

SZIMULÁLT HŰTÉS KERESŐ ALGORITMUS ALKALMAZÁSA FLEXIBILIS JOB SHOP ÜTEMEZÉSI FELADATOK MEGOLDÁSÁRA

Hablik Róbert

szoftverfejlesztő, ETEO Software Factory Kft.
3526 Miskolc, Macropolis, e-mail: robert.hablik@eteo-sf.hu

Szabó Martin

egyetemi tanársegéd, Miskolci Egyetem, Informatika Intézet, Alkalmazott Informatikai Intézeti Tanszék
3515 Miskolc, Miskolc-Egyetemváros, e-mail: uitszabo@uni-miskolc.hu

Absztrakt

A széles körben elterjedt és alkalmazott korszerű termelésinformatikai eszközök és rendszerek a termelő tevékenységet folytató vállalatok hatékonyabb és gazdaságosabb működését teszik lehetővé. A vállalatok célja mindig a nyereség maximalizálása és a termékminőség növelése. A munkavégzés hatékonyságának növelése fejlett informatikai módszerekkel és a termelést célszerűen támogató gépi eszközök bevezetésével lehetséges. Az emberi munkaerő alkalmazása továbbra is nélkülözhetetlen, tehát célszerű az olyan rendszerek létrehozása, amelyek lehetővé teszik az emberi beavatkozást is (felügyelet, vezérlés és irányítás). Implementáltunk egy olyan termelésinformatikai alkalmazást, amely alkalmas a rugalmas, több utas, több operációs termelési feladatok ütemezésére, a kiválasztott ütemezési szabály és a kiválasztott célfüggvény alapján. A cikk a kísérleti szoftver azon részletét mutatja be, amely a rugalmas ütemezési feladatokat szimulált hűtés kereső algoritmus bevezetésével oldja meg. Az elméleti alapok és az alkalmazott kutatási módszerek áttekintését követően bemutatjuk az algoritmussal elért eredményeket, továbbá a publikáció egy összehasonlító elemzést is tartalmaz. A cikk Hablik Róbert „Diszkrét ütemezési feladatok reaktív megoldási módszereinek implementálása és összehasonlító elemzése” című szakdolgozatának a kivonata.

Kulcsszavak: termelésinformatika, diszkrét termelésütemezés, szimulált hűtés, lokális keresési módszerek, globális keresési módszerek

Abstract

The use of modern production IT tools and systems is essential for the more efficient and economical operations of production companies. The goal of companies is always to maximize profits and minimize production waste. Increasing work efficiency is possible only with the introduction of different IT methods and machine tools that expediently support production. However, the use of human labor is still essential, so it is advisable to set up systems that also allow for human intervention (supervision, control, and management). We have implemented a production informatics application that is suitable for scheduling one-path, one-pass, multi-operational tasks based on a preselected scheduling rule and objective function. The paper introduces an experimental software module that can schedule tasks with a simulated annealing algorithm, implemented to achieve better results. After presenting the theoretical foundations and the applied research methods, the publication also includes the results obtained with the algorithm and comparative analysis. The article is an excerpt from Róbert Hablik's dissertation entitled “Diszkrét ütemezési feladatok reaktív megoldási módszereinek implementálása és összehasonlító elemzése”

Keywords: *production informatics, discrete production scheduling, simulated annealing, local search methods, global search methods*

1. Bevezetés

Napjainkban a vállalatok egyik legnagyobb belső mérőszáma a beérkezett megrendelések teljesítésének mutatója. Ezt a mérőszámot több módszerrel is magasan lehet tartani, de nem mindegyik egyformán hatásos. Vegyük az előregyártás esetét, ahol tervezhető nagyobb készletszint, amennyiben a vállalat előre tudja a jövőbeni vevői igényeket. Ha a vevői igény megváltozik, nagy mennyiségű, nehezen eladható készletük maradhat a raktárban, amivel a vállalat profitot veszít. Továbbá az ilyen módon történő gyártás sérti a Lean [1][2] elveken alapuló *pull*, azaz a húzó elvű gyártást, ami alapján a gyártás csak a beérkezett megrendeléseket veszi figyelembe. Ez a modern vállalatok egyik alapelve, alapja a Toyota módszer.

