مجله تحقیق در عملیات در کاربردهای آن، سال سیزدهم، شماره چهارم (پیاپی ۵۱)، زمستان ۹۵، ص ص ۳۷-۲۱
شاپا ۷۲۸۶-۲۲۵۱

ترکیبی جدید از الگوریتم‌های فرا ابتکاری ژنتیک و شبیه‌سازی تبرید برای حل مساله زمان‌بندی کارگاهی انعطاف‌پذیر

محمدعلی بهشتی‌نیا*[۱]، ملیحه حسنی بیدگلی[۲]
۱- استادیار، دانشگاه سمنان، مهندسی صنایع، سمنان، ایران
۲- دانشجوی کارشناسی ارشد، دانشگاه پیام نور تهران مرکز، مهندسی صنایع، تهران، ایران

رسید مقاله: ۱۹ تیر ۱۳۹۵
پذیرش مقاله: ۸ آذر ۱۳۹۵

چکیده
در سال‌های اخیر، مساله زمان‌بندی سیستم‌های تولید کارگاهی انعطاف‌پذیر موردتوجه محققین مختلفی قرار گرفته است. این مقاله به بررسی مساله زمان‌بندی تولید کارگاهی انعطاف‌پذیر در حالت مونتاژ و زمان‌های پردازش وابسته به‌توالی می‌پردازد. هدف تخصیص عملیات به مجموعه‌ای از ماشین‌آلات و تعیین توالی پردازش هر یک از آن‌ها برای کمینه نمودن مجموع زمان تکمیل محصولات نهایی می‌باشد. با توجه به عدم بررسی این مساله در ادبیات موضوع، ابتدا مدل ریاضی مساله ارایه می‌شود. با توجه به NP-hard بودن مساله، سه الگوریتم فرا ابتکاری شامل الگوریتم‌های ژنتیک، شبیه‌سازی تبرید و یک الگوریتم ترکیبی که دو الگوریتم قبلی را با یکدیگر ادغام نموده، برای حل مساله پیشنهاد شده است. در انتها نیز به مقایسه الگوریتم‌ها با یکدیگر پرداخته می‌شود. نتایج محاسباتی نشان از کارایی بهتر الگوریتم ترکیبی نسبت به دو الگوریتم دیگر دارد.

کلمات کلیدی: زمان‌بندی، محیط کارگاهی منعطف، الگوریتم ژنتیک، شبیه‌سازی تبرید، خط مونتاژ، زمان‌های آماده‌سازی وابسته به‌توالی.

۱ مقدمه
تخصیص بهینه منابع محدود در طی زمان مساله‌ای است که از دیرباز موردتوجه بشر بوده و امروزه مطالعات گسترده‌ای روی این موضوع با عنوان زمان‌بندی و توالی عملیات صورت گرفته است. توالی عملیات، ترتیبی را که فعالیت‌ها باید مطابق آن انجام گردند، تعیین می‌نماید. زمان‌بندی نیز تعیین می‌کند که هر فعالیت در چه زمانی باید انجام گیرد.

* عهده‌دار مکاتبات
آدرس الکترونیکی: beheshtinia@semnan.ac.ir

زمان‌بندی سیستم‌های تولید کارگاهی (JSS)[1] به دلیل جایگاه ویژه آن در مراکز تولیدی، بسیار موردتوجه مدیران واحدهای تولیدی می‌باشد. در این مدل فرض می‌شود که n سفارش وجود دارند که هر یک از آن‌ها دارای m عملیات مختلف می‌باشد. برای انجام هر نوع عملیات (به‌عنوان‌مثال پرس‌کاری) یک ماشین وجود دارد و اولویت انجام عملیات مختلف برای هر سفارش برای تکمیل شدن آن ممکن است، متفاوت باشد. زمان‌بندی سیستم تولید کارگاهی انعطاف‌پذیر (FJSS)[2] شکل توسعه‌یافته سیستم تولید کارگاهی می‌باشد با این تفاوت که در آن فرض می‌شود برای انجام هر عملیات مجموعه‌ای از ماشین‌ها در دسترس هستند.

از سوی دیگر در بسیاری از کارخانه‌هایی که محصولاتی متشکل از چند قطعه تولید می‌کنند، حالت مونتاژ وجود دارد. تولید قطعات مختلف در هر ایستگاهی مستلزم آماده‌سازی دستگاه‌ها می‌باشد. این مقاله به بررسی مساله زمان‌بندی تولید کارگاهی انعطاف‌پذیر در حالت مونتاژ و زمان‌های پردازش وابسته به‌توالی برای کمینه نمودن مجموع زمان تکمیل محصولات مختلف می‌پردازد. این مساله تاکنون در ادبیات موضوع موردبررسی قرار نگرفته است.

نوآوری‌های این مقاله شامل موارد زیر می‌باشد:

- در نظر گرفتن مساله زمان‌بندی تولید کارگاهی انعطاف‌پذیر در حالت مونتاژ و زمان‌های پردازش وابسته به‌توالی
- ارایه مدل ریاضی مساله
- ارایه سه الگوریتم برای حل مساله شامل الگوریتم‌های ژنتیک (GA)[3]، شبیه‌سازی تبرید (SA)[4] و یک الگوریتم ترکیبی به نام GA-SA که دو الگوریتم قبلی را با یکدیگر ادغام نموده است.

در ادامه در بخش ۲ ادبیات مساله زمان‌بندی تولید کارگاهی انعطاف‌پذیر موردبررسی قرار می‌گیرد. سپس مساله و محدودیت‌های آن در بخش ۳ تعریف شده و مدل ریاضی مساله ارایه می‌شود. در بخش ۴ الگوریتم‌های حل مساله شرح داده می‌شود. در بخش ۵ به ارزیابی کارایی الگوریتم‌های پیشنهادی و مقایسه آن‌ها با دیگر پرداخته می‌شود. در انتها نیز در بخش ۶ نتیجه‌گیری و زمینه‌های تحقیقات آتی تبیین می‌شود.

۲ مرور ادبیات

تاکنون تحقیقات متعددی در زمینه مساله زمان‌بندی تولید در محیط کارگاهی منعطف صورت پذیرفته است.

برندیمارت[۱] با استفاده از الگوریتم جستجوی ممنوع[5] به تعیین نحوه تخصیص عملیات به ماشین‌ها و تعیین توالی عملیات پرداخته است. چویی و چویی[۲] یک الگوریتم جستجوی محلی برای مسایل زمان‌بندی تولید کارگاهی ارایه کردند. آن‌ها علاوه بر اینکه برای هر عملیات امکان وجود یک عملیات جانشین را در نظر گرفتند، زمان راه‌اندازی وابسته به عملیات قبلی را نیز لحاظ کرده‌اند.

[1] Job Shop Scheduling
[2] Flexible Job Shop scheduling
[3] Genetic Algorithm
[4] Simulated Annealing Algorithm
[5] Tabu search

ژیا و وو[۳] یک نگرش بهینه‌سازی ترکیبی برای مسایل زمان‌بندی تولید کارگاهی انعطاف‌پذیر ارایه کردند. در این تحقیق از ترکیب دو الگوریتم بهینه‌سازی ازدحام ذرات (PSO)[1] برای تخصیص و شبیه‌سازی تبرید برای تعیین توالی استفاده‌شده است. روسی و دینی[۴] یک بهینه‌سازی کلونی مورچه مبتنی بر نرم‌افزار سیستم ارایه نمودند که محدودیت‌های مسیریابی منعطف، زمان حمل‌ونقل و راه‌اندازی وابسته به‌توالی نیز در نظر گرفته و عملگرهای کلونی مورچه را برای بهبود کیفیت راه‌حل در نظر می‌گیرند.

