مهندسی صنایع و مدیریت شریف
دوره‌ی ۱ - ۲۶، شماره‌ی ۲، ص. ۴۷-۵۸

# کمینه‌سازی بیشینه‌ی زمان درگردش کارها در مسئله‌ی کارکارگاهی پویای انعطاف‌پذیر با ماشین‌های موازی

**نسیم نهاوندی (استادیار)**
**محمّد عباسیان (کارشناس ارشد)**
**دانشکده‌ی فنی و مهندسی، دانشگاه تربیت مدرس**

در محیط‌های واقعی ساخت و تولید، مجموعه کارهایی که باید زمان‌بندی شوند با گذشت زمان تغییر می‌یابد که این خود بیان‌گر پویایی مسائل زمان‌بندی است. همچنین به‌منظور سازگاری سیستم‌های ساخت و تولید با نوساناتی همچون ازکارافتادگی ماشین‌آلات و ایجاد ماشین‌های گلوگاهی، انواع انعطاف‌پذیری‌های کف کارگاهی نیز در این سیستم‌ها لحاظ می‌شود. در نوشتار حاضر، برای نخستین‌بار پارامترهای انعطاف‌پذیری عملیات و انعطاف‌پذیری ناشی از ماشین‌های موازی (PM)[1] با سرعت‌های غیریکنواخت در محیط کارکارگاهی پویا (DJS)[2] و در قالب مسئله‌ی کارکارگاهی انعطاف‌پذیر با ماشین‌های موازی در محیط پویا (FDJSPM)[3]، در نظر گرفته شد. پس از مدل‌سازی مسئله، الگوریتمی مبتنی بر اصول الگوریتم ژنتیک (GA)[4] با کروموزوم‌های دوبعدی پویا، برای حل آن پیشنهاد شد. نتایج حاصل از اجرای الگوریتم پیشنهادی و مقایسه‌ی آن با یک روش فرابتکاری موجود در ادبیات، نشان‌دهنده‌ی بهبود جواب‌ها به‌میزان ۱/۳۴ درصد برای ابعاد مختلف مسئله است.

**واژگان کلیدی:** کارکارگاهی پویا (DJS)، ماشین‌های موازی (PM)، بیشترین زمان گردش قطعات، الگوریتم ژنتیک (GA).

N_nahavandi@modares.ac.ir
M_abbasian@modares.ac.ir

## مقدمه

برنامه‌ی زمان‌بندی (تخصیص منابع به وظایف) یکی از تصمیمات مهم در استفاده‌ی مطلوب از امکانات و تجهیزات به‌شمار می‌رود. تحقیقات در این زمینه را می‌توان به دو قسمت ـ محیط‌های ایستا و محیط‌های پویا ـ تقسیم کرد. اغلب روش‌های بررسی‌شده در ادبیات برای حل مسائل زمان‌بندی ایستا کاربرد دارند و رخ‌دادهای پویا مثل زمان‌های متفاوت ورود قطعات اولیه به کارگاه، زمان‌های متفاوت سفارشات جدید، کاهش سطح نیروی انسانی یا خرابی ماشین‌آلات در طول زمان لحاظ نمی‌شوند. در مسائل زمان‌بندی اگر مجموعه کارهای در دسترس برای زمان‌بندی در طول زمان تغییر نکند، سیستم ساخت و تولید را ایستا می‌نامند. در مقابل اگر در طول زمان کار جدیدی به مجموعه‌ی کارها افزوده شود، این سیستم ساخت و تولید را پویا می‌نامند.[1] به‌عبارت دیگر نمونه‌ی ایستای مسئله‌ی کارکارگاهی به وضعیتی اطلاق می‌شود که در آن همه‌ی کارها به‌طور همزمان (و در یک زمان صفر) برای اجرا دسترس‌پذیر باشند. اما در بسیاری از مسائل توالی عملیات واقعی لازم است که زمان‌های متفاوت راه‌اندازی کارها با یکدیگر وفق داده شود. مسئله‌ی زمان‌بندی در عمل پویا است و دلیل اصلی آن این است که در طول زمان برخی از پارامترهای ذکرشده تغییر می‌کند. لذا برای حل مسائل عملی، پویایی مسئله‌ی برنامه‌ی زمان‌بندی باید مورد توجه قرار گیرد.[2] ثابت شده است که بررسی مدل‌های ایستا آسان‌تر از مدل‌های پویاست و بررسی‌های گسترده‌یی نیز درمورد آنها انجام شده است.[1]

تحقیق حاضر علاوه بر در نظرگرفتن پویایی محیط ساخت و تولید (به‌سبب ورود کارها در یک زمان غیرصفر به کارگاه)، از انعطاف‌پذیری عملیات و انعطاف‌پذیری ناشی از ماشین‌های موازی نیز برخوردار است. به‌طور کلی، در مدل‌های کلاسیک مسائل کارکارگاهی، وجود تنها یک مسیر پردازش واحد برای هر کار به عدم انعطاف‌پذیری در چنین سیستم‌های تولیدی اشاره دارد که تولید محصولات در مسیرهای تولیدی مختلف و بر سایر ماشین‌های موجود در کارگاه را برای سیستم مجاز نمی‌داند. در حالی که امروزه محیط‌های تولیدی مدرن، با دوره‌ی عمر کوتاه‌و افزایش تنوع در محصولات شناخته می‌شوند. در این محیط‌ها، انعطاف‌پذیریِ تولید به‌عنوان یک سلاح رقابتی کلیدی به‌شمار می‌رود به‌طوری که امروزه داشتن چنین قابلیت‌هایی برای این سیستم‌ها تبدیل به یک مزیت رقابتی شده و سبب توانایی آنها در پاسخ‌گویی سریع به تغییرات غیر قابل پیش‌بینی، به‌عنوان عامل اصلی فشار رقابتی بازار می‌شود. به‌طور نمونه، داشتن برخی از این انعطاف‌پذیری‌های در زمینه‌ی تولید صنعتی، برای این سیستم‌های قدرت‌مند ساخت و تولید امکان سازگاری با نوساناتی مانند ازکارافتادگی

*تاریخ: دریافت ۱۳۸۷/۱۱/۷، داوری ۱۳۸۸/۱۰/۱، پذیرش ۱۳۸۸/۱۲/۲۲.*

ماشین‌ها و ایجاد ماشین‌های گلوگاهی را فراهم می‌کند و هرکدام از ماشین‌های موجود در کارگاه را به‌طور مؤثر قادر به تولید چندین نوع محصول می‌کند. امروزه در اکثر کارگاه‌های تولیدی به‌منظور رفع مشکل گلوگاهی (مثلاً به سبب زمان‌های پردازش بلندمدت برخی از قطعات و یا به‌عللی مانند خرابی ماشین‌آلات) و نیز افزایش تولید و بهبود عملکرد سیستم معمولاً چندین نسخه از هر ماشین در هر ایستگاه کاری موجود است (انعطاف‌پذیری ناشی از ماشین‌های موازی) یا ممکن است برخی از این عملیات، نه‌تنها توسط یک ایستگاه بلکه بر روی مجموعه‌یی از ایستگاه‌های حاضر در کارگاه قابل پردازش باشند (انعطاف‌پذیری ناشی از عملیات). این حالات به‌نوبه‌ی خود بر پیچیدگی این دسته از مسائل می‌افزاید به‌طوری که حتی یافتن جواب‌های بهینه‌ی تقریبی برای این مسائل نیز به‌طور چشم‌گیری پیچیده و سخت می‌شود.[۳] این دسته از مسائل در ادبیات به مسائل زمان‌بندی انعطاف‌پذیر معروف‌اند. در بررسی جامعی که در سال ۲۰۰۰ انجام شد، ابعاد این انعطاف‌پذیری‌های به ۱۵ نوع توسعه داده شد.[۴]

در نوشتار حاضر، مسئله‌ی زمان‌بندی کارگاهیِ انعطاف‌پذیر با ماشین‌های موازی در محیط کارگاهی پویا (FDJSPM) مطالعه و بررسی شده است. در این مسئله، به تعداد $n$ قطعه و $m$ ایستگاه کاری وجود دارد که علاوه بر این که تکمیل هر قطعه به مجموعه عملیات مشخصی (مانند $O$) نیازمند است، ترتیب حرکت هر قطعه در بین ایستگاه‌های کاری برای قطعات مختلف نیز متفاوت است. در تحقیق حاضر به‌منظور انطباق هرچه بیشتر مسئله با محیط‌های واقعی ساخت و تولید، محیط کارگاه «پویا» در نظرگرفته شده است و بنابراین هر یک از قطعات در یک زمان $r$ (یک زمان غیرصفر و تصادفی) وارد کارگاه می‌شوند. از سوی دیگر، به‌منظور اجتناب از ایجاد گلوگاه و افزایش نرخ تولید، علاوه بر این که هریک از ایستگاه‌های کاری قابلیت پردازش بیش از یک نوع قطعه را دارند (به‌سبب وجود انعطاف‌پذیری عملیات) هر ایستگاه کاری $M$، مرکب از مجموعه‌یی از $L$ ماشین موازی است که با سرعت‌های غیریکنواخت $S$ (و به‌سبب وجود انعطف‌پذیری ناشی از وجود پردازنده‌های موازی با سرعت‌های غیریکنواخت در هر ایستگاه کاری) از توانایی پردازش کار تخصیص داده شده به آن ایستگاه کاری برخوردارند.

هدف این نوشتار، مدل‌سازی و حل مسئله‌ی کمینه‌سازی بیشترین زمان گردش کاری قطعات ($F_{Max}$) با در نظرگرفتن پویایی ایجادشده در کارگاه، انعطاف‌پذیری عملیات و انعطاف‌پذیری ناشی از ماشین‌های موازی با سرعت‌های غیریکنواخت است. در این نوشتار، برای نخستین‌بار دو پارامتر «انعطاف‌پذیری عملیات» و «انعطاف‌پذیری ناشی از ماشین‌های موازی با سرعت‌های غیریکنواخت» برای محیط کارگاهی پویا، تواماً در نظرگرفته شد.

براین اساس، مسئله‌ی FDJSPM به دو زیرمسئله‌ی تخصیص و تعیین توالی عملیات تجزیه می‌شود. زیرمسئله‌ی تخصیص ناشی از انعطاف‌پذیری ایجادشده در مسئله‌ی تولید کارگاهی است و شدیداً بر پیچیدگی مسئله می‌افزاید.

نوشتار حاضر در قالب هشت بخش تهیه شده است. در این بخش به ذکر مقدمه و ضرورت تحقیق پرداخته شد. در بخش دوم ادبیات مسئله و پیشینه‌ی تحقیق در دو زمینه‌ی کارگاهی انعطاف‌پذیر پویا و ایستا ارائه می‌شود. در بخش سوم مسئله‌ی تحقیق تعریف شده و مدل‌سازی ریاضی آن بیان می‌شود. در بخش چهارم رویکرد حل مسئله بیان، و روش حل مسئله با استفاده از الگوریتم ژنتیک ارائه خواهد شد. طراحی آزمایش‌های عددی در بخش پنجم بیان شده و سپس نتایج حاصل از آن به‌منظور نشان‌دادن کارایی روش پیشنهادی در بخش ششم بررسی می‌شود. در بخش هفتم نتایج و دستاوردهای تحقیق ذکر شده و زمینه‌های مناسب برای انجام تحقیقات آتی پیشنهاد خواهد شد. در ادامه، مرور ادبیات و بررسی پیشینه‌ی تحقیق، در زمینه‌های «کار کارگاهی»، «کار کارگاهی انعطاف‌پذیر» و «مسائل زمان‌بندی با ماشین‌های موازی با سرعت‌های یکسان و غیریکسان» آورده می‌شود.

