

زمان بندی مقاوم و پایدار برای محیط کار کارگاهی منعطف با شکست تصادفی ماشین

مصطفی زندیه^۱، احسان احمدی^۲

چکیده: پژوهش پیش رو، رویکردی را به منظور ایجاد زمان بندی مقاوم و پایدار برای محیط کار کارگاهی منعطف، زمانی که شکست تصادفی ماشین وجود دارد، پیشنهاد می کند. به منظور بررسی وضعیت شکست ماشین از شبیه سازی استفاده شد که برای دستیابی به زمان بندی مقاوم و پایدار به کمک الگوریتم های فراابتکاری یکپارچه شده است. الگوریتم پیشنهادی دو مرحله را دربرمی گیرد. در مرحله اول، از آنجاکه زمان تکمیل برنامه اولین هدف هر برنامه زمان بندی است، این شاخص بهبود می یابد و سپس در مرحله دوم سه شاخص زمان تکمیل برنامه، مقاومت و پایداری به صورت خطی ترکیب شده و تابع هدف را تشکیل خواهد داد. در مدل پیشنهادی، برنامه ریز می تواند میزان اهمیت هریک از شاخص ها را در تابع ترکیب خطی مشخص کند و مسیر بهبود الگوریتم را در جهت شاخص های مد نظر تغییر دهد. نتایج محاسباتی نشان می دهد که دستیابی به زمان بندی مقاوم و پایدار بدون افت در شاخص زمان تکمیل برنامه امکان پذیر است. در نهایت از آزمون فرضیه آماری به منظور مقایسه عملکرد دو الگوریتم فراابتکاری استفاده شده است.

واژه های کلیدی: الگوریتم تکامل تفاضلی، الگوریتم ژنتیک، زمان بندی مقاوم و پایدار، شکست ماشین، کار کارگاهی منعطف.

۱. دانشیار مدیریت صنعتی، دانشگاه شهید بهشتی، تهران، ایران

۲. کارشناس ارشد مدیریت صنعتی، دانشگاه شهید بهشتی، تهران، ایران

تاریخ دریافت مقاله: ۱۳۹۲/۰۹/۰۳

تاریخ پذیرش نهایی مقاله: ۱۳۹۳/۰۳/۰۵

نویسنده مسئول مقاله: احسان احمدی

E-mail: Ahmady.Ehsan@gmail.com

مقدمه

امروزه انعطاف‌پذیری ساخت و تولید به صورت گسترده‌ای عامل اساسی برای دستیابی به مزیت رقابتی شناخته شده است. مسئله زمان‌بندی کار کارگاهی منعطف، یکی از مهم‌ترین مسائل مطرح در مدیریت تولید است و از سخت‌ترین مباحث بهینه‌سازی ترکیباتی به شمار می‌رود، اما به دلیل پیچیدگی محاسباتی دستیابی به جواب بهینه برای این مسئله با روش‌های متداول بهینه‌سازی، بسیار دشوار است. تا کنون در مسائل زمان‌بندی مطالعه شده فرض می‌کردند تمام پارامترهای برنامه از قبل مشخص است و هیچ‌گونه رخداد تصادفی وجود نخواهد داشت، یا در صورت بروز واقعه غیر منتظره‌ای، از قبل پیش‌بینی‌شدنی است. بنابراین چنین فرضی بازتاب مسئله‌ای با نام وقایع پیش‌بینی‌نشده که در سیستم‌های تولید واقعی اتفاق می‌افتد، نیست. شکستگی تصادفی ماشین یکی از وقایعی است که می‌تواند در هر محیط تولیدی رخ دهد و تمام برنامه‌ریزی‌های پیش‌بینی‌شده را با چالشی بزرگ مواجه کند. بنابراین دستیابی به زمان‌بندی‌ای که هنگام اجرا بتواند این رخدادهای خارجی را جذب کند و حداقل هزینه را به سازمان تحمیل کند، حیاتی خواهد بود.

هنگامی می‌توان به حداقل هزینه دست یافت که بعد از شکست ماشین و انجام زمان‌بندی مجدد، کمترین انحراف در زمان‌بندی هم از نظر مدت تکمیل کار (شاخص زمان تکمیل برنامه) و هم از نظر تخصیص عملیات به ماشین‌ها و زمان تکمیل هر عملیات (شاخص مقاومت و پایداری) بین دو زمان‌بندی اولیه (زمان‌بندی پیش‌گویانه) و زمان‌بندی بعد از شکست ماشین وجود داشته باشد. به بیان ساده‌تر، به زمان‌بندی‌ای نیازمندیم که بتواند آثار شکست ماشین را حذف کند.

دستیابی به زمان‌بندی پیش‌گویانه، به معنای ارضای دو هدف ناسازگار است، اول اینکه باید به‌طور مؤثر از منابع استفاده شود و دوم باید اجازه انعطاف‌پذیری کافی برای تغییرات را بدهد. این موضوع با هدف اولیه کمینه‌کردن زمان تکمیل کارها و سپس هدف دوگانه مقاومت و پایداری دست‌یافتنی است.

پژوهش پیش رو به بررسی مقاومت و پایداری زمان‌بندی کار کارگاهی انعطاف‌پذیر می‌پردازد؛ جایی که انتظار می‌رود اختلال‌ها در شکست ماشین ایجاد شود. هدف اصلی این پژوهش ارائه مدلی است که بتواند سه معیار متضاد زمان تکمیل برنامه، مقاومت و پایداری را به صورت هم‌زمان بهینه کند. برای دستیابی به این هدف از رویکردهای فرا ابتکاری و شبیه‌سازی برای رسیدن به زمان‌بندی مد نظر، استفاده شده است.

پیشینه نظری پژوهش

کار کارگاهی منعطف (FJS)^۱

کار کارگاهی منعطف، یکی از فراگیرترین سیستم‌های تولید قطعات متنوع است که تعمیم‌یافته دو سیستم کلاسیک کار کارگاهی و کارگاه ماشین‌های موازی است. این محیط تولیدی شامل چند ایستگاه کاری می‌شود که در هر مرحله یک یا بیش از یک ماشین موازی وجود دارد. از آنجا که مسئله زمان‌بندی کار کارگاهی منعطف توسعه‌یافته کار کارگاهی است، پس مسئله مذکور نیز NP-Hard است.

هیورینک، جوربیچ و ثوله (۱۹۹۴) الگوریتم جست‌وجوی ممنوعی (TS) ارائه دادند که در آن تخصیص مجدد و توالی مجدد، به‌مثابه دو نوع جهش متفاوت به کار گرفته شده است. مسقونی، حمدی و بورنه (۱۹۹۷) برای حل این مسئله الگوریتم ژنتیکی را توسعه دادند و جزئیات عملکرد الگوریتم را با مثالی تشریح کردند.

کوولیس، دانیل و وایرکتاراکیس (۲۰۰۰)، مونتمانی (۲۰۰۷)، ویو و کوایان (۲۰۰۹) و لیو، گایو و زی (۲۰۰۷)، کارگاه تک‌ماشینه‌ای را با توجه به شکست ماشین در نظر گرفتند. علاوه‌براین، سواکس و سورنسن (۲۰۰۴) الگوریتم ژنتیک اصلاح‌شده‌ای را برای یافتن جواب‌های مقاوم در محیط تک‌ماشینه با توجه به زمان ارسال تصادفی کارها، بررسی کردند.

کوتانوکلو و ویو (۲۰۰۴) به‌منظور تقسیم مسئله کلاسیک زمان‌بندی کار کارگاهی با زمان فرایند نامشخص به یک‌سری از زیرمسئله‌ها، نوعی تجزیه ابتکاری را به کار بردند و به‌صورت تکراری پارامترهای مسئله را برای تحلیل آثار نوسان زمان فرایند با استفاده از اطلاعات پیشین بررسی کردند.

شفایی و برون (۲۰۰۰) از رویکرد زمان غلتان برای بررسی مقاومت زمان‌بندی استفاده کردند. جنسن (۲۰۰۱ و ۲۰۰۳) برای بهبود مقاومت و انعطاف‌پذیری زمان‌بندی کار کارگاهی زمانی که بیشترین تأخیر، مجموع تأخیرها، کل زمان فرایند و زمان کل ساخت را حداقل می‌کند، الگوریتم فراابتکاری‌ای پیشنهاد کرد. در هر دو مطالعه ضمن استفاده از الگوریتم ژنتیک توسعه‌داده‌شده متفلد (۱۹۹۶)، دو شاخص مقاومت (شاخص مقاومت بر مبنای همسایگی^۲ و شاخص مقاومت بر مبنای تأخیر^۳) در نظر گرفته شده است.

آنگلانی و گرایکو، گیوریرو و میوسمانو (۲۰۰۵) مدل ریاضی فازی از زمان‌بندی ماشین‌های موازی با هزینه وابسته به توالی در شرایط عدم قطعیت در زمان‌های پردازش را در نظر گرفتند.

