کمینه‌سازی هزینه توقف خط مونتاژ با استفاده از الگوریتم ژنتیک

رضا توکلی مقدم*، مسعود ربانی* و محمدعلی بهشتی**
گروه مهندسی صنایع، دانشکده فنی، دانشگاه تهران
گروه مهندسی صنایع، دانشگاه علوم و فنون مازندران

(دریافت مقاله: ۸۴/۷/۱۹- دریافت نسخه نهایی: ۸۶/۱۰/۹)

چکیده – در این مقاله، یک مدل برنامه‌ریزی غیرخطی عدد صحیح ترکیبی با هدف کاهش هزینه توقف خط در خطوط مونتاژ چند مدلی ارایه می‌شود. امروزه با توجه به دلیل تنوع زیاد محصولات برای پاسخگویی سریع به تقاضاهای متنوع مشتریان، اکثر شرکتهای تولیدی از خطوط مونتاژ چند مدلی استفاده می‌کنند. از آنجا که معمولا زمان مونتاژ، ترکیب و مقدار قطعات مورد نیاز برای هر مدل متفاوت است، حداقل کردن توقف خط یک عامل عمده که در تعیین توالی پردازش محصولات مطرح می‌شود. توقف خط موجب بیکاری اپراتورها، بیکاری ماشینها، کاهش تولید و افزایش هزینه سربار و در نهایت کاهش بهره‌وری می‌شود. به علت پیچیدگی این مدل که جزء خانواده مسایل NP-hard است، یک روش فراابتکاری بر اساس الگوریتم ژنتیک برای به دست آوردن حلهای نزدیک به بهینه در ابعاد بزرگ و در زمان معقول طراحی و پیشنهاد می‌شود. برای نشان دادن کارایی این الگوریتم پیشنهادی، نتایج محاسباتی با حلهای به دست آمده از نرم‌افزار لینگو مقایسه شده است.

واژگان کلیدی : خط مونتاژ چند مدلی، هزینه توقف، الگوریتم ژنتیک

Minimizing Stoppage Cost of an Assembly Line Using Genetic Algorithm

R. Tavakkoli-Moghaddam, M. Rabbani, and M.A. Beheshti

Department of Industrial Engineering, Faculty of Engineering, University of Tehran
Department of Industrial Engineering, Mazandaran University of Science and Technology

Abstract: ***This paper presents a nonlinear mixed-integer programming model to minimize the stoppage cost of mixed-model assembly lines. Nowadays, most manufacturing firms employ this type of line due to the increasing varieties of products in their attempts to quickly respond to diversified customer demands. Advancement of new technologies, competitiveness, diversification of products, and large customer demand have encouraged practitioners to use different methods of improving production lines. Minimizing line stoppage is regarded as a main factor in determining the sequence of processing products. Line stoppage results***

* – دانشیار ** – کارشناس ارشد

in idleness of operators and machines, reduced throughput, increased overhead costs, and decreased overall productivity. Due to the complexity of the model proposed, which belongs to a class of NP-hard problems, a meta-heuristic method based on a genetic algorithm (GA) is proposed to obtain near-optimal solutions in reasonable time, especially for large-scale problems. To show the efficiency of the proposed GA, the computational results are compared with those obtained by the Lingo software.

Keywords: *Mixed-model assembly lines, Stoppage cost, GA.*

۱- مقدمه

در این مقاله، هدف اصلی نشان دادن کارایی الگوریتم ژنتیک در مقایسه با حل بهینه برای مسئله خط مونتاژ چندمدلی در اندازه کوچک است. ابتدا مدلی برای مسئله مورد نظر با هدف تعیین توالی خط مونتاژ چندمدلی بر اساس حداقل کردن توقفات خط مونتاژ در سیستم تولید به موقع در نظر گرفته می‌شود. سپس رویه حل بر اساس الگوریتم ژنتیک ارائه می‌شود و جوابهای به دست آمده با حل بهینه مقایسه خواهد شد.

محصولات، از مونتاژ قسمتهای مختلف یک سیستم تولیدی تشکیل می‌شوند. مونتاژ، عمل پیوند دادن دو یا چند قطعه مجزا و تشکیل یک نهاد جدید است. خط مونتاژ، که یک نوع سیستم مونتاژ با جریان کار است، برای تولید با حجم زیاد و نگهداری موجودی کم استفاده می‌شود. کار می‌تواند به فعالیتهای کوچک به نام عناصر کاری تقسیم شود و امکان تخصیص فعالیتها به ایستگاهها روی خط وجود دارد، یک خط مونتاژ شامل دو عنصر اصلی است: یکی انجام‌دهنده عملیات یا اپراتور[۱] و دیگری سیستم انتقال مواد[۲].

یک خط مونتاژ شامل چند ایستگاه کاری است که اپراتورها در آنها عملیات روی قطعه‌کار را انجام می‌دهند. محصول در حین انجام عملیات ممکن است به مصرف قطعه یا قطعاتی نیاز داشته باشد. سیستم انتقال مواد معمولا به صورت نقاله متحرک است که جریان کار بین ایستگاهها را انجام می‌دهد. قطعه کار در هر ایستگاه کاملتر شده و در نهایت به صورت محصول تکمیل شده از خط خارج می‌شود [۱].

در خط مونتاژ چند مدلی، چند نوع مدل از یک نوع محصول تولید می‌شود. بنابراین در این حالت علی‌رغم وجود تشابه فرایند هر مدل در خط مونتاژ به علت تفاوت مشخصات مدلها، زمان عملیات هر مدل، مشخصات قطعات و میزان مصرف آنها در مدلهای مختلف، سیستم مونتاژ دارای پیچیدگی بیشتری نسبت به حالت تک مدلی است [۱و۲].

سیستم تولید به موقع[۳] (JIT)، ابتدا در شرکت تویوتا با عنوان سیستم تولیدی تویوتا مطرح و مورد استفاده قرار گرفت. ایده اصلی سیستم تولیدی تویوتا برقراری یک جریان پیوسته محصولات درکارخانه‌ها به منظور انعطاف‌پذیری برای تطابق با تغییرات تقاضای بازار است. تحقق این ایده از طریق JIT و اطمینان از کیفیت میسر است. تولید JIT به این معنی است که اقلام لازم را به مقدار لازم و در زمان لازم تولید کنیم. به عنوان مثال در فرایند مونتاژ اتومبیل ،انواع زیر مونتاژهای مورد نیاز باید در زمان لازم و به مقدار لازم از فرایندهای قبلی به خط مونتاژ وارد شوند. اگر JIT در تمام بخشهای شرکت انجام شود آن گاه موجودی زائد در کارخانه تقریبا به طور کامل حذف می‌شود و به دنبال آن لزوم وجود انبارهای زائد نیز حذف خواهد شد، هزینه‌های نگهداری موجودی حداقل می‌شود و نرخ برگشت سرمایه افزایش خواهد یافت [۲].

۲- ادبیات موضوع

مسئله تعیین توالی خط مونتاژ چندمدلی، از دهه ۱۹۶۰ مطرح شد. فعالیتهای محققان را می‌توان به دو بخش عمده تحقیقات قبل و پس از ظهور سیستم تولیدی JIT تقسیم کرد.

در بخش اول، مسئله تعیین توالی تنها با توجه به معیار بازدهی خط تعریف می‌شدند. بخش دوم بیشتر تحقیقات متوجه معیار بازدهی تولید قطعات‌اند. مقالاتی در زمینه بازدهی خط مونتاژ و یا هر دو معیار فوق الذکر نیز وجود دارد که تحقیقات مرتبط در ادامه ارائه می‌شود.

ماندن [۲] علاوه بر بیان روشهای Goal-Chasing که تنها به بازدهی تولید قطعات می‌پردازند، به روش شرکت تویوتا برای درنظر گرفتن معیار بازدهی خط مونتاژ در تعیین توالی محصولات نیز اشاره می‌کند. در سیستم تولیدی تویوتا، تعادل خط به صورتی انجام می‌شود که مدل اتومبیلی که دارای زمان مونتاژ کل بزرگتری است همیشه دارای زمان عملیات بیشتری در هر فرایند (ایستگاه) خط است. برای رفع امکان ایجاد ترتیبهای متوالی از محصولاتی با زمانهای بزرگ، تمام اتومبیلها براساس زمان کل مونتاژ به سه طبقه بزرگ، متوسط و کوچک تقسیم می‌شوند. یک محصول از بین این سه طبقه طوری انتخاب می‌شود که تعدیل در زمانهای مونتاژ کل در طول افق برنامه‌ریزی ایجاد شود یا به عبارت دیگر سرعت ثابت خط حفظ شود.

یاماشینا و اوکامورا [۳] در حالت خط مونتاژ پیوسته فرمولبندی جدیدی با هدف حداقل کردن هزینه توقف نوار نقاله ارائه می‌کنند و سپس یک روش فراابتکاری کارایی را برای مسائل با مقیاس بالا معرفی می‌کنند. در این مدل برای حداقل کردن هزینه توقف نوار نقاله سعی می‌شود که حداکثر نقطه اتمام عملیات محصولات روی ایستگاهها حداقل شود. آنها ادعا می‌کنند که اگر حداکثر نقطه ابتدای عملیات مدلها روی ایستگاهها نیز حداقل شود آن گاه جواب یکسانی با حالت قبل وجود خواهد داشت. در روش فراابتکاری آنها، ابتدا یک ترتیب اولیه تهیه می‌شود و مقدار فوق به دست می‌آید. سپس با تعویض محصولات در ترتیب سعی می‌شود که این مقدار کاهش یابد.

