

کمینه‌سازی بیشینه‌ی زمان در گردش کارها در مسئله‌ی کار کارگاهی 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)^۱ با سرعت‌های غیریکنواخت در محیط کار کارگاهی پویا (DJS)^۲ و در قالب مسئله‌ی کار کارگاهی انعطاف‌پذیر با ماشین‌های موازی در محیط پویا (FDJSPM)^۳، در نظر گرفته شد. پس از مدل‌سازی مسئله، الگوریتمی مبتنی بر اصول الگوریتم ژنتیک (GA)^۴ با کروموزوم‌های دوبعدی پویا، برای حل آن پیشنهاد شد. نتایج حاصل از اجرای الگوریتم پیشنهادی و مقایسه‌ی آن با یک روش فراابتکاری موجود در ادبیات، نشان‌دهنده‌ی بهبود جواب‌ها به میزان ۱/۳۴ درصد برای ابعاد مختلف مسئله است.

واژگان کلیدی: کار کارگاهی 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 استفاده مطلوب از امکانات و تجهیزات به‌شمار می‌رود. تحقیقات در این زمینه را می‌توان به دو قسمت - محیط‌های ایستا و محیط‌های پویا - تقسیم کرد. اغلب روش‌های بررسی‌شده در ادبیات برای حل مسائل زمان‌بندی ایستا کاربرد دارند و رخ داده‌های 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 مجموعه کارها افزوده شود، این سیستم ساخت و تولید را پویا می‌نامند.^[۱] به‌عبارت دیگر نمونه‌ی ایستای مسئله‌ی کار کارگاهی 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.^[۲] ثابت شده است که بررسی مدل‌های ایستا تر از مدل‌های پویاست و بررسی‌های گسترده‌ی نیز در مورد آنها انجام شده است.^[۱] تحقیق حاضر علاوه بر در نظر گرفتن 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 کار کارگاهی، 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 کوتاه و افزایش تنوع در محصولات شناخته می‌شوند. در این محیط‌ها، انعطاف‌پذیری 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

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، نه تنها توسط یک ایستگاه بلکه بر روی مجموعه‌ی 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 کار کارگاهی انعطاف‌پذیر با ماشین‌های موازی در محیط کار کارگاهی پویا (FDJSPM) مطالعه و بررسی شده است. در این مسئله، به تعداد m 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 نظر گرفته شده است و بنابراین هر یک از قطعات در یک زمان n (یک زمان غیرصفر و تصادفی) وارد کارگاه می‌شوند. از سوی دیگر، به‌منظور اجتناب از ایجاد گلوگاه و افزایش نرخ تولید، علاوه بر این که هر یک از ایستگاه‌های کاری قابلیت پردازش بیش از یک نوع قطعه را دارند (به‌سبب وجود انعطاف‌پذیری عملیات) هر ایستگاه کاری M ، مرکب از مجموعه‌ی L ماشین موازی است که با سرعت‌های غیریکنواخت S (و به‌سبب وجود انعطاف‌پذیری 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 کمیته‌سازی بیشترین زمان گردش کاری قطعات (F_{Max}) با در نظر گرفتن پویایی ایجادشده در کارگاه، انعطاف‌پذیری عملیات و انعطاف‌پذیری ناشی از ماشین‌های موازی با سرعت‌های غیریکنواخت است. در این نوشتار، برای نخستین بار دو پارامتر «انعطاف‌پذیری عملیات» و «انعطاف‌پذیری ناشی از ماشین‌های موازی با سرعت‌های غیریکنواخت» برای محیط کار کارگاهی پویا، توأم در نظر گرفته 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 کار 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 در باره‌ی 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.^[۵] با توجه به پیچیدگی مسئله، روش‌های تحلیلی فوق‌الذکر غالباً مسائل دارای یک ماشین را حل می‌کند. 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 اولین بار به‌منظور حل مسئله‌ی برنامه‌ریزی میان‌مدت به کار گرفته شد.^[۶] 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 هر کار وجود دارد که این به عدم انعطاف‌پذیری در سیستم تولید اشاره دارد. کار کارگاهی انعطاف‌پذیر (FJS)^۹ حالت توسعه‌یافته‌ی کار کارگاهی (JS) است که در آن هر عملیاتی توسط مجموعه‌ی ایستگاه‌ها 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. در FJS، 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 مجموعه‌ی ایستگاه‌های 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 نظر گرفت. در حقیقت نوع جدیدی از انعطاف‌پذیری عملیات در سیستم‌های کار کارگاهی 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 سه‌مرحله‌ی با ساختار ویژه‌ی یک ماشین در مراحل اول و سوم، و دو ماشین در مرحله‌ی دوم با هدف کمیته‌سازی