## مرور ادبیات و پیشینه‌ی تحقیق

زمان‌بندی کار کارگاهی (JS)[۵] کلاسیک و کار کارگاهی پویا (DJS) از مهم‌ترین مباحث مدیریت تولید، و جزء سخت‌ترین مباحث بهینه‌سازی ترکیبی به‌شمار می‌روند. به‌طور کلی تحقیقات انجام‌شده درباره‌ی برنامه‌ی زمان‌بندی در محیط‌های پویا یا براساس نظریه‌ی صف و یا براساس تکنیک افق برنامه‌ریزی زمانی رولی[۶] صورت گرفته است. در حالت اول مسئله‌ی زمان‌بندی به‌صورت یک سیستم صف در نظر گرفته می‌شود. در این حالت سفارشات جدید پس از ورود براساس فرایند مورد نظر روی دستگاه‌ها رفته و براساس اولویت‌بندی کار روی آنها انجام می‌پذیرد. روش‌های تحلیلی حل این مسئله براساس نظریه‌ی صف است.[۵و۶] با توجه به پیچیدگی مسئله، روش‌های تحلیلی فوق‌الذکر غالباً مسائل دارای یک ماشین را حل می‌کنند. تکنیک افق برنامه‌ریزی زمانی رولی روشی کارا در ارائه‌ی برنامه‌های زمانی برای مسائل پویاست که در آن مسئله‌ی زمان‌بندی پویا به چند مسئله‌ی زمان‌بندی ایستا تبدیل می‌شود چنان که در هر مرحله مسئله به‌صورت ایستا حل شود. این روش اولین‌بار به‌منظور حل مسئله‌ی برنامه‌ریزی میان‌مدت به کار گرفته شد.[۱] سپس به‌منظور بررسی مسئله‌ی زمان‌بندی مورد استفاده قرار گرفت و نتایج قابل توجهی در این باره ارائه شد.[۷] بعدها محققین از راه‌حلی عمومی به‌منظور استفاده از این روش در برنامه‌ی زمان‌بندی به طرف عقب استفاده کردند.[۸] اخیراً با استفاده از روش RTH مسائل برنامه‌ی زمان‌بندی در محیط پویا و نامعین نیز بررسی، و عملکرد روش‌های فوق در شرایط مختلف ارزیابی شده است.

در مسئله‌ی کار کارگاهی (JS) کلاسیک، مسیر کارها ثابت و مشخص است و وجود مسیر یکسان برای همه‌ی کارها ضرورت ندارد. در این مسئله فرض بر این است که تنها یک مسیر پردازش برای هرکار وجود دارد که این به عدم انعطاف‌پذیری در سیستم تولید اشاره دارد. کار کارگاهی انعطاف‌پذیر (FJS)[۷] حالت توسعه‌یافته‌ی کار کارگاهی (JS) است که در آن هر عملیاتی توسط مجموعه‌یی از ماشین‌ها قابل پردازش است. با توجه به تعریف، مسئله‌ی زمان‌بندی FJS شامل دو زیرمسئله‌ی مسیریابی (تخصیص ماشین به عملیات) و زمان‌بندی عملیات (تعیین توالی عملیات) است.[۹]

زمان‌بندی کار کارگاهی انعطاف‌پذیر با ماشین‌های موازی (FJSPM) بسط دو نظریه‌ی سنتی FJS و ماشین‌های موازی (PM) است. درFJS، در هر مرحله از پردازش فقط یک ماشین وجود دارد ولی در FJSPM، حداقل در یک مرحله تعداد ماشین‌ها بیشتر از یکی است و در هر مرحله مجموعه‌یی از ماشین‌های موازی در کنار هم قرار می‌گیرند که هر یک از آنها قابلیت پردازش عملیات تخصیص داده شده به آن مرحله را دارد. بنابراین در هر مرحله مسیرهای متفاوتی برای پردازش یک کار می‌توان در نظرگرفت. در حقیقت نوع جدیدی از انعطاف‌پذیری عملیات در سیستم‌های کارکارگاهی تعریف می‌شود که در ادبیات موضوع چندان به این مسئله پرداخته نشده است. از اهداف اصلی به‌کارگیری این انعطاف‌پذیری در محیط‌های کارگاهی می‌توان به افزایش تولید، رفع مشکل گلوگاهی و استفاده از آن به‌صورت یک مزیت رقابتی در محیط‌های اقتصادی اشاره کرد.

در سال ۱۹۹۸ مسئله‌ی جریان کارگاهی مختلط سه‌مرحله‌یی با ساختار ویژه‌ی یک ماشین در مراحل اول و سوم، و دو ماشین در مرحله‌ی دوم با هدف کمینه‌سازی

بیشترین زمان تکمیل کارها ($C_{Max}$) مورد بررسی قرار گرفت.[۱۰] در همین سال یک الگوریتم جست‌وجوی ممنوع (TS)[۸] نیز برای حل مسئله‌ی جریان کارگاهی با ماشین‌های موازی با هدف کمینه‌سازی $C_{Max}$ ارائه شد.[۱۱]

در سال ۱۹۹۹ یکی از محققین مسئله‌ی برنامه‌ی فرایند انعطاف‌پذیر با تابع هدف چندگانه را با یک الگوریتم بهینه‌سازی دقیق حل کرد.[۱۲] او مسئله‌ی بارگذاری ماشین را به‌عنوان مدل عدد صحیح دومعیاره فرمول‌بندی کرد و دو روش ابتکاری و فراابتکاری مختلف ـ یکی بر پایه‌ی نظریه‌ی دوگان و دیگری بر پایه‌ی الگوریتم ژنتیک ـ را برای آن پیشنهاد داد. در این سال همچنین یک رویکرد ترکیبی از روش‌های فراابتکاری برمبنای الگوریتم ژنتیک برای حل مسئله‌ی FJS به‌منظور کمینه‌سازی $C_{Max}$ ارائه شد.[۱۳] پس از آن در سال ۲۰۰۱، به‌منظور حل مسئله‌ی کارکارگاهی انعطاف‌پذیر با/بدون در نظر گرفتن قطع عملیات یک الگوریتم چندجمله‌یی ارائه شد.[۱۴] در سال ۲۰۰۲ برای اولین بار مسئله‌ی FJS در حالت چندمنظوره مورد بررسی قرار گرفت[۱۵] که در آن روش ارائه‌شده ترکیبی از منطق فازی و الگوریتم‌های تکاملی است و انعطاف‌پذیری مسئله تنها به انعطاف‌پذیری عملیات محدود می‌شود. از منطق فازی برای جست‌وجو در فضای اهداف، به‌منظور دسترسی به جواب‌های بهینه‌ی پارتو استفاده شده است. در همین سال، یک الگوریتم ژنتیک برای حل مسئله‌ی‌مشابه FJS در زنجیره‌ی تأمین با تابع هدف کمینه‌سازی $C_{Max}$ ارائه شد.[۱۶] در مدل ارائه‌شده فرض وجود منابع بیرونی، فرض جایگزینی ماشین برای هر عملیات و نیز فرض توالی عملیات چندگانه برای هر قطعه در نظر گرفته شده است.

در سال ۲۰۰۳ با استفاده از روش جست‌وجوی هوش مصنوعی[۹]، روشی جدید تحت نام الگوریتم تکاملی همزیستی[۱۰] (نوعی الگوریتم هم‌تکاملی) ارائه شد.[۱۷] در همین سال محققین اقدام به حل مسئله‌ی کارکارگاهی انعطاف‌پذیر به‌کمک روشی ابتکاری با رویکرد منطق فازی کردند.[۱۸] در سال ۲۰۰۴ انعطاف‌پذیری در کارکارگاهی تنها به انعطاف‌پذیری در عملیات محدود شد.[۱۹] هدف این تحقیق کمینه‌سازی دیرکردها بود که با استفاده از الگوریتم جست‌وجوی ممنوع (TS) به حل مسئله می‌پردازد. در سال ۲۰۰۴ یک الگوریتم ژنتیک برای حل مسئله‌یی که محققین در سال ۲۰۰۲ مطرح کرده بودند، ارائه شد.[۲۰] در این تحقیق نخست زیرمسئله‌ی تخصیص توسط قاعده‌ی اولویت‌بندی SPTحل می‌شود، سپس بر این اساس که انتخاب یک بازنمایی مناسب از کروموزوم‌ها مهم‌ترین مرحله در به دست آوردن جواب‌های با کیفیت است یک الگوریتم ژنتیک با بازنمایی کروموزوم‌ها براساس عملیاتِ پایه برای حل مسئله ارائه شده است. نتایج حاصل از محاسبات نشان‌گر کارایی الگوریتم در برخورد با مسائل بزرگ است. در سال ۲۰۰۵ با استفاده از یک الگوریتم ابتکاری برمبنای الگوریتم شبیه‌سازی حرارتی (SA)[۱۱] نسبت به حل مسئله‌ی زمان‌بندی جریان کارگاهی چندمرحله‌یی با ماشین‌های موازی متفاوت اقدام شد[۲۱] که در آن هدف کمینه‌سازی کل زمان جاری است. در همین سال همچنین مسئله‌ی مطالعه‌شده در سال‌های پیش[۱۵] مورد بررسی قرار گرفت[۲۲] و طی آن یک رویکرد ترکیبی سلسله‌مراتبی برای حل مسئله پیشنهاد شد. در رویکرد پیشنهادی از الگوریتم بهینه‌سازی تجمعی ذرات (PSO)[۱۲] برای حل زیرمسئله‌ی مسیریابی، و از الگوریتم شبیه‌سازی حرارتی برای حل زیرمسئله‌ی تعیین توالی عملیات استفاده می‌شود. در سال ۲۰۰۴ برای حل مسئله‌ی زمان‌بندی اقتصادی گروه چندمحصولی با چرخه‌ی عمومی، در کارکارگاهی انعطاف‌پذیر روشی بیان شد.[۲۳] در همین سال مجدداً یک روش عملی سلسله‌مراتبی کارا به‌منظور حل مسئله‌ی کارکارگاهی انعطاف‌پذیر با تابع هدف چندگانه پیشنهاد شد.[۲۴] این روش از بهینه‌سازی تجمعی ذرات برای تخصیص عملیات به ماشین‌ها، و از حرارت‌دهی شبیه‌سازی شده برای زمان‌بندی عملیات روی هر ماشین بهره می‌برد. پس از آن الگوریتمی مبتنی بر سیستم ایمنی بدن برای حل مسئله‌ی کارکارگاهی انعطاف‌پذیر با چرخش مجدد کار ارائه شد.[۲۵]

در سال ۲۰۰۶ محققین مسئله‌ی زمان‌بندی کارکارگاهی انعطاف‌پذیر با ماشین‌های موازی یکنواخت را با تابع هدف کمینه‌سازی $C_{Max}$ مورد مطالعه قرار دادند.[۲۶] آنها دسته‌یی از روش‌های فراابتکاری موجود برای مسئله‌ی کارکارگاهی را گسترش دادند و نشان دادند که روش فراابتکاری مبتنی بر روش جمع برداری، هنگامی که تعداد کارها بسیار زیاد باشد، به‌طور مجانبی بهینه است. در سال ۲۰۰۷ پژوهش‌گران از سه نوع انعطاف‌پذیری کلی (به‌صورت FJSP-۱۰۰)، متوسط (به‌صورت FJSP-۵۰) و جزئی (به‌صورت FJSP-۲۰) در کار خود استفاده کردند.[۲] آنها عنوان کردند که انعطاف‌پذیری C٪، به‌این معنی است که تمامی عملیات‌های کارهای مختلف، می‌توانند حداکثر توسط C٪ از تمامی ماشین‌های موجود درکارگاه پردازش شوند.[۱] همچنین در سال ۲۰۰۷ در حل مسئله‌ی کارکارگاهی انعطاف‌پذیر با فرض گردش مجدد کارها،[۲۷] محققین موضوعِ اتکای بیش از حد الگوریتم‌های تکاملی به سازوکارهای ترکیب مجدد و انتخاب تصادفی را از عمده محدودیت‌های این روش عنوان کردند. به‌منظور غلبه بر این محدودیت‌ها و به دست آوردن جواب‌های کارآمد برای مسائل کار کارگاهی با ماشین‌های موازی (FJSP)[۱۳]، آنها سعی کردند از تعامل بین یادگیری و تکامل بهره بگیرند، و به‌همین منظور الگوریتم ژنتیک پیشنهادی خود را تحت عنوان LEGA ارائه کردند. آنها این الگوریتم را ترکیبی توأمان از الگوریتم تکاملی، الگوی یادگیری و تولید جمعیت در نظر گرفتند. در این رویکرد برای تولید جمعیت از یک قاعده‌ی توزیعی ساده و برای الگوی یادگیری نیز از روش همسایگی K-nearst استفاده شده بود. در سال ۲۰۰۷ برای حل مسئله‌ی زمان‌بندی تولید کارگاهی انعطاف‌پذیر با سه هدف، از ترکیب الگوریتم ژنتیک با روش ابتکاری انتقال گلوگاه استفاده شد.[۲۸] آنها برای بازنمایی کروموزوم‌های الگوریتم ژنتیک خود از روش‌های مطرح‌شده‌ی قبلی[۲۰] بهره جستند که البته در آن برای تخصیص ماشین و توالی عملیات به‌جای یک بردار، از دو بردار استفاده کردند. آنها همچنین در روش ترکیبی خود از قابلیت جست‌وجوی سراسری الگوریتم ژنتیک و نیز از قابلیت جست‌وجوی محلی الگوریتم ابتکاری انتقال گلوگاه برای حل مسئله‌ی خود بهره گرفتند.