1. Flexible Job Shop
2. Neighborhood-base
3. Lateness-base

بیویاچیا، بلالونا، جایلت و قدیرا (۲۰۱۰) تعمیم احتمالی‌ای برای طراحی مقاومت زمان‌بندی پیشنهاد کردند که فرض می‌کند بخشی از کارها روی ماشین‌های موازی به‌عنوان متغیر تصادفی با توجه به وزن کل زمان جریان، پردازش می‌شود.

الچینایی و المکاوای (۲۰۱۱ الف) الگوریتمی را برای شکست ماشین پیشنهاد کردند که در آن سطح خرابی کارگاه را در نظر نگرفتند، آنها از توزیع یکنواخت برای شکست ماشین استفاده کردند.

مسئله زمان‌بندی در کار کارگاهی منعطف، شامل دو زیرمسئله مسیریابی و تعیین توالی است. در زیرمسئله مسیریابی، هر عملیات به ماشینی از مجموعه ماشین‌های امکان‌پذیر تخصیص می‌یابد و زیرمسئله تعیین توالی، شامل تعیین توالی عملیات تخصیص‌یافته به ماشین‌ها برای دستیابی به زمان‌بندی امکان‌پذیر بهینه است.

انعطاف‌پذیری FJSP به دو دسته زیر تقسیم می‌شود:

۱. انعطاف‌پذیری کلی (T-FJSP): زمانی است که هر عملیات می‌تواند روی تمام

ماشین‌های درون کارگاه پردازش شود.

۲. انعطاف‌پذیری بخشی (P-FJSP): زمانی است که هر عملیات می‌تواند روی یک یا

مجموعه‌ای از ماشین‌های درون کارگاه پردازش شود (کاصم، حمدی و برونه ۲۰۰۲ الف).

فرضیه‌های مسئله زمان‌بندی کار کارگاهی منعطف

کارها مستقل از یکدیگرند و در زمان صفر در دسترس‌اند. مکان پردازش دو عملیات از یک کار به‌صورت هم‌زمان، وجود ندارد. هر کار دارای توالی عملیات ثابتی است. هر ماشین در هر لحظه فقط یک عملیات را می‌تواند انجام دهد. ماشین‌ها ممکن است بیکار شوند. ماشین‌ها، هرگز خراب نمی‌شوند و در تمام دوره زمان‌بندی در دسترس‌اند. زمان‌های آماده‌سازی، مستقل از توالی عملیات هستند. از زمان جابه‌جایی کارها بین ماشین‌ها چشم‌پوشی می‌شود. انبار کار در حین ساخت، مجاز است. همه پارامترها معلوم‌اند و تصادفی نیستند.

زمان‌بندی مقاوم و پایدار

با توجه به مرور ادبیات، رویکردهایی که برای دستیابی به زمان‌بندی مقاوم استفاده شده است به دو دسته اصلی تقسیم می‌شوند: ۱. رویکرد حفظ کیفیت جواب^۳ و ۲. رویکرد متمایل به‌اجرا^۴.

-
1. Total Flexibility
 2. Partial Flexibility
 3. Preservation of solution quality approach
 4. Execution-oriented quality approach

در رویکرد اول، مقاومت به معنای توانایی حفظ سطحی از کیفیت جواب در نظر گرفته می‌شود؛ مانند حفظ زمان تکمیل برنامه در مطالعه لئون، ویو و استورر (۱۹۹۴) و جنسن (۲۰۰۳) یا حفظ تأخیرات و کل زمان در جریان در مطالعه جنسن (۲۰۰۱).

در رویکرد متمایل به اجرا، زمان‌بندی مقاوم از طریق ایجاد بخشی از زمان‌بندی امکان‌پذیر است و زمان‌بندی نهایی به زمان اجرا یا نزدیک به اجرا به تأخیر انداخته می‌شود. این موضوع در پژوهش‌های دیگری نیز بررسی شده است (کوتانگولا، ۱۹۹۸ و پلیسلا، اودی، اسمیت و سستا، ۲۰۰۴). بنابراین دو تعریف برای زمان‌بندی مقاوم وجود دارد:

۱. زمان‌بندی ای مقاوم در نظر گرفته می‌شود که هنگام مواجهه با اختلال از جهش به راست^۱ به مثابه الگوریتم زمان‌بندی مجدد استفاده می‌کند و در مقایسه با زمان‌بندی‌های دیگر هزینه کمی دارد (جنسن، ۲۰۰۳).

۲. زمان‌بندی ای مقاوم در نظر گرفته می‌شود که بتواند اختلال‌های خارجی را بدون از دست دادن سازگاری، جذب کند و این تغییر روی کل اجزای زمان‌بندی اثری نداشته باشد (پلیسا و همکاران، ۲۰۰۴).

بیشتر مسائل زمان‌بندی شامل FJSP، مسائل NP-HARD در نظر گرفته می‌شود. بنابراین رویکردهای ابتکاری و فرا ابتکاری در ادبیات بیشتر در کانون توجه قرار گرفته است. در ادامه به خلاصه مطالعاتی اشاره می‌شود که بر زمان‌بندی تصادفی تمرکز کرده‌اند. مطالعات بسیاری از جمله کوولیس (۲۰۰۰)، موتمانی (۲۰۰۷) و وی (۲۰۰۹)، به بررسی مسئله تصادفی تک‌ماشینه با زمان‌های پردازش کار نامشخص همراه با شکست ماشین پرداخته‌اند. لئون و همکاران (۱۹۹۴) شاخص مقاومت زمان تأخیرمحوری را برای تحلیل اثرهای شکست ماشین و تغییرات زمان پردازش روی کیفیت جواب مسئله کار کارگاهی کلاسیک، پیشنهاد کردند. شاخص سنجش مقاومت پیشنهادی آنها به شرح رابطه ۱ است.

$$Z_{r=1}(s) = MS_{min} - RD3(S) \quad \text{رابطه ۱}$$

که در آن MS_{min} زمان تکمیل کار از زمان‌بندی S و $RD3(S)$ میانگین تأخیر عملیات در زمان‌بندی S است.

بیون (۲۰۰۱) نوعی تجزیه ابتکاری را به منظور تقسیم مسئله زمان‌بندی کار کارگاهی کلاسیک با زمان‌های پردازش نامشخص به یک سری از مسائل جزئی به کار برد. وی به صورت تکراری، پارامترهای مسئله را برای تحلیل زمان‌های پردازش متغیر، به روزرسانی کرد. شفایی و

برون از رویکرد زمان گردش برای بررسی مقاومت زمان بندی استفاده کردند. جنسن (۲۰۰۱) تلاش کرد که مقاومت و انعطاف پذیری زمان بندی کار کارگاهی را به منظور کمینه کردن حداکثر تأخیرها، مجموع تأخیرها، کل زمان در جریان و زمان تکمیل کار، بهبود دهد. در هر دو مطالعه، ضمن استفاده از الگوریتم ژنتیک، دو شاخص مقاومت در نظر گرفته شده است: ۱. شاخص مقاومت همسایگی محور و ۲. شاخص مقاومت تأخیرمحور. او شاخص مقاومت همسایگی $N_1(S)$ را به صورت رابطه ۲ تعریف کرد.

$$R_{C_{msp}}(S) = \frac{1}{|N_1(S)|} \sum_{S' \in N_1(S)} C_{msp}(S') \quad \text{رابطه ۲}$$

که $N_1(S)$ شامل زمان بندی S و تمام زمان بندی های موجه است و می تواند از از کبه طریق جابه جایی دو عملیات متوالی روی یک ماشین ایجاد شود و $C_{msp}(S')$ زمان تکمیل کار زمان بندی S' است. پولیسلا، سستا، اودی و اسمیت (۲۰۰۵) از رویکرد دومرحله ای برای ایجاد یک زمان بندی انعطاف پذیر بخشی برای مسائل زمان بندی منابع محدود با حداقل و حداکثر زمان تأخیر استفاده کرد. رنگساریتراسامی، فرل و کیورتز (۲۰۰۴) روشی برای زمان بندی مجدد بر اساس الگوریتم ژنتیک جست و جوی محلی برای مسئله کار کارگاهی با ورود کارها به صورت پویا پیشنهاد کردند. الگوریتم پیشنهادی آنها همزمان کارایی را با حفظ زمان تکمیل برنامه^۱ و تأخیرها و پایداری را از طریق حداقل سازی انحرافات در زمان شروع کار، در نظر می گیرد. در مطالعه آنها زمان بندی مجدد در بازه های زمانی مشخص و با استفاده از تمام کارهای در دسترس، انجام می گیرد. ماتسویچوک، سوتسکوف و اگوروا (۲۰۰۹) چارچوبی برای تصمیم گیری دومرحله ای، به منظور زمان بندی کارگاه جریان کاری دومرحله ای با وقفه در زمان پردازش پیشنهاد کردند.

روش شناسی پژوهش

تطبیق الگوریتم ژنتیک برای مسئله مورد مطالعه. الگوریتم ژنتیک روش فرا ابتکاری مؤثری برای حل مسائل بهینه سازی ترکیباتی مانند FJSP است. انواع الگوریتم های ژنتیکی که برای حل FJSP پیشنهاد شده اند، در روش نمایش کروموزوم، روش ایجاد جمعیت اولیه و راهبردهای ایجاد فرزند، با یکدیگر تفاوت دارند.