Jó mutatókat lehet elérni azzal, ha vállalat egy megbízható ellátási láncsal dolgozik, azonban így is történhetnek késések, megrendelés módosítások, alapanyag hibák. Ilyen esetekben, annak érdekében, hogy a gyártás ne álljon meg, újra kell ütemezni az elvégzendő feladatok listáját. Az ütemezés elvégzésére több különféle módszer létezik, azonban hatékonyságukat mindig az adott feladat határozza meg. Ezért a megfelelő módszer kiválasztása minden esetben magas szakmai tudást feltételez. Például más módszert érdemes alkalmazni több műveletű, kis vagy közepes mennyiségű munkák esetén, ha kevés gép áll rendelkezésre, és más módszert, ha sok munkát kell végrehajtanunk, viszont több gép áll rendelkezésre, de szorosabb határidők vannak megszabva.

A cikkben kiértékelt teszteseteket egy rugalmas gyártórendszerek [3] reaktív és prediktív szabály alapú ütemezésének megoldására implementált, saját fejlesztésű alkalmazással készítettük el. A szoftvert kiegészítettük egy olyan modullal, amely a szimulált hűtés algoritmus bevezetésével támogatja az NP-nehéz ütemezési feladatok megoldását. Ebben az esetben prediktív ütemezést alkalmaztunk, tehát kizárólag az előre ismert munkák ütemezésével foglalkoztunk. Az algoritmus a Critical Ratio (CR) ütemezési szabály által adott eredményeket használja fel, mivel a megelőző kutatások során, az alkalmazott feltételek mellett, ez a szabály biztosította a legmegfelelőbb eredményeket nagy mennyiségű beérkező munka esetén. Jelen publikáció csak ennek a modulnak a bemutatását, az elért eredményeket és egy összehasonlító elemzést tartalmaz.

2. Szimulált hűtés kereső algoritmus bemutatása

A szimulált hűtés egy olyan lokális kereső algoritmus (meta-heurisztikus), amely nem áll meg el egy lokális optimum elérése esetén, hanem végrehajtása véletlenszerű csúcspont kiválasztásokkal tovább folytatódik. A megfelelő hűtési stratégia mellett az algoritmus véges lépésben megközelíti az optimum értékét. Az egyszerű megvalósíthatóság, az algoritmus konvergenciára vonatkozó tulajdonságai és a helyi optimum elkerülése érdekében alkalmazott hegymászó lépések miatt két évtized alatt népszerű és széles körben alkalmazott technikává vált. Általában diszkrét, de kisebb mértékben a folytonos optimalizálási problémák megoldására is használják. A szakirodalomban több olyan cikket találhatunk, amelyek jó áttekintést adnak a szimulált hűtés elméleti fejlődéséről és alkalmazási területeiről. Eglese [4] bemutatja a szimulált hűtés algoritmusát és néhány jelentős elméleti eredmény segítségével az alkalmazásában rejlő lehetőségeket. A szerzők végig vezetnek az algoritmus implementációjának lépésein és áttekintést nyújtanak az elvégzett kísérleteik eredményeiről. Továbbá javaslatot tesznek az algoritmus teljesítményének javítására, a „tisztá” megközelítési módszer módosításával. Az [5][7] publikációk áttekintést nyújtanak a tématerület aktuális állapotáról és a legfontosabb eredményekről.

Továbbá a szerzők bemutatják saját megvalósításaikat is, például a [5] publikáció egy gráf elméleti probléma optimális megoldásának megkeresésével illusztrálja az algoritmus alkalmazhatóságát. Koulamas et al. [6] operációkutatási feladatokon vizsgálja meg az algoritmus alkalmazhatóságát. Arra a következtetésre jutottak, hogy a szimulált hűtés jól alkalmazható a hagyományos (egygépes, flow shop, job shop, stb.) és nem hagyomány (gráf színezés, számok particionálása, stb.) problémák megoldására is, viszont a lokális optimum elkerülése („beragadás”) érdekében lényegesen több számítási erőforrást igényel. Aarts és Korst [8], valamint Van Laarhoven és Aarts [9] könyveket is jelentettek meg a témában. Előbbi [8] a kombinatorikus optimalizálás sztochasztikus megközelítési módszereit ismerteti. A könyv három nagyobb részre bontható fel. Az első a korlátozó feltételek nélküli optimalizációval, a második a lineáris programozási problémák megoldására használt módszerekkel, a harmadik pedig a nemlineáris, az integer és a dinamikus programozással foglalkozik. Aarts és Lenstra [10] egy külön fejezetben mutatják be a diszkrét optimalizálási problémák megoldására alkalmazható lokális keresési algoritmusokat.