به‌طور کلی از نتایج حاصل از مطالعه‌ی جامعِ پیشینه‌ی تحقیق در زمینه‌های «کارکارگاهی»، «کارکارگاهی انعطاف‌پذیر» و «مسائلِ زمان‌بندی با ماشین‌های موازی با سرعت‌های یکسان و غیریکسان»، مشخص شد که در بسیاری از توالی عملیات فرض می‌شود همه‌ی کارها برای اجرا به‌طور هم‌زمان دسترس‌پذیرند؛ اما در بیشتر محیط‌های واقعی ساخت و تولید، مجموعه‌ی کارهایی که باید زمان‌بندی شوند با گذشت زمان تغییر می‌یابد و این موضوع نمونه‌ی «پویایی مسئله کارگاهی» را پدید می‌آورد. به‌علاوه لزوم سازگاری سیستم‌های ساخت و تولید با نوساناتی مانند ازکار افتادگی ماشین‌ها و ایجاد ماشین‌های گلوگاهی از اهمیت ویژه‌یی برای این سیستم‌ها برخوردار است. این سازگاری‌ها با درنظرگرفتن انواع انعطاف‌پذیری‌های کف کارگاهی ایجاد می‌شود به‌طوری که امروزه سیستم‌های ساخت و تولید با بهره‌گیری از انواع انعطاف‌پذیری‌های کف کارگاهی، علاوه بر فائق آمدن بر چنین نوساناتی قادرند از این قابلیت‌ها به‌عنوان مزیت رقابتی خود بهره جویند.

بنابراین با بررسی ادبیات موضوع و شناسایی خلاءهای موجود، در این نوشتار، مسئله‌ی زمان‌بندی کارکارگاهی انعطاف‌پذیر با ماشین‌های موازی در محیط ساخت و تولید پویا (FDJSPM) مورد مطالعه قرار گرفته است. در این نوشتار، برای نخستین‌بار دو پارامتر «انعطاف‌پذیری عملیات» و «انعطاف‌پذیری ناشی از ماشین‌های موازی با سرعت‌های غیریکنواخت» برای محیط کارکارگاهی پویا، تواماً در نظر گرفته شد. به‌عبارت دیگر در این نوشتار، علاوه بر پویایی کارگاه ــ به‌سبب متغیربودن

زمان‌های ورود قطعات به کارگاه ــ مسئله‌یی ترکیبی از زمان‌بندی تولید کارکارگاهی با قابلیت انعطاف‌پذیری عملیات در ایستگاه‌های کاری و نیز وجود پردازنده‌های موازی با سرعت‌های غیریکسان در هر ایستگاه بررسی شد. در ادامه مدل‌سازی مسئله با تابع کمینه‌سازی $F_{\max}$، به‌عنوان تابع هدف زمان‌بندی، بررسی می‌شود.

## تعریف مسئله و مدل‌سازی ریاضی آن

در این بخش نخست متغیرها و مجموعه‌های مورد استفاده در این نوشتار را معرفی خواهیم کرد. سپس با تعریف مسئله‌ی FDJSPM و ارائه‌ی یک مدل برنامه‌ریزی عدد صحیح مختلط و اثبات NP-hard بودن مسئله به این بخش پایان می‌دهیم.

### ۱. تعریف مسئله

مسئله‌ی FDJSPM چنین تعریف می‌شود: برای تعداد $n$ کار (قطعه) $m$ مرحله‌ی پردازش (ایستگاه کاری) وجود دارد به‌طوری که انجام هرکار درکارگاه پویا به مجموعه عملیات مشخصی نیازمند است. هر قطعه مانند قطعه‌ی $i$ام، برای پردازش درکارگاه در یک زمان غیر صفر ($r_i$) وارد کارگاه پویا می‌شود. قطعه‌ی $J_i$ شامل زنجیره‌یی از عملیات $\{o_{i,1}, o_{i,2}, ..., o_{i,n_i}\}$ است که بدون از دست دادن کلیت مسئله می‌توان چنین فرض کرد که ترتیب انجام عملیات در کارگاه نیز به‌همین صورت باشد. ترتیب حرکت هرکار بین ایستگاه‌های کاری و برای کارهای مختلف متفاوت است (خصوصیت اصلی مسائل کارکارگاهی). تعداد عملیات لازم برای تکمیل هر کار ممکن است کوچک‌تر یا مساوی تعداد مراحل پردازش است. در هر مرحله از پردازش مانند $M_k$ به‌تعداد $l_k$ نسخه از ماشین‌های موازی با سرعت‌های مختلف وجود دارد که هریک توانایی پردازش عملیات تخصیص یافته به آن مرحله را دارند و دست‌کم در یک ایستگاه کاری منعطف ازکارگاه مقدار این پارامتر ($l_k$) بزرگ‌تر از ۱ است. هر عملیات دست‌کم توسط یکی از ایستگاه‌های کاری قابل پردازش است و به‌دلیل وجود انعطاف‌پذیری عملیات درکارگاه، حداقل یک عملیات وجود دارد به‌طوری که در بیش از یک ایستگاه کاری قابل پردازش باشد. عملیات هریک از کارهای موجود درکارگاه را حداکثر با یک ماشین در یک زمان می‌توان انجام داد. طول زمان پردازش عملیات $O_{i,j}$ روی ماشینی با سرعت واحد ($S_{k,pm} = ۱$) در مرحله‌ی $k$ام برابر $P_{pm,j,k}$ در نظرگرفته می‌شود و اگر پردازش این عملیات روی ماشینی با سرعت ($S_{k,pm} > ۱$) انجام شود طول زمان پردازش عملیات مذکور به $P_{i,j,k}/S_{k,pm}$ کاهش خواهد یافت. هدف این نوشتار، کمینه‌سازی حداکثر زمان گردش کاری قطعات ($F_{Max}$) است.

با توجه به تعریف، مسئله‌ی FDJSPM به دو زیرمسئله‌ی تخصیص (اختصاص کارها به ایستگاه‌های منعطف) و تعیین توالی عملیات (تعیین توالی پردازش عملیات کارها دربین‌کارهای موجود در صف انتظار هرایستگاه) تجزیه می‌شودکه زیرمسئله‌ی تخصیص نیز به‌نوبه‌ی خود به دو زیرمسئله تخصیص ایستگاه و تخصیص ماشین تجزیه می‌شود. زیرمسئله‌ی تخصیص به‌دلیل انعطاف‌پذیری (عملیات و ماشین‌های موازی) ایجادشده در کار کارگاهی به وجود می‌آید که شدیداً بر پیچیدگی مسئله می‌افزاید. هدف کمینه‌سازی $F_{Max}$ با سایر مفروضات زیر است:

۱. هر ماشین در هر لحظه تنها می‌تواند یک عملیات را پردازش کند.

۲. کارگاه پویا بوده و زمان‌های ورود کارها به کارگاه غیرصفر است.

۳. هرکار در هر لحظه تنها می‌تواند به‌وسیله‌ی یک ماشین پردازش شود.

۴. همه‌ی ماشین‌ها از زمان صفر در دسترس‌اند و خراب نمی‌شوند.

۵. قطع کردن عملیات مجاز نیست و زمان‌های پردازش عملیات‌ها نیز قطعی و معلوم است.

۶. انبار پای کار، مجاز و ظرفیت آن نامحدود است.

۷. زمان‌های آماده‌سازی بین عملیات ناچیزاست یا شامل زمان‌های پردازش می‌شود و زمان حمل و نقل قابل چشم‌پوشی است.

بعضی از فرض‌های فوق عملاً غیرواقعی است و فقط برای ساده‌ترشدن مدل و امکان‌پذیر شدن آن در نظرگرفته شده است.

### ۲. تعریف متغیرها و مجموعه‌ها

$n$: تعداد کارهای واردشده به کارگاه در زمان‌های غیرصفر

$n_i$: تعداد عملیات‌های قطعه‌ی $i$ام

$r_i$: زمان ورود قطعه‌ی $i$ام به کارگاه

$m$: تعداد مراحل پردازش (ایستگاه‌های کاری)

$k$: شاخص مرحله (ایستگاه‌های کاری) $k = ۱, ..., m$

$M_k$: $k$امین مرحله‌ی پردازش

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

$o_{i,j}$: $i$امین عملیات کار $j$

$St_{i,j}$: مرحله‌یی که عملیات $o_{i,j}$ در آن پردازش خواهد شد

$l_k$: تعداد ماشین‌های موازی در مرحله $k$ام

$L_{max}$: بیشینه‌ی $l_k$ها

$pm$: شاخص ماشین‌های موازی در ایستگاه‌های کاری $pm = ۱, ..., l_k$

$S_{k,r}$: سرعت ماشین $M_{k,r}$

$P_{i,j}$: تعداد واحد زمانی برای پردازش عملیات $o_{i,j}$ روی ماشینی با سرعت واحد

$F_{Max}$: بیشترین زمان گردش کاری

$c_{i,j}$: زودترین زمان تکمیل عملیات $o_{i,j}$

$Ft_{k,pm}^{(i,j)}$: زمان پایان عملیات $o_{i,j}$ روی ماشین $M_{k,pm}$

$a_k^{(i,j)} = \begin{cases} ۱ \\ ۰ \end{cases}$: اگر ماشین‌های $k$امین مرحله پردازش جزء ماشین‌های علی‌البدل ممکن برای $o_{i,j}$ باشد برابر ۱ و در غیر این صورت برابر صفر در نظر گرفته می‌شود.

$X_{k,pm}^{(i,j)} = \begin{cases} ۱ \\ ۰ \end{cases}$: اگر عملیات $o_{i,j}$ روی ماشین $M_{k,pm}$ پردازش شود برابر ۱ و در غیر این‌صورت برابر صفر در نظرگرفته می‌شود.

$R_{k,pm}^{(i,j)(p,q)} = \begin{cases} ۱ \\ ۰ \end{cases}$: اگر عملیات $o_{a,b}$ زودتر از عملیات $o_{i,j}$ روی ماشین $M_{k,pm}$ صورت‌پذیرد برابر ۱ و در غیر این‌صورت برابر صفر در نظرگرفته می‌شود.

