Analiza primene pravila prioriteta na problemu raspoređivanja u fleksibilno protočnoj proizvodnji

ZORAN M. RAKIĆEVIĆ, Univerzitet u Beogradu,
Fakultet organizacionih nauka, Beograd
MIRKO B. VUJOŠEVIĆ, Univerzitet u Beogradu,
Fakultet organizacionih nauka, Beograd

Originalni naučni rad
UDC: 658.527

U ovom radu analizirana je grupa jednostavnih heurističkih metoda koje se koriste u rešavanju problema raspoređivanja u proizvodnji i pružanju usluga. Analiza je izvršena na primeru raspoređivanja u fleksibilno-protočnoj proizvodnji koji je poznat po engleskom akronimu FFS (Flexible-Flow Shop). Zadatak je odrediti raspored obrade više proizvoda na više mašina, pri čemu svi proizvodi slede isti redosled obrade i za svaku obradu postoji na raspolaganju više mašina. Za opisani problem FFS dat je odgovarajući matematički model mešovitog celobrojnog programiranja. Od potencijalnih metoda za rešavanje postavljenog zadatka detaljnije se razmatraju jednostavne heuristike jer je originalni zadatak NP tvrd i nalaženje tačnog optimalnog rešenja zahtevalo bi neprihvatljivo dugo računarsko vreme. Heurističke metode počivaju na pravilima prioriteta koja se izvode na osnovu relacija važnosti između proizvoda i trajanja njihovih obrada na pojedinačnim mašinama. Heurističke metode imaju široku primenu u rešavanju praktičnih problema i to je bila motivacija za analizu čiji se rezultati saopštavaju u ovom radu. Cilj analize je da se na jednom hipotetičkom primeru problema raspoređivanja odrede ona pravila prioriteta koja daju dobra rešenja pri čemu se ocenjivanje obavlja korišćenjem različitih kriterijumskih funkcija. Analiza je obavljena pomoću računarskog programa LEKIN. Kao glavni rezultat analize pokazano je da pravila prioriteta daju različita rešenja za problem FFS i da svako od tih rešenja predstavlja značajno dobar rezultat sa aspekta neke od razmatranih kriterijumskih funkcija.

Ključne reči: *Problemi raspoređivanja, protočna proizvodnja, fleksibilno-protočna proizvodnja, pravila prioriteta*

1. UVOD

Problem raspoređivanja (*Scheduling*) je u opštem slučaju određivanje prostornog i vremenskog rasporeda izvršavanja proizvodnih aktivnosti. On je deo procesa planiranja u svakodnevnom poslovanju proizvodnih preduzeća. Pronaći najbolji, tj. optimalan raspored, predstavlja veoma kompleksan problem kome je u naučnoj i stručnoj literaturi posvećena velika pažnja [1].

Problem raspoređivanja u širem smislu obuhvata: (1) problem asignacije (*Assignment*) - dodeljivanja određenih proizvodnih i uslužnih aktivnosti radnim centrima (koji čine ljudi i mašine koji imaju ograničen kapacitet), kao i (2) problem određivanja redosleda obavljanja određenih proizvodnih i uslužnih aktivnosti (*Sequencing*) po mašinama i izvršiocima i njihovo terminiranje tj. definisanja vremenskih termina za početak i završetak [2, 3, 4, 5].

Proizvodne sisteme karakterišu brojni faktori: broj proizvodnih resursa i/ili mašina, njihove karakteristike i konfiguracija, nivo automatizacije, tip sistema za manipulisanje u unutrašnjem transportu. Razlike u brojnim karakteristikama prouzrokuju pojavu različitih tipova proizvodnog procesa. Različiti tipovi proizvodnih procesa iziskuju upotrebu odgovarajućih modela za rešavanje zadataka planiranja proizvodnje [6]. Najviše korišćenu klasifikaciju problema raspoređivanja dali su Graham i dr. [7]. Oni modele raspoređivanja klasifikuju korišćenjem notacije $\alpha|\beta|\gamma$. Polje α opisuje proizvodno okruženje problema raspoređivanja. Polje β označava informacije o karakteristikama proizvoda koje je potrebno obraditi i ograničenjima procesa raspoređivanja. Polje γ sadrži infor-

Adresa autora: Zoran Rakićević, Univerzitet u Beogradu, Fakultet organizacionih nauka, Beograd, Jove Ilića 154

Rad primljen: 26.03.2014.

Rad prihvaćen: 03.04.2014.

macije o kriterijumskoj funkciji koja se koristi u procesu optimizacije problema raspoređivanja.