بارد و همکاران [۴] در حالت خط مونتاژ پیوسته با توجه به مشخصه‌های مهم طراحی خط، چند مدل بهینه‌سازی تعیین توالی را ارائه می‌کنند. آنها از ترکیب این مشخصات و دو هدف طراحی شامل حداقل کردن طول خط و حداقل کردن مدت زمان اتمام کل تولید چند مدل بهینه‌سازی ارائه می‌کنند. تامپولس [۵] برای خط مونتاژ پیوسته، مدلی را ارائه می‌دهد. در این مدل، ترکیبی از چهار نوع وضعیت بسته، باز، بسته به راست و باز به چپ، بسته به چپ و باز به راست برای ایستگاهها در نظر گرفته می‌شود. چهار نوع معیار عدم کارایی خط را با توجه به وضعیت ایستگاه تعریف می‌کند.

کوتر و دارل [۶] برای حالت خط مونتاژ پیوسته، مدلهایی را در دو حالت جداگانه ایستگاههای باز و بسته ارائه می‌کنند. هدف مدل آنها این است که برای یک تقاضای مشخص، ترتیبی به دست آورند که طول خط مونتاژ حداقل شود. سپس الگوریتمهای فراابتکاری را برای حل مدلها ارائه می‌کنند.

خیابو و اونو [۷و۸] با ارائه دو مقاله به فاصله سه سال از هم در مورد توقف خط بحث می‌کنند. آنها در مقاله اول با استفاده از روش شاخه و کران، می‌خواهند ترتیب بهینه‌ای بیابند که کل زمان توقف خط را حداقل کند و در مقاله دوم با بیان مجدد الگوریتم شاخه و کران که تنها برای مسایل کوچک می‌توان استفاده کرد، استفاده از روش SA[۴] برای به دست آوردن حلهای تقریبا بهینه برای مسایل بزرگ را مفید می‌دانند.

بولات و همکاران [۹] یکی از مسایل خطوط مونتاژ چند مدلی را پیدا کردن ترتیب بهینه‌ای که هزینه‌های ناشی از ناکارایی ایستگاهها را حداقل کند، معرفی می‌کنند. آنها این ناکارایی را ناشی از آماده سازیها[۵] و مطلوبیت کارها[۶] می‌دانند. آنها مسئله را به صورت یک برنامه‌ریزی ریاضی عدد صحیح مختلط فرموله کرده و در ادامه از الگوریتم شاخه و کران برای حل آن استفاده کرده‌اند. آنها با توجه به محدودیت الگوریتم شاخه و کران از نظر محاسبات، دو الگوریتم فراابتکاری برای حل مسایل واقعی با زمان محاسبه قابل قبولی ایجاد کرده‌اند.

توکلی مقدم و رحیمی واحد [۱۰] بیان می‌کنند که خطوط مونتاژ چند مدلی نوعی از خطوط مونتاژ هستند که مدلهای مختلف محصولات با مشخصات مشابه در سیستم تولیدی JIT مونتاژ می‌شوند. آنها سه هدف را به طور همزمان بررسی کردند که این سه هدف عبارت‌اند از : کل هزینه کار مفید، کل هزینه انحراف نرخ تولید و کل هزینه آماده‌سازی. سپس این سه هدف را بر اساس اهمیت وابستگی وزن‌دهی کردند و مدل ریاضی جدیدی ارائه کردند. برای حل این مدل و تعیین توالی مناسب الگوریتم ممتیک[۷] (MA) را پیشنهاد کرده و کارایی روش MA پیشنهادی را با نرم‌افزار لینگو ۶ مقایسه کرده‌اند. تعداد مسایل

آزمایشی اجرا شده، توانایی خوب MA پیشنهادی را با توجه به زمان محاسبات و کیفیت جوابها تایید می‌کند. نتایج محاسباتی نشان می‌دهد که MA، نتایج رضایت‌بخشی را به خصوص در مسایل با ابعاد بزرگ ارائه می‌دهد.

رحیمی واحد و همکاران [۱۱] توالی تولید برای خط مونتاژ چند مدلی در سیستمهای JIT با توجه به مدل چند هدفه ارائه کرده‌اند. این مدل، سه هدف اصلی را به طور همزمان حداقل می‌کند: کل کار مفید، کل انحراف نرخ تولید و کل هزینه آماده‌سازی. به علت اینکه مسئله مورد نظر جزو مسایل NP-hard است، لذا یک روش جدید بر اساس MOPS[۸] برای جستجوی موضعی مرز بهینه پارتو طراحی کردند. برای اعتباردهی کارایی الگوریتم پیشنهادی، مسایل آزمایشی مختلف حل شده‌اند و قابلیت اطمینان الگوریتم پیشنهادی بر اساس برخی استانداردهای مقایسه‌ای، با سه الگوریتم ژنتیک چند هدفه متمایز PS-NC GA ، NSGA-II و SPEA-II مقایسه شد. مقایسه‌ها نشان می‌دهد که MOPS نتایج بهتری نسبت به الگوریتمهای ژنتیک چند هدفه ارائه می‌کند.

رحیمی واحد و همکاران [۱۲] سه هدف را به طور همزمان بررسی کرده‌اند که این اهداف عبارت‌اند از: حداقل کردن کل کار مفید، انحراف کل نرخ تولید و کل هزینه آماده‌سازی. یک مسئله توالی چند هدفه و فرمولبندی آن را تشریح کرده‌اند. از آنجایی که مسئله مورد نظر جزو مسایل NP-hard است، لذا یک روش جدید بر اساس MOSS[۹] برای جستجوی موضعی مرز بهینه پارتو برای مسئله طراحی کرده‌اند. برای اعتباردهی کارایی الگوریتم پیشنهاد شده برحسب کیفیت جواب و تنوع سطح، مسایل آزمایشی مختلف ساخته شده‌اند و قابلیت اطمینان الگوریتم پیشنهادی بر اساس برخی استانداردهای مقایسه‌ای، با سه الگوریتم ژنتیک چند هدفه مهم یعنی PS-NC GA، NSGA-II و SPEA-II مقایسه شد. نتایج محاسباتی نشان می‌دهد که MOSS کارایی بیش از انتظار نسبت به الگوریتمهای ژنتیک موجود به خصوص برای مسایل با ابعاد بزرگ، ارائه می‌کند.

جیانفنگ و همکاران [۱۳] مسایل برنامه‌ریزی زمانبندی به دلیل دارا بودن فضای شدنی بسیار بزرگ و تعداد زیاد جوابهای بهینه موضعی به دست آمده، جزو مسایل ترکیبی مشکل به حساب می‌آیند. آنها یک الگوریتم ژنتیک چند هدفه را به عنوان یک الگوریتم هوشمند برای برنامه زمانبندی خط مونتاژ چند مدلی ارائه کردند. روش رتبه‌دهی پارتو و روش پراکندگی فاصله‌ای را برای ارزیابی برازندگی استفاده کردند. نتایج محاسباتی نشان می‌دهد که الگوریتم ژنتیک چند هدفه پیشنهادی، کاراست.

سیماریا و ویلارینهو [۱۴] یک مدل برنامه‌ریزی ریاضی و یک رویه الگوریتم ژنتیک تکرارشونده برای مسئله تعادل خط مونتاژ چند مدلی در ایستگاههای کاری موازی را ارائه کرده‌اند، که در آن هدف حداکثر کردن نرخ تولید خط با تعداد اپراتورهای معین است. مسئله مذکور، در مورد برخی مسایل مربوط به محدودیتهای عملیاتی خطوط مونتاژ در دنیای واقعی، همانند محدودیتهای منطقه‌بندی و بالانس بارکاری نیز صادق است. همچنین به تصمیم‌گیرنده اجازه می‌دهد که ایجاد ایستگاههای کاری موازی را تحت کنترل خود درآورد.

منصوری [۱۵] یک الگوریتم ژنتیک چند هدفه با نگرش بر مشکل توالی سیستم JIT را ارائه کرد، که در آن انحراف نرخ تولید و تعداد آماده‌سازیها به طور همزمان بهینه می‌شوند. این دو هدف به نوعی ارتباط معکوس با یکدیگر دارند، بنابراین بهینه‌سازی همزمان هر دو آنها مشکل است. این نوع مسئله جزو مسایل NP-hard بوده و دستیابی به راه‌حل برنامه‌ریزی خطی، برنامه‌ریزی عدد صحیح یا به وسیله راه‌حل TE[۱۰] محاسباتی، مشکل است. در جایی که حداقل‌سازی انحرافات نرخ تولید و تعداد آماده‌سازیها به طور همزمان مورد نظر باشد، روش الگوریتم ژنتیک چند هدفه برای یافتن بهینه موضعی پارتو به کار می‌رود. کارایی الگوریتم ژنتیک چند هدفه پیشنهادی، در مقابل رویه TE برای مسایل کوچک و همچنین در مقابل سه روش جستجوی ابتکاری برای مسایل کوچک، متوسط و بزرگ مقایسه شده است. نتایج عملی نشان می‌دهد که کارایی الگوریتم ژنتیک چند هدفه در مقایسه با TE، بسیار

خوب بوده و به طور قابل ملاحظه‌ای زمان کمتری داشته است.