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 دوروش ابتکاری و فرابابتکاری مختلف - یکی بر پایه‌ی نظریه‌ی دوگان و دیگری بر پایه‌ی الگوریتم ژنتیک - را برای آن پیشنهاد داد. در این سال همچنین یک رویکرد ترکیبی از روش‌های فرابابتکاری برمبنای الگوریتم ژنتیک برای حل مسئله‌ی FJS به‌منظور کمینه‌سازی C_{Max} ارائه شد.^[۱۳] پس از آن در سال ۲۰۰۱، به‌منظور حل مسئله‌ی کار کارگاهی 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 مسئله‌ی مشابه 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 کار کارگاهی انعطاف‌پذیر به‌کمک روشی ابتکاری با رویکرد منطق فازی کردند.^[۱۸] در سال ۲۰۰۴ انعطاف‌پذیری در کار کارگاهی 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 مسئله‌ی که محققین در سال ۲۰۰۲ مطرح کرده بودند، ارائه شد.^[۲۰] در این تحقیق نخست زیرمسئله‌ی تخصیص توسط قاعده‌ی اولویت‌بندی SPT حل می‌شود، سپس بر این اساس که انتخاب یک بازنامی مناسب از کروموزوم‌ها مهم‌ترین مرحله در به دست آوردن جواب‌های با کیفیت است یک الگوریتم ژنتیک با بازنامی کروموزوم‌ها براساس عملیات 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 کار کارگاهی انعطاف‌پذیر روشی بیان شد.^[۲۳] در همین سال مجدداً یک روش عملی سلسله‌مراتبی کارا به‌منظور حل مسئله‌ی کار کارگاهی 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

با چرخش مجدد کار ارائه شد.^[۲۵] در سال ۲۰۰۶ محققین مسئله‌ی زمان‌بندی کار کارگاهی انعطاف‌پذیر با ماشین‌های موازی یک‌نواخت را با تابع هدف کمینه‌سازی C_{Max} مورد مطالعه قرار دادند.^[۲۶] آنها دسته‌یی از روش‌های فرابابتکاری موجود برای مسئله‌ی کار کارگاهی را گسترش دادند و نشان دادند که روش فرابابتکاری 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.^[۲۷] آنها عنوان کردند که انعطاف‌پذیری C٪، به‌این معنی است که تمامی عملیات‌های کارهای مختلف، می‌توانند حداکثر توسط C٪ تمامی ماشین‌های موجود در کارگاه پردازش شوند.^[۲۸] همچنین در سال ۲۰۰۷ در حل مسئله‌ی کار کارگاهی انعطاف‌پذیر با فرض گردش مجدد کارها،^[۲۹] محققین موضوع 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 توانمند 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-neighbor 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 بازنامی 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 جامع پیشینه‌ی تحقیق در زمینه‌های «کار کارگاهی»، «کار کارگاهی انعطاف‌پذیر» و «مسائل زمان‌بندی با ماشین‌های موازی با سرعت‌های یکسان و غیریکسان»، مشخص شد که در بسیاری از توالی‌های 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 در نظر گرفتن 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 خلأ‌های موجود، در این نوشتار، مسئله‌ی زمان‌بندی کار کارگاهی انعطاف‌پذیر با ماشین‌های موازی در محیط ساخت و تولید پویا (FDJSPM) مورد مطالعه قرار گرفته است. در این نوشتار، برای نخستین بار در پارامتر «انعطاف‌پذیری عملیات» و «انعطاف‌پذیری ناشی از ماشین‌های موازی با سرعت‌های غیریک‌نواخت» برای محیط کار کارگاهی پویا، تواماً در نظر گرفته شد. به‌عبارت دیگر در این نوشتار، علاوه بر پویایی کارگاه -- به‌سبب متغیر بودن

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 ناچیز است یا شامل زمان‌های پردازش می‌شود و زمان حمل و نقل قابل چشم‌پوشی است.
- بعضی از فرض‌های فوق عملاً غیر واقعی است و فقط برای ساده‌تر شدن مدل و امکان‌پذیر شدن آن در نظر گرفته شده است.