U ovom radu se opisuje jedan karakterističan problem raspoređivanja u fleksibilnoj protočnoj proizvodnji (*Flexible Flow Shop* - FFS). Problemu FFS odgovara matematički model mešovitog celobrojnog programiranja. U opštem slučaju, radi se o vrlo složenom modelu tako da problem pripada klasi NP tvrdih problema. Zbog toga se za rešavanje praktičnih problema FFS, po pravilu, koriste jednostavne heurističke metode poznate pod nazivom pravila prioriteta (*Dispatching or Priority rules*) [8], [9], [10].

U sledećoj sekciji biće opisan problem raspoređivanja u fleksibilno-protočnoj proizvodnji i dat odgovarajući model matematičkog programiranja. U trećoj sekciji biće predstavljen jedan karakterističan hipotetički primer i opisane heurističke metode za njegovo rešavanje. Predstavljen je numerički eksperiment na osnovu koga se uz pomoć programa LEKIN, vrši poređenje grupe metoda rešavanja problema FFS. Na osnovu analize rezultata (u delu 4) pokušava se dati odgovor na istraživačko pitanje: Koja od navedenih metoda rešavanja problema FFS, generiše dobra rešenja sa aspekta različitih performansi tj. kriterijuma vrednovanja?

2. PROBLEM RASPOREĐIVANJA U FLEKSIBILNO-PROTOČNOJ PROIZVODNJI

Problem raspoređivanja u protočnoj proizvodnji je problem rasporeda n proizvoda na m mašina u kome svi proizvodi imaju identičnu putanju obrade koju slede, a koja se sastoji od unapred određenog niza operacija na mašinama. Ukoliko u ovom proizvodnom sistemu za pojedine faze obrade postoji više istorodnih mašina u paralelnoj vezi onda je reč o jednom obliku fleksibilno-protočne proizvodnje. FFS je problem u kome od m mašina postoji *c* istorodnih grupa mašina namenjenih proizvodima sa istim tipom obrade. Na taj način raspoređivanje proizvoda na obradu postaje mnogo fleksibilnije, jer proizvodi mogu biti obrađivani na bilo kojoj mašini iz grupe sa istim procesom obrade. Ovaj problem raspoređivanja FFS je u literaturi poznat i pod terminima: Hybrid flow shop, Compound flow shop, Multi-processor flow-shop [1], [2].

Matematički je problem FFS opisan u [3]:

Razmatramo problem rasporeda n proizvoda J_j, gde je $j=1,2,...n.$, tako da u proizvodnom sistemu FFS ukupna vremenska dužina rasporeda bude minimalna. FFS se sastoji od $m \geq 2$ proizvodnih faza ili mašinskih centara sa l faza koje imaju $k_l \geq 1$ identičnih mašina sa istom vrstom obrade $P_{l1}, P_{l2},..., P_{lk}$. Za svaki proizvod J_j, vektor $[p_{1j}, p_{2j},...,p_{mj}]^T$ vremena obrade su unapred poznata, gde je $p_{lj} \geq 0$ za svako (l,j). Operacija obrade T_{ij} proizvoda J_j bi mogla da se izvrši na svakoj od k_l mašina.

Proizvodi moraju da posete sve faze obrade u istom redosledu počev od faze l kroz narednih m faza. Mašina može da obradi najviše jedan proizvod u datom trenutku i svaki proizvod može biti obrađivan na samo jednoj mašini u datom trenutku. Pravo preče obrade nije dozvoljeno. Problem raspoređivanja se sastoji u dodeljivanju operacija obrade proizvoda mašinama na svakoj fazi obrade i definisanje redosleda kojim se proizvodi obrađuju na mašinama a sve u cilju minimizacije kriterijumske funkcije. Treba napomenuti da vreme obrade p_{lj} ne zavisi od mašine koja je dodeljena proizvodu J_j na fazi l. Vreme završetka obrade J_j proizvoda u fazi obrade l se obeležava sa C_{jl}.

Matematički model mešovitog celobrojnog programiranja za problem $FFm| k_1, k_2, ...,k_m|C_{max}$ se predstavlja dalje.

