

ارزیابی الگوریتم های زمانبندی تولید کارگاهی انعطاف پذیر و مقایسه آنها با الگوریتم ژنتیک دوبخشی

محمدعلی بهشتی نیا^{۱*} و نیلوفر قاضی وکیلی^۲

اطلاعات مقاله	چکیده
دریافت مقاله:	در این مقاله مساله زمانبندی تولید کارگاهی انعطاف پذیر مورد بررسی قرار می گیرد. این مساله تعمیم مساله زمانبندی تولید کارگاهی می باشد. اهداف مساله کمینه کردن حداکثر زمان تکمیل آخرین سفارش (C_{max}) و حداکثر بارکاری ماشینها (W_{max}) و مجموع بار کاری ماشینها (W_T) در نظر گرفته شده اند. این مساله جز مسائل NP-hard می باشد، بنابراین بدست آوردن جواب بهینه به ازای هر نمونه مساله با ابعاد دلخواه در زمان چند جمله ای امکان پذیر نیست. به همین منظور یک الگوریتم ژنتیک پیشنهادی به نام الگوریتم ژنتیک دو بخشی برای حل مساله ارائه شده است. برای بررسی کارایی الگوریتم پیشنهادی از دو مجموعه داده محک استفاده شده است و با الگوریتم های موجود در ادبیات موضوع مقایساتی صورت پذیرفته است. نتایج محاسباتی نشان از کارایی الگوریتم ژنتیک دو بخشی برای حل مساله زمانبندی تولید کارگاهی انعطاف پذیر دارد.
پذیرش مقاله:	
واژگان کلیدی: زمانبندی، تولید کارگاهی انعطاف پذیر، الگوریتم ژنتیک، مقداردهی اولیه، نمایش کروموزم.	

۱- مقدمه

شامل n_1 عملیات است که باید توسط ماشینهایی مشخص پردازش شوند. مسیر پردازش همه سفارشات مشخص بوده و برای سفارشات مختلف می تواند متفاوت باشد. هدف پیدا کردن توالی مناسب پردازش عملیات روی ماشینها بمنظور بهینه کردن یک یا چند تابع هدف می باشد.

در مساله زمانبندی تولید کارگاهی انعطاف پذیر (FJSP)^۳ فرض می شود هر عملیات اجازه دارد روی مجموعه ای از ماشینهای در دسترس پردازش شود. زمانبندی تولید کارگاهی انعطاف پذیر بسیار مشکل تر از زمانبندی تولید کارگاهی است زیرا شامل مساله اختصاص عملیات به ماشینها نیز می شود [۴].

در این مقاله به بررسی مساله زمانبندی تولید کارگاهی انعطاف پذیر می پردازیم. مساله زمانبندی تولید کارگاهی از مسائل NP-hard محسوب می شود. بنابراین زمانبندی تولید کارگاهی انعطاف پذیر که بسط یافته زمانبندی تولید

زمانبندی اختصاص منابع برای اجرای مجموعه ای از کارها در یک دوره زمانی مشخص می باشد. یافتن زمانبندی بهینه بسته به محیط کارگاه، محدودیت های فرآیند و شاخصهای عملکرد می تواند بسیار آسان و یا بسیار مشکل باشد [۱]. یکی از مشکل ترین مسائل در این حوزه مسائل زمانبندی تولید کارگاهی^۲ می باشد. تولید کارگاهی یکی از موضوعات بارز در برنامه ریزی تولید می باشد. زمانبندی تولید کارگاهی از اهمیت بسزایی برخوردار است زیرا نقشه فرآیند و قابلیت های فرآیند برای بیشتر صنایع را تعیین می کند [۲]. در مساله زمانبندی تولید کارگاهی مجموعه ای از n سفارش (کار) باید توسط m مرحله مختلف پردازش شوند. هر مرحله شامل یک ماشین بوده و هر سفارش i

* پست الکترونیک نویسنده مسئول: beheshtinia@profs.semnan.ac.ir

۱. استادیار، دانشکده مهندسی صنایع، دانشگاه سمنان

۲. دانشجوی کارشناسی ارشد MBA- تولید و عملیات، دانشگاه سمنان

² Job Shop Scheduling

³ Flexible Job Shop Scheduling Problem

جزئی و مساله زمانبندی تولید کارگاهی انعطاف پذیر کلی تقسیم نموده اند. اگر مجموعه ماشین در دسترس برای عملیات زیر مجموعه ای از مجموعه ماشینهای مساله باشد به آن مساله زمانبندی تولید کارگاهی انعطاف پذیر جزئی می گویند.

اگر تمام عملیات بتوانند از تمام ماشینهای مساله استفاده کنند به آن مساله زمانبندی تولید کارگاهی انعطاف پذیر کلی می گویند. مساله زمانبندی تولید کارگاهی انعطاف-پذیر جزئی سخت تر از مساله زمانبندی تولید کارگاهی انعطاف پذیر کل می باشد.

روشهای حل مختلفی برای حل مساله زمانبندی تولید کارگاهی انعطاف پذیر پیشنهاد شده است. این روشها را می توان به دو دسته اصلی تقسیم نمود: رویکرد سلسله مراتبی^۴ و رویکرد یکپارچه^۵. در رویکرد سلسله مراتبی، تخصیص عملیات به ماشینها و توالی عملیات روی ماشینها به صورت جداگانه انجام می شود. بعبارت دیگر ابتدا مساله تخصیص عملیات به ماشینها حل می شود. پس از تخصیص عملیات، توالی عملیات تخصیص داده شده به هر ماشین تعیین می گردد. رویکرد سلسله مراتبی براساس ایده تجزیه می باشد که پیچیدگی را کاهش می دهد. این رویکرد در تحقیقات مختلفی مورد استفاده قرار گرفته است [۷-۹]. رویکردهای یکپارچه تخصیص عملیات به ماشینها و تعیین توالی عملیات روی ماشینها را به طور همزمان در نظر گرفته و معمولا منجر به نتایج بهتری می شوند.

برندیمارت [۷] رویکردی سلسله مراتبی را برای حل مساله بر اساس تجزیه مساله ارائه داده است. برندیمارت مساله را به زیر مساله هایی تقسیم نموده و با استفاده از الگوریتم جستجوی ممنوع مساله را حل نموده است.

کاسم و همکاران [۸] رویکردی محلی را برای تخصیص منابع ارائه داده است و یک روش تکاملی چند هدفه را به منظور حل مساله زمانبندی تولید کارگاهی انعطاف پذیر پیشنهاد داده است.

اکسیا و وو [۹] بر رویکرد ترکیبی برای مساله زمانبندی تولید کارگاهی انعطاف پذیر چند هدفه تمرکز نموده اند. رویکرد آنها از بهینه سازی ازدحام ذرات^۶ برای تخصیص

کارگاهی انعطاف پذیر می باشد نیز از مسائل NP-hard می باشد. بنابراین بدست آوردن جواب بهینه برای مسائل با سایز متوسط و بزرگ در زمان معقول امکان پذیر نمی باشد [۳] و نیاز به استفاده از الگوریتمهای ابتکاری یا فراابتکاری برای حل مساله می باشد. به همین منظور یک الگوریتم ژنتیک پیشنهادی به نام الگوریتم ژنتیک دو بخشی برای حل مساله ارائه شده است که ساختار کروموزومهای آن از دو بخش تشکیل شده است.

نوآوری این مقاله، ارائه یک روش حل جدید بر مبنای الگوریتم ژنتیک برای مساله است. نحوه کد نمودن کروموزوم ها نقش موثری در جستجوی فضای جوابهای مساله دارد که پیاده سازی عملگر تقاطع و جهش روی این ساختار کروموزوم باعث جستجوی نقاط متفاوت فضای جواب مساله می شود. در این مقاله سعی شده است با ارائه ساختاری نوین برای کروموزومها که در ادبیات موضوع مشابه آن وجود ندارد، قابلیت جستجوی فضای جواب برای الگوریتم ژنتیک افزایش یابد. همچنین تطبیق عملگرهای تلفیق و جهش موجود با این ساختار کروموزوم پیشنهادی از نوآوریهای این تحقیق است.

در ادامه در بخش ۲ مروری بر مقالات مرتبط با مساله زمانبندی تولید کارگاهی انعطاف پذیر می پردازیم. سپس مساله و محدودیتهای مساله در بخش ۳ تعریف می شود. در بخش ۴ حل مساله با استفاده از الگوریتم ژنتیک دو بخشی شرح داده می شود. در بخش ۵ به ارزیابی کارایی الگوریتم پیشنهادی و مقایسه آن با سایر الگوریتمهای موجود در ادبیات موضوع با استفاده از داده های محک پرداخته می شود. در انتها نیز در بخش ۶ نتیجه گیری و زمینه های تحقیقات آتی تبیین می شوند.

۲- مروری بر ادبیات

همانگونه که اشاره شد، در این تحقیق به مساله زمانبندی تولید کارگاهی انعطاف پذیر پرداخته می شود. مساله تولید کارگاهی انعطاف پذیر اولین بار در سال ۱۹۹۰ توسط براکر و اسچلای [۵] بررسی گردید. آنها یک الگوریتم ابتکاری چندجمله ای برای تخصیص سفارشات و زمانبندی آنها ارائه نموده اند.

کاسم و همکاران [۶] مساله زمانبندی تولید کارگاهی را به دو دسته ی مساله زمانبندی تولید کارگاهی انعطاف پذیر

⁴ Hierarchical Approach

⁵ Integrated Approach

⁶ Particle Swarm Optimization

استراتژی های تصادفی و هیوریستیک برای مقدار دهی اولیه حافظه هارمونی معرفی می شود که در کیفیت و تراکم معین اتفاق می افتد. در مرحله بعدی یک جستجوی محلی در الگوریتم نهاده شده تا توانایی اکتشاف محلی را افزایش دهد. نتایج در داده های محک مختلف کارایی و اثربخشی الگوریتم را تایید می کند.