۲. تعریف متغیرها و مجموعه‌ها

- n : تعداد کارهای وارد شده به کارگاه در زمان‌های غیر صفر
- 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) $k = 1, \dots, m$
- M_k : کمترین مرحله‌ی پردازش
- $M_{k,r}$: کمترین ماشین موازی مرحله k ام
- $o_{i,j}$: کمترین عملیات کار j
- $St_{i,j}$: مرحله‌ی j 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
- $pm = 1, \dots, l_k$: شاخص ماشین‌های موازی در ایستگاه‌های کاری
- $S_{k,r}$: 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 $o_{i,j}$
- $F_{k,pm}^{(i,j)}$: زمان پایان عملیات $o_{i,j}$ روی ماشین $M_{k,pm}$
- $$a_k^{(i,j)} = \begin{cases} 1 & \text{اگر ماشین‌های } k \text{امین مرحله پردازش جزء ماشین‌های} \\ 0 & \text{علی‌البدل ممکن برای } o_{i,j} \text{ باشد برابر } 1 \text{ و در غیر این صورت برابر صفر در نظر} \\ & \text{گرفته می‌شود.} \end{cases}$$
- $$X_{k,pm}^{(i,j)} = \begin{cases} 1 & \text{اگر عملیات } o_{i,j} \text{ روی ماشین } M_{k,pm} \text{ پردازش شود برابر} \\ 0 & \text{و در غیر این صورت برابر صفر در نظر گرفته می‌شود.} \end{cases}$$
- $$R_{k,pm}^{(i,j)(p,q)} = \begin{cases} 1 & \text{اگر عملیات } o_{a,b} \text{ زودتر از عملیات } o_{i,j} \text{ روی ماشین} \\ 0 & \text{صورت‌پذیرد برابر } 1 \text{ و در غیر این صورت برابر صفر در نظر گرفته می‌شود.} \end{cases}$$

۳. مدل ریاضی

الف) تابع هدف مسئله

Minimization of Flow Time

$$F = \{Max \{C_i | i = 1, \dots, N\} - r_i\}$$

زمان‌های ورود قطعات به کارگاه -- مسئله‌ی ترکیبی از زمان‌بندی تولید کار کارگاهی با قابلیت انعطاف‌پذیری عملیات در ایستگاه‌های کاری و نیز وجود پردازنده‌های موازی با سرعت‌های غیر یکسان 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 به‌طوری‌که انجام هر کار در کارگاه پویا به مجموعه عملیات مشخصی نیازمند است. هر قطعه مانند قطعه‌ی i ام، برای پردازش در کارگاه در یک زمان غیر صفر (r_i) وارد کارگاه پویا می‌شود. قطعه‌ی J_i شامل زنجیره‌ی عملیات $\{o_{i,1}, o_{i,2}, \dots, o_{i,n_i}\}$ است که بدون از دست دادن کلیت مسئله می‌توان چنین فرض کرد که ترتیب انجام عملیات در کارگاه نیز به همین صورت باشد. ترتیب حرکت هر کار بین ایستگاه‌های کاری و برای کارهای مختلف متفاوت است (خصوصیت اصلی مسائل کار کارگاهی). تعداد عملیات لازم برای تکمیل هر کار ممکن است کوچک‌تر یا مساوی تعداد مراحل پردازش است. در هر مرحله از پردازش مانند M_k به تعداد l_k نسخه از ماشین‌های موازی با سرعت‌های مختلف وجود دارد که هر یک توانایی پردازش عملیات تخصیص یافته به آن مرحله را دارند و دست‌کم در یک ایستگاه کاری منعطف از کارگاه مقدار این پارامتر (l_k) بزرگ‌تر از ۱ است. هر عملیات دست‌کم توسط یکی از ایستگاه‌های کاری قابل پردازش است و به دلیل وجود انعطاف‌پذیری عملیات در کارگاه، حداقل یک عملیات وجود دارد به‌طوری‌که در بیش از یک ایستگاه کاری قابل پردازش باشد. عملیات هر یک از کارهای موجود در کارگاه را حداکثر با یک ماشین در یک زمان می‌توان انجام داد. طول زمان پردازش عملیات $o_{i,j}$ روی ماشینی با سرعت واحد ($S_{k,pm} = 1$) در مرحله‌ی k ام برابر $P_{pm,j,k}$ در نظر گرفته می‌شود و اگر پردازش این عملیات روی ماشینی با سرعت ($S_{k,pm} > 1$) 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 در بین کارهای موجود در صف انتظار هر ایستگاه) تجزیه می‌شود که زیرمسئله‌ی تخصیص نیز به نوبه‌ی خود به دو زیرمسئله‌ی تخصیص ایستگاه و تخصیص ماشین تجزیه می‌شود. زیرمسئله‌ی تخصیص به دلیل انعطاف‌پذیری (عملیات و ماشین‌های موازی) ایجاد شده 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 غیر صفر است.
۳. هر کار در هر لحظه تنها می‌تواند به وسیله‌ی یک ماشین پردازش شود.