فتاحی و همکاران[۵] با رویکرد نوین اشتراک عملیات[2] به حل مساله پرداخته‌اند. در بسیاری از تولید کارگاهی انعطاف‌پذیر، تقاضای مشتری می‌تواند برای هر کار آزاد شود به‌طوری‌که تقاضا مقدار پایان هر یک از کارها را به سفارش مشتری تعیین می‌کند. در این مدل هر کار ممکن است یک یا چند تقاضا داشته باشد. به دلیل اینکه این مسایل NP-hard هستند، یک رویکرد سلسله مراتبی که از الگوریتم شبیه‌سازی باز پخت استفاده می‌کند برای حل مسایل بزرگ ارایه‌شده است. در اضافه، یک متد برنامه‌ریزی خطی یکپارچه ترکیبی نیز ارایه شده است.

گاوو و همکاران[۶] برای مسایل زمان‌بندی تولید کارگاهی انعطاف‌پذیر از ترکیب الگوریتم ژنتیک با یک روش جستجوی محلی خلاقانه انتقال گلوگاه[3] استفاده نمودند. تابع هدف در این مقاله از سه هدف کمینه کردن زمان تکمیل آخرین سفارش، کمینه ماکزیمم جریان تشکیل می‌شود. گاوو و همکاران[۷] یک الگوریتم ژنتیک ترکیبی را برای مساله زمان‌بندی تولید کارگاهی انعطاف‌پذیر توسعه دادند. زینگ و همکارانش[۸] روش حل ارایه‌شده توسط ژیا و وو[۳] را به‌صورت تجربی اجرا نموده و با اصلاح مدل ریاضی آن قابلیت اجرایی بهتری به این مقاله داده‌اند.

یزدانی و همکاران[۹] الگوریتم جستجوی همسایگی متغیر موازی را برای حل مساله زمان‌بندی تولید کارگاهی انعطاف‌پذیر برای کمینه کردن حداکثر زمان تکمیل آخرین سفارش ارایه دادند. باقری و همکاران [۱۰] الگوریتم سیستم ایمنی مصنوعی(AIA)[4] بر اساس رویکرد یکپارچه ارایه داده است. این الگوریتم چندین استراتژی برای تولید جمعیت اولیه و انتخاب اشخاص برای ترکیب دارد. چندین عملگر جهش نیز برای ترکیب اشخاص جدید ارایه شده است.

باقری و زندیه[۱۱] در مقاله خود زمان آماده‌سازی وابسته به‌توالی را برای به حداقل رساندن زمان تکمیل آخرین سفارش و متوسط تأخیری در محیط‌های زمان‌بندی تولید کارگاهی انعطاف‌پذیر در نظر گرفته‌اند. چن و همکارانش[۱۲] بر اساس الگوریتم ژنتیک و گروه‌بندی الگوریتم ژنتیک، یک برنامه زمان‌بندی برای مساله تولید کارگاهی و برنامه‌ریزی کار با ماشین‌های موازی ارایه کرده‌اند که تابع هدف آن مینیمم کردن هزینه تحویل، تأخیر و در مجموع کل زمان بیکاری ماشین و زمان اتمام کار تعریف شده است.

شن و باسچر[۱۳] مساله JSP با تابع هدف حداقل کردن زمان اتمام کار را ارایه نموده است. محصولاتی را که دارای زمان راه‌اندازی وابسته به هم هستند در یک دسته قرار داده و از الگوریتم جستجوی ممنوع برای حل

[1] Particle Swarm Optimization
[2] overlapping
[3] bottleneck shifting
[4] Artificial Immune Algorithm

استفاده کرده است. موزیاه و ماج کومر[۱۴] دو الگوریتم کلونی مورچگان[1] و ژنتیک را برای حل مساله پیشنهاد داده‌اند و آن‌ها را با یکدیگر مقایسه نموده‌اند. شن[۱۵] مساله را با فرض وجود زمان تنظیم ماشین‌آلات مورد بررسی قرار داده و الگوریتم جستجوی ممنوعی طراحی نموده است که با استفاده از گراف‌ها و اولویت حرکات داخلی درون گراف، نسبت به حرکات خارجی آن توانستند جواب بهتری نسبت به الگوریتم شبیه‌سازی تبرید به دست آورد.

چن و لی[۱۶] نیز با بهره گرفتن از روش دسته‌بندی با تابع هدف کمینه نمودن میزان تأخیر، مساله را حل نموده‌اند. دسته‌بندی تعریف شده برمبنای موعد زمان اتمام کار حل انجام تعریف شده است. پن و همکارانش[۱۷] به تعریف جدیدی در دسته‌بندی عملیات پرداختند. در این روش، دسته‌ها در مکان‌هایی که تجمع کار بیش‌تر است تعریف شده و با حداقل کردن این دسته‌ها در سیستم انتخاب، توانستند سیستم را بهبود ببخشند.

یاوو و شن[۱۸] مساله را با توابع هدف کارایی و انطباق سریع با محیط جدید اجرا نموده‌اندکه الگوریتم ارایه‌شده برمبنای روش واکنش مبتکرانه[2] در مقایسه با روش‌های قبلی دارای جواب‌های بهتر و همچنین زمان کم‌تری نیز می‌باشد. ابدلمگوید[۱۹] مساله FJSP را با استفاده از الگوریتم جستجوی ممنوع و جستجوی محلی ابتکاری حل نموده است. لین[۲۰] و پلاسیوس و همکاران وی[۲۱] مساله FJSP را با شرایط زمان‌های پردازش فازی مورد بررسی قرار داده‌اند.

تحقیقات صورت پذیرفته در این حوزه را می‌توان از دیدگاه‌های مختلفی تقسیم‌بندی نمود. از منظر تابع هدف موردبررسی، تحقیقات را می‌توان به دو دسته تقسیم نمود. دسته اول تحقیقاتی هستند که به‌صورت تک هدفه مساله را بررسی نموده‌اند [۱–۳، ۹]. دسته دوم تحقیقاتی هستند که به‌صورت چند هدفه مساله را مورد بررسی قرار داده‌اند. توابع هدف مورد بررسی معمولاً شامل کمینه کردن زمان تأخیر در تحویل[۱۱، ۱۶]، ماکزیمم کارایی و انطباق با محیط[۱۸]، کمینه کردن ماکزیمم جریان کاری ماشین‌آلات[۶] و کمینه کردن زمان بیکاری ماشین‌آلات[۱۲] و ... می‌باشند.

از منظر جنبه نوآوری تحقیق، مقالات را می‌توان به دو دسته تقسیم نمود. دسته اول تحقیقاتی هستند که روش‌های حل جدیدی برای مساله ارایه نموده‌اند [۲، ۶، ۷، ۹–۱۱، ۱۵]. دسته‌ی دوم شامل مقالاتی هستند که با اضافه کردن محدودیت‌ها و شرایط مختلف به مساله، مساله‌ای نزدیک‌تر به فضای حقیقی سالن‌های تولیدی را بررسی نموده‌اند. ازجمله‌ی این محدودیت‌ها می‌توان به مواردی مانند تعریف دسته‌بندی بر روی محدودیت‌ها[۴]، اشتراک‌گذاری عملیات[۵]، و ... اشاره نمود.

بیش‌ترین روش‌های حل به‌کاررفته در ادبیات موضوع الگوریتم ژنتیک[۶، ۷، ۱۲]، است.

