- A rendszerben minden feladat (Work Type) útvonala determinisztikus.
- A termékigény már a rendszer indításakor ismert.
- A rendszerbe annyi alapanyag érkezik be, amennyi biztosítja a vizsgált időszak alatti folyamatos áramlást.
- Ha ugyanazon gépen egy feladat (Work Type) feldolgozását egy másiké követ, akkor a gépen átszerelést kell végrehajtani, mielőtt a következő termékfajta feldolgozásába kezdhetne (changeover).
- A gépek a feldolgozandó termékek beérkezési sorrendjében priorizálják a kötegek feldolgozását.
- Egy adott gép csak akkor kezdi el egy feladat kötegének feldolgozását, ha a köteg minden eleme rendelkezésre áll. (Mivel a vizsgált időszak alatt folyamatos áramlást biztosítunk, ezért a maradékok feldolgozásával nem kell foglalkoznunk, viszont a modellben ennek kezelése megoldott.)
- A rendszerben az anyagmozgatási időket figyelmen kívül hagyjuk.
- A modell futási ideje (vizsgált időszak) 20 műszak.
- Az elszállítandó termékek igénye naponta a munkaidő végén jelenik meg. Az igények generált, egyenletes eloszlású értékek. A bizonyos termékek átlagosan igényelt darabszámai között nincs szignifikáns eltérés.
- Késő termékek elszállítását napközben is megengedjük.

Az optimum feltétel

Az optimum feltétel megalkotásánál a költségminimalizálást tartottam szem előtt. A tárolási és átállási költségeket *1 pénzegység/perc*-ként vettem alapul, majd ezeket súlyoztam az általam vélt általános költségarányok szerint.

A célfüggvény értékét növelő elemek:

kötegnagyság optimális kombinációja. Kombináció esetén a változók számát a lehetőségek számának hatványára kell emelnünk, azaz ez esetben 46^{10} számú lehetőségünk van. Ez, ha egy futtatás 10mp-ig tart, akkor 13,45 milliárd évig futtathatnánk a szimulációt újra és újra, hogy minden lehetőséget kipróbálhassunk. Ezt viszonyítva ahhoz, hogy az ősröbbanás körülbelül 13,7 ($\pm 0,2$) milliárd évvel ezelőtt következett be, a kilátásaink enyhén szólva kétségbeejtőek. Természetes a számítási teljesítmény számos módon növelhető, de ennek költségei igen magasak is lehetnek. A következőkben e probléma egy megoldási alternatíváját fejteném ki.

Metaheurisztikák

A metaheurisztikákat gyakran használják kombinatorikus diszkrét problématerű optimalizálási feladatok megoldásához, de folyamatos problématerekben is használhatóak. Általános keretet nyújtanak fekete-doboz rendszerek optimalizálásához, ezáltal bármilyen típusú probléma megoldásához használható komolyabb változtatás nélkül. A metaheurisztikus algoritmusokon alapuló optimalizáló modulok uralkodó szerepet töltenek be a kereskedelmi szimulációs szoftverek piacán. Működésük során olyan algoritmusok szerint próbálják megtalálni az optimalizálandó rendszerek célfüggvényeinek legkedvezőbb értékét eredményező változókat, melyek lényeges mértékben redukálják annak megtalálásához szükséges próbálkozások számát.

Az 50-es évek óta számos optimalizációs algoritmust kifejlesztettek, mint például a tabukeresés, a szóródás keresés, valamint a genetikus, bakteriális, hangya-kolóniás, részecske-rajos, valamint iterált mohó algoritmusok.

Genetikus algoritmusok (Genetic Algorithms)

A genetikus algoritmusok olyan optimalizálási módszerek, amik a darwini, biológiai reprodukció során történő természetes kiválasztódás és mutáció analógiáján alapulnak. Ezek az algoritmusok potenciális megoldást hordoznak magukban specifikus problémák megoldására.

A genetikus algoritmus alkalmazásához öt komponens szükséges (Jósvai, Perger, 2010):

- A probléma kódolásához megfelelő megoldás – adott hosszúságú jelölési rendszer.
- Kiértékelési függvény, amely minden megoldás számára meghatároz egy értéket.
- Eljárás a populációk létrehozására.
- Operátorok, amelyek a szülők reprodukciója során biztosítják a genetikai összetétel változását, mint például a keresztezés, mutáció, ill. más terület specifikus operátorok.
- Az algoritmus paraméter beállítása, operátorok, és így tovább.

Működése során egy általunk létrehozott kezdőpopulációt, megadott számú generáción keresztül szaporítja. Ez alatt folyamatosan figyeli a célérték függvény értékét (fitness), és azon az egyedek génjeit szaporítja tovább, melyek esetén annak értéke jobb. Generációról generációra az egyedek megváltoznak a rekombinációs, mutációs operátoroknak köszönhetően. Számos szelekciós, illetve rekombinációs és mutációs operátort alkalmaznak a gyakorlatban, melyekre dolgozatomban nem térek ki.

