International Journal of Industrial Engineering & Production Management (2017)

September 2017, Volume 28, Number 1
pp 15- 26

http://IJIEPM.iust.ac.ir/

Discrete Particle Swarm Optimization for Job Shop Scheduling Problem with Parallel Machine

Masoud Shayanian & Javad Behnamian *

Masoud Shayanian, *Department of Industrial Engineering, Faculty of Engineering, Bu-Ali Sina University*
Javad Behnamian, *Department of Industrial Engineering, Iran University of Science & Technology*

Keywords

Job shop scheduling, Parallel machines, Metaheuristic, Particle swarm optimization.

ABSTRACT

Since the production scheduling is covered the wide range of manufacturing and services systems, the types of related issues are highly diverse. In advanced manufacturing environments, because the production flexibility property is taking into account as a competitive advantage, in this paper a special kind of flexibility in the job shop scheduling problem is defined in which, for jobs processing, each workstation have multiple parallel machines. Processing speed of each machine can also be different. The objective of this problem is to minimize the maximum completion time (makespan). Due to NP-hardness of problem, we proposed a metaheuristic algorithm. In the proposed approach, due to the structure of the problem and its discrete enviorment, we modified a particle swarm optimization as a new discrete algorithm. Finally to evaluate the performance of the algorithm, the proposed algorithm has been compared with several heuristics existing in the literature.

© 2017 IUST Publication, IJIEPM Vol. 28, No. 1, All Rights Reserved

نشـــریه بین المللـــی مهنـــدسی صنـــایع و مدیـــــریت تولیـــد

شمـــــاره ۱، جلـــد ۲۸، بهار ۱۳۹۶

صفحـــــه ۱۶- ۲۶

نشریه بین المللی مهندسی صنایع و مدیریت تولید دانشگاه علم و صنعت ایران

ISSN: 2008-4870

http://IJIEPM.iust.ac.ir/

ارائه الگوریتم بهینه سازی گروه ذرات گسسته برای زمانبندی مساله تولید کارگاهی انعطاف پذیر با ماشین های موازی

مسعود شایانیان و جواد بهنامیان *

کلمات کلیدی

زمانبندی،
تولید کارگاهی،
ماشینهای موازی،
الگوریتم فراابتکاری،
بهینه سازی گروه ذرات.

چکیده:

زمانبندی[۱] تولید حوزه وسیعی از مسایل اکثر سیستم های تولیدی و خدماتی را تحت پوشش قرار می دهد به همین دلیل ساختار این نوع مسایل بسیار متنوع هستند. در محیط های تولیـدی پیشـرفته، خاصـیت انعطاف پذیری تولید به عنوان یک مزیت رقابتی به حساب می آید. به همین دلیل در این مقاله نوع خاصی از انعطاف پذیری در مساله سیستم زمانبندی تولید کارگاهی[۲] تعریف شده که در آن در هر ایستگاه کاری، ممکن است چندین ماشین موازی[۳] برای پردازش کارها وجـود داشـته باشـد. سـرعت پـردازش هریـک از ماشین ها برای عملیات مختلف نیز می تواند متفاوت از یکدیگر باشد. تابع هدف این مساله، کمینه سازی بازه ساخت[۴] بوده و به علت NP-hard بودن آن، از یک الگوریتم فراابتکاری[۵] جهت حل مسئله استفاده شده است. از اینرو و به دلیل ساختار مساله مورد نظر و گسسته بودن فضای جواب آن، یک روش جدید گسسته بر پایه بهینه سازی گروه ذرات[۶] پیشنهاد شده است. همچنین در انتها نیز برای ارزیابی عملکرد الگوریتم، از چند روش ابتکاری موجود در ادبیات برای بررسی تحلیل و مقایسه استفاده شده است.

۱. مقدمه

زمانبندی تولید حوزه وسیعی از مسایل اکثر سیستم های تولیدی و خدماتی را تحت پوشش قرار می دهد به همین دلیل این مسایل بسیار متنوع هستند. هدف زمانبندی تولید تخصیص منابع محدود در طول زمان برای انجام گروهی از فعالیت های مختلف است. این نوع مسایل دارای ویژگی های گونـاگون، اهـداف و اولویـت‌هـای مختلف هستند، به همین دلیل مدل سازی این گونه مسایل بسیار مشکل است. زمانبندی سیسـتم تولیـد کارگـاهی جـزء یکـی از پیچیده ترین مباحث بهینه سازی ترکیبی بـه شـمارمی رود. در مساله تولید کارگاهی فرض می شود که n کار وجود دارد که باید توسط m ایستگاه کاری مورد پردازش قـرار گیرنـد، کـه در هـر ایستگاه کاری تنها یک ماشین وجود دارد، تمام کارهایی که باید در آن ایستگاه پردازش شوند، توسط آن ماشین مورد پردازش قرار می گیرند. تولید کارگاهی انعطاف پذیر[۷] توسعه مساله تولید کارگاهی است. تعیین برنامه زمانبندی و توالی عملیات در مسایل برنامه ریزی تولید به عنوان یکی از عوامل کلیدی موفقیت در هر سازمان تولیدی نقش مهم و موثری دارد، زیرا زمانبندی صحیح تولید می تواند فواید زیادی را برای ذینفعان امر به دنبال داشته باشد.

در این مقاله براساس تعریفی که از انعطاف پذیری می شود در هر ایستگاه، چند ماشین برای پردازش کارها وجود دارد و هر کار در هر ایستگاه می تواند یکی از ماشین ها را برای پردازش انتخاب کند. همچنین سرعت پردازش هریک از این کارها می تواند با سایر ماشین های موجود در ایستگاه تفاوت داشته باشد.

تابع هدف این نوع مساله، کمینه سازی بازه ساخت است. در محیط های تولیدی مدرن امروز، انعطاف پذیری تولید به عنوان یک ابزار رقابتی کلیدی به شمار می رود، بطوریکه داشتن چنین قابلیت هایی برای این سیستم ها تبدیل به یک مزیت رقابتی شده است و سبب توانایی آنها در پاسخ گویی سریع به تغییرات

تاریخ وصول: ۹۳/۰۸/۰۳

تاریخ تصویب: ۹۴/۰۲/۰۹

مسـعود شـایانیان، دانشـکده مهندسـی، دانشـگاه بـوعلی سـینا، همـدان، Masoud.shayanian@gmail.com

*نویسنده مسئول مقاله: **جواد بهنامیان**، دانشکده مهندسی، دانشگاه بوعلی سینا، همدان، Behnamian@Basu.ac.ir

غیر قابل پیش بینی، به عنوان عامل اصلی فشار رقابتی بازار می شود. این نوع انعطاف پذیری می تواند خصوصیات مختلفی را دارا باشد که مهمترین آن می تواند افزایش ضریب اطمینان سیستم تولیدی تلقی شود. حل مساله زمانبندی سیستم تولید کارگاهی منعطف با ماشین های موازی[۸] شامل دو زیر مساله مسیریابی (تخصیص ماشین به عملیات) و زمانبندی عملیات (تعیین توالی عملیات) می باشد:

۱ – مساله تخصیص: در این مساله نحوه تخصیص ماشین ها به عملیات ها مورد بررسی قرار می گیرد (هر عملیات توسط کدام ماشین پردازش شود).