A diszkrét optimalizálási feladatoknál alkalmazott szimulált hűtési algoritmus minden iterációjában a célfüggvény két megoldásához generál értékeket (aktuális és az újonnan kiválasztott megoldásokhoz), majd ezeket összehasonlítja. A jobb megoldásokat minden esetben elfogadják, az alacsonyabb szintű, nem javító megoldások egy részét szintén elfogadják abban bízva, hogy így elkerülhetik a lokális optimumot a globális optimum megtalálása érdekében. A nem javító megoldások elfogadásának valószínűsége egy hőmérsékleti paramétertől függ, melynek értéke az algoritmus iterációi során jellemzően nem növekszik. A módszer egyik legfontosabb tulajdonsága, hogy a lokális optimum elkerülése érdekében hegymászó lépéseket alkalmazhat (például olyan lépéseket, amelyek a célfüggvény értékét elrontják). Ha a hőmérséklet értéke nullára csökken, a hegymászó lépések előfordulása is csökken. A megoldás eloszlása összekapcsolható az inhomogén Markov-lánccal, amely az algoritmust olyan formára konvergálja, amelyben az összes előforduló valószínűség a globális optimummal rendelkező megoldások halmazára koncentrálódik. Feltéve, hogy az algoritmus konvergens, ellenkező esetben a lokális optimumhoz konvergál, amely vagy a globális optimum vagy sem.

Az algoritmus pszeudó-kódja a következő [4]:

1. Kezdeti megoldás kiválasztása $\omega \in \Omega$
2. Hőmérséklet változtatási értékének kiválasztása: $k = 0$
3. Hűtési ütemterv kiválasztása: t_k
4. Kezdeti hőmérséklet kiválasztása: $T = t_0 \geq 0$
5. Ismétlési ütemterv kiválasztása (M_k). A terv meghatározza az egyes hőmérsékleteken végrehajtott ismétlések számát (t_k).
6. Ismétlés (ciklus):
 1. Ismétlési számláló beállítása: $m = 0$
 2. Ismétlés (ciklus):
 1. Megoldás előállítása: $\omega' \in N(\omega)$
 2. $\Delta_{\omega, \omega'} = f(\omega') - f(\omega)$ kiszámítása
 3. Ha $\Delta_{\omega, \omega'} \leq 0$, akkor $\omega \leftarrow \omega'$
 4. Ha $\Delta_{\omega, \omega'} > 0$, akkor $\omega \leftarrow \omega' \exp(-\Delta_{\omega, \omega'} / t_k)$ valószínűséggel
 5. $m \leftarrow m + 1$
 3. Ismétlés vége, ha $m = M_k$

4. $k \leftarrow k + 1$

7. Ismétlés vége, ha a megállítási feltétel teljesül

3. Szimulált hűtés alkalmazása diszkrét ütemezési feladatokra

Jelen esetben, a kísérleti alkalmazás többi moduljától eltérő módon, prediktív termelésütemezés mellett döntöttünk. A cikkben tárgyalt termelésütemezési példa célfüggvénye a munkák késésének minimalizálására irányult. A keresési algoritmus egy korábbi kutatásunkra alapozva a *Critical Ratio* (CR) ütemezés eredményeit használja fel, mivel ez adta a legjobb eredményt az adott feltételek, nagymennyiségű beérkező munka és valósághoz közeli paraméterek esetén.