کدینگ کروموزوم. نمایش کروموزوم مناسب، بر موفقیت استفاده از GA تأثیر بسزایی دارد. چنگ، گن و تسوجیمورا (۱۹۹۶) پژوهشی درباره انواع روش های مختلف نمایش کروموزوم برای

حل مسائل کلاسیک JSP ارائه کردند. از نتایج پژوهش آنها و محققان دیگری مانند هو، تای و لای (۲۰۰۷)، می‌توان نتیجه گرفت روش نمایش کروموزوم به صورت عملیات‌محور^۱ کل فضای جست‌وجو را پوشش می‌دهد و هر ترکیبی از عملیات، جواب موجه ایجاد می‌کند. این روش نمایش به شکل ماتریسی است که در هر سطر آن رشته‌ای از ترکیب سه‌تایی (i, j, k) به نام ژن قرار گرفته است. i بیان‌کننده کار فعلی است؛ j شماره عملیات مربوط به کار i است و k شماره ماشینی است که به عملیات j تخصیص داده می‌شود. این روش نمایش، دو مرحله تخصیص ماشین و تعیین توالی را در یک ژن ساده یکپارچه می‌کند. مزیت دیگر استفاده از این روش نمایش کروموزوم، توانایی آن در مدل‌سازی مسیرهای تولید جایگزین با استفاده از تغییر شاخص k است. طول کروموزوم برابر با کل تعداد عملیاتی است که باید زمان‌بندی شود. برای مثال ماتریس شکل ۱ نمونه‌ای از نمایش کروموزوم برای سه کار و سه ماشین در زمان‌های پردازش جدول ۱ است.

جدول ۱. زمان‌های پردازش FJSP برای سه کار و سه ماشین

| J | O_{ij} | M_1 | M_2 | M_3 |
|-------|----------|-------|-------|-------|
| J_1 | O_{11} | ۳ | ۴ | ۱ |
| | O_{12} | ۴ | ۱ | - |
| J_2 | O_{21} | ۱ | ۱ | ۵ |
| | O_{22} | ۱ | - | ۲ |
| | O_{23} | - | ۲ | ۲ |
| J_3 | O_{31} | ۲ | ۲ | ۲ |
| | O_{32} | ۱ | ۲ | ۲ |
| | O_{33} | - | ۱ | ۱ |

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

شکل ۱. نمایش کروموزوم

جمعیت اولیه. کارایی الگوریتم‌های فرا ابتکاری که از جواب‌های اولیه به‌منزله نقطه شروع استفاده می‌کنند، با ایجاد جواب‌های بهتر در جمعیت اولیه می‌توانند بهبود یابند. در این پژوهش از رویکرد ابتکاری پیشنهادشده الحینایی (۲۰۱۱) به نام Ini-Pop Gen استفاده شده است. در این رویکرد، زمان‌های پردازش و همچنین بار کاری ماشین، هنگام تخصیص عملیات به ماشین در نظر گرفته می‌شود. بنابراین یک عملیات لزوماً به ماشینی با زمان پردازش کمتر تخصیص داده نمی‌شود، بلکه به ماشینی تخصیص داده می‌شود که می‌تواند آن را زودتر از ماشین‌های دیگر انجام دهد. این رویه برای ایجاد جواب‌های اولیه با در نظر گرفتن تابع زمان تکمیل برنامه، مؤثر است.

انتخاب. انتخاب عامل مهمی در الگوریتم ژنتیک است. این فرایند بدین منظور انجام می‌گیرد که افراد^۱ برای تولید مجدد انتخاب شوند. در این پژوهش از روش چرخ رولت برای انتخاب‌دهنده و گیرنده و انتقال آنها به حوزه جفت‌گیری استفاده شده است. چرخ رولت برای انتخاب کروموزوم‌ها یا افرادی به کار می‌رود که بر اساس احتمال انتخاب رتبه‌بندی شده‌اند. احتمال انتخاب هر کروموزوم بر اساس رابطه^۳ محاسبه می‌شود.

$$ind = 1, 2, 3, \dots, Npop \quad P_s = \frac{F_{ind}}{F_{tot}} \quad \text{رابطه ۳}$$

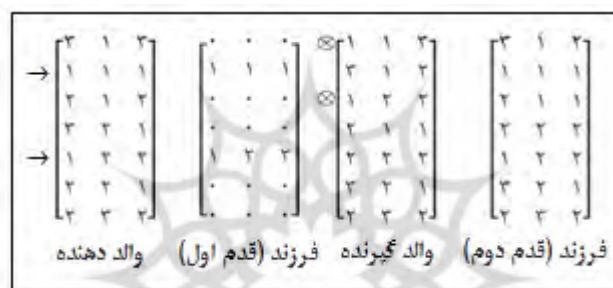
که در آن؛ P_s احتمال انتخاب ind ام فرد، $Npop$ اندازه جمعیت، F_{ind} تابع برازش ind امین فرد و F_{tot} تابع برازش کل افراد موجود در جمعیت است.

اپراتورهای ژنتیک. عملکرد الگوریتم ژنتیک برای تکامل، به عملکرد اپراتورهای ژنتیک استفاده‌شده در آن بستگی دارد (چنگ، ۱۹۹۶). بنابراین طراحی اپراتورهای ژنتیک برای موفقیت هر الگوریتم ژنتیکی حیاتی است. از آنجا که در روش نمایش جایگشت‌محور، دو زیر مسئله توالی و تخصیص با استفاده از سه‌تایی (کار، عملیات، ماشین) به‌صورت یکپارچه در نظر گرفته می‌شود، باید اپراتوری که محدودیت‌های مسئله را در نظر می‌گیرد به‌دقت طراحی شود، در غیر این صورت به سازوکار اصلاحی نیاز خواهد بود. بدین منظور اپراتورهای پیشنهادی POX^۲ (کاظم و همکاران، ۲۰۰۲ الف)، PBM^۳ و MBM^۴ پیشنهادی متفلد (۱۹۹۶) انتخاب شده است.

-
1. Individuals
 2. Precedence preserving order based crossover
 3. Position Based Mutation
 4. Machine Based Mutation

عملگر تقاطع. در پژوهش حاضر از روش پیاده‌سازی عملگر تقاطع POX پیشنهادی الحینایی و المکاوی (۲۰۱۱ الف) استفاده شده است؛ بدین صورت که ابتدا ژنی به صورت تصادفی از والد دهنده انتخاب می‌شود و تمام ژن‌هایی که در بردارنده عملیات مربوط به کاری که در ژن انتخاب شده وجود دارد، به فرزند منتقل می‌شود. در حین کار پس از یافتن هر عملیات مشابه در والد گیرنده، آن عملیات حذف خواهد شد. در مرحله دوم ژن‌های باقی‌مانده فرزند از طریق انتقال عملیات باقی‌مانده والد دهنده به ژن‌های فرزند، به همان ترتیب موجود تکمیل می‌شود.

شکل ۲ فرایند کار برای دو کروموزوم ایجادشده تصادفی را در مسئله 3×3 (جدول ۱) نشان می‌دهد.



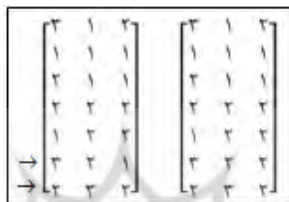
شکل ۲. نحوه کار عملگر POX

فرض کنید ژن $[1 \ 1 \ 1]$ از والد دهنده به صورت تصادفی انتخاب شده است. حال باید تمام ژن‌هایی که در والد دهنده دربرگیرنده عملیات مربوط به کار ۱ است، انتخاب شود. دو ژن انتخاب شده در والد دهنده با پیکان نشان داده شده است و علامت \otimes عملیات مشابه در والد گیرنده را نمایش می‌دهد. ژن‌هایی که در والد گیرنده با علامت \otimes نمایش داده شده است باید حذف شوند. سپس عملیات مربوط به کارها به صورت یکسان از والد به فرزند به همان ترتیب منتقل می‌شود، بنابراین از محدودیت توالی کارها تجاوز نمی‌شود و کروموزوم ناموجهی ایجاد نخواهد شد، پس نیازی به سازوکار اصلاحی نیست.

عملگر جهش. بعد از ایجاد تعداد جمعیت، درصدی از کروموزوم‌ها در معرض جهش قرار می‌گیرند. عملگر جهش به منظور حفظ تنوع کروموزوم‌ها در جمعیت اجرا می‌شود. در جریان اجرای جهش از دو راهبرد پیروی می‌شود.