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 کارگاه‌های دست‌کم دو محدودیت مشابه دارند که در صورت نقض آنها مسئله از حالت کارگاه‌های 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:

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

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

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

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

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

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

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

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

۴. شرح مدل

برای این که هر برنامه‌ی زمانی، راه‌حل امکان‌پذیر محدودیت‌های منطقی نیز باشد همه‌ی موارد عملیات انجام شده در چارچوب هر کار 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

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 هر کار در هر مرحله از ماشین‌های موازی، حداکثر روی یک ماشین و فقط یک‌بار پردازش می‌شود. این معادلات به‌دلیل لحاظ کردن انعطاف‌پذیری مسئله که ناشی از حضور ماشین‌های موازی با سرعت غیریکنواخت در ایستگاه‌های کاری کارگاه است ایجاد شده است. در کنار این معادلات، نامعادلات ۶ نیز بیان می‌کنند که اگر هیچ‌یک از ماشین‌های مرحله‌ی k ام به عملیات O_i اختصاص داده نشوند، زمان تکمیل این عملیات روی تمامی ماشین‌های این مرحله باید صفر در نظر گرفته شود. مجموعه‌ی این معادلات و نامعادلات برای در نظر گرفتن منعطف بودن مسئله به‌دلیل وجود عملیات انعطاف‌پذیر در کارگاه ایجاد شده است.

ب) از طرف دیگر چون تابع هدف مسئله (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 در نظر گرفتن پویایی مسئله به‌دلیل متفاوت بودن زمان ورود کارها به کارگاه ایجاد شده است.

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

۵. NP-hard بودن مسئله

پیچیدگی یک مسئله‌ی ترکیبی عبارت است از مقدار زمانی که در یک الگوریتم برای حل آن مسئله صرف می‌شود. مسئله FDJSPM، توسعه‌یافته‌ی مدل کارگاه‌های انعطاف‌پذیر (FDJSP) با در نظر گرفتن انعطاف‌پذیری ناشی از ماشین‌های موازی در یک محیط ساخت و تولید پویا است و در نتیجه همه‌ی پیچیدگی‌ها و سختی‌های آن را نیز به‌همراه خواهد داشت. چنانچه فقط انعطاف‌پذیری عملیات در این مسائل ملحوظ شود، حتی بر پیچیدگی یافتن جواب‌های بهینه‌ی تقریبی نیز به‌طور چشم‌گیری افزوده می‌شود.^[۱] از آنجا که مسئله FDJSP با انعطاف‌پذیری عملیات قویاً NP-hard است،^[۳] 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. رویه‌ی کلی الگوریتم ژنتیک پیشنهادی در شکل ۱ نشان داده شده است.

عوامل اصلی توسعه‌ی الگوریتم ژنتیک عبارت است از: نمایش کروموزوم‌ها، ایجاد جمعیت اولیه، معیار ارزیابی، عملکرد تقاطع، عملگر جهش و راهکار انتخاب. به علاوه پارامترهای الگوریتم از جمله 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 کار کارگاهی، یا هر مسئله‌ی بهینه‌سازی دیگر به کمک الگوریتم ژنتیک، بازنمایی جواب‌های مسئله به شکل کروموزوم است.^[۱۹] یکی از قابلیت‌های جست‌وجو به وسیله‌ی الگوریتم ژنتیک، امکان طراحی 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) به نحو 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 به لحاظ پیچیدگی محاسباتی مسئله و نیز اجتناب از به وجود 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 به طور یک پارچه و هم‌زمان قادر به حل هر دو زیرمسئله‌ی یادشده باشد. برای این منظور یک کروموزوم دوبعدی پویا - با در نظر گرفتن پویایی کارهای وارد شده به کارگاه - 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 نظر گرفته شده است. بنابراین هر جواب مسئله به صورت یک آرایه‌ی دوبعدی نمایش داده می‌شود. این روش مشابه روش به کار گرفته شده در سال ۲۰۰۲ است که البته در آن رشته‌ی تخصیص به صورت دو رشته‌ی مجزا و شامل رشته‌ی تخصیص ایستگاه کاری و نیز رشته‌ی تخصیص ماشین در نظر گرفته 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 تخصیص یافته 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 روبرو نمی‌شویم. به طور 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 به گونه‌ی مطرح شده که در آن، هر کار دارای ۳ عملیات و هر ایستگاه به ترتیب دارای ۲، ۳، ۲ و ۲ ماشین موازی یکسان است. همچنین کارگاه در عملیات به طور کامل انعطاف‌پذیر است. در این صورت کروموزوم متناظر با جواب دارای ۳ رشته به طول 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 به همین صورت قابل تفسیرند.

$$nTOP = \sum_{i=1}^n O_i$$

شماره عملیات	۱	۲	۳	...	nTOP
تعیین مرحله پردازش					
تعیین شماره ماشین					
تعیین ترتیب پردازش					

شکل ۲. بازنمایی کروموزوم‌ها در GA پیشنهادی.