برای بررسی جامع بر روی منابع و امکان مقایسه آن‌ها برخی مقالات فارسی، برخی از مقالاتی که در این زمینه کار شده عبارت است از:

[1] Artificial Bee Colony
[2] proactive–reactive method

فتاحی، ارکات و اصلاحی [۲۳] در این تحقیق به حل مساله زمان‌بندی سیستم‌های تولید کارگاهی انعطاف‌پذیر با فرض پویا بودن محیط تولیدی پرداخته یعنی ورود سفارش‌های جدید به سیستم در حین دوره زمان‌بندی مشخص شده است. در این تحقیق ابتدا مدل ریاضی مربوطه توسعه داده‌شده و سپس با توجه به پیچیدگی بالای مساله، یک الگوریتم تقریبی برپایه الگوریتم ژنتیک برای حل آن ارایه شده است. نتایج آزمایش‌های عددی، کارایی الگوریتم پیشنهادی در رسیدن به جواب‌های باکیفیت خوب را نشان می‌دهد.

بهشتی و قاضی‌وکیلی [۲۴] برای دستیابی به اهداف کمینه کردن ماکزیمم زمان اتمام کار و حداکثر بار کاری ماشین‌آلات از الگوریتم ژنتیک دوبخشی استفاده نموده‌اند. نتایج نشان از کارایی الگوریتم ارایه شده دارد. نصیری و میرعابدینی [۲۵] در این مقاله از یک رویکرد جدید مبتنی بر ترکیب الگوریتم رقابت استعماری و الگوریتم قورباغه جهنده با هدف کاهش زمان پردازش خاتمه کل کارها ارایه کرده‌اند. نعمتی، رفاهی و کردرستمی[۲۶] نیز به حل مساله زمان بدی ماشین‌های موازی با استفاده از الگوریتم ژنتیک پرداخته‌اند.

بررسی ادبیات موضوع نشان می‌دهد که تاکنون مساله زمان‌بندی در محیط کارگاهی منعطف در حالت مونتاژ تاکنون مورد بررسی قرار نگرفته است. در این تحقیق به بررسی مساله مذکور در حالت زمان‌های آماده‌سازی وابسته به‌توالی پرداخته می‌شود و سه الگوریتم فرا ابتکاری ژنتیک، شبیه‌سازی تبرید و یک الگوریتم ترکیبی برای حل مساله ارایه می‌شود.

۳ تعریف مساله

در این بخش ابتدا تعریف مساله بیان می‌گردد و سپس مدل ریاضی مساله ارایه می‌شود.

۳-۱ فرضیات مساله

همان‌گونه که اشاره شد، این تحقیق به بررسی مساله زمان‌بندی در محیط کارگاهی منعطف در حالت مونتاژ و زمانه‌ای آماده‌سازی وابسته به‌توالی می‌پردازد. در مساله موردبررسی در این تحقیق فرض می‌شود که P محصول وجود دارند که هر یک از مونتاژ تعدادی قطعه درست می‌شوند. برای تولید هر قطعه نیز تعدادی عملیات موردنیاز است. مسیر پردازش عملیات موردنیاز برای تکمیل هر قطعه به‌منظور تکمیل شدن آن با مسیر پردازش عملیات موردنیاز برای سایر سفارش‌ها ممکن است، متفاوت باشد. جدول ۱ اطلاعات یک مساله شامل ۲ محصول را نشان می‌دهد. محصول اول از مونتاژ دو قطعه A و B و محصول دوم از مونتاژ سه قطعه D، E و F به دست می‌آیند. برای تولید هر قطعه نیز چند عملیات مختلف موردنیاز است.

جدول ۱. اطلاعات یک مساله نمونه

محصول	قطعات مورد نیاز	عملیات اول	عملیات دوم	عملیات سوم
محصول ۱	A	O_{A1}	O_{A2}	ـــــ
	B	O_{B1}	O_{B2}	$O_{\square 3}$
محصول ۲	C	$O_{\square 1}$	$O_{\square 2}$	ـــــ
	D	$O_{\square 1}$	$O_{\square 2}$	ـــــ
	E	$O_{\square 1}$	$O_{\square 2}$	$O_{\square 3}$

سایر فرضیات مساله به‌صورت زیر می‌باشد:

- مسیر پردازش قطعات مشخص هستند. هرچند این مسیر برای قطعات مختلف ممکن است متفاوت می‌باشد.
- قطعات مورد نیاز برای مونتاژ هر محصول نهایی مشخص هستند.
- زمان‌های پردازش وابسته به‌توالی هستند. به‌عبارت‌دیگر زمان پردازش هر قطعه روی یک ماشین وابسته به قطعه‌ای است که بلافاصله قبل از آن روی ماشین مورد پردازش قرارگرفته است. این امر مرتبط با زمان آماده‌سازی هر قطعه می‌شود؛ زیرا اگر نوع قطعات متفاوت باشد، زمانی جهت آماده‌سازی آن مورد نیاز است که با توجه به نوع قطعات نیز ممکن است این زمان متفاوت باشد.
- اگر حتی یک قطعه از قطعات مورد نیاز برای تولید محصول نهایی آماده نشده باشد، امکان مونتاژ محصول نهایی وجود ندارد.
- تابع هدف کمینه کردن مجموع زمان تکمیل محصولات نهایی می‌باشد.

سایر فرضیات، فرضیات رایج در مسایل زمان‌بندی هستند. در ادامه مدل ریاضی مساله ارایه می‌شود.

۳-۲ مدل ریاضی مساله

در این قسمت به بیان مدل ریاضی مساله پرداخته می‌شود. در ابتدا به تعریف پارامترها و متغیرهای به‌کاررفته در مدل ریاضی پرداخته می‌شود. شاخص‌های به کار رفته در مدل ریاضی مساله به شرح زیر می‌باشد.

w, i: شاخص قطعه

k, k': شاخص شماره عملیات

j : شاخص ماشین

p : شاخص محصول

s: شاخص اولویت پردازش روی ماشین

پارامترهای مورد استفاده در مدل نیز به‌صورت زیر هستند.

p : تعداد محصولات

n: تعداد کل قطعات

m: تعداد ماشین‌ها

k_i: تعداد عملیات قطعه i

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

A_{ikj}: یک پارامتر صفر و یک می‌باشد. اگر O_{ik} بتواند در ماشین j اجرا شود مقدار ۱ و در غیر این صورت مقدار ۰ دارد.

$\Pr_{wk'ik}$: زمان پردازش عمل O_{ik} اگر بعد از $O_{wk,}$ انجام شود.

$\Pr_{ik}$: زمان پردازش عمل O_{ik} اگر اولین عملیات روی ماشین مربوطه باشد.

$\Pr A_p$: زمان مورد نیاز برای مونتاژ محصول p

B: یک ماتریس صفر و یک با ابعاد P در N است که $B(p,i)=۱$ بیانگر این مطلب است که قطعه i زیرمجموعه محصول p است و بالعکس.