یان و اکسیو [۱۷] در مقاله ای دیگر مساله زمانبندی تولید کارگاهی انعطاف پذیر را توسط دو الگوریتم جستجوی هارمونی ترکیبی و جستجوی همسایه بزرگ^{۱۲} (LNS) حل نموده اند. الگوریتم جستجوی هارمونی ترکیبی، مبتنی بر تکامل است در حالی که جستجوی همسایه بزرگ نمونه ای از رویکرد مبتنی بر محدودیت می باشد. در این مقاله این دو الگوریتم را ترکیب کرده و الگوریتمی تحت عنوان HHS/LNS ارائه نموده اند. الگوریتم پیشنهادی با HHS شروع شده و سپس راه حل توسط LNS بهبود می یابد. نتایج این الگوریتم نشان می دهد که الگوریتم از عملکرد خوبی برخوردار است و همچنین برای مسائل بزرگ FJSP که در داده های محک حل نشده هستند جواب بدست آورده است.

ونگ و همکاران [۱۸] الگوریتم کلونی زنبور عسل^{۱۳} و الگوریتم تخمین توزیع^{۱۴} را برای مساله زمانبندی تولید کارگاهی انعطاف پذیر پیشنهاد داده اند که هر دو بر تعادل بین اکتشاف کلی و بهره برداری محلی تاکید دارند. روشهای فراابتکاری مبتنی بر تکنیکهای برنامه ریزی محدود، توانایی در حل این مسائل را افزایش داده است.

تامانو و فوانگ [۱۹] الگوریتم کلونی زنبور عسل ترکیبی را برای حل مساله زمانبندی تولید کارگاهی انعطاف پذیر پیشنهاد داده اند که مساله اصلی را به دو زیر مساله انتخاب بهترین ماشین برای هر عملیات و تعیین توالی عملیات روی هر ماشین تقسیم می کند. در الگوریتم پیشنهادی، ابتدا چند قانون و الگوریتم جستجوی هارمونی برای ایجاد راه حل اولیه استفاده شده است. اگر همگرایی زودرس به یک بهینه محلی اتفاق بیافتد، الگوریتم شبیه سازی بازپخت برای اجتناب از رسیدن به نقاط بهینه محلی استفاده می شود. در غیر این صورت، از یک

عملیات به ماشینها و از الگوریتم شبیه سازی بازپخت^۷ برای زمانبندی عملیات به هر ماشین (جستجوی محلی) استفاده می کند.

چن و همکاران [۱۰] الگوریتم ژنتیک را برای حل مساله زمانبندی تولید کارگاهی انعطاف پذیر ارائه دادند. ژانگ و جن [۱۱] یک عملیات چند مرحله ای بر اساس الگوریتم ژنتیک از نقطه نظر برنامه ریزی پویا ارائه دادند.

اونگ و همکاران [۱۲] یک الگوریتم سیستم ایمنی مصنوعی^۸ بنام کلونال فلکس (clonaFLEX) پیشنهاد دادند. این الگوریتم بر اساس اصول انتخاب کلونال طراحی شده است که مساله زمانبندی تولید کارگاهی انعطاف پذیر را حل می کند.

پزللا و همکاران [۱۳] الگوریتم ژنتیکی را برای مساله زمانبندی تولید کارگاهی انعطاف پذیر ارائه نمود که از قوانین مختلفی برای تولید جمعیت، انتخاب اشخاص و عملگر ترکیب استفاده می نمود.

گاوو و همکاران [۱۴] ترکیبی از الگوریتم ژنتیک و جستجوی همسایگی متناوب^۹ را برای حل مساله استفاده نموده اند. جستجوی همسایگی متناوب شامل دو رویکرد جستجوی محلی است: جستجوی محلی جابجایی یک عملیات و جستجوی محلی جابجایی دو عملیات.

فتاحی و همکاران [۱۵] یک مدل ریاضی و دو رویکرد فراابتکاری^{۱۰} برای حل مساله زمانبندی تولید کارگاهی انعطاف پذیر ارائه داده اند. همچنین با در نظر گرفتن دو رویکرد فراابتکاری و رویکردهای سلسله مراتبی و یکپارچه شش الگوریتم جستجوی متفاوت ارائه داده اند.

یان و اکسیو [۱۶] الگوریتم جستجوی هارمونی ترکیبی (HHS)^{۱۱} را برای مساله زمانبندی تولید کارگاهی انعطاف پذیر ارائه داده اند. در ابتدا برای ساخت الگوریتم جستجوی هارمونی مطابق با مساله زمانبندی تولید کارگاهی انعطاف پذیر، تکنیکهای تبدیل توسعه یافتند تا بردار هارمونی پیوسته به نوعی از دو بردار گسسته تبدیل شود. سپس بردار هارمونی به جدول فعال مناسب توسط رمزگشایی موثر نگاشت می شود که فضای جستجو را کاهش می دهد. در مرحله سوم، مقدار دهی اولیه ترکیب

⁷ Simulates annealing

⁸ Artificial Immune System

⁹ Variable Neighborhood Search

¹⁰ Metaheuristics

¹¹ Hybrid harmony search

¹² Large neighborhood search

¹³ Artificial bee colony (ABC) algorithm

¹⁴ Estimation of distribution algorithm (EDA)

باید توسط یک ماشین پردازش شوند. هر عملیات O_{ik} یعنی عمل k از کار i می تواند روی یک از مجموعه از ماشینها که با S_{ik} نشان داده می شود، پردازش شود (S_{ik} زیرمجموعه ای از S است) که یکی از آنها بمنظور پردازش باید انتخاب شود.

همه ی کارها و ماشینها در زمان صفر در دسترس هستند و هر ماشین می تواند در یک لحظه فقط یک عمل را پردازش کند. وقفه غیرمجاز است، یعنی هر عمل باید بدون وقفه، بعد از شروع شدن پردازش شود تا تکمیل شود. زمان پردازش عمل O_{ik} روی ماشین M_j مثبت و از قبل مشخص است. هدف تخصیص عملیات به ماشینهای مناسب و تعیین توالی عملیات روی هر ماشین به نحوی است که سه تابع هدف کمینه شوند.

سه تابع هدف برای مساله در نظر گرفته شده است که کمینه کردن زمان تکمیل آخرین سفارش^{۱۶} (C_{max})، حداکثر بارکاری ماشینها^{۱۷} (W_{max}) و بارکاری کل^{۱۸} (WT) یا عبارتی مجموع بار کاری ماشینها می باشد.

به منظور ارائه مدل ریاضی مساله مورد نظر ابتدا به تعریف متغیرها و پارامترهای مساله می پردازیم.

i : شاخص سفارش (کار)

k : شاخص شماره عملیات

j : شاخص ماشین

n : تعداد سفارشات

m : تعداد ماشینها

k_i : تعداد عملیات سفارش i

n_j : حداکثر تعداد عملیاتی که می توان به ماشین j تخصیص داد.

$A_{ikj} = 1$ یک پارامتر صفر و یک می باشد. اگر O_{ik} بتواند در ماشین j اجرا شود مقدار ۱ و در غیر اینصورت مقدار ۰ دارد.

P_{ikj} : زمان پردازش عمل O_{ik} اگر در ماشین j اجرا شود. ($P_{ikj} > 0$)

PS_{ik} : زمان پردازش عملیات O_{ik} بعد از انتخاب یک ماشین

L : یک عدد بزرگ

الگوریتم فیلتر استفاده می شود. در نهایت، عمل تقاطع به منظور افزایش قابلیت بهره برداری ارائه شده است. نتایج تجربی در مجموعه داده های معیار نشان می دهد که الگوریتم پیشنهادی به طور موثر می تواند این مساله را حل کند.

تیکنگ و تامانو [۲۰] یک الگوریتم ژنتیک تغییر یافته برای حل مساله زمانبندی تولید کارگاهی انعطاف پذیر ارائه داده اند. الگوریتم پیشنهادی شامل روش انتخاب چرخ رولت فازی و عملگر تقاطعی که از خوشه بندی سلسله مراتبی برای خوشه بندی جمعیت در هر نسل استفاده می کند، است. همچنین عملگر جهش جدیدی ارائه شده است که به تراکم جمعیت کمک کرده و از همگرایی زودرس اجتناب می کند.

آک و کوک [۲۱] زمانبندی موازی ماشین و زمانبندی تولید کارگاهی انعطاف پذیر را با استفاده از الگوریتم ژنتیک حل نموده اند. در این مقاله ساختارهای متفاوتی برای عملگرهای الگوریتم ژنتیک در نظر گرفته شده است. زجکول و همکاران [۲۲] یک چارچوب فراابتکاری جستجوی ممنوع^{۱۵} توزیع یافته برای خوشه بندی ارائه داده اند. در این تحقیق در واقع یک جستجوی ممنوع موازی سازی ترکیبی پیشنهاد شده است. این متدولوژی مساله تولید کارگاهی انعطاف پذیر را با استفاده از بهینه سازی گسسته حل می نماید.

همانگونه که اشاره گردید، این مقاله به حل مساله زمانبندی تولید کارگاهی انعطاف پذیر جزئی با استفاده از یک الگوریتم ژنتیک پیشنهادی به نام الگوریتم ژنتیک دو بخشی می پردازد. در بخش بعد به تبیین فرضیات مساله پرداخته می شود. سپس الگوریتم ژنتیک پیشنهادی ارائه می شود.