مکان	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
اندیس عملیات	O_{11}	O_{11}	O_{13}	O_{11}	O_{21}	O_{21}	O_{23}	O_{21}	O_{23}	O_{31}	O_{31}	O_{33}
رشته تخصیص ایستگاه	M_1	M_1	M_3	M_1	M_2	M_2	M_3	M_2	M_3	M_1	M_1	M_3
رشته تخصیص ماشین	۳	۲	۱	۲	۱	۱	۱	۲	۳	۲	۳	۲
رشته توالی عملیات	۱	۸	۱۰	۵	۶	۹	۲	۳	۱۲	۴	۷	۱۱

شکل ۳. نمایش کروموزوم متناظر با جواب مسئله.

- 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 $P(t)$ 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.

۲. جمعیت اولیه

می‌گیرد. در صورت انجام موفقیت‌آمیز این آزمون، کروموزوم مذکور جهش می‌یابد. در الگوریتم ژنتیک پیشنهادی، به‌خاطر ساختار ویژه 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 ارائه شده برای عملگر جهش، در صورت مثبت بودن 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.

پس از تعیین روش تبدیل هر جواب به کروموزوم، باید جمعیت اولیه‌ی آن کروموزوم‌ها تولید شود. چگونگی تولید اعضای جمعیت اولیه نقش اساسی در مشخص کردن کیفیت جواب نهایی در هر روش جست‌وجوی محلی - از جمله در الگوریتم ژنتیک - دارد. تولید تصادفی جمعیت اولیه سبب حفظ تنوع کروموزوم‌ها در تولید جمعیت، کاهش احتمال هم‌گرایی 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 تصادفی 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.

۵. رویه‌ی انتخاب

در الگوریتم ژنتیک پیشنهادی از فضای نمونه‌گیری توسعه یافته استفاده می‌شود. این فضا چنین تعریف می‌شود: اگر اندازه‌ی جمعیت اولیه برابر α باشد و بعد از اعمال عملگرهای تقاطعی و جهشی به تعداد β 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 $Pop_Size + \mu$ 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 λ 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.

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 هم‌مکان ژن‌های کپی نشده در گام نخست، از کروموزوم P_2 در عملیات شرکت کرده و به همان مکان از O_1 منتقل خواهند شد.

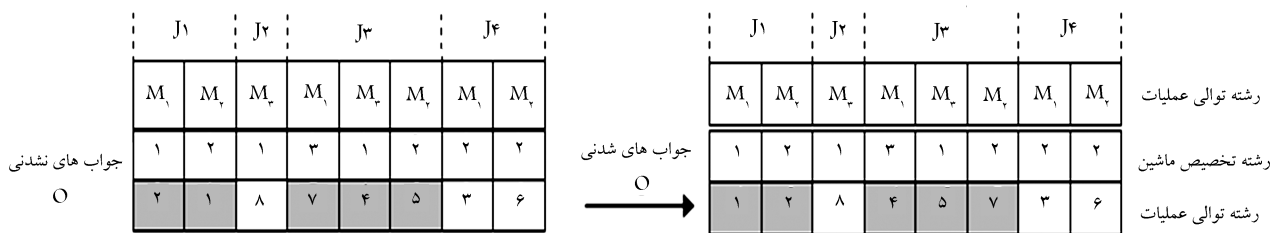
عملگر تقاطعی RMX بر روی رشته‌ی توالی عملیات (رشته‌ی سوم کروموزوم) اعمال می‌شود. این روش که توسط گلدبرگ و لینگل معرفی شد درحقیقت همان عملگر دو نقطه‌ی برش موجود در ادبیات است که برای استفاده در حالت‌های خاص بسط داده شده است. در این روش باید نخست هر دو کروموزوم P_1 و P_2 را در دو نقطه که به‌صورت تصادفی 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 کرد، البته تا جایی 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 زودرس الگوریتم جلوگیری می‌کند. به همین 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

۶. تابع برازندگی

تابع برازندگی میزان اثربخشی هر جواب منتخب یا هر زمان‌بندی ایجاد شده را اندازه‌گیری می‌کند. در مسئله‌ی مورد مطالعه‌ی ما، هدف کمیته‌سازی 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.

اول از رویکرد RKGا، و برای تخصیص کارها به ماشین‌ها در مراحل بعدی از ابداعات SPTCH (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 RKGا زمان بندی می‌شود و برای زمان بندی 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 RKGا و عملیات بعدی را با استفاده از همان ابداعات زمان بندی کردیم.^[۳۲]

۲. تولید مسائل تصادفی

برای تولید مسائل تصادفی نخست ۶ پارامتر مسئله را مطابق جدول ۱ شناسایی می‌کنیم. برای پنج پارامتر نخست از الگوریتم RKGا استفاده کردیم.^[۳۱،۳۰] و برای سرعت پردازش ماشین‌ها توزیع $U(۱,۳)$ 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 به صورت 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 هر کدام تولید می‌شود.