پونامبالام و همکاران [۱۶] کارایی الگوریتم ژنتیک را برای مسایل توالی در خطوط مونتاژ چند مدلی بررسی کردند. مسئله‌ای که در ابتدا مورد بررسی قرار گرفته، مقایسه‌ای بین الگوریتم موجود و الگوریتم ژنتیک پیشنهادی برای رسیدن به استفاده یکنواخت هر قطعه مورد استفاده در خط، با در نظر گرفتن تغییرات در چهار سطح (محصول، زیر مونتاژ، اجزا و مواد اولیه) بوده است. آنها الگوریتم پیشنهاد شده توسط میلتنبرگ و سینامون در سال ۱۹۹۲را با الگوریتم ژنتیک پیشنهادی به کار برده شده برای خط مونتاژ چند مدلی مقایسه کرده‌اند. نتایج ارزیابی نشان می‌دهد که کارایی الگوریتم ژنتیک در بیش از ۲۵ مسئله از ۴۰ مسئله بررسی شده، بهتر است. مسئله حل شده دیگر، مسئله توالی چند هدفه در خطوط مونتاژ چند مدلی است. سه هدف مهم کاربردی عبارت‌اند از: حداقل‌سازی کل کار مفید با حفظ نرخ یکنواخت مصرف قطعات، حداقل‌سازی تغییرپذیری در مصرف قطعات و حداقل‌سازی هزینه آماده‌سازی کل. آنها کارایی مکانیزمهای انتخاب سطح مقعر پارتو و انتخاب بر پایه مقدار تابع برازش عددی با توجه به اهداف حداقل‌سازی انحراف مصرف قطعات، حداقل‌سازی کل کارمفید و حداقل‌سازی کل هزینه آماده‌سازی را با هم مقایسه کرده‌اند. نتایج ارزیابی نشان می‌دهد که الگوریتم ژنتیک که از رویه سطح مقعر پارتو استفاده کرده، کارایی بهتری نسبت به الگوریتم ژنتیک که از مکانیزمهای انتخاب دیگر استفاده می‌کنند، دارد.

اردال ارل و همکاران [۱۷] از یک روش BS[11] برای حل مسئله توالی مدل در خطوط مونتاژ چند مدلی استفاده می‌کنند. به ویژه، آنها روی شش الگوریتم BS برای انحراف مصرف قطعات و سنجش کارایی تسطیح بار کار کردند. نتایج محاسباتی آزمایشها نشان می‌دهد که روشهای پیشنهادی BS با روشهای ابتکاری قابل رقابت‌اند.

مینگژو جین و دیوید وو [۱۸] روش GS[12] را یک الگوریتم عمومی در سیستم JIT، برای مسئله بالانس خط مونتاژ چندمدلی بیان کرده‌اند. آنها تعریفی برای اجزای قطعات و توالی ماندگاری قطعات در نظر گرفتند و ارتباط آنها را با مقدار تابع هدف مربوط به جوابهای بهینه، تجزیه و تحلیل کردند. آنها یک الگوریتم ابتکاری جدیدی به نام "الگوریتم واریانس" ایجاد کردند. آزمایشات عددی نشان می‌دهند که الگوریتم جدید می‌تواند جوابهای بهتری را با محاسبات اضافی کمتر ارائه دهد.

باکچین و رابینوویچ [۱۹] یک فرض عمومی در ادبیات بالانس خط مونتاژ چند مدلی آن است که یک فعالیت مشترک در مدلهای گوناگون، باید به یک ایستگاه تخصیص یابد. آنها این محدودیت را برداشته، و به یک فعالیت مشترک اجازه می‌دهند که به ایستگاههای مختلف برای مدلهای گوناگون تخصیص یابد. آنها به دنبال حداقل‌سازی مجموع هزینه‌های ایستگاهها و فعالیت تکراری اند. سپس یک روش حل بهینه بر پایه الگوریتم شاخه و کران برگشتی ایجاد می‌کنند و کارایی آن را به وسیله مجموعه‌ای از آزمایشات بزرگ مورد ارزیابی قرار می‌دهند. سپس یک روش ابتکاری بر پایه روش شاخه و کران برای حل مسایل با ابعاد بزرگ بسط داده شده است. جوابهای ابتکاری با یک حد پایین مقایسه شده و آزمایشات نشان می‌دهند که روشهای ابتکاری حلهای خیلی بهتری نسبت به روشهای سنتی ارائه می‌دهند.

نورالحق و همکاران [۲۰] تولید انبوه صرفه‌جویی هزینه‌های زیادی را برای تولید کننده‌ها در صنایع گوناگون فراهم می‌آورد. با رشد روند تغییرات بیشتر و دوره عمر کوتاهتر محصولات، تولید انبوه سنتی با خطوط مونتاژ جایگزین می‌شود. بازار کنونی کاملا رقابتی و مشتری محور است. خطوط مونتاژ چند مدلی در بسیاری از محیطهای صنعتی رو به افزایش هستند. آنها به بالانس خطوط مونتاژ چند مدلی برای n مدل می‌پردازند و یک روش الگوریتم ژنتیک کلاسیک را برای حداقل‌سازی تعداد ایستگاههای کاری به کار می‌برند. آنها همچنین یک رویه الگوریتم ژنتیک تلفیقی را به کار گرفتند، که جواب را از روش رتبه‌دهی اصلاحی برای حلهای اولیه برای کاهش فضای جستجو نسبت به کل فضا، از راه کاهش زمان جستجو، به

دست می‌آورد. این رویه در قالب چندین مثال نشان داده شده است. نرم افزار مورد استفاده برای برنامه‌نویسی این رویه، زبان C++ است.

۳- مسئله حداقل کردن توقفات خط

این مسئله در خطوط مونتاژ چند مدلی از نوع پیوسته ایجاد می‌شوند. در این حالت چون اپراتور ضمن انجام عملیات حرکت می‌کند، بنابراین عملیات در هر ایستگاه باید در محدوده ایستگاه انجام گیرد. اگر طول ایستگاه m را L_m و سرعت حرکت خط را ثابت Vc بنامیم آن گاه زمان ایستگاه (STm) یا حداکثر مدت زمانی که می‌توان در طول L_m عملیاتی را انجام داد برابر است با:

$$ST_m = \frac{L_m}{V_c}$$

در خطوط مونتاژ چندمدلی، تمام مدلها دارای زمان عملیات یکسانی در هر ایستگاه نیستند. اگر فرض کنیم که زمان عملیات تمام مدلها در یک ایستگاه کوچکتر از STm باشد باز هم به علت توالی عملیات قطعه‌کارها ممکن است اپراتور نتواند در طول محدوده ایستگاه، عملیات یک قطعه کار را تکمیل کند. در این صورت یک قطعه کار ناتمام در آن ایستگاه به وجود می‌آید. این توقفات (به طور کلی وقوع قطعه کار ناتمام) سبب تاخیر در تکمیل عملیات مدلها و افزایش زمان کل مونتاژ محصول می‌شود [۳ و ۴]. بنابراین مسئله ترتیب بندی محصولات روی خط مونتاژ چند مدلی بدین معناست که با چه ترتیبی محصولات روی خط مونتاژ تولید می‌شوند به نحوی که اهداف مطلوب در نظر گرفته شوند.

۳-۱- تشریح نظرات موجود با مسئله

ماندن [۲] اشاره می‌کند که برای حداقل کردن هزینه توقف خط باید از دادن چند محصول با زمانهای عملیاتی نسبتا بزرگ به‌طور متوالی به خط پرهیز کرد و مناسب است که پس از هر محصول با زمان عملیاتی نسبتا بزرگ، یک محصول با زمان عملیاتی نسبتا کوچک قرار گیرد. در مدل در نظر گرفته شده از نظر ماندن مبنی بر جلوگیری از وقوع توالی محصولات با زمانهای عملیاتی نسبتا بزرگ استفاده می‌شود.

یاماشینا و اوکومورا [۳] نیز مدلی را برای حداقل کردن هزینه توقف خط ارائه می‌کنند و روش حل فراابتکاری آنها در جهت کاهش حداکثر نقطه اتمام عملیات محصولات یا کاهش حداکثر نقطه شروع عملیات در ایستگاهها عمل می‌کند. مؤلفان روشهایی را برای یافتن ترتیبهای اولیه و تعداد آنها برای دستیابی به ترتیب بهینه با یک فاصله اطمینان مشخص و همچنین نحوه تعویض محصولات در ترتیب را نیز ارائه می‌دهند.

۳-۲- فرضیات مدل

فرضیات در نظر گرفته شده برای مدل عبارت‌اند از:

- خط انتقال به صورت پیوسته عمل می‌کند.
- خط دارای نرخ تغذیه ثابت است.
- قبلا تعادل خط مونتاژ صورت گرفته است.
- زمان آماده‌سازی صفر است و ترتیب محصولات، تاثیری در زمانهای عملیاتی ندارد.
- در هر ایستگاه، یک اپراتور یا یک گروه اپراتور که به‌طور همزمان روی یک محصول عمل می‌کنند وجود دارد.
- هزینه اپراتور و هزینه‌های خط مونتاژ در نظر گرفته نمی‌شوند.

