

برنامه‌ریزی و زمان‌بندی پیشرفته با در نظر گرفتن اثر یادگیری در سیستم‌های ساخت کارگاهی انعطاف‌پذیر

محمد باقر فخرزاد^{*}، اسماعیل علی‌نژاد[†]

^{*} استادیار گروه مهندسی صنایع، دانشکده فنی و مهندسی، دانشگاه یزد، یزد
[†] کارشناسی ارشد مهندسی صنایع، گروه مهندسی صنایع، دانشکده فنی و مهندسی، دانشگاه یزد، یزد

خلاصه

برنامه‌ریزی و زمان‌بندی پیشرفته، نوعی فرآیند مدیریت تولید است که در آن منابع و ظرفیت‌های تولیدی به طور بهینه به تقاضاهای مشتریان تخصیص داده می‌شود. این رویکرد به طور خاص در محیط‌هایی که روش‌های برنامه‌ریزی ساده نمی‌توانند جوابگوی شرایط پیچیده باشند کاربرد دارد. یک فرض رایج در مسأله برنامه‌ریزی و زمان‌بندی پیشرفته این است که زمان پردازش یک محصول ثابت و مستقل از جایگاهش در توالی تولید است؛ اما در بسیاری از موارد عملی، عملکرد اپراتور به طور پیوسته با گذشت زمان بهبود و زمان پردازش کارها کاهش می‌یابد که این پدیده به عنوان اثر یادگیری شناخته می‌شود. در این مقاله، یک چارچوب بهینه‌سازی توسعه‌یافته برای مسأله برنامه‌ریزی و زمان‌بندی پیشرفته در محیط تولید انعطاف‌پذیر با در نظر گرفتن اثر یادگیری ارائه شده است تا توانایی بیشتری در بیان شرایط واقعی محیط‌های تولیدی فراهم و عوامل انسانی را در امر زمان‌بندی دخیل کند. همچنین به دلیل پیچیدگی محاسباتی بالای مدل پیشنهادی، یک الگوریتم حل ژنتیک چند مرحله‌ای نیز پیشنهاد شده است. نتایج عددی نشان می‌دهد که الگوریتم حل پیشنهادی، توانایی رسیدن به جواب‌های بهینه/ نزدیک بهینه را در زمان‌های محاسباتی بسیار کمتر از روش‌های دقیق دارد.

اطلاعات مقاله

تاریخچه مقاله:

دریافت ۱۳۹۲/۲/۱۶

پذیرش ۱۳۹۲/۵/۲۶

کلمات کلیدی:

برنامه‌ریزی و زمان‌بندی

پیشرفته،

اثر یادگیری موقعیت‌گرا،

الگوریتم ژنتیک چند مرحله‌ای

۱- مقدمه

می‌کند. با استفاده از سیستم‌های برنامه‌ریزی یکپارچه، آن‌ها می‌توانند نسبت به تغییرات به سرعت پاسخ دهند و محصولات با کیفیت بالاتر و هزینه‌های تولید کمتر ارائه کنند [۱-۳]. با شکل‌گیری مفهوم یکپارچه‌سازی، سیستم‌های ساخت یکپارچه شکل گرفتند تا به شرکت‌ها در داشتن سطح خدمت‌دهی عالی برای غلبه بر رقبا در بازار پرتلاطم امروزی کمک کنند. برنامه‌ریزی و زمان‌بندی پیشرفته (APS)^۱ شامل یک سری از تکنیک‌های زمان‌بندی در حالت ظرفیت محدود (از سطح کارگاه تا سطوح بالای سیستم‌های یکپارچه ساخت) و یک گام تکاملی جدید در راستای یکپارچه‌سازی

امروزه با توجه به رقابتی شدن شرایط بازارها، سازمان‌ها برای حفظ و بقا باید به دنبال افزایش کارایی و بهینه‌سازی عملیات تولیدی خود باشند. به همین دلیل باید عوامل متفاوتی را در راستای بهینه‌سازی برنامه زمان‌بندی خود و برای پاسخ‌گویی به تقاضاهای مشتریان مد نظر قرار دهند. این شرایط شرکت‌ها را به حرکت از فرآیندهای برنامه‌ریزی مجزا به سمت برنامه‌ریزی یکپارچه ترغیب

۱. نویسنده مسئول.

تلفن: ۰۳۵۱-۸۱۲۲۴۰۵، پست الکترونیکی: Mfakhrzad@yazd.ac.ir

2. Advanced Planning and Scheduling (APS)

دهد. لی و همکاران [۷] مدلی جامع برای مسأله APS در حالت مجاز بودن برون‌سپاری پیشنهاد دادند و برای اولین بار مفهوم تنوع مسیرهای پردازش را در مدل‌های APS لحاظ کردند. آن‌ها برای حل مدل پیشنهادی یک روش ابتکاری کارا بر مبنای الگوریتم ژنتیک ارائه کردند. چن و جی [۸] یک مدل MIP کارا با محدودیت‌های ظرفیت، ترتیب عملیات، زمان انجام سفارش و موعدهای تحویل برای یکپارچه‌سازی مسأله برنامه‌ریزی تولید و زمان‌بندی عملیات در یک محیط چند محصولی ارائه کردند. تابع هدف آن‌ها، کمینه‌سازی هزینه بر مبنای دو معیار زمان بیکاری سیستم و هزینه‌های زودکرد-دیرکرد سفارش‌ها بود. یکی از نکات مهم در مقالاتی که از الگوریتم ژنتیک استفاده کرده‌اند نحوه مناسب رمزگذاری و رمزگشایی جواب‌هاست. یانگ و تانگ [۹] یک الگوریتم ژنتیک تطبیقی^۱ برای حل مسأله APS در حالت چند هدفه ارائه و برای نشان دادن کارایی و سرعت الگوریتم پیشنهادی خود از مثال‌های عددی موجود در مقاله ژانگ و گن [۱۰] استفاده کردند.

به طور معمول، ماشین‌ها توانایی تولید چند نوع قطعه را دارند اما این نکته در مدل‌سازی مسائل APS لحاظ نشده است. به همین دلیل، ارسلان اورنگ و همکاران [۱۱] مدل چن و جی [۸] را با استفاده از مفهوم گراف انفصالی^۲ توسعه دادند و برای قطعات قابلیت پردازش روی ماشین‌های متفاوت در نظر گرفتند. از جمله معایب عمده این پژوهش این است که هیچ گونه روش حل و یا مثال عددی برای درک بهتر مدل‌سازی ارائه نشده است. چن و همکاران [۱۲] یک مطالعه موردی برای مسأله APS در کارخانجات ساخت وسائل روشنایی در چین با استفاده از مدل چن و جی [۸] انجام و برای حل آن یک روش هوشمند ابتکاری بر مبنای مفهوم الگوریتم ژنتیک و کلیدهای تصادفی^۳ ارائه دادند. در تمامی مقالات موجود در ادبیات موضوع، فرض شده که زمان پردازش کارها ثابت و مستقل از توالی بوده و با گذشت زمان تغییر نمی‌کند. در حالی که در چند سال اخیر مطالعات پژوهشگران در مورد پدیده‌هایی مانند اثر یادگیری، اثر خستگی و زمان‌های پردازش متغیر و قابل کنترل، صحت این فرض را مورد تردید قرار داده است. به همین دلیل در این مقاله، مدل موجود در [۱۱] به گونه‌ای توسعه داده شده است که بتوان فرض متغیر بودن زمان‌های پردازش (به عنوان مثال اثر یادگیری) را در مدل‌سازی لحاظ و آن را به شرایط دنیای واقعی محیط‌های تولیدی نزدیکتر کرد. در قسمت بعدی تاریخچه مختصری از مبحث اثر یادگیری بیان شده است تا لزوم اعمال این اثر در مدل‌های زمان‌بندی مشخص شود.