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

عوامل		سطوح عوامل	
تعداد کارها	ثابت	متغیر	۳۰-۱۰۰
توزیع ماشین‌ها	ثابت	متغیر	متغیر
تعداد ماشین‌ها	۲-۱۰	$U(۱,۴)-U(۱,۱۰)$	$U(۱,۴)-U(۱,۶)$
تعداد مراحل			۲-۴-۸
زمان‌های پردازش			$U(۵۰,۷۰)-U(۲۰,۱۰۰)$
سرعت پردازش ماشین‌ها			$U(۱,۳)$

برابر $f(i) = 1/F_{max}(i)$ تعریف شده است؛ $f(i)$ مقدار برازندگی کروموزوم نام $F_{max}(i)$ بیشینه زمان گردش کاری کارها در کروموزوم نام 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 ارائه شده راهکار اصلاحی برای برخورد با محدودیت‌ها به کار گرفته شده است. در این روش، کروموزوم غیرموجه به جای حذف شدن 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 نشان دادن کارایی الگوریتم پیشنهادی، از الگوریتم RKGا که در سال ۲۰۰۴ برای مسئله‌ی FSPM ارائه شد،^[۳۱،۳۰] استفاده کرده‌ایم. الگوریتم RKGا که یک الگوریتم فراابتکاری است، برای حل مسئله‌ی FSPM، با توجه به انعطاف پذیری ناشی از وجود ماشین‌های موازی با سرعت‌های یکسان و نیز انعطاف پذیری عملیات در بین ایستگاه‌های کاری طراحی و ارائه شده است. به همین دلیل، مسئله‌ی حل شده توسط این الگوریتم بیشترین شباهت را با مسئله‌ی FDJSPM دارد و به راحتی - بدون از دست دادن کلیت الگوریتم - برای این مسئله قابل استفاده است. بدین منظور در این مقایسه ما حالت خاص مسئله‌ی FDJSPM، یعنی همان مسئله‌ی JSPM را در نظر گرفته‌ایم.

ارائه دهندگان الگوریتم RKGا، برای حل مسئله‌ی خود از رویکرد سلسله‌مراتبی استفاده کرده و مسئله‌ی خود را به دو زیرمسئله‌ی تخصیص (اختصاص کارها به ایستگاه‌های منطف) 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 بر اثر انعطاف پذیری (عملیات و ماشین‌های موازی) ایجاد شده، به وجود آمده است.

در تحقیقات انجام شده در سال ۲۰۰۴، برای زمان بندی کارها در مرحله‌ی

۳. طراحی روش انجام آزمایش‌ها

در ادامه به منظور انجام مقایسه، الگوریتم‌های ارائه شده (هر دو الگوریتم پیشنهادی و الگوریتم RKGGA) به زبان C کد شده و با رایانه‌ی با مشخصات (of RAM Pentium IV (CPU ۳GHz, ۱GB در محیط Borland C++ ۵٫۰۲ اجرا می‌شوند. هر کدام از الگوریتم‌ها (GA پیشنهادی و RKGGA) با همان ۷۲۰ مجموعه اطلاعات تولید شده اجرا می‌شوند.

۴. تنظیم پارامترها

سطوح مختلف پارامترها بر کیفیت 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 (p_c) ، (p_m) و (μ') به کار خواهیم گرفت. برای این منظور از هر سناریوی آزمایشی که در آن ۱۰ مجموعه داده تولید شده، یک مجموعه و در کل ۷۲ مجموعه داده به صورت 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 (μ') : ۰٫۵، ۱٫۰، ۱٫۵، ۲٫۰ و ۲٫۵.

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

احتمال جهشی (p_m) : ۰٫۰۵، ۰٫۱، ۰٫۱۵، ۰٫۲، ۰٫۲۵ و ۰٫۳.

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 به جز 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 (μ') ، (p_c) و (p_m) به ترتیب ۰٫۷، ۰٫۲ و ۰٫۲ است. (جدول ۲)

نتایج آزمایش‌ها

در این مرحله، کارایی الگوریتم ژنتیک پیشنهادی با روش فراابتکاری RKGGA موجود در ادبیات، برای حل مسئله‌ی FDJSPM با تابع هدف کمینه‌سازی بیشترین زمان گردش کاری کارها برای حل ۲۱۶ سناریوی مختلف و هر کدام به تعداد ۱۰ بار (طبق

جدول ۲. مقادیر پارامترهای الگوریتم ژنتیک پس از تنظیم.