۳-۳- تعریف علائم

علائم مورد استفاده در مدل عبارت‌اند از:

M= تعداد ایستگاهها

I= تعداد مدلهای محصول

i= شمارنده نوع مدل (i=1,2,…, I)

m= شمارنده ایستگاه (m=1,2,…, M)

d_i = تقاضای تولید مدل i

Q= کل تقاضای تولید $Q = \sum_{i=1}^{I} d_i$

k= شمارنده موقعیت یک مدل در توالی. k=1,2,…,Q

t_{im} = زمان عملیات یک واحد از مدل i در ایستگاه m

α'_m = ضریب تفاوت ایستگاهها در وقوع توقف خط

1	2	3	4	5	6	7	8	9	10
2	1	2	1	3	3	2	2	3	2

شکل ۱- ساختار کروموزوم

$$\alpha'_m = \frac{Max_i\{t_{im}\}}{L_m}$$

$\overline{T}_m$ = متوسط زمان عملیاتی یک واحد در ایستگاه m

$$\overline{T}_m = \frac{\sum d_i t_{im}}{Q}$$

L_m = طول ایستگاه m

X_{ik} = ۱؛ اگر یک واحد از مدل i به موقعیت ترتیب k تخصیص یابد و ۰ در غیر این صورت

۳-۴- مدل پیشنهادی

مدل بهینه‌سازی در نظر گرفته شده با هدف حداقل کردن هزینه توقف خط به صورت زیر خواهد بود:

$$\text{Min} \sum_{m=1}^{M} \alpha'_m \sum_{k=2}^{Q} \left| \sum_{i=1}^{I} t_{im}\left(X_{ik} + X_{i(k-1)}\right) - 2\overline{T}_m \right|$$

s.t.

$$\sum_k X_{ik} = d_i \quad i=1,2,\ldots,I \qquad (1)$$

$$\sum_i X_{ik} = 1 \quad k=1,2,\ldots,Q \qquad (2)$$

$$X_{ik} \in \{0,1\} \quad i=1,2,\ldots,I \;;\; k=1,2,\ldots,Q$$

محدودیت گروه (۱)، تولید تقاضای مدلها در افق برنامه‌ریزی و محدودیت گروه (۲)، تخصیص یافتن یک واحد محصول به یک موقعیت ترتیب را تضمین می‌کنند.

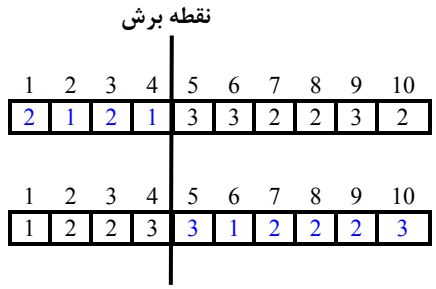
۴- طراحی الگوریتم ژنتیک

الگوریتم ژنتیک روشی است که در آن از انتخاب تصادفی کدها در فضایی مشخص به جواب قابل توجهی در مسایل رسیده می‌شود. الگوریتم ژنتیک یک روش آماری برای بهینه‌سازی و جستجوست که بخش محاسبات تکاملی است که خود زیر شاخه‌ای از هوش مصنوعی است. ویژگیهای خاص این الگوریتم باعث می‌شود که نتواند آن را یک جستجوگر تصادفی قلمداد کرد در واقع ایده اولیه این روش از نظریه تکاملی داروین الهام گرفته شده است و کارکرد آن بر اساس ژنتیک طبیعی استوار است. جواب مسئله‌ای که از طریق الگوریتم ژنتیک حل می‌شود، مرتبا بهبود می‌یابد. الگوریتم ژنتیک با مجموعه‌ای از جوابها که از طریق کروموزومها نشان داده می‌شوند شروع می‌شود. این مجموعه جواب جمعیت اولیه نام دارند. در این الگوریتم جوابهای حاصل از یک جمعیت برای تولید جمعیت بعدی استفاده می‌شوند. در این فرایند امید است که جمعیت جدید نسبت به جمعیت قبلی بهبود یابد. انتخاب بعضی از جوابها از میان کل جوابها (والدین[۱۳]) به منظور ایجاد جوابهای جدید یا همان فرزندان[۱۴] بر اساس مطلوبیت آنهاست. طبیعی است که جوابهای مناسبتر شانس بیشتری برای تولید مجدد داشته باشند. این فرایند تا ارضای شروط از پیش تعیین شده (مانند تعداد جمعیت یا میزان بهبود جواب) ادامه می‌یابد.

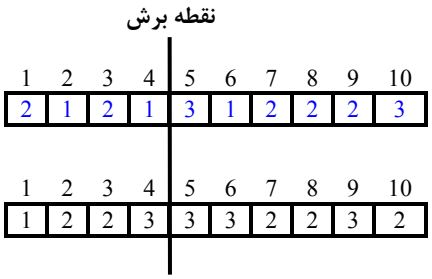
۱-۴- ساختار کروموزوم

کروموزوم بیانگر ساختار حل یا جواب شدنی مسئله مورد نظر در قالب ژنتیکی است. برای مسایلی که قبلا مدلسازی شده‌اند، متغیرهای تصمیم می توانند الگوی مناسبی برای طراحی کروموزوم باشند. با توجه به ماهیت متغیر تصمیم مدل ارائه شده؛ می‌توان کروموزوم را به صورت یک رشته خطی با Q ژن در نظر گرفت به طوری که هر ژن می تواند مقداری بین ۱ تا I اختیار کند. این نحوه ارائه؛ به طور خودکار محدودیت (۲) مدل را ارضاء خواهد کرد. برای ارضای محدودیت (۱) مدل، لازم است تعداد d_i ژن از کروموزوم؛ مقدار i را اختیار کنند. این مورد باید توسط یک سابروتین بعد از تولید هر کروموزوم کنترل شود.

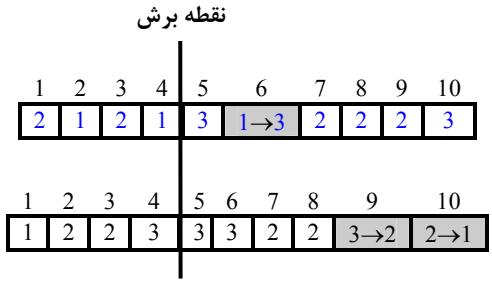
شکل (۱) ساختار کروموزوم برای یک مسئله نمونه با I=3 و Q=10 به طوری که d_1=2، d_2=5 و d_3=3 نشان می‌دهد.



شکل ۲ – دو حل شدنی به عنوان کروموزوم های والد



شکل ۳- دو فرزند حاصل از عملگر تقاطعی کلاسیک



شکل ۴- دو فرزند حاصل از عملگر تقاطعی اصلاح شده

۴-۲- عملگرهای ژنتیکی

برای جستجو و پیمایش فضای شدنی، نیاز به طراحی عملگرهای ژنتیکی مناسب با توجه به ساختار ارائه شده برای کروموزوم است. در این طراحی از سه عملگر تقاطعی اصلاح شده، جهش و معکوس استفاده شده است. همچنین از عملگر ترکیب مجدد[۱۵] نیز برای استفاده از خبره‌ترین حلهای نسل قبل استفاده شده است. در زیر به تفصیل نحوه عملکرد هر عملگر و اصلاحات آنها برای مسئله مورد بررسی آورده شده است.

۴-۲-۱- عملگر تقاطعی اصلاح شده

عملگر تقاطعی عبارت است از آمیزش دو عضو از جمعیت والد و تولید دو فرزند جدید؛ به طوری که هر فرزند نیمی از خصوصیات ژنتیکی و ساختاری والدین خود را در برداشته باشد. اعمال عملگر تقاطعی به صورت استاندارد به احتمال زیاد موجب تولید کروموزومهای نشدنی خواهد شد. چرا که رویکرد استاندارد به راحتی موجب نقض محدودیت (۱) مدل خواهد شد. در نتیجه این عملگر برای مسئله کاهش ریسک خط مورد تجدید نظر قرار گیرد. به عنوان نمونه؛ کروموزومهای شکل (۲) را به عنوان والد در نظر بگیرید؛ در حالت کلاسیک یک نقطه برش در طول کروموزوم به صورت تصادفی و یکنواخت انتخاب می‌کنیم (مثلا نقطه ۴). سپس بخشهای متناظر والدین را با یکدیگر تعویض می‌کنیم. بدین ترتیب فرزندان جدید به صورت شکل (۳) تولید می‌شوند. به وضوح مشاهده می‌شود که هر دو کروموزوم تولید شده در شکل (۳) نشدنی هستند زیرا هر دو کروموزوم محدودیت (۱) را نقض کرده‌اند. یافتن یک نقطه برش آرمانی که شدنی ماندن فرزندان جدید را تضمین کند؛ عملا ممکن نیست. برای غلبه بر این مشکل از یک رویکرد ترتیبی و جایگزینی استفاده شده است به طوری که حتی‌المقدور فرزندان تولید شده؛ بیشترین خصوصیات را از والدین خود به ارث ببرند.

در این روش، ابتدا همان عملگر تقاطعی کلاسیک را بر روی والدین اعمال می‌کنیم. سپس مقادیر ژنی را در کروموزومهای جدید از مکان برش از چپ به راست (یا برعکس) در صورت نیاز به گونه‌ای اصلاح می کنیم که محدودیت (۱) مدل ارضا شود. به عنوان نمونه، مجددا کروموزومهای شکل (۲) را به عنوان والد در نظر بگیرید؛ و فرض کنید نقطه برش همان مکان ۴ باشد. شکل (۴) فرزندان اصلاح شده را بعد از اعمال عملگر تقاطعی نشان می‌دهد. فرزندان جدید به غیر از چند ژن معدود تا حد امکان به والدین خود شبیه‌اند.