L: یک عدد بزرگ

همچنین متغیرهای تصمیم مساله عبارت‌اند از:

C_i: زمان تکمیل قطعه i

F_p: زمان تکمیل محصول p

PS_{ik}: زمان پردازش واقعی عملیات O_{ik} در فرایند زمان‌بندی

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

$$Min\ Z=\sum_{p=۱}^{P}F_p$$

s.t.

$$\sum_{j}^{m}Y_{ikj}=۱, \qquad i=۱,...,n,\quad k=۱,...,k_{i-۱}, \tag{۱}$$

$$\sum_{j}^{m}\sum_{s}^{nj}X_{ikjs}=۱, \qquad i=۱,...,n,\quad k=۱,...,k_{i-۱}, \tag{۲}$$

$$C_i \geq T_{ik}+PS_{ik}, \qquad i=۱,...,n,\quad k=۱,...,k_i, \tag{۳}$$

$$PS_{ik} \geq \Pr_{ik}-L\left(۱-X_{ikj۱}\right), \tag{۴}$$

$$PS_{ik} \le \Pr_{ik} + L\left(۱ - X_{ikj۱}\right), \qquad i = ۱,...,n, \quad k = ۱,...,k_{i-۱},$$
$$PS_{ik} \ge \Pr_{wk'ik} - L\left(۲ - X_{ikjs} - X_{wk'j(s-۱)}\right), \qquad s = ۲,...,n_j,;$$
$$PS_{ik} \le \Pr_{wk'ik} + L\left(۲ - X_{ikjs} - X_{wk'j(s-۱)}\right), \qquad i,w = ۱,...,n, \quad k,k' = ۱,...,k_{i-۱}, \tag{۵}$$

$$T_{ik} + PS_{ik} \le T_{i(k+۱)}, \qquad i = ۱,...,n, \quad k = ۱,...,k_{i-۱}, \tag{۶}$$

$$TM_{js} + PS_{ik} \le TM_{j(s+۱)} + L\left(۱ - X_{ikjs}\right), \qquad j = ۱,...,m, \quad s = ۱,...,n_{j-۱}, \tag{۷}$$

$$TM_{js} \le T_{ik} + L\left(۱ - X_{ikjs}\right), \qquad j = ۱,...,m, \quad s = ۱,...,n_j,$$
$$TM_{js} \ge T_{ik} - L\left(۱ - X_{ikjs}\right), \qquad i = ۱,...,n, \quad k = ۱,...,k_{i-۱}, \tag{۸}$$

$$Y_{ikj} \le A_{ikj}, \qquad j = ۱,...,m, \quad i = ۱,...,n, \quad k = ۱,...,k_{i-۱}, \tag{۹}$$

$$\sum_{s}^{nj} X_{ikjs} = Y_{ikj}, \qquad i = ۱,...,n, \quad k = ۱,...,k_{i-۱}, \quad j = ۱,...,m, \tag{۱۰}$$

$$F_p \ge C_i + \Pr A_p, \qquad p = ۱,...,P, \quad i = ۱,...,n \mid B(p,i) = ۱, \tag{۱۱}$$

$$C_i, F_p, PS_{ik}, T_{ik}, TM_{js} \ge ۰, \qquad Y_{ikj}, X_{ikjs} \in \{۰,۱\}. \tag{۱۲}$$

تابع هدف کمینه کردن مجموع زمان تکمیل محصولات نهایی را نشان می‌دهد. محدودیت ۱ تعیین می‌کند که هر عملیات باید فقط به یک ماشین تخصیص یابد. محدودیت ۲ تعیین می‌کند که هر عملیات باید در یک اولویت مشخص از ماشین تخصیص‌یافته به آن، پردازش شود. محدودیت ۳ زمان تکمیل قطعه را با توجه به زمان شروع عملیات مرتبط با آن تعیین می‌کند. محدودیت ۴ زمان پردازش واقعی عملیات O_{ik} درصورتی‌که اولین اولویت پردازش روی ماشین را داشته باشد و محدودیت ۵ زمان پردازش واقعی هر عملیات را درصورتی‌که به اولویت های دیگر ماشین اختصاص‌یافته باشند (با توجه به سفارش‌های قبلی) تعیین می‌کند. محدودیت ۶ زمان شروع هر عملیات را با توجه به زمان شروع عملیات پیش‌نیازی آن تعیین می‌کند. محدودیت ۷ بیان می‌دارد که هر ماشین فقط در یک‌لحظه یک عمل را پردازش کند. محدودیت ۸ زمان شروع فعالیت‌های تخصیص یافته به هر ماشین را تعیین می‌کند. محدودیت ۹ مجاز بودن تخصیص هر عملیات به هر ماشین را تعیین می‌کند. محدودیت ۱۰ بیان‌کننده این امر است که اگر عملیاتی به یک ماشین تخصیص نیابد، به هیچ‌یک از اولویت‌های پردازش آن ماشین نیز تخصیص نخواهد یافت. محدودیت ۱۱ زمان تکمیل هر محصول را با توجه به زمان تکمیل شدن قطعات مرتبط با آن تعیین می‌کند. محدودیت ۱۲ نیز علامت و نوع متغیرها را تعریف می‌کند.

۴ روش حل

حالت خاصی از مساله موردبررسی در این تحقیق که هر محصول از یک قطعه تشکیل‌شده باشد و زمان پردازش هر عملیات به ازای هر نوع عملیات قبل از آن یکسان در نظر گرفته شود، یک مساله FJSP معمولی است. با توجه به این‌که مساله FJSP جز مسایل Np-hard محسوب می‌شود، بنابراین مساله موردبررسی در این مقاله نیز از نوع Np-hard خواهد بود و برای حل آن باید از الگوریتم‌های ابتکاری و یا فرا ابتکاری استفاده نمود. در این بخش

سه الگوریتم شامل الگوریتم‌های ژنتیک، شبیه‌سازی تبرید و یک الگوریتم ترکیبی که دو الگوریتم قبلی را با یکدیگر ادغام نموده است به نام GA-SA برای حل مساله پیشنهاد می‌شود.

۴-۱ الگوریتم ژنتیک

الگوریتم ژنتیک، روشی برای یافتن راه‌حل تقریبی برای مسایل بهینه‌سازی است که زیرمجموعه‌ای از الگوریتم‌های تکاملی به شمار می‌آید. این الگوریتم از عملگرهای تلفیق، جهش و انتخاب جهت جستجوی فضای جواب مساله استفاده می‌کند.

مشخصه‌های الگوریتم ژنتیک استفاده‌شده در این مقاله به‌قرار ذیل می‌باشد:

ساختار کرومزوم: کارایی مؤثر جستجوی فضای جواب توسط الگوریتم ژنتیک وابسته به تعریف ساختار کرومزوم مناسب جهت تبیین هر یک از جواب‌ها و منطبق شدن آن با عملگرهای الگوریتم است. در این مقاله ساختاری نوین از کروموزم ارایه‌شده است. ساختارکروموزم پیشنهادی از دو رشته تشکیل‌شده است. رشته اول مربوط به‌توالی عملیات و رشته دوم مربوط به تخصیص ماشین به هر عملیات است.

عناصر رشته اول شامل شماره قطعات مورد نیاز جهت تکمیل محصولات می‌شود. در این رشته هر قطعه به تعداد عملیات مورد نیاز جهت تکمیل خود تکرار می‌شود. این رشته تعیین کننده اولویتی پردازش عملیات موردنیاز جهت تکمیل محصولات است. عناصر رشته دوم شامل شماره ماشین‌هایی است که عملیات متناظر در رشته اول باید توسط آن‌ها پردازش شوند. برای تبیین بیش‌تر مثال زیر بیان می‌شود.

مثال ۱- فرض کنید یک محصول وجود دارد که از دو قطعه تشکیل شده است. این قطعات باید توسط سه ماشین مورد پردازش قرار گیرند. قطعه اول شامل سه عملیات و قطعه دوم شامل دو عملیات است. پس در مجموع ۵ عملیات وجود دارد. نحوه‌ی کدگذاری این مساله به این صورت می‌باشد که دو رشته به طول تعداد کل عملیات (در این مثال ۵ عملیات) ایجاد می‌شود.

شکل ۱ یک کرومزوم موجه برای این مساله را نشان می‌دهد. در رشته اول (مرتبط یا توالی عملیات)، مقدار آرایه اول عدد ۱ می‌باشد. این نشان‌دهنده آن است که اولین عمل از قطعه اول بالاترین اولویت پردازش را داراست. مقدار آرایه دوم عدد ۲ است که این نشان‌دهنده آن است که اولین عملیات از قطعه دوم دارای دومین اولویت پردازش است. سپس مقدار آرایه سوم عدد ۱ است که بیانگر دومین عمل از قطعه اول می‌باشد. مقدار آرایه چهارم نیز عدد ۱ می‌باشد که بیانگر سومین عملیات قطعه اول است. مقدار آرایه پنجم عدد ۲ می‌باشد که دومین عمل از قطعه دوم را نشان می‌دهد.

