

Simon Pál<sup>1</sup> – Varga Zoltán<sup>2</sup>

## GYÁRTÓRENDSZEREKBE NAPJAINKBAN ALKALMAZOTT TERMELÉSÜTEMEZÉSI MÓDSZEREK BEMUTATÁSA<sup>3</sup>

*Napjaink termelő és szolgáltató vállalatai egyre nagyobb figyelmet fordítanak a logisztikai igények teljesítésére. Az egyre szélesebb körben elterjedő "lean" filozófia egyre szigorúbb követelményeket támaszt a termelési és logisztikai költségek csökkentése terén. A legnagyobb mértékű költségcsökkentést hatékony termelés ütemezéssel és logisztikai optimalizálással lehet elérni. A két kutatási terület között számos hasonlóság, és szoros összefüggés fedezhető fel. A termelés ütemezés célját úgy lehet definiálni, hogy a rendelkezésre álló erőforrások optimális kiosztása oly módon, hogy az igények által támasztott feltételeket teljesítsék. Ezekben a feltételekben számos logisztikai szempont is fontos szerepet játszik. Jelen cikk célja bemutatni, összehasonlítani a manapság alkalmazott ütemező módszereket, amikkel a termelő rendszerek hatékonysága növelhető.*

### EXAMINATION OF SCHEDULING METHODS FOR PRODUCTION SYSTEMS

*Nowadays manufacturing and service companies pay more attention to meet logistical demands. The widespread lean philosophy establishes claims to reduce production and logistic costs. The biggest cost reduction can be obtained by effective scheduling algorithms and logistics optimization. Several similarities and a close relationship can be seen between the two research areas. The aim of production scheduling can be defined as the allocation of available production resources in order to satisfy the criteria set by demands. These criteria contain a lot of logistical aspects, which also play important roles. Typically, the scheduling problem involves a set of tasks and an objective function, which aims to find a balance between early completion, stock and frequent production changeovers. Since the production processes can be diverse and unique, there are several different production models and scheduling algorithms. The aim of this article is to present and compare the nowadays applied different scheduling algorithms, with which the efficiency of production systems can be increased.*

### A LOGISZTIKA ÉS A TERMELÉSÜTEMEZÉS KAPCSOLATA

A logisztikai egyik speciális területe a termelési rendszerekhez kötődik, ezt nevezik termelési logisztikának. Definíciója szerint, a termelési folyamatokhoz szükséges anyagok, gyártóeszközök, és a termelési folyamat részfolyamatainak összhangjához szükséges anyag- és hozzá kapcsolódó információáramlási folyamatok összessége. A termelési logisztikai rendszer feladata a megfelelő anyagellátás biztosítása a termelés során. Ennek megfelelően a termelésütemezés adta követelményeket figyelembe véve meghatározza, hogy például melyik technológiai sorhoz, géphez, milyen anyagmozgató eszközzel kell a bemenő tárolójáról kiszállítani. További feladatai közé tartozik az átfutási idők, a különböző ráfordítások csökkentése és a készletek minimalizálása. Ezek a feladatok a termelésütemezés során jelennek meg, melyek kulcsfontosságúak az optimális gyártás megvalósítása érdekében.

<sup>1</sup> mérnök informatikus, Nokia Networks pal.simon.8@gmail.com

<sup>2</sup> PhD hallgató, Miskolci Egyetem, zoli.varga86@gmail.com

<sup>3</sup> Lektorálta: Prof. Dr. Kovács Ferenc professzor emeritus, az MTA doktora, bgtkf@uni-miskolc.hu