### ۳. مدل ریاضی

#### الف) تابع هدف مسئله

$Minimization\ of\ Flow\ Time$

$F = \{Max\{C_i | i = ۱, ..., N\} - r_i\}$

**ب) محدودیت‌های مسئله**

$$Ft_{k,pm}^{(i,j+1)} - Ft_{k',pm'}^{i,j} + L \times (1 - a_k^{(i,j+1)} \times X_{k,pm}^{(i,j+1)}) \geq P_{i,j+1,k}/S_{k,pm} \quad \begin{matrix} 1 \leq i \leq n;\ 1 \leq j \leq (n_i - 1); \\ 1 \leq k, k' \leq m; \\ 1 \leq pm, pm' \leq l_{k'}; \end{matrix} \quad (1)$$

$$Ft_{k,pm}^{(i,j)} - Ft_{k,pm}^{(p,q)} \times X_{k,pm}^{(i,j)} + L \times R_{k,pm}^{(i,j)(p,q)} \geq X_{k,pm}^{(i,j)} \times P_{i,j,k}/S_{k,pm} \quad \begin{matrix} i = 1, ..., n-1 \,\&\, q = i+1, ..., n; \\ 1 \leq j \leq n_j;\ 1 \leq p \leq n_q; \end{matrix} \quad (2)$$

$$Ft_{k,pm}^{(p,q)} - Ft_{k,pm}^{(i,j)} \times X_{k,pm}^{(p,q)} + L \times (1 - R_{k,pm}^{(i,j)(p,q)}) \geq X_{k,pm}^{(p,q)} \times P_{p,q,k}/S_{k,pm} \quad \begin{matrix} 1 \leq k \leq m; \\ 1 \leq pm \leq l_k; \end{matrix} \quad (3)$$

$$\sum_{k=1}^{m} \sum_{pm=1}^{l_k} a_k^{(i,j)} \times X_{k,pm}^{(i,j)} = 1 \quad \begin{matrix} 1 \leq i \leq n; \\ 1 \leq j \leq n_j; \end{matrix} \quad (4)$$

$$X_{k,pm}^{(i,j)} \leq a_k^{(i,j)} \quad \begin{matrix} 1 \leq i \leq n;\ \forall i: 1 \leq j \leq n_j; \\ 1 \leq k \leq m; \\ 1 \leq pm \leq l_k; \end{matrix} \quad (5)$$

$$Ft_{k,pm}^{(i,j)} \leq L \times X_{k,pm}^{(i,j)} \quad \begin{matrix} 1 \leq i \leq n;\ \forall i: 1 \leq j \leq n_j; \\ 1 \leq k \leq m; \\ 1 \leq pm \leq l_k; \end{matrix} \quad (6)$$

$$C_i \geq \sum_{k=1}^{m} \sum_{pm=1}^{l_k} Ft_{k,pm}^{(i,n_j)} \quad 1 \leq i \leq n; \quad (7)$$

$$Ft_{k,pm}^{(i,1)} \geq X_{k,pm}^{(i,1)} \times P_{i,1,k}/S_{k,pm} + r_i \quad \begin{matrix} 1 \leq i \leq n; \\ 1 \leq k \leq m; \\ 1 \leq pm \leq l_k; \end{matrix} \quad (8)$$

## ۴. شرح مدل

- برای این که هر برنامه‌ی زمانی، راه‌حل امکان‌پذیر محدودیت‌های منطقی نیز باشد همه‌ی موارد عملیات انجام شده در چارچوب هرکار معین، باید با ترتیبی تقدمی و بدون تداخل با یکدیگر، بر روی منابع تخصیص داده شوند.[۱] نامعادلات ۱ تضمین می‌کند که مجموعه‌ی توالی عملیات کارها تداخل زمانی نداشته باشند، به‌عبارتی مجموعه این نامعادلات باعث می‌شود هر عملیات از یک توالی در صورتی شروع شود که عملیات متقدم آن انجام شده باشد.
- از آنجا که پردازش دو عملیات به‌طور هم‌زمان بر روی یک ماشین ممکن نیست، باید هر برنامه‌ی زمانی معتبر یک راه‌حل ممکن برای محدودیت‌های منابع نیز باشد.[۱] مجموعه محدودیت‌های ۲ و ۳ به‌طور هم‌زمان تضمین می‌کنند که هیچ یک از عملیاتی که روی یک ماشین انجام می‌شوند، تداخل زمانی نداشته باشند.

تمامی مدل‌های ارائه‌شده برای مسائل کارکارگاهی دست‌کم دو محدودیت مشابه دارند که در صورت نقض آنها مسئله از حالت کارکارگاهی خارج می‌شود. بنابراین تمامی مسائل کارگاهی، نامعادلات ۱ و ۲ و ۳ را، که در مدل ریاضی مطرح در نوشتار ارائه شده، شامل می‌شوند. نوآوری این نوشتار در مدل‌سازی عبارت است از:

الف) مجموعه معادلات ۴ و نامعادلات ۵ در مدل ریاضی ارائه‌شده در نوشتار تضمین می‌کنند که هر یک از مجموعه عملیات مربوط به یک کار فقط می‌تواند به یکی از ماشین‌های علی‌البدل تخصیص یابد. به‌عبارت دیگر معادلات ۴ متضمن این نکته است که هرکار در هر مرحله از ماشین‌های موازی، حداکثر روی یک ماشین و فقط یک‌بار پردازش می‌شود. این معادلات به‌دلیل لحاظ‌کردن انعطاف‌پذیری مسئله که ناشی از حضور ماشین‌های موازی با سرعت غیریکنواخت در ایستگاه‌های کاری کارگاه است ایجاد شده است. درکنار این معادلات، نامعادلات ۶ نیز بیان می‌کنند که اگر هیچ‌یک از ماشین‌های مرحله‌ی $k$ام به عملیات $O_{i,j}$ اختصاص داده نشوند، زمان تکمیل این عملیات روی تمامی ماشین‌های این مرحله باید صفر در نظرگرفته شود. مجموعه‌ی این معادلات و نامعادلات برای درنظرگرفتن منعطف‌بودن مسئله به‌دلیل وجود عملیات انعطاف‌پذیر درکارگاه ایجاد شده است.

ب) از طرف دیگر چون تابع هدف مسئله ($F_{Max}$) مبتنی بر معیاری عادی از زمان تکمیل کارها است، بنابراین در نامعادلات ۷ بیشترین زمان تکمیل کارها محاسبه می‌شود، و برای این که زمان تکمیل اولین عملیات کارها بزرگ‌تر یا مساوی طول زمان پردازش آن عملیات، به‌علاوه‌ی زمان حضور کار مذکور در کارگاه باشد، مجموعه‌ی نامعادلات ۸ در مدل آورده شده است. مجموعه‌ی این نامعادلات نیز به‌منظور درنظرگرفتن پویایی مسئله به‌دلیل متفاوت بودن زمان ورود کارها به کارگاه ایجاد شده است.

به‌منظور سنجش صحت مدل‌ارائه‌شده برای مسئله‌ی FDJSPM مدل برای حل مسائل در مقیاس کوچک توسط نرم‌افزار Lingo اجرا و آزموده شده است. نتایج حاصله مؤید صحت مدل ارائه‌شده برای مسئله‌ی FDJSPM است.

## ۵. NP-hardبودن مسئله

پیچیدگی یک مسئله‌ی ترکیبی عبارت است از مقدار زمانی که در یک الگوریتم برای حل آن مسئله صرف می‌شود. مسئله‌ی FDJSPM، توسعه‌یافته‌ی مدل کار کارگاهی انعطاف‌پذیر (FDJSP) با درنظرگرفتن انعطاف‌پذیری ناشی از ماشین‌های موازی در یک محیط ساخت و تولید پویا است و درنتیجه همه‌ی پیچیدگی‌ها و سختی‌های آن را نیز به‌همراه خواهد داشت. چنانچه فقط انعطاف‌پذیری عملیات در این مسائل ملحوظ شود، حتی بر پیچیدگیِ یافتن جواب‌های بهینه‌ی تقریبی نیز به‌طور چشم‌گیری افزوده می‌شود.[۱] از آن‌جا که مسئله‌ی FDJSP با انعطاف‌پذیری عملیات قویاً Np-hard است،[۲] ملحوظ داشتن انعطاف‌پذیری ناشی از ماشین‌های موازی در محیط ساخت و تولید پویا نیز قویاً Np-hard خواهد بود.

## ۶. رویکرد حل مسئله

برای حل این‌گونه مسائل زمان‌بندی، راهکار بهینه‌سازی کارآمدی مورد نیاز است تا با در نظر داشتن پیچیدگی‌های ناشی از افزایش نمایی فضای جواب، مقادیر مناسبی برای معیار عملکرد مسئله به دست آید. در تحقیق حاضر سعی شده است از قابلیت‌های

الگوریتم ژنتیک در طراحی روشی مناسب برای حل این مسئله استفاده شود. در ادامه ساختار الگوریتم ژنتیک پیشنهادی برای حل مسئله‌ی FDJSPM معرفی و پس از تنظیم پارامترها، کارایی الگوریتم پیشنهادی برای حل این مسئله براساس معیار عملکرد مسئله و از طریق آزمایشات عددی نشان داده خواهد شد.

## الگوریتم ژنتیک پیشنهادی

از آنجا که مسئله‌ی FDJSPM جزء مسائل NP-hard است، الگوریتمی که بتواند در زمان معقول به جواب بهینه‌ی مسئله با اندازه‌ی بزرگ و یا حتی متوسط دست یابد وجود نخواهد داشت و همچنین الگوریتمی که بتواند به جواب بهینه برسد زمان بسیار زیادی را لازم خواهد داشت. بنابراین در این بخش برای حل مسئله، یک الگوریتم ژنتیک ارائه خواهد شد تا در زمان معقول بتواند به جواب خوبی از مسئله دست یابد. روبه‌ی کلی الگوریتم ژنتیک پیشنهادی در شکل ۱ نشان داده شده است.

عوامل اصلیِ توسعه‌ی الگوریتم ژنتیک عبارت است از: نمایش کروموزوم‌ها، ایجاد جمعیت اولیه، معیار ارزیابی، عملگر تقاطع، عملگر جهش و راهکار انتخاب. به‌علاوه پارامترهای الگوریتم ازجمله تعداد جمعیت، تعداد تکرارها، احتمال عمل تقاطع ($p_c$)، احتمال عمل جهش ($p_m$) باید قبل از اجرای الگوریتم مشخص شوند.[۱۵]

### ۱. نمایش کروموزوم‌ها (کدگذاری مسئله)

اولین مرحله‌ی حل مسئله‌ی زمان‌بندی کارگاهی، یا هر مسئله‌ی بهینه‌سازی دیگر به‌کمک الگوریتم ژنتیک، بازنمایی جواب‌های مسئله به‌شکل کروموزوم است.[۲۹]

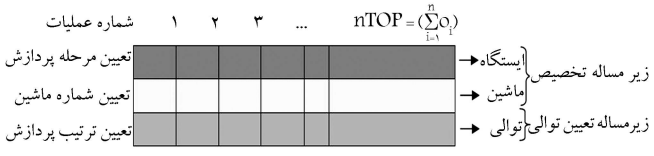
یکی از قابلیت‌های جست‌وجو به‌وسیله‌ی الگوریتم ژنتیک، امکان‌طراحی کروموزوم‌های متناظر با جواب‌ها (در اینجا برنامه‌های زمانی) است. بنابراین اگر در استفاده از این الگوریتم‌ها، طراحی کروموزوم‌ها و عملگرهای کلاسیکی که روی این کروموزوم‌ها اعمال می‌شوند (عملگرهای جهشی و تقاطعی) به‌نحو مطلوبی انجام گیرد از به وجود آمدن کروموزوم‌های غیرموجه در هر تکرار اجتناب می‌شود، و لذا در طول اجرا استفاده از راهکارهایی نظیر راهکار جریمه یا راهکار اصلاحی[۱۴] ضروری نیست. در ادبیات استفاده از الگوریتم ژنتیک برای حل مسائل، معیارهای زیادی در طراحی کروموزوم‌ها در نظر گرفته شده است. ولی معمولاً در نظر داشتن معیارهایی همچون کم‌ترین مقدار نیاز به فضا و زمان به‌لحاظ پیچیدگی محاسباتی مسئله و نیز اجتناب از به‌وجود آمدن کروموزوم‌های غیرموجه در طراحی کروموزوم‌ها حائز اهمیت است.[۱]