Az optimalizáló szoftvercsomagok

Ma már a legtöbb szimulációs szoftvercsomag professzionális verziója tartalmaz optimalizáló modult. Mivel ezen szoftvereket problémák széles skálájának megoldására használják, ezért olyan optimalizációs megoldásra van szükség, mely hatékonyan elboldogul black-box jelegű problémákkal. E szoftverek széles körben elterjedtek a vállalati gyakorlatban, ezért figyelembe kell venni azt is, hogy ezen programok többnyire egyszerű PC-ken futnak, melyek számítási teljesítménye meg sem közelíti a nagy munkaállomásokét. A 6. ábra kereskedelmi szimulációs szoftverekhez kapható legfontosabb optimalizációs modulokat, azok fejlesztőit, valamint keresési stratégiáikat sorolja fel.

6. ábra.

A futtatások adatait tartalmazó Excel tábla

Optimization Package (Simulation Platform)	Vendor (URL)	Primary Search Strategies
AutoStat (AutoMod)	AutoSimulations, Inc. (www.autosim.com)	evolutionary, genetic algorithms
OptQuest (Arena, Crystal Ball, et al.)	Optimization Technologies, Inc. (www.opttek.com)	scatter search and tabu search, neural networks
OPTIMIZ (SIMUL8)	Visual Thinking International Ltd. (www.simul8.com)	neural networks
SimRunner (ProModel)	PROMODEL Corp. (www.promodel.com)	evolutionary, genetic algorithms
Optimizer (WITNESS)	Lanner Group, Inc. (www.lanner.com/corporate)	simulated annealing, tabu search

Forrás: Fu, 2002 p.2

Az OptQuest

Az OptQuest egy egyedülálló programcsomag, ami képes a piac számos szimulációs szoftvercsomagjával együttműködni. Széles körben alkalmazzák véges elem szimulációs szoftverek optimalizáló moduljaként (pl. Arena, Simul8, ED Developer, stb.). A szoftver több optimumkeresési algoritmust egyesít, mint a szóródás keresés és a tabukeresés a neurális hálózatok mentén. A program működése során a szimulációt fekete dobozként kezeli, egy teljesen különálló szoftverről van szó. A hangsúly a keresési algoritmusra helyeződik. (Fu, 2002)

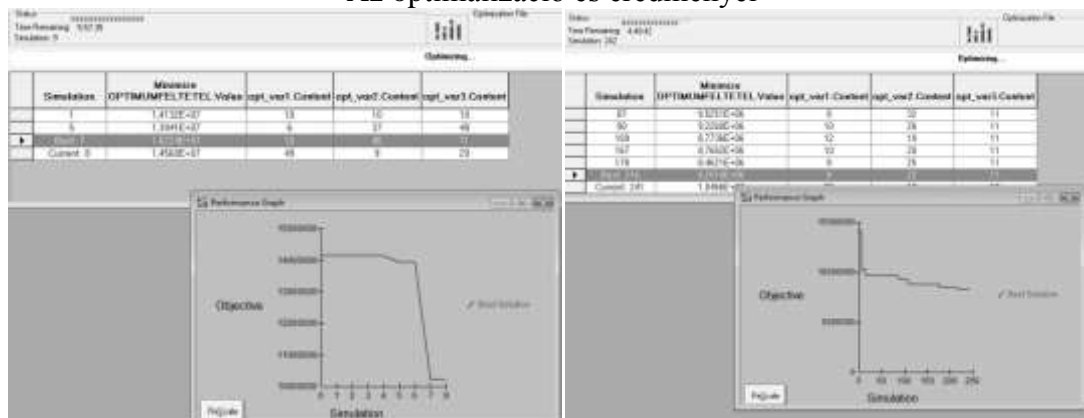
A vizsgált modell optimalizálásának eredményei

Először a 0 fejezetben létrehozott szimulációs modell termelési tétel nagyságainak optimalizációját futtattam le az OptQuestben. A problémateret minden optimum változónál az 5-50-ig terjedő intervallumban korlátoztam. Az algoritmus már a 169. próbálkozásból megtalálta az optimumot, ami egy órát sem vett igénybe. Ezután a futtatás még több mint 5 órán keresztül folytatódott, de nem vezetett jobb eredményre. Az optimalizáció teljesítménygrafikonja és eredményei a mellékletben találhatóak.

A második teszt során a szimulációs modell paramétereit úgy változtattam, hogy azok az egyes feladatok optimális kombinációjának szórását lényegesen megnövelje. Ebben az esetben is körülbelül egy óra alatt született optimálisnak tekinthető megoldás. A 216. futtatás vezetett az optimumhoz. A következő 5 óra és körülbelül 1200 futtatás sem talált a célfüggvénynek jobban megfelelő kombinációt. A futtatás részletes adatairól ad részletesebb képet a 7. ábra.

7. ábra.