مدل‌های برنامه‌ریزی است [۴]. APS بیشتر در سازمان‌هایی قابل به‌کارگیری است که یک یا چند مورد از حالات زیر اتفاق بیفتند: (۱) تولید برای سفارش انجام شود نه برای انبارش، (۲) محصولات نیازمند تعداد زیادی قطعه یا عملیات باشند، (۳) تولید با محدودیت سرمایه و منابع مواجه باشد، (۴) محصولات برای استفاده از منابع با هم در رقابت باشند (محیط چند محصولی)، (۵) شرایط ناپایدار باشد به گونه‌ای که امکان زمان‌بندی منابع وجود نداشته باشد و (۶) استفاده از یک رویکرد تولید انعطاف‌پذیر اجتناب ناپذیر باشد. تا کنون در تمامی مقالات مربوط به حیطه APS و تولید کارگاهی یک فرض رایج این بوده که زمان پردازش کارها ثابت و مستقل از توالی است. در حالی که در بسیاری از موارد عملی با تکرار کارهای مشابه (و یا متفاوت)، توانایی و مهارت اپراتور افزایش و در نتیجه آن، زمان پردازش کارها کاهش می‌یابد. این امر باعث بهبود مستمر عملکرد تسهیلات تولیدی مخصوصاً نیروی انسانی می‌شود و در ادبیات موضوع با نام اثر یادگیری شناخته می‌شود. بنابراین در این مقاله، برای اولین بار یک مدل برنامه‌ریزی و زمان‌بندی پیشرفته به گونه‌ای توسعه داده شده است که بتواند مفهوم اثر یادگیری را لحاظ کند. این مدل نه تنها برای اعمال اثر یادگیری در مدل‌های زمان‌بندی کارگاهی انعطاف‌پذیر، بلکه برای مدل‌های موزی و خطوط مونتاژ که در آن‌ها زمان پردازش عملیات (بنا به هر دلیلی) تغییر می‌کند نیز قابل استفاده است. با توجه به پیچیدگی محاسباتی بالای مسأله، برای حل مدل پیشنهادی نیز یک الگوریتم ژنتیک چند مرحله‌ای پیشنهاد شده است.

در ادامه مقاله، در بخش دوم به مروری اجمالی بر ادبیات APS و اثر یادگیری پرداخته شده است. سپس در بخش سوم یک مدل ریاضی کاربردی برای مسأله APS با در نظر گرفتن اثر یادگیری ارائه و برای حل آن در بخش چهارم یک الگوریتم ژنتیک چند مرحله‌ای توسعه داده شده است. بخش پنجم به نتایج محاسباتی اختصاص یافته و در نهایت در بخش ششم نتیجه‌گیری و پیشنهاد کارآیندگان آورده شده است.

۲- ادبیات موضوع

به طور کلی می‌توان پژوهش‌ها در حیطه مسائل APS را به دو دسته کلی کیفی و کمی تقسیم‌بندی کرد. در پژوهش‌های کیفی از دید مدیریتی به جنبه‌های تحلیلی، انتقادی و توصیفی پرداخته شده است که از موضوع این مقاله خارج است. علاقه‌مندان به مقالات کیفی می‌توانند به عنوان نمونه به مراجع [۵، ۶] مراجعه کنند. در پژوهش‌های کمی، معمولاً ابتکار در ارائه مدل‌های ریاضی و یا بهبود روش‌های حل موجود بوده است. در حیطه پژوهش‌های کمی، مسائل زمان‌بندی در حالت برون‌سپاری نظر محققان زیادی را به خود جلب کرده است. این‌گونه مسائل معمولاً هنگامی کاربرد دارد که تولیدکننده بخواهد سفارش‌ها را دقیقاً در یک موعد مقرر تحویل

1. Adaptive Genetic Algorithm

2. Disjunctive graph

3. Random keys

۲-۱- اثر یادگیری و تاریخچه آن

که اگر تراکنش‌های انسانی اثر قابل ملاحظه‌ای در طی پردازش کارها داشته باشد، بهتر است که اثر یادگیری به صورت وابسته به زمان در نظر گرفته شود [۱۹].

در طی چند سال اخیر، اثر یادگیری به دلیل اهمیت آن در مباحث نوین مدیریتی، صنایع و بخش‌های تجاری، توجه خاصی را در مسائل زمان‌بندی به خود جلب کرده است [۲۰] که رشد نمایی حجم مقالات ارایه شده در این حیطه گواه این ادعاست.

دژونگ با استفاده از نتایج مطالعات زمان‌سنجی، یک مدل یادگیری جدید و کاربردی برای تقریب میزان کاهش زمان پردازش کارها در تکرارهای آتی ارایه داد (رابطه (۱)) که در مراجع به این فرمول اشاره مستقیم شده است [۱۴، ۱۵].

$$p_{j,r} = p_j (M + (1-M)r^\alpha) \quad j \geq 1, r \leq n \quad (1)$$

که در آن، p_j زمان پردازش نرمال کار، $\alpha \leq 0$ شاخص یادگیری، M فاکتور تراکم‌پذیری r و موقعیت (جایگاه) کنونی کار (قطعه) در برنامه زمان‌بندی است. دژونگ برای کارهای اپراتور-محور $M = 0.25$ و برای کارهای ماشین-محور $M = 0.5$ را پیشنهاد کرد. او کولوفسکی و گاوچنویچ [۱۴] بر مبنای رابطه (۱)، مدلی برای زمان‌بندی ماشین‌های موازی با هدف کمینه‌سازی دامنه عملیات ارائه و ادعا کردند که اولین کسانی هستند که مدل دژونگ را به عنوان اثر یادگیری در حیطه مسائل زمان‌بندی وارد کرده‌اند. هرچند که به نظر می‌رسد قبل از آن‌ها، بهلولی و ذگردی [۱۵] این رابطه را در نظر گرفته بودند. در ادامه به برخی از مهم‌ترین و جدیدترین پژوهش‌ها در زمینه اثر یادگیری اشاره می‌شود:

جی وانگ و جیان وانگ [۲۱] هر دو رویکرد یادگیری موقعیت‌گرا و مجموع زمان پردازش را در نظر گرفته و یک مدل یادگیری ترکیبی ارایه کردند. ژانگ و همکاران [۲۲] بر مدل‌های مجموع لگاریتمی زمان‌های پردازش و پردازش موقعیت‌گرا تمرکز و رابطه‌ای برای مجموع لگاریتمی زمان‌های پردازش ارائه کردند. امینی و همکاران [۲۳] یک مدل ریاضی برای مسأله ماشین‌های موازی یکسان در حالت وجود اثر یادگیری در زمان‌های آماده‌سازی و اثر خستگی در تخلیه ارائه و جواب‌های مدل قطعی خود را با روش‌های ابتکاری مقایسه کردند. همتا و همکاران [۲۴] یک مدل بهینه‌سازی چند هدفه را برای مدل بالانس خط مونتاژ در حالت نامشخص بودن زمان پردازش کارها ارائه و با ترکیب دو تکنیک موزون و ماکسی‌مین، یک روش حل برای مدل خود ارائه کردند. در ادامه همین پژوهش، همتا و همکاران [۲۵] برای حل مسأله در ابعاد بزرگ، یک الگوریتم بهینه‌سازی انبوه ذرات با جستجوی متغیر همسایگی پیشنهاد دادند. بهنامیان و فاطمی قمی [۲۶] مسأله فلوشاپ ترکیبی چند هدفه را در حالت زمان پردازش وابسته به زمان مورد بررسی قرار دادند و یک روش نمایش جواب و الگوریتم فراابتکاری استوار جدید پیشنهاد دادند.

یک فرض رایج در مسائل زمان‌بندی کلاسیک این است که زمان تولید یک محصول (پردازش کار) ثابت و مستقل از جایگاهش در توالی تولید است. برای مثال، بیکر فرض می‌کند زمان‌های پردازش ثابت، مشخص و مستقل از توالی کارهاست [۱۳]. از آن جا که مسائل کلاسیک زمان‌بندی برای محیط‌های چندماشین محدود و ناکارا هستند، با تکامل این‌گونه مسائل از اوایل دهه ۱۹۹۰ مدل‌های نوینی در مسائل زمان‌بندی به وجود آمد؛ از جمله مسأله زمان‌بندی با زمان‌های پردازش قابل کنترل، زمان‌بندی کارها با زمان‌های پردازش وابسته به زمان و زمان‌بندی کارها روی ماشین‌های چندمنظوره [۱۴]. بسیاری از محققین با بررسی برنامه زمان‌بندی تجربی و تئوری به این نتیجه رسیده‌اند که زمان‌های پردازش کارها تراکم‌پذیرند [۱۵]. یکی از رویکردها برای کاهش زمان پردازش کارها و در نظر گرفتن تراکم‌پذیری، استفاده از مبحث یادگیری است [۱۶]. در بیشتر مواقع با تکرار یک کار، توانایی و مهارت کارگر افزایش و زمان پردازش کارها کاهش می‌یابد. این امر باعث بهبود مستمر عملکرد تسهیلات تولیدی (مخصوصاً نیروی انسانی) می‌شود و در ادبیات موضوع با نام اثر یادگیری شناخته می‌شود.