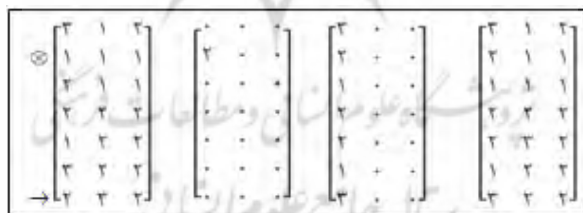
راهبرد اول، جهش ماشین‌محور (MBM) است. در این راهبرد چند عملیات به صورت تصادفی انتخاب می‌شود و به ماشین دیگری تخصیص می‌یابد. براساس آزمایش‌هایی که

الحنایی (۲۰۱۱) انجام داد، تعداد ژن‌های مناسب در هر کروموزوم به منظور اعمال عملگر جهش، عددی با توزیع یکنواخت در بازه (۱, ۳) است. بعد از آنکه تعداد ژن مد نظر انتخاب شد، به همان تعداد عملیات به صورت تصادفی انتخاب می‌شود و به ماشین‌های مناسب دیگر تخصیص می‌یابد. برای مثال، فرض کنید ۲ ژن انتخاب شده است (دو ژن انتخاب شده با علامت پیکان در شکل ۳ نمایش داده شده است). عملیات مربوط به دو ژن به ماشین‌های دیگر اختصاص می‌یابد. در این مثال عملیات ۲ از کار ۳ که روی ماشین ۱ انجام می‌شده است به ماشین ۲ منتقل می‌شود، ولی عملیات ۳ از کار ۲ فقط می‌تواند روی ماشین ۲ انجام شود، لذا بدون تغییر باقی می‌ماند.



شکل ۳. عملگر جهش MBM

راهبرد دوم، جهش موقعیت‌محور (PBM) است. در این جهش عملیاتی از فرزند به صورت تصادفی انتخاب می‌شود و سپس در موقعیت دیگری در فرزند جدید درج می‌شود. سپس عملیات باقی‌مانده با رعایت محدودیت‌های توالی به فرزند جدید کپی می‌شود. شکل زیر رویه این جهش را نشان می‌دهد (ژن انتخاب شده با علامت پیکان و موقعیت جدید با علامت \otimes نمایش داده است).



شکل ۴. عملگر جهش PBM

مرحله شروع مجدد. مرحله شروع مجدد در واقع برای حفظ تنوع در جمعیت طراحی شده است؛ بدین ترتیب که پس از اجرای الگوریتم ژنتیک، زمانی که بهترین جواب با میانگین جواب‌ها در ۳۰ نسل متوالی بر هم منطبق شد، مرحله شروع مجدد اجرا می‌شود. در این مرحله نیمی از جمعیت فعلی برای نسل بعد انتخاب می‌شود و جمعیت باقی‌مانده نیز به صورت تصادفی ایجاد خواهد شد.

تطبیق الگوریتم تکامل تفاضل برای مسئله مورد مطالعه. رابطه ۴ بیان‌کننده عملگر جهش برای الگوریتم تکامل تفاضلی است (محمودی و لطفی، ۱۳۹۱).

$$X_{New} = X_{Selected} + F \times (X_{Best} - X_{Selected}) \quad \text{رابطه ۴}$$

F عددی تصادفی است که گام‌های حرکتی به سمت بهترین جواب را تعیین می‌کند. بنابراین در الگوریتم تکامل تفاضلی گسسته، نیاز به روشی است که این حرکت و رسیدن به بهترین جواب در هر تکرار را مدل کند. در ادامه مفاهیم تفریق، ضرب و جمع موجود در رابطه ۴ برای فضای گسسته باز تعریف می‌شود.

تفریق: برای به‌دست آوردن اختلاف دو بردار در فضای گسسته، فهرست حرکات ضروری برای تغییر از راه حل S_j به S_i با عنوان $M_{j \rightarrow i}$ تعریف شده است که از طریق تغییر عناصر به‌دست می‌آید. اختلاف بین دو راه حل در قالب رابطه ۵ نشان داده شده است.

$$M_{j \rightarrow i} = S_i \ominus S_j \quad \text{رابطه ۵}$$

رابطه ۵ به این معناست که اگر حرکات در فهرست $M_{j \rightarrow i}$ روی راه حل S_j اعمال شود، راه حل S_i به‌دست خواهد آمد.

ضرب: با در نظر گرفتن $F \in [0, 1]$ که یک عدد اعشاری است و $M' = F \times |M|$ ، عملگر ضرب به این معناست که تعدادی از حرکات موجود در فهرست $M_{j \rightarrow i}$ انتخاب می‌شود. در واقع برای تکمیل کردن عملگر، از پارامتر F استفاده می‌شود که گامی تصادفی به سمت بهترین جواب تعریف شده است.

جمع: اگر x موقعیت و M' فهرست حرکات باشد، $x + M'$ به معنای اعمال فهرست حرکات M' روی بردار x است که موقعیت جدیدی در فضا ایجاد می‌کند. برای تفهیم این ایده شکل ۵ را در نظر بگیرید:

$$X_{Best} = \begin{bmatrix} 3 & 1 & 2 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \\ 2 & 2 & 2 \\ 1 & 2 & 2 \\ 3 & 2 & 1 \\ 2 & 3 & 2 \end{bmatrix} \quad X_{Selected} = \begin{bmatrix} 3 & 1 & 2 \\ 3 & 2 & 2 \\ 2 & 1 & 1 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \\ 1 & 2 & 2 \\ 2 & 3 & 2 \end{bmatrix}$$

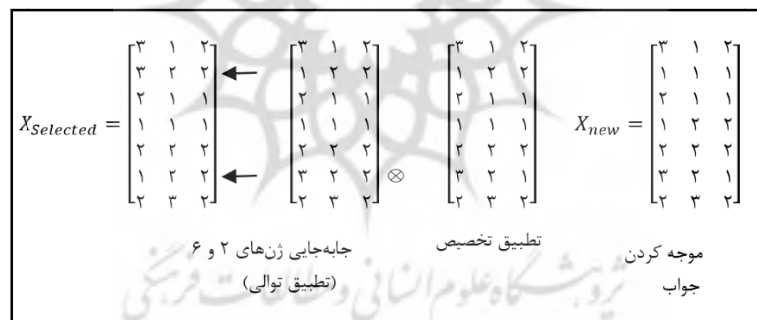
شکل ۵. بهترین کروموزوم و کروموزوم انتخاب‌شده برای جهش

در کروموزوم انتخاب‌شده برای جهش، موقعیت‌هایی که با پیکان نمایش داده شده‌اند، با بهترین جواب متفاوت‌اند. $X_{Best} - X_{Selected}$ باید فهرست حرکاتی تعریف شود که وقتی روی

$X_{Selected}$ اعمال شود، X_{Best} تولید شود. فهرست حرکات برای این مثال به شکل رابطه ۶ است.

$$M = X_{Best} - X_{Selected} = \{(2,6); (4,2); (5,4); (6,5)\} \quad \text{رابطه ۶}$$

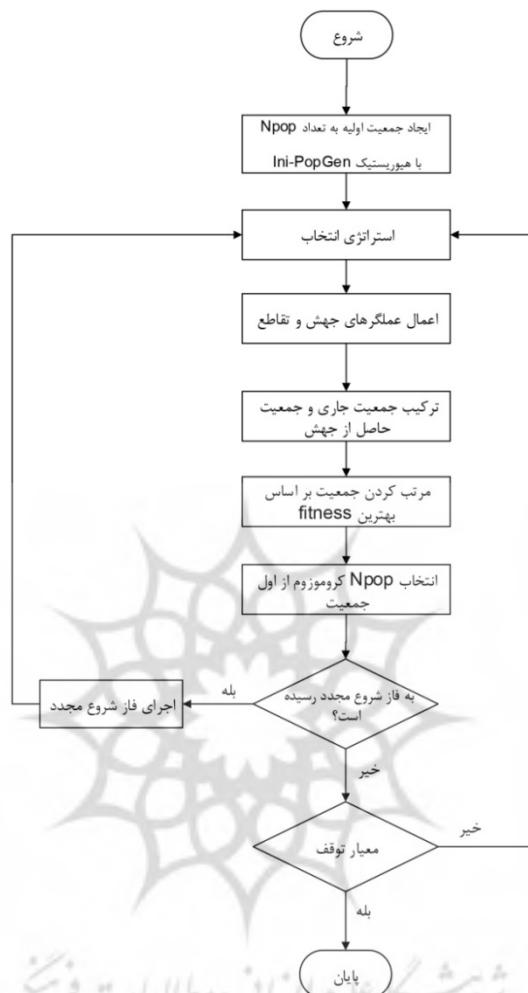
این بدین معناست که موقعیت ژن دوم در کروموزوم انتخاب شده باید با موقعیت ژن ششم جابه‌جا شود تا با موقعیت ششم بهترین کروموزوم یکسان شود. این جابه‌جایی ژن فقط توالی عملیات را روی ماشین تغییر می‌دهد؛ اگر در کروموزوم انتخاب شده عملیات روی ماشین متفاوت از بهترین کروموزوم پردازش شود، باید نوع ماشین را نیز با آن هماهنگ کرد. اعمال این حرکات به $X_{Selected}$ در واقع X_{Best} را تولید می‌کند. در این مثال و در بهترین کروموزوم، عملیات ۲ از کار ۳ روی ماشین ۱ پردازش شده است، اما در کروموزوم انتخاب شده عملیات ۲ از کار ۳ روی ماشین ۲ پردازش می‌شود. بنابراین پس از جابه‌جایی توالی در کروموزوم انتخاب شده باید تخصیص ماشین نیز با بهترین کروموزوم، هماهنگ شود. در نهایت چنانچه جواب خارج از فضای موجه باشد، باید با سازوکار اصلاحی به فضای موجه برگردد. این فرایند در شکل ۶ نشان داده شده است.