در رشته دوم کروموزوم (تخصیص ماشین) مقدار آرایه اول برابر عدد ۲ است. این به این معناست که عملیات متناظر با آن (یعنی عملیات اول از قطعه اول) باید توسط ماشین ۲ پردازش شود. مقدار آرایه دوم رشته دوم نیز ۲ است که نشان‌دهنده ماشین اختصاصی به عمل اول از قطعه دوم است. به همین ترتیب مقدار آرایه سوم نشان‌دهنده ماشین اختصاص‌یافته به عمل دوم قطعه اول می‌باشد و ... .

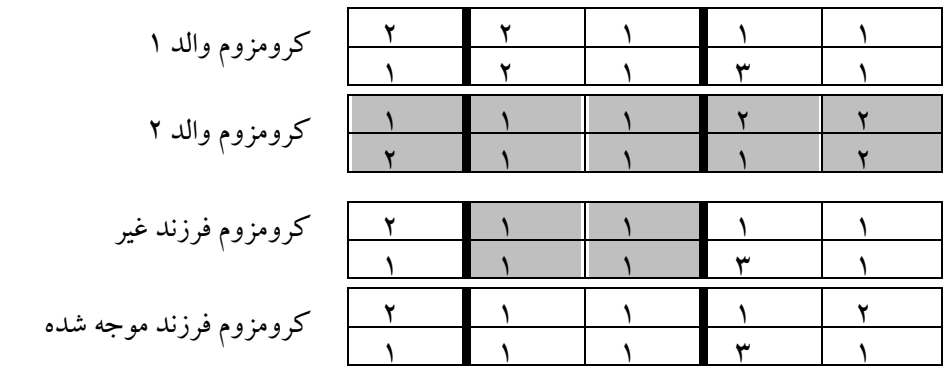
پس از تعیین تخصیص عملیات به ماشین‌ها هر یک از آن‌ها روی ماشین مربوطه پردازش می‌شوند. اولویت پردازش سفارش‌هایی که به یک ماشین تخصیص‌یافته‌اند نیز توسط رشته اول تعیین می‌شود. پس از تعیین تخصیص و توالی سفارش‌ها به ماشین‌ها به زمان‌بندی کرومزوم مربوطه پرداخته می‌شود و مقدار تابع هدف متناظر با کرومزوم مربوطه محاسبه می‌شود.

۱	۲	۱	۱	۲
۲	۲	۱	۳	۱

شکل ۱. نمایش کروموزوم

عملگر تقاطع: عملگر تقاطع مورداستفاده در این تحقیق از نوعِ دو نقطه برش می‌باشد. در حین انجام عمل تقاطع ممکن است کرومزوم‌های فرزند غیرموجه ایجاد شوند که توسط یک‌رویه اصلاحی، فرزندهای ایجادشده تبدیل به کرومزوم‌های موجه می‌شوند. ایجاد کرومزوم‌های غیرموجه تنها مربوط به رشته اول کرومزوم می‌باشند. برای تبیین بیش تر مثال زیر ارایه می‌شود.

مثال ۲- شکل ۲ را در نظر بگیرید. این شکل دو کرومزوم والد موجه برای مثال ۱ را نشان می‌دهد. همچنین نقاط برش نیز در این شکل نشان داده شده است. پس از انجام عمل تقاطع، کرومزوم فرزندی ایجاد می‌شود که در سطر اول آن ۴ عدد ۱ وجود دارد. این در صورتی است که قطعه ۱ تنها دارای ۳ عملیات می‌باشد. تخصیص اضافی عملیات برای هر قطعه معادل تخصیص کم‌تر عملیات برای قطعه‌های دیگر است. رویه اصلاحی به این صورت عمل می‌کند که ابتدا فهرستی از عملیاتی که کم‌تر از میزان موردنیاز تخصیص داده شده است تشکیل می‌شود که آن را مجموعه S می‌نامیم. سپس از ابتدای سطر اول کرومزوم شروع کرده و تعداد عملیات هر قطعه شمارش می‌شوند. به محض اینکه تعداد عملیات تخصیص داده شده به هر قطعه بیش از عملیات مورد نیاز قطعه شد، عملیات مذکور از کرومزوم حذف شده و یک عملیات به‌صورت تصادفی از مجموعه S انتخاب شده و جایگزین عملیات حذف‌شده از کرومزوم می‌شود. سپس مجموعه S به‌روز می‌گردد.



شکل ۲. عمل تقاطع و رویه اصلاحی

عملگر جهش: برای انجام عمل جهش از روش جابجایی[1] استفاده‌شده است. از اضافه شدن کرومزوم‌های جدید ایجاد شده توسط عملگرهای جهش و تقاطع باید اندازه جمعیت به حالت اولیه خود بازگردد. برای باید تعدادی از کرومزوم‌های فعلی برای نسل بعد انتخاب شوند.

عملگر انتخاب برای انتقال کرومزوم‌ها به نسل بعد از سه روش نخبه‌گرایی، چرخ رولت و انتخاب تصادفی استفاده‌شده است. به این معنی که ۲۰٪ از کرومزوم‌های نسل بعد از بهترین کرومزوم‌های نسل قبل انتخاب می‌شوند. ۴۰٪ با استفاده از روش چرخ رولت و ۴۰٪ نیز به‌صورت تصادفی از نسل قبل انتخاب می‌شوند.

معیار توقف: معیار توقف در این الگوریتم به این صورت است که اگر ۱۰ تکرار متوالی بهترین کرومزوم تغییری پیدا نکرد یا تعداد تکرارها بیش‌تر از ۲۰۰ شد، الگوریتم خاتمه پیدا می‌کند.

پس از اجراهای متعدد و به‌صورت تجربی مشخص گردید مقادیر ۲۰۰ برای جمعیت اولیه، ۰/۵ برای نرخ تلفیق و ۰/۵ برای نرخ جهش منجر به ایجاد جواب‌های نسبتاً مناسب در زمان حل معقول می‌گردد.

۴-۲ الگوریتم شبیه‌سازی تبرید

الگوریتم شبیه‌سازی تبرید، یک الگوریتم بهینه‌سازی فراابتکاری اثربخش در حل مسایل بهینه‌سازی است. منشأ الگوریتم شبیه‌سازی تبرید شبیه‌سازی‌شده، کارهای کریک پاتریک و همکاران[۲۲] است. آن‌ها برای حل مسایل سخت بهینه‌سازی، روشی مبتنی بر تکنیک انجماد تدریجی پیشنهاد نمودند. تکنیک انجماد تدریجی، روشی برای رسیدن به حالتی است که در آن انرژی ماده جامد، به‌خوبی و به‌طور یکنواخت کمینه شده باشد. این تکنیک شامل قرار دادن ماده در دمای بالا و سپس کم کردن تدریجی این دماست.

گام‌های الگوریتم شبیه‌سازی تبرید توسعه داده‌شده برای مساله به‌صورت زیر است:

۱- دمای اولیه برای مساله در نظر بگیرید. همچنین یک جواب اولیه به‌صورت تصادفی تولید و مقدار تابع هدف آن را محاسبه می‌گردد.

۲- گام‌های زیر k بار در دمای فعلی تکرار می‌شود:

۲-۱- یک همسایگی برای جواب فعلی تولید کنید.