U matematičkom modelu, promenljive odlučivanja definišu redosled obrade proizvoda na mašinama u svakoj fazi obrade:

x_{ijkl} = 1, ako se proizvod J_j u redosledu obrađuje odmah posle proizvoda J_i na mašini P_k u fazi obrade l, x_{ijkl} = 0, u suprotnom;

x_{0jkl} = 1, ako je proizvod J_j prvi proizvod u redosledu obrade na mašini P_k u fazi l, x_{0jkl} = 0, u suprotnom;

x_{i0kl} = 1, ako je proizvod J_i poslednji u redosledu obrade na mašini P_k u fazi l, x_{i0kl} = 0, u suprotnom;

i vremenski trenutak završetka obrade proizvoda u svakoj fazi: C_{jl} = vreme završetka obrade proizvoda J_j u fazi l;

$$(\min) C_{max} \qquad (1.1)$$

p.o.

$$\sum_{i=0, i\neq j}^{n} \sum_{k=1}^{k_l} x_{ijkl} = 1, \quad \forall j = 1,..n, l = 1,..m \qquad (1.2)$$

$$\sum_{j=0}^{n} x_{ijkl} \leq 1, \quad \forall i = 0,...,n, k = 1,...,k_l, l = 1,...,m \qquad (1.3)$$

$$\sum_{i=0, i\neq h}^{n} x_{ihkl} - \sum_{j=0, j\neq h}^{n} x_{hjkl} = 0, \quad \forall h = 1,...,n, k = 1,...,k_l, \; l = 1,...,m \qquad (1.4)$$

$$C_{il} + \sum_{k=1}^{k_l} x_{ijkl} \cdot p_{lj} + \left(\sum_{k=1}^{k_l} x_{ijkl} - 1 \right) \cdot B \leq C_{jl}$$
$$\forall i = 1,...,n, j = 1,...,n, l = 1,...,m \qquad (1.5)$$

$$C_{jl-1} + p_{lj} \leq C_{jl}, \quad \forall j = 1,...,n, l = 2,...,m \qquad (1.6)$$

$$C_{jl} \leq C_{max}, \quad \forall j = 1,...,n, l = 1,...,m \qquad (1.7)$$

$$x_{ijkl} \in \{0,1\} \quad \forall i = 1,...,n, j = 1,...,n, \; k = 1,...,k_l, l = 1,...,m \qquad (1.8)$$

$$C_{jl} \geq 0 \quad \forall j = 1,...,n, l = 1,...,m \qquad (1.9)$$

Funkcija cilja C_{max} (1.1) je vreme završetka obrade svih proizvoda. Tu funkciju je potrebno minimizovati. Ograničenja (1.2), (1.3) i (1.4) obezbeđuju da se svaki proizvod obradi tačno jednom na svakoj fazi obrade. Posebno, ograničenje (1.2) garantuje da za svaku fazu obrade l za svaki proizvod J_j postoji jedna mašina tako da se ili J_j obrađuje prvi ili posle nekog drugog proizvoda na toj mašini. Nejednakost (1.3) govori da na svakoj fazi obrade postoji mašina na kojoj proizvod koji se obrađuje ima proizvod koji ga sledi u nizu ili se taj proizvod obrađuje poslednji. Ograničenje (1.4) kaže da na svakoj fazi za svaki proizvod postoji samo jedna mašina koja zadovoljava prethodna dva uslova. Ograničenje (1.5) i (1.6) vode računa o vremenu završetka obrade proizvoda. Nejednačina (1.5) obezbeđuje da vremena završetka C_{il} i C_{jl} proizvoda J_i i J_l rasporede uzastopno na istoj mašini poštujući redosled. B je vrlo veliki broj tj. konstanta koja je veća od sume svih vremena obrade proizvoda. Sa druge strane, nejednačina (1.6) određuje da proizvodi prolaze kroz faze u istom redosledu od faze l do faze m. Ograničenje koje govori da vreme završetka obrade svih proizvoda nije manje od vremena završetka obrade bilo kog proizvoda je izraženo sa nejednačinom (1.7). Poslednja dva ograničenja (1.8) i (1.9) definišu domen upravljačkih promenljivih.

S obzirom da je problem FFS *NP*-tvrd [11], [12], pristupi njihovom rešavanju se mogu podeliti u grupe [13]:

Egzaktne metode – one koje za prihvatljive dimenzije problema garantuju pronalaženje optimalnog rešenja sa aspekta jednog kriterijuma ili Pareto optimalnog rešenja u slučaju više kriterijuma.

Programiranje ograničenja (*Constraint Programming*) – pristup u kome se traži rešenje koje zadovoljava sva ograničenja a koje ne mora biti optimalno [14].

Heurističke metode – metode efikasnog rešavanja zasnovane na nekim zdravo-razumskim logičkim pravilima. Ovakve metode ne garantuju pronalaženje optimalnog rešenja, ali vrlo efikasno mogu pronaći dovoljno dobro rešenje, ili niz rešenja, koja zadovoljavaju postavljena ograničenja. Heuristike se najčešće dele na: konstruktivne heuristike koje generišu samo jedno rešenje i heuristike lokalne pretrage koje generišu niz dopustivih rešenja u svakoj iteraciji.