Adott a kezdőhőmérséklet, amely minden ciklus lefutása után csökken, úgy, hogy az aktuális értéke szorozódik egy 0 és 1 közötti valós számmal. A hűtés akkor ér véget, ha a kezdőhőmérséklet lecsökken egy adott pontra, amely a vizsgált esetben 0,06 fok. A hűtés során a munka két véletlenül kiválasztott operációja megcserélődik, majd az új sorrendet alkalmazva a számítás újból lefut. Ha az új sorrendben előállított munkák összes késése kevesebb, mint az aktuális sorrendben előállított munkák összes késése, akkor az új sorrend rögzítésre kerül és a továbbiakban az algoritmus ezt használja az összehasonlítás alapjául. Ha a kapott megoldás nem javító jellegű, akkor egy adott valószínűség alapján kerülhet rögzítésre, amelynek a kiszámítása az alábbi függvény alapján megy végbe [11]:

$$\text{Math.exp}((\text{prevLate} - \text{currLate}) / \text{temperature}) > \text{Math.random()} \quad (1)$$

Látható tehát, hogy a kezdi időszakban a keresés nagyobb eséllyel haladhat rossz irányba, azaz olyan kiszámított eredmények kerülhetnek rögzítésre, amelyek az előző eredményeknél rosszabbak [12].

A szimulált hűtés az ütemezési sorrend felállítása során az előzőleg elmentett ütemezés sorrendjét veszi alapul, majd megcserél két tetszőleges operációt. Mivel a műveletek végrehajtási sorrendje kötött, ezért az azonos munkához tartozó operációk sorrendjének módosítására nincs lehetőség. Ennek biztosítására minden operáció rendelkezik egy további paraméterrel, melynek értéke az azt kötelezően megelőző művelet azonosító száma. Ha azonos munkán belül nincs ilyen operáció, tehát a vizsgált művelet a munka legelső végrehajtandó eleme, akkor ez az érték -1.

A csere megvalósítására egy segédtablát használtunk, amely tartalmazza az operációk sorrendjét (a megelőző lefutás alapján), majd kiválasztottunk belőle két tetszőleges műveletet. Mivel a tábla mérete a tartalmazott operációk mennyiségétől függően dinamikusan változik, a csere végrehajtásakor a sorrend alapján később következő operáció törlésének meg kell előznie a tőle korábban bekövetkező operáció eltávolítását. Fontos, hogy az operációk beillesztésekor a kisebb sorszámú művelet kerül előbb a listába. Ellenkező esetben a korábban beillesztett elemek is elmozdulnak egy hellyel.

A cserét követően a kísérleti alkalmazás elvégzi az újra-ütemezést az új operáció végrehajtási lista alapján. Az iteráció addig folytatódik, amíg az algoritmus el nem éri a globális optimumot. Mivel azonban ez nem garantálható, az iterációt a beállított futási korlát is leállíthatja.

4. Eredmények

Az ütemezési folyamat egy rugalmas többutas (Flexible Job Shop) modellen alapszik, tehát egy adott operáció elvégezhető az arra alkalmas gépek (munkahelyek) bármelyikén. Ezt a modell típust a szakirodalomban az gyakran az FJ vagy FJS rövidítéssel jelölik [3]. Az implementált modell a következőket feltételezi:

- Az operációk végrehajtási sorrendje kötött.
- Az operációk közötti átállási időket a műveleti idő tartalmazza.

- Az azonos típusú operációk eltérő műveleti idővel is létrehozhatók.
- Azonos típusú gépekből több is rendelkezésre állhat.

A kereső algoritmus célfüggvénye a munkák csúszásának (T) minimalizálása. A cél nem a késő munkák darabszámának (N_L) csökkentése, hanem az összes csúszás (T_{sum}) csökkentése. Ebből következik, hogy több munka is átlépheti saját határidejét, azonban, ennek ellenére a korábban tesztelt módszerekkel összehasonlítva az összes késés még így is kevesebb. A munka késésének értékét a munka befejezési időpontjának és határidejének különbsége adja meg. Amennyiben a kiszámolt érték negatív, a munka siet. Ha a késés értéke 0, akkor a munka éppen időben végzett, ha az érték pozitív, a munka késik. A munka csúszásának értékét a kiszámolt késés értéke adja meg. Pozitív érték esetén a késés és a csúszás értéke megegyezik. Negatív és 0 érték esetén a csúszás értéke 0 lesz.