رایت [۱۷] اولین کسی بود که به تحقیق در مورد مفهوم اثر یادگیری در چارچوب یک کار علمی پرداخت. او با استفاده از آزمایش‌ها علمی در صنایع هواپیمایی، فرضیه مشهور ۸۰٪ را مطرح کرد که «با دو برابر شدن خروجی تولید، زمان پردازش هر واحد ۲۰٪ کاهش می‌یابد». با توجه به محسوس بودن نقش نیروی انسانی در محیط‌های زمان‌بندی، تعداد عملیاتی که در آن احتمال وقوع یادگیری وجود داشته باشد زیاد است؛ بنابراین در نظر گرفتن یادگیری در محیط‌های زمان‌بندی معقول به نظر می‌رسد. اولین بار بیسکاپ [۱۸] مفهوم اثر یادگیری را در مسایل زمان‌بندی بکار برد. این مفهوم پس از آن مورد توجه بسیاری از پژوهشگران قرار گرفت و امروزه پژوهش‌های متعددی در این زمینه به چاپ رسیده است. برای اعمال اثر یادگیری در مدل‌های زمان‌بندی دو رویکرد عمده و متفاوت وجود دارد که هر یک اعتبار خاص خود را دارند.

۱- رویکرد موقعیت‌گرا^۱ که در آن، یادگیری تحت تأثیر تعداد کارهایی است که تاکنون پردازش شده‌اند. معمولاً این رویکرد در مواردی استفاده می‌شود که پردازش واقعی کارها عمدتاً ماشین‌محور بوده و با مداخله کمتر انسان انجام می‌شود.

۲- رویکرد مجموع زمان پردازش^۲ که در آن، یادگیری تحت تأثیر مجموع زمان کارهایی است که تاکنون پردازش شده‌اند. به عنوان مثال‌های عملی از این رویکرد می‌توان فرآیندهایی را نام برد که انجام آن‌ها بسیار پیچیده و خطا ساز است. توصیه پژوهشگران بر آن است

1. Position-Based Learning
2. The Sum of Processing Time

۳- آرایه مدل ریاضی و بررسی پیچیدگی آن

معمولاً مدل‌های موجود در ادبیات موضوع برنامه‌ریزی و زمان‌بندی پیشرفته یکسان و یا دارای مفهوم مشابهی هستند. با این حال مقاله اورنک و همکاران [۱۱]، یکی از جدیدترین و کامل‌ترین مدل‌های آرایه‌شده در حیطه APS است که قابلیت مدل‌سازی محیط‌های کارگاهی انعطاف‌پذیر را داراست. اما یکی از معایب این پژوهش آن است که زمان‌های پردازش کارها ثابت و مستقل از توالی فرض شده است. به همین دلیل این مدل به گونه‌ای توسعه داده شده است که با حذف فرض ثابت بودن زمان پردازش کارها، بتوان اثر یادگیری را در مدل‌های APS لحاظ کرد. قابل ذکر است که با توجه به پیش‌نیازهای استفاده از مدل‌های یادگیری، در این پژوهش بخش قابل توجهی از ساختار مدل‌سازی و نحوه تعریف متغیرهای تصمیم نسبت به مقاله مینا تغییر کرده است.

یک محیط تولیدی چند محصولی را در نظر بگیرید که با استفاده از یک سری ماشین‌آلات اقدام به تولید سفارش‌های مشتریان می‌کند. هر محصول ساختار درختی داشته و از یک سری قطعه و زیر مونتاژ تشکیل شده و برای تولید هر قطعه، ماشین‌های متفاوتی در اختیار است. ماشین‌ها می‌توانند در زمانی غیر از صفر نیز در دسترس باشند اما همه سفارش‌ها در زمان صفر در دسترس هستند. هدف مدل، آرایه یک برنامه زمان‌بندی برای تولید قطعات تمامی سفارش‌هاست که گونه‌ای که دامنه عملیات کمینه شود. در این برنامه زمان‌بندی دقیقاً مشخص می‌شود که کدام قطعه از کدام محصول، باید چه موقع و توسط چه ماشینی تولید شود. در اینجا به توضیح ساختار مدل پرداخته می‌شود:

اندیس‌ها و مجموعه‌ها:

$i, j = 1, 2, \dots, n$: اندیس محصولات (سفارش‌های مشتری)

n : تعداد کل محصولات مورد برنامه‌ریزی (کل سفارشات مشتریان)

$A(P_i)$: مجموعه قطعات متفاوت مورد نیاز برای تولید محصول i

$N(A(P_i))$: تعداد قطعات متفاوت مورد نیاز برای تولید محصول i

b : کل قطعات مورد نیاز برای برآورده کردن سفارش‌های تمامی مشتریان که از رابطه (۲) بدست می‌آید.

$$b = \sum_{i=1}^n N(A(P_i)) \quad (2)$$

$q, p = 1, 2, \dots, b$: اندیس قطعات

N_{ip} : تعداد قطعه مورد نیاز p برای تولید یک واحد محصول i

$l, k = 1, 2, \dots, m$: اندیس ماشین‌آلات

m : تعداد کل ماشین‌ها

r : اندیس موقعیت (جایگاه) انجام کار که حداکثر برابر با کل قطعاتی است که باید تولید شوند یعنی $r = 1, 2, \dots, b$

پارامترها:

wh: ساعات کار روزانه

p_i : قطعه نهایی محصول i یعنی بالاترین قطعه در درخت محصول

Q_i : تعداد سفارش محصول i

t_{ipk} : مدت زمان نرمال تولید قطعه p از سفارش محصول i روی

ماشین k در حالت نرمال (بدون در نظر گرفتن موقعیت انجام کار)

M_{ipk} : عامل تراکم‌ناپذیری: (درصدی از زمان پردازش قطعه p از

سفارش i روی ماشین k که بدون دخالت انسان انجام می‌شود).

β : شاخص یادگیری در انجام کارهای یکسان

\bar{t}_{ipk} : میانگین زمان نرمال انجام کار p از سفارش i روی ماشین k که

از رابطه (۳) بدست می‌آید. در این رابطه با استفاده از زمان پردازش

نرمال، حجم سفارش محصول و نرخ یادگیری برای کارهای یکسان،

میانگین زمان نرمال انجام کار بدست می‌آید. به عبارت دیگر،

میانگین زمان نرمال (\bar{t}_{ipk}) همان زمان نرمال (t_{ipk}) است که در آن

اثر یادگیری اعمال شده است.

$$\bar{t}_{ipk} = t_{ipk} \times \frac{\sum_{a=1}^{N_{ip}Q_i} (M_{ipk} + (1 - M_{ipk}) \alpha^{\beta_{ipk}})}{N_{ip}Q_i} \quad \forall i, p, k \quad (3)$$

α : شاخص کلی یادگیری که برابر است با لگاریتم نرخ یادگیری در

مبنای دو

$t_{ipk[r]}$: مدت زمان واقعی لازم برای تولید قطعه p از محصول i در

موقعیت r روی ماشین k که از رابطه (۴) محاسبه می‌شود.

$$t_{ipk[r]} = \bar{t}_{ipk} (M_{ipk} + (1 - M_{ipk}) r^\alpha) \quad \forall i, p, k, r \quad (4)$$

RT_k : زمان دسترسی ماشین k

d_i : موعد تحویل سفارش محصول i

I : هزینه بیکاری سیستم در واحد زمان (ساعت)

M : عددی مثبت و به اندازه کافی بزرگ

R_i : مجموعه زوج مرتب‌هایی از قطعات (q, p) از سفارش محصول

i به گونه‌ای که قطعه q پیش‌نیاز مستقیم قطعه p باشد.

F_p : مجموعه ماشین‌هایی که قادرند قطعه p را تولید کنند.

متغیرهای تصمیم:

C_{\max} : زمان تکمیل مجموعه کارها (دامنه عملیات)

$Cr_{k[r]}$: زمان تکمیل جایگاه r ماشین k

$Z_{ipk[r]}$: متغیری باینری که مقدار یک می‌گیرد اگر قطعه p از

سفارش محصول i روی ماشین k در جایگاه r زمان‌بندی شود.