Složenost algoritama prva dva pristupa je eksponencijalna funkcija dimenzije problema i zato su često neefikasni za rešavanje realnih problema iz prakse. Zato se za rešavanje složenijih modela skoro isključivo koriste jednostavne heurističke metode. Neke od njih biće predstavljene u tekstu koji sledi.

3. HEURISTIČKE METODE I NUMERIČKI EKSPERIMENT

Efikasnost heurističkih metoda biće analizirana na primeru FFS koji se u $\alpha|\beta|\gamma$ notaciji označava kao: $FF5|r_j, d_j, w_j,|\gamma$, pri čemu su konkretni podaci prikazani u Tabeli 1. Radi se o raspoređivanju 15 proizvoda (P1-P15) na pet mašinskih centara (C1-C5). Za svaki proizvod su dati sledeći parametri: p_{ij} – vremensko trajanje operacije obrade j-tog proizvoda na i toj mašini, r_j – vremenski trenutak kada j-ti proizvod postaje raspoloživ za obradu, d_j - vremenski trenutak do koga je neophodno da se j-ti proizvod završi sa obradom i w_j – težinski koeficijent tj. prioritet j-tog proizvoda u odnosu na druge proizvode. Struktura mašinskih centara je sledeća: u prvoj fazi obrade u okviru prvog mašinskog centra postoje dve istorodne mašine, u drugoj i trećoj fazi obrade po tri istorodne mašine, u četvrtoj fazi obrade dve istorodne mašine i u petoj fazi obrade tri istorodne mašine.

Tabela 1. Matrica podataka koji se koriste u analizi problema

				t_{ij}				
	r_j	d_j	w_j	C1	C2	C3	C4	C5
P1	0	25	4	4	2	5	8	6
P2	0	26	1	6	4	7	7	2
P3	0	30	2	4	6	6	4	2
P4	0	30	1	7	7	4	6	5
P5	0	32	2	6	6	4	2	3
P6	3	48	2	6	6	14	2	15
P7	5	40	1	12	6	10	4	3
P8	3	24	4	6	6	4	2	3
P9	0	40	2	6	8	4	12	5
P10	2	36	2	8	6	4	2	8
P11	2	40	4	10	6	4	2	8
P12	0	36	2	8	6	4	2	8
P13	2	80	2	8	16	14	12	18
P14	2	36	1	2	8	6	4	4
P15	2	34	2	2	6	4	6	4

Za rešavanje postavljenog problema koristiće se heuristike poznate pod nazivom pravila prioriteta (*Dispatching Rules*). Ona su posebno pogodna u kompleksnim, dinamičnim i nepredvidivim proizvodno-uslužnim okruženjima, radi generisanja brzih rešenja. Primena ovih pravila pri procesu raspoređivanja, podrazumeva da se na slobodnu mašinu rasporedi proizvod spreman za obradu, koji na najbolji način zadovoljava usvojeno pravilo raspoređivanja.

Postoji veliki broj pravila prioriteta [2], [3], [4]. Ovde će biti analizirana sledeća:

SPT (*Shortest Processing Time*) - Pravilo koje neraspoređenom proizvodu sa najmanjim vremenskim normativom obrade daje prvi prioritet pri raspoređivanju.

WSPT (*Weighted Shortest Processing Time*) - Prvo rasporediti proizvod sa najvećim količnikom težinskog koeficijenta i vremena obrade $max\ (w_j / p_j)$.

LPT (*Longest Processing Time*) - Prvo rasporediti proizvod sa najvećim vremenskim normativom tj. najzahtevniji proizvodi.

EDD (*Earliest Due Date*) - Prvo rasporediti proizvod sa najranijim rokom završetka. Ovo pravilo je poznato i pod nazivom Jackson's rule [15].

MS (*Minimum Slack*) - Prvo rasporediti proizvod sa najmanjim vremenskim zalihama. Vremenska zaliha nekog *j*-tog proizvoda se izračunava preko formule $\max\{d_j - p_j - t, 0\}$;

ERD (*Earliest Release Date*) - Prvo rasporediti proizvod koji najranije postaje raspoloživ za obradu. Ovo pravilo je ustvari pravilo: "prvi došao, prvi uslužen" (*First Come First Served* - FCFS).