۳- تعریف مساله

در این بخش به معرفی مساله و بیان فرضیات موجود در آن پرداخته می شود. مساله شامل n کار (سفارش) و m ماشین می باشد. در این مساله فرض می شود، مجموعه ای از ماشینها $S = \{M_1, \dots, M_m\}$ و مجموعه ای از کارها $J = \{J_1, \dots, J_n\}$ وجود دارند. هر کار J_j شامل n_j عملیات $O_{i1}, O_{i2}, \dots, O_{in}$ می باشد که هر کدام از آنها

¹⁶ Makespan

¹⁷ Maximal Machine Workload

¹⁸ Total Workload

¹⁵ Tabu search

$$W_{\max} \geq W_j \quad (12)$$

for $j=1, \dots, m$

$$C_{\max}, T_{ik}, TM_{js}, W_{\max}, W_j \geq 0 \quad (13)$$

$$Y_{ikj}, X_{ikjs} \in \{0,1\}$$

محدودیت (۱) کمینه کردن زمان تکمیل آخرین سفارش را نشان می‌دهد. محدودیت (۲) زمان پردازش عملیات O_{ik} را توسط انتخاب ماشین تعیین می‌کند. محدودیت (۳) زمان شروع هر عملیات را با توجه به زمان شروع عملیات پیش‌نیازی آن تعیین می‌کند. محدودیت (۴) بیان می‌دارد که هر ماشین فقط در یک لحظه یک عمل را پردازش کند. محدودیت‌های (۵) و (۶) زمان شروع فعالیت‌های تخصیص یافته به هر ماشین را تعیین می‌کنند. محدودیت (۷) مجاز بودن تخصیص هر عملیات به هر ماشین را تعیین می‌کند. محدودیت (۸) تعیین می‌کند که هر عملیات باید فقط به یک ماشین تخصیص یابد. محدودیت (۹) تعیین می‌کند که هر عملیات باید در یک اولویت مشخص از ماشین تخصیص یافته به آن، پردازش شود. محدودیت (۱۰) بیان کننده این امر است که اگر عملیاتی به یک ماشین تخصیص نیابد، به هیچ یک از اولویت‌های پردازش آن ماشین نیز تخصیص نخواهد یافت. محدودیت (۱۱) بار کاری هر ماشین را تعیین می‌کند. محدودیت (۱۲) حداکثر بار کاری ماشینها را تعیین می‌کند. محدودیت (۱۳) نیز علامت و نوع متغیرها را تعریف می‌کند.

مثالی از این مساله در جدول ۱ آورده شده است. در این مثال سه کار و سه ماشین در نظر گرفته شده است. اولین کار دارای سه عملیات، دومین کار دارای چهار عملیات و سومین کار دارای دو عملیات می‌باشد. در این جدول سطرها عملیات و ستونها ماشینها را نشان می‌دهد. اعداد درون جدول زمان پردازش عملیات روی هر ماشین می‌باشند. در این جدول نماد Inf ماشینهایی را نشان می‌دهد که نمی‌توانند عملیات مورد نظر را اجرا کنند. این حالت بیانگر مساله زمانبندی تولید کارگاهی انعطاف‌پذیر جزئی است.

همچنین متغیرهای تصمیم مساله عبارتند از:

C_{\max} : زمان تکمیل آخرین سفارش

W_j : بار کاری ماشین j

W_{\max} : حداکثر بار کاری ماشینها

Y_{ikj} : یک متغیر صفر و یک که اگر ماشین j برای عمل O_{ik} انتخاب شود، مقدار ۱ و در غیر اینصورت مقدار صفر می‌گیرد.

X_{ikjs} : یک متغیر صفر و یک که اگر O_{ik} روی ماشین j و در اولویت s اجرا شود، مقدار ۱ و در غیر اینصورت مقدار ۰ می‌گیرد.

T_{ik} : زمان شروع پردازش عملیات O_{ik}

TM_{js} : زمان شروع عملیاتی که در اولویت s ام روی ماشین j پردازش می‌شود.

مدل ریاضی مساله به شرح زیر است:

$$\begin{aligned} & \text{Minimize } Z_1 = C_{\max} \\ & \text{Minimize } Z_2 = W_{\max} \\ & \text{Minimize } Z_3 = \sum_{j=1}^m W_j \\ & C_{\max} \geq T_{ikj} + PS_{ikj} \end{aligned} \quad (1)$$

$$\begin{aligned} & \text{for } i=1, \dots, n \\ & \sum_j Y_{iki} \times P_{iki} = PS_{ik} \end{aligned} \quad (2)$$

$$\begin{aligned} & \text{for } i=1, \dots, n; k=1, \dots, k_{j-1} \\ & T_{ik} + PS_{ik} \leq T_{i(k+1)} \end{aligned} \quad (3)$$

$$\begin{aligned} & \text{for } i=1, \dots, n; k=1, \dots, k_{j-1} \\ & TM_{js} + PS_{ik} \times X_{ikjs} \leq TM_{j(s+1)} \end{aligned} \quad (4)$$

$$\begin{aligned} & \text{for } j=1, \dots, m; s=1, \dots, n_{j-1} \\ & TM_{js} \leq T_{ik} + (1 - X_{ikjs}) \times L \end{aligned} \quad (5)$$

$$\begin{aligned} & \text{for } j=1, \dots, m; s=1, \dots, n_j; i=1, \dots, n; \\ & k=1, \dots, k_j \\ & TM_{js} + (1 - X_{ikjs}) \times L \geq T_{ik} \end{aligned} \quad (6)$$

$$\begin{aligned} & \text{for } j=1, \dots, m; s=1, \dots, n_j; i=1, \dots, n; \\ & k=1, \dots, k_j \\ & Y_{ikj} \leq A_{ikj} \end{aligned} \quad (7)$$

$$\begin{aligned} & \text{for } j=1, \dots, m; i=1, \dots, n; k=1, \dots, k_j \\ & \sum_j Y_{ikj} = 1 \end{aligned} \quad (8)$$

$$\begin{aligned} & \text{for } i=1, \dots, n; k=1, \dots, k_j \\ & \sum_j \sum_s X_{ikjs} = 1 \end{aligned} \quad (9)$$

$$\begin{aligned} & \text{for } i=1, \dots, n; k=1, \dots, k_j \\ & = Y_{ikj} \sum_s X_{ikjs} \end{aligned} \quad (10)$$

$$\begin{aligned} & \text{for } i=1, \dots, n; k=1, \dots, k_j; j=1, \dots, m \\ & W_j = \sum_i \sum_k Y_{ikj} \times P_{ikj} \end{aligned} \quad (11)$$

for $j=1, \dots, m$

بررسی جامعی بر نمایش کروموزمها در مساله زمانبندی تولید کارگاهی انعطاف پذیر انجام داده اند. مسگونی و همکاران [۲۴] نمایش موازی کارها را برای حل مساله زمانبندی تولید کارگاهی انعطاف پذیر ارائه داده اند. کروموزم پیشنهادی آنها توسط یک ماتریس نمایش داده می شود که هر سطر ترتیب توالی هر کار است. هر عامل سطر شامل دو قسمت است که قسمت اول ماشین پردازش عملیات و قسمت دوم زمان شروع این عملیات می باشد. این رویکرد نیاز دارد یک مرتب سازی در ساختار کروموزوم انجام دهد و ارائه رمزگشایی کروموزومهای آن بسیار پیچیده است.

همچنین هو و همکاران [۲۵] کروموزوم را به دو رشته تقسیم کرده اند، که اولین قسمت نشان دهنده ترتیب عملیات و دیگری نشان دهنده ماشینها است که توسط یک آرایه با ارزش دودویی به نمایش در می آید. ساختار می تواند مساله را به روشنی و به راحتی نشان دهد، اما کد باینری فضای حافظه را افزایش می دهد، زیرا برای نشان دادن یک جایگشت معمولی از n سفارش نیاز به یک ماتریس صفر و یک دو بعدی $n \times n$ است. بنابراین تا زمانی که مقیاس مساله بزرگ باشد، فضای حافظه و زمان محاسباتی افزایش می یابد.

براساس تحلیل رویکرد ادبیات بالا، در این مقاله نمایشی نوین از کروموزمها ارائه شده است. نمایش کروموزوم به این صورت است که هر کروموزوم به دو بخش تقسیم شده و هر بخش از یک رشته تشکیل می شود. بخش اول مربوط به تعیین اولویت عملیات است. بخش دوم مربوط به ماشینهایی است که عملیات متناظر باید توسط آنها پردازش شوند.

بخش اول کروموزوم از یک رشته اعداد به طول $\sum_{i=1}^n k_i$ تشکیل شده است. مقدار هر ژن در کروموزوم بصورت تصادفی انتخاب شده و یک کار در آن قرار می گیرد. در این رشته، هر کار باید به تعداد عملیاتش تکرار شود. بخش دوم کروموزوم از یک رشته به اندازه رشته اول تشکیل شده است و ژنهای آن بصورت تصادفی از ماشینها تشکیل می شود. بعبارت دیگر هر ژن نشان دهنده ماشینی است که عملیات متناظر با آن در رشته اول باید توسط آنها پردازش شود.

به منظور تبیین بیشتر مثال زیر بیان می شود. فرض کنید دو کار و پنج ماشین موجود است. در این مساله کار اول

جدول ۱- زمان های پردازش عملیات مثال

عملیات	M1	M2	M3
O11	۶	۶	Inf
O12	Inf	۵	Inf
O13	۴	۵	۵
O21	Inf	۶	Inf
O22	Inf	۵	۷
O23	۷	۹	Inf
O24	۶	۳	Inf
O31	۵	۳	۳
O32	۴	Inf	Inf

اگر در جدول ۱ همه زمانهای پردازش عدد باشد و از نماد Inf استفاده نشود نشان دهنده مساله زمانبندی تولید کارگاهی انعطاف پذیر کلی می باشد. در بخش ۴ به ارائه روش حل برای مساله فوق پرداخته می شود.

۴- حل مساله

در این مقاله برای حل مساله زمانبندی تولید کارگاهی انعطاف پذیر از الگوریتم ژنتیک استفاده شده است. الگوریتم ژنتیک یک مدل از یادگیری ماشین است که رفتار آن از مکانیسم تکامل در طبیعت الهام گرفته شده است. این روش با ایجاد جمعیتی از افراد که هر یک در قالب کروموزوم ارائه می شوند به حل مساله می پردازند. الگوریتم ژنتیک با عملگرهای تقاطع^{۱۹}، جهش^{۲۰} و انتخاب^{۲۱} نقاط مختلف فضای جواب مساله را جستجو می کند. در این قسمت الگوریتم ژنتیک پیشنهادی با نام الگوریتم ژنتیک دو بخشی (2p-GA)^{۲۲} با تمام پارامترهایش بیان می شود.