۲ – مساله تعیین توالی: در این مساله هم توالی عملیات های تخصیص یافته به یک ماشین مورد بررسی قرار می گیرد.

از آنجا که بهینه بودن هر کدام از مسایل بالا بطور جداگانه الزاما بهینه بودن تابع هدف را تضمین نمی کند، حل اینگونه مسایل نیازمند نگرشی است تا بتوان بوسیله آن تابع هدف نهایی را با در نظر گرفتن توالی و تخصیص قابل قبول، بهینه کرد. ساده ترین نگرش، نگرش سلسله مراتبی است بطوریکه دو مساله تخصیص و توالی عملیات را بتوان بصورت جدا از هم فرض کرد. در چنین نگرشی به ازای هر تخصیص، یک توالی بهینه برای این تخصیص بدست آمده و این روند تا رسیدن به جواب بهینه کلی مساله اصلی ادامه می یابد.

در ادامه ابتدا در بخش ۲ مرور ادبیات مورد مطالعه در این تحقیق ارایه می گردد. سپس مدل ریاضی مساله در بخش ۳ ارائه می گردد. در بخش ۴ نیز الگوریتم گروه ذرات معرفی و روش حل مساله به کمک این الگوریتم توضیح داده می شود. در انتها نیز در بخش ۵ به کمک طراحی آزمایش های عددی و تولید مساله تصادفی و تنظیم پارامترهای موجود در الگوریتم، نتایج آزمایشات عددی گزارش شده و نهایتا مقاله با ارائه نتیجه گیری و چند زمینه کاری برای مطالعات آتی خاتمه می یابد.

۲. پیشینه تحقیق

مساله زمانبندی تولیدکارگاهی یکی از مسایل قدیمی حوزه زمانبندی شمرده می شود که در سال ۱۹۷۴ معرفی شد [۱]. در حالیکه پیدایش مساله تولیدکارگاهی انعطاف پذیر به سال ها بعد از آن برمی گردد. در مقالات بسیاری مانند مقاله گنزالیز و ساهنی [۲] و یا در مقاله گری و همکارانش [۳] *NP-hard* بودن مساله زمانبندی تولید کارگاهی به اثبات رسیده است.

انعطاف پذیری به معنای توانایی پاسخ در مواجهه با بروز تغییرات محیطی تعریف می شود. آنچه که امروزه با عنوان تولیدکارگاهی منعطف شناخته می شود برای نخستین بار توسط توسط براکر و اسچلای [۴] در سال ۱۹۹۰ مطرح شد آنها در مقاله خود مفهوم وجود ماشین های عمومی را در فضای تولید مطرح کردند. ماسترو لیلی و گمبرلا [۵] از الگوریتم فراابتکاری جستجوی همسایگی برای بدست آوردن جواب مطلوب مساله تولید کارگاهی انعطاف پذیر بهره بردند این مساله پایه تحقیقات آتی بسیاری از محققان دیگر قرار گرفت.

مدل های زیادی انعطاف پذیری مساله FJS را از جوانب مختلف بررسی نموده اند. اسکریچ و همکارانش [۶] با در نظر گرفتن کمینه سازی دیرکردها به عنوان تابع هدف مساله تولیدکارگاهی انعطاف پذیر را به کمک الگوریتم جستجوی همسایگی مورد بررسی قرار دادند. عبدلمگویید [۷] مساله تولید کارگاهی منعطف را با در نظر گرفتن زمان راه اندازی وابسته به توالی انجام عملیات ها بررسی نمود. همچنین برای حل این مدل از روش جستجوی ممنوع با استفاده از توابع جستجوی همسایگی تصادفی پیشنهادی استفاده شده است. یوسفی یگانه و همکارانش [۸] مساله FJS را با در نظر گرفتن فرض امکان ایجاد وقفه در فعالیت ها بررسی و الگوریتم مبتنی بر جمعیت ممتیک را برای حل آن پیشنهاد نمودند. این فرض زمانی مناسب است که به عنوان نمونه خرابی ماشین آلات می تواند باعث بروز وقفه در انجام فعالیت ها شود.

وجود ماشین های موازی در هر ایستگاه عملیاتی یکی از فرضیاتی است که به عنوان دسته ای از انعطاف پذیری مورد توجه محققان قرار گرفته است. در این زمینه چن و همکارانش [۹] از الگوریتمی ابتکاری را برای حل مسایل صنعتی واقعی کارکارگاهی منعطف با ماشین های موازی استفاده نمود.

بالین [۱۰] عملیات جدیدی را درغالب الگوریتم ژنتیک برای مطابقت با مساله برنامه ریزی ماشین های موازی غیر مشابه با رویکرد حداقل سازی حداکثر زمان ساخت پیشنهاد داد. هم چنین دلفرد و محمدی [۱۱] از دو الگوریتم شبیه سازی تبرید و ژنتیک ترکیبی برای حل مساله تولیدکارگاهی منعطف چندهدفه با ماشین های موازی در محیط پویا با درنظر گرفتن محدودیت تعمیرات برای ماشین ها استفاده کرد، سپس نتایج خود را با نرم افزار لینگو مورد مقایسه قرار دادند. چن و همکارانش[۱۲] برای حل مساله برنامه ریزی کارکارگاهی منعطف با ماشین های موازی الگوریتم های ژنتیک و ژنتیک گروهی را ارایه نمودند. به این منظور آنها برای روش خود دو مدل اساسی تعریف کردند. ابتدا مدلی برای انتخاب ماشین پردازش کننده هر عملیات از میان ماشین های مشابه و موازی موجود و دیگری مدلی برای تعیین توالی انجام کارها برروی یک ماشین. در تحقیقی مشابه بیلجسو و اردم [۱۳] الگوریتم ژنتیک را برای حل مساله برنامه ریزی کارکارگاهی منعطف با ماشین های موازی پیشنهاد دادند. تعریفی که آنان از ماشین های موازی ارایه داده اند امکان پردازش همه کارها توسط

ماشین های مشابه با سرعت های یکسان یا متفاوت بوده است. تحقیقی که برای مقایسه کارایی الگوریتم بهینه سازی توده ذرات پیشنهادی از آن استفاده شده است، توسط ذگردی و همکارانش [۱۴] ارایه شد. آنان در تحقیقشان برای مساله تولید کارگاهی انعطاف پذیر با ماشین های موازی با درنظر گرفتن تابع هدف حداقل سازی بازه ساخت، ضمن مدلسازی مساله و اثبات NP-$hard$ بودن آن، پنج الگوریتم ابتکاری برای حل مساله ارایه نمودند و سپس به مقایسه آنها در ابعاد مختلف مساله پرداختند.