$fin_{ipk[r]}$: متغیری مثبت که برای خطی‌سازی مدل استفاده می‌شود.

توابع هدف متعددی در حیطه مسایل زمان‌بندی وجود دارد که از

جمله پرکاربردترین و مشهورترین آن‌ها، کمینه‌سازی دامنه عملیات

است. کمینه‌سازی عملیات معادل استفاده حداکثری از ماشین‌آلات و

کمینه‌سازی کار در جریان ساخت است. رابطه (۵) نشان‌دهنده این

تابع هدف است

$$\text{Min } Z = C_{\max} \quad (5)$$

خطی شدن مدل می‌شود، با استفاده از روابط جایگزین (۱۰) تا (۱۲) مدل خطی‌سازی شده است. قابل ذکر است که پس از خطی‌سازی، دیگر نیازی به استفاده از رابطه (۹) نیست. محدودیت (۱۳) تضمین می‌کند که زمان شروع تولید یک نوع قطعه (p), بزرگ‌تر یا مساوی زمان تکمیل قطعات پیش‌نیازش (q) باشد. رابطه (۱۴) تضمین می‌کند که هر قطعه فقط روی یک ماشین و در یک جایگاه پردازش شود. محدودیت (۱۵) بیان می‌کند که باید در هر جایگاه از هر ماشین حداکثر یک کار قرار بگیرد. رابطه (۱۶) برای رعایت پیوستگی توالی زمان‌بندی در نظر گرفته شده است و بدان معنی است که تا جایگاهی پر نشده جایگاه بعدی آن پر نشود. اگر قطعه p متعلق به محصول i نباشد از همان ابتدا باید مقادیر Z آن روی همه ماشین‌ها صفر باشد. این مطلب در محدودیت (۱۷) بیان شده است. رابطه (۱۸) تضمین می‌کند که کارها روی ماشین‌هایی زمان‌بندی شوند که قابلیت پردازش آن را دارند.

تحقیقات زیادی نشان داده است که مسأله برنامه‌ریزی و زمان‌بندی پیشرفته از جمله مسائل NP-Hard است [۸،۲]. از طرفی، حل مسائلی با در نظر گرفتن اثر یادگیری، نیازمند تلاش‌های محاسباتی به مراتب بیشتری نسبت به مدل‌های گذشته است [۱۶]؛ زیرا افزودن این اثر، باعث افزایش یک بُعد (اندیس) جدید به متغیرهای تصمیم مسأله می‌شود. می‌توان اثبات کرد که مدل پیشنهادی این مقاله نیز از نوع NP-Hard خواهد بود (با فرض $\alpha, \beta = 0$ می‌توان به مدل‌های موجود در ادبیات APS رسید که همگی NP-hard هستند. چون در مسائل NP-hard نمی‌توان از روش‌های دقیق برای بدست آوردن جواب‌های مناسب در زمان‌های محاسباتی منطقی استفاده کرد، به همین دلیل در بخش بعد یک الگوریتم ژنتیک چند مرحله‌ای برای حل مدل، پیشنهاد داده شده است.

۴- طراحی الگوریتم پیشنهادی (روش حل)

الگوریتم ژنتیک یکی از مشهورترین و پرکاربردترین روش‌ها برای حل مسائل گوناگون و به خصوص مدل‌های APS است. مفهوم الگوریتم ژنتیک برای اولین بار توسط جان هلند^۱ در دانشگاه میشیگان، آن‌اربر^۲ مطرح شد. پس از آن وی و دانشجویانش به گسترش این الگوریتم پرداختند. این الگوریتم جزو کلاس الگوریتم‌های جستجوی تصادفی^۳ قرار داشته و برای بهینه‌سازی مسائل پیچیده با مدل‌هایی با فضای جستجوی ناشناخته مناسب است [۲۵]. الگوریتم ژنتیک شامل یک سری گام‌ها و اجزای مهم است که اگر به خوبی تعریف نشده باشد در یافتن جواب‌های مناسب مسأله ناتوان خواهد بود. برخی از این اجزا عبارتند از: نحوه نمایش جواب شدنی، ایجاد جواب

محدودیت‌های مسأله به صورت زیر است که در ادامه به طور دقیق و تفکیکی توضیح داده خواهد شد.

$s.t.$

$$Cr_{k[1]} \geq \sum_{i=1}^n \sum_{p \in A(p_i)} t_{ipk[1]} \times Z_{ipk[1]} \times Q_i \times N_{ip} + RT_k \quad (6)$$

$$\forall i, p \in A(p_i)$$

$$Cr_{k[r]} \geq Cr_{k[r-1]} + \sum_{i=1}^n \sum_{p \in A(p_i)} t_{ipk[r]} \times Z_{ipk[r]} \times Q_i \times N_{ip} \quad (7)$$

$$\forall i, p \in A(p_i), r = 2, \dots, b-1$$

$$Cr_{k[r]} \leq C_{\max} \quad k = 1, \dots, m, r = b \quad (8)$$

$$fin_{ipk[r]} = Z_{ipk[r]} \times Cr_{kr} \quad \forall i, k, p, r \quad (9)$$

$$fin_{ipk[r]} \geq -M(1 - Z_{ipk[r]}) + Cr_{k[r]} \quad (10)$$

$$\forall i, p \in A(p_i), k \in F_p, r = 1, \dots, b-1$$

$$fin_{ipk[r]} \leq M(1 - Z_{ipk[r]}) + Cr_{k[r]} \quad (11)$$

$$\forall i, p \in A(p_i), k \in F_p, r = 1, \dots, b-1$$

$$fin_{ipk[r]} \leq M \times Z_{ipk[r]} \quad (12)$$

$$\forall i, p \in A(p_i), k \in F_p, r = 1, \dots, b-1$$

$$\sum_{k=1}^m \sum_{r=1}^b (fin_{ipk[r]} - t_{ipk[r]} \times Z_{ipk[r]} \times Q_i \times N_{ip}) \geq \quad (13)$$

$$\sum_{k=1}^m \sum_{r=1}^b fin_{ipk[r]} \quad \forall i, (q, p) \in R_i$$

$$\sum_{k=1}^m \sum_{r=1}^b Z_{ipk[r]} = 1 \quad \forall i, p \in A(p_i) \quad (14)$$

$$\sum_{i=1}^n \sum_{p \in A(p_i)} Z_{ipk[r]} \leq 1 \quad \forall k, r \quad (15)$$

$$Z_{ipk[r+1]} \leq \sum_{\substack{i=1 \\ q \in A(q_i) \\ q \neq p}}^n \sum_{r=1}^b Z_{iqk[r]} \quad \forall i, k, r, p \in A(p_i) \quad (16)$$

$$\sum_{k \in F_p} \sum_{r=1}^b Z_{ipk[r]} = 0 \quad \forall i, p, N_{ip} = 0 \quad (17)$$

$$\sum_{r=1}^b Z_{ipk[r]} = 0 \quad \forall i, p, k \notin F_p \quad (18)$$

$$C_{\max} \geq 0 \quad (19)$$

$$C_{k[r]} \geq 0 \quad \forall k, r \quad (20)$$

$$Z_{ipk[r]} \in \{0, 1\} \quad \forall i, p, k, r \quad (21)$$

محدودیت (۶) برای محاسبه زمان تکمیل جایگاه اول هر ماشین و محدودیت (۷) برای دیگر جایگاه‌ها استفاده می‌شود. رابطه (۸) حدود دامنه عملیات را مشخص می‌کند. معمولاً در مدل‌های مختلط عدد صحیح با افزایش تعداد متغیرها، بُعد مسأله به صورت نمایی افزایش می‌یابد که این امر موجب افزایش زمان حل نیز می‌شود. بنابراین در این پژوهش سعی شده در مدل‌سازی، کم‌ترین تعداد متغیر تعریف و تا حد امکان از ترکیب متغیرهای موجود استفاده شود. به همین دلیل از رابطه ترکیبی (۹) برای محاسبه زمان تکمیل قطعه (با توجه به زمان تکمیل جایگاه) استفاده شده است. چون رابطه (۹) باعث غیر

1. Jan Holland
2. University of Michigan, Ann Arbor
3. Stochastic Search Technique

- قطعه دارای پیش‌نیاز نباشد: در این حالت قطعه زمان‌بندی شده و با حذف آن از UIL، قطعات پس‌نیاز آن آزاد می‌شود.
- تعیین کارهای دوم به بعد نیز با استفاده از مکانیزم مشابه صورت می‌گیرد. پس از اتمام مرحله اول، با ثابت نگه داشتن توالی بدست آمده، مرحله دوم آغاز می‌شود.
- ۲- تخصیص ماشین‌آلات: در این مرحله، به طور تصادفی یک ماشین از بین ماشین‌های مجاز برای پردازش قطعه انتخاب می‌شود.