۲-۲- جواب همسایگی را با جواب اولیه مقایسه کنید. در صورت برتری جواب همسایگی نسبت به جواب فعلی، آن با جواب فعلی جایگزین کنید و در غیر یک عدد تصادفی بین صفر و یک تولید کنید. درصورتی‌که عدد تصادفی تولیدشده از مقدار تابع احتمال پذیرش جواب‌های بد کم‌تر بود، جواب همسایگی جایگزین جواب فعلی شده و در غیر این صورت تغییری در جواب فعلی داده نمی‌شود.

۳- با استفاده از تابع زیر دما را کاهش دهید:

$$T_{new} = \alpha T_{old}$$

۴- درصورتی‌که به دمای انجماد رسیده‌اید، الگوریتم را خاتمه دهید. در غیر این صورت به گام ۲ بروید.

[1] Swap

در این الگوریتم نیز ساختار رمزنگاری جواب‌ها مشابه الگوریتم ژنتیک در نظر گرفته‌شده است. پس از اجراهای متعدد و به‌صورت تجربی مشخص گردید مقادیر ۳۰۰ برای دمای اولیه، ۵ برای پارامتر k، ۰/۹۹ برای پارامتر α و ۱ برای دمای انجماد منجر به جواب‌های نسبتاً خوب در زمان حلی معقول می‌گردد.

همچنین تابع احتمال پذیرش جواب‌های بد به‌صورت $e^{-\frac{\Delta f}{T}}$ در نظر گرفت شده است که Δf اختلاف جواب فعلی با بهترین همسایگی و T درجه حرارت فعلی می‌باشد.

۴-۳ الگوریتم ترکیبی GA-SA

به‌منظور حل مساله، یک الگوریتم ترکیبی به نام GA-SA که دو الگوریتم قبلی را با یکدیگر ادغام نموده است نیز ارایه شده است. الگوریتم پیشنهادی در ابتدا مانند الگوریتم ژنتیک عمل می‌کند؛ اما درصورتی‌که بهترین کرومزوم یافت شده طی ۵ نسل آخر تغییری پیدا نکرد یا الگوریتم ژنتیک ۱۰ بار تکرار شد، (نصف شرایط خاتمه الگوریتم ژنتیک) الگوریتم شبیه‌سازی تبرید فعال می‌شود. بدین منظور، بهترین جواب حاصله از الگوریتم ژنتیک به‌عنوان جواب اول الگوریتم تبرید انتخاب شده و با همان مشخصات الگوریتم شبیه‌سازی تبرید که در بخش قبل بیان شده است، سایر محاسبات انجام می‌گیرد. الگوریتم اجرایی الگوریتم پیشنهادی به‌قرار ذیل می‌باشد:

گام ۱- جمعیت اولیه‌ای از کرومزوم‌ها انتخاب کنید.

گام ۲- با استفاده از عملگرهای جهش و تلفیق نسل فعلی را افزایش دهید.

گام ۳- اگر ۵ تکرار متوالی است که بهترین کرومزوم تغییری پیدا نکرده است و یا تعداد تکرارها بیش‌تر از ۱۰۰ شده است (نصف شرایط خاتمه توسط الگوریتم ژنتیک) بهترین کرومزوم نسل جاری را به‌عنوان جواب ورودی الگوریتم شبیه‌سازی تبرید در نظر گرفته و به گام ۴ بروید. در غیر این صورت بر اساس عملگر انتخاب کرومزوم‌هایی را برای رفتن به نسل بعد انتخاب کنید و به گام ۲ بازگردید.

گام ۴- با استفاده از الگوریتم شبیه‌سازی تبرید شرح داده شده در بخش ۴-۲ جواب فعلی را بهبود دهید.

ساختار کرومزوم، تمامی عملگرها و مقادیر پارامترها مشابه دو الگوریتم قبلی است با این تفاوت که شرط رفتن از الگوریتم ژنتیک به الگوریتم شبیه‌سازی تبرید نصف شرایط خاتمه الگوریتم ژنتیک در نظر گرفته شده است.

۵ آزمایش‌های عددی

در این بخش به بررسی و مقایسه عملکرد هر سه الگوریتم پیشنهادی پرداخته می‌شود. به این منظور طیف گسترده‌ای از مثال‌های تصادفی ایجاد و توسط هر سه الگوریتم حل‌شده‌اند. در ادامه نحوه تولید مثال‌های تصادفی و نتایج به‌دست‌آمده توسط الگوریتم‌های پیشنهادی مورد مقایسه قرار می‌گیرد.

۵-۱ تولید داده‌های اولیه

مساله موردبررسی دارای پارامترهای متعددی است که ما ۴ پارامتر را برای ایجاد مسایل تصادفی موردبررسی قرار داده‌ایم. این پارامترها عبارت است از:

۱) تعداد محصولات، ۲) تعداد قطعات ، ۳) تعداد مراحل کاری، ۴) تعداد ماشین در هر مرحله

برای هر یک از پارامترهای بیان‌شده، سه سطح کوچک، متوسط و بزرگ در نظر گرفته‌شده است. پارامترهای تعداد سفارش و قطعات از به‌صورت قطعی و پارامترهای تعداد عملیات و تعداد ماشین‌آلات از توزیع یکنواخت انتخاب‌شده‌اند. جدول ۲ مقدار سطوح مختلف را نشان می‌دهد. با توجه به سطوح نشان داده‌شده در جدول ۲، در مجموع ۸۱ (۳*۳*۳*۳) مساله تصادفی ایجاد شده است. کلیه برنامه‌های کامپیوتری این مقاله توسط زبان برنامه‌نویسی Matlab نوشته شده و توسط یک رایانه با مشخصات intel Core i3, 3.1 GHZ اجرا گردیده است. در ادامه به بررسی نتایج محاسباتی الگوریتم‌ها می‌پردازیم.

جدول ۲. پارامترها و سطوح مورد بررسی

پارامتر	سطح کوچک	سطح متوسط	سطح بزرگ
تعداد محصولات	۵	۱۰	۱۵
تعداد قطعات	۵	۱۰	۵۰
تعداد عملیات	U{۱ ۵}	U{۵ ۱۰}	U{۱۰ ۱۵}
تعداد ماشین‌آلات	U{۱ ۵}	U{۵ ۱۰}	U{۱۰ ۱۵}

۵-۲ نتایج محاسباتی

نتایج به‌دست‌آمده از اجرای الگوریتم‌ها در جداول ۳ و ۴ آورده شده است. جدول شماره ۳ شامل میانگین جواب‌ها و زمانه‌ای حل به‌دست‌آمده از هر الگوریتم است. جدول شماره ۴ نیز اطلاعاتی در مورد تعداد دفعاتی که هر یک از الگوریتم‌ها نسبت به دیگری جواب بهتر (NBR[1]) و یا بدتری (NWR[2]) داده است را ارایه می‌دهد. همان‌گونه که در جدول ۳ مشخص‌شده است، ترکیب دو الگوریتم از هریک از آن‌ها به‌صورت جداگانه جواب بهتری را ارایه می‌دهد. همچنین الگوریتم ژنتیک نیز از الگوریتم شبیه‌سازی تبرید نتایج بهتری را ارایه می‌دهد.