در زمینه حل مسائل تولید کار کارگاهی منعطف روش های متعددی ارائه شده است که اغلب آنها فرا ابتکاری و مبتنی بر جمعیت بوده و کمتر از الگوریتم های تک جمعیتی استفاده شده است.

گوا و همکارانش [۱۵] مساله FJS را با فرض به وجود آمدن کار جدید در پروسه تولید، به کمک الگوریتم کلونی زنبور عسل در دو مرحله بررسی نمودند. مرحله اول برای برنامه ریزی اصلی مساله و مرحله دوم برای برنامه ریزی مساله در زمانی است که کار جدیدی اضافه می شود. برای عملکرد بهتر روش حل از یک الگوریتم گروهی جستجوی محلی مبتنی بر جمعیت استفاده شده است. هم چنین تانگ و همکارانش [۱۶] الگوریتمی ترکیبی را برای حل این مساله ارایه نمودند. آنان از الگوریتم گروه ذرات و ژنتیک برای حل مساله بهره بردند. شهسواری پور و قاسمی شبانکاره [۱۷] مساله تولیدکارگاهی انعطاف پذیر را بصورت چند هدفه در نظر گرفتند. توابع آنها حداقل سازی حداکثر زمان ساخت، بیشترین حجم کاری ماشین بحرانی و مجموع کارکرد همه ماشین آلات بود. آنها از روش حل ترکیبی الگوریتم ژنتیک و شبیه سازی تبرید استفاده نمودند که در فرآیند خود از راه حل های بهینه پارتو استفاده می کند. در اینصورت به جای جواب بهینه فردی، مجموعه ای از جواب های بهینه بدست می آید.

از آنجا که موفقیت روش های فراابتکاری در مساله تولید کارگاهی منعطف وابسته به فرآیند جستجوی محلی است [۱۹-۱۸] لازم از روش هایی برای حل استفاده شود که بتوانند فضای جواب را به خوبی کاوش نمایند؛ الگوریتم های مبتنی بر جمعیت می توانند جواب های مطلوبی در این مساله ارائه نمایند [۷] به همین دلیل در این مقاله از یک الگوریتم جمعیتی استفاده شده است.

همانطور که اشاره شد فرضیات مختلفی در مساله تولید کارگاهی منعطف در نظر گرفته شده اند که هریک می توانند باعث توسعه مدل اولیه خود باشند. در این مقاله فرض می شود که در هر ایستگاه چندین ماشین موازی با سرعت های متفاوت برای پردازش کارها وجود دارند. مسیر انجام عملیات های یک کار ثابت نیست و هر کار می تواند مسیر مختص به خود را برای پردازش عملیات هایش داشته باشد. به عبارتی دیگر مسیر انجام عملیات های یک کار از پیش تعیین شده است ولی می تواند با مسیرهای انجام کارهای دیگر متفاوت باشد. مدل ریاضی مساله تولید کارگاهی منعطف با ماشین های موازی با توضیح فرضیات آن در بخش بعدی تشریح شده است.

۳. مدل سازی ریاضی

در این بخش ابتدا فرضیات حاکم بر مساله توضیح داده می شود و سپس پارامترهای مورد استفاده معرفی خواهند شد. در انتهای این بخش مدل ریاضی ارائه می شود و محدودیت های مساله معرفی می شوند [۱۴].

۳-۱. فرضیات

فرصیاتی که مساله تحت آنها مدلسازی و حل شده، به شرح زیر است.

- تعداد n کار و m ایستگاه عملیاتی
- وجود چند ماشین بطورموازی در هر ایستگاه با سرعت های مختلف
- پردازش یک کار در هر لحظه توسط فقط یک ماشین
- در دسترس بودن همه ی ماشین ها از لحظه صفر
- مجاز نبودن قطع عملیات
- وجود نداشتن خرابی در ماشین ها
- وجود انبار با ظرفیت نامحدود میان ایستگاه ها
- قطعی و معلوم بودن زمان پردازش عملیات ها
- ناچیز بودن زمان آماده سازی عملیات ها و حمل و نقل بین ایستگاه ها

۳-۲. معرفی پارامترها

شمارنده و پارامترهای استفاده شده در مدلسازی در ادامه ذکر شده است.

تعداد کارها: n

j امین کار: J_j

شاخص کار: $j = 1, \dots, n$

تعداد مراحل پردازش: m

شاخص عملیات: $i = 1, \dots, m$

k امین مرحله پردازش: m_k

شاخص مرحله: $k = 1, \dots, m$

r امین ماشین موازی در مرحله k ام: $M_{k,r}$

تعداد ماشین های موازی مرحله k ام: l_k

شاخص ماشین : $r = 1, \dots, l_k$

i امین عملیات از کار j ام: o_{ij}

سرعت ماشین $M_{k,r}$: $S_{k,r}$

زمان پردازش o_{ij}: P_{ij}

یک مقدار ثابت بسیار بزرگ : L

مرحله پردازش عملیات o_{ij} : St_{ij}
سرعت ماشین r ام در مرحله ای که عملیات o_{ij} در آن انجام می شود: $S_{St_{i,j},r}$
علاوه بر این متغیرهای تصمیم مدل به شرح زیر است:
زمان اتمام عملیات o_{ij}: C_{ij}
حداکثر زمان ساخت کارها: C_{max}

اگرکار jام روی ماشین r ام مرحله $St_{i,j}$ انجام شود یک در غیر اینصورت صفرخواهد بود. $X_{i,j,r} = \begin{cases}1\\0\end{cases}$
اگر عملیات o_{ab} زودتر از عملیات o_{ij} انجام شود برابر یک در غیر اینصورت صفر در نظرگرفته می شود: $Y_{a,b,i,j} = \begin{cases}1\\0\end{cases}$

۳-۳. مدل ریاضی مساله

مسئله مورد بررسی بصورت زیر مدل بندی می گردد.

$$\min Z = C_{max} \quad (۱)$$

$$C_{i,j} \geq C_{(i-1),j} + \sum_{r=1}^{l_{St_{i,j}}} (X_{i,j,r} \times \frac{P_{i,j}}{S_{St_{i,j},r}}) \quad \forall i: 1 \leq j \leq n \\ \forall j: 2 \leq i \leq m \quad (۲)$$

$$C_{1,j} \geq \sum_{r=1}^{l_{St_{i,j}}} (X_{i,j,r} \times \frac{P_{1,j}}{S_{St_{1,j},r}}) \quad \forall j: 1 \leq j \leq n \quad (۳)$$

$$\sum_{r=1}^{l_{St_{i,j}}} X_{i,j,r} = 1 \quad \forall i: 1 \leq j \leq n \\ \forall j: 1 \leq i \leq m \quad (۴)$$

$$C_{i,j} \geq C_{a,b} + \frac{P_{i,j}}{S_{St_{i,j},r}} - L \times \left(3 - Y_{a,b,i,j} - X_{i,j,r} - X_{a,b,r}\right) \\ \forall j, b \in \{1 \leq j, b \leq n | j \neq b\} \\ \forall r \in \left\{1, \dots, l_{St_{i,j}}\right\} \quad (۵)$$