CR (*Critical Ratio*) - Prvo rasporediti operaciju *j*-tog proizvoda sa najmanjom vrednošću kritičnog koeficijenta (odnos vremena preostalog do isteka roka za završetak i vremena preostalog za obradu nekog proizvoda):

$$CR = \frac{d_j - t}{\sum_{i=k}^{m} p_{ij}}$$

Proizvodi čiji je kritični koeficijent: (CR<1), su oni proizvodi koji kasne u odnosu na plan; (CR=1) su oni proizvodi koji će se izvršiti na vreme; (CR>1) su oni proizvodi koji se završavaju pre vremena i imaju vremensku rezervu.

ATCS (*Apparent Tardiness Cost with Setups*) – Pravilo raspoređivanja predstavlja kombinaciju više drugih pravila. Prema ovom pravilu raspoređivanja, u vremenskom trenutku *t*, prioritet pri obradi na *i*-toj mašini koja je slobodna, ima proizvod sa najvećom vrednošću funkcije:

$$I_j(t,l) = \frac{w_j}{p_j} e^{\left(\frac{-\max(d_j - p_j - t, 0)}{k_1 \bar{p}}\right)} e^{\left(-\frac{s_{lj}}{k_2 \bar{s}}\right)}$$

U ovoj funkciji oznake su sledeće**:**

t - aktuelni vremenski period;

l - indeks proizvoda koji je upravo završen;

w_j – težinski koeficijent za *j*-ti proizvod;

p_j – vreme obrade *j*-tog proizvoda (za koji se izračunava vrednost funkcije pravila *ATCS*);

d_j – rok za završetak *j*-tog proizvoda;

$\bar{p}$ – prosečno vreme obrade svih proizvoda;

$\bar{s}$ – prosečno vreme pripreme svih proizvoda;

s_{lj} – pripremno vreme potrebno kada proizvod *j* dolazi na obradu posle proizvoda *l*;

k_1 *i* k_2 – parametri za skaliranje funkcije. Njihovu vrednost može definisati donosilac odluke, ali se može koristiti i preporuka data u literaturi [16]:

$$k_1 = 4{,}5 + R, za R \le 0.5$$

$$k_1 = 6 - 2R, za R > 0.5$$

$$k_2 = \frac{\tau}{2\sqrt{\eta}}$$

gde su:

$R = \frac{(d_{\max} - d_{\min})}{C_{\max}}$ - raspon roka završetka (*Due Date Range*), gde je: d_{min} – najmanja vrednost roka završetka od svih proizvoda; d_{max} – najveća vrednost roka završetka od svih proizvoda;

$\tau = 1 - \frac{\bar{d}}{C_{\max}}$ - uzanost roka završetka (*Due Date Tightness*), gde je: $\bar{d}$ – prosečna vrednost roka završetka svih proizvoda;

$\eta = \frac{\bar{s}}{\bar{p}}$ - strogost prosečnog vremena pripreme u odnosu na prosečno vreme obrade poslova (*Setup Time Severity*).

Što se tiče ostalih, složenijih heuristika, u analizi rezultata su korišćene:

Heuristika premeštanja uskih grla (*Shifting Bottleneck heuristics*) [17]. Ova heuristika deli problem rasporeda nekoliko proizvoda na *m* mašina, na *m* problema rasporeda na jednoj mašini. U slučaju FFS to je *s* problema rasporeda na mašinske centre koji imaju $m \ge 1$ mašina u paralelnoj vezi. U svakoj iteraciji postoje mašine na kojima je izvršeno raspoređivanje u prethodnim iteracijama. Raspoređivanje se prvo vrši na nekoj novoj mašini, koja predstavlja usko grlo, prema rešenju tj. redosledu proizvoda dobijenom rešavanjem problema raspoređivanja samo na toj mašini sa ciljem minimizacije određene kriterijumske funkcije.

Usko grlo je ona mašina ili mašinski centar koji je najkritičniji sa aspekta vrednosti te kriterijumske funkcije. Nakon raspoređivanja na mašini koja u datom trenutku predstavlja usko grlo, heuristika pokušava da ponovo revidira već raspoređene proizvode kako bi se dobio bolji raspored. U hipotetičkom primeru koji se analizira, koristi se heuristika premeštanja uskih grla sa različitim kriterijumskim funkcijama za determinaciju uskog grla i raspoređivanje na jednoj mašini (U tabeli 2 Shifting Bottleneck/ $\sum w_j T_j$; C_{max}; T_{max}; $\sum C_j$; $\sum T_j$; $\sum w_j C_j$)

Hibridna metoda rešavanja SB-LS (Shifting Bottleneck & Local Search). Ova heuristika predstavlja kombinaciju heuristike uskih grla koja generiše inicijalni raspored po mašinama, i heuristike lokalne pretrage koja pokušava da pronađe bolje rasporede u okolini prethodnog inicijalnog rasporeda.