Az optimalizáció és eredményei



Forrás: Saját szerkesztés

Fiktív modellekről lévén szó a kapott kombinációkat nem értékelem (a célérték-függvény definiálásával már úgyszólván előre megvettem). Egy 97336 elemű problémátér vélt optimumát a 169, illetve 216 futtatással megtalálta a szoftver, körülbelül egy óra alatt a lehetséges kombinációknak csak 0,2%-át megvizsgálva. Mindezt egy átlagos kétmagos processzorral rendelkező PC-n. Figyelembe kell vennünk, hogy a számítási teljesítmény növelésére számtalan alternatíva áll rendelkezésre, amik akár egy kkv-nak is megfizethetőek.

Összegzés

Dolgozatomban rávilágítottam a kistételes (small-lot) termelés előnyeire (a nagytételes-készletező (batch-and-queue) munkamódszerrel szemben) és létrehoztam egy szimulációs modellt, ami ennek megvalósítását nem csak, hogy támogatja, de még képes is az igényekhez és lehetőségekhez mért optimális ütemezési paraméterek kiszámítására is. Fejlesztése során a recirkulációt megengedő rugalmas job shop (műhelyrendszerű) és a gyártási tétel nagyság ütemező modelleket (*lot scheduling models*) vettem alapul. Segítségével kidolgozhatunk ütemezési alternatívákat a batch méretek optimumon tartására és kisebb átalakításokkal problémák széles skálájának megoldására lehet képes diszkrét és folyamatos rendszerek esetén egyaránt.

Körüljártam a felmerülő problémákhoz kapcsolódó témákat, mint a termelési modellek; a diszkrét, eseményvezérelt szimuláció és a metaheurisztikus optimalizációs eszközök. A későbbiekben a modellem különböző ütemezési stratégiák vizsgálatára, valamint konkrét, céges problémák megoldására alkalmazható és az ezekből nyert tapasztalatok segítségével fejleszhető.

Irodalom

- Bangsow, S. (2010): *Manufacturing Simulation with Plant Simulation and Simtalk*. Springer, Berlin
- Banks, J. (1998): *Handbook of simulation: principles, methodology, advances, applications, and practice*. Wiley-IEEE, h.n.
- Banks, J. et. al. (2009): *Discrete-Event System Simulation*. Prentice Hall, h.n.
- Concannon, K. et. al (2007): *Simulation Modeling with SIMUL8*. Visual Thinking International, h.n.
- Földesi Péter (2006): *Logisztika I-II*. Széchenyi István Egyetem (elektronikus jegyzet), Győr
- Fu, M. C. (2002): Feature Article: Optimization for simulation: Theory vs. Practice. *Journal on Computing archive*, 4., 192-215.
- Hauge, J. W.-Paige, K. N (2004): *Learning SIMUL8: The complete guide*. Plain Vu Publishers, Bellingham
- Jávor András (2000): *Diszkrét szimuláció*. BME Információmenedzsment Tanszék, Budapest
- Jones, A., Rabelo, L. (1998): Survey of Job Shop Scheduling Techniques. National Institute of Standards and Technology, California Polytechnic State University, Elérve: http://www.nist.gov/customcf/get_pdf.cfm?pub_id=821200 [letöltve: 2012.04.13.]
- Jósvai J., Perger J. (2010): Digitális gyár, módszerek és alkalmazások termelési környezetben. *Logisztikai innovációs füzetek*, 3., 2-10.
- Kardos K., Jósvai J. (2008): Termelési rendszerek számítógépes kezelése Gyártási és logisztikai folyamatok tervezése szimulációs eljárással. *Gyártástrend*, 4., 36-37.
- Kardos Károly, Jósvai János (2006): *Gyártási folyamatok tervezése*. Széchenyi István Egyetem (elektronikus jegyzet), Győr
- Pinedo, M. L. (2005): *Planning and scheduling in manufacturing services*. Springer Science+Business Media, h.n.
- Prezenszki J. (1999) (szerk.): *Logisztika II (Módszerek, eljárások)*. Logisztikai Fejlesztési Központ, Budapest

- Prezenszki J. (2004) (szerk.): *Logisztika I (Bevezető fejezetek)*. BME Mérnöktovábbképző Intézet, Budapest
- Szily István (1994): *Döntéselőkészítés I-II*. SZIF, Győr
- Whitley, D. (1995): A Genetic Algorithm Tutorial. *Statistics and Computing*, 4., 65-85.
- Womack, James P.-Jones, Daniel T (2009): *Lean szemlélet*. HVG Kiadó, Budapest
- Zapata, J. C. - Pekny, J.- Reklaitis, G. V. (2011): *Simulation-Optimization in Support of Tactical and Strategic Enterprise Decisions* In: Kempf, K. G.- Keskinocak, P. - Uzsoy, R. (ed.) (2011): *Planning Production and Inventories in the Extended Enterprise*. Springer Science+Business Media, h.n.