Az ütemezés osztott erőforrások elosztása időről időre konkurens tevékenységek között. Ez a téma jelentős mennyiségű tudományos irodalmi mű témája volt. Ezekben a hangsúlyt olyan ütemezési problémákra helyezték ahol a tevékenységek munkák voltak, az erőforrások pedig megmunkáló gépek. Ezekben az esetekben általában egy gép, egy adott időben egy munkát tud végrehajtani. Az ütemezési probléma az egyik legfontosabb és legnehezebb kombinatorikus optimalizálási probléma, annak összetettsége és gyakorlati alkalmazásokban való gyakori alkalmazása miatt. Az ütemezés általános indoka erőforrások halmazát hozzárendelni munkákhoz, Pinedo szerint [1]. Mióta a szisztematikus ütemezési módszer megjelent az 1950-es években, azóta több száz cikk született különböző ütemezési problémákra. Ezek a következő kategóriába sorolhatók: egy-gépes, párhuzamos-gépes, open shop, job shop, flow shop ütemezési feladatok.

### Egy-gépes ütemezés

Az ütemezés koncepciója a késő '70-es évek óta releváns kutatási terület, amikor az alapvető fogalmakat bevezették [2]. Adott  $n$  munka, amelyeket  $J_1, \dots, J_n$  jelölhetünk és adott a gépek halmaza, amelyek egyszerre csak egy munkát hajtanak végre, ezekre a gépekre kell ütemezni a feladatokat. A gépek konfigurációjától függően megkülönböztetünk egy-gépes ütemezést, párhuzamos-gépes ütemezést és műhely modellt. Minden  $J_i$  ( $i = 1, \dots, n$ ) munkának van  $p_i$  végrehajtási ideje, amely egy  $J_i$  munka végrehajtásához szükséges idő. Adott egy  $\pi$  ütemterv, ahol a  $J_i$  munkának a kezdési ideje  $S_i(\pi)$  és a befejezési ideje  $C_i(\pi)$ ,  $\pi$  elhagyható, ha egyértelmű melyik ütemtervről van szó. Ha munkák megszakítása nem engedett, akkor  $C_i = S_i + p_i$ . Egy  $J_i$  munka végrehajtása függ az elengedési időtől  $r_i$ , amely a kezdési idő alsó korlátja vagy a határidőtől  $d_i$ , ami a kezdési idő felső korlátját határozza meg. Egy  $J_i$  munkának lehet  $w_i$  súlya a fontosságának meghatározására. Egy adott ütemtervben  $L_i = C_i - d_i$  definiálja a  $J_i$  munka késését,  $T_i$  pedig definiálja a lassúságát  $T_i = \max\{0, C_i - d_i\}$ . A maximális késés általánosítható maximális költségnek  $f_{max}$ , ami a következőképpen definiálható  $f_{max}(C) = \max_i\{f_i(C_i) \mid i = 1, \dots, n\}$ , ahol minden  $J_i$  ( $i=1, \dots, n$ ) munkának saját  $f_i(C_i)$  költség függvénye van. Bevezethetünk egy indikátor függvényt, ami jelzi, hogy az adott  $J_i$  munka késik ( $U_i = 1$ ) vagy időben van ( $U_i = 0$ ) egy adott ütemtervben. A késés ellentéte egy  $J_i$  munkának a koraiság, amit a következőképpen definiálhatunk  $E_i = \max\{0, d_i - C_i\}$ .

A leggyakrabban használt teljesítmény kritériumok a következők:

- maximális befejezési idő vagy átfutás

$$C_{max} = \max\{C_i \mid i = 1, \dots, n\}$$

- összesített (súlyozott) befejezési idő

$$\sum_{i=1}^n w_i C_i$$

- maximális késés

$$L_{max} = \max\{L_i \mid i = 1, \dots, n\}$$

- maximális lassúság

$$T_{max} = \max\{T_i \mid i = 1, \dots, n\}$$

- maximális költség

$$f_{max} = \max\{f_i(C_i) \mid i = 1, \dots, n\}$$

- összesített (súlyozott) késés

$$\sum_{i=1}^n w_i T_i$$

- maximális koraiság

$$E_{max} = \max\{E_i \mid i = 1, \dots, n\}$$

- összesített (súlyozott) koraiság

$$\sum_{i=1}^n w_i E_i$$

- (súlyozott) összege a lassú munkáknak

$$\sum_{i=1}^n w_i U_i$$

Nem feltétlenül csak egy kritérium használható, hanem a fent felsoroltak közül egyszerre több is. Gyakorlatban leginkább használt a késés és a koraiság kritériumok kombinációja, amely így Just in Time követelménynek való megfelelést határozza meg, azáltal, hogy mind a korai, mind a kései befejezést bünteti.