۴-۱- نمایش کروموزوم

نحوه نمایش کروموزومها نقش حیاتی در عملکرد الگوریتم ژنتیک برای پیدا نمودن نقاط مختلف فضای جوابهای مساله دارد. همچنین تغییر و منطبق کردن عملگرهای الگوریتم ژنتیک با این ساختار کروموزوم جز اجتناب ناپذیر استفاده از الگوریتم ژنتیک می باشد. هو و همکاران [۲۳]

¹⁹ Crossover

²⁰ Mutation

²¹ Selection

²² 2Part- Genetic Algorithm

غیر موجه است. اما استراتژی دوم پذیرش کروموزوم‌های غیرموجه در جمعیت و اختصاص جریمه‌های سنگین به مقدار شایستگی^{۲۳} آنها می‌باشد. در الگوریتم ژنتیک ارائه شده در این مقاله از استراتژی دوم استفاده شده است. در این حالت می‌توان مساله زمانبندی تولید کارگاهی انعطاف‌پذیر جزئی را به مساله زمانبندی تولید کارگاهی انعطاف‌پذیر کلی تبدیل نمود، با این تفاوت که اگر یک ماشین قادر به پردازش عملیات متناظر با آن نباشد جریمه سنگینی به مقدار شایستگی آن اختصاص می‌یابد.

۴-۲- جمعیت اولیه

جمعیت اولیه یکی از ویژگی‌های الگوریتم‌های تکاملی است که بر سرعت همگرایی و کیفیت راه حل نهایی تاثیر می‌گذارد [۲۵]. در این مقاله از جمعیت اولیه تصادفی استفاده شده است. نکاتی که در نمایش کروموزوم ارائه شد در تقاطع، جمعیت اولیه و جهش باید در نظر گرفت. بخش اول کروموزوم باید توسط تابع تصادفی مقداردهی اولیه شود که هر سفارش به تعداد عملیاتی که دارد، تکرار می‌شود. در بخش دوم کروموزوم، هر زن از اعداد تصادفی که به صورت صحیح بین یک تا تعداد ماشین آلات موجود هستند، انتخاب می‌شود.

به طور مثال در شکل ۱ به علت اینکه کار یک، سه عملیات داشته سه بار تکرار شده است و کار دو دارای دو عملیات می‌باشد که دو بار تکرار شده است. بخش دوم از اعداد تصادفی صحیح بین ۱ تا ۵ ایجاد شده است. بدیهی است اگر ماشینی قادر به پردازش عملیات متناظر با آن نباشد جریمه سنگینی به مقدار شایستگی آن کروموزوم تخصیص می‌یابد.

۴-۳- عملگر انتخاب

گزینش کروموزومها برای انتقال به نسل بعد و همچنین گزینش آنها برای انجام عملگرهای تقاطع و جهش بر عهده عملگر انتخاب است.

در این مقاله برای عملگر انتخاب، از انتخاب چرخ رولت^{۲۴} و انتخاب نخبه‌گرایی^{۲۵} استفاده شده است. عملگر چرخ رولت برای انتخاب کروموزومها به منظور عمل تقاطع و جهش مورد استفاده قرار می‌گیرد و عملگر نخبه‌گرایی به

شامل سه عملیات و کار دوم شامل دو عملیات می‌باشد. پس در مجموع ۵ عملیات وجود دارد.

کروموزوم را به دو بخش تقسیم می‌کنیم. شکل ۱ نمونه‌ای از این کروموزوم را نشان می‌دهد. پنج خانه به بخش توالی عملیات و پنج خانه به بخش تخصیص ماشینها اختصاص دارد.

در بخش توالی عملیات خانه اول عدد ۱ می‌باشد. این امر نشان دهنده آن است که اولین عمل از کار اول بیشترین اولویت برای پردازش را دارد. خانه دوم عدد ۲ است که نشان دهنده آن است که اولین عمل از کار دوم اولویت بعدی را دارد. سپس خانه سوم عدد ۱ است که بیانگر دومین عمل از کار اول می‌باشد. خانه چهارم نیز عدد ۱ می‌باشد که سومین عمل از کار اول را نشان می‌دهد. خانه پنجم عدد ۲ می‌باشد که دومین عمل از کار دوم را نشان می‌دهد.

اولین خانه‌ی بخش دوم کروموزوم (تخصیص ماشین)، دارای مقدار ۴ می‌باشد. یعنی عمل اول از کار اول توسط ماشین ۴ پردازش می‌شود. خانه دوم نیز ۴ است که با توجه به رشته اول، اختصاص به عمل اول از کار دوم دارد. به همین ترتیب خانه سوم رشته اول، مربوط به عمل دوم کار اول می‌باشد که توسط ماشین ۵ پردازش می‌شود.

رشته توالی عملیات	۱	۲	۱	۱	۲
رشته تخصیص ماشین	۴	۴	۵	۳	۱

شکل ۱- نمایش کروموزوم

اگر هر عملیات قادر به پردازش روی تمام ماشینها باشد با مساله زمانبندی تولید کارگاهی انعطاف‌پذیر کلی مواجه هستیم. اما اگر هر عملیات تنها توسط بخشی از ماشین آلات قابل پردازش باشد با زمانبندی تولید کارگاهی انعطاف‌پذیر جزئی روبرو هستیم. در این تحقیق مساله زمانبندی تولید کارگاهی انعطاف‌پذیر جزئی با سه تابع هدف ذکر شده در نظر گرفته شده است. در الگوریتم‌های ژنتیک دو استراتژی در تولید کروموزومها وجود دارد استراتژی اول ایجاد کروموزومهای موجه و حذف کروموزوم

²³ Fitness Value

²⁴ Roulette wheel selection

²⁵ Elitism selection

گام ۱- دو کروموزوم را برای انجام عمل تقاطع انتخاب کنید و آنها را والد ۱ و والد ۲ بنامید.

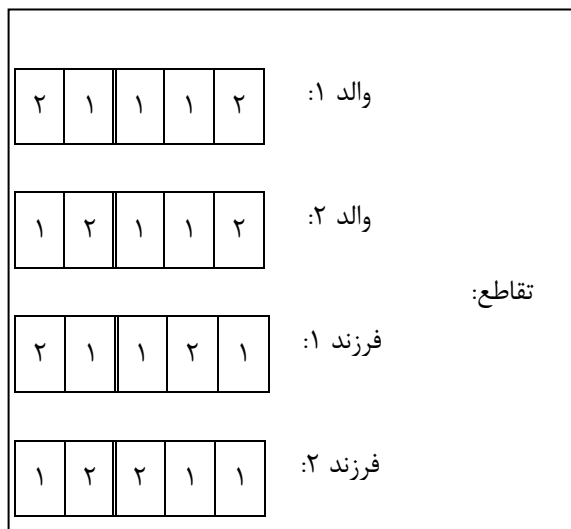
گام ۲- یک مکان تصادفی به نام S از کروموزوم والد ۱ انتخاب کنید و از آن قسمت برش دهید.

گام ۳- تکه اول کروموزوم والد ۱ را به کروموزوم فرزند ۱ انتقال دهید.

گام ۴- سایر مکانهای کروموزوم فرزند ۱ را (از نقطه S به بعد) به ترتیب از ابتدای رشته والد ۲ به کروموزوم فرزند ۱ انتقال دهید، به نحویکه تعداد تکرار هر کار از تعداد عملیاتش بیشتر نشود.

برای کروموزوم فرزند ۲ نیز گامهای ۲ تا ۴ را انجام دهید، اما جای والد ۱ و والد ۲ را تعویض کنید.

شکل ۲ عملگر تقاطع قسمت توالی عملیات را شرح می-دهد. یک عدد تصادفی S ($S=2$) انتخاب شده است. از خانه ۱ تا S از یک والد و از خانه $S+1$ تا آخر از والد دیگر به کروموزوم فرزند انتقال داده شده است. اگر کپی کردن ژنها موجب شود که تعداد تکرار هر کار از تعداد عملیاتش بیشتر شود، ژن مربوطه کپی نمی شود.



شکل ۲- تقاطع قسمت توالی عملیات

۴-۴-۲- تقاطع بخش تخصیص ماشین

عملگر تقاطع تخصیص ماشین در بخش تخصیص ماشین کروموزوم اجرا می شود تا اختصاص های مختلف عملیات به ماشین را انجام دهد. در این مثال دو نقطه به صورت تصادفی (a و b) انتخاب می کنیم در اینجا دو نقطه $a=1$ و $b=3$ می باشد. از خانه ۱ تا خانه a از کروموزوم ۱ و از

منظور انتخاب کروموزومها برای انتقال به نسل بعد مورد استفاده قرار گرفته است.

انتخاب چرخ رولت از عملگرهای بسیار رایج می باشد که به کروموزومهای دارای مقدار شایستگی بیشتر، شانس بیشتری برای انتخاب و به کروموزومهای دارای مقدار شایستگی کمتر، شانس کمتری برای انتخاب می دهد.

در نخبه سالاری، بهترین عضو هر جمعیت زنده می ماند و در جمعیت بعد حضور می یابد، به عبارت دیگر عضوی که بالاترین شایستگی را دارد به طور خودکار به جمعیت جدید منتقل می شود. عملگر نخبه سالاری در الگوریتم ژنتیک، معمولاً باعث حل سریعتر مساله می شود. نحوه شکل گیری جمعیت نسل بعد به این صورت در نظر گرفته شده است که کروموزومهای جدید بدست آمده از عمل تقاطع و جهش مستقیماً به نسل بعد منتقل می شوند. به منظور رسیدن تعداد کروموزومهای نسل بعد به اندازه جمعیت، مسایر کروموزومها با استفاده از عملگر انتخاب نخبه گرایی از نسل قبل به نسل جدید منتقل می شوند. عبارت دیگر مجموع نرخ تقاطع، جهش و نرخ گزینش کروموزومهای برتر باید برابر ۱ شود.