$$C_{a,b} \geq C_{i,j} + \frac{P_{a,b}}{S_{St_{a,b},r'}} - L \times \left(2 + Y_{a,b,i,j} - X_{i,j,r} - X_{a,b,r}\right) \\ \forall i, a \in \left\{1 \leq i, a \leq m | St_{i,j} = St_{a,b}\right\} \\ \forall r' \in \left\{1, \dots, l_{St_{a,b}}\right\} \quad (۶)$$

$$C_{max} \geq C_{m,j} \quad \forall j: 1 \leq j \leq n \quad (۷)$$

تابع هدف مساله یعنی کمینه سازی بزرگترین زمان ساخت کارها توسط رابطه (۱) نشان داده شده است. رابطه (۲) باعث تضمین عدم تداخل زمانی برای مجموعه توالی عملیات های یک کار می شود، در واقع این محدودیت باعث می شود که هریک از عملیات های یک کار تنها زمانی شروع شود که عملیات متقدم آن کار تکمیل شده باشد. رابطه (۳) زمان تکمیل اولین عملیات بر روی ماشین تخصیص داده شده برای انجام آن عملیات را مشخص می کند. رابطه (۴) از بابت اینکه هر کار در هر مرحله فقط می بایست بروی یک ماشین انجام شود درون مدل قرارگرفته است. رابطه (۵) و (۶) هم برای اینکه مجموعه عملیاتی که روی یک ماشین خاص در یک مرحله انجام می شوند، تداخل زمانی نداشته باشند در نظر گرفته شده است. اگر عملیات o_{ab} زودتر از عملیات o_{ij} انجام شود در اینصورت در محدودیت (۵) ضریب L صفر شده و این محدودیت فعال می شود علاوه بر این در چنین حالتی در محدودیت (۶) ضریب L یک خواهد شد و این محدودیت زائد می شود. زمانی که عملیات o_{ij} زودتر از عملیات o_{ab} انجام شود محدودیت (۶) فعال شده و محدودیت (۵) زائد خواهد بود. نهایتا رابطه (۷) حداکثر زمان تکمیل کارها را بر اساس زمان تکمیلشان مشخص می کند. پس از معرفی مدل در بخش بعدی الگوریتم پیشنهادی برای حل مساله ارائه می شود.

۴. الگوریتم پیشنهادی

۴-۱. معرفی الگوریتم

الگوریتم گروه ذرات در سال ۱۹۹۵ توسط جیمز کندی و راسل ابرهات [۲۰] معرفی گردید. کار آنان منجر به ایجاد الگوریتمی قوی برای بهینه سازی، به نام الگوریتم گروه ذرات شد. این روش از عملکرد دسته جمعی گروهای حیوانات مانند پرندگان و ماهی ها اقتباس شده است. در این الگوریتم تعدادی از موجودات وجود دارند، که آن ها را ذره[۹] می نامیم ودر کل فضای جستجو پخش شده اند. هر ذره مقدار تابع هدف را در موقعیتی از فضا که در ان قرار گرفته است، محاسبه می کند. سپس با استفاده از ترکیب اطلاعات محل فعلی اش و بهترین محلی که قبلا در آن بوده است و همچنین اطلاعات یک یا چند ذره از بهترین ذرات موجود در جمع، جهتی را برای حرکت انتخاب می کند. پس از انجام حرکت جمعی، یک مرحله از الگوریتم به پایان می رسد. این مراحل چندین بار تکرار می شوند تا آنکه جواب مورد نظر به دست بیاید. روابط (۸) و (۹)،

معادلات برای سرعت و مکان ذره i در لحظه $t+1$ را مشخص می کند:

$$v^i[t+1] = wv^i[t] + c_1r_1\left(x^{i,best}[t] - x^i[t]\right) + c_2r_2\left(x^{g,best}[t] - x^i[t]\right) \quad (۸)$$

$$x^i[t+1] = x^i[t] + v^i[t+1] \quad (۹)$$

روش فوق در محیط بهینه سازی پیوسته پیشنهاد شده و کاربرد آن در فضای گسسته مسئله ما امکان پذیر نبوده و به همین دلیل در این تحقیق، یک روش جدید گسسته بر پایه بهینه سازی گروه ذرات پیشنهاد شده است.

۲-۴. حل مساله به کمک الگوریتم گروه ذرات

برای بهتر مشخص شدن نحوه حل مساله، دو کار با سه مرحله تولیدی در نظر می گیریم:

در مرحله اول دو ماشین مشابه، در مرحله دوم یک ماشین و در مرحله سوم سه ماشین مشابه وجود دارند. ماشین هایی که در مرحله مشابهند دارای سرعت های مختلف هستند. کار اول دارای سه عملیات می باشد که زمان پردازش هر عملیات بروری ماشین های آن مرحله مطابق مقادیر زیر مشخص شده اند.

برای کار اول $j=1$

$$o_{11}: \begin{cases} m_1: 10 \\ m_2: 15 \end{cases} \qquad o_{21}: m_3: 7 \qquad o_{31}: \begin{cases} m_4: 6 \\ m_5: 3 \\ m_6: 9 \end{cases}$$

شکل ۱. اطلاعات مربوط به کار اول مثال عددی

کار دوم دارای دو عملیات می باشد که در مراحل اول و سوم می بایست مورد پردازش قرار گیرند که زمان پردازش هر عملیات بروری ماشین های آن مرحله مطابق مقادیر زیر مشخص شده اند.

برای کار دوم $j=2$

$$o_{12}: \begin{cases} m_1: 3 \\ m_2: 5 \end{cases} \qquad o_{22}: \begin{cases} m_4: 7 \\ m_5: 4 \\ m_6: 9 \end{cases}$$

شکل ۲. اطلاعات مربوط به کار دوم مثال ارایه شده

ماتریس زیر کلیه اطلاعات عملیات های ممکن بر روی ماشین ها را مشخص می کند. نمونه ای تقریبی از این ماتریس در مقاله فتاحی و همکارانش [۲۱] آورده شده است.

1	1	1	1	1	1	2	2	2	2	2
عملیات 1		2	3	3	3	1	1	2	2	2
کارگاه 1		2	3	3	3	1	1	3	3	3
m_1	m_2	m_3	m_4	m_5	m_6	m_1	m_2	m_4	m_5	m_6
ماشین										
10	15	7	6	3	9	3	5	7	4	9
زمان										

شکل ۳. ماتریس نمایش اطلاعات کلی مثال ارایه شده

ماتریس تولید زیر را در شکل ۴ در نظر بگیرید، این ماتریس ورودی در روند حل مساله خواهد بود:

	o_{11}	o_{21}	o_{31}	o_{12}	o_{22}
m_1	10	0	0	3	0
m_2	15	0	0	5	0
m_3	0	7	0	0	0
m_4	0	0	6	0	7
m_5	0	0	3	0	4
m_6	0	0	9	0	9

شکل ۴. ماتریس ورودی اطلاعات در الگوریتم پیشنهادی

تعداد سطرها به تعداد کل ماشین های موجود است و تعداد ستون ها برابر کلیه عملیات های موجود است. زمان پردازش هر عملیات به صورت ستونی برروی ماشین های ممکن در آن مرحله پردازش، عناصر ماتریس خواهد بود.