Az egy-gépes ütemezés jelentős figyelmet kapott a '70-es évektől, amikor az alapjait bemutatták [3]. Ezeket az alapokat tovább építve az utóbbi évtizedekben több jelentős eredményt publikáltak a termelés ütemezési feladatok megoldására a gyakorlatban. Ezekben a tanulmányokban az átfutási idő a felhasznált erőforrások mennyiségétől függ, amely Panwalkar és Pajagopalan (1992) [4], Li (1995) [5], Dirk Biskup (1999) [6] és Wen-Hung Kuo és Dar-Li Yang (2006) [7] publikációiban láthatóak.

### Párhuzamos-gépes ütemezés

Párhuzamos-gépes ütemezési feladatokban egynél több gép áll rendelkezésre, ezek a munkafolyamatok végrehajtása szempontjából egyformák és párhuzamosak. Jelölje ezeket  $M_1, \dots, M_m$ , ha adott  $n$  feladat  $J_1, \dots, J_n$  és ezeknek a végrehajtási idejük  $p_1, \dots, p_n$  egy  $m$  azonos, párhuzamos gépen, akkor az egy-gépes ütemezés teljesítmény kritériumai itt is ugyanúgy használhatóak.

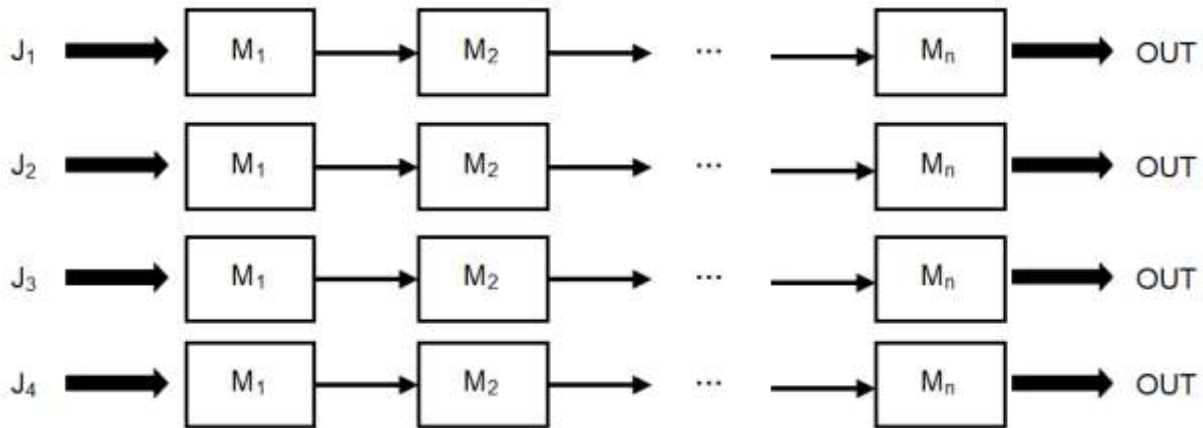
A párhuzamos-gépes ütemezést nem alkalmazzák gyakran, mert napjainkban már nem aktuális ütemezési feladatokban. A legtöbb kutatás a '90-es években zajlott és ekkor fektették le az eljárás alapjait [8][9]. Továbbá bizonyítást nyert, hogy ez egy NP-nehéz feladat, amelyek megoldására heurisztikus algoritmusokat célszerű alkalmazni [10][11].