۴-۲-۲- عملگر جهش استاندارد

ساختار ارائه شده برای کروموزوم به گونه‌ای است که

1	2	3	4	5	6	7	8	9	10
2	1	2	1→2	3	3	2→1	2	3	2

شکل ۵- نحوه اعمال عملگر جهش کلاسیک

1	2	3	4	5	6	7	8	9	10
2	1	3	3	1	2	2	2	3	2

شکل ۶- فرزند جدید حاصل از عملگر معکوس

می‌توان از جهش استاندارد بخوبی استفاده کرد. برای این کار یک کروموزوم از جمعیت والد، مانند شکل (۲)، به تصادف انتخاب می‌کنیم؛ سپس دو مکان متمایز در بازه [1,Q] نیز به تصادف انتخاب کرده و مقادیر ژن این دو مکان را بر روی کروموزوم منتخب مانند شکل (۵) تعویض می کنیم.

۴-۲-۳- عملگر معکوس

عملگر معکوس حالت تعمیم یافته‌ای از عملگر جهش است که بر روی یک زیر رشته از کروموزوم منتخب اعمال شده و مقادیر ژن زیر رشته را معکوس می کند. برای این‌کار یک کروموزوم از جمعیت والد، مانند شکل (۲)، به تصادف انتخاب می کنیم؛ سپس دو مکان متمایز در بازه [1,Q] که معرف یک زیر رشته است، نیز به تصادف انتخاب کرده و مقادیر ژن بین این دو مکان را به روی کروموزوم منتخب تعویض می‌کنیم. در شکل (۶) زیر رشته بین مکانهای ۳ و ۶ معکوس شده است.

۴-۳- تولید جمعیت اولیه (نسل اول)

برای تولید نسل اول، با فرض ثابت بودن جمعیت در هر نسل، از یک تخصیص تصادفی ساده بر حسب توزیع یکنواخت گسسته استفاده شده است. بدین ترتیب که برای هر محصول i، تعداد d_i ژن به تصادف انتخاب و مقدار ژن آن برابر i تنظیم می‌شود. یک سابروتین خاص، عدم تشابه کروموزومهای تولید شده در هر نسل را بررسی می‌کند.

۴-۴- نحوه انتخاب جمعیت والد

برای انتخاب جمعیت والد از نسل فعلی؛ به منظور تولید نسل بعدی، از روش نرمالیزه کردن برازندگی کروموزومهای نسل فعلی استفاده می‌شود. بدین ترتیب که میانگین و واریانس برازندگی هر نسل محاسبه و مقدار هر برازندگی به صورت زیر نرمالیزه می‌شود.

$$z_i^g = \frac{f_i^g - \mu_g}{\delta_g} \in [-1,1]$$

$$\delta_g = \left(\frac{\sum_{i=1}^{K} \left(f_i^g - \mu_g \right)}{K} \right)^{\frac{1}{2}}$$

نسل g+1		
کروموزومها	مقدار برازندگی	برازندگی نرمالیزه
$x_1^{(g+1)}$	$f_1^{(g+1)}$	$Z_1^{(g+1)}$
$x_2^{(g+1)}$	$f_2^{(g+1)}$	$Z_2^{(g+1)}$
$x_K^{(g+1)}$	$f_K^{(g+1)}$	$z_K^{(g+1)}$
میانگین و واریانس	$\mu_{(g+1)}, \delta_{(g+1)}$	

نسل g		
کروموزومها	مقدار برازندگی	برازندگی نرمالیزه
x_1^g	f_1^g	z_1^g
x_2^g	f_2^g	z_2^g
x_K^g	f_K^g	z_K^g
میانگین و واریانس	μ_g, δ_g	

جمعیت والد، آن دسته از کروموزومهای نسل فعلی‌اند که $z_i^g \leq 0$ چرا که تابع هدف از نوع کمینه سازی است و هدف یافتن کروموزومها با برازندگی کمتر است. کروموزمهای نسل g+1 با استفاده از جمعیت والد و عملگرهای ژنتیکی تولید می‌شوند.

۴-۵- معیار توقف

برای توقف الگوریتم، از معیارهای زیر استفاده شده است:

الف) حداکثر تعداد نسلها.

ب) حداقل مقدار واریانس مجاز نسل.

ج) تساوی میانگین و واریانس دو نسل متوالی.

د) حداکثر زمان اجرای الگوریتم.

۴-۶- گام‌های الگوریتم ژنتیک

گام ۰- پارامترهای اولیه را به صورت زیر مقداردهی کنید:

جدول ۱- مسئله نمونه با توجه به زمان عملیات، تقاضای محصول و طول ایستگاه در بازه [۵-۱]

مسئله نمونه						
Station (ایستگاه)	۲	۳	۴	۵	۷	۹
Product (محصول)	۳	۴	۵	۶	۸	۱۰
مسئله نمونه						
Station (ایستگاه)	۳	۴	۵	۶	۸	۱۰
Product (محصول)	۲	۳	۴	۵	۷	۹

جدول ۲- مسئله نمونه با توجه به زمان عملیات و تقاضای محصول در بازه [۱۰-۱] و طول ایستگاه در بازه [۵-۱]

مسئله نمونه						
Station (ایستگاه)	۲	۳	۴	۵	۷	۹
Product (محصول)	۳	۴	۵	۶	۸	۱۰
مسئله نمونه						
Station (ایستگاه)	۳	۴	۵	۶	۸	۱۰
Product (محصول)	۲	۳	۴	۵	۷	۹

K : تعداد جمعیت در هر نسل.

G : حداکثر تعداد نسلهای مجاز.

δ : حداقل واریانس مجاز برای هر نسل.

r_i : نرخ انتخاب عملگر نوع i.

- شمارنده‌ها:

g : شمارنده نسل $g = 1,2,\ldots,G$.

گام ۱- قرار دهید $g = 1$ و جمعیت اولیه را طبق بند ۴-۳ تولید کنید.

گام ۲- جمعیت والد را طبق بند ۴-۴ تولید کنید.

گام ۳- با استفاده از کروموزومهای جمعیت والد و انتخاب تصادفی عملگرها با نرخ معین، تعداد K فرزند تولید و در نسل جدید قرار دهید. $G = g + 1$ را قراردهید و میانگین و واریانس نسل جدید را محاسبه کنید. اگر یکی از معیارهای بند ۴-۵ ارضا شده است توقف کرده و بهترین جواب نسل آخر را به خروجی بفرست در غیر این صورت به گام ۲ بروید.

۵- نتایج محاسباتی

۵-۱- طراحی مسائل نمونه

در این مقاله، برای نشان دادن کارایی الگوریتم ژنتیک در مسئله حداقل‌کردن توقفات خط باید مسایلی طراحی شود تا توانایی و نقاط ضعف این الگوریتم را نشان دهد و همچنین برای سنجش کارایی الگوریتم ژنتیک، با استفاده از مقادیر مختلف، مسئله مورد بررسی قرار می‌گیرد. جداول (۱) و (۲) مسایل طراحی شده را نشان می‌دهد.

۵-۲- حل مسئله و تحلیل نتایج

در این قسمت، نتایج حل مسئله با استفاده از الگوریتم ژنتیک برای حداقل‌کردن توقفات خط ارایه می‌شود و برای آزمودن صحت جواب به دست آمده از الگوریتم، از نرم‌افزار Lingo استفاده شده است. سپس در جدولی زمان اجرا توسط نرم‌افزار و الگوریتم ژنتیک، مقدار تابع هدف به دست‌آمده از نرم‌افزار و مقدار اختلاف بهترین تابع هدف به دست آمده از نرم‌افزار و الگوریتم ژنتیک ارایه می‌شود، جداول (۳) و (۴) و نمودارهای (۱) و (۲). سپس با توجه به جداول ارایه شده، میانگین زمان اجرا و مقدار خطای الگوریتم ژنتیک با نرم‌افزار مقایسه می‌شود. در نتیجه می‌توان گفت که الگوریتم ژنتیک در زمانی قابل قبول جوابی مناسب با مقدار خطای کم را نتیجه می‌دهد. بنابراین الگوریتم پیشنهادی مناسب وکارایی آن بالا ارزیابی می‌شود.

با توجه به اطلاعات هر مسئله و جوابهای به دست آمده از الگوریتم ژنتیک می‌توان تعداد کل عملگرهای استفاده شده در الگوریتم ژنتیک را مشاهده کرد، جداول (۵) و (۶) و نمودارهای (۳) و (۴).