۴-۳- تابع برازندگی

فرض کنید مقدار تابع هدف جواب فعلی مسأله f است. چون مسأله از نوع کمینه‌سازی است از معکوس تابع f استفاده می‌شود. اما چون مقدار تابع هدف در ابعاد بزرگ معمولاً زیاد است و معکوس آن، تفاوت بین جواب‌ها را به خوبی نشان نمی‌دهد در الگوریتم پیشنهادی از روش رتبه‌بندی استفاده شده است.

مرحله بعد، ایجاد دومین نسل از جامعه بر اساس نسل فعلی است که با استفاده از عملگرهای ژنتیک صورت می‌گیرد. این عملگرها عبارتند از: انتخاب و تولید مجدد، تقاطع و جهش.

۴-۴- عملگر انتخاب و تولید مجدد

این عملگر وظیفه انتخاب مناسب‌ترین کروموزوم‌ها را از میان نسل فعلی برای تشکیل نسل آینده به عهده دارد. هدف اصلی عملگر تولید مجدد، مضاعف کردن جواب‌های خوب و حذف جواب‌های بد در یک جمعیت با ثابت نگه‌داشتن اندازه جمعیت است. فرآیند انتخاب باید به گونه‌ای باشد که مناسب‌ترین کروموزوم‌ها انتخاب شوند. با این کار حتی ضعیف‌ترین عناصر هم شانس انتخاب داشته و از گرفتار شدن در جواب‌های محلی و بهینگی زودرس^۲ جلوگیری می‌شود. در این پژوهش، برای تشکیل نسل جدید از میان کروموزوم‌های نسل جاری، از تکنیک چرخ رولت استفاده شده است. ضمناً از عملگر تولید مجدد نیز برای انتقال درصد ثابتی از بهترین کروموزوم‌های نسل فعلی به نسل آتی استفاده شده است. این عملگر، تعداد مشخصی از کروموزوم‌های خوب جمعیت فعلی را بدون هیچ تغییری در جمعیت آتی کپی می‌کند. ایجاد جواب‌های جدید با استفاده از عملگرهای تقاطع و جهش صورت می‌گیرد که در ادامه توضیح داده شده است:

۴-۵- عملگر تقاطع

از این عملگر برای تولید فرزند از بین کروموزوم‌های موجود در استخر جفت‌گیری استفاده می‌شود. برخی از عملگرهای تقاطع مشهور عبارتند از تقاطع یک نقطه‌ای، دو نقطه‌ای و ترتیب-محور^۳ با توجه به ساختار خاص APS و وجود روابط پیش‌نیازی و ماشین‌های متناوب، بیشتر این عملگرها فرزندان ناموجه ایجاد می‌کنند. این

اولیه، محاسبه تابع برازندگی، عملگرها و معیار توقف که در ادامه به تفکیک توضیح داده خواهد شد.

۴-۱- نحوه نمایش جواب شدنی

برنامه‌ریزی و زمان‌بندی پیشرفته شامل یکسری محدودیت‌های پیش‌نیازی و ماشین‌های متناوب برای تولید قطعات است. در این حالت برای ایجاد یک جواب شدنی، نحوه نمایش جواب‌ها باید به گونه‌ای باشد که بتوان تمامی جواب‌های شدنی را در نظر گرفت. در این مقاله، برای نمایش جواب‌ها از روش چند مرحله‌ای استفاده شده که بهترین و رایج‌ترین روش در ادبیات موضوع APS است.

به طور کلی فرآیند کدگذاری جواب‌ها شامل دو مرحله است: در مرحله اول توالی کارها مشخص شده و سپس با ثابت نگه داشتن این توالی، در مرحله دوم برای هر قطعه یک ماشین (از بین ماشین‌های مجاز) انتخاب می‌شود. به همین دلیل، کروموزوم دو بُعدی تعریف شده است که بُعد اول شامل بردار توالی و بُعد دوم شامل بردار ماشین‌آلات است. برای مثال، محصولی متشکل از ۴ قطعه را در نظر بگیرید که باید توسط سه ماشین تولید شود. اگر قطعه ۱ والد و سه قطعه ۲، ۳ و ۴ فرزند باشند، یکی از جواب‌های شدنی مسأله را می‌توان در قالب شکل (۱) کد کرد.

| | | | |
|---|---|---|---|
| ۱ | ۳ | ۴ | ۳ |
| ۲ | ۲ | ۱ | ۳ |

شکل (۱): نحوه نمایش جواب شدنی به صورت کروموزوم

در این شکل، توالی انجام کارها به صورت ۲، ۴، ۳ و ۱ است که برای تولید آن‌ها به ترتیب از ماشین‌های ۳، ۱، ۲ و ۲ استفاده شده است.

۴-۲- ایجاد جواب اولیه

پس از مشخص شدن نحوه کدگذاری جواب‌های مسأله، چندین جواب به صورت تصادفی برای ایجاد نسل اول تولید می‌شود به طوری که این مرحله خود شامل دو مرحله است:

۱- ایجاد یک توالی شدنی: برای ایجاد یک جواب اولیه شدنی برای مسأله، از مفهوم کلیدهای تصادفی استفاده شده است. در این روش، ابتدا همه قطعات در فهرست قطعات برنامه‌ریزی نشده (UIL)^۱ قرار داده شده و برای هر کدام از آن‌ها، یک عدد تصادفی منحصر به فرد در بازه [۰، ۱] تولید می‌شود. سپس قطعه‌ای با کوچک‌ترین عدد تصادفی به عنوان کاندید برای قرارگیری در جایگاه اول برنامه زمان‌بندی انتخاب می‌شود. دو حالت اتفاق می‌افتد:

- قطعه دارای پیش‌نیاز باشد: در این حالت قطعه به UIL برگردانده و قطعه دیگری با مکانیزم مشابه انتخاب می‌شود.

2. Premature
3. Order-Based Crossover

1. Unscheduled Item List (UIL)

در ابعاد کوچک و برای نشان دادن سرعت و کیفیت آن نیز از مثال‌های عددی در ابعاد متوسط و بزرگ استفاده شده است. از آنجا که مسأله مطرح‌شده در این مقاله از نظر ساختاری کاملاً جدید و متفاوت با مدل‌های موجود در ادبیات موضوع است، نمی‌توان جواب‌های آن را با جواب‌های موجود در ادبیات موضوع مقایسه کرد. به همین دلیل جواب‌های مسئله با جواب‌های روش حل دقیق مقایسه شده است. برای محاسبه جواب دقیق مسأله، از نرم‌افزار GAMS22.2 و برای پیاده‌سازی الگوریتم ژنتیک پیشنهادی از نرم‌افزار MATLAB R2010 استفاده شده است. همچنین تمامی محاسبات مربوط به این پژوهش در یک کامپیوتر شخصی با مشخصات CPU 2.66 GHZ و 4GB RAM انجام شده است.

۵-۱- اعتبارسنجی مدل پیشنهادی

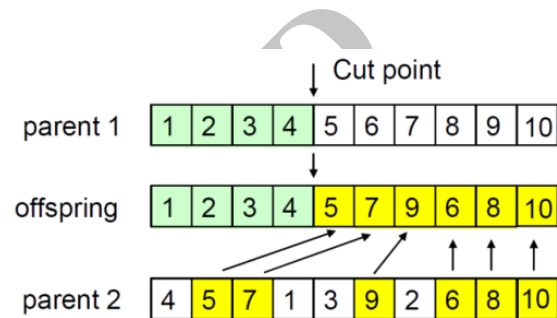
در این بخش با استفاده از یک مثال عددی در ابعاد کوچک و مقایسه جواب‌های حاصل از روش حل دقیق آن با جواب حاصل از روش شمارش کامل، اعتبار مدل در بدست آوردن جواب‌های منطقی نشان داده می‌شود. یک محصول سه قطعه‌ای را در نظر بگیرید که در آن قطعات ۲ و ۳ پیش‌نیاز قطعه ۱ هستند. زمان پردازش نرمال و هزینه تولید هر قطعه روی ماشین‌های مجاز در جدول (۱) آورده شده است.