مسئله‌ی FDJSPM به دو زیرمسئله‌ی «تخصیص» و «تعیین توالی عملیات» تجزیه می‌شود. الگوریتم ژنتیک ارائه‌شده نیز طوری طراحی شده است که بتواند به‌طور یک‌پارچه و هم‌زمان قادر به حل هر دو زیرمسئله‌ی یادشده باشد. برای این منظور یک کروموزوم دوبعدی پویا ـ با درنظرگرفتن پویایی کارهای وارد شده به کارگاه ـ ارائه شده است (شکل ۲). در این بازنمایی، طول کروموزوم برابر تعداد کل عملیات کارهای موجود برای زمان‌بندی (nTOP) و عرض آن نیز برابر ۳ در نظرگرفته شده است. بنابراین هر جواب مسئله به‌صورت یک آرایه‌ی دوبعدی نمایش داده می‌شود. این روش مشابه روش به‌کارگرفته شده در سال ۲۰۰۲ است که البته در آن رشته‌ی تخصیص به‌صورت دو رشته‌ی مجزا و شامل رشته‌ی تخصیص ایستگاه کاری و نیز رشته‌ی تخصیص ماشین در نظرگرفته شده است. در این بازنمایی رشته‌ی اول کروموزوم مشخص‌کننده‌ی ایستگاه کاری (مرحله‌ی) پردازش‌کننده‌ی عملیات مورد نظر است و رشته‌ی دوم نیز شماره‌ی ماشینی است که در ایستگاه کاری مورد نظر عملیات مربوطه را پردازش خواهد کرد. رشته‌ی سوم این بازنمایی نیز نشان‌گر اولویت تخصیص‌یافته به هر عملیات است. هر عنصر سطر سوم همواره عددی است بین ۱ و nTOP، با این ویژگی که اولویت دو عملیات نمی‌تواند یکسان باشد. لذا همواره جواب مسئله موجه خواهد بود. بنابراین در این طراحی هیچ‌وقت با کروموزوم ناموجه روبه‌رو نمی‌شویم. به‌طور کلی می‌توان گفت که دو رشته‌ی اول کروموزوم، تخصیص و رشته سوم توالی عملیات را مشخص می‌کند.

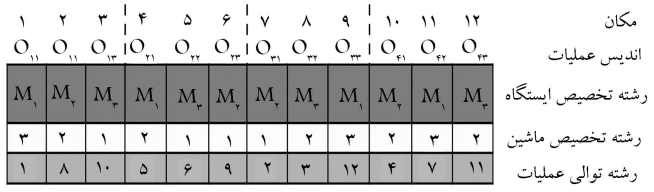
برای درک بهتر سیستم کدگذاری، مثالی را در نظر بگیرید که دارای ۴ کار و ۳ ایستگاه کاری است و برای یک سیستم ساخت و تولید انعطاف‌پذیر با ماشین‌های موازی به‌گونه‌یی مطرح شده که در آن، هرکار دارای ۳ عملیات و هر ایستگاه به‌ترتیب دارای ۳، ۲ و ۲ ماشین موازیِ یکسان است. همچنین کارگاه در عملیات به‌طور کامل انعطاف‌پذیر است. در این صورت کروموزوم متناظر با جواب دارای ۳ رشته به‌طول ۱۲ است و مطابق شکل ۳ کدگذاری می‌شود. مفهوم این نحوه‌ی نمایش کروموزوم چنین بیان می‌شود: براساس رشته توالی عملیات، نخستین عملیات برای پردازش عملیات اول کار یک است که با توجه به رشته‌ی تخصیص ایستگاه، در ایستگاه کاری اول کارگاه و نیز با توجه به رشته‌ی تخصیص ماشین، بر روی ماشین سوم از مجموعه ماشین‌های موازی موجود در این ایستگاه کاری پردازش می‌شود. سایر مکان‌های کروموزوم نیز به‌همین صورت قابل تفسیرند.

رویه الگوریتم ژنتیک پیشنهادی:
۱. شروع؛
۲. شاخص شمارنده جمعیت (t) را برابر صفر قرار بده؛
۳. اولین کروموزوم دوبعدی پویا را طراحی کن؛
۴. تا زمانی که شرط توقف برقرار نشده ادامه بده؛
۵. فرزند C(t) را با اعمال ترکیب مجدد از والد P(t) تولید کن؛
۶. مقدار برازندگی فرزند C(t) را حساب کن؛
۷. از بین C(t) و والد P(t)،P(t+۱) را تعیین کن؛
۸. یک واحد به t اضافه کرده و دوباره شرط خط ۴ را تست کن؛
۹. پایان (شرط ۴)؛
۱۰. پایان (کل الگوریتم).

**شکل ۱. رویه‌ی کلی الگوریتم ژنتیک پیشنهادی.**




**شکل ۲. بازنمایی کروموزوم‌ها در GA پیشنهادی.**




**شکل ۳. نمایش کروموزوم متناظر با جواب مسئله.**

## ۲. جمعیت اولیه

پس از تعیین روش تبدیل هر جواب به کروموزوم، باید جمعیت اولیه‌یی از کروموزوم‌ها تولید شود. چگونگی تولید اعضای جمعیت اولیه نقش اساسی در مشخص‌کردن کیفیت جواب نهایی در هر روش جست‌وجوی محلی ـ ازجمله در الگوریتم ژنتیک ـ دارد. تولید تصادفی جمعیت اولیه سبب حفظ تنوع کروموزوم‌ها در تولید جمعیت، کاهش احتمال هم‌گراییِ زودرس، و نیز به دام افتادن در نقاط بهینه‌ی محلی می‌شود.[۲۸]

در الگوریتم ژنتیک پیشنهادی نیز به‌منظور اجتناب از هم‌گرایی زودرس از هیچ‌گونه روش ابتکاری برای تولید جمعیت اولیه استفاده نمی‌شود. بنابراین سطر اول و دوم در بازنمایی کروموزوم‌ها با اختصاص تصادفیِ یکی از ایستگاه‌های کاری منعطف و نیز یک ماشین از میان مجموعه‌ی جایگزین‌های ممکن برای ماشین‌ها در انجام هر عملیات پر می‌شود و سطر سوم نیز با ترتیبی تصادفی از اعداد ۱ تا nTOP تکمیل می‌شود.

## ۳. عملگر تقاطعی

در الگوریتم ژنتیک پیشنهادی از دو عملگر تقاطعی مکان‌محور و بخشی‌نگاشته (یا RMX) به‌صورت ترکیبی و با نرخ $P_c$ استفاده شده است. عملگر تقاطع مکان‌محور[۱۵] بر روی رشته‌ی تخصیص ماشین (رشته‌ی اول و دوم کروموزوم) و عملگر تقاطعی RMX بر رشته‌ی توالی عملیات (رشته‌ی سوم کروموزوم) اعمال می‌شود. در عملیات تقاطعی مکان‌محور، ابتدا تعدادی از ژن‌های موجود در کروموزوم والد اول ($P_1$) به‌طور تصادفی انتخاب، و عیناً در کروموزوم فرزند ($O_1$) کپی می‌شوند. برای تکمیل باقی ژن‌های $O_1$ از کروموزوم $P_2$ کمک گرفته شده و ژن‌های هم‌مکانِ ژن‌های کپی‌نشده در گام نخست، از کروموزوم $P_2$ در عملیات شرکت کرده و به همان مکان از $O_1$ منتقل خواهند شد.

عملگر تقاطعی RMX بر روی رشته‌ی توالی عملیات (رشته‌ی سوم کروموزوم) اعمال می‌شود. این روش که توسط گلدبرگ و لینگل معرفی شد درحقیقت همان عملگر دو نقطه‌ی برش موجود در ادبیات است که برای استفاده در حالت‌های خاص بسط داده شده است. در این روش باید نخست هر دو کروموزوم $P_1$ و $P_2$ را در دو نقطه که به‌صورت تصادفی برگزیده می‌شوند برش داد. این نقاط را «مقاطع نگاشت» می‌نامند. ژن‌های موجود در میان این دو مقطع نگاشت، بین $P_1$ و $P_2$ مبادله می‌شوند. حال برای ایجاد رشته‌ی سوم $O_1$ باید اقدام به جایگذاری ژن‌های $P_1$ و $P_2$ در $O_1$ و $O_2$ کرد، البته تاجایی که هیچ‌گونه ناسازگاری (به‌لحاظ غیر یکسان بودن شماره‌های توالی‌ها) در کروموزوم‌ها به وجود نیاید. حال برای تکمیل کروموزوم‌ها باید اقدام به جانشینی ژن‌ها با ژن‌های دیگری کرد که ایجاد ناسازگاری می‌کنند. این ژن‌های جایگزین از طریق تناظر یک به یک حاصله از فرایند تبادل ژن‌های میان دو نقطه‌ی برش به دست می‌آیند.

## ۴. عملگر جهش

عموماً عملیات جهش در الگوریتم ژنتیک ضمن ایجاد تغییرات تصادفی برنامه‌ریزی نشده در کروموزوم‌ها، امکان جست‌وجوی قسمت‌های وسیع‌تری از فضای جواب را فراهم، و از همگرایی زودرسِ الگوریتم جلوگیری می‌کند. به‌همین دلیل در الگوریتم پیشنهادی نیز پس از این که عملگر تقاطعی روی دو رشته‌ی والد اعمال شد، عملگر جهش به‌صورت مجزا به این دو رشته اعمال می‌شود. این روش اعمال نیز بدین صورت است که ابتدا برای تک‌تک عناصر یک رشته، آزمون احتمال جهش صورت می‌گیرد. در صورت انجام موفقیت‌آمیز این آزمون، کروموزوم مذکور جهش می‌یابد. در الگوریتم ژنتیک پیشنهادی، به‌خاطر ساختار ویژه‌ی مسئله و کروموزوم پیشنهادی، یک عملگر جهشی ابتکاری ارائه شده که مبتنی است بر عملگر وارونگی خطی که یک قاعده‌ی ابتکاری پایه‌ریزی شده است به‌طوری که این عملگر با دو نرخ جهشی متفاوت کار می‌کند و برای زیرمسئله‌ی تخصیص از قاعده‌ی ابتکاری عملگر جهش با نرخ جهشی ($p_{m_1}$)، و برای زیرمسئله‌ی تعیین توالی عملیات از عملگر وارونگی با نرخ جهشی ($p_{m_2}$) استفاده می‌کند. در قاعده‌ی ابتکاری ارائه‌شده برای عملگر جهش، در صورت مثبت‌بودن نتیجه‌ی آزمون احتمال جهش بر روی یک کروموزوم، یکی از ژن‌های کروموزوم منتخب به‌صورت تصادفی انتخاب، و رشته‌ی تخصیص ایستگاه آن به عدد دیگری که معرف یکی از ایستگاه‌های کاری منعطف برای پردازش عملیات متناظر با آن ژن است، تغییر می‌یابد و سپس رشته‌ی تخصیص ماشین آن نیز به‌صورت تصادفی تعیین می‌شود.

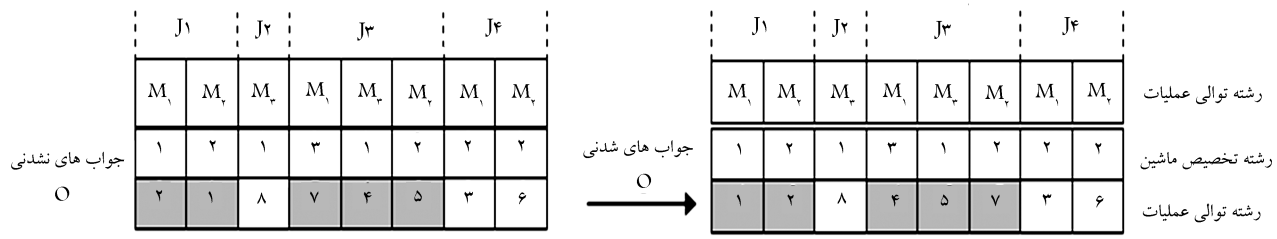
## ۵. رویه‌ی انتخاب

در الگوریتم ژنتیکِ پیشنهادی از فضای نمونه‌گیری توسعه‌یافته استفاده می‌شود. این فضا چنین تعریف می‌شود: اگر اندازه‌ی جمعیت اولیه برابر $\alpha$ باشد و بعد از اعمال عملگرهای تقاطعی و جهشی به تعداد $\beta$ فرزند ایجاد شود، اندازه فضای نمونه‌گیری برابر $\alpha+\beta$ خواهد بود. سازوکار نمونه‌گیری نیز برای تشکیل نسل‌های بعدی ترکیبی از روش نخبه‌گزینی و روش چرخ رولت خواهد بود.