شکل ۶. نحوه عملکرد عملگر جهش الگوریتم تکامل تفاضلی

با توجه به اینکه $F \in [0,1]$ ، اگر $F = 0/3$ ، $M' = [F \times |M|]$ برابر با حرکت اول فهرست M به شکل $M' = \{(2,6)\}$ است.

نمودار الگوریتم‌های تکاملی پیشنهادی. ساختار کلی الگوریتم‌های ژنتیک و تکامل تفاضلی پیشنهادی این پژوهش در شکل ۷ نشان داده شده است.



شکل ۷. نمودار الگوریتم‌های تکاملی پیشنهادی

شبیه‌سازی شکست ماشین. برای شبیه‌سازی فرایند شکست ماشین از رویکرد پیشنهادی غلامی و زندیه (۱۳۸۶) استفاده شده است. تشخیص آنکه برای ماشین m در طول پردازش کار t ، خرابی رخ داده است یا نه، ضروری است. بعد از رخداد شکست، زمان تعمیر (مدت زمان تعمیر) محاسبه می‌شود. طول عمر هر ماشین (مدت زمان استفاده شده از هر ماشین، از زمان رخداد آخرین خرابی برای آن ماشین تا به حال) باید مشخص شود. از آنجاکه خرابی‌ها فقط طی زمان پردازش کار روی ماشین رخ می‌دهد، فقط زمان پردازش کارهایی که روی یک ماشین در فاصله بین هر دو خرابی پردازش شده‌اند، به‌مثابه طول عمر آن ماشین در نظر گرفته خواهد شد.

همچنین زمان‌های بیکاری و نیز زمان‌های آماده‌سازی در طول عمر یک ماشین در نظر گرفته نخواهد شد. متعاقب آن، زمان تعمیر به زمان تکمیل کار مد نظر z که در زمان خرابی ماشین در حال پردازش بوده است، اضافه می‌شود. به سبب طبیعت احتمالی مسئله باید محاسبه شبیه‌ساز را برای هر توالی ورودی چندین بار (Nsim) تکرار کرد. در حالیکه تمام مشخصات مسئله ثابت باقی می‌ماند؛ شاخص پایداری برای هر تکرار شبیه‌ساز محاسبه خواهد شد. پس از پایان تکرارهای شبیه‌ساز، میانگین شاخص پایداری محاسبه و مقدار برازندگی هر توالی ورودی در الگوریتم ژنتیک در نظر گرفته خواهد شد.

درباره شکست ماشین، فاصله زمانی بین وقوع هر دو شکست، از توزیع نمایی با میانگین MTBF تبعیت می‌کند و زمان تعمیر ماشین نیز از توزیع نمایی با میانگین MTTR^۱ پیروی خواهد کرد. علاوه بر این، تمام ماشین‌ها مقادیر یکسانی از MTBF و MTTR را اختیار دارند. هنگامی که یک شکست اتفاق می‌افتد، باید اقدام اصلاحی برای تغییر زمان‌بندی انجام شود. زمان‌بندی مجدد عملیاتی که مستقیم یا غیر مستقیم متأثر از شکست ماشین بوده‌اند، بهتر از زمان‌بندی مجدد کل برنامه است. بنابراین از روش جهش به راست^۳ برای زمان‌بندی مجدد عملیات متأثر از شکست استفاده شده است.

معیارهای سنجش زمان‌بندی مقاوم و پایدار. برای سنجش مقاومت زمان‌بندی از شاخص همسایگی محور N_1 پیشنهاد شده جنسن (۲۰۰۳) استفاده شده است. همسایگی $N_1(s)$ شامل s و همه زمان‌بندی‌هایی است که از s با جابه‌جایی دو عملیات متوالی روی یک ماشین می‌تواند ایجاد شود. به‌طور مثال برای مسئله نمونه MK-08 سایز همسایگی، $|N_1(s)|$ ، کمابیش برابر ۲۱۰ تا ۲۲۰ است. این شاخص به‌صورت رابطه ۷ تعریف می‌شود و در آن $MSP(s')$ زمان تکمیل برنامه برای زمان‌بندی s' است.

$$RBT(s) = \frac{1}{|N_1(s)|} \sum_{s' \in N_1(s)} MSP(s') \quad \text{رابطه ۷}$$

برای محاسبه پایداری زمان‌بندی، الحینایی و همکارانش (۲۰۱۱) سه شاخص را معرفی کرده‌اند. در این پژوهش از شاخص پایداری میانگین اختلاف زمان تکمیل برنامه، بین زمان‌بندی پیش‌بینی شده و زمان‌بندی‌ای که بعد از شکست ماشین به‌دست می‌آید (زمان‌بندی واقعی)، استفاده شده است. این شاخص به‌صورت رابطه ۸ است.

-
1. Mean time between failure
 2. Mean time to repair
 3. Right-Shifting

$$STB = \min \frac{\sum_{i=1}^n \sum_{j=1}^{q_i} |CT_{ijP} - CT_{ijR}|}{\sum_{i=1}^n O_i} \quad \text{رابطه ۸}$$

که در آن n تعداد کارها؛ q_i تعداد عملیات کار نام؛ CT_{ijP} زمان تکمیل عملیات نام از کار نام برای زمان بندی پیش بینی شده؛ CT_{ijR} زمان تکمیل عملیات نام از کار نام برای زمان بندی واقعی و O_i تعداد کل عملیات است.

یکپارچه سازی الگوریتم های فرا ابتکاری با شبیه سازی. شکل ۸ نمودار مدل پیشنهادی برای یکپارچه کردن شبیه سازی با الگوریتم فرا ابتکاری و همچنین ساختار کلی مدل استفاده شده در این پژوهش را نشان می دهد. این مدل شامل دو مرحله است. پیش از اجرای الگوریتم دوم مرحله ای، هریک از سه شاخص مقاومت، پایداری و زمان تکمیل برنامه، به صورت سه تابع هدف مجزا در نظر گرفته می شود و الگوریتم فرا ابتکاری تک هدفه برای هریک از انجام می گیرد. این فرایند به منظور دستیابی به جواب بهینه محلی به نام Loc Opt برای هر یک از سه شاخص انجام گرفته است. جواب بهینه محلی به دست آمده در مرحله دوم الگوریتم ژنتیک در تابع هدف ترکیبی استفاده خواهد شد. در ادامه مراحل مختلف الگوریتم پیشنهادی تشریح شده است.

در مرحله اول، الگوریتم یکپارچه با جمعیت اولیه Ini-PopGen و با تابع هدف حداقل سازی زمان تکمیل برنامه اجرا شده است. هنگامی که الگوریتم به معیار توقف می رسد، جواب های تکراری در نسل آخر حذف می شوند و جواب های منحصربه فرد استخراج خواهند شد. برای محاسبه شاخص پایداری، این جواب ها وارد شبیه سازی می شوند. در الگوریتم شبیه ساز، ماشین ها در معرض شکست قرار می گیرند و شبیه ساز $Nsim$ مرتبه تکرار می شود. در هر مرتبه شاخص پایداری برای هر زمان بندی، براساس رابطه ۸ محاسبه و نگهداری می شود. در نهایت میانگین $Nsim$ شاخص پایداری محاسبه می شود و در تابع ترکیب خطی مرحله دوم به کار می رود. شاخص مقاومت نیز برای این جواب ها بر اساس رابطه ۹ (جنسن، ۲۰۰۳) محاسبه می شود.

در مرحله دوم الگوریتم یکپارچه، این سه شاخص به صورت خطی ترکیب شده اند. تابع هدف ترکیبی به صورت رابطه ۹ است و به منظور تکامل مرحله دوم استفاده خواهد شد.

$$Z_{com} = \min (\alpha C_{MSPN} + \beta C_{RBTN} + \gamma C_{STBN}) \quad \text{رابطه ۹}$$