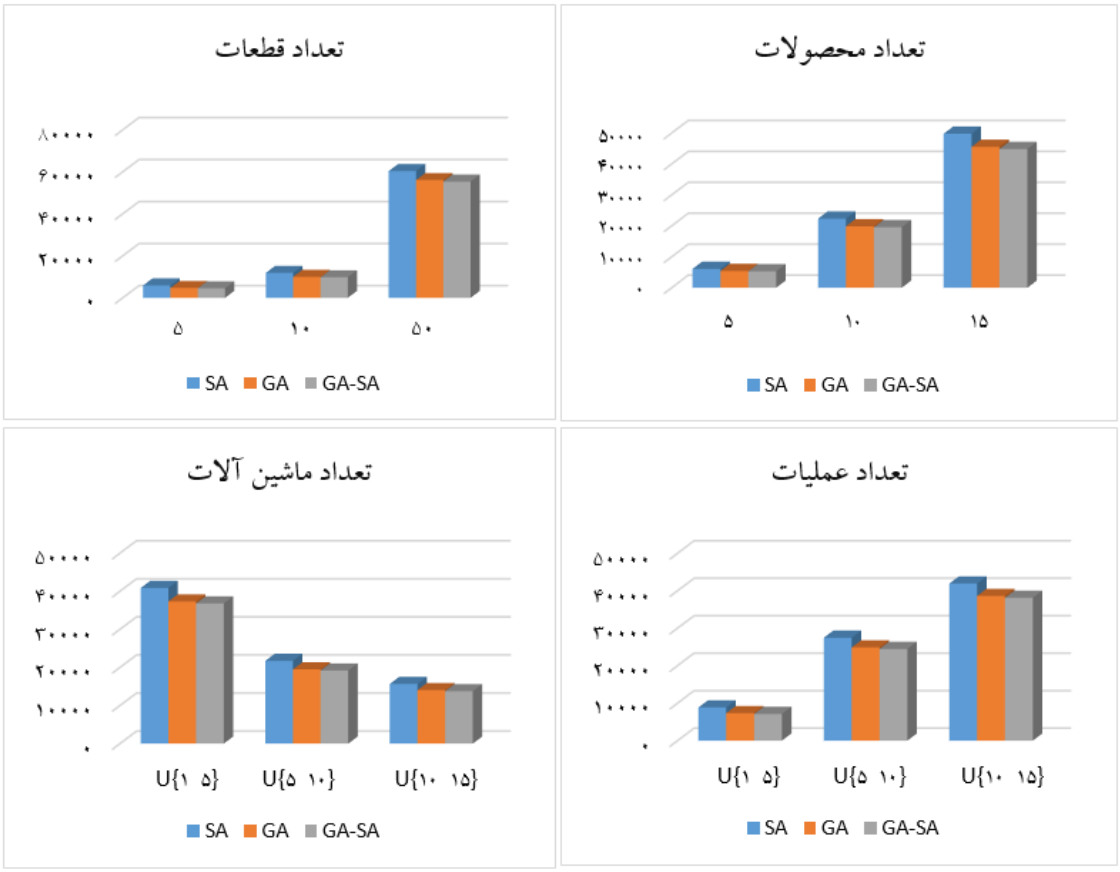
همچنین نتایج جدول ۳ نشان می‌دهد که با افزایش تعداد محصولات، تعداد قطعات و تعداد عملیات میانگین جواب‌ها و زمان‌های حل افزایش می‌یابد؛ولی با افزایش تعداد ماشین‌آلات میانگین جواب‌ها کم‌تر می‌شود.

همچنین داده‌های موجود در جدول شماره ۴، نشان می‌دهد که الگوریتم تبرید، در هیچ‌کدام از ۸۱ مساله نتوانسته است جواب بهتری را از الگوریتم ترکیبی ارایه دهد. البته با توجه به ساختار این الگوریتم و وابستگی زیاد آن به جواب اولیه، این نتیجه دور از واقعیت نمی‌باشد.

[1] Number of Better Results
[2] Number of Equivalent Results

جدول ۳. مقایسه نتایج الگوریتم‌ها

مشخصه مورد مقایسه	سطوح	میانگین جواب‌ها			میانگین زمان‌های حل		
		SA	GA	GA-SA	SA	GA	GA-SA
تعداد محصولات	۵	۶۱۵۰/۰۰	۵۴۴۳/۹۳	۵۳۳۷/۴۱	۰/۱۲	۲۴۳/۰۸	۵۱۹/۷۹
	۱۰	۲۲۳۳۹/۲۶	۱۹۸۸۴/۰۰	۱۹۵۷۲/۹۶	۰/۲۷	۹۳۵/۰۱	۱۶۱۰/۰۸
	۱۵	۴۹۸۰۰/۱۵	۴۵۵۴۲/۵۶	۴۴۸۳۷/۰۰	۰/۲۷	۳۵۱۰/۵۹	۴۱۴۸/۷۴
تعداد قطعات	۵	۵۹۳۵/۰۷	۴۷۱۴/۰۰	۴۵۵۰/۲۲	۰/۰۸	۵۵/۴۷	۱۵۶/۰۷
	۱۰	۱۱۹۳۴/۴۸	۱۰۰۳۴/۱۱	۹۸۲۰/۴۸	۰/۱۹	۱۷۲/۳۶	۳۴۶/۱۶
	۵۰	۶۰۴۱۹/۸۵	۵۶۱۲۲/۳۷	۵۵۴۱۲/۶۷	۰/۳۸	۴۴۶۱/۱۴	۵۷۷۶/۳۸
تعداد عملیات	U{۱ ۵}	۸۸۷۸/۳۷	۷۳۵۶/۱۵	۷۱۳۷/۷۸	۰/۱۰	۲۱۶/۲۸	۴۳۷/۰۶
	U{۵ ۱۰}	۲۷۴۷۵/۴۱	۲۴۸۸۸/۱۱	۲۴۴۹۶/۱۵	۰/۲۱	۱۲۷۸/۶۸	۲۱۴۴/۵۷
	U{۱۰ ۱۵}	۴۱۹۳۵/۶۳	۳۸۶۲۶/۲۲	۳۸۱۵۲/۴۴	۰/۳۵	۳۱۹۴/۰۲	۳۶۹۶/۹۸
تعداد ماشین‌آلات	U{۱ ۵}	۴۰۸۷۵/۱۵	۳۷۳۱۷/۴۸	۳۶۸۱۰/۹۳	۰/۱۹	۱۶۱۶/۱۶	۱۸۲۰/۷۹
	U{۵ ۱۰}	۲۱۷۳۶/۷۴	۱۹۵۱۶/۵۶	۱۹۲۰۲/۱۱	۰/۱۹	۱۱۱۲/۹۷	۲۲۴۴/۹۹
	U{۱۰ ۱۵}	۱۵۶۷۷/۵۲	۱۴۰۳۶/۴۴	۱۳۷۷۰/۳۳	۰/۲۸	۱۹۵۹/۸۴	۲۲۱۲/۸۳
تمام حالات		۲۶۰۹۶/۴۷	۲۳۶۲۳/۴۹	۲۳۲۶۱/۱۲	۰/۲۲	۱۵۶۲/۹۹	۲۰۹۲/۸۷



شکل ۳. نتایج حاصل از مقایسه میانگین جواب الگوریتم‌ها

جدول ۴ تعداد دفعات برتری جواب‌ها نسبت به یکدیگر

مشخصه مورد مقایسه	سطوح	NBR			NWR		
		GA نسبت به SA	GA-SA نسبت به SA	GA-SA نسبت به GA	GA نسبت به SA	GA-SA نسبت به SA	GA-SA نسبت به GA
تعداد محصولات	۵	۲۷	۲۷	۲۳	۰	۰	۳
	۱۰	۲۷	۲۷	۲۳	۰	۰	۴
	۱۵	۲۷	۲۷	۲۵	۰	۰	۲
تعداد قطعات	۵	۲۷	۲۷	۲۳	۰	۰	۳
	۱۰	۲۷	۲۷	۲۱	۰	۰	۶
	۵۰	۲۷	۲۷	۲۷	۰	۰	۰
تعداد عملیات	U{۱ ۵}	۲۷	۲۷	۲۵	۰	۰	۱
	U{۵ ۱۰}	۲۷	۲۷	۲۳	۰	۰	۴
	U{۱۰ ۱۵}	۲۷	۲۷	۲۳	۰	۰	۴
تعداد ماشین‌آلات	U{۱ ۵}	۲۷	۲۷	۲۴	۰	۰	۳
	U{۵ ۱۰}	۲۷	۲۷	۲۳	۰	۰	۴
	U{۱۰ ۱۵}	۲۷	۲۷	۲۴	۰	۰	۲