به‌منظور غنی‌کردن هر نسل از جواب‌های خوب و افزایش شانس رسیدن به جواب‌های بهینه،[۲۸] از روش نخبه‌گزینی نیز استفاده می‌شود. براساس روش نخبه‌گزینی که در روش پیشنهادی به کار رفته است، به تعداد کروموزوم‌های غیر یکسان منتخب برای انتقال به نسل بعد ($Best$) از تعداد کل کروموزوم‌های جمعیت $\mu+Pop_Size$ کروموزوم، بدون هیچ‌گونه پیش‌شرطی انتخاب و به نسل بعد منتقل می‌شوند. اگر مقادیر کروموزوم‌های متوالی یکسان باشند از انتخاب این کروموزوم مشابه صرف‌نظر شده و بدون این که به مقدار $Best$ افزوده شود، کروموزوم بعدی آزمایش می‌شود. با به کار بردن روش نخبه‌گزینی در الگوریتم ژنتیک پیشنهادی به تعداد $\lambda$ تا از بهترین عضوهای هر جمعیت به‌طور مستقیم در نسل بعد قرار می‌گیرند. انگیزه‌ی استفاده از این روش دو چیز است: ۱. به‌طور معمول در هر جمعیت، تعدادی از عناصر این جمعیت از بالاترین درجه‌ی برازندگی یا ارزیابی برخوردارند. بنابراین کیفیت مواد ژنی این کروموزوم‌ها نسبت به سایر کروموزوم‌ها بهتر است و ممکن است نماد معرف به این رشته از کروموزوم‌ها نسبت به نمادهای معرف رشته‌های دیگر به حل بهینه‌ی مسئله نزدیک‌تر باشد. از سوی دیگر این رشته کروموزوم‌ها، با توجه به داشتن ارزیابی بهتر نسبت به سایر رشته‌ها، ممکن است دارای فرزندان برازنده‌تری در نسل‌های بعد باشند. ۲. استفاده از این روش، دادن شانس بیشتر به والدین برازنده‌تر در تولید عناصر نسل‌های بعد است.

## ۶. تابع برازندگی

تابع برازندگی میزان اثربخشی هر جواب منتخب یا هر زمان‌بندی ایجادشده را اندازه‌گیری می‌کند. در مسئله‌ی مورد مطالعه‌ی ما، هدف کمینه‌سازی بیشترین زمان گردش کاری کارها ($F_{\max}$) است و لذا به هر جواب منتخب با $F_{\max}$ بزرگ یک مقدار برازندگی پایین تخصیص داده می‌شود. در اینجا تابع برازندگی برای هر کروموزوم




شکل ۴. نحوه‌ی انجام عملیات اصلاح بر روی رشته توالی عملیات.

برابر $f(i) = ۱/F_{\max}(i)$ تعریف شده است؛ $f(i)$ مقدار برازندگی کروموزوم $i$ام و $F_{\max}(i)$ بیشینه زمان گردش کاری کارها در کروموزوم $i$ام است.

## ۷. راهکار برخورد با محدودیت‌ها

بحث دیگری که در اجرای الگوریتم ژنتیک وجود دارد چگونگی برخورد با محدودیت‌های مسئله است، زیرا عملگرهای ژنتیک مورد استفاده در الگوریتم باعث تولید کروموزوم‌های غیر موجه می‌شوند. در الگوریتم ارائه‌شده راهکار اصلاحی برای برخورد با محدودیت‌ها به‌کارگرفته شده است. در این روش، کروموزوم غیرموجه به‌جای حذف‌شدن تبدیل به یک کروموزوم موجه می‌شود (شکل ۴).

## ۸. معیار توقف

الگوریتم بعد از رسیدن به تعداد تکرار (max_gen) متوقف می‌شود.

# طراحی آزمایش‌های عددی

## ۱. روش مورد مقایسه

از آنجا که مسئله‌ی FDJSPM تاکنون مورد مطالعه قرار نگرفته، الگوریتم مشابهی وجود ندارد که بتوان با مقایسه‌ی آن کارایی الگوریتم پیشنهادی را نشان دهیم. از این رو برای نشان‌دادن کارایی الگوریتم پیشنهادی، از الگوریتم RKGA که در سال ۲۰۰۴ برای مسئله‌ی FSPM ارائه شد،[۳۰و۳۱] استفاده کرده‌ایم. الگوریتم RKGA که یک الگوریتم فرابتکاری است، برای حل مسئله‌ی FSPM، با توجه به انعطاف‌پذیری ناشی از وجود ماشین‌های موازی با سرعت‌های یکسان و نیز انعطاف‌پذیری عملیات در بین ایستگاه‌های کاری طراحی و ارائه شده است. به‌همین دلیل، مسئله‌ی حل‌شده توسط این الگوریتم بیشترین شباهت را با مسئله‌ی FDJSPM دارد و به‌راحتی ـ بدون از دست دادن کلیت الگوریتم ـ برای این مسئله قابل استفاده است. بدین‌منظور در این مقایسه ما حالت خاص مسئله‌ی FDJSPM، یعنی همان مسئله‌ی JSPM را در نظر گرفته‌ایم.

ارائه‌دهندگان الگوریتم RKGA، برای حل مسئله‌ی خود از رویکرد سلسله‌مراتبی استفاده کرده و مسئله‌ی خود را به دو زیرمسئله‌ی تخصیص (اختصاص کارها به ایستگاه‌های منعطف) و تعیین توالی عملیات (تعیین توالی پردازش عملیات کارها) تجزیه کرده‌اند. سپس آنها زیرمسئله‌ی تخصیص خود را به دو زیرمسئله‌ی تخصیص ایستگاه و تخصیص ماشین تجزیه کرده‌اند که البته در این مسئله نیز زیرمسئله‌ی تخصیص براثر انعطاف‌پذیری (عملیات و ماشین‌های موازی) ایجادشده، به وجود آمده است.

در تحقیقات انجام شده در سال ۲۰۰۴، برای زمان‌بندی کارها در مرحله‌ی اول از رویکرد RKGA، و برای تخصیص کارها به ماشین‌ها در مراحل بعدی از ابداعات SPTCH (SPT چرخشی) و قاعده‌ی جانسون استفاده می‌شود، بدین طریق که هرکار به ماشینی تخصیص می‌یابد که امکان تکمیل کارها در زودترین زمان ممکن به وجود آید. در حقیقت در این الگوریتم عملیات اول کارها توسط RKGA زمان‌بندی می‌شود و برای زمان‌بندی عملیات بعدی کارها از ابداعات یادشده استفاده می‌شود.[۳۱] ما از این مفهوم هم‌ارز برای مسئله‌ی FDJSPM نیز استفاده کردیم؛ یعنی عملیات اول کارها را توسط RKGAو عملیات بعدی را با استفاده از همان ابداعات زمان‌بندی کردیم.[۳۲]

## ۲. تولید مسائل تصادفی

برای تولید مسائل تصادفی نخست ۶ پارامترِ مسئله را مطابق جدول ۱ شناسایی می‌کنیم. برای پنج پارامتر نخست از الگوریتم RKGA استفاده کردیم.[۳۰و۳۱] و برای سرعت پردازش ماشین‌ها توزیع U(۱و۳) را در نظر گرفتیم.

در حالت کلی، همه‌ی ترکیبات این سطوح آزمایش خواهد شد. بعضی از محدودیت‌های دیگر نیز ذیلاً معرفی می‌شوند. عامل توزیع ماشین به‌صورت متغیر نیازمند آن است که دست‌کم در یک مرحله تعداد متفاوتی از ماشین‌ها نسبت به دیگر مراحل وجود داشته باشد؛ همچنین بیشترین تعداد ماشین در یک مرحله باید از تعداد کارها کم‌تر باشد و دست‌کم در یک مرحله تعداد ماشین‌های موازی بزرگ‌تر از ۱ باشد. بنابراین، ترکیب ۱۰ ماشین در هر مرحله و ۶ کار حذف، و ترکیب ۶ ماشین در هر مرحله و ۶ کار اضافه می‌شود. همچنین ترکیب ۱-۱۰ ماشین در هر مرحله با ۶ کار به ترکیب ۱-۶ ماشین در هر مرحله با ۶ کار تغییر می‌کند و ترکیب ۱ ماشین در هر مرحله حذف می‌شود. در اینجا ۷۲ سناریوی آزمایشی وجود دارد که ۱۰ مجموعه داده برای هرکدام تولید می‌شود.

جدول ۱. جدول سطوح عوامل برای اجرای الگوریتم ژنتیک (GA).

| عوامل | سطوح عوامل | | | |
|---|---|---|---|---|
| تعداد کارها | ۳۰-۱۰۰ | | ۶ | |
| توزیع ماشین‌ها | ثابت | متغیر | ثابت | متغیر |
| تعداد ماشین‌ها | ۲-۱۰ | U(۱و۱۰)-U(۱و۴) | ۲-۶ | U(۱و۶)-U(۱و۴) |
| تعداد مراحل | ۲-۴-۸ | | | |
| زمان‌های پردازش | U(۵۰و۷۰)-U(۲۰و۱۰۰) | | | |
| سرعت پردازش ماشین‌ها | U(۱و۳) | | | |

## ۳. طراحی روش انجام آزمایش‌ها

در ادامه به‌منظور انجام مقایسه، الگوریتم‌های ارائه شده (هر دو الگوریتم پیشنهادی و الگوریتم RKGA) به زبان C کد شده و با رایانه‌یی با مشخصات (CPU ۳GHz, ۱GB of RAM Pentium IV در محیط Borland C++ ۵٫۰۲اجرا می‌شوند. هرکدام از الگوریتم‌ها (GA پیشنهادی و RKGA) با همان ۷۲۰ مجموعه اطلاعاتِ تولیدشده اجرا می‌شوند.

## ۴. تنظیم پارامترها

سطوح مختلف پارامترها برکیفیت جواب‌های حاصل از الگوریتم ژنتیک تأثیر زیادی دارد و ترکیب‌های مختلف مقادیر پارامترها جواب‌های متفاوتی از الگوریتم ارائه می‌دهد. در اینجا فقط رویه‌ی تنظیم پارامترها را برای تنظیم سه پارامتر ($\mu'$)، ($p_c$)، ($p_m$)) به‌کار خواهیم گرفت. برای این منظور از هر سناریوی آزمایشی که در آن ۱۰ مجموعه داده تولیدشده، یک مجموعه و درکل ۷۲ مجموعه داده به‌صورت تصادفی انتخاب می‌کنیم و سپس در رویه‌ی تنظیم پارامترها، با تولید هر مجموعه پارامتر جدید، GA روی ۷۲ مجموعه داده اجرا می‌شود. برای هر اجرا متوسط توابع هدف مورد استفاده قرار خواهد گرفت. نخست سطوح زیر را برای پارامترها در نظر می‌گیریم:

درصد انتخاب بهترین کروموزوم برای نسل بعد ($\mu'$): ۵٪، ۱۰٪، ۱۵٪، ۲۰٪ و ۲۵٪.

احتمال تقاطعی ($p_c$): ۰٫۳، ۰٫۴، ۰٫۵، ۰٫۶، ۰٫۷، ۰٫۸ و ۰٫۹.

احتمال جهشی ($p_m$): ۰٫۰۰۵، ۰٫۰۱، ۰٫۰۱۵، ۰٫۰۲ و ۰٫۰۲۵.

تعداد جمعیت اولیه ($pop_size$): ۱۰۰.

تعداد تکرارها ($Max_gen$): ۱۲۵.