جدول (۱): زمان پردازش نرمال و هزینه متغیر تولید قطعات

| ماشین‌ها | | قطعات |
|-----------|-----------|-------|
| ماشین دوم | ماشین اول | |
| (۵,۵۰) | - | ۱ |
| (۱۰,۱۰۰) | (۱۵,۸۰) | ۲ |
| (۵,۱۰۰) | (۱۰,۷۰) | ۳ |

هزینه بیکاری سیستم ۵۰ واحد پولی در ساعت و تعداد ساعات کاری روزانه ۸ ساعت است. زمان دسترسی ماشین‌ها به ترتیب ۰ و ۳ است. شاخص‌های یادگیری ۰/۳۲۲- و ضریب تراکم‌ناپذیری ۰/۵ است. موعد تحویل سفارش ۲ روز از لحظه دریافت سفارش (مبدأ زمان) است. هدف، یافتن بهترین برنامه زمانی تولید است به نحوی که زمان تکمیل مجموعه قطعات کمینه شود. با استفاده از روش شمارش کامل جواب بهینه مسأله برابر است با: $c_{\max} = 17.5$ که گانت چارت آن در شکل (۳) آورده شده است. مثال عددی مذکور با استفاده از مدل MIP پیشنهادی، در نرم‌افزار GAMS22.2 کد نویسی و اجرا شد که خروجی آن (مقدار تابع هدف و متغیرهای اصلی مسأله) در شکل (۴) نشان داده شده است. با توجه به شکل (۳) و (۴) مشاهده می‌شود که جواب مسأله در هر دو حالت یکسان بدست آمده است که این امر نشان دهنده توانایی مدل پیشنهادی در مدل کردن مسأله و یافتن جواب واقعی مسأله است.

اتفاق معمولاً در بُعد اول کروموزوم یعنی توالی کارها رخ می‌دهد. عملگر تقاطع ترتیب- محور یکی از عملگرهای تقاطع مشهور است که در عین داشتن کارایی، موجه بودن کروموزوم‌های حاصله را نیز تضمین می‌کند. به همین دلیل از این عملگر به طور گسترده در ادبیات APS استفاده شده است [۳،۱۰]. در عملگر تقاطع ترتیب- محور (با احتمال انتخاب P_c) برای تولید فرزند اول، تعدادی از ژن‌های والد اول ثابت می‌ماند و ژن‌های باقی‌مانده به ترتیب ظاهر شدن آن‌ها در والد دیگر مرتب می‌شوند. فرزند دوم نیز به روشی مشابه و با استفاده از والد دوم تولید می‌شود. نمونه‌ای از این نوع عملگر (حالت تک نقطه‌ای) در شکل (۲) آمده است.



شکل (۲): نمونه‌ای از عملگر ترتیب-محور تک نقطه‌ای

۴-۶- عملگر جهش

عملگر جهش با یک احتمال مشخص (p_m)، محتوای یک یا چند ژن کروموزوم را تغییر داده و باعث ایجاد تنوع در جمعیت می‌شود. این عملگر مانع از همگرایی زودرس شده و احتمال گیر افتادن در جواب‌های بهینه محلی را کاهش می‌دهد. در عملگر جهش مورد استفاده در این پژوهش، یک قطعه به طور تصادفی انتخاب و ماشین آن به طور تصادفی با یکی دیگر از ماشین‌های مجاز تعویض می‌شود.

۴-۷- معیار توقف

معیار توقف در الگوریتم پیشنهادی، رسیدن به تعداد ثابتی از نسل‌ها و یا عدم بهبود تابع هدف در تعداد مشخصی از تکرارهای الگوریتم است.

۵- ارزیابی و اعتبارسنجی نتایج

چون نوآوری این پژوهش در هر دو قسمت مدل‌سازی و روش حل است، تحلیل نتایج محاسباتی در دو بخش تعیین اعتبار مدل و تعیین کارایی روش حل انجام می‌شود: در بخش (۵-۱) اعتبار مدل پیشنهادی با استفاده از مثال‌های عددی در ابعاد کوچک و با مقایسه جواب‌های روش حل دقیق آن با روش شمارش کامل نشان داده می‌شود. در بخش (۵-۲) برای نشان دادن توانایی الگوریتم در همگرایی به سمت جواب‌های بهینه، از جواب دقیق مثال‌های عددی

مقایسه شده‌اند که در ادامه در مورد آن‌ها توضیح داده شده است.

جدول (۳): مناسب‌ترین مقادیر تنظیمی پارامترهای الگوریتم

| پارامترها | مقدار | | |
|-----------------|-----------|------------|-----------|
| | بُعد کوچک | بُعد متوسط | بُعد بزرگ |
| نرخ تقاطع | ۰/۶ | ۰/۶ | ۰/۶۵ |
| نرخ جهش | ۰/۲ | ۰/۲ | ۰/۱۵ |
| اندازه جمعیت | ۱۰ | ۱۰ | ۱۵ |
| تعداد تکرار نسل | ۵۰۰ | ۸۰۰ | ۱۰۰۰ |
| نرخ تولید مجدد | ۰/۵ | ۰/۵ | ۰/۵ |

مقایسه MSGA و GAMS از نظر کیفیت جواب‌ها

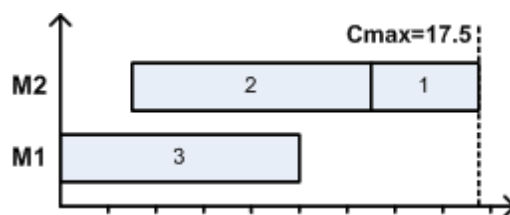
جواب‌های به‌دست آمده برای مسائل در ابعاد مختلف در جدول (۴) و شکل (۵) نشان داده شده است. هدف از مقایسه جواب‌های دو روش MSGA و GAMS در بُعد کوچک، ارزیابی توانایی MSGA در رسیدن به جواب‌های بهینه یا نزدیک بهینه است.

با توجه به جدول (۴) و شکل (۵) مشاهده می‌شود که در چهار مسئله ۱، ۲، ۳ و ۷، جواب‌های MSGA منطبق بر جواب‌های GAMS است و تنها در یک مورد (مسئله ۴)، جواب MSGA از جواب بهینه چهار درصد فاصله دارد که این نکته حاکی از توانایی الگوریتم پیشنهادی در رسیدن به جواب‌های مناسب است. ضمناً در مسائل ۵، ۶، ۸ و ۹، GAMS توانایی رسیدن به جواب بهینه را در محدوده زمانی مشخص شده (۴۰۰۰ ثانیه) نداشت و به همین دلیل در جدول (۴) ستون "جواب" GAMS، علامت (-) گذاشته شده است.

هدف از مقایسه دو الگوریتم MSGA و GAMS در بُعد متوسط و بزرگ، نشان دادن توانایی MSGA در رسیدن به جواب‌های بهینه/ نزدیک بهینه و عدم توانایی روش‌های حل دقیق در این امر است. با توجه به نتایج جدول (۴) و شکل (۵) مشاهده می‌شود که در بُعد متوسط و بزرگ (مسائل ۱۰ تا ۲۷) نرم‌افزار GAMS حتی توانایی رسیدن به یک جواب اولیه شدنی در زمان محاسباتی تعیین‌شده (۴۰۰۰ ثانیه) را نداشته است به همین دلیل در شکل (۵) برای مسائل ۱۰ تا ۲۷ برای روش دقیق، نموداری وجود ندارد. با توجه به نتایج ابعاد مختلف، می‌توان چنین استنباط کرد که MSGA در حل مسائل با ابعاد مختلف کاراست.