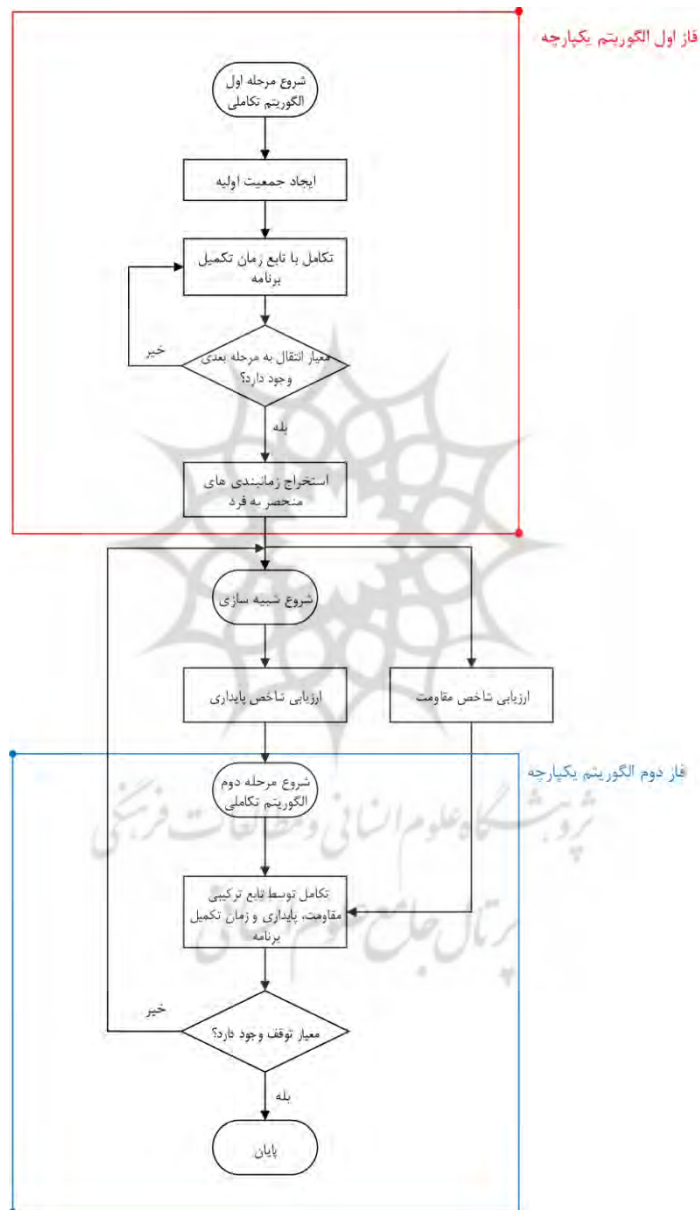
که در آن؛ $\alpha + \beta + \gamma = 1$ ، $\alpha, \beta, \gamma \in [0,1]$

$$C_{mN} = \frac{A \lg Sol_m - LB_m}{A \lg Sol_m} \quad \text{مقدار نرمال شده هر شاخص است.}$$

$$m \in \{MSP, RBT, STB\}$$

$LB_m = 0.8 \times LocOpt_m$ حد پایین شاخص m است.

$AlgSol_m$ جواب الگوریتم را برای هر شاخص نشان می‌دهد.



شکل ۸. مدل پیشنهادی برای یکپارچگی الگوریتم فرا ابتکاری با شبیه‌سازی

در این مدل وزنی که هریک از شاخص‌ها اختیار می‌کند، تعیین‌کننده اهمیت آن خواهد بود و مسیر تکامل الگوریتم را تغییر خواهد داد. این وزن را کاربر بر اساس نوع محیط تعیین می‌کند. حد پایینی که برای هر شاخص استفاده شده به میزان $0/8$ جواب بهینه محلی (LocOpt) در نظر گرفته شده است.

طراحی پارامترها و تولید داده‌های آزمون

برای بررسی اثری که ضریب هریک از شاخص‌های این مطالعه بر تکامل الگوریتم ژنتیک و تکامل تفاضلی دارند، چهار ترکیب خطی طبق جدول ۲ برای تابع هدف مرحله دوم آزمایش شدند.

جدول ۲. جدول ترکیب مختلف پارامترها

| نوع پارامتر | α | β | γ |
|-------------|----------|---------|----------|
| نوع ۱ | $0/15$ | $0/10$ | $0/75$ |
| نوع ۲ | $0/33$ | $0/33$ | $0/33$ |
| نوع ۳ | $0/30$ | $0/50$ | $0/20$ |
| نوع ۴ | $0/50$ | $0/30$ | $0/20$ |

تمام الگوریتم‌های این پژوهش به کمک نرم‌افزار MATLAB R2009a اجرا شده است. تعداد جمعیت الگوریتم ژنتیک و تکامل تفاضلی برای مسائل کوچک ۱۰۰ و برای مسائل متوسط و بزرگ ۲۰۰، درصد عملگرهای جهش و تقاطع به ترتیب $0/3$ و $0/8$ ، نرخ عملگر جهش در الگوریتم تفاضلی $0/4$ و تعداد نسل برای گذار به مرحله دوم ۳۰۰ نسل در نظر گرفته شده است. تعداد تکرار الگوریتم شبیه‌ساز در هر مرحله ۱۰ مرتبه و تعداد نسل در مرحله دوم الگوریتم ژنتیک و تکامل تفاضلی، ۲۰۰ نسل در نظر گرفته شده است. این مقادیر بر اساس مشاهدات و تجربه به دست آمده است. به منظور پوشش کاملی از انواع مسائل با اندازه‌های مختلف، مسئله‌های نمونه براساس جدول ۳ انتخاب شده است.

بر اساس آن انتخاب شده است. این مقیاس سطح خرابی کارگاه یا به بیان دیگر درصدی از زمان که ماشین‌ها به دلیل خرابی فعالیت نمی‌کنند را نشان می‌دهد. به پیروی از هولتاس (۱۹۹۹) مقدار $MTTR$ برابر \bar{p} و $A_g = 0/05$ در نظر گرفته شده است. \bar{p} بیان‌کننده میانگین زمان پردازش یک عملیات است. برای مثال در مسئله ۰۵ MK، $A_g = 0/05$ ، $MTTR = \bar{p} = 6/79$ ، $MTBF$ آنگاه برابر با $MTBF = (6/79 \div 0/05) - 6/79 = 129/01$ خواهد بود؛ به این معنا که به طور متوسط

یک ماشین ۱۳۹/۰۱ واحد زمانی در دسترس خواهد بود و سپس ماشین با میانگین زمان تعمیر ۶/۷۹ دچار خرابی می‌شود.

جدول ۳. مسائل نمونه انتخاب شده

| نوع مسئله | نام | اندازه مسئله | تعداد عملیات | منبع |
|-----------|------|--------------|--------------|--------------------------|
| T-FJSP | I۳ | ۱۰ × ۱۰ | ۳۰ | کاصم و همکاران، ۲۰۰۲ الف |
| | I۴ | ۱۰ × ۱۵ | ۵۶ | |
| P-FJSP | MK۰۱ | ۱۰ × ۶ | ۵۶ | |
| | MK۰۴ | ۱۵ × ۸ | ۹۰ | |
| | MK۰۵ | ۱۵ × ۴ | ۱۰۶ | برندیمارت (۱۹۹۳) |
| | MK۰۸ | ۲۰ × ۱۰ | ۲۲۵ | |
| | MK۰۹ | ۲۰ × ۱۰ | ۲۵۰ | |
| | MK۱۰ | ۲۰ × ۱۵ | ۲۴۰ | |

تحلیل اثر پارامترها در تابع ترکیب خطی

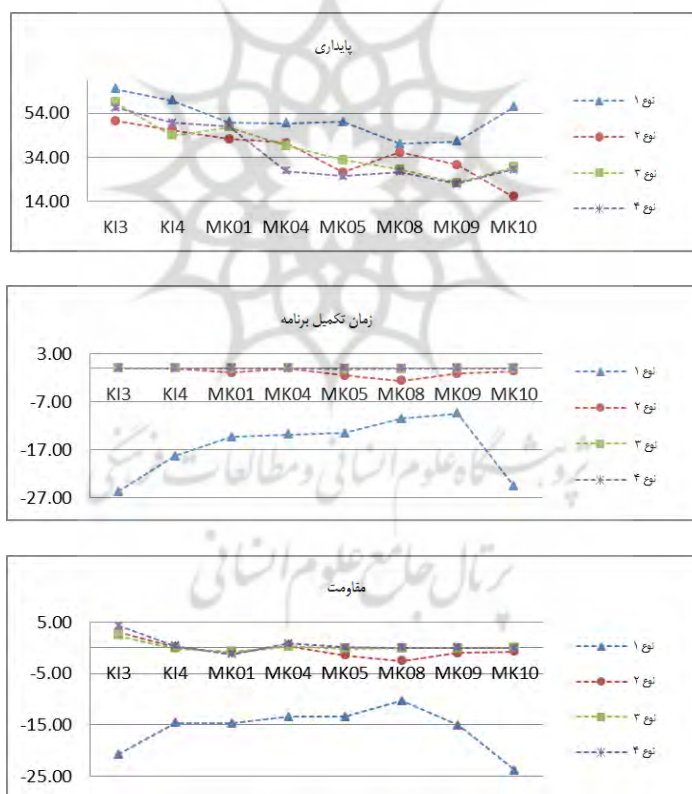
برای بررسی کارایی مدل پیشنهادی و دستیابی به زمان‌بندی مقاوم و پایدار، نتایجی که در مرحله دوم برای هر ترکیب به دست آمد (از نظر شاخص‌ها) با نتایج خروجی از شبیه‌سازی (قبل از ورود به مرحله دوم) مقایسه شده است. برای این کار هشت مسئله نمونه آزمایش شد؛ بدین ترتیب که هر یک به صورت جداگانه، پنج بار با در نظر گرفتن چهار ترکیب مختلف اجرا شد. بنابراین در مجموع $۱۶۰ = (۴ \times ۵ \times ۸)$ نمونه توسط هر الگوریتم آزمایش شده است. برای هر مسئله درصد بهبود هر شاخص بر اساس رابطه ۱۰ محاسبه شده است.

$$PIM_i = \frac{M_i^1 - M_i^2}{M_i^1} \times 100 \quad (\text{رابطه ۱۰})$$

که در آن: PIM_i درصد بهبود شاخص مربوطه را نشان می‌دهد؛ $i = MSP, RBT, STB$ است؛ M_i^1 شاخص بررسی شده در مرحله اول الگوریتم و M_i^2 شاخص بررسی شده در مرحله دوم الگوریتم است. علامت مثبت برای این شاخص نشانه بهبود آن و علامت منفی نشانه افت شاخص است. در ادامه به تحلیل اثر ضرایب استفاده شده در دو الگوریتم ژنتیک و تکامل تفاضلی پرداخته می‌شود.