۵-۳- تحلیل و بررسی جواب مسئله

چندین بار مسئله با توجه به پارامتر تعریف شده اجرا شده است که نتایج آنها به صورت جداولی ارائه شد. حال به مقایسه

جدول۳- نتایج حل مسئله با توجه به زمان عملیات، تقاضای محصول و طول ایستگاه در بازه [۵-۱]

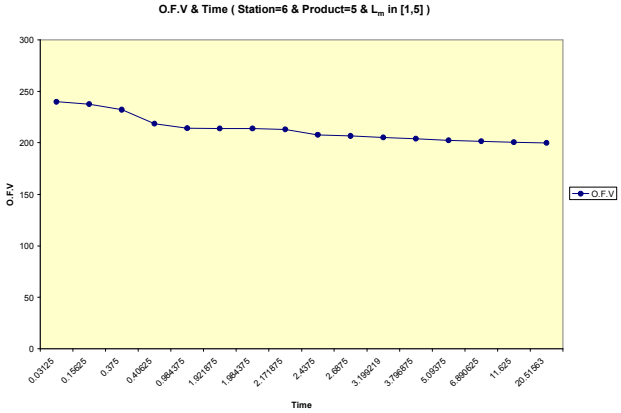
ردیف	مسئله نمونه		اطلاعات مسئله		حل با Lingo		حل با GA		Gap of OFV	T_{AVG}
	Station	Product	$\sum Q$	$\sum L_m$	OFV	CPU Time (s)	OFV	CPU Time (s)		
۱	۲	۳	۹	۵	۹/۴۴۴۴	۰	۹/۴۴۴۴	۰/۲۸۱۲	۰	۰/۱۴۰۶
۲	۳	۴	۱۵	۱۰	۴۹/۳۰۶۷	۳۱۱	۴۹/۳۰۶۷	۳/۷۳۴۴	۰	۱۵۷/۳۶۷۲
۳	۴	۵	۱۴	۱۱	-----	۳۷۱۲	۹۹/۵۷۱۴	۱۶/۳۱۲	-----	۱۵۶/۱۸۶۴
۴	۵	۶	۲۳	۱۵	-----	۳۶۹۶●	۱۸۳/۰۶۰۹	۷۵/۹۶۸۷	-----	۲۰۲۲/۴۸۴۳
۵	۷	۸	۲۲	۲۰	-----	۴۲۵۰●	۲۷۸/۱۹۶۹	۴۲/۴۳۷۰	-----	۲۱۴۶/۲۱۸۵
۶	۹	۱۰	۲۵	۲۶	-----	۴۳۲۰●	۳۲۲/۳۰۹۹	۲۲۲/۳۷۵۰	-----	۲۲۷۱/۱۸۷۵
۷	۳	۲	۵	۹	۱۷/۵۵	۱	۱۷/۵۵۰۰	۰	۰	۰/۵
۸	۴	۳	۱۰	۱۵	۴۲/۲۴	۱	۴۲/۲۳۹۹	۰/۴۳۳۶	۰/۰۰۰۱	۰/۷۱۶۸
۹	۵	۴	۱۱	۱۴	۶۷/۳۷۱۲	۳۱	۶۷/۳۷۱۲	۰	۰	۱۵/۵
۱۰	۶	۵	۲۱	۱۷	-----	۳۶۹۰●	۱۹۹/۸۹۵۲	۲۰/۵۱۵۶	-----	۱۸۵۵/۲۵۷۸
۱۱	۸	۷	۲۰	۲۲	-----	۳۹۰۳●	۲۸۳/۱۴۹۹	۴۱/۵۷۹۰	-----	۱۹۷۲/۲۸۹۵
۱۲	۱۰	۹	۲۲	۲۹	-----	۴۱۰۹●	۳۶۴/۶۸۱۸	۸۶/۴۸۳۹	-----	۲۰۹۷/۷۴۱۹

● تا این زمان جواب بهینه‌ای برای مسئله توسط نرم‌افزار لینگو به دست نیامده است.

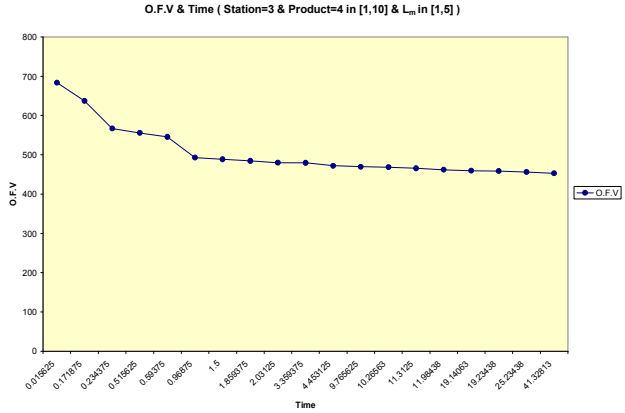
جدول۴- نتایج حل مسئله با توجه به زمان عملیات، تقاضای محصول در بازه [۱۰-۱] و طول ایستگاه در بازه [۵-۱]

ردیف	مسئله نمونه		اطلاعات مسئله		حل با Lingo		حل با GA		Gap of OFV	T_{AVG}
	Station	Product	$\sum Q$	$\sum L_m$	OFV	CPU Time (s)	OFV	CPU Time (s)		
۱	۲	۳	۱۷	۵	۳۴/۷۵	۱	۳۴/۷۵	۰/۷۹۶۹	۰	۰/۸۹۸۴
۲	۳	۴	۳۱	۱۰	-----	۳۷۷۲●	۴۵۲/۷۱۲۹	۴۱/۳۲۸۱	-----	۱۹۰۶/۶۶۴۰
۳	۴	۵	۲۶	۱۱	-----	۳۷۵۵●	۷۸۵/۷۲۴۴	۱۸/۰۳۱۲	-----	۱۸۸۶/۵۱۵۶
۴	۵	۶	۴۴	۱۵	-----	۳۹۶۰●	۱۴۲۳/۵۷۵۸	۷۶/۶۸۷۱	-----	۲۰۱۸/۳۴۳۵
۵	۷	۸	۳۶	۲۰	-----	۴۲۹۶●	۲۰۵۱/۹۴۴۵	۱۱۹/۷۳۵۱	-----	۲۲۰۷/۸۶۷۵
۶	۹	۱۰	۴۵	۲۶	-----	۴۴۵۲●	۲۳۷۶/۴۷۹۱	۵۵۹/۳۷۵	-----	۲۵۰۵/۶۸۷۵
۷	۳	۲	۹	۹	۸۵/۷۵	۰	۸۵/۷۵۰۰	۰	۰	۰
۸	۴	۳	۲۱	۱۵	۳۵۷/۵۸۱۰	۷۸۴	۳۵۷/۵۸۰۹	۲/۴۵۳۱	۰,۰۰۰۱	۳۹۳/۲۲۶۵
۹	۵	۴	۲۱	۱۴	-----	۳۹۰۹●	۵۸۳/۱۹۰۵	۷/۸۹۰۶	-----	۱۹۵۸/۴۴۵۳
۱۰	۶	۵	۴۱	۱۷	-----	۴۰۴۵●	۱۶۱۳/۱۵۱۲	۸۲/۲۹۶۹	-----	۲۰۶۳/۶۴۸۴
۱۱	۸	۷	۳۲	۲۲	-----	۴۳۴۹●	۲۱۰۰/۲۶۵۶	۸۱/۰۴۷۱	-----	۲۲۱۵/۰۲۳۵
۱۲	۱۰	۹	۴۰	۲۹	-----	۴۴۶۷●	۳۰۰۶/۳۵۱۶	۳۰۶/۹۳۸	-----	۲۳۸۶/۹۶۹۰

●تا این زمان جواب بهینه‌ای برای مسئله توسط نرم‌افزار لینگو به دست نیامده است.



نمودار ۱- حل مسئله نمونه توسط GA با توجه به زمان عملیات، تقاضای محصول و طول ایستگاه در بازه [۵-۱]



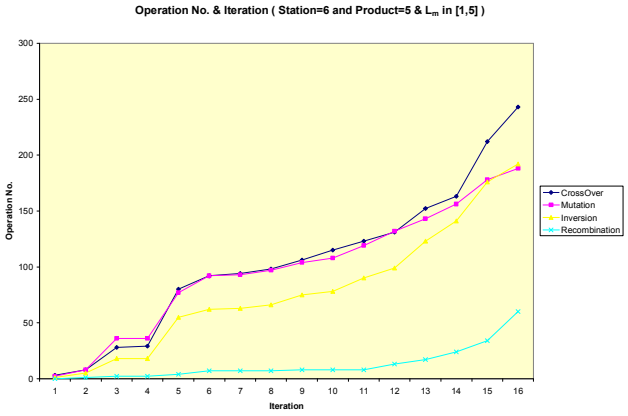
نمودار۲- حل مسئله نمونه توسط GA با توجه به زمان عملیات، تقاضای محصول در بازه [۱۰-۱] و طول ایستگاه در بازه [۵-۱]

جدول۵- نتایج حل مسئله با توجه به زمان عملیات، تقاضای محصول و طول ایستگاه در بازه [۵-۱]