پارامتر	اندازه جمعیت اولیه	تعداد نسل‌ها	μ'	p_c	p_m
Value	۱۰۰	۱۲۵	۲۰٪	۰٫۷	۰٫۲

جدول ۱) مقایسه می‌شود. برای مقایسه‌ی این دو روش حل، از یک شاخص اصلی 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 بر اساس شاخص «کم‌ترین مقدار تابع هدف کمینه‌سازی بیشترین زمان گردش کاری کارها» برای ۱۰ اجرا هر یک از ۷۲ سناریو مسئله، در ابعاد کوچک، متوسط و بزرگ و به تفکیک تعداد مراحل، به ترتیب با میانگین ۰٫۵۷٪، ۱٫۰۲٪ و ۱٫۱۷٪ درصد بهبود و در کل با میانگین ۰٫۹۲٪ بهبود، برتری محسوسی نسبت به RKGGA دارد.

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

۳. روش پیشنهادی با توجه به شاخص «بهبود تابع هدف» در ابعاد کوچک، متوسط و بزرگ به طور میانگین و به ترتیب با ۵۰، ۴۶٫۷ و ۵۲ بار بهبود و در کل با میانگین ۴۹٫۶ بار و در مجموع با ۴۴۶ بار (از ۲۱۶ اجرا) یا ۲۰٫۶۵٪ درصد بهبود از RKGGA کارا تر است.

۴. روش پیشنهادی با توجه به شاخص میانگین تابع هدف در ابعاد کوچک، متوسط و بزرگ به طور میانگین و به ترتیب 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 RKGGA کارا تر است.

۵. با در نظر گرفتن شاخص میانگین زمان حل، روش پیشنهادی در ابعاد کوچک با RKGGA تفاوت چندانی ندارد ولی در ابعاد متوسط و بزرگ به ترتیب با میانگین ۱۴٫۵ و ۳۳٫۸ ثانیه افزایش و در کل با میانگین ۱۶٫۶ ثانیه افزایش، در مقایسه با RKGGA عملکرد نامطلوبی دارد.

معیار «تفاضل میانگین توابع هدف الگوریتم ژنتیک پیشنهادی با روش فراابتکاری RKGGA» برای ابعاد مختلف مسئله، شاخص مناسبی به منظور نشان دادن کارایی الگوریتم پیشنهادی است و نشان‌دهنده‌ی بهبود به میزان ۱٫۱۷٪، ۱٫۳۵٪ و ۱٫۴۹٪ و به طور 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 فراابتکاری RKGGA موجود در ادبیات افزایش می‌یابد.

در نهایت، با توجه به نتایج ملاحظه می‌شود که الگوریتم ژنتیک ارائه شده برای حل مسئله در مقایسه با RKGGA نیازمند زمان حل بیشتری است، اما با توجه به میزان بهبود انجام شده در شاخص‌های کم‌ترین مقدار و میانگین تابع هدف بر اساس 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 ارائه شده نسبت به روش RKGGA تأیید می‌شود.

جدول ۳. مقایسه ی الگوریتم ژنتیک پیشنهادی و الگوریتم RKGA.

ابعاد مسئله		کوچک			متوسط			بزرگ		
تعداد کار		۶			۳۰			۱۰۰		
تعداد مراحل		میانگین بعد کوچک	۲	۴	۸	میانگین بعد متوسط	۲	۴	۸	میانگین بعد بزرگ
روش RKGA	کمترین Fmax	۹۵,۶۲	۹۱,۸۵	۱۲۲,۶	۱۳۹,۴۷	۲۸۳,۹	۳۰۷,۱	۳۶۲,۷	۳۱۷,۸۹	
	میانگین Fmax	۲۰۲,۸۶	۵۲۷,۴۳	۶۴۶,۸۳	۸۴۵,۲۳	۶۷۳,۱۶	۱۷۴۳,۸۰	۱۹۰۵,۸۸	۲۰۸۷,۵۳	
	میانگین زمان حل (ثانیه)	۰,۰۲۹	۰,۲۰۱	۰,۲۴۸	۰,۲۲۷	۱,۶۰۲	۱,۵۵۱	۱,۷۴۵	۱,۶۳۳	
الگوریتم ژنتیک پیشنهادی	کمترین Fmax	۹۴,۸۵	۹۳,۳۵	۱۱۸,۰۱	۱۳۷,۷۹	۲۷۸,۱۲	۳۰۷,۹۸	۳۵۶,۲۹	۳۱۴,۱۳	
	میانگین Fmax	۲۰۰,۰۱	۵۲۷,۶۹	۶۳۵,۲۱	۸۲۵,۸۷	۶۶۲,۹۲	۱۷۳۱,۳۲	۱۸۷۰,۸۹	۲۰۵۴,۹۳	
	میانگین زمان حل (ثانیه)	۰,۰۷۷	۰,۷۳۵	۲,۱۵۲	۷,۹۸۲	۳,۶۲۳	۲,۶۵۳	۱۸,۷۴۵	۵۹,۸۸	
میزان بهبود (درصد)	کمترین Fmax	٪۰,۵۷	٪-۱,۶۳	٪۳,۷۳	٪۰,۹۶	٪۱,۰۲	٪-۰,۲۸	٪۱,۷۶	٪۱,۱۷	
	میانگین Fmax	٪۱,۱۷	٪-۰,۰۵	٪۱,۸۰	٪۲,۲۹	٪۱,۳۵	٪۰,۷۲	٪۱,۸۴	٪۱,۴۹	
تعداد دفعات بهبود	کمترین Fmax	۵۰	۲۳	۶۶	۵۱	۶۲	۴۱	۵۳	۵۲	
	میانگین Fmax	۱۳,۳	۱۰	۱۹	۱۴	۱۸	۱۵	۱۶	۱۶,۳	
بهبود زمان حل (ثانیه)		-۰,۷۵	-۳,۱۷۱۴	-۱,۴	-۲,۶۵۶۷	-۷,۶۷۷۴	-۱۴,۵	-۱۱,۰۸۵۸	-۸۹,۶۸۰۲	