اعدادی که در این ماتریس صفر می باشند به آن معنی است که این عملیات خاص بر روی آن ماشین قابیت پردازش شدن نخواهد داشت، به همین دلیل در کلیه ماتریس هایی که در روند حل مساله وجود دارد این عناصر صفر در نظر گرفته می شود.

۳-۴. نحوه نمایش جواب

همانطور که اشاره شد، مساله تولید کار کارگاهی منعطف با ماشین های موازی نیازمند تعیین دو شاخص تخصیص هرکار به یک ماشین و تعیین توالی انجام این کارها بر روی ماشین هاست. اما از آنجا که الگوریتم گروه ذرات تنها تخصیص کارها به ماشین ها را مشخص می کند، برای تعیین توالی بهینه به ازای تخصیص صورت گرفته از قاعده ای استفاده می شود که برای هر ماشین، ابتدا کاری زودتر بر روی آن ماشین انجام می شود که مقدار بزرگتری در ماتریس موقعیت هر ذره داشته باشد.

برای مثال اگرفهرست اولیه کارها (تخصیص اولیه برای هر ماشین) توسط الگوریتم مانند الگوی زیر باشد:

$$[23624]$$

در ماتریس فوق عملیات O_{11} برروی ماشین ۲، عملیات O_{21} برروی ماشین ۳، عملیات O_{31} بروی ماشین ۶ و به همین صورت تخصیص ماشین برای انجام هر عملیات مشخص شده است.

دراینصورت برای ماشین ۲ ام که ۲ عملیات را می بایست انجام دهد (عملیات های O_{11} و O_{12}) تعیین اولویت انجام آنها براساس ماتریس ذره مانند الگوی زیر است:

$$[21111]$$

در اینصورت اولویت انجام کار برای ماشین دوم به ترتیب عملیات پنجم و عملیات اول است و ماتریس خروجی مساله که نشان دهنده تخصیص و توالی هرعملیات می باشد بصورت زیر خواهد بود:

$$\begin{bmatrix} 2 & 3 & 6 & 2 & 4 \\ 2 & 1 & 1 & 1 & 1 \end{bmatrix}$$

تعداد ستون های این ماتریس به اندازه ی تعداد کل عملیات هاست که به ترتیب از اولین عملیات کار نخست شروع و به آخرین عملیات کار پایانی ختم می شود. عناصر موجود در سطر اول، نشان دهنده شماره ماشین تخصیص یافته برای انجام آن عملیات است. همچنین عناصر سطر دوم، توالی انجام هرعملیات را بر روی ماشین های اختصاص یافته سطر اول نشان می دهد.

۳-۴. روند حل مثال عددی

در ابتدا به ازای هر ذره یک ماتریس با اعداد تصادفی، به ازای عناصری که در ماتریس ورودی بزرگتر از صفر هستند تولید می شود. برای هر عملیات بیشترین مقدار آن ستون به ماشین مربوط به آن تخصیص پیدا می کند و به این ترتیب تخصیص کارها به ماشین ها در مرحله نخست انجام می شود. این روند برای ذره اول بصورت زیر خواهد بود. تولید ماتریس تصادفی برای ذره اول در تکرار اول بصورت زیر خواهد بود.

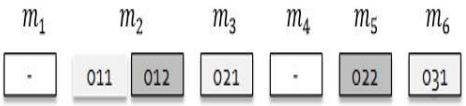
$$X^1[1] = \begin{matrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \\ m_6 \end{matrix} \begin{bmatrix} 0.05 & 0 & 0 & 0.35 & 0 \\ 0.65 & 0 & 0 & 0.54 & 0 \\ 0 & 0.18 & 0 & 0 & 0 \\ 0 & 0 & 0.73 & 0 & 0.29 \\ 0 & 0 & 0.72 & 0 & 0.35 \\ 0 & 0 & 0.87 & 0 & 0.19 \end{bmatrix}$$

$$\begin{matrix} O_{11} & O_{21} & O_{31} & O_{12} & O_{22} \\ [2 & 3 & 6 & 2 & 5] \end{matrix}$$

شکل ۵. نحوه تخصیص هرعملیات به هریک از ماشین ها

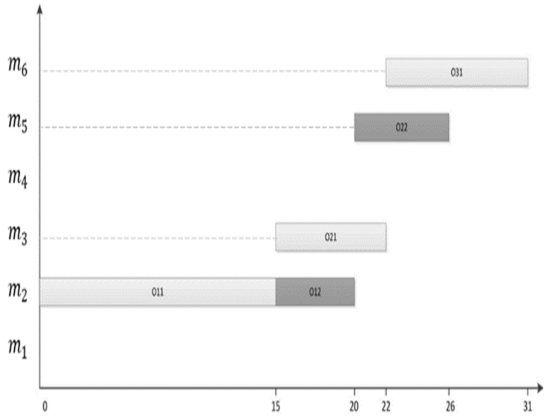
ماتریس تخصیص درشکل ۵ به این معنی خواهد بود که عملیات اول از کار اول بر روی ماشین دو، عملیات دوم از کار اول بر روی ماشین سه، عملیات سوم از کار اول بر روی ماشین شش، عملیات اول از کار دوم بر روی ماشین دوم و عملیات دوم از کار دوم بر روی ماشین پنجم انجام خواهد شد. اما از آنجا که چندین عملیات به یک ماشین تخصیص داده شده است، هر ماشین برای انجام این عملیات ها نیازمند یک اولویت بندی و ترتیب انجام این عملیات ها خواهد بود. اولویت بندی انجام عملیات ها برای یک ماشین به این صورت خواهد بود که کاری که بیشترین مقدار را در ماتریس ذره داراست در اولویت ۱ برای انجام و دیگر عملیات ها به ترتیب نزولی مقادیرشان در ماتریس ذره اولویت بندی می شوند. در اینصورت ماتریس تخصیص-توالی بصورت زیر خواهد بود:

$$\begin{cases} m_1: - \\ m_2: O_{11}, O_{12} \\ m_3: O_{21} \\ m_4: - \\ m_5: O_{22} \\ m_6: O_{31} \end{cases} \quad \begin{matrix} [23625] \\ \& \\ [11121] \end{matrix} \quad \Rightarrow \quad \begin{bmatrix} 2 & 3 & 6 & 2 & 5 \\ 1 & 1 & 1 & 2 & 1 \end{bmatrix}$$



شکل ۶. نحوه تخصیص عملیات ها به ماشین ها و اولویت بندی آنها برروی هر ماشین

و تعیین *Makespan* که مطابق شکل شماره ۷ است.