در رویه تنظیم پارامترها نخست با تغییر مقدار یکی از پارامترها در دامنه تعریف شده و ثابت نگه داشتن سایر پارامترها در حداقل مقدار دامنه‌شان بهترین مقدار را برای پارامتر متغیر با توجه به تابع هدف تعریف شده مشخص می‌کنیم. سپس با تثبیت این مقدار برای پارامتر مورد نظر، در مرحله بعدی به تغییر پارامتر دیگر می‌پردازیم. در این مرحله نیز به‌جز مقادیر پارامترهایی که مقدارشان تثبیت شده است سایر پارامترها در حداقل مقدار دامنه‌شان نگه داشته می‌شوند و سپس همانند مرحله قبلی به انتخاب بهترین مقدار پارامتر متغیر می‌پردازیم و این رویه را تا زمانی که تمامی پارامترهای متغیر مقدار تثبیت شده بگیرند ادامه می‌دهیم. نتایج آزمایشات نشان می‌دهد که بهترین مقادیر حاصله برای پارامترهای ($\mu'$)، ($p_c$) و ($p_m$) به‌ترتیب ۲۰٪، ۰٫۷ و ۰٫۰۲ است.(جدول ۲)

## نتایج آزمایش‌ها

در این مرحله، کارایی الگوریتم ژنتیک پیشنهادی با روش فرابتکاری RKGA موجود در ادبیات، برای حل مسئله‌ی FDJSPM با تابع هدف کمینه‌سازی بیشترین زمان گردش کاری کارها برای حل ۲۱۶ سناریوی مختلف و هرکدام به تعداد ۱۰ بار (طبق جدول ۱) مقایسه می‌شود. برای مقایسه‌ی این دو روش حل، از یک شاخص‌اصلی تحت عنوان «میانگین تابع هدف کمینه‌سازی حداکثر زمان گردش کاری کارها» و نیز از چندین شاخص کمکی دیگر همچون «کم‌ترین مقدار تابع هدف کمینه‌سازی بیشترین زمان گردش کاری کارها در طی اجراهای مختلف»، «میانگین زمان حل هر کدام از سناریوها»، «میزان و تعداد دفعات بهبود در تابع هدف» و در ابعاد مختلف مسئله (کوچک، متوسط و بزرگ) استفاده شده است. نتایج حاصل از این مقایسه‌ها، با توجه به اطلاعات جداول ۳ و ۴، عبارت است از:

۱. الگوریتم ژنتیک پیشنهادی براساس شاخص «کم‌ترین مقدار تابع هدف کمینه‌سازی بیشترین زمان گردش کاری کارها» برای ۱۰ اجرا هریک از ۷۲ سناریو مسئله، در ابعاد کوچک، متوسط و بزرگ و به‌تفکیک تعداد مراحل، به‌ترتیب با میانگین ۰٫۵۷٪، ۱٫۰۲٪ و ۱٫۱۷٪ درصد بهبود و درکل با میانگین ۰٫۹۲٪ بهبود، برتری محسوسی نسبت به RKGA دارد.

۲. «میانگین تابع هدف» برای ۱۰ اجرای هریک از ۷۲ مجموعه داده، بهترین معیار برای ارزیابی الگوریتم‌ها به نظر می‌رسد. چنان که در جدول ۳ نیز ملاحظه می‌شود، الگوریتم ژنتیک پیشنهادی براساس این شاخص در ابعاد کوچک، متوسط و بزرگ به‌ترتیب با میانگین ۱٫۱۷٪، ۱٫۳۵٪ و ۱٫۴۹٪ درصد بهبود و درکل با میانگین ۱٫۳۴٪ درصد بهبود، عملکرد بهتری نسبت به RKGA دارد.

۳. روش پیشنهادی با توجه به شاخص «بهبود تابع هدف» در ابعاد کوچک، متوسط و بزرگ به‌طور میانگین و به‌ترتیب با ۵۰، ۴۶٫۷ و ۵۲ بار بهبود و درکل با میانگین ۴۹٫۶ بار و در مجموع با ۴۴۶ بار (از ۲۱۶۰ اجرا) یا ۲۰٫۶۵٪ درصد بهبود از RKGA کاراتر است.

۴. روش پیشنهادی با توجه به شاخص میانگین تابع هدف در ابعاد کوچک، متوسط و بزرگ به‌طور میانگین و به‌ترتیب با ۴۰، ۴۲ و ۴۹ بار بهبود از ۷۲ بار و در کل با میانگین ۴۹٫۶ بار و در مجموع با ۱۳۱ بار (از ۲۱۶ اجرا) یا ۶۰٫۶۵٪ درصد بهبود از RKGA کاراتر است.

۵. با در نظرگرفتن شاخص میانگین زمان حل، روش پیشنهادی در ابعاد کوچک با RKGA تفاوت چندانی ندارد ولی در ابعاد متوسط و بزرگ به‌ترتیب با میانگین ۱۴٫۵ و ۳۳٫۸ ثانیه افزایش و درکل با میانگین ۱۶٫۶ ثانیه افزایش، در مقایسه با RKGA عملکرد نامطلوبی دارد.

معیار «تفاضل میانگین توابع هدف الگوریتم ژنتیک پیشنهادی با روش فرابتکاری RKGA» برای ابعاد مختلف مسئله، شاخص مناسبی به‌منظور نشان‌دادن کارایی الگوریتم پیشنهادی است و نشان‌دهنده‌ی بهبود به‌میزان ۱٫۱۷٪، ۱٫۳۵٪ و ۱٫۴۹٪ و به‌طور متوسط ۱٫۳۴٪ برای ابعاد کوچک، متوسط و بزرگ است و مؤید این مطلب است که با افزایش ابعاد مسئله، میزان بهبود روش پیشنهادی نسبت به روش فرابتکاری RKGA موجود در ادبیات افزایش می‌یابد.

در نهایت، با توجه به نتایج ملاحظه می‌شود که الگوریتم ژنتیک ارائه‌شده برای حل مسئله در مقایسه با RKGA نیازمند زمان حل بیشتری است، اما با توجه به میزان بهبود انجام‌شده در شاخص‌های کم‌ترین مقدار و میانگین تابع هدف براساس میزان و تعداد دفعات بهبود، این افزایش زمان قابل توجیه است. بنابراین با توجه به نتایج حاصل از آزمایش‌های عددی، در مجموع کارایی روش حل ارائه‌شده نسبت به روش RKGA تأیید می‌شود.

جدول ۲. مقادیر پارامترهای الگوریتم ژنتیک پس از تنظیم.

| پارامتر | اندازه جمعیت اولیه | تعداد نسل‌ها | $\mu'$ | $P_c$ | $P_m$ |
|---|---|---|---|---|---|
| Value | ۱۰۰ | ۱۲۵ | ٪۲۰ | ۰٫۷ | ۰٫۰۲ |

جدول ۳. مقایسه‌ی الگوریتم ژنتیک پیشنهادی و الگوریتم RKGA.

| ابعاد مسئله | | کوچک | | | | متوسط | | | | بزرگ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| تعداد کار | | ۶ | | | | ۳۰ | | | | ۱۰۰ | | | |
| تعداد مراحل | | ۲ | ۴ | ۸ | میانگین بعد کوچک | ۲ | ۴ | ۸ | میانگین بعد متوسط | ۲ | ۴ | ۸ | میانگین بعد بزرگ |
| روش RKGA | کمترین Fmax | ۴۰٫۳۶ | ۷۳٫۹۳ | ۱۷۲٫۶ | ۹۵٫۶۲ | ۹۱٫۸۵ | ۱۲۲٫۶ | ۲۰۴ | ۱۳۹٫۴۷ | ۲۸۳٫۹ | ۳۰۷٫۱ | ۳۶۲٫۷ | ۳۱۷٫۸۹ |
| | میانگین Fmax | ۱۲۱٫۰۹ | ۱۸۴٫۶۵ | ۳۰۲٫۸۳ | ۲۰۲٫۸۶ | ۵۲۷٫۴۳ | ۶۴۶٫۸۳ | ۸۴۵٫۲۳ | ۶۷۳٫۱۶ | ۱۷۴۳٫۸۰ | ۱۹۰۵٫۸۸ | ۲۶۱۲٫۹۰ | ۲۰۸۷٫۵۳ |
| | میانگین زمان حل(ثانیه) | ۰٫۰۲۵ | ۰٫۰۲۸ | ۰٫۰۳۵ | ۰٫۰۲۹ | ۰٫۲۰۱ | ۰٫۲۴۸ | ۰٫۲۳۳ | ۰٫۲۲۷ | ۱٫۶۰۲ | ۱٫۵۵۱ | ۱٫۷۴۵ | ۱٫۶۳۳ |
| الگوریتم ژنتیک پیشنهادی | کمترین Fmax | ۳۹٫۷۴ | ۷۴٫۹۵ | ۱۶۹٫۸۶ | ۹۴٫۸۵ | ۹۳٫۳۵ | ۱۱۸٫۰۱ | ۲۰۲٫۰۲ | ۱۳۷٫۷۹ | ۲۷۸٫۱۲ | ۳۰۷٫۹۸ | ۳۵۶٫۲۹ | ۳۱۴٫۱۳ |
| | میانگین Fmax | ۱۱۹٫۶۶ | ۱۸۴٫۷۱ | ۲۹۵٫۶۷ | ۲۰۰٫۰۱ | ۵۲۷٫۶۹ | ۶۳۵٫۲۱ | ۸۲۵٫۸۷ | ۶۶۲٫۹۲ | ۱۷۳۱٫۳۲ | ۱۸۷۰٫۸۹ | ۲۵۶۲٫۵۹ | ۲۰۵۴٫۹۳ |
| | میانگین زمان حل (ثانیه) | ۰٫۰۳۵ | ۰٫۰۴۹ | ۰٫۱۴۶ | ۰٫۰۷۷ | ۰٫۷۳۵ | ۲٫۱۵۲ | ۷٫۹۸۲ | ۳٫۶۲۳ | ۲٫۶۵۳ | ۱۸٫۷۴۵ | ۱۵۸٫۲۳۷ | ۵۹٫۸۸ |
| میزان بهبود (درصد) | کمترین Fmax | ٪۱٫۵۴ | ٪-۱٫۳۸ | ٪۱٫۵۶ | ٪۰٫۵۷ | ٪-۱٫۶۳ | ٪۳٫۷۳ | ٪۰٫۹۶ | ٪۱٫۰۲ | ٪۲٫۰۲ | ٪-۰٫۲۸ | ٪۱٫۷۶ | ٪۱٫۱۷ |
| | میانگین Fmax | ٪۱٫۱۷ | ٪-۰٫۰۳ | ٪۲٫۳۷ | ٪۱٫۱۷ | ٪-۰٫۰۵ | ٪۱٫۸۰ | ٪۲٫۲۹ | ٪۱٫۳۵ | ٪۰٫۷۲ | ٪۱٫۸۴ | ٪۱٫۹۳ | ٪۱٫۴۹ |
| تعداد دفعات بهبود | کمترین* Fmax | ۵۸ | ۲۹ | ۶۳ | ۵۰٫۰ | ۲۳ | ۶۶ | ۵۱ | ۴۶٫۷ | ۶۲ | ۴۱ | ۵۳ | ۵۲٫۰ |
| | میانگین** Fmax | ۱۵ | ۷ | ۱۸ | ۱۳٫۳ | ۱۰ | ۱۹ | ۱۳ | ۱۴٫۰ | ۱۸ | ۱۵ | ۱۶ | ۱۶٫۳ |
| بهبود زمان حل (ثانیه) | | -۰٫۴ | -۰٫۷۵ | -۳٫۱۷۱۴ | -۱٫۴ | -۲٫۶۵۶۷ | -۷٫۶۷۷۴ | -۳۳٫۲۵۸ | -۱۴٫۵ | -۰٫۶۵۶۰۵ | -۱۱٫۰۸۵۸ | -۸۹٫۶۸۰۲ | -۳۳٫۸ |

* کمترین مقدار Fmax، از میان ۷۰ اجرا از سناریوها که بر اساس تعداد مراحل کاری تفکیک شده‌اند، انتخاب شده اند و شامل مقادیر یکسان با RKGA نیز نمی‌شوند.