A kereső algoritmus kiindulási alapja a korábbi CR ütemezési szabály alapján kiszámított megoldás (1. táblázat), mivel a korábbi kutatások során szerzett tapasztalatok alapján ez a szabály biztosítja darabszámra a legkevesebb csúszó munkát. Ebből következik, hogy azoknál a munkáknál, ahol a határidő előtti teljesítés jóval az elvárt mutatószám felett volt, lehetőség van egy - a vizsgált időpontban - potenciálisan kritikus állapotban lévő munka késleltetésre.

1. táblázat. Munkák késései (az értékek másodpercben értendők)

A munkák késései					
Név	Indítási idő	Futási idő	Határidő	Befejezési idő	Csúszás
Job_1	0	178	208	225	17
Job_4	1	112	140	152	12
Job_15	5	171	197	193	0
Job_2	6	185	218	250	32
Job_10	8	148	177	161	0
Job_7	12	78	115	157	42
Job_6	17	145	176	249	73
Job_11	18	118	145	280	135
Job_13	23	129	162	287	125
Job_9	33	179	205	232	27
Job_3	35	70	90	180	90
Job_12	36	135	158	181	23
Job_5	21	91	123	184	61
Job_8	23	109	133	246	113
Job_14	19	120	153	297	144
T_{sum}					894

A szimulált hűtés tesztelése során a bemenő munkák számát 15-re, a munkákhoz tartozó operációk számát maximum 3-ra állítottuk be. Az algoritmus kezdőértékét 5000-re határoztuk meg, így csökkentve az esélyét annak, hogy a szimulált hűtés során, kezdeti magas hőmérséklet esetén rossz értékeket vehessen fel az algoritmus. A kilépési feltétel, azaz a hűtés végpontja 0.06 fokra lett beállítva, az alfa értéke pedig 0.9. Ezzel a konfigurációval az ütemezési folyamat 108 alkalommal futott le. A kiértékelés

során a várt hatás érvényesült, azaz sikerült csökkenteni az operációk késését. A 2. táblázatban láthatóak a kereső algoritmus által elért eredmények. Látható, hogy a határidő előtt befejeződött munkák száma nem változott (N_T), azonban a cél az összes csúszás (T_{sum}) csökkentése volt, ami jelen esetben a korábbi 894 másodpercről 763 másodpercre csökkent.

2. táblázat. A munkák késései szimulált hűtést alkalmazva.

A munkák késései					
Név	Indítási idő	Futási idő	Határidő	Befejezési idő	Csúszás
Job_1	0	178	208	258	50
Job_4	1	112	140	154	14
Job_15	5	171	197	193	0
Job_2	6	185	218	323	105
Job_10	8	148	177	161	0
Job_7	12	78	115	151	36
Job_6	17	145	176	202	26
Job_11	18	118	145	215	70
Job_13	23	129	162	252	90
Job_9	33	179	205	289	84
Job_3	35	70	90	136	46
Job_12	36	135	158	181	23
Job_5	21	91	123	184	61
Job_8	23	109	133	243	110
Job_14	19	120	153	251	98
T_{sum}					763

A 1. és 2. táblázatot összehasonlítva látható, hogy a legtöbb munka befejezési ideje és ez által a késés mértéke is megváltozott. A legtöbb esetben a csúszás továbbra is pozitív értéket vett fel, viszont összességében sikerült csökkenést elérni.

5. Összefoglalás

A kísérleti alkalmazás elsődleges funkciója, hogy a bemeneti adatok alapján, különböző ütemezési szabályok alkalmazásával előállítson egy megfelelő ütemezési tervet. Az adatok felvitele történhet manuálisan is, viszont a tesztelési folyamatok gyorsítását elősegítve, beállítható korlátozó feltételek mellett lehetőség van az adatok generálására is. A szoftver egy különálló modulja szimulált hűtés kereső algoritmust használ, amely a kutatások alapján, a szabály alapú módszerekkel összehasonlítva, polinomiális futási időben nem megoldható ütemezési feladatok esetén jobb eredmény elérését teszi lehetővé. A modul prediktív, azaz előidejű ütemezést alkalmaz. Az ütemezési folyamat rugalmas többutas modellen alapszik. Több ugyanolyan típusú gép is rendelkezésre áll, a megegyező típusú operációk bármelyiken elvégezhetők. Az operációk sorrendje kötött, viszont a műveleti idők operáció-típuson belül is eltérhetnek. Az ütemezési folyamat végrehajtása után az eredmények táblázatos formában jelennek meg és Gantt-diagramok segítségével elemezhetők.