Az utóbbi évtizedekben a kutatások a független párhuzamos-gépes ütemezési feladatokra irányultak [12][13][14].

### Shop ütemezés

A shop ütemezés gyűjtő fogalom, ami magába foglalja a job shop, flow shop feladatokat, amelyek széles körben használtak ipari termelési folyamatok modellezésére. Ezek a feladatok speciális esetei az általános shop feladatnak. Egy általános job shop modell látható az 1. ábrán.

Az általános shop feladat a következőképpen írható le. Adott  $n$  feladat  $J_1, \dots, J_n$  és  $m$  gép  $M_1, \dots, M_m$ . Minden  $i$  feladat  $O_{ij}$  ( $j=1, \dots, n$ ) műveletek halmazából áll, ahol  $p_{ij}$  a műveleti idő. Minden egyes  $O_{ij}$  művelet egy gépen kell végrehajtani a  $\{M_1, \dots, M_m\}$  gépek halmazából. Egy művelet csak egy gép által végezhető és egy gép csak egy műveletet tud végrehajtani egy időben. A cél egy végrehajtható ütemterv előállítása, a korábban már ismertetett kritériumoknak megfelelően.



1. ábra Általános Job Shop modell

## JOB SHOP ÜTEMEZÉSI MÓDSZEREK

A „Job Shop” ütemezési feladatok jellemzően a legösszetettebb ütemezési feladatok közé tartoznak [15]. Kombinatorikus optimalizáláson alapuló nem determinisztikus polinomiális idejű NP nehéz problémaként ismertek. A szakirodalomban számos jelölés elterjedt, többek között a JSS (Job Shop Scheduling), JSP (Job Shop Problem), vagy a JSSP (Job Shop Scheduling Problem). Jelen cikkben a JSSP rövidítést alkalmazzuk a későbbiekben. A Job Shop probléma munkák (job-ok) és azokon végzett műveletek halmazaként értelmezhető, mely munkákat gépeken hajtanak végre. A flow shop feladatokhoz képest a legnagyobb eltérés, hogy a job-ok több különböző gépen is végrehajthatók, azonban a műveletek sorrendje kötött.

Legfontosabb tulajdonságai:

- egy gép egyszerre egy műveletet képes elvégezni;
- ha egy művelet elkezdődött azt nem lehet félbeszakítani;
- egyes job-ok azonos műveletei megkötésekkel másik gépen is elvégezhetőek lehetnek bizonyos esetekben;
- a műveleti idők és a gépek száma előzetesen ismertek.

A JSSP-t számos helyen alkalmazzák, például gyártórendszerek esetében, melyek összetett komplex problémákat jelentenek. Jellemzően ilyen esetben minden feladat egy speciális problémát definiál, többek között ezért is lehetséges az ilyen jellegű feladatok megoldása számos módon, matematikai eljárásokkal, optimalizálási módszerekkel. A sok megoldási lehetőség közül azonban csak néhány eredményez egzakt megoldást, ezek is jellemzően csak bizonyos egyszerűbb, vagy speciális esetekre igazak. Napjaink a termelésütemezési feladatai nagyon jó kutatási területei a különböző mesterséges intelligencia módszereknek, sikerrel alkalmazhatóak a feladatok megoldására a mesterséges neurális háló, a hangyafarm (ant colony) vagy a méhkolónia (bee

colony) algoritmusok. Ezek a módszerek azonban heurisztikus jellegűek, vagyis nem egy optimális eredményt, állapotot eredményeznek, hanem csak egy ahhoz közelit. A közelítő érték többek között függ az algoritmus jóságától, és a feladat komplexitásától. Az eredeti JSSP jellemzően egy kritériumot szem elé helyező optimalizálási feladat, mely napjainkban nem állja meg a helyét. Manapság több célú optimalizálási feladatokról beszélhetünk, melyek sokkal közelebb állnak a valósághoz és mai kor igényeihez. Ezek célja, hogy több célfüggvény együttes szem előtt tartásával álljon elő a kívánt megoldás. A szakirodalom több célú JSSP csoporton belül is megkülönböztet dinamikus, és flexibilis alcsoportokat. Fontos megemlíteni az értelmezések sokszor az irodalomban is keverednek, ezért számos esetben nehéz élesen elválasztani, hogy pontosan milyen típusú JSSP-ről is beszélünk. A következő bekezdésekben ezek bemutatása következik.