Kod *FFS* problema, ako je broj proizvoda koje je potrebno rasporediti znatno veći od broja mašina, ovaj algoritam se pokazuje znatno uspešnijim u odnosu na ostale algoritme [18].

Eksperiment je obavljen da bi se došlo do odgovora koja od navedenih metoda rešavanja problema FFS, generiše dobra rešenja sa aspekta različitih performansi tj. kriterijuma vrednovanja.

Za eksperimentisanje je korišćen je računarski program LEKIN, u kome su implementirane navedene metode [19].

4. ANALIZA REZULTATA

Rezultati primene predstavljeni su u tabeli 2. Rešenja dobijena različitim pravilima prioriteta međusobno su upoređivana prema izabranim performansama tj. kriterijumskim funkcijama. Performanse se odnose na proizvode i označene su na sledeći način:

C_{max} – vreme završetka obrade svih proizvoda, ili vreme kada se poslednji proizvod završi sa obradom, $C_{max}=max_j(C_j)$, gde je C_j vreme kada se j-ti proizvod završi;

T_{max} - maksimalno zakašnjenje; zakašnjenje se definiše kao pozitivna razlika između vremena stvarnog završetka obrade određenog proizvoda i vremenskog roka do koga se očekuje njegov završetak, tj.

$Tj=(Cj-dj)+;$ $T_{\max} = \max_j (T_1,...,T_n)$;

$\sum_{j=1}^{n} T_j$ - ukupno zakašnjenje, zbir kašnjenja svih proizvoda. Ova kriterijumska funkcija može biti predstavljena i sa težinskim koeficijentima za svaki proizvod $\sum_{j=1}^{n} w_j T_j$. Težinski koeficijenti mogu označavati troškove kašnjenja j-tog proizvoda po jedinici vremena ili kvantifikovati važnost svakog proizvoda.

$\sum_{j=1}^{n} C_j$ – zbir vremena završetka obrade svih proizvoda. Ovaj kriterijum može biti pomnožen težinskim koeficijentima svakog da bi se dobila sledeća funkcija $\sum_{j=1}^{n} w_j C_j$.

$\sum_{j=1}^{n} U_j$ - ukupan broj proizvoda koji su u zakašnjenju. Ukoliko j-ti proizvod kasni $C_j>d_j$, onda je $U_j=1$, u suprotnom je $U_j=0$.

Na osnovu rezultata koji su predstavljeni Tabelom 2 nad problemom raspoređivanja $FF5|\ r_j, d_j, w_j,|\gamma$ mogu se izvesti sledeći zaključci:

Heuristika premeštanja uskih grla *(*SB*)* za dimenzije problema (n, m) = (15, 5) pruža odlične rezultate. To se može videti po „boldovanim“ vrednostima u tabeli 2. za kriterijumske funkcije:

$$\left(C_{\max}, T_{\max} \sum C_j, \sum T_j\right)$$

Hibridna heuristika (SB-LS) se pokazala vrlo uspešnom na definisanom problemu generisanjem dobrog rešenja sa aspekta vrednosti kriterijumskih funkcija:

$$\left(\sum U_j, \sum w_j C_j, \sum w_j T\right)$$

Pravila prioriteta su dala raznolike rezultate za zadati problem:

- Pravilo raspoređivanja SPT generiše dobro rešenje za kriterijumske funkcije:

  $\sum U_{j,}$ $\sum C_j$, $\sum T_j$;

- Pravilo raspoređivanja *WSPT* generiše dobro rešenje za kriterijumske funkcije:

  $\sum U_j$, $\sum w_j C_j$;

- Pravilo raspoređivanja *LPT* generiše dobro rešenje za kriterijumsku funkciju C_{max};

- Pravilo raspoređivanja *EDD* generiše dobro rešenje sa aspekta kriterijumskih funkcija:

  T_{max}, $\sum T_j$;

- Pravilo raspoređivanja *MS* generiše dobro rešenje sa aspekta kriterijumske funkcije T_{max};

- Pravilo raspoređivanja *ERD* generiše dobro rešenje za kriterijumske funkcije: $\sum U_j$, $\sum T_j$;

- Pravilo raspoređivanja *CR* generiše dobro rešenje sa aspekta kriterijumske funkcije T_{max};