شکل ۷. نمودار گانت حداکثر بازه ساخت مثال ارایه شده

به این ترتیب برای ذره اول مقادیر پارامترهای الگوریتم گروه ذرات در تکرار اول بدست می آید و این مقادیر مبنای محاسبات آتی قرار خواهند گرفت تازمانی که شرایط معیار توقف الگوریتم ایجاد شود.

۵. طراحی آزمایش های عددی

۵-۱. روش مورد مقایسه

از آنجا که در ادبیات مورد مطالعه، تحقیقات مشابه ای انجام نشده است، به منظور ارزیابی کارایی الگوریتم پیشنهادی به دلیل تعریف مشابه مساله موردنظر در مقاله ذکردی و همکارانش از سه الگوریتم ابتکاری H_1, H_2 و H_3 استفاده شده است. معیار مقایسه میانگین جواب ها در ده تکرار هر مساله و بدترین جواب یافت شده و بهترین جواب بدست آمده با جواب ناشی از الگوریتم های ابتکاری می باشد.

۵-۲. تولید مساله تصادفی

برای تولید مساله تصادفی، پنج متغیر مورد بحث مساله تعریف شده با عنوان های تعداد کارها، تعداد ماشین ها، تعداد مراحل، زمان های پردازش و سرعت پردازش ماشین ها شناسایی شده اند. برای تعیین مقادیر برخی از عوامل، از مقاله دسوکی و همکارانش [۲۲] کمک گرفته شده است. این مقادیر درجدول ۱ نشان داده شده اند.

جدول ۱. مقادیر مختلف سطوح عوامل شناسایی شده در مساله

عوامل	سطوح عوامل
تعداد کارها	۲۰ – ۲۵ – ۳۰
تعداد ماشین های هرمرحله	U(1,5)
تعداد مراحل	۲ – ۴ – ۸
زمان های پردازش	$U(1,40 \times \sum_{r=1}^{l_k} S_{k,r})$
سرعت پردازش ماشین ها	U(1,3)

۵-۳. نحوه ایجاد جواب اولیه

در ابتدا به ازای هر ذره یک ماتریس با اعداد تصادفی، به ازای عناصری که در ماتریس ورودی بزرگتر از صفر هستند تولید می شود. درواقع به ازای هر عملیاتی که قابلیت پردازش روی یک ماشین را دارا باشد، درمحل درایه مربوط به آن در ماتریس تولید، عددی تصادفی دربازه (۰،۱) ایجاد می شود. برای هر عملیات بیشترین مقدار آن ستون به ماشین مر بوط به آن تخصیص پیدا می کند و به این ترتیب تخصیص کارها به ماشین ها در مرحله نخست انجام می شود.

۵-۴. محیط انجام آزمایش ها

به منظور انجام مقایسه، الگوریتم ارایه شده به همراه الگوریتم های ابتکاری، در محیط نرم افزاری *Matlab* با رایانه ای شخصی با مشخصات رم $4G$ و پردازنده $2.5GHz$ انجام شده است.

۵-۵. تنظیم پارامترها

مقادیر مختلف پارامترهای موجود، برکیفیت جواب های حاصل از الگوریتم گروه ذرات تاثیر چشمگیری دارد. پارامتر مهمی که درتوانایی بهینه سازی الگوریتم گروه ذرات موثر است، ضریب اینرسی W می باشد که بر روی همگرایی الگوریتم تاثیر مستقیم دارد. به عبارتی تاثیر سرعت های گذشته را بر سرعت های زمان حال کنترل می نماید. یک مقدار مناسب برای W باعث ایجاد تعادل بین جستجوهای محلی و سراسری می شود و در اغلب اوقات باعث کاهش تعداد تکرارهای لازم، برای همگرایی به یک جواب مناسب می شود. ژیا و وو [۲۳] مقدار رابطه (۱۰) را برای ضریب اینرسی پیشنهاد دادند.

$$W = W_{max} - \frac{W_{max} - W_{min}}{iter_{max}} * iter \qquad (۱۰)$$

W_{max} مقدار ابتدایی ضریب اینرسی
W_{min} مقدار پایانی ضریب اینرسی
$iter_{max}$ حداکثر تعداد تکرارهای الگوریتم
$iter$ شماره تکرار در جریان

از طریق این رابطه، مقدار W در مراحل ابتدایی یک مقدار بزرگ در نظر گرفته می شود تا یک جستجوی سراسری و کامل از فضای جستجو صورت گیرد. سپس در طی مراحل اجرای الگوریتم، مقدار W به تدریج کاهش داده می شود تا الگوریتم به مرز همگرایی نزدیک شود.

مقادیر $W_{max} = 1.2 , W_{min} = 0.4 , iter_{max} = 500$ برای هریک از پارامترها در نظر گرفته شده است. هم چنین از تعداد ۱۰۰ ذره برای جستجوی فضای جواب بهره برده شده است.

C_1, C_2 ضرایب یادگیری برای هر ذره هستند، بطوریکه C_1 ضریب یادگیری مربوط به تجارب شخصی هر ذره و C_2 ضریب یادگیری مربوط به تجارب کل جمع می باشد. ابرهارت و شی [۲۴] معتقدند، مقدار کم برای این پارامترها به ذره اجازه می دهد تا در فضای جواب قبل از برگشت به مقادیر بهینه هر ذره، جستجوی بیشتری انجام دهد اما مقدار بیشتر این پارامتر باعث تمایل به سمت مقادیر گذشته می شود. در نتیجه مقادیر هریک از این پارامترها برابر با ۱،۴۹ در نظر گرفته شده است.

۵-۶. شرط توقف

معیاری که برای توقف اجرای الگوریتم گروه ذرات درنظر گرفته شده است، محدودیت تعداد تکرار[۱۰] است. این حداکثر تعداد

تکرار برای همه حالت‌های مساله یکسان در نظر گرفته شده است [۲۵].