### **Több célú JSSP, Dinamikus JSSP, flexibilis JSSP**

Egy speciális több célú ütemezési feladat (Multi objective JSSP) a dinamikus JSSP. Ezen feladatok olyan valós idejű eseményeket is figyelembe vesznek, mint például hirtelen fellépő gép leállások, vagy hirtelen fellépő munka (job) igények. Ezeket a fellépő állapotokat a legtöbb korábban kifejlesztett algoritmus nem kezeli megfelelően ezért számos kutató ezek megoldására saját mesterséges intelligenciát alkalmazó algoritmusokat fejlesztett ki, mint a tanuló ügynökök, genetikusan algoritmusok [16][17]. Adibi, Zandieh és Amiri kutatásaik során VNS (Variable neighborhood search) módszereket alkalmaztak [18].

### **Flexibilis JSSP**

A másik alcsoport a flexibilis JSSP, mely szintén sokkal közelebb áll a valósághoz, mint az egy célfüggvényű eredeti változat, ez egyfajta kiterjesztése annak. A flexibilitást az eredményezi, hogy az egyes műveletek bármely a relációban található géppel elvégezhetőek. Ennek köszönhetően a flexibilis JSSP egy komplexebb feladatot is jelent. Azáltal, hogy a gépek jelentős része képes más feladatok elvégzésére a hirtelen fellépő esetleges gép meghibásodásokból adódó termelési hatékonyság csökkenése sokkal kezelhetőbb [19].

2014-ben Yang és Gu újszerű tenyésztési eljárás alapján genetikusan Tabu algoritmust fejlesztettek ki (QSCGTA – Quad-space Cultural Genetic Tabu Algorithm), a probléma megoldására [20], míg Pérez és Raupp egy újfajta hierarchikus algoritmuson alapuló megoldási javaslatot tettek [21].

## **FLOW SHOP ÜTEMEZÉSI MÓDSZEREK**

A flow shop feladat egy speciális esete a shop feladatnak, ami  $n$  feladat  $m$  gépen való végrehajtásának tervezése miközben a cél adott célfüggvények minimalizálása és a következő feltételek és korlátok mellett:

- minden feladat  $m$  műveletből áll, melyek különböző műveleti idővel rendelkeznek;
- a megmunkálási folyamatok gépekhez rendelésének sorrendje kötött, pl.: minden munkadarabot először az 1. gépen kell megmunkálni, majd a 2. gépen, 3. gépen stb.

A flow shop feladat az egyik legnépszerűbb ütemezési feladat, napjainkban a termelő rendszerek, összeszerelő sorok, információ szolgáltató rendszerek majdnem negyedére jellemző ez a struktúra [22], [23] és [24]. Egyszerű két gépes esetben a Johnson algoritmus használható, ami



pontos megoldást eredményez. Azonban ha több mint két gép adott, a feladat túl összetett pontos megoldás kiszámításához, ezért heurisztikus vagy metaheurisztikus algoritmusok kerültek kidolgozásra az idők során.

Ilyen heurisztikus módszer lehet a szimulált hűtés (SA), tabu keresés (TS) és a genetikus algoritmus (GA). Általános flow shop feladat megoldására használható a szimulált hűtés, ami fémek hűtését modellezve keres közel optimális megoldást [25][26]. Ezzel szemben a tabu keresés permutációs flow shop feladatokra alkalmazható, ahol az összes feladat műveleti sorrendje egyforma. Könnyű implementálni és nem eredményez érezhetően rosszabb eredményt, mint az optimális általános flow shop [27][28].