- Pravilo raspoređivanja *ATCS* generiše dobro rešenje za kriterijumske funkcije: $\sum w_j C_j$, $\sum w_j T_j$

Tabela 2. Rešenje problema FF5| r_j, d_j, w_j,| γ

Metoda	vreme[1]	C_{max}	T_{max}	$\sum U_j$	$\sum C_j$	$\sum T_j$	$\sum w_j C_j$	$\sum w_j T_j$
Shifting Bottleneck / $\sum w_j T_j$	7	92	37	11	729	182	1396	256
Shifting Bottleneck / C_{max}	6	**76**	49	12	885	346	1847	723
Shifting Bottleneck / T_{max}	6	108	**28**	10	761	229	1686	574
Shifting Bottleneck / $\sum C_j$	6	88	31	11	**705**	179	1482	376
Shifting Bottleneck / $\sum T_j$	7	86	31	11	**701**	**161**	1430	300
Shifting Bottleneck / $\sum w_j C_j$	6	90	38	11	728	203	1433	341
Hybrid method (SB-LS)	6	100	35	**9**	717	176	**1385**	**254**
Shortest processing time (SPT)	1	109	35	11	732	200	1533	421
Weighted SPT (WSPT)	1	106	38	11	776	227	1460	328
Longest processing time (LPT)	1	78	53	13	1006	461	2156	1020
Earliest due date (EDD)	1	113	34	13	759	202	1572	412
Minimum slack (MS)	1	109	33	13	800	243	1674	487
Earliest release date (ERD)	1	95	36	11	753	203	1585	439
Critical Ratio (CR)	1	112	35	13	791	234	1637	477
Apparent Tardiness Cost with Setups (2,1)[2]	1	106	37	12	771	220	1451	315

[1] Vreme dolaska do rešenja, izraženo u sekundama, prilikom rada na računaru (Intel Core i3- 2.30GHz, 4 GB RAM)
[2] Vrednost parametara za skaliranje data je u zagradi kao uređeni par (k_1, k_2)

5. ZAKLJUČAK

U ovom radu je predstavljen problem raspoređivanja u fleksibilno-protočnoj proizvodnji i uz pomoć softvera LEKIN na jednom primeru pokazana primenljivost nekoliko jednostavnih heurističkih metoda raspoređivanja. Dobijena rešenja su međusobno upoređena prema različitim performansama. Na osnovu dobijenih rezultata u numeričkom eksperimentu, primenom jednostavnih pravila prioriteta može se zaključiti sledeće:

Za kriterijum C_{max} može se koristiti pravilo *LPT*; Za kriterijum T_{max} mogu se koristiti pravila *MS, ED, CR i SPT;* Za kriterijum $\sum U_j$ pravila prioriteta *SPT, WSPT, ERD*. Za kriterijum $\sum C_j$, pravilo prioriteta *SPT;* Za kriterijum $\sum T_j$ pravila *SPT, EDD, ERD;* Za kriterijum $\sum w_j C_j$, pravila *ATCS* i *WSPT;* Za kriterijum $\sum w_j T_j$, pravilo prioriteta *ATCS*.

Sva prethodno navedena pravila prioriteta imaju ogromnu prednost u odnosu na egzaktne metode rešavanja i složenije heurističke metode sa aspekta vremenske efikasnosti. Iako ne garantuju dobijanje optimalnih rešenja, ona brzo daju relativno dobra rešenja za razliku od egzaktnih metoda koje praktično mogu biti sasvim neprimenljive zbog zahtevanog obima računanja. Posmatranjem rezultata primene pravila prioriteta, u hipotetičkom primeru koji je analiziran u ovom radu, može se zaključiti da ne postoji univerzalno pravilo za raspoređivanje, već se preporučuje da se na problemu raspoređivanja primeni više pravila prioriteta u generisanju rasporeda i izvrši izbor „najboljeg“ rasporeda po željenim performansama donosioca odluke.

Ograničenja ovog istraživanja se odnose na primenu određenog broja metoda rešavanja problema koje su implementirane u računarski program LEKIN.

Definisanje novih heurističkih i egzaktnih metoda za rešavanje problema raspoređivanja ostaje stalni praktični i istraživački izazov.

LITERATURA

[1] Ruiz, R., Vazquez-Rodriguez, J. A., The hybrid flow shop scheduling problem, European Journal of Operational Research, 205, p. 1-18 , 2010.

[2] Leung, J. Y. T. (editor), Handbook of Scheduling: Algorithms, Models, and Performance Analysis, CRC Press, Boca Raton, FL, USA, 2004.