* کمترین مقدار Fmax، از میان ۷۰ اجرا از سناریوها که بر اساس تعداد مراحل کاری تفکیک شده اند، انتخاب شده اند و شامل مقادیر یکسان با RKGA نیز نمی شوند.
 ** کمترین مقدار میانگین Fmax از میان ۲۱ اجرا از سناریوها که بر اساس تعداد مراحل کاری تفکیک شده اند انتخاب شده اند، و شامل مقادیر یکسان با RKGA نیز نمی شوند.
 اعداد منفی جدول نشان دهنده برتری جواب حاصل از RKGA نسبت به روش حل پیشنهادی است.

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

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

نتیجه گیری

در این نوشتار یک مسئله ی ترکیبی از زمان بندی کار کارگاهی انعطاف پذیر با در نظر گرفتن ماشین های موازی (با سرعت های پردازش غیر یکسان) در هر ایستگاه کاری، برای یک سیستم ساخت و تولید پویا (به سبب متغیر بودن زمان های ورود کارها به کارگاه) معرفی شد. سپس یک مدل ریاضی از نوع برنامه ریزی غیرخطی عدد صحیح مختلط برای مسئله ارائه شد. سپس کمینه سازی تابع F_{max} به عنوان هدف زمان بندی و به منظور بهره برداری کارا از ماشین آلات از طریق کاهش مدت زمان استفاده از منابع درگیر و افزایش نرخ و سرعت فرایند ساخت - که یکی از اهداف مهم در زنجیره ی تأمین ساخت و تولید است - مورد بررسی قرار گرفت. برای حل مسئله از قابلیت های الگوریتم ژنتیک در حل مسائل زمان و زمان بندی استفاده شد. به دلیل پویایی کارگاه، طراحی کروموزومها به گونه ای انجام شد که پویایی کارها نیز در آنها مد نظر قرار گیرد؛ به عبارتی دیگر کروموزوم های دوبعدی پویا برای اجرای الگوریتم طراحی شد. طراحی کروموزوم و انتخاب عملگرهای ژنتیک به گونه ای صورت گرفت که در برخورد با محدودیت ها و تولید کروموزوم غیرموجه کمترین مشکل ایجاد شود. در الگوریتم ژنتیک پیشنهادی، جمعیت اولیه به صورت تصادفی تولید شده و فاقد

این مقایسه، نشان‌دهنده‌ی بهبود به میزان ۱۷٪، ۳۵٪ و ۴۹٪ و به‌طور متوسط ۳۴٪ برای ابعاد کوچک، متوسط و بزرگ از مسئله است. بنابراین با توجه به نتایج حاصل از آزمایش‌های عددی، در مجموع کارایی روش حل پیشنهادی نسبت به روش RKGا تأیید می‌شود. از جمله 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.

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 RKGا موجود در ادبیات مورد مقایسه قرار گرفت و ملاحظه شد که الگوریتم ژنتیک ارائه شده برای حل مسئله‌ی تحقیق در مقایسه با RKGا نیازمند زمان حل بیشتری است، اگرچه با توجه به میزان بهبود انجام شده در شاخص‌های کم‌ترین مقدار و میانگین تابع هدف براساس میزان و تعداد دفعات بهبود، این افزایش زمان قابل توجیه است. نتایج

پانویس

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-shop 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 shop 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 flow-lines 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).