مقایسه MSGA و GAMS از نظر زمان اجرا

یکی از معیارهای مهم برای سنجش کیفیت عملکرد یک الگوریتم، مدت زمان اجرای آن است که در برخی مقالات از آن به عنوان سرعت اجرا نیز یاد می‌شود. این معیار وقتی اهمیت بیشتری پیدا می‌کند که ابعاد و پیچیدگی مسئله افزایش یابد. اطلاعات مربوط به



شکل (۳): گانت چارت زمان‌بندی بهینه مثال عددی

| | LOWER | LEVEL | UPPER | MARGINAL |
|---------------|-------|--------|-------|----------|
| --- VAR Z_obj | -INF | 17.500 | +INF | . |
| --- VAR cmax | . | 17.500 | +INF | . |

شکل (۴): خروجی نرم‌افزار GAMS برای مثال عددی

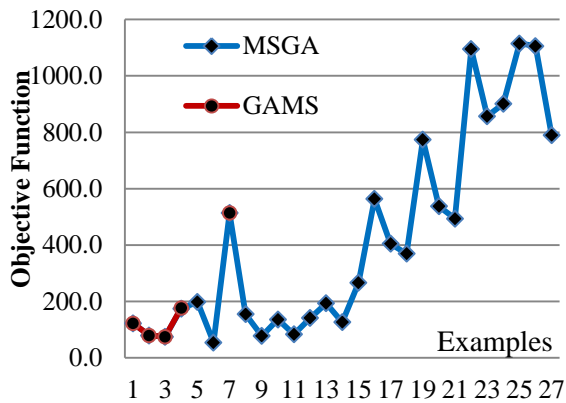
۲-۵- اعتبارسنجی الگوریتم پیشنهادی و تحلیل نتایج

برای ارزیابی کارایی الگوریتم، ۲۷ مسئله در ابعاد بزرگ، متوسط و کوچک تولید شد که داده‌های مربوط به آن‌ها در جدول (۲) آورده شده است. این داده‌ها تا حد امکان از پژوهش‌های پیشین [۸، ۱۲] استخراج و در مواردی که داده‌های بیشتری مورد نیاز بوده، اعداد تصادفی طبق یک الگوی مناسب تولید شده است.

جدول (۲): داده‌های کلی مربوط به مسائل تصادفی تولید شده

| شماره مسئله | مقادیر | | | تعداد |
|-------------|------------|--------------|---------------|------------------|
| | بُعد کوچک | بُعد متوسط | بُعد بزرگ | |
| ۲۷ | ۱۲ و ۸ و ۵ | ۳۰ و ۲۰ و ۱۵ | ۱۰۰ و ۷۰ و ۵۰ | n |
| | ۱ و ۵ و ۲ | ۱۵ و ۱۰ و ۵ | ۲۰ و ۱۵ و ۱۰ | m |
| ۱ | U[0,1] | U[0,1] | U[0,1] | t _{ipk} |
| ۱ | U[0,30] | U[0,30] | U[0,30] | RT _k |
| ۱ | -۰/۳۲۲ | -۰/۳۲۲ | -۰/۳۲۲ | α,β |
| ۲۷ | مجموع | | | |

به دلیل ماهیت تصادفی الگوریتم ژنتیک، این الگوریتم برای هر مسئله ۱۰ بار اجرا شد و بهترین جواب به‌دست آمده به عنوان جواب نهایی مسئله در نظر گرفته شد. به منظور افزایش کیفیت جواب‌های MSGA، مناسب‌ترین مقادیر پارامترهای مورد نیاز از طریق آزمایش‌های عددی به‌دست آمد که مقادیر آن‌ها در جدول (۳) آورده شده است. برای سنجش کارایی الگوریتم پیشنهادی، از معیارهای کیفیت جواب‌ها (مقدار تابع هدف) و زمان اجرا استفاده شده است. با توجه به این دو معیار، دو روش حل دقیق (GAMS) و الگوریتم پیشنهادی (MSGA) در سه بُعد کوچک، متوسط و بزرگ با یکدیگر



شکل (۵): مقایسه جواب دو روش MSGA و GAMS در ابعاد مختلف

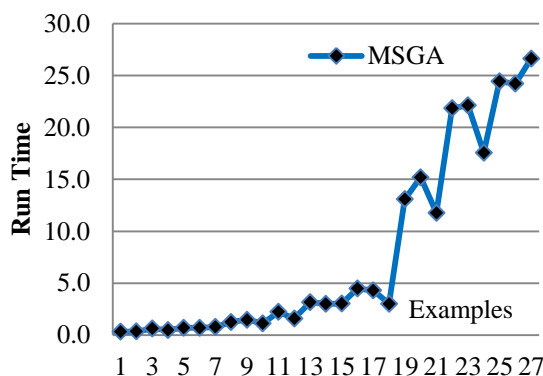
زمان اجرای این دو روش در جدول (۴) و شکل (۶) آورده شده است. با توجه به اختلاف زیاد زمان اجرای روش حل دقیق و پیشنهادی، مقایسه این دو روش از نظر زمان اجرا به نظر مناسب نمی‌رسد. به همین دلیل در این قسمت فقط به تحلیل زمان اجرای الگوریتم پیشنهادی در ابعاد مختلف پرداخته می‌شود. به طور کلی در شکل (۶) مشاهده می‌شود که زمان اجرای MSGA با افزایش بُعد مسأله (تعداد کارها) به صورت نمایی افزایش می‌یابد. دلیل پلکانی بودن این نمودار از مسأله ۱۸ به بعد این است که پیچیدگی مسأله و زمان اجرای الگوریتم، بیشتر وابسته به n است نه m ($O(n^2m)$). به همین دلیل مشاهده می‌شود که در مسائل ۱۸ به بعد، هر سه نقطه متوالی که n یکسان و m متفاوت دارند تقریباً زمان اجرای یکسان دارند و با تغییر n (مانند تغییرات بین نقطه ۲۱ و ۲۲)، زمان اجرا افزایش قابل ملاحظه و ناگهانی داشته است.

جدول (۴): اطلاعات مربوط به جواب‌های MSGA و GAMS در ابعاد مختلف

| بُعد | ردیف | مسأله ($n \times m$) | GAMS | | MSGA | |
|-------|------|------------------------|--------|-------------------|--------|--------------|
| | | | جواب | زمان اجرا (ثانیه) | جواب | انحراف معیار |
| | ۱ | 5×2 | ۱۲۲/۵ | ۱۶۲ | ۱۲۲/۵ | ۰ |
| | ۲ | 5×5 | ۷۹ | ۳/۹ | ۷۹ | ۰ |
| | ۳ | 5×10 | ۷۴/۶ | ۲۳/۲ | ۷۴/۶ | ۰ |
| | ۴ | 8×2 | ۱۶۷/۳۵ | ۱۵/۴ | ۱۷۴/۷ | ۰ |
| کوچک | ۵ | 8×5 | - | - | ۱۹۸/۱ | ۰/۰۳ |
| | ۶ | 8×10 | - | - | ۵۴/۲ | ۱۷/۲۸ |
| | ۷ | 12×2 | ۵۱۳/۳ | ۵۶/۵ | ۵۱۳/۳ | ۰ |
| | ۸ | 12×5 | - | - | ۱۵۴/۶ | ۱۹/۴۰ |
| | ۹ | 12×10 | - | - | ۷۷/۲ | ۱۷/۳۷ |
| | ۱۰ | 15×5 | - | - | ۱۳۶ | ۹/۹۰ |
| | ۱۱ | 15×10 | - | - | ۸۲/۹ | ۱۷/۷۰ |
| | ۱۲ | 15×15 | - | - | ۱۴۱ | ۷/۹۴ |
| | ۱۳ | 20×5 | - | - | ۱۹۳/۲ | ۱۱/۴۷ |
| متوسط | ۱۴ | 20×10 | - | - | ۱۲۶/۵ | ۱۷/۵۲ |
| | ۱۵ | 20×15 | - | - | ۲۶۵/۸ | ۲۷/۰۴ |
| | ۱۶ | 30×5 | - | - | ۵۶۳/۶ | ۵۸/۴۸ |
| | ۱۷ | 30×10 | - | - | ۴۰۴/۴ | ۴۶/۴۹ |
| | ۱۸ | 30×15 | - | - | ۳۶۹/۶ | ۳۰/۲۵ |
| | ۱۹ | 50×10 | - | - | ۷۷۴/۱ | ۵۷/۵۴ |
| | ۲۰ | 50×15 | - | - | ۵۳۷/۵ | ۷۰/۹۷ |
| | ۲۱ | 50×20 | - | - | ۴۹۳/۶ | ۶۵/۱۳ |
| | ۲۲ | 70×10 | - | - | ۱۰۹۵ | ۹۱/۱۱ |
| | ۲۳ | 70×15 | - | - | ۸۵۶/۸ | ۷۴/۹۸ |
| بزرگ | ۲۴ | 70×20 | - | - | ۹۰۱ | ۸۰/۰۸ |
| | ۲۵ | 100×10 | - | - | ۱۱۱۵/۱ | ۱۲۱/۱۲ |
| | ۲۶ | 100×15 | - | - | ۱۱۰۵/۵ | ۱۳۶/۵۳ |
| | ۲۷ | 100×20 | - | - | ۷۹۰/۳ | ۹۶/۴۵ |