Az SA és TS után a GA-t is alkalmazták FSSP megoldására. Reeves volt az első, aki alkalmazta és tanulmányában rávilágított arra, hogy az SA és GA-hoz hasonlóan jó megoldást eredményez a GA is, továbbá nagy feladatokra hatékonyabb, gyorsabban talál közel optimális megoldást [29].

Az FSP egy speciális esete mikor több párhuzamos gép van jelen egy-egy művelet végrehajtására, ezt a hybrid flow shop feladatnak (HFSSP) nevezik. Rubén és Vázquez-Rodríguez készített egy áttekintést a pontos, heurisztikus és metaheurisztikus módszerekről, különböző célfüggvényeket használva és a HFSSP változatokról [30].

A HFSSP megoldására több módszer is alkalmazható, mint például Mirsanei által bemutatott SA algoritmus, ami új és hatékonyabb szomszéd kiválasztó eljárást használ, ezáltal jobb eredményt ér el [31]. Božejko egy TS algoritmust mutatott be, amely hatékonyan futtatható több szálon [32]. Engin szintén több szálon futtatható GA algoritmust mutatott be, amiben új párhuzamosítható feldolgozást lehetővé tevő keresztező operátort használ [33]. Liu egy részecske raj optimalizáló (PSO) algoritmust mutatott be speciális lokális kereső operátorral és adaptív lokális keresést alkalmazott [34].

Napjainkban a just-in-time (JIT) kiterjedt alkalmazása a termelő rendszerekben szükségessé teszi, hogy a késés és a koraiság kritériumok együttes alkalmazását. A lot-streaming az egyik leghatékonyabb eljárás ezeknek a kritériumoknak együttes megfelelésére. Lot-streaming-nek a feladatok rész műveletekre bontását nevezzük. Ez a leghatékonyabb módszer idő alapú kritériumoknak megfelelő ütemezésre [35]. A részműveletekre bontás lehetővé teszi a műveletek átfedését, ezáltal csökkentve a gépek várakozási idejét.

Raj intelligenciát alkalmazó algoritmusok is hatékonyan alkalmazható ilyen feladatok megoldására. Tseng és Liao egy diszkrét PSO algoritmust mutattak be egy úgynevezett mozgási háló előny algoritmussal, amely biztosítja az optimális kezdési és befejezési idejét a rész műveleteknek egy adott feladatban [36]. Pan egy diszkrét mesterséges méhkolónia algoritmust (ABC) mutatott be lot-streaming flow shop feladatok megoldására, súlyozott koraiság és késés büntetéssel ideális és nem ideális esetekben. Az eredeti ABC algoritmushoz képest a bemutatott DABC algoritmusban az ételforrást a feladatok sorrendjei jelképezik és diszkrét műveleteket alkalmaz új szomszédos ételforrások előállítására a méhek számára [37].

Napjaink termelő, és gyártó vállalatai egyre nagyobb figyelmet fordítanak a termelési logisztika optimális működésére. Ennek részeként cikkünkben bemutatásra kerültek különböző termelés-ütemezési módszerek, mint az egy-gépes, párhuzamos-gépes, és különböző shop ütemezési módszerek, valamint manapság ezek megoldására alkalmazott módszerek. Mivel az ütemezési módszerek NP nehéz problémák, ezért egzakt megoldásuk csak speciális esetben létezik, illetve a feladatok megoldása valós időben jellemzően nem lehetséges. Ennek köszönhetően a metaheurisztikus megoldások terjedtek el az 1970-80-as évektől kezdődően, mint például a szimulált hűtés, a tabu keresés, vagy a különböző genetikus algoritmusok. Manapság a kutatások egyre inkább a raj intelligencia irányába mutatnak, párhuzamos algoritmusokkal. Miután a metaheurisztikus módszerek csak egy optimumhoz közeli megoldást eredményeznek, ezért számos kutató töri a fejét az újabb és pontosabb megoldásokon, ezért a jövőben nagy számú újabb és egyre jobb algoritmus megjelenése várható.