۴-۴-۴- تقاطع

هدف از تقاطع بدست آوردن کروموزوم های جدیدتر است. این عملگر با توجه به خصوصیات کروموزومهای انتخابی برای انجام عمل تقاطع، این خصوصیات را به کروموزومهای جدید منتقل می کند. در این مقاله از دو نوع تقاطع استفاده شده است. برای بخش تخصیص ماشین از تقاطع برش دو نقطه ای^{۲۶} استفاده می شود. برای بخش توالی عملیاتها به دلیل اینکه تکرار عملیات رخ می دهد، استفاده از تقاطع برش دو نقطه ای باعث ظهور کروموزومهای غیر موجه می شود. به همین دلیل از تقاطع پیشنهادی زیر استفاده شده است.

۴-۴-۱- تقاطع بخش توالی عملیات

عملگر تقاطع بخش توالی عملیات با عملگر تقاطع بخش تخصیص ماشین متفاوت است. الگوریتمی برای عملگر تقاطع بخش توالی عملیات ارائه شده است. این الگوریتم به صورت زیر است.

²⁶ Two cut points crossover

موضوع استفاده شده است. کلیه کدنویسی‌ها نرم افزار Matlab صورت پذیرفته است و در یک لپ تاپ شخصی با ۱,۱۷GHZ و ۳,۳۶ RAM اجرا شده است. ماهیت غیر قطعی این الگوریتم باعث می‌شود چندین اجرا برای یک مساله انجام داده تا نتایج معنی داری بدست آورد. بهترین جواب برای هر مساله، بهترین جواب بدست آمده از ۱۰ اجرا الگوریتم ژنتیک دو بخشی بدست آمده است. در این بخش عملکرد الگوریتم ارائه شده با دیگر الگوریتمهای مشهور با تابع هدف حداکثر زمان تکمیل آخرین سفارش (C_{max}) و همچنین مجموع هر سه تابع هدف (Z) بررسی شده است. نرخ تقاطع، نرخ جهش و نرخ انتخاب نخبه‌گرایی با روش آزمون و خطا به ترتیب ۰,۶۵، ۰,۳ و ۰,۰۵ در نظر گرفته شده‌اند.

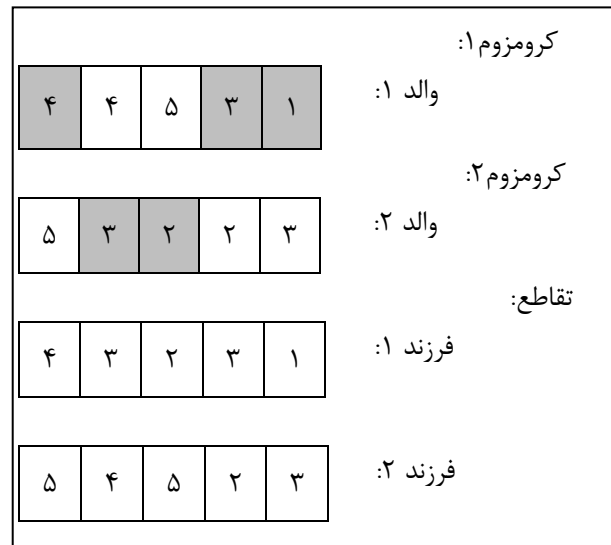
۵-۱- مجموعه داده‌ها و مسائل مختلف

در این مقاله به منظور ارزیابی الگوریتم ژنتیک پیشنهادی از دو مجموعه داده استفاده شده است که به شرح زیر می‌باشد:

(۱) داده Kacem: این مجموعه مسائل نمونه شامل ۳ مساله می‌باشد که توسط کاسم و همکاران [۸] تهیه شده است. مساله ۱ مساله ۸*۸، مساله ۲ مساله ۱۰*۱۰ و مساله ۳ مساله ۱۰*۱۵ می‌باشد. مساله ۸*۸ نمونه‌ای از مساله زمانبندی تولید کارگاهی انعطاف‌پذیری جزئی است که شامل ۸ کار با ۲۷ عملیات که توسط ۸ ماشین پردازش می‌شود (جدول ۲). مساله ۱۰*۱۰ نوعی از انعطاف‌پذیری کلی است که شامل ۱۰ کار با ۳۰ عملیات است که در ۱۰ ماشین پردازش می‌شود (جدول ۳). مساله ۱۰*۱۵ نمونه‌ای از انعطاف‌پذیری کلی است که شامل ۱۵ کار با ۵۶ عملیات که توسط ۱۰ ماشین پردازش می‌شوند (جدول ۳).

(۲) BRdata: این مجموعه داده شامل ۱۰ مساله نمونه می‌باشد که توسط برندیمارت ارائه شده است [۷]. پارامترهای هر یک از مسائل این مجموعه به صورت تصادفی و با استفاده از توزیع یکنواخت بین دو حد تولید می‌شود. تعداد کارها از ۱۰ تا ۲۰، تعداد ماشین ۴ تا ۱۵، تعداد عملیات برای هر کار ۵

خانه $a+1$ تا خانه b از کروموزوم ۲ و از خانه $b+1$ تا خانه آخر از کروموزوم ۱ انتخاب می‌کنیم.



شکل ۳- تقاطع قسمت توالی ماشین

۴-۵- عملگر جهش

جهش یکی دیگر از عملگرهای الگوریتم ژنتیک است که تنوع جمعیت را افزایش می‌دهد. از این عملگر نیز دوبار استفاده شده است (یکبار برای بخش توالی عملیات و یکبار برای بخش تخصیص ماشین). در این الگوریتم برای بخش توالی عملیات از عملگر جهش جابجایی^{۲۷} استفاده شده است و برای جهش بخش تخصیص ماشین از عملگر جهش مقدار^{۲۸} استفاده شده است که در آن یک ژن به تصادف انتخاب شده و مقدار آن به صورت تصادفی تغییر می‌کند.

معیار خاتمه تولید تعدادی مشخص از نسلهای می‌باشد که در این مقاله ۱۰۰۰ نسل فرض شده است.

۵- آزمایشات عددی

در این بخش به منظور ارزیابی کارایی الگوریتم ژنتیک پیشنهادی، به مقایسه آن با تعدادی از روشهای موجود در ادبیات موضوع که برای حل مساله زمانبندی تولید کارگاهی منعطف ارائه شده‌اند، می‌پردازیم. به منظور انجام آزمایشات، از دو مجموعه داده محک موجود در ادبیات

²⁷ Swap mutation

²⁸ Value mutation

تا ۱۵ و تعداد عملیات برای همه کارها ۵۵ تا ۲۴۰

می باشد [۷].

جدول ۲- مساله ۸*۸

M8	M7	M6	M5	M4	M3	M2	M1	عملیات	
۹	۱۰	۹۹۹	۳	۳	۵	۳	۵	O ₁₁	J ₁
۶	۹	۹	۳	۸	۵	۹۹۹	۱۰	O ₁₂	
۵	۴	۲	۶	۵	۹۹۹	۱۰	۹۹۹	O ₁₃	
۹۹۹	۹	۹۹۹	۸	۹	۳	۷	۵	O ₂₁	J ₂
۹	۱۰	۷	۶	۲	۵	۸	۹۹۹	O ₂₂	
۷	۱	۴	۶	۵	۹۹۹	۱۰	۹۹۹	O ₂₃	
۹۹۹	۹۹۹	۷	۴	۶	۹	۸	۱۰	O ₂₄	
۴	۲	۵	۶	۷	۹۹۹	۹۹۹	۱۰	O ₃₁	J ₃
۹۹۹	۱۰	۹	۸	۴	۶	۱۰	۹۹۹	O ₃₂	
۷	۹۹۹	۱۰	۹۹۹	۶	۵	۴	۱	O ₃₃	
۴	۸	۷	۹	۵	۶	۱	۳	O ₄₁	J ₄
۹	۶	۵	۱۰	۸	۷	۱۱	۱۲	O ₄₂	
۷	۵	۹	۳	۱۰	۲	۶	۴	O ₄₃	
۹۹۹	۱۰	۹۹۹	۹	۸	۷	۶	۳	O ₅₁	J ₅
۹۹۹	۶	۸	۹	۴	۷	۹۹۹	۱۰	O ₅₂	
۹۹۹	۷	۲	۴	۷	۸	۹	۹۹۹	O ₅₃	
۶	۳	۵	۷	۶	۹۹۹	۹	۱۱	O ₅₄	
۱۰	۹۹۹	۹	۶	۴	۱	۷	۶	O ₆₁	J ₆
۴	۶	۷	۹	۹	۹	۹۹۹	۱۱	O ₆₂	
۹۹۹	۱۰	۹۹۹	۱۱	۱۰	۹	۵	۱۰	O ₆₃	
۹۹۹	۱۰	۹۹۹	۷	۶	۲	۴	۵	O ₇₁	J ₇
۵	۱۰	۹	۱۱	۹	۹۹۹	۹	۹۹۹	O ₇₂	
۱۰	۹۹۹	۶	۸	۳	۹	۸	۹۹۹	O ₇₃	
۱۰	۹۹۹	۴	۹۹۹	۹	۵	۸	۲	O ₈₁	J ₈
۹۹۹	۱۰	۹۹۹	۹	۸	۷	۴	۷	O ₈₂	
۱	۷	۶	۵	۸	۹۹۹	۹	۹	O ₈₃	
۹۹۹	۸	۵	۱	۷	۳	۹۹۹	۹	O ₈₄	