Az elkészített szoftver segítségével sikerült meghatározni, hogy különböző szélsőséges feltételek mellett, mely ütemezési szabályok alkalmazása a célszerű. Továbbá a szimulált hűtés kereső algoritmus hozzáadásával sikerült bizonyítani, hogy NP-nehez feladatok esetén, az optimálshoz közeli állapot eléréshez egyetlen ütemezési szabály alkalmazása nem feltétlenül elegendő, szükség van valamilyen kereső algoritmus használatára is, így jobb eredmények érhetők el. A kutatás során elért eredmények jó kiindulási alapként szolgálnak hasonló körülmények között végrehajtandó munkák ütemezésére.

6. Köszönetnyilvánítás

A kutató munka az Európai Unió és a magyar állam támogatásával, az Európai Regionális Fejlesztési Alap társfinanszírozásával, a GINOP-2.3.4-15-2016-00004 projekt keretében valósult meg, a felsőoktatás és az ipar együttműködésének elősegítése céljából.

Irodalom

- [1] Jasti, N. V. K., Kodali, R. *Lean production: literature review and trends*, International Journal of Production Research, 53(3), (2015) pp. 867-885.
<https://doi.org/10.1080/00207543.2014.937508>
- [2] Gupta, S., Jain, S. K. *A literature review of lean manufacturing*, International Journal of Management Science and Engineering Management, 8(4), (2013) pp. 241-249.
<https://doi.org/10.1080/17509653.2013.825074>
- [3] Tóth, T. *Termelési rendszerek és folyamatok*, Miskolc, 2004.
- [4] Eglese, R. W. *Simulated annealing: a tool for operational research*, European Journal of Operational Research, 46, (1990) pp. 271–281. [https://doi.org/10.1016/0377-2217\(90\)90001-R](https://doi.org/10.1016/0377-2217(90)90001-R)
- [5] Fleischer, M. A. *Simulated annealing: past, present, and future*, In: C. Alexopoulos, K. Kang, W.R. Lilegdon and D. Goldsman (eds.), Proceedings of the 1995 Winter Simulation Conference, IEEE Press, pp. 155–161.
- [6] Koulamas, C., Antony, S. R. Jaen, R. *A survey of simulated annealing applications to operations-research problems*, OMEGA-International Journal of Management Science, 22, (1994) pp. 41–56. [https://doi.org/10.1016/0305-0483\(94\)90006-X](https://doi.org/10.1016/0305-0483(94)90006-X)
- [7] Romeo, F., Sangiovanni-Vincentelli, A. *A theoretical framework for simulated annealing*, Algorithmica, 6, (1991) pp. 302–345. <https://doi.org/10.1007/BF01759049>
- [8] Aarts, E. H. L., Korst, J. *Simulated Annealing and Boltzmann Machines: A stochastic approach to combinatorial optimization and neural computing*, John Wiley & Sons, Chichester, England, 1989.
- [9] van Laarhoven, P. J. M. *Theoretical and computational aspects of simulated annealing*, Centrum voor Wiskunde en Informatica, Amsterdam, Netherlands, 1988.
- [10] Aarts, E. H. L., Lenstra, J. K. *Local search in combinatorial optimization*, John Wiley & Sons, Chichester, England, 1997.
- [11] van Laarhoven, P. J. M., Aarts, E. H. L. *Simulated annealing: Theory and Applications*, D. Reidel, Kluwer Academic Publishers, Dordrecht, Boston, Norwell, Massachusetts, 1987.
<https://doi.org/10.1007/978-94-015-7744-1>
- [12] Aszalós, L., Bakó M. *Fejlett keresőalgoritmusok*, Debreceni Egyetem, Debrecen, 2012.