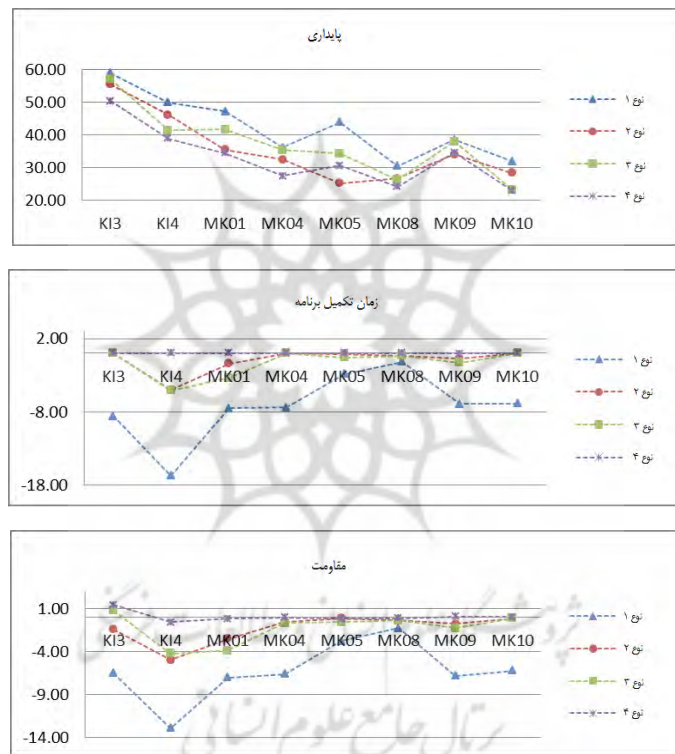
اثر پارامترها بر الگوریتم ژنتیک. شکل ۹ نمودار درصد بهبود سه شاخص را برای مسائلی که آزمایش شده‌اند، نشان می‌دهد. خطوطی که بالای محور صفر قرار گرفته‌اند، نشان‌دهنده بهبود

آن شاخص‌اند و خطوط زیر محور صفر، افت آن شاخص را نشان می‌دهند. خطوطی که روی محور صفر جای گرفته‌اند، مشخص می‌کنند شاخص مربوطه نسبت به مرحله اول در سطح اولیه خود باقی‌مانده است. از آنجاکه شاخص زمان تکمیل برنامه در فاز اول به‌اندازه کافی بهبود داده شده است، در مرحله دوم انتظار بهبود آن نمی‌رود و حفظ آن در سطح اولیه خود ($PIM_{MSP}=0$) می‌تواند نشان‌دهنده دستیابی به هدف باشد. همان‌طور که در نمودار پایداری شکل ۹ مشاهده می‌شود، میزان بهبود نوع چهارم به دلیل ضریب بالای آن در تابع ترکیب خطی (۰/۷۵) بیشتر از انواع دیگر است و به همین نسبت به‌شدت سبب افت شاخص زمان تکمیل برنامه شده است. ترکیب نوع دوم نیز مانع افت شاخص زمان تکمیل برنامه نبوده است. ترکیب نوع سوم در مسئله‌های MK۰۵ و MK۰۸ افت این شاخص را در پی داشته است. ترکیب نوع چهارم به بهبود نسبی شاخص پایداری در تمام مسائل منجر شده و سطح شاخص زمان تکمیل برنامه را در سطح اول خود نگه داشته است.



شکل ۹. نمودارهای درصد بهبود برای سه شاخص مقاومت، پایداری و زمان تکمیل برنامه با استفاده از الگوریتم ژنتیک

اثر پارامترها بر الگوریتم تکامل تفاضلی. در شکل ۱۰ نمودارهای مربوط به ترکیب نوع اول برای شاخص پایداری، بالاتر از ترکیبات دیگر مشاهده می‌شود و دلیل آن ضریب بالای این شاخص در تابع ترکیب خطی است. این موضوع سبب افت شاخص‌های زمان تکمیل برنامه و مقاومت شده است. ترکیب نوع چهارم شاخص زمان تکمیل برنامه را برای تمام مسائل، در سطح اولیه خود حفظ کرده است. اگرچه این ترکیب افت ناچیز شاخص مقاومت را در پی داشته است، درصد افت آن به مراتب کمتر از ترکیب‌های دیگر است (نمودار آن بالاتر از سایر نمودارهاست). نوسان‌های ترکیب‌های نوع دوم و سوم کمابیش بین ترکیب‌های نوع اول و چهارم قرار دارند.



شکل ۱۰. نمودارهای درصد بهبود برای سه شاخص مقاومت، پایداری و زمان تکمیل برنامه با استفاده از الگوریتم تکامل تفاضلی

نتایج این آزمایش به خوبی نشان می‌دهد با توجه به محیط‌های کاری متفاوت و درجه اهمیت هریک از این شاخص‌ها، برنامه‌ریز می‌تواند به هدف مد نظر خود دست یابد. برای مثال در محیط‌هایی که میزان شکست ماشین بسیار زیاد است، زمان‌بندی‌ای لازم است که پایداری بیشتری داشته باشد. بنابراین بهبود شاخص پایداری بیش از دو شاخص دیگر اهمیت دارد که در

این حالت برنامه‌ریز می‌تواند ترکیب نوع اول را انتخاب کند و در محیط‌هایی که به بهبود نسبی پایداری و حفظ کیفیت اولیهٔ زمان تکمیل برنامه احتیاج است، ترکیب نوع چهارم می‌تواند در کانون توجه قرار گیرد.

مقایسه کارایی الگوریتم‌های ژنتیک و تکامل تفاضلی. در این بخش از پژوهش به منظور مقایسه عملکرد دو الگوریتم، آزمون فرض آماری روی نتایج به دست آمده از اجرای الگوریتم ژنتیک و تکامل تفاضلی برای چهار زوج مختلف انجام گرفته است. بدین منظور مقادیر تابع ترکیب خطی (Z_{com}) به ازای هر چهار نوع ترکیب مختلف و برای هر پنج اجرا محاسبه شده است و از آزمون t استیودنت زوجی برای مقایسه میانگین این مقادیر بین دو الگوریتم ژنتیک و تکامل تفاضلی در سطح اطمینان ۹۸ درصد استفاده شده است. آزمون فرضیه آماری به صورت زیر است:

$$H_0: \mu_{type,GA} - \mu_{type,DE} = 0$$

$$H_1: \mu_{type,GA} - \mu_{type,DE} \neq 0$$

فرضیه H_0 بیان می‌کند میانگین نتایج محاسباتی برای تابع ترکیب خطی به ازای انواع مختلف - که در این پژوهش چهار نوع بود - برای دو روش حل الگوریتم‌های ژنتیک و تکامل تفاضلی، اختلاف معناداری با یکدیگر ندارند. فرضیه H_1 نیز بیان می‌کند میانگین نتایج محاسباتی برای تابع ترکیب خطی به ازای انواع مختلف برای دو روش حل الگوریتم‌های ژنتیک و تکامل تفاضلی، اختلاف معناداری با یکدیگر دارند.

نتایج آزمون فرضیه‌ها در جدول ۴ نشان داده شده است. همان‌طور که در این جدول مشاهده می‌شود، میزان P-Value برای ترکیب‌های نوع اول، دوم و چهارم از مقدار خطای آزمون ($\alpha/2$) کمتر است، بنابراین در سطح تشخیص ۲ درصد فرض H_0 رد می‌شود؛ به این معنا که بین میانگین تابع ترکیب خطی در دو روش حل الگوریتم ژنتیک و تکامل تفاضلی تفاوت معناداری وجود دارد، اما در ترکیب نوع سوم چون مقدار P-Value از مقدار خطای آزمون بیشتر است، فرضیه H_0 پذیرفته می‌شود، پس تفاوتی در استفاده از این دو روش وجود ندارد.