### FELHASZNÁLT IRODALOM

- [1] Pinedo, M. (2012) Scheduling: theory, algorithms, and systems. Springer, e-ISBN 978-1-4614-2361-4.
- [2] Graham, R. L.; Lawler, E. L.; Lenstra, J. K.; Kan, A. H. G. (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of discrete mathematics*, Vol. 5. pp. 287-326. ISBN 978-0-08-086767-0
- [3] Baker, K. R.; Baker, K. R. (1974) Introduction to sequencing and scheduling. New York: Wiley. p. 305
- [4] Panwalkar, S. S.; Rajagopalan, R. (1992) Single-machine sequencing with controllable processing times. *European Journal of Operational Research*, Vol. 59(2) pp. 298-302.
- [5] Li, C. L.; Sewell, E. C.; Cheng, T. C. E. (1995) Scheduling to minimize release-time resource consumption and tardiness penalties. *Naval Research Logistics (NRL)*, Vol. 42(6) pp. 949-966.
- [6] Biskup, D. (1999) Single-machine scheduling with learning considerations. *European Journal of Operational Research*, Vol. 115(1) pp. 173-178.
- [7] Kuo, W. H.; Yang, D. L. (2006) Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect. *European Journal of Operational Research*, Vol. 174(2) pp. 1184-1190.
- [8] Chen, Z. L. (1996) Parallel machine scheduling with time dependent processing times. *Discrete Applied Mathematics*, Vol. 70(1) pp. 81-93.
- [9] Cheng, T. C. E.; Sin, C. C. S. (1990) A state-of-the-art review of parallel-machine scheduling research. *European Journal of Operational Research*, Vol. 47(3) pp. 271-292.
- [10] Cheng, R.; Gen, M. (1997) Parallel machine scheduling problems using memetic algorithms. *Computers & Industrial Engineering*, Vol. 33(3) pp. 761-764.
- [11] Mosheiov, G. et al. (2001) Parallel machine scheduling with a learning effect. *Journal of the Operational Research Society*, Vol. 52(10) pp. 1165-1169.
- [12] Rabadi, G.; Moraga, R. J.; Al-Salem, A. (2006) Heuristics for the unrelated parallel machine scheduling problem with setup times. *Journal of Intelligent Manufacturing*, Vol. 17(1) pp. 85-97.
- [13] Fanjul-Peyro, L.; Ruiz, R. (2010) Iterated greedy local search methods for unrelated parallel machine scheduling. *European Journal of Operational Research*, Vol. 207(1) pp. 55-69.
- [14] Vallada, E.; Ruiz, R. (2010) A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. *European Journal of Operational Research*, Vol. 211(3) pp. 612-622.
- [15] French, S. (1982) Sequencing and scheduling: an introduction to the mathematics of the job-shop. Chichester: Ellis Horwood.
- [16] Sha, D. Y.; Liu, C. H. (2005) Using data mining for due date assignment in a dynamic job shop environment. *International Journal of Advanced Manufacturing Technology*, Vol. 25. pp. 1164-1174.
- [17] Dimitrov, T.; Baumann, M. (2011) Genetic algorithm with genetic engineering technology for multi-objective dynamic job shop scheduling problems. In: *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*. ACM, pp. 833-834.