** کمترین مقدار میانگین Fmax از میان ۲۱ اجرا از سناریوها که بر اساس تعداد مراحل کاری تفکیک شده‌اند انتخاب شده اند، و شامل مقادیر یکسان با RKGA نیز نمی‌شوند.

اعداد منفی جدول نشان دهنده برتری جواب حاصل از RKGA نسبت به روش حل پیشنهادی است.

جدول ۴. مقایسه‌ی بهبود و درصد بهبود اجرای GA پیشنهادی نسبت به الگوریتم RKGA.

| | بعد کوچک | بعد متوسط | بعد بزرگ | جمع |
|---|---|---|---|---|
| کل تعداد دفعات بهبود کمترین Fmax | ۱۵۰ | ۱۴۰ | ۱۵۶ | ۴۴۶ |
| درصد کل دفعات بهبود کمترین Fmax | ٪۲۰٫۸۳ | ٪۱۹٫۴۴ | ٪۲۱٫۶۷ | ٪۲۰٫۶۵ |
| | | | | |
| | بعد کوچک | بعد متوسط | بعد بزرگ | جمع |
| کل تعداد دفعات بهبود میانگین Fmax | ۴۰ | ۴۲ | ۴۹ | ۱۳۱ |
| درصد کل دفعات بهبود میانگین Fmax | ٪۵۵٫۵۶ | ٪۵۸٫۳۳ | ٪۶۸٫۰۶ | ٪۶۰٫۶۵ |

## نتیجه‌گیری

در این نوشتار یک مسئله‌ی ترکیبی از زمان‌بندی کارکارگاهی انعطاف‌پذیر با در نظر گرفتن ماشین‌های موازی (با سرعت‌های پردازش غیریکسان) در هر ایستگاه کاری، برای یک سیستم ساخت و تولید پویا (به‌سبب متغیر بودن زمان‌های ورود کارها به کارگاه) معرفی شد. سپس یک مدل ریاضی از نوع برنامه‌ریزی غیرخطی عدد صحیح مختلط برای مسئله ارائه شد. سپس کمینه‌سازی تابع $F_{max}$ به‌عنوان هدف زمان‌بندی و به‌منظور بهره‌برداری کارا از ماشین‌آلات از طریق کاهش مدت زمان استفاده از منابع درگیر و افزایش نرخ و سرعت فرایند ساخت ـ که یکی از اهداف مهم در زنجیره‌ی تأمین ساخت و تولید است ـ مورد بررسی قرار گرفت. برای حل مسئله از قابلیت‌های الگوریتم ژنتیک در حل مسائل زمان و زمان‌بندی استفاده شد. به‌دلیل پویایی کارگاه، طراحی کروموزوم‌ها به‌گونه‌یی انجام شد که پویایی کارها نیز در آنها مد نظر قرار گیرد؛ به‌عبارتی دیگر کروموزوم‌های دوبعدی پویا برای اجرای الگوریتم طراحی شد. طراحی کروموزوم و انتخاب عملگرهای ژنتیک به‌گونه‌یی صورت گرفت که در برخورد با محدودیت‌ها و تولید کروموزوم غیرموجه کم‌ترین مشکل ایجاد شود. در الگوریتم ژنتیک پیشنهادی، جمعیت اولیه به‌صورت تصادفی تولید شده و فاقد

هرگونه اطلاعات در رابطه با مسئله بود. این امر سبب حفظ تنوع کروموزوم‌ها در جمعیت و کاهش احتمال هم‌گرایی سریع و پیش از موعد شد. همچنین با جلوگیری از ورود کروموزوم‌های تکراری به نسل جدید، مانع از تسلط ابرکروموزوم‌ها بر جمعیت شده و از همگرایی سریع به جواب بهینه‌ی محلی جلوگیری می‌شود. پس از تنظیم پارامترهای الگوریتم ژنتیک پیشنهادی، عملکرد آن در مقایسه با رویکرد RKGA موجود در ادبیات مورد مقایسه قرارگرفت و ملاحظه شد که الگوریتم ژنتیک ارائه‌شده برای حل مسئله‌ی تحقیق در مقایسه با RKGA نیازمند زمان حل بیشتری است، اگرچه با توجه به‌میزان بهبود انجام‌شده در شاخص‌های کم‌ترین مقدار و میانگین تابع هدف براساس میزان و تعداد دفعات بهبود، این افزایش زمان قابل توجیه است. نتایج این مقایسه، نشان‌دهنده‌ی بهبود به‌میزان ۱٫۱۷٪، ۱٫۳۵٪ و ۱٫۴۹٪ و به‌طور متوسط ۱٫۳۴٪ برای ابعاد کوچک، متوسط و بزرگ از مسئله‌است. بنابراین با توجه به‌نتایج حاصل از آزمایش‌های عددی، در مجموع کارایی روش حل پیشنهادی نسبت به روش RKGA تأیید می‌شود. ازجمله مسیرهای آتی تحقیق، افزایش کارایی الگوریتم پیشنهادی به‌منظور استفاده‌ی هرچه بیشتر از دانش مدل در یافتن جواب‌های بهینه و نیز اجتناب از گیرافتادن در نقاط بهینه‌ی محلی و نیز کاهش زمان اجراست. از طرفی در مسائل ساخت و تولید واقعی، تصمیم‌گیرنده معمولاً با چند هدف روبه‌رو است. انتخاب ترکیب مناسب اهداف و بررسی مسئله‌ی تحقیق در حالت چندهدفی و نیز بررسی مسئله در محیط احتمالی از ضروریات است.

## پانوشت

1. parallel machines (PM)
2. dynamic job shop (DJS)
3. flexible dynamic job-shop with parallel machines (FDJSPM)
4. genetic algorithm (GA)
5. job shop (JS)
6. rolling time horizon
7. flexible job shop (FJS)
8. taboo search (TS)
9. artificial intelligent
10. symbiotic evolutionary algorithm
11. simulated annealing (SA)
12. particle swarm optimization (PSO)
13. flexible job shop with parallel machines (FJSP)
14. repairing strategy
15. position based crossover

## منابع


1. Baker K.R., *Introduction to Sequencing and Scheduling*, New York, John Wiley & sons (1974).
2. Tay, J.C. and Ho, N.B. "Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems", *Computer & Industrial engineering*, **54**, pp. 453-473 (2007). Available from <www.elsevier.com>.
3. Pinedo, M. and Chao, X., *Operations Scheduling with Applications in Manufacturing and Services*, chapter 3, McGraw-Hill, New York (1999).
4. Vokurka, R.J. and O'Leary-Kelly, S.W. "A review of empirical research on manufacturing flexibility", *Journal of Operations Management,* **18**, pp. 485-501 (2000).
5. Bruker, P.; Jurisch, B. and Sievers, B. "A fast branch and bound algorithm for the job-shop scheduling problem", Discrete Applied Mathematics, **49**, pp. 107-127 (1994).
6. Carlier, J. and Pinson, E., Management Science, **35**, pp. 164-176 (1989).
7. Muhlemann, A.P.; Lockett, A.G. and Farn C.K. "Job-shop scheduling heuristics and frequency of scheduling" *International Journal of Production Research,* **20**, pp. 227-241 (1982).
8. Sun, D. and Lin, L. "A dynamic job shop scheduling framework: A backward Approach" *International Journal of Production Research,* **32**, pp. 967-985 (1994).
9. Scrich, C.A.; Armentano, V.A. and Laguna, M. "Tardiness minimization in a flexible job shop: A taboo search approach", *Journal of Intelligent Manufacturing,* **15**, pp. 103-115 (2004).
10. Riane, F.; Artiba, A. and Elmaghraby, S.E. "A hybrid three-stage flow shop problem: Efficient heuristics to minimize makespan", *European Journal of Operational Research,* **109**, pp. 321-329 (1998).
11. Nowicki, E. and Smutniciki, C. "The flow shop with parallel machines: A taboo search approach", *European Journal of Operational Research,* **106**, pp. 226-253 (1998).
12. Brandimarte, P. "Theory and methodology, exploiting process plan flexibility in production scheduling: A multi-objective approach", *European Journal of Operational Research,* **114**, pp. 59-71 (1999).
13. Ghedjati, F. "Genetic algorithms for the job-shop scheduling problem with unrelated parallel constraints: Heuristic mixing method machines and precedence", *Computer & Industrial Engineering,* **37**, pp. 39-42 (1999).
14. Jansen, K. "Approximation algorithms for flexible job shop problems", *International Journal of Foundations of Computer Science,* **12** pp.521-534 (2001).
15. Kacem, I.; Hammadi, S. and Borne, P. "Approach by localization and multi objective evolutionary optimization for flexible job-shap scheduling problems", *IEEE Transaction on Systems, Man and Cybernetics,* **32**(1), pp. 1-13 (2002).
16. Lee, Y.H.; Jeong, C.S. and Moon, C. "Advanced planning and scheduling with outsourcing in manufacturing supply chain", *Computer & Industrial Engineering,* **43**, pp. 351-374 (2002).
17. Kim, Y.K.; Park, K. and Ko, J. "A symbiotic evolutionary algorithm for the integration of process planning and

job shop scheduling", *Computers & Operations Research,* **30**, pp. 1151-1171 (2004).

18. Allet, S. "Handling flexibility in a "generalised job shop" with a fuzzy approach", *European Journal of Operational Research,* **147**, pp. 312-333 (2003).
19. Scrich, C.A.; Armentano, V.A. and Laguna, M. "Tardiness minimization in a flexible job shop: A taboo search approach", *Journal of Intelligent Manufacturing,* **15**, pp. 103-115 (2004).
20. Gen, M.; Cheng, R. *Genetic Algorithm in Search, Optimization and Machine Learning*, Addition-Wesley, Reading, MA, (2004).
21. Low, C. "Simulated annealing heuristic for flow shop scheduling problem with unrelated parallel machines", *Computer & Operation Research,* **32**, pp. 2013-2025 (2005).
22. Xia, W. and Wu, Z. "An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems", *Computers & Industrial Engineering,* **48**, pp. 409-425 (2005).
23. Torabi, S.A.; Karimi, B. and Fatemi-Ghomi, S.M.T. "The common cycle economic lot scheduling in flexible job shops: The finite horizon case", *Int. J. Production Economics,* **97**, pp. 52-65 (2005).
24. Xia J. and Wang, X. "Complexity and algorithms for two-stage flexible flowshop scheduling with availability constraints", *Computers and Mathematics with Applications,* **50**, pp. 1629-1638 (2005).
25. Tay, J.C. and Wibowo, D. "An effective chromosome representation for evolving flexible job-shop scheduling", Genetic and Evolutionary Computation Conference (2004).
26. Kyparisis, G.J. and Koulamas, C. "Flexible flow shop scheduling with uniform parallel machines", *European Journal of Operational Research,* **168**, pp. 985-997 (2004).
27. Ho, N.B.; Tay, J.C. and Lai, E. "An effective architecture for learning and evolving flexible job shap schedules", *European Journal of Operational Research,* **179**, pp. 316-333 (2007).
28. Gao, J.; Gen, M. and Sun, L. "Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm", *Journal Intelligent Manufacturing,* **17**, pp. 493-507 (2006).
29. Park, B.J.; Choi, H.R. and Kim, H.S. "A hybrid genetic algorithm for the job shop scheduling problems", *Computer & Industrial Engineering,* **45**, pp. 597-613 (2003).
30. Kurz, M.E. and Askin, R.G. "Scheduling flexible flowlines with sequence-dependent setup times", *European Journal of Operational Research,* **159,** pp. 66-82 (2004).
31. Kurz, M.E. and Askin, R.G. "Comparing scheduling rules for flexible flow-lines", *Int. J. Production Economics,* **85**, pp. 371-388 (2003).
32. Abbasian, M. and Nahavandi, N., *Minimization Flow Time in a Flexible Dynamic Job Shop with Parallel Machines,* Tehran, Tarbiat Modares University, Engineering Department of Industrial Engineering, Master of Science Thesis (2008).