۶ نتیجه‌گیری

در این مقاله مساله زمان‌بندی تولید کارگاهی انعطاف‌پذیر در حالت مونتاژ و زمانه‌ای پردازش وابسته به‌توالی بررسی شد. این محیط تولیدی در بسیاری از صنایع نظیر خودروسازی، الکترونیک، شیمیایی و ... کاربرد دارد. پس از ارایه مدل ریاضی مساله با توجه به Np-Hard آن، سه الگوریتم فرا ابتکاری شامل الگوریتم‌های ژنتیک، شبیه‌سازی تبرید و یک الگوریتم ترکیبی برای حل مساله ارایه‌شده است. نتایج نشان می‌دهد که الگوریتم ترکیبی از هریک از الگوریتم‌های ژنتیک و شبیه‌سازی تبرید جواب بهتری را ارایه می‌دهد.

همچنین نتایج نشان می‌دهد که با افزایش تعداد محصولات، تعداد قطعات و تعداد عملیات میانگین جواب‌ها و زمان‌های حل افزایش می‌یابد؛ ولی با افزایش تعداد ماشین‌آلات میانگین جواب‌ها کم‌تر می‌شود.

بررسی مساله موردبررسی در این تحقیق در حالت چندهدفه می‌تواند زمینه‌ای برای تحقیقات آتی باشد. ارایه سایر الگوریتم‌های فرا ابتکاری نظیر الگوریتم زنبورعسل و مورچگان می‌تواند زمینه‌ای دیگر برای تحقیقات آتی باشد. اضافه کردن محدودیت‌های دیگر نظیر محدودیت‌های نگهداری و تعمیرات و یا در نظر گرفتن مساله در حالت غیرقطعی می‌توانند زمینه‌هایی دیگر از تحقیقات آتی باشند.

منابع

[۲۳] فتاحی، پ.، ارکات، ج.، صالحی، م.، (۱۳۸۶). زمان‌بندی سیستم‌های تولید کارگاهی انعطاف‌پذیر همراه بازمان راه‌اندازی وابسته به خانواده کارها. پنجمین کنفرانس ملی مهندسی صنایع، تهران، انجمن مهندسی صنایع ایران، دانشگاه علم و صنعت ایران.

[۲۴] بهشتی نیا، م.، قاضی وکیلی، ن.، (۱۳۹۴). ارزیابی الگوریتم‌های زمان‌بندی تولید کارگاهی انعطاف‌پذیر و مقایسه آن‌ها با الگوریتم ژنتیک دوبخشی، فصلنامه مدل‌سازی در مهندسی، ۱۳(۴۰)، ۱-۱۶.

[۲۵] نصیری، م.، میرعابدینی، س.، (۱۳۹۵). یک روش بهینه‌سازی ترکیبی فرا ابتکاری برای مساله زمان‌بندی جریان کارگاهی، سومین کنفرانس بین‌المللی علوم و مهندسی، استانبول- کشور ترکیه، موسسه مدیران ایده پرداز پایتخت ویرا.

[۲۶] نعمتی، خ.، رفاهی، ا.، کردرستمی، س.، (۱۳۹۵)، بهینه‌سازی زمان‌بندی الگوریتم‌های موازی با استفاده از الگوریتم ژنتیک، مجله تحقیق در عملیات در کاربردهای آن، ۱۳ (۲)، ۳۵-۵۲.

[1] Brandimarte, P., (1993). Routing and scheduling in a flexible job shop by tabu search. Annals of Operations Research, 41(3), 157-183.

[2] Choi, I. C., Choi, D. S., (2002). A local search algorithm for jobshop scheduling problems with alternative operations and sequence-dependent setups. Computers & Industrial Engineering, 42(1), 43-58.

[3] Xia, W., Wu, Z., (2005). An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. Computers & Industrial Engineering, 48(2), 409-425.

[4] Rossi, A., Dini, G., (2007). Flexible job-shop scheduling with routing flexibility and separable setup times using ant colony optimisation method. Robotics and Computer-Integrated Manufacturing, 23(5), 503-516.

[5] Fattahi, P., Saidi Mehrabad, M., Jolai, F., (2007). Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. Journal of Intelligent Manufacturing, 18(3), 331-342.

[6] Gao, J., Gen, M., Sun, L., Zhao, X., (2007). A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems. Computers & Industrial Engineering, 53(1), 149-162.

[7] Gao, J., Sun, L., Gen, M., (2008). A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Computers & Operations Research, 35(9), 2892-2907.

[8] Xing, L. N., Chen, Y. W., Yang, K. W., (2009). Comments on “An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems” [Comput. Ind. Eng. 48 (2005) 409–425]. Computers & Industrial Engineering, 56(4), 1735-1736.

[9] Yazdani, M., Amiri, M., Zandieh, M., (2010). Flexible job-shop scheduling with parallel variable neighborhood search algorithm. Expert Systems with Applications, 37(1), 678-687.

[10] Bagheri, A., Zandieh, M., Mahdavi, I., Yazdani, M., (2010). An artificial immune algorithm for the flexible job-shop scheduling problem. Future Gener. Comput. Syst., 26(4), 533-541.

[11] Bagheri, A., Zandieh, M., (2011). Bi-criteria flexible job-shop scheduling with sequence-dependent setup times—Variable neighborhood search approach. Journal of Manufacturing Systems, 30(1), 8-15.

[12] Chen, J. C., Wu, C. C., Chen, C. W., Chen, K. H., (2012). Flexible job shop scheduling with parallel machines using Genetic Algorithm and Grouping Genetic Algorithm. Expert Systems with Applications, 39(11), 10016-10021.

[13] Shen, L., Buscher, U., (2012). Solving the serial batching problem in job shop manufacturing systems. European Journal of Operational Research, 221(1), 14-26.

[14] Muthiah, A, Rajkumar, R., (2014). A Comparison of Artificial Bee Colony algorithm and Genetic Algorithm to Minimize the Makespan for Job Shop Scheduling. Procedia Engineering, 97(0), 1745-1754.

[15] Shen, L., (2014). A tabu search algorithm for the job shop problem with sequence dependent setup times. Computers & Industrial Engineering, 78(0), 95-106.

[16] Li, S. S., Chen, R. X., (2014). Single-machine parallel-batching scheduling with family jobs to minimize weighted number of tardy jobs. Computers & Industrial Engineering, 73(0), 5-10.

[17] Pan, J. C. H., Shih, P. H., Wu, M. H., (2015). Order batching in a pick-and-pass warehousing system with group genetic algorithm. Omega.

[18] Shen, X. N., Yao, X., (2015). Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems. Information Sciences, 298(0), 198-224.

[19] Abdelmaguid, T. F., (2015). A neighborhood search function for flexible job shop scheduling with separable sequence-dependent setup times. Applied Mathematics and Computation, 260(0), 188-203.

[20] Lin, J., (2015). A hybrid biogeography-based optimization for the fuzzy flexible job-shop scheduling problem. Knowledge-Based Systems, 78, 59-74.

[21] Palacios, J. J., González, M. A., Vela, C. R., González-Rodríguez, I., Puente, J., (2015). Genetic tabu search for the fuzzy flexible job shop problem. Computers & Operations Research, 54(0), 74-89.
[22] Kirkpatrick, S., Jr., C. D. G., Vecchi, M. P., (1983). Optimization by Simulated Annealing. sciencemag, 220(4598), 671-680.