- [18] Adibi, M. A.; Zandieh, M.; Amiri, M. (2010) Multi-objective scheduling of dynamic job shop using variable neighborhood search. *Expert Systems with Applications*, Vol. 37(1) pp. 282-287.
- [19] Jia, S.; Hu, Z. H. (2014) Path-relinking Tabu search for the multi-objective flexible job shop scheduling problem. *Computers & Operations Research*, Vol. 47. pp. 11-26.
- [20] Yang, Y.; Gu, X. (2014) Cultural-Based Genetic Tabu Algorithm for Multiobjective Job Shop Scheduling. *Mathematical Problems in Engineering*, Vol. 2014, Article ID 230719, 14 pages, 2014. doi:10.1155/2014/230719
- [21] Pérez, M. A. F.; Raupp, F. M. P. (2014) A Newton-based heuristic algorithm for multi-objective flexible job-shop scheduling problem. *Journal of Intelligent Manufacturing*, pp. 1-8. DOI: 10.1007/s10845-014-0872-0
- [22] Lee, W. C.; Wu, C. C. (2009) Some single-machine and m-machine flowshop scheduling problems with learning considerations. *Information Sciences*, Vol. 179(22) pp. 3885-3892.
- [23] Tavakkoli-Moghaddam, R.; Rahimi-Vahed, A.; Mirzaei, A. H. (2007) A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: weighted mean completion time and weighted mean tardiness. *Information Sciences*, Vol. 177(22) pp. 5072-5090.
- [24] Yin, Y. et al. (2009) Some scheduling problems with general position-dependent and time-dependent learning effects. *Information Sciences*, Vol. 179(14) pp. 2416-2425.
- [25] Osman, I. H.; Potts, C. N. (1989) Simulated annealing for permutation flow-shop scheduling. *Omega*, Vol. 17(6) pp. 551-557.
- [26] Ishibuchi, H.; Misaki, S.; Tanaka, H. (1995) Modified simulated annealing algorithms for the flow shop sequencing problem. *European Journal of Operational Research*, Vol. 81(2) pp. 388-398.
- [27] Nowicki, E.; Smutnicki, C. (1996) A fast tabu search algorithm for the permutation flow-shop problem. *European Journal of Operational Research*, Vol. 91(1) pp. 160-175.
- [28] Ben-Daya, M.; Al-Fawzan, M. (1998) A tabu search approach for the flow shop scheduling problem. *European Journal of Operational Research*, Vol. 109(1) pp. 88-95.
- [29] Reeves, C. R. (1995) A genetic algorithm for flowshop sequencing. *Computers & operations research*, Vol. 22(1) pp. 5-13.
- [30] Ruiz, R.; Vázquez-Rodríguez, J. A. (2010) The hybrid flow shop scheduling problem. *European Journal of Operational Research*, Vol. 205(1) pp. 1-18.
- [31] Mirsanei, H. S. et al. (2011) A simulated annealing algorithm approach to hybrid flow shop scheduling with sequence-dependent setup times. *Journal of Intelligent Manufacturing*, Vol. 22(6) pp. 965-978.
- [32] Bożejko, W.; Pempera, J.; Smutnicki, C. (2013) Parallel tabu search algorithm for the hybrid flow shop problem. *Computers & Industrial Engineering*, Vol. 65(3) pp. 466-474.
- [33] Engin, O.; Ceran, G.; Yilmaz, M. K. (2011) An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems. *Applied Soft Computing*, Vol. 11(3) pp. 3056-3065.
- [34] Liu, B.; Wang, L.; Jin, Y. H. (2007) An effective PSO-based memetic algorithm for flow shop scheduling. *Systems, Man, and Cybernetics, Part B: Cybernetics*, IEEE Transactions on, Vol. 37(1) pp. 18-27.
- [35] Chang, J. H.; Chiu, H. N. (2005) A comprehensive review of lot streaming. *International Journal of Production Research*, Vol. 43(8) pp. 1515-1536.
- [36] Tseng, C. T.; Liao, C. J. (2008) A discrete particle swarm optimization for lot-streaming flowshop scheduling problem. *European Journal of Operational Research*, Vol. 191(2) pp. 360-373.
- [37] Pan, Q. K. et al. (2011) A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information sciences*, Vol. 181(12) pp. 2455-2468.