[3] Blazewicz, J., Ecker, K., Pesch, E., Schmidt, G., Weglarz, J., Handbook on Scheduling: From Theory to Applications, International Handbook on Information Systems, Springer, 2007.

[4] Pinedo, M., Planning and Scheduling in manufacturing and service, 2nd edition, Springer, 2009.

[5] Pinedo, M., Scheduling: Theory, Algorithms, and Systems, 4th edition, Springer, 2012.

[6] Heizer, J., Render, B., Operations Management, Prentice Hall, New Jersey, 2011.

[7] Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Optimization and approximation in deterministic sequencing and scheduling: a survey,

Annals of Discrete Mathematics, 5, p. 287-326, 1979.

[8] Lee, G. C., Kim, Y. D., Kim, J. G., Choi, S.H., A dispatching rule-based approach to production scheduling in a printed circuit board manufacturing system, Journal of the Operational Research Society, 54, 10, p. 1038-1049, 2003.

[9] Parthanadee, P., Buddhakulsomsiri, J., Simulation modeling and analysis for production scheduling using real-time dispatching rules: A case study in canned fruit industry, Computers and electronics in agriculture, 70, 1, p. 245-255, 2010.

[10] Joo, B. J., Choi, Y. C., Xirouchakis, P., Dispatching rule-based algorithms for a dynamic flexible flow shop scheduling problem with time-dependent process defect rate and quality feedback, Procedia *CIRP*, 7, p. 163-168, 2013.

[11] Gupta, J. N. D., Two-stage, hybrid flow shop scheduling problem, Journal of the Operational Research Society, 39, 4, p. 359-364, 1988.

[12] Hoogeveen, J. A., Lenstra, J. K., Veltman, B., Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard, European Journal of Operational Research, 89, 1, p. 172-175, 1996.

[13] Krčevinac, S., Čangalović, M., Kovačević-Vujčić, V., Martić, M., Vujošević, M., Operaciona istraživanja, FON, Beograd, 2004.

[14] Vujošević, M., Metode optimizacije u inženjerskom menadžmentu, AINS-FON, Beograd, 2012.

[15] Kellerer, H., Minimizing the maximum lateness, Chapter 10. p.2, in Leung, J.Y.T. (editor), Handbook of Scheduling: Algorithms, Models, and Performance Analysis, CRC Press, Boca Raton, FL, USA, 2004.

[16] Lee, Y. H., Bhaskaran, K., Pinedo, M., A heuristic to minimize the total weighted tardiness with sequence-dependent setups, IIE Transactions, 29, 1, p 45–52, 1997.

[17] Asadathorn, N., Scheduling of Assembly Type of Manufacturing Systems: Algorithms and Systems Development, Ph.D Thesis, Department of Industrial Engineering, New Jersey Institute of Technology, Newark, New Jersey, 1997.

[18] Yang, Y., Kreipl, S., Pinedo, M., Heuristics for Minimizing Total Weighted Tardiness in Flexible Flow Shops, Journal of Scheduling, 3, 2, p 89–108, 2000.

[19] http://community.stern.nyu.edu/om/software/lekin/ (pristupano 21.01.2014).

SUMMARY

ANALYSIS OF DISPATCHING RULES APPLICATION ON SCHEDULING PROBLEM IN FLEXIBLE-FLOW SHOP PRODUCTION

In this paper we analyzed a group of simple heuristic methods, which are used for solving the scheduling problem in manufacturing and services. The analysis was performed on the scheduling problem in a flexible-flow shop production, which is known by the English term – Flexible-Flow Shop (FFS). The task is to determine the schedule of processing multiple products on multiple machines, where all the products have the same sequence of processing and for each process there are multiple machines available. For this FFS problem we present the corresponding mathematical model of mixed integer programming. Among potential methods for solving the set task, we consider simple heuristics because the original scheduling problem is NP-hard and finding the exact optimal solution would require unacceptably long computing time. Heuristic methods are based on priority rules that are performed based on the relations of importance between products and their processing time on individual machines. Heuristic methods are widely used for solving practical problems, which was the motivation for the analysis performed in this paper. The aim of the analysis is to identify those priority rules, from a set of considered, which provide a good solution to a hypothetical scheduling problem example, where the evaluation of solution is performed using different criteria functions. The analysis that is presented in the paper was obtained by using the computer program LEKIN. The main results of the analysis indicated that priority rules give different solutions to the problem of FFS and that each of these solutions is a significantly good result in terms of some of the considered criteria functions.

Key words: *Scheduling, flow shop, flexible-flow shop (FFS), dispatching (priority) rules*