جدول ۳- مساله ۱۰*۱۰

M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	عملیات	
۵	۹	۸	۲	۵	۳	۹	۶	۴	۱	O ₁₁	J ₁
۴	۱۱	۴	۱۰	۸	۴	۳	۱	۱	۴	O ₁₂	
۳	۱۰	۵	۹	۶	۵	۱	۵	۲	۳	O ₁₃	
۴	۸	۱۵	۴	۸	۹	۵	۴	۱۰	۲	O ₂₁	J ₂
۱	۷	۱۰	۱	۶	۹	۱	۷	۸	۴	O ₂₂	
۲	۹	۱۴	۵	۳	۵	۷	۲	۱۱	۶	O ₂₃	
۱	۸	۳	۵	۳	۴	۹	۸	۵	۸	O ₃₁	J ₃
۲	۷	۱	۴	۶	۲	۱	۶	۳	۹	O ₃₂	
۴	۳	۲	۱	۹	۴	۵	۸	۱	۷	O ₃₃	
۶	۱	۷	۱	۵	۹	۴	۶	۱۰	۵	O ₄₁	J ₄
۴	۸	۹	۶	۴	۷	۸	۳	۲	۴	O ₄₂	
۲	۵	۳	۸	۵	۶	۱	۱۲	۳	۷	O ₄₃	
۶	۲	۱۵	۵	۳	۶	۵	۴	۱۰	۷	O ₅₁	J ₅
۷	۱	۶	۸	۲	۸	۹	۳	۶	۵	O ₅₂	
۹	۱۲	۱۱	۳	۴	۱۰	۱	۴	۱	۶	O ₅₃	
۱۰	۳	۸	۷	۲	۴	۸	۱۰	۹	۸	O ₆₁	J ₆
۱۵	۲	۹	۶	۳	۴	۵	۱۲	۳	۷	O ₆₂	
۱۱	۱	۵	۱	۴	۳	۶	۳	۷	۴	O ₆₃	
۷	۱۰	۱۳	۴	۹	۴	۳	۸	۷	۱	O ₇₁	J ₇
۳	۱۲	۲	۱۱	۶	۳	۲	۱	۸	۳	O ₇₂	
۷	۵	۱۴	۸	۱	۲	۱	۲	۴	۵	O ₇₃	
۸	۱۲	۵	۸	۹	۲	۳	۱۱	۷	۵	O ₈₁	J ₈
۴	۸	۶	۴	۱۳	۵	۷	۱۰	۳	۸	O ₈₂	
۵	۹	۷	۵	۳	۴	۵	۱۳	۲	۶	O ₈₃	
۴	۵	۷	۶	۱	۸	۳	۱	۹	۳	O ₉₁	J ₉
۷	۶	۹	۱	۳	۷	۵	۲	۶	۴	O ₉₂	
۱۲	۱۰	۳	۲	۱	۶	۸	۴	۵		O ₉₃	
۶	۲۰	۶	۲	۱	۷	۶	۱	۳	۴	O _{10,1}	J ₁₀
۱۵	۱۷	۴	۱	۴	۹	۱	۸	۱	۳	O _{10,2}	

جدول ۳- مساله ۱۰*۱۰

M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	عملیات
۲۳	۱۰	۴	۲	۵	۳	۲	۴	۲	۹	O _{10,3}

جدول ۴- مساله ۱۰*۱۵

M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	عملیات	
۴	۹	۸	۲	۵	۳	۹	۶	۴	۱	O ₁₁	J ₁
۳	۴	۱۱	۴	۱۰	۸	۴	۳	۱	۱	O ₁₂	
۲	۳	۱۰	۵	۹	۶	۵	۱	۵	۲	O ₁₃	
۴	۴	۸	۱۵	۴	۸	۹	۵	۴	۱۰	O ₁₄	
۱	۷	۱۰	۱	۶	۹	۱	۷	۸	۴	O ₂₁	J ₂
۲	۹	۱۴	۵	۳	۵	۷	۲	۱۱	۶	O ₂₂	
۲	۸	۳	۵	۳	۴	۹	۸	۵	۸	O ₂₃	
۲	۷	۱	۴	۶	۲	۱	۶	۳	۹	O ₂₄	
۴	۳	۲	۱	۹	۴	۵	۸	۱	۷	O ₃₁	J ₃
۶	۱	۷	۱	۵	۹	۴	۶	۱۰	۵	O ₃₂	
۴	۸	۹	۶	۴	۷	۸	۳	۲	۴	O ₃₃	
۲	۵	۳	۸	۵	۶	۱	۱۲	۳	۷	O ₃₄	
۴	۵	۶	۳	۲	۱	۴	۵	۲	۶	O ₄₁	J ₄
۵	۸	۵	۳۶	۲	۱	۴	۷	۵	۸	O ₄₂	
۲	۵	۶	۳	۱	۵	۴	۲	۶	۹	O ₄₃	
۱	۲	۴	۵	۷	۲	۶	۵	۴	۱۱	O ₄₄	
۱	۲	۴	۷	۸	۵	۳	۲	۹	۶	O ₅₁	J ₅
۵	۴	۷	۲۸	۲	۵	۳	۶	۴	۵	O ₅₂	
۹	۷	۴	۲	۵	۶	۳	۴	۲	۶	O ₅₃	
۵	۷	۴	۵	۲	۳	۲	۴	۵	۶	O ₅₄	
۲	۴	۵	۸	۹	۶	۲	۳	۱	۴	O ₆₁	J ₆
۵	۶	۴	۵	۷	۴	۵	۶	۳	۱	O ₆₂	
۴	۵	۸	۹	۶	۳	۵	۲	۴	۱	O ₇₁	J ₇
۵	۲	۴	۵	۳	۲	۵	۴	۱	۲	O ₇₂	
۷	۸	۵	۱	۴	۵	۲	۶	۳	۲	O ₈₁	J ₈
۵	۲	۱	۴	۵	۳	۲	۶	۵	۴	O ₈₂	
۵	۴	۵	۸	۴۹	۵	۲	۴	۵	۳	O ₈₃	
۲	۱۱	۴	۶	۳	۲	۵	۳۶	۲	۱	O ₈₄	
۱	۵	۲۳	۱۰	۱۱	۴۴	۲۲	۲	۳	۶	O ₉₁	J ₉
۱۶	۱۸	۱۴	۱۲	۱۰	۱۵	۱۲	۲	۳	۲	O ₉₂	
۶	۵	۷	۴	۶	۹	۵	۱۲	۱۷	۲۰	O ₉₃	
۲	۵۶	۴	۷	۸	۵	۴	۷	۸	۹	O ₉₄	
۱	۴	۵	۲	۳	۵۶	۴	۷	۸	۵	O _{10,1}	J ₁₀
۴	۵	۲	۴	۵	۸	۹	۶	۵	۲	O _{10,2}	
۱	۲	۵	۴	۷	۴	۵	۲	۳	۶	O _{10,3}	
۲	۵	۴	۷	۸	۵	۶	۵	۲	۳	O _{10,4}	
۱	۲	۴	۱	۲	۵	۶	۳	۲	۱	O _{11,1}	J ₁₁
۱	۱۲	۱۰	۴	۱	۲	۳	۶	۳	۲	O _{11,2}	
۵	۲	۳	۶	۴	۸	۵	۲	۶	۳	O _{11,3}	
۴	۲	۲۵	۱	۴	۲	۶	۴۵	۱	۴	O _{11,4}	
۲	۴	۲	۵	۶	۳	۶	۵	۸	۹	O _{12,1}	J ₁₂
۲۱	۵	۶	۶۳	۷۵	۴	۵	۹	۸	۵	O _{12,2}	
۵	۲	۴	۵	۲	۳	۶	۴	۵	۱۲	O _{12,3}	
۴	۸	۵	۲	۳	۶	۵	۹	۷	۸	O _{12,4}	
۲	۶	۴	۶	۵	۸	۶	۵	۲	۴	O _{13,1}	J ₁₃
۲	۳	۶	۶	۸	۵	۷	۴	۵	۳	O _{13,2}	
۲	۴	۵	۶	۴	۵	۸	۵	۴	۵	O _{13,3}	
۴	۶	۵	۸	۴	۵	۶	۵	۲	۳	O _{13,4}	
۵	۴	۸۵	۴	۵	۶	۴	۵	۳	۲	O _{14,1}	J ₁₄
۶	۲	۴	۵	۶	۸	۵	۴	۲	۶	O _{14,2}	
۴	۵	۲	۳	۶	۵	۸	۴	۲۵	۳	O _{14,3}	
۴	۵	۸	۶	۳	۲	۴	۶	۵	۸	O _{14,4}	
۴	۵	۲	۳	۶	۵	۸	۶	۵	۲	O _{15,1}	J ₁₅
۵	۲	۳	۵	۲	۴	۵	۲	۶	۵	O _{15,2}	
۵	۷	۴	۸	۲	۵	۳	۲	۵	۴	O _{15,3}	
۸	۴	۵	۶	۳	۲	۱۴	۱۱	۲	۶	O _{15,4}	

جدول ۵- نتایج بر مجموعه داده Kacem

مسائل	توابع هدف	AL+CGA	PSO+SA	moGA	clonaFLEX	2p-GA
مساله ۸*۸	C_{max}	۱۵	۱۶	۱۴		۱۱
	Z		۱۰۲	۱۰۲		۸۴
مساله ۱۰*۱۰	C_{max}	۷	۷	۷	۷	۶
	Z		۵۷	۵۵		۵۳
مساله ۱۰*۱۵	C_{max}	۲۴	۱۲		۱۲	۱۱
	Z	۱۲۶	۱۱۴			۱۱۵

دومین راه حل فقط از رویکرد محلی سازی استفاده شده است که مقدار C_{max} را ۱۶ بدست آورده است.

دومین الگوریتم PSO+SA است که توسط اکسیا و وو ارائه شده است [۹]. در این الگوریتم نیز دو جواب برای این مساله بدست آورده است. در اولین جواب مقدار C_{max} برابر ۱۵ و مقدار Z (مجموع سه تابع هدف مساله) ۱۰۲ می باشد. در دومین جواب مقدار C_{max} برابر ۱۶ و مقدار Z نیز برابر ۱۰۲ شده است. این در حالی است که الگوریتم پیشنهادی برای C_{max} مقدار ۱۱ و برای Z مقدار ۸۴ را ارائه می دهد.