۶. نتایج عددی

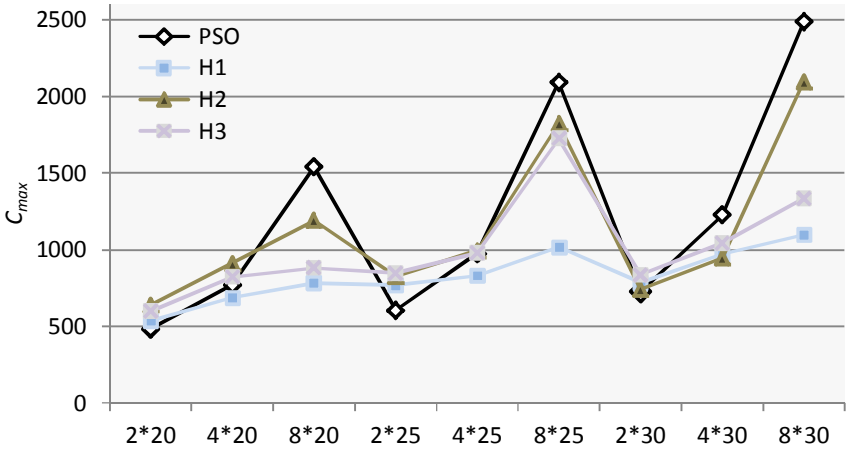
در این قسمت، مقایسه کارایی الگوریتم گروه ذرات نوشته شده با الگوریتم های ابتکاری H_3 , H_2, H_1 موجود در ادبیات مورد مطالعه [۱۴] در جدول ۲ ارایه شده است. برای هر بعد از مساله، الگوریتم گروه ذرات با مشخصات یاد شده ده با اجرا و نتایج آن ثبت شده است. شاخص های انتخاب شده برای مقایسه، بهترین جواب یافت شده، بدترین جواب بدست آمده و میانگین جواب های این ده تکرار می باشند. ذکر این نکته قابل توجه است که از تنظیم پارامترهای الگوریتم گروه ذرات بنا به ابعاد مساله در حالت های مختلف، جواب های به مراتب بهتری بدست آمده است، اما در این شرایط زمان اجرای الگوریتم هم نیز دچار تغییر خواهد شد. به همین دلیل برای حل مساله از یک ساختار ثابت از مشخصات الگوریتم مانند تعداد تکرارها و تعداد ذرات، استفاده شده است.

شکل ۸ نمودار رسم شده بهترین جواب بدست آمده از الگوریتم گروه ذرات و سه الگوریتم ابتکاری H_3 , H_2 , H_1 مقاله ذگردی را برای حالت های مختلف (تعداد کار* تعداد مرحله) نمایش می دهد

جدول ۲. مقایسه الگوریتم های مختلف

تعدادکار		۲۰			۲۵			۳۰		
تعداد مراحل		۲	۴	۸	۲	۴	۸	۲	۴	۸
الگوریتم پیشنهادی توده ای ذرات	کمترین C_{max}	۴۸۳	۷۷۰,۱	۱۵۴۰	۶۰۴,۱	۹۷۴,۳	۲۰۹۲	۷۲۶	۱۲۲۶,۵	۲۴۸۶,۷
	بیشترین C_{max}	۵۳۷	۹۰۲	۱۶۷۳	۶۴۱	۱۲۳۶	۲۴۰۸	۸۱۳,۵	۱۴۸۶,۵	۲۹۹۲
	میانگین جوابها	۵۰۷,۱	۸۲۸,۱	۱۶۰۱,۹	۶۲۴	۱۰۸۱,۵	۲۲۳۱	۷۷۶,۲	۱۳۷۱,۱	۲۷۰۹,۳
الگوریتم ابتکاری H_1		۵۳۷	۶۸۸	۷۸۴	۷۷۰	۸۳۰,۶	۱۰۱۷,۸	۷۸۴,۵	۹۷۰,۷	۱۰۹۷,۳
الگوریتم ابتکاری H_2		۶۳۸,۳	۹۱۱,۸	۱۱۹۱	۸۲۴	۹۹۳,۶	۱۸۲۴,۲	۷۴۱,۵	۹۴۷,۳	۲۰۹۵,۲
الگوریتم ابتکاری H_3		۵۹۸,۷	۸۲۲	۸۸۱,۳	۸۴۸	۹۷۹,۶	۱۷۲۵	۸۳۵,۵	۱۰۴۵,۲	۱۳۳۴,۷

همانطور که از جدول شماره ۲ و همچنین نمودار رسم شده شکل ۸ مشخص است، کارایی هرکدام از الگوریتم ها بصورت مطلق بهتر از سایرین نمی باشد. زمانی که تعداد ایستگاه ها و مراحل کاری کم در نظر گرفته می شود، الگوریتم پیشنهادی گروه ذرات نتایج به مراتب بهتری از سه الگوریتم دیگر ارایه می دهد. اما زمانی که تعداد مراحل کاری افزایش پیدا می کند، از کارایی الگوریتم کاسته می شود. دلیل این اختلاف عملکرد، به این علت است که زمانی که تعداد مراحل مساله تولیدکارگاهی منعطف با ماشین های موازی کم باشد، الگوریتم پیشنهادی به راحتی در فضای جواب جستجو و جواب مطلوب را پیدا خواهدکرد. اما زمانی که تعداد مراحل مساله افزایش می یابد فضای جواب مساله افزایش پیدا خواهد کرد و الگوریتم گروه ذرات برای بدست آوردن جواب مطلوب، نیازمند تعداد ذرات و تعداد تکرارهای بیشتر و در حالت کلی تنظیم پارامترهای خود خواهد بود. کیفیت جواب ها ارایه شده در الگوریتم بیش از سایر موارد مانند تعداد کارها، تحت تاثیر تعداد مراحل مساله می باشد، کما اینکه در هر سه حالت ۲۰ ، ۲۵ و ۳۰ کار جواب های الگوریتم پیشنهادی زمانی که تعداد مرحله برابر با ۲ در نظر گرفته شده است، مطلوب تر بوده است. علاوه بر این به علت ماهیت تصادفی بودن الگوریتم توده ای ذرات لزوما مطلوب بودن جواب آن برای یک حالت باعث تضمین مطلوبیت جواب در حالت های دیگر مساله نخواهد بود.



شکل ۸. نمودارنمایش عملکرد الگوریتم ها در ابعاد مختلف مساله

۷. نتیجه گیری و پیشنهاد تحقیقات آتی

در این مقاله مساله زمانبندی سیستم تولید کارگاهی انعطاف پذیر با ماشین های موازی مورد بررسی قرار گرفت. برای حل مساله از الگوریتم گروه ذرات استفاده گردید. ساختار الگوریتم برای حل مساله مشخص و نحوه چگونگی تطبیق الگوریتم گروه ذرات با مساله مورد نظر نشان داده شد. به دلیل جدید بودن ادبیات مورد مطالعه از تولید مساله تصادفی با استفاده از راهکارهای مشخص بهره برده شد. به منظور نشان دادن کارایی الگوریتم پیشنهادی از سه روش ابتکاری ارایه شده در ادبیات استفاده شد. هر چهار روش در محیط نرم افزاری یکسان کدنویسی و نتایج آن مشخص گردید. الگوریتم پیشنهادی در محیط های تولیدی با تعداد مراحل کم نتایج مطلوبتری در مقایسه با الگوریتم های ابتکاری موجود در ادبیات ارایه می دهد. اما با افزایش تعداد مراحل، از کارایی الگوریتم کاسته می شود. این موضوع در حالتی است که پارامترهای الگوریتم برای تمامی حالت ها ثابت در نظر گرفته شده است. اما زمانی که پارامترهای الگوریتم با توجه به شرایط مساله دچار تغییر می شوند عملکرد الگوریتم به مراتب از حالت قبل مطلوبتر خواهد شد. الگوریتم پیشنهاد شده در این ارایه با افزایش ابعاد مساله الگوریتم چندان عملکرد مناسبی نمی تواند داشته باشد، در واقع زمانی که تعداد کارهای تخصیص یافته به یک ماشین افزایش می یابد کارایی الگوریتم کاهش می یابد و این بدان علت است که الگوریتم گروه ذرات تنها برای تخصیص کارها به ماشین مورد استفاده قرار گرفته است و برای تعیین توالی کارهای تخصیص یافته از الگویی تکاملی استفاده نشده است و فرآیند استفاده شده در الگوریتم حالت تصادفی داشته و در اکثر موارد نمی تواند در ابعاد بزرگ مساله جواب مورد قبول ارایه دهد. بنابراین برای نتایج بهتر برای قسمت دوم مساله یعنی تعیین توالی بهینه عملیات اختصاص یافته به یک ماشین و نهایتا در کل مساله، می بایست از الگوریتم های تکاملی و یا حداقل سازوکاری تعریف شده و مشخص مانند رویه های ابتکاری استفاده شود تا با هر تخصیص بهینه یک توالی بهینه برای انجام عملیات ها بر روی ماشین ها بدست آید و نهایتا جواب قابل قبول تری دریافت گردد. استفاده از الگوریتم های ترکیبی در حالت های مختلف می تواند سازوکار مناسبی برای حل مساله در تحقیقات آتی باشد.