۷- مراجع

- [1] Gen, M., Cheng, R. and Lin, L. (2008), Advanced Planning and Scheduling Models, in Network Models and Optimization Decision Engineering, p.p. 297-417.
- [2] Moon, C. and Seo, Y. (2005), Evolutionary algorithm for advanced process planning and scheduling in a multi-plant, Computers & Industrial Engineering, vol. 48, p.p.311-325.
- [3] Moon, C., Seo, Y. and Yun, Y. (2006) "Adaptive genetic algorithm for advanced planning in manufacturing supply chain," J Intell Manuf, vol. 17: 509-522.
- [4] Blazewicz, J., Ecker, K. H., Schmidt, G. and Weglarz, J., (1994), Scheduling in Computer and Manufacturing Systems," Springer.
- [5] Bermudez, J., (1998), Advanced Planning and Scheduling: Is it as good as it sounds?, The report on Supply Chain Management, March, p.p. 3-18.
- [6] Marjolein and van Eck, (2003), Is logistics everything, a research on the use(fullness) of advanced planning and scheduling systems, in BWI paper, April.
- [7] Lee, Y.H., Jeong, C.S. and Moon, C., (2002), Advanced planning and scheduling with outsourcing in manufacturing supply chain, Computers & Industrial Engineering, vol. 43, pp. 351-374,.
- [8] Chen, K. and Ji, P. A., (2007), mixed integer programming model for advanced planning, European Journal of Operational Research, vol. 181, pp. 515-522.
- [9] Yang, J., Tang, W., (2009), Preference-based adaptive genetic algorithm for multiobjective advanced planning and scheduling problem, IEEE International Conference, pp. 1935 - 1940.
- [10] Zhang, H. and Gen, M., (2006), Effective Genetic Approach for Optimizing Advanced Planning and Scheduling in Flexible Manufacturing System, GECCO, pp. 1841-1848.
- [11] Örnek, A., Özpeynirci, S. and Öztürk, C., (2010), A note on "A mixed integer programming model for advanced planning, European Journal of Operational Research, vol. 203, pp. 784-785.
- [12] Chen, K., Ji, P. and Wang, Q., (2011), A case study for advanced planning and scheduling (APS), J Syst Sci Syst Eng, Dec, vol. 20, no. 4, pp. 460-474.
- [13] Baker, K. R., (1974), Introduction to sequencing and scheduling, John Wiley & Sons.
- [14] Okołowski, D. and Gawiejnowicz, S., (2010), Exact and heuristic algorithms for parallel-machine scheduling with DeJong's learning effect, Computers & Industrial Engineering, vol. 59, pp. 272-279.
- [15] ذگردی، سید حسام‌الدین؛ بهلولی، احسان (۱۳۸۸). زمان‌بندی گروهی با در نظر گرفتن اثر یادگیری در سیستم تولید سلولی، نشریه بین‌المللی مهندسی صنایع و مدیریت تولید، شماره ۲، جلد ۲۰.
- [16] Biskup, D. A., (2008), state-of-the-art review on scheduling with learning effects, European Journal of Operational Research, vol. 188, pp. 315-329.
- [17] Wright, T.P., (1936) Factors affecting the cost of airplanes, Journal of Aeronautical Sciences 3, pp. 122-128.
- [18] Biskup, D., (1999), Single-machine scheduling with



شکل (۶): نمودار زمان اجرای MSGA. در ابعاد مختلف

۶- نتیجه‌گیری

تاکنون یک فرض رایج در مسائل برنامه‌ریزی و زمان‌بندی پیشرفته این بود که زمان تولید یک محصول، ثابت و مستقل از جایگاهش در توالی تولید است. در حالی که بسیاری از محققین با بررسی برنامه زمان‌بندی تجربی و تئوری به این نتیجه رسیده‌اند که زمان‌های پردازش کارها تراکم‌پذیر هستند که یکی از رویکردها برای کاهش زمان پردازش کارها و در نظر گرفتن تراکم‌پذیری، استفاده از پدیده یادگیری است. به همین دلیل در این مقاله برای اولین بار عامل‌های انسانی در امر مدل‌سازی دخیل و یک رویکرد جدید برای اعمال اثر یادگیری در مسأله برنامه‌ریزی و زمان‌بندی پیشرفته ارائه شد. همچنین با توجه به پیچیدگی محاسباتی بالای مدل حاصله، یک الگوریتم ژنتیک چندمرحله‌ای نیز برای حل مدل پیشنهاد و کارایی آن با توجه به معیارهای سرعت و کیفیت (از طریق مقایسه آن با روش‌های حل دقیق) آزمون شد.

نتایج به دست آمده نشان داد که الگوریتم ژنتیک پیشنهادی تنها در یک مورد از مقدار بهینه مسأله چهار درصد انحراف دارد و در بقیه موارد (که در آن‌ها جواب دقیق مسأله وجود داشته) توانایی دست یافتن به جواب‌های بهینه مسأله را دارد.

با توجه به کاربردی بودن مدل ارائه شده در این مقاله، توصیه می‌شود علاقه‌مندان به عنوان تحقیقات آتی به عنوان مطالعه موردی به بررسی پیاده‌سازی این مبحث در مدل‌های تولید چند محصولی و مونتاژی و تعیین نرخ یادگیری در محیط‌های تولیدی ایران بپردازند. همچنین علاقه‌مندان به حیطه روش‌های حل فراابتکاری می‌توانند الگوریتم ژنتیک چندمرحله‌ای پیشنهادی این پژوهش را با دیگر الگوریتم‌های فراابتکاری ترکیب کرده و کارایی آن را بهبود بخشند.

- International Conference on, Singapore, pp. 1271 - 1274.
- [24] Hamta, N., Fatemi Ghomi, S.M.T., Asiabar, M.H. and Hooshangi Tabrizi, P., (2011), Multi-objective Assembly Line Balancing Problem with Bounded Processing Times, Learning Effect, and Sequence-dependent Setup Times, in Industrial Engineering and Engineering Management (IEEM), 2011 IEEE International Conference on, Singapore, pp. 768-772.
- [25] Hamta, N., FatemiGhomi, S.M.T., Jolai, F. and Akbarpour Shirazi, M., (2012), A hybrid PSO algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect, Int. J. Production Economics: <http://dx.doi.org/10.1016/j.ijpe.2012.03.013>.
- [26] Behnamian, J. and Fatemi Ghomi, S.M.T., (2011), Hybrid flowshop scheduling with machine and resource-dependent processing times, Applied Mathematical Modelling, vol. 35, pp. 1107-1123.
- [۲۷] دب، کالیانموی. الگوریتم‌های ژنتیک با رویکرد بهینه‌یابی چندهدفه؛ رضایی، جعفر؛ داودی منفرد (پاییز ۱۳۸۷)، منصور؛ انتشارات دانشگاه ولی عصر رفسنجان.
- learning considerations, European Journal of Operational Research, vol. 115, pp. 173-178.
- [19] Kuo, W-H. and Yang, D-L., (2006), Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect, European Journal of Operational Research, vol. 174, pp. 1184-1190.
- [20] 20] Janiak, A. and Rudek, R., (2008), A new approach to the learning effect: Beyond the learning curve restrictions, Computers & Operations Research, vol. 35, pp. 3727 – 3736.
- [21] Wang, J-B. and Wang, J-J., (2012), Scheduling jobs with a general learning effect model, Appl. Math. Modelling, pp. article in press.
- [22] Zhang, X., Yan, G., Huang, W. and Tang, G., (2012), A note on machine scheduling with sum-of-logarithm-processing-time-based and position-based learning effects, Information Sciences, vol. 187, pp. 298-304.
- [23] Amini, A., TavakkoliMoghaddam, R. and Niakan, F. A., (2011), Multi-Objective Identical Parallel Machine Scheduling with Setup and Removal Times with Deteriorating and Learning Effects, in Industrial Engineering and Engineering Management (IEEM), IEEE