در مساله ۱۰*۱۰ نیز هم در تابع هدف اول و هم در تابع هدف دوم جواب های بهتری نسبت به سایر الگوریتم ها بدست آمده است.

در مساله ۱۰*۱۵ الگوریتم 2p-GA از نظر تابع هدف اول نسبت به سایر الگوریتم ها جواب بهتری ارائه داده است. همچنین الگوریتم پیشنهادی از نظر تابع هدف دوم نسبت به الگوریتم AL+CGA جواب بهتری ارائه داده است. اما در این حالت الگوریتم PSO+SA جواب بهتری نسبت به الگوریتم 2p-GA ارائه نموده است.

تحقیقات اخیر نشان می دهد که مساله زمانبندی تولید کارگاهی انعطاف پذیر حل شده است که الگوریتم hGA را پیشنهاد داده است [۱۴].

به منظور مقایسات بیشتر در جدول ۶ الگوریتم 2p-GA با الگوریتم hGA که توسط گاوو و همکاران [۱۴] پیشنهاد شده است و همچنین الگوریتم AIA که توسط باقری و همکاران [۴] ارائه شده اند، به ازای اندازه جمعیت های مختلف مقایسه شده است.

در مساله ۸*۸ در هر دو تابع هدف الگوریتم 2p-GA به نتایج بهتری نسبت به الگوریتم های دیگر رسیده است. در

۵-۲- نتایج محاسبات

اولین داده برای انجام محاسبات از مجموعه داده Kacem استفاده شده است. جدول ۵ بهترین راه حل از ۱۰ اجرا الگوریتم ژنتیک پیشنهادی در داده Kacem نشان می دهد که با الگوریتم AL+CGA ترکیب رویکرد محلی سازی و الگوریتم ژنتیک توسط کاسم و همکاران [۸]، PSO+SA ترکیب بهینه سازی ازدحام ذرات و الگوریتم شبیه سازی بازپخت توسط اکسیا و وو [۹]، moGA الگوریتم ژنتیک پیشنهادی توسط ژانگ و جن [۱۱] و clonaFLEX یک الگوریتم ایمنی است که توسط اونگ و همکاران [۱۲] و AIA که یک الگوریتم ایمنی مصنوعی است که توسط باقری و همکاران [۴] مقایسه شده است. داده های نتایج الگوریتم ها روی مجموعه داده Kacem از مقاله باقری [۴] آورده شده است. الگوریتم ژنتیک دو بخشی (2p-GA) الگوریتم پیشنهادی این مقاله می باشد.

علت خالی بودن برخی خانه های این جدول این است که مقالات مرتبط با الگوریتم های مورد نظر یا تابع هدف مذکور را در مساله در نظر نگرفته اند یا برای این مجموعه داده نتایجی را ارائه نداده اند.

همانطور که در جدول ۲ مشاهده می شود، در مساله ۸*۸ بهترین C_{max} برای الگوریتم ژنتیک دو بخشی می باشد. اولین الگوریتمی که در جدول ۲ بررسی شده است، الگوریتم AL+CGA که توسط کاسم و همکاران [۸] ارائه شده است. آنها مساله ۸*۸ را که دارای انعطاف پذیری جزئی است توسط رویکرد محلی سازی و الگوریتم ژنتیک حل نموده اند. دو راه حل برای این مساله بدست آمده است که اولین راه حل ترکیب رویکرد محلی سازی و الگوریتم ژنتیک مقدار C_{max} را ۱۵ بدست آورده است.

سفارش (C_{max}) در نظر گرفته شده است. نتایج محاسبات سایر الگوریتمها از مقاله باقری [۴] آورده شده است.

در جدول ۸ الگوریتم 2p-GA با ۴ الگوریتم مقایسه شده است که ۴ الگوریتم شامل الگوریتم ژنتیک GA-chen که توسط چن و همکاران [۱۰]، الگوریتم ژنتیک GA-Jia که توسط جیا و همکاران [۲۶]، الگوریتم ایمنی مصنوعی AIA توسط باقری و همکاران [۴] و الگوریتم ژنتیک GA-Pezzalla توسط پزلا [۱۳] پیشنهاد شده است.

مساله ۱۰*۱۰ نیز مانند مساله ۸*۸ نتایج بهتری حاصل شده است. در مساله ۱۰*۱۵ در تابع هدف اول نتایج یکسان است ولی در تابع هدف دوم الگوریتم 2p-GA با الگوریتم AIA نتایج مشابهی با سائز جمعیت یکسان بدست می‌دهد. در این حالت الگوریتم hGA نسبت به دو الگوریتم دیگر نتیجه بهتری داده است.

مجموعه داده دوم برای مقایسات، مجموعه BRdata می‌باشد که توسط برندیمارت ارائه شده است [۷]. این مجموعه داده در جدول ۷ آورده شده است. در این مجموعه داده ۱۰ مساله مورد بررسی قرار گرفته است. تابع هدف کمینه کردن حداکثر زمان تکمیل آخرین

جدول ۶- مقایسه الگوریتم های 2p-GA، AIA، hGA بر مجموعه داده Kacem

مسائل	توابع هدف	hGA		AIA		2p-GA	
		نتایج	سائز جمعیت	نتایج	سائز جمعیت	نتایج	سائز جمعیت
مساله ۸*۸	C_{max}	۱۴	۳۰۰	۱۴	۴۰۰	۱۱	۴۰۰
	Z	۱۰۳		۱۰۳		۸۴	
مساله ۱۰*۱۰	C_{max}	۷	۳۰۰	۷	۲۰۰۰	۶	۲۰۰۰
	Z	۵۵		۵۵		۵۳	
مساله ۱۰*۱۵	C_{max}	۱۱	۵۰۰	۱۱	۵۰۰۰	۱۱	۵۰۰۰
	Z	۱۱۳		۱۱۵		۱۱۵	

جدول ۷- مجموعه داده BRdata

مسائل	njob	nmachine	noperation	meq	proc
MK01	۱۰	۶	۷-۵	۳	۷-۱
MK02	۱۰	۶	۷-۵	۶	۷-۱
MK03	۱۵	۸	۱۰-۱۰	۵	۲۰-۱
MK04	۱۵	۸	۱۰-۳	۳	۱۰-۱
MK05	۱۵	۴	۱۰-۵	۲	۱۰-۵
MK06	۱۰	۱۵	۱۵-۱۵	۵	۱۰-۱
MK07	۲۰	۵	۵-۵	۵	۲۰-۱
MK08	۲۰	۱۰	۱۰-۵	۲	۲۰-۵
MK09	۲۰	۱۰	۱۵-۱۰	۵	۲۰-۵
MK10	۲۰	۱۵	۱۵-۱۰	۵	۲۰-۵

meq: ماکزیمم تعداد ماشین در دسترس در هر عملیات

proc: مینیمم و ماکزیمم زمان پردازش در هر عملیات

جدول ۸- نتایج بر مجموعه داده BRdata

مسائل	LB	2p-GA	GA-chen	GA-Jia	AIA	GA-Pezzalla
-------	----	-------	---------	--------	-----	-------------

Dev	C _{max}	Dev	C _{max}	Dev	C _{max}	Dev	C _{max}	C _{max}		
.	۴۰	.	۴۰	.	۴۰	.	۴۰	۴۰	۳۶	MK01
.	۲۶	.	۲۶	+۷,۱۴	۲۸	+۱۰,۳۴	۲۹	۲۶	۲۴	MK02
.	۲۰۴	.	۲۰۴	.	۲۰۴	.	۲۰۴	۲۰۴	۲۰۴	MK03
.	۶۰	.	۶۰	+۱,۶۴	۶۱	+۴,۷۶	۶۳	۶۰	۴۸	MK04
.	۱۷۳	.	۱۷۳	+۱,۷۰	۱۷۶	+۴,۴۲	۱۸۱	۱۷۳	۱۶۸	MK05
.	۶۳	.	۶۳	-۱,۶۱	۶۲	-۵,۰۰	۶۰	۶۳	۳۳	MK06
.	۱۳۹	+۰,۷۱	۱۴۰	+۴,۱۳	۱۴۵	+۶,۰۸	۱۴۸	۱۳۹	۱۳۳	MK07
.	۵۲۳	.	۵۲۳	.	۵۲۳	.	۵۲۳	۵۲۳	۵۲۳	MK08
.	۳۱۱	+۰,۳۲	۳۱۲	-۰,۳۲	۳۱۰	-۰,۹۷	۳۰۸	۳۱۱	۲۹۹	MK09
+۰,۴۷	۲۱۲	+۱,۴۰	۲۱۴	+۲,۳۱	۲۱۶	+۰,۴۷	۲۱۲	۲۱۱	۱۶۵	MK10
+۰,۰۴		+۰,۲۴		+۱,۵		+۲,۰۱				میانگین بهبود

تطبيق یافته عملگرهای جهش و تقاطع دانست که به صورت همزمان، هم توالی و هم تخصیص ماشین را تغییر داده و در نتیجه تمام فضای جوابهای مساله را بررسی می کند.

در جدول ۸ نتایج هر الگوریتم با الگوریتم پیشنهادی توسط شاخصی به نام *Dev* ارزیابی شده است که این شاخص از رابطه زیر بدست می آید.

$$Dev = [(C_f - C_{best}) / C_f] * 100\%$$

۶- نتیجه گیری

در این مقاله به مساله زمانبندی تولید کارگاهی انعطاف پذیر پرداخته شده است که یکی از موضوعات بارز در برنامه ریزی تولید می باشد. اهداف مساله کمینه کردن حداکثر زمان تکمیل آخرین سفارش (C_{max})، حداکثر بارکاری ماشینها (W_{max}) و بارکاری کل (W_T) یعنی مجموع بار کاری تمام ماشینها است. مساله زمانبندی تولید کارگاهی انعطاف پذیر جز مسائل NP-hard محسوب می شود. بنابراین بدست آوردن جواب بهینه برای مسائل یا سائز متوسط و بزرگ در زمان معقول امکان پذیر نیست و باید از روشهای ابتکاری یا فراابتکاری برای حل مساله استفاده نمود. به همین منظور یک الگوریتم ژنتیک به نام الگوریتم ژنتیک دو بخشی (2p-GA) برای حل این مساله پیشنهاد شده است.