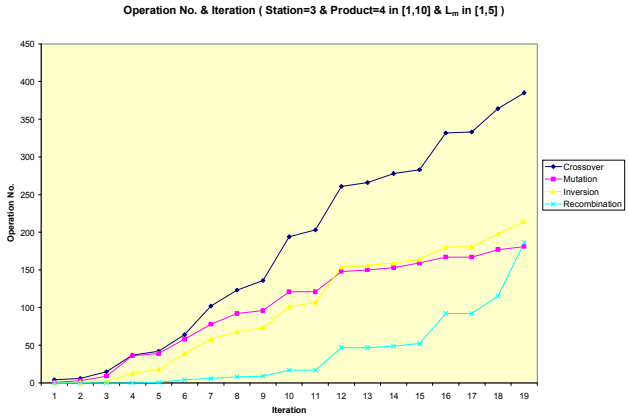
ردیف	مسئله نمونه		اطلاعات مسئله		تعداد عملگر			
	Station	Product	$\sum Q$	$\sum L_m$	Crossover	Mutation	Inversion	Recombination
۱	۲	۳	۹	۵	۲۸	۱۵	۲۲	۴
۲	۳	۴	۱۵	۱۰	۱۶۰	۹۳	۸۹	۱۹
۳	۴	۵	۱۴	۱۱	۱۹۸	۱۴۱	۱۶۰	۱۰۱
۴	۵	۶	۲۳	۱۵	۴۷۱	۲۶۵	۳۰۹	۲۸۶
۵	۷	۸	۲۲	۲۰	۳۵۳	۳۵۰	۳۶۴	۱۰۶
۶	۹	۱۰	۲۵	۲۶	۵۴۰	۳۹۵	۶۱۸	۴۳۷
۷	۳	۲	۵	۹	-----	-----	-----	-----
۸	۴	۳	۱۰	۱۵	۶۱	۳۷	۲۴	۳
۹	۵	۴	۱۱	۱۴	-----	-----	-----	-----
۱۰	۶	۵	۲۱	۱۷	۲۴۳	۱۸۸	۱۹۲	۶۰
۱۱	۸	۷	۲۰	۲۲	۳۱۵	۱۹۲	۲۱۹	۱۴۲
۱۲	۱۰	۹	۲۲	۲۹	۴۱۷	۲۴۴	۳۳۰	۱۹۷

جدول۶- نتایج حل مسئله با توجه به زمان عملیات، تقاضای محصول در بازه [۱۰-۱] و طول ایستگاه در بازه [۵-۱]

ردیف	مسئله نمونه		اطلاعات مسئله		تعداد عملگر			
	Station	Product	$\sum Q$	$\sum L_m$	Crossover	Mutation	Inversion	Recombination
۱	۲	۳	۱۷	۵	۳۱	۳۷	۲۸	۸
۲	۳	۴	۳۱	۱۰	۳۸۵	۱۸۱	۲۱۴	۱۸۶
۳	۴	۵	۲۶	۱۱	۳۲۲	۲۲۳	۱۸۳	۷۳
۴	۵	۶	۴۴	۱۵	۵۳۰	۴۰۱	۳۹۹	۱۷۶
۵	۷	۸	۳۶	۲۰	۷۶۴	۴۱۰	۴۵۷	۲۵۰
۶	۹	۱۰	۴۵	۲۶	۱۰۴۱	۵۵۴	۷۵۷	۷۴۶
۷	۳	۲	۹	۹	-----	-----	-----	-----
۸	۴	۳	۲۱	۱۵	۱۲۷	۸۱	۷۱	۱۲
۹	۵	۴	۲۱	۱۴	۲۰۷	۱۴۰	۱۲۳	۳۷
۱۰	۶	۵	۴۱	۱۷	۴۴۶	۳۰۱	۳۵۲	۱۹۰
۱۱	۸	۷	۳۲	۲۲	۵۱۳	۴۰۹	۳۷۷	۱۷۸
۱۲	۱۰	۹	۴۰	۲۹	۵۴۶	۴۸۷	۶۳۷	۴۰۸



نمودار ۳- تعداد عملگرهای استفاده شده درمسئله نمونه با توجه به زمان عملیات، تقاضای محصول و طول ایستگاه در بازه [۵-۱]



نمودار۴- تعداد عملگرهای استفاده شده درمسئله نمونه با توجه به زمان عملیات، تقاضای محصول در بازه [۱۰-۱] و طول ایستگاه در بازه [۵-۱]

زمان حل دو روش با توجه به زمان عملیات، تقاضای محصول و طول ایستگاه می‌پردازیم، جداول (۷) و (۸) و نمودارهای (۵) و (۶).

همان طور که مشاهده می‌شود در صورتی که زمان عملیات، تقاضای محصول و طول ایستگاه در بازه [۵-۱] قرار گیرد و مسئله $\text{station} = ۳ \text{ ,product} = ۴$ باشد، زمان لینگو برابر ۳۱۱ ثانیه و زمان الگوریتم ژنتیک برابر ۳/۷۳۴۴ ثانیه و یا در حالت $\text{station} = ۴, \text{product} = ۵$، زمان لینگو ۳۷۱۲ ثانیه و زمان الگوریتم ژنتیک ۱۶/۳۱۲ ثانیه است. همچنین در صورتی که زمان عملیات، تقاضای محصول در بازه [۱۰-۱] و طول ایستگاه در بازه [۵-۱] قرار گیرد و مسئله $\text{station} = ۲, \text{product} = ۳$ باشد، زمان لینگو برابر ۱ ثانیه و زمان الگوریتم ژنتیک برابر ۰/۷۹۶۹ ثانیه و یا در حالت $\text{station} = ۴, \text{product} = ۳$، زمان لینگو ۷۸۴ ثانیه و زمان الگوریتم ژنتیک ۲/۴۵۳۱ ثانیه است.

با توجه به مقایسه صورت گرفته می‌توان نتیجه‌گیری کرد که با افزایش حجم مسئله و بیشتر شدن تعداد ایستگاهها و محصولات، توانایی نرم‌افزار لینگو کاهش می‌یابد و در زمان طولانی به جواب بهینه دست می‌یابد و چنانچه حجم مسئله گسترده‌تر شود، توانایی نرم‌افزار در انجام محاسبات میسر نمی‌شود، حال آنکه با افزایش حجم مسئله و بیشتر شدن تعداد ایستگاههای کاری و متنوع شدن مدلها، روش ابتکاری پیشنهادی سریعتر به جواب می‌رسد و این خود گویای برتری الگوریتم و کاراتر بودن آن در زمانی کوتاه نسبت به نرم‌افزار لینگوست.

۶- نتیجه‌گیری و پیشنهادها

استفاده از مدل پیشنهادی، برای مسایل بزرگ با استفاده از نرم‌فزار لینگو، به دلیل مشکلات محاسباتی و زمان زیادی که برای حل نیاز دارد، مناسب نیست. از این رو یک روش حل فراابتکاری برای حل مسئله در یک زمان قابل قبول پیشنهاد شد و رویه حل آن توسط الگوریتم ژنتیک صورت گرفت، سپس برای ارزیابی الگوریتم، اطلاعات ورودی به الگوریتم داده شد و جوابها مورد بررسی قرار گرفت. همچنین اطلاعات ورودی، به نرم‌افزار لینگو نیز داده شد و جوابهای به دست آمده مورد تجزیه و تحلیل و مقایسه با جوابهای حاصل از الگوریتم، قرار گرفت. بررسی نتایج حاکی از آن است که با افزایش ایستگاه و محصول، زمان پردازش (CPU) در الگوریتم ژنتیک به صورت قابل توجه، نسبت به نرم‌افزار لینگو کمتر است، که این عملکرد، قابل قبول بودن الگوریتم ژنتیک پیشنهادی را نسبت به نرم‌افزار لینگو نشان می‌دهد. نتایج به دست آمده را می‌توان به شرح زیر خلاصه کرد:

جدول۷- مقایسه زمان حل دو روش با توجه به زمان عملیات تقاضای محصول و طول ایستگاه در بازه [۵-۱]

ردیف	مسئله نمونه		اطلاعات مسئله		حل با Lingo	حل با GA
	Station	Product	$\sum Q$	$\sum L_m$	CPU Time (s)	CPU Time (s)
۱	۲	۳	۹	۵	۰	۰/۲۸۱۲
۲	۳	۴	۱۵	۱۰	۳۱۱	۳/۷۳۴۴
۳	۴	۵	۱۴	۱۱	۳۷۱۲	۱۶/۳۱۲
۴	۵	۶	۲۳	۱۵	۳۹۶۹●	۷۵/۹۶۸۷
۵	۷	۸	۲۲	۲۰	۴۲۵۰●	۴۲/۴۳۷۰
۶	۹	۱۰	۲۵	۲۶	۴۳۲۰●	۲۲۲/۳۷۵۰
۷	۳	۲	۵	۹	۱	۰
۸	۴	۳	۱۰	۱۵	۱	۰/۴۳۳۶
۹	۵	۴	۱۱	۱۴	۳۱	۰
۱۰	۶	۵	۲۱	۱۷	۳۶۹۰●	۲۰/۵۱۵۶
۱۱	۸	۷	۲۰	۲۲	۳۹۰۳●	۴۱/۵۷۹۰
۱۲	۱۰	۹	۲۲	۲۹	۴۱۰۹●	۸۶/۴۸۳۹

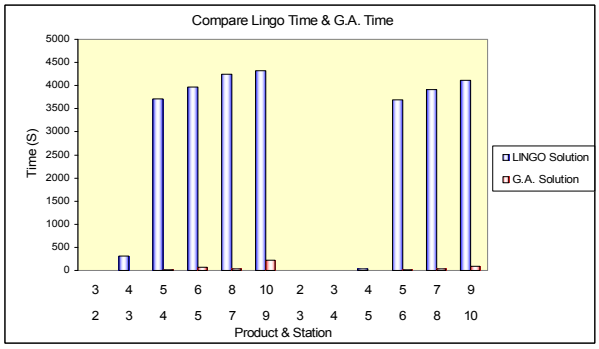
●تا این زمان جواب بهینه‌ای برای مسئله توسط نرم‌افزار لینگو به دست نیامده است

جدول ۸- مقایسه زمان حل دو روش با توجه به زمان عملیات تقاضای محصول در بازه [۱۰-۱] و طول ایستگاه در بازه [۵-۱]