جدول ۴. نتایج آزمون فرضیه‌ها

| Pairs | Test Case | Std. Deviation | t | Df | P-Value |
|--------|----------------------|----------------|--------|----|---------|
| Pair ۱ | Type۱-GA , Type۱-DEA | ۰/۱۱۱۳۵ | -۳/۲۱۳ | ۳۹ | ۰/۰۰۳ |
| Pair ۲ | Type۲-GA , Type۲-DEA | ۰/۰۴۷۱۵ | -۳/۰۲۵ | ۳۹ | ۰/۰۰۴ |
| Pair ۳ | Type۳-GA , Type۳-DEA | ۰/۰۵۴۲۳ | -۱/۴۸۱ | ۳۹ | ۰/۱۴۷ |
| Pair ۴ | Type۴-GA , Type۴-DEA | ۰/۰۳۵۰۳ | -۳/۷۶۵ | ۳۹ | ۰/۰۰۱ |

نتیجه‌گیری و پیشنهادها

این پژوهش رویکردی را برای مواجهه با شکست ماشین در مسئله FJSP به‌منظور بهینه‌سازی سه هدف زمان تکمیل برنامه، مقاومت و پایداری، به‌طور هم‌زمان بررسی کرده است. از آنجا که زمان تکمیل برنامه هدف اصلی تمام مسائل بهینه‌سازی است، در این پژوهش نیز دو الگوریتم ژنتیک و تکامل تفاضلی پیشنهاد شده است که هر یک شامل دو مرحله می‌شوند. در مرحله اول زمان تکمیل برنامه بهبود داده شد و در مرحله دوم تابع هدف به ترکیب خطی از این سه هدف تغییر پیدا کرد و با توجه به ضرایبی که هر یک از شاخص‌ها اختیار می‌کنند، مسیر الگوریتم را در جهت بهبود آنها تغییر می‌دهند. نتایج آزمایش به خوبی نشان داد که برنامه‌ریز می‌تواند با توجه به محیط کاری و اهمیتی که هر یک از این شاخص‌ها دارند، از طریق تغییر ضرایب تابع ترکیب خطی، مسیر بهبود الگوریتم را در جهت شاخص مد نظر تغییر دهد. به‌دلیل ماهیت متضادی که این سه هدف با یکدیگر داشتند (بدین معنا که بهبود شاخص پایداری به‌شدت بر شاخص زمان تکمیل برنامه و مقاومت تأثیر می‌گذارد)، ترکیبی از این ضرایب که بتواند علاوه بر بهبود پایداری، کیفیت جواب دو شاخص دیگر را در سطح اولیه نگه دارد، از اهمیت خاصی برخوردار است. این پژوهش ترکیبی که بتواند این نیاز را برآورده کند، شناسایی و معرفی کرد و در نهایت به مقایسه نتایج محاسباتی دو الگوریتم نسبت به هم پرداخت. هیچ‌یک از پژوهش‌های انجام‌گرفته رویکرد شبیه‌سازی را برای سنجش مقاومت و پایداری زمان‌بندی انجام‌شده، به‌کار نبرده‌اند. در این پژوهش با پیشنهاد الگوریتم یکپارچه متشکل از الگوریتم ژنتیک و الگوریتم شبیه‌ساز، مشخص شد که امکان دستیابی به جواب مقاوم و پایدار هنگام وقوع تصادفی شکست ماشین، ضمن حفظ کیفیت اولیه جواب زمان تکمیل برنامه، وجود دارد. در مطالعات آتی سیاست شکست قطعات می‌تواند ادامه‌ناپذیر فرض شود یا سایر رخدادهای نامنتظره مانند لغو کار، ورود کار جدید، زمان‌های پردازش متغیر توسعه یا ترکیب سایر الگوریتم‌های فرا ابتکاری بررسی شود.

References

- Al-Hinai, N., ElMekkawy, T.Y. (2011a). An efficient hybridized genetic algorithm architecture for the flexible job-shop scheduling problem. *Flexible Services and Manufacturing Journal*, 23 (1): 64-85.
- Al-Hinai, N., ElMekkawy, T.Y. (2011b). Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm. *International Journal of Production Economics*, 132 (2): 279-291.
- Anglani, A., Grieco, A., Guerriero, E., Musmanno, R., (2005). Robust scheduling of parallel machines with sequence-dependent set-up costs. *European Journal of Operational Research*, 161 (3): 704-720.

- Bouyahia, Z., Bellalouna, M., Jaillet, P., Ghedira, K. (2010). A priori parallel machines scheduling. *Computers & Industrial Engineering*, 58 (3): 488-500.
- Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations Research*, 41(3): 157-183.
- Byeon, E., Wu, S.D. and Storer, R.H. (1998). Decomposition heuristics for robust job-shop scheduling. *IEEE Transactions on Robotics and Automation*, 4 (2): 303-313.
- Cheng, R., Gen, M. and Tsujimura, Y. (1996). A tutorial survey of job-shop scheduling problems using genetic algorithms – I. Representation. *Computers & Industrial Engineering*, 30 (4): 983-997.
- Gen, M. & Cheng, R. (2000). *Genetic algorithms & engineering optimization*, Wiley Series in Engineering Design and Automation, New York: John Wiley & Sons.
- Gholami, M., Zandieh, M. (2009). *Scheduling a hybrid flow shop with sequence dependent setup times and machines with random breakdowns*. Tehran: Shahid Beheshti University. (in Persian)
- Ho, N.B., Tay, J.C. & Lai, E. M-K. (2007). An effective architecture for learning and evolving flexible job-shop schedules. *European Journal of Operational Research*, 179 (2): 316-333.
- Holthaus, O. (1999). Scheduling in job shops with machine breakdowns: an experimental study. *Computers and Industrial Engineering*, 36 (1):137-162.
- Hurink, J., Jurisch, B., Thole, M. (1994). Tabu search for the job shop scheduling problem with multi-purpose machines. *Operations Research-Spektrum*, 15 (4): 205-215.
- Jensen, M.T. (2001b). Improving robustness and flexibility of tardiness and total flow time job shops using robustness measure. *Applied Soft Computing*, 1 (1): 35-52.
- Jensen, M.T. (2003). Generating robust and flexible job shop schedules using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 7 (3): 275-288.
- Kacem, I., Hammadi, S. & Borne, P. (2002a). Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man and Cybernetics*, 32 (1): 1-13.
- Kouvelis, P., Daniels, R. L. & Vairaktarakis, G. (2000). Robust scheduling of a two machine flow shop with uncertain processing times. *IIE Transactions*, 32 (5): 421- 432.
- Kutanoglu, E. & Wu, S.D. (1998). Improving schedule robustness via stochastic analysis and dynamic adaptation. *IMSE Technical Report 98T-001*. Available in: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.30.8534&rep=rep1&type=pdf>.
- Leon, V. J., Wu, S.D. & Storer, R. H. (1994). Robustness measures and robust scheduling for job shops. *IIE Transactions*, 26 (5): 32-43.

- Liu, L., Gu, H., Xi, Y. (2007). Robust and stable scheduling of a single machine with Random machine breakdowns. *International Journal of Advanced Manufacturing Technology*, 31 (7-8): 645-654.
- Mahmoudi, S., Lotfi, S. (2012). *Discrete Manufacturing Cuckoo Search algorithm Case study: Graph Coloring Problem*. Tabriz: University collage of Nabi Akram. (in Persian)
- Matsveichuk, N.M., Sotskov, Yu. N., Egorova, N.G. and Lai, T.C. (2009). Schedule execution for two-machine flow-shop with interval processing times. *Mathematical and Computers Modeling*, 49 (5-6): 991-1011.
- Mattfeld, D.C. (1996). *Evolutionary search and the job shop: investigations on genetic algorithms for production scheduling*, Germany: Heidelberg: Physica - Verlag.
- Mesghouni, K., Hammadi, S., and Borne, P. (1997). Evolution programs for job-shop scheduling. *IEEE International Conference on Systems, Man, and Cybernetics*, Orlando, Florida, 1: 720-725.
- Montemanni, R. (2007). A mixed integer programming formulation for the total flow time single machine robust scheduling problem with interval data. *Journal of Mathematical Modeling Algorithms*, 6 (2): 287-296.
- Policella, N., Cesta, A., Oddi, A. & Smith, S.F. (2005). Schedule robustness through broader solve and robustify search for partial order schedules. *Lecture Notes in Artificial Intelligence*, 3673: 160-172.
- Policella, N., Oddi, A., Smith, S.F. and Cesta, A. (2004). Generating robust partial order schedules. *M. Wallace (Ed.): CP 2004, Lecture Notes in Computer Science*, 3258: 496-511.
- Rangsaritratsamee, R., Ferrell, W.G. & Kurtz, M.B. (2004). Dynamic rescheduling that simultaneously considers efficiency and stability. *Computers & Industrial Engineering*, 46 (1): 1-15.
- Sevaux, M., Sorensen, K. (2004). A genetic algorithm for robust schedules in a one machine environment with ready times and due dates. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 4OR (2): 129-147.
- Shafaei, R. & Brunn, P. (2000). Workshop scheduling using practical (inaccurate) data Part 3: A framework to integrate job releasing, routing and scheduling functions to 139 create a robust predictive schedule. *International Journal of Production Research*, 38 (1): 85 – 99.
- Wei, Q. & Qiaoyun, L. (2009). Solving the flexible job shop scheduling problem based on the adaptive genetic algorithm. *2009 International Forum on Computer Science-Technology and Applications*, 1: 97-100.