در الگوریتم ژنتیک دو بخشی پیشنهادی روشی نوین جهت نمایش کروموزوم ارائه شده است. در این الگوریتم ساختار کروموزوم از دو بخش تشکیل شده است که بخش اول مربوط به توالی عملیات و بخش دوم مربوط به تخصیص عملیات به ماشینها می باشد. همچنین عملگرهای تقاطع و جهش سازگار با این ساختار کروموزوم پیشنهاد شده اند.

در این مقاله به منظور اعتبارسنجی الگوریتم ژنتیک پیشنهادی از دو مجموعه داده محک استفاده شده است. اولین داده برای انجام مقایسات از مجموعه داده Kacem استفاده می کند [۹]. به منظور ارزیابی الگوریتم ژنتیک

در رابطه با C_{best} بهترین C_{max} بدست آمده الگوریتم پیشنهادی و C_f بهترین C_{max} از الگوریتمی است که مورد مقایسه قرار داده است. این شاخص توسط باقری و همکاران [۴] مورد استفاده قرار گرفته است. مثبت بودن مقدار شاخص *Dev*، نشان از برتری الگوریتم ژنتیک دو بخشی نسبت به الگوریتم مورد مقایسه و منفی بودن این شاخص نشان از برتری الگوریتم مورد مقایسه نسبت به الگوریتم ژنتیک دو بخشی دارد.

از میانگین گیری روی *Dev*، میانگین بهبود بدست می آید که در واقع بهبود الگوریتم پیشنهادی نسبت به الگوریتم های مورد مقایسه را به صورت کلی نشان می دهد. علامت مثبت برای این شاخص، نشان دهنده برتری الگوریتم ژنتیک پیشنهادی نسبت به الگوریتم مورد مقایسه است. در جدول ۸ الگوریتم پیشنهادی 2p-GA از تمامی الگوریتم های مورد مقایسه نتیجه بهتری داده است.

نحوه کد نمودن کروموزومها نقش موثری در جستجوی فضای جوابهای مساله و کارایی الگوریتم ژنتیک دارد. زیرا پیاده سازی عملگرهای تقاطع و جهش روی این ساختار کروموزوم است که باعث جستجوی نقاط متفاوت فضای جواب مساله می شود. در این مقاله سعی شده است با ارائه ساختاری نوین برای کروموزومها که در ادبیات موضوع مشابه آن وجود ندارد، قابلیت جستجوی فضای جواب برای الگوریتم ژنتیک افزایش یابد. می توان علت برتری الگوریتم ژنتیک پیشنهادی را ساختار کروموزوم و نوع

این مجموعه داده الگوریتم ژنتیک پیشنهادی با چهار الگوریتم دیگر مقایسه شده است که نتایج نشان از برتری الگوریتم ژنتیک پیشنهادی در این تحقیق دارد. در تحقیقات آتی می‌توان اضافه نمودن فرضیات جدیدی نظیر دسته بندی سفارشات، وجود محدودیتهای تقدمی بین سفارشات و یا قطع عملیات را در نظر گرفت. همچنین در نظر گرفتن سایر توابع هدف نظیر کمینه نمودن مجموع دیرکرد سفارشات یا تعداد کارهایی که دچار دیرکرد می‌شوند نیز می‌تواند زمینه دیگری برای تحقیقات آتی باشد. ارائه روشهای حل متفاوت برای حل مساله مذکور نیز حوزه دیگری از تحقیقات آتی می‌باشد.

پیشنهادی آن را با الگوریتم های ارائه شده توسط ۴ تحقیق مختلف [۸، ۹، ۱۱ و ۱۲] مقایسه نموده‌ایم. نتایج نشان از برتری الگوریتم ژنتیک پیشنهادی نسبت به سایر الگوریتم ها دارد.

همچنین الگوریتم ژنتیک پیشنهادی در این تحقیق به ازای اندازه جمعیت‌های مختلف با الگوریتم hGA، ارائه شده توسط گاوو و همکاران [۱۴] و الگوریتم AIA، ارائه شده توسط باقری و همکاران [۴] مقایسه شده است که نتایج نشان از برتری الگوریتم ژنتیک پیشنهادی در این تحقیق دارد.

مجموعه داده دوم برای مقایسات مجموعه داده BRdata می‌باشد که توسط برندیمارت [۷] ارائه شده است. روی

مراجع

- [1] Pinedo, M.(2002)."Scheduling: Theory, Algorithms and Systems". Prentice-Hall, Englewood Cliffs, NJ.
- [2]Roshanaei, V., Naderi, B., Jolai, F., Khalili, M. (2009). " A variable neighborhood search for job shop scheduling with set-up times to minimize makespan". Future Generation Computer Systems, Vol. 25, pp. 654-661.
- [3] Ombuki, B., Ventresca, M.(2004)."Local search genetic algorithms for the job shop scheduling problem". Applied Intelligence, Vol. 21, pp. 99-109.
- [4] Bagheri, A., Zandieh M., Mahdavi I., Yazdani M.(2010). "An artificial immune algorithm for the flexible job-shop scheduling problem". Future Generation Computer Systems, Vol. 26, pp. 533-541.
- [5] Brucker, P., Schile, R. (1990). "Job-shop scheduling with multi-purpose machines". Computing, Vol. 45, pp.369-375.
- [6] Kacem, I., Hammadi, S., Borne, P. (2002). "Pareto-optimality approach for flexible job-shop scheduling problems: Hybridization of evolutionary algorithms and fuzzy logic". Mathematics and Computers in Simulation, Vol. 60, pp. 245-276.
- [7] Brandimarte, P.(1993). "Routing and scheduling in a flexible job shop by taboo search". Annals of operations Research, Vol. 41, pp. 157-183.
- [8] Kacem, I., Hammadi, S., Borne, P.(2002). "Approach by localization and multi objective evolutionary optimization for flexible job-shop scheduling problems". IEEE Transactions on Systems, Man, and Cybernetics, Vol. 32, pp. 1-13.
- [9] Xia, W., Wu, Z.(2005). "An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problem". Computers and Industrial Engineering, Vol. 48, pp. 409-425.
- [10] Chen, H., Ihlow, J., Lehmann, C.(1999). "A genetic algorithm for flexible Job-shop scheduling", IEEE International Conference on Robotics and Automation, Detroit,pp.1120-1125.
- [11] Zhang,H., Gen, M. (2005). Multistage-based genetic algorithm for flexible job-shop scheduling problem, Journal of Complexity International, Vol. 11, 223-232.
- [12] Ong, Z.X., Tay, J.C., Kwoh, C.K.(2005). "Applying the Clonal Selection Principle to Find Flexible Job-Shop Schedules", in: LNCS, Vol. 3627, pp. 442-455.
- [13] Pezzella, F., Morganti, G., Ciaschetti, G.(2008). "A genetic algorithm for the flexible jobshop scheduling problem". Computers and Operations Research, Vol. 35 (10), pp. 3202-3212.

- [14] Gao, J., Sun, L., Gen, M.(2008). "A hybrid genetic and variable neighborhood descent for flexible job shop scheduling problems", *Computers and Operations Research*, Vol. 35 (9), pp. 2892-2907.
- [15] Fattahi, P., Saidi Mehrabad, M., Jolai, F.(2007). "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems". *Journal of Intelligent Manufacturing*, Vol. 18 (3), pp. 331-342.
- [16] Yuan, Y. , Xu, H. (2013). A hybrid harmony search algorithm for the flexible job shop scheduling problem. *Applied Soft Computing*, Volume 13, Issue 7, Pages 3259–3272.
- [17] Yuan, Y. , Xu, H. (2013). An integrated search heuristic for large-scale flexible job shop scheduling problems. *Computers & Operations Research*, Volume 40, Issue 12, Pages 2864–2877.
- [18] Wang, L., Wang, S., Xu, Y., Zhou, G., & Liu, M. (2012). A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem. *Computers & Industrial Engineering*, 62(4), 917–926.
- [19]Thammano, A., Phu-ang, A.(2013). A Hybrid Artificial Bee Colony Algorithm with Local Search for Flexible Job-Shop Scheduling Problem. *Procedia Computer Science* 20 ,Pages 96 – 101.
- [20] Teekeng, W., Thammano,A. (2012). Modified Genetic Algorithm for Flexible Job-Shop Scheduling Problems. *Procedia Computer Science* 12,Pages 122 – 128.
- [21] Ak, B., Koc, E.(2012). A guide for genetic algorithm based on parallel machine scheduling and flexible job-shop scheduling. *Procedia - Social and Behavioral Sciences* 62, Pages 817 – 823.
- [22] Zejko, W., Hejducki, Z., Uchroński, M., Wodecki,M.(2012). Solving the Flexible Job Shop Problem on Multi-GPU. *Procedia Computer Science* 9, Pages 2020 – 2023.
- [23] Ho, N. B., Tay, J. C., Edmund, M., Lai, K. (2007). "An effective architecture for learning and evolving flexible job shop schedules". *European Journal of Operational Research*, Vol. 179, pp. 316–333.
- [24] Mesghouni, K., Hammadi, S., Borne, P. (1997). "Evolution programs for job-shop scheduling". In *Proceedings of the IEEE international conference on computational cybernetics and simulation* ,Vol. 1, pp. 720–725.
- [25] Shahryar, R., Hamid, R. T., Magdy, M. A. S. (2007). "A novel population initialization method for accelerating evolutionary algorithms". *Computers and Mathematics with Application*, Vol. 53, pp. 1605–1614.
- [26] Jia, H.Z., Nee, A.Y.C., Fuh, J.Y.H., Zhang, Y.F.(2003). "A modified genetic algorithm for distributed scheduling problems". *International Journal of Intelligent Manufacturing*, Vol. 14, pp. 351-362.