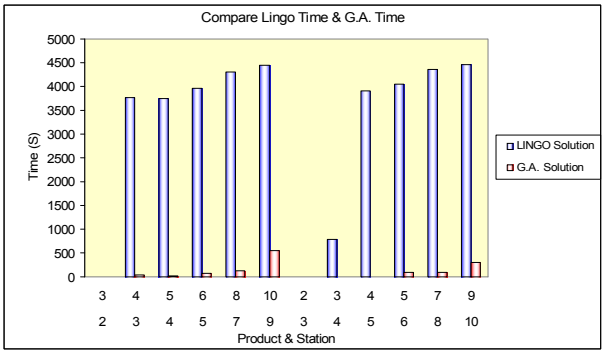
ردیف	مسئله نمونه		اطلاعات مسئله	حل با Lingo		حل با GA
	Station	Product	$\sum Q$	$\sum L_m$	CPU Time (s)	CPU Time (s)
۱	۲	۳	۱۷	۵	۱	۰/۷۹۶۹
۲	۳	۴	۳۱	۱۰	۳۷۷۲●	۴۱/۳۲۸۱
۳	۴	۵	۲۶	۱۱	۳۷۵۵●	۱۸/۰۳۱۲
۴	۵	۶	۴۴	۱۵	۳۹۶۰●	۷۶/۶۸۷۱
۵	۷	۸	۳۶	۲۰	۴۲۹۶●	۱۱۹/۷۳۵۱
۶	۹	۱۰	۴۵	۲۶	۴۴۵۲●	۵۵۹/۳۷۵
۷	۳	۲	۹	۹	۰	۰
۸	۴	۳	۲۱	۱۵	۷۸۴	۲/۴۵۳۱
۹	۵	۴	۲۱	۱۴	۳۹۰۹●	۷/۸۹۰۶
۱۰	۶	۵	۴۱	۱۷	۴۰۴۵●	۸۲/۲۹۶۹
۱۱	۸	۷	۳۲	۲۲	۴۳۴۹●	۸۱/۰۴۷۱
۱۲	۱۰	۹	۴۰	۲۹	۴۴۶۷●	۳۰۶/۹۳۸۰

● : تا این زمان جواب بهینه‌ای برای مسئله توسط نرم‌افزار لینگو به دست نیامده است

- نتایج حاصل نشان می‌دهد که الگوریتم ژنتیک در این مسئله (با توجه به فرضیات آن) عملکرد قابل قبولی داشته و زمانهای رسیدن به حل بهینه یا نزدیک بهینه بسیار کم است.
- با توسعه و گسترش مسایل، زمان محاسباتی توسط لینگو افزایش می‌یابد در حالی‌که این زمان در حالت استفاده از الگوریتم ژنتیک کم و تقریبا افزایش نامحسوسی در مقابل زمان محاسباتی توسط لینگو دارد.
- با توجه به افزایش تنوع محصولات و حرکت صنایع به سمت تولید بر اساس سفارش مشتری، در آینده با افزایش تعداد مدلهای مختلف روبرو خواهیم بود، همچنین افزایش



نمودار ۵- مقایسه زمان حل دو روش با توجه به زمان عملیات، تقاضای محصول و طول ایستگاه در بازه [۵-۱]



نمودار ۶- میله‌ای مقایسه زمان حل دو روش با توجه به زمان عملیات و تقاضای محصول در بازه [۱۰-۱] و طول ایستگاه در بازه [۵-۱]

ظرفیت تولید از اهداف اصلی صنایع مونتاژ می‌باشد، لذا استفاده از روشهای فراابتکاری برای تعیین توالی محصولات نسبت به روشهای معمولی بیشتر مورد توجه خواهد بود.

برای گسترش این تحقیق می‌توان پیشنهادهایی را بیان کرد. با توجه به اینکه مدل ارائه شده در این تحقیق با هدف حداقل‌کردن توقفات خط مونتاژ با دو روش حل مقایسه شده، لذا این موضوع را از جهات مختلف می‌توان توسعه داد؛ که بخشی از این موضوعات عبارت‌اند از:

طرح مسایل در شرایط احتمالی و پویا ، تغییرات در فرضیات مسئله، استفاده از عملگرهای مختلف به منظور افزایش امکان جستجوی بیشتر الگوریتم و هموار کردن همگرایی، اعمال محدودیتهای جدید به مسئله با در نظرگرفتن محدودیتهای موجود، به کار بردن روشهای ابتکاری دیگر مانند TS، SA، شبکه‌های عصبی، سیستمهای فازی و یا تلفیقی از این روشها.

واژه‌نامه

1. operator
2. material handling system
3. just-in-time
4. simulated annealing
5. setups
6. work (or job) utility
7. memetic algorithm
8. multi-objective particle swarm
9. multi-objective scatter search
10. total enumeration
11. beam search
12. goal chasing
13. parents
14. off spring
15. recombination

مراجع

1. Grover, M.P., *Automation, Production System, and Computer Integrated Manufacturing*, 2nd Ed., Prentice Hall, 2001.
2. Monden, Y., *Toyota Production System , an Integrated Approach to Just in Time*, 2nd Ed., Georgia, Institute of Industrial Engineers, 1993.
3. Okamura, K., and Yamashina, H., "A Heuristic Algorithm for the Assembly Line Model Mix Sequencing Problem to Minimize the Cost of Stopping the Conveyor," *International of Production Research*, Vol. 17, No. 3, PP. 233-247, 1979.
4. Bard, J.F., Dar–El, E., and Shtub, A., "An Analytic Framework for Sequencing Mixed Model Assembly Lines," *International of Production Research*, Vol. 13, No. 1, PP. 35-48, 1992.
5. Thomopoulos, N.T., "Line Balancing Sequencing for Mixed Model Assembly," *Management Science*, Vol. 14, No. 2, PP. B 59 -75, 1967.
6. Dar–El, E.M., and Cother, R.F., "Assembly Line Sequencing for Model Mix," *International of production Research*", Vol. 13, No. 5, PP. 463-477, 1975.
7. Xiabo, Z., and Ohno, K., "Algorithms for Sequencing Mixed Models on an Assembly Line in a JIT Production System," *Computers and Industrial Engineering*, Vol. 32, No. 1, PP. 47-56, 1997.
8. Xiabo, Z., and Ohno, K., "A Sequencing Problem for a Mixed Model Assembly Line in a Jit Production System," *Computers and Industrial Engineering*, Vol. 27, No. 1-4, PP. 71-74, 1994.
9. Bolat, A., Savsar, M., and Al-Fawzan, M.A., "Algorithms for Real Time Scheduling of Jobs on

Mixed Model Assembly Lines," *Computers and Operations Research*, Vol. 21, No. 5, PP. 487-498, 1994.

10. Tavakkoli-Moghaddam, R., and Rahimi-Vahed, A.R., "Multi-Criteria Sequencing Problem for a Mixed-Model Assembly Line in a JIT Production System," *Applied Mathematics and Computation*, Vol. 181, PP. 1471-1481, 2006.
11. Rahimi-Vahed, A.R., Mirghorbani, S.M., and Rabbani, M., "A New Particle Swarm Algorithm for a Multi-Objective Mixed-Model Assembly Line Sequencing Problem," *Soft Computing*, Vol. 11, PP. 997-1012, 2007.
12. Rahimi-Vahed, A.R., Rabbani, M., Tavakkoli-Moghaddam, R., Torabi, S.A., and Jolai, F., "A Multi-Objective Scatter Search for a Mixed-Model Assembly Line Sequencing Problem," *Advanced Engineering Information*, Vol. 21, PP. 85-99, 2007.
13. Yu, J., Yin, Y,. and Chen, Z., "Scheduling of an Assembly Line with a Multi-Objective Genetic Algorithm," *Int. J. Adv. Manufacturing Technology*, Vol. 28, PP. 551-555, 2006.
14. Simaria, A.S., and Vilarinho, P.M., "A Genetic Algorithm Based Approach to the Mixed-Model Assembly Line Balancing Problem of Type II," *Computers & Industrial Engineering*, Vol. 47, PP. 391-407, 2004.
15. Mansouri, S.A., "A Multi-Objective Genetic Algorithm for Mixed-Model Sequencing on JIT Assembly Lines," European Journal of Operational Research, Vol. 167, PP. 696-716, 2005.
16. Ponnambalam, S.G., Aravindan, P., and Rao M.S., "Genetic Algorithms for Sequencing Problems in Mixed Model Assembly Lines," *Computers & Industrial Engineering*, Vol. 45, PP. 669-690, 2003.
17. Erel, E., Gocgun, Y., and Sabuncuoglu, I., "Mixed-Model Assembly Line Sequencing Using Beam Search," *Article in Press, International Journal of Production Research*, DOI: 10.1080/00207540600806497, 2006.
18. Jin, M., and Wu, S.D., "A New Heuristic Method for Mixed Model Assembly Line Balancing Problem," *Computers & Industrial Engineering*, Vol. 44, PP. 159-169, 2002.
19. Bukchin, Y., and Rabinowitch, I., "A Branch-and-Bound Based Solution Approach for the Mixed-Model Assembly Line-Balancing Problem for Minimizing Stations and Task Duplication Costs," *European Journal of Operational Research*, Vol. 174, PP. 492-508, 2006.
20. Noorul Haq, A., Jayaprakash, J., and Rengarajan, K., "A Hybrid Genetic Algorithm Approach to Mixed-Model Assembly Line Balancing," *Int. Adv. Manufacturing Technology*, Vol. 28, PP. 337-341, 2006.