پی نوشت

1. Scheduling
2. Jobshop
3. Parallel machine
4. Makespan (C_{max})
5. Metaheuristic
6. Particle swarm optimization (PSO)
7. Flexible job shop
8. Flexible job shop with parallel machine
9. Particle
10. Max-generation

منابع

[1] Brucker K.R., Introduction to sequencing and scheduling, John Wiley, New York, (1974).

[2] Gonzaleaz T., Sahni S., "Flow shop and job shop schedule: complexity and approximation", Opration research, (1978), Vol. 26, pp. 36-52.

[3] Garey M.R., Johnson D.S., sethi R., "The complexity of flow shop and job shop

scheduling", Mathematical of Opration research, (1976), Vol. 1, pp. 117-129.

[4] Brucker P., Schlie R., "Job shop scheduling with Multi-purpose machines", Computing, (1990), Vol. 45, pp. 369-375.

[5] Mastrolilli M., Gambardella L. M., "Effective neighbourhood functions forthe flexible job shop problem", Journal of Scheduling, (2000), Vol. 3, pp. 3–20.

[6] Scrich C.A., Armentano V.A., Laguna M., "Tardiness Minimization in a Flexible Job Shop: A Tabo Search Approach", Journal of Intelligent Manufacturing, (2004), Vol. 15, pp. 103-115.

[7] Abdelmaguid T.F., "A neighborhood search function for flexible job shop scheduling with separable sequence-dependent setup times", Applied Mathematics and Computation, (2015), Vol. 260, pp. 188–203.

[8] Yousefi Yegane B., Khanlarzade N., Rahimi Fard A., "Memetic Algorithm for Flexible Job Shop Scheduling with Preemption", International Journal of Industrial Engineering & Production Management, (2012), Vol. 22, No. 4, , pp. 331-340.

[9] Chen J. C., Chen K. H., Wu J. J., Chen C. W., "A study of flexible job shopscheduling problem with parallel machine and reentrant process", International Journal of Advanced Manufacturing Technology, (2008), Vol. 39, pp. 344–354.

[10] Balin S. "Non-identical parallel machine scheduling using genetic algorithm", Expert Systems with Applications, (2011), Vol. 38, No. 6, pp. 6814–6821.

[11] MajaziDalfard V., Mohammadi Gh., "Two meta-heuristic algorithms for solving multi-objective flexible job-shop scheduling with parallel machine andmaintenance constraints", Computers and Mathematics with Applications, (2012), Vol. 64, pp. 2111–2117.

[12] Chen J. C., Wu C. C., Wen C. C., Chen K. H., "Flexible job shop scheduling with parallel machines using Genetic Algorithm and Grouping Genetic Algorithm", Expert Systems with Applications, (2012), Vol. 39, pp. 10016–10021.

[13] Bilgesu A., Erdem K., "A guide for genetic algorithm based on parallel machine scheduling and flexible job-shop scheduling", Procedia - Social and Behavioral Sciences, (2012), Vol. 62, pp. 817 – 823.

[14] Zegordi S.H., Beheshtinia M.A., Jafari Hajagha N., "Minimizing makespan in job shop scheduling problem with parallel machines", International Journal of Industrial Engineering & Production Management, (2009), Vol. 20, No. 2, pp. 11-21.

[15] Gao K. Z., Suganthan P. N., Chua T. J., Chong C. S., Cai T. X., Pan Q. K., "A two-stage artificial bee colony algorithm scheduling flexible job shop scheduling problem with new job insertion", Expert Systems with Applications, (2015), Vol. 42, pp. 7652–7663.

[16] Tang J., Zhang G., Lin B., Zhang B., "A Hybrid Algorithm for Flexible Job-shop Scheduling Problem", Procedia Engineering, (2011), Vol. 15, pp. 3678 – 3683.

[17] Shahsavari-Pour N., Ghasemishabankareh B., "A novel hybrid meta-heuristic algorithm for solving multi objective flexible job shop scheduling", Journal of Manufacturing Systems, (2013), Vol. 32, pp. 771– 780.

[18] Mastrolilli M., Gambardella L., "Effective neighborhood functions for the flexible job shop problem", Scheduling, (2000), Vol. 3, No. 1, pp. 3-20.

[19] Mesghouni K., Hammadi S., Borne P., "Evolution programs for job shop scheduling", IEEE International Conference on Systems, Man, Cybernetics, (1997), pp.720–724.

[20] Kennedy J., Eberhart R., "Particle swarm optimization", In Proceedings of the 1995 IEEE international conference on neural network, (1995), Vol. 4, No. 4, pp. 1942–1948.

[21] Fattahi P., Saidi M., Jolai F., "Mathematical modeling and heuristic approaches to flexible job scheduling problems", Journal of intelligent Manufacturing, (2007), Vol. 18, pp. 331-342.

[22] Dessouky M.M., Dessouky M.I., Verma S.K., "Flowshop Scheduling with Identical Jobs and Uniform Parallel Machines", European Journal of Operational Research, (1997), Vol. 109, pp. 620-631.

[23] Xia W.J., Wu Z., "An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems", Computers & Industrial Engineering, (2005), Vol. 48, No. 2, pp. 409–425.

[24] Eberhart R., Shi Y., "Particle swarm optimization: Developments, applications and resources", Proceedings of the 2001 IEEE international conference on evolutionary computation, (2001), pp. 81-86.

[25] Fasihi M, Jolai F, Tavakkoli-Moghaddam R., "Minimizing the maximum tardiness on an m-machine reentrant permutation flowshop problem", International Journal of Industrial Engineering & Production Management, (2011), Vol. 21, No. 4, pp. 179-190.

.