



MULTIPLE ROUTE JOB SHOP SCHEDULING USING PARTICLE SWARM OPTIMIZATION APPROACH

Hamid Reza Golmakani* & Ali Reza Birjandi

Hamid Reza Golmakani, Industrial Engineering Department, Tafresh University

Ali Reza Birjandi, Industrial Engineering Department, Tafresh University

Keywords

Multiple-route,
Job Shop Scheduling,
Particle Swam,
Optimization Algorithm
Make span,

ABSTRACT

The Job shop Scheduling Problem (JSP) is classified in the group of NP-hard problems. The Multiple Route JSP (MRJSP) is an extension to JSP where each part may have more than one route for its production and the numbers of operations in routes are not necessarily equal. The solution to this problem has become even more complicated due to the existence of multiple alternative routes for parts and, thus, meta-heuristic algorithms are the only approaches that can efficiently obtain good slotions within a reasonable computational time. In this paper, mathematical formulation of the MRJSP is first presented. Then, a meta-heuristic approach based on Particle Swam Optimization (PSO) algorithm is proposed for the minimization of makes pan. In order to evaluate the effectiveness of the proposed approach, 30 problems in small, medium, and large size, which have already been used by other researchers, are adopted and re-solved using the proposed approach. Problems are also formulated and solved using Lingo software and the results are compared. The computational results show that the proposed approach outperforms the approach based on the immune algorithm, already reported by other researchers, and can effectively generate high-quality schedules in a timely fashion.

©2015 IUST Publication, IJIEPM. Vol. 26, No. 2, All Rights Reserved



زمانبندی کارِ کارگاهی چند مسیره با استفاده از الگوریتم بهینه سازی گروه ذرات

حمیدرضا گل‌مکانی* و علیرضا بیرجندی

چکیده:

مسئله زمانبندی کارِ کارگاهی، در گروه مسائل NP-hard طبقه بندی می‌شود. وجود چند مسیر پردازش برای هر کار نیز موجب پیچیده‌تر شدن این مسئله شده و لذا استفاده از الگوریتم‌های فراابتکاری، به عنوان روشی موثر در حل اینگونه مسائل، مورد توجه بسیاری از محققین بوده است. در این مقاله، ابتدا مدل برنامه‌ریزی غیرخطی عدد صحیح مختلط از مسئله زمانبندی کارِ کارگاهی چند مسیره ارائه خواهد شد. سپس یک الگوریتم فراابتکاری، بر مبنای الگوریتم بهینه سازی گروه ذرات، با هدف کمینه سازی زمان اتمام کارها، برای حل مسئله مذکور ارائه می‌گردد. همچنین جهت ارزیابی عملکرد الگوریتم پیشنهادی، ۳۰ مسئله در ابعاد کوچک، متوسط و بزرگ، که قبلاً با استفاده از الگوریتم ایمنی مصنوعی توسط دیگر محققین حل و نتایج آن گزارش شده است، با الگوریتم پیشنهادی حل و نتایج با یکدیگر مقایسه شده‌اند. از نقطه نظر زمان مورد نیاز برای تولید جواب و مقدار بدست آمده برای تابع هدف، نتایج حاکی از عملکرد مطلوب الگوریتم پیشنهادی است.

کلمات کلیدی

زمانبندی کارِ کارگاهی چند مسیره، الگوریتم بهینه سازی گروه ذرات، کمینه سازی زمان اتمام کارها

۱. مقدمه

زمانبندی عبارت از تخصیص منابع، در طول دوره‌ای از زمان، برای انجام مجموعه‌ای از کارها با در نظر گرفتن محدودیت‌ها، نیازمندی‌ها و اهداف زمانبندی است [۱]. زمانبندی به عنوان یک پروسه تصمیم‌گیری، نقش مهمی را در اغلب صنایع تولیدی و خدماتی، برای استفاده مؤثرتر از منابع موجود، ایفا می‌کند. صنایع کارِ کارگاهی، از صنایع تولیدی شناخته شده و رایجی هستند که برای تولید انواع متنوعی از محصولات، با حجم اندک، به کار می‌روند [۲]. تولید ویفر در صنایع تولید نیمه رساناها، صنایع خودروسازی، دارویی، شیمیایی، صنایع تولید فولاد و بسیاری از صنایع تولیدی دیگر، نمونه‌هایی از صنایع کارِ کارگاهی به شمار می‌آیند.

حل ریاضی و قطعی مسئله‌ی زمانبندی کارِ کارگاهی، حتی در ابعاد کوچک، بعضاً بسیار پیچیده و زمان‌بر است. به عبارت دیگر، با توجه به تعداد زیاد متغیرها و محدودیت‌ها در مدل ریاضی، حل مدل ریاضی و یافتن جواب بهینه در زمان متعارف میسر نیست [۳]. از این رو برای حل چنین مسائلی از الگوریتم‌های ابتکاری و فراابتکاری مختلفی استفاده شده است که در ادامه به مواردی از آن‌ها اشاره می‌شود. الگوریتم ژنتیک، با الهام از شبیه سازی تکامل طبیعی و به عنوان یک پروسه‌ی بهینه‌یابی، بطور وسیعی مورد استفاده محققین در حوزه‌های مختلف قرار گرفته است. از الگوریتم ژنتیک برای حل مسائل زمانبندی کارِ کارگاهی نیز استفاده‌های زیادی گزارش شده است. به عنوان مثال در [۴] از الگوریتم ژنتیک، برای حل مسئله‌ی زمانبندی کارِ کارگاهی و با هدف کمینه سازی زمان اتمام کارها، استفاده شده است. در [۵] نیز با ترکیب الگوریتم ژنتیک و روشی برای جستجوی محلی به منظور بهبود جواب‌های نهایی، مسئله کارِ کارگاهی مورد بررسی قرار گرفته است. مبنای جستجوی محلی، تعویض‌های (جابجایی‌های) دوتایی بین عملیات‌هایی است که در مسیر بحرانی (طولانی‌ترین مسیری که از توالی پی در پی عملیات‌ها روی ماشین‌های مختلف، تا زمان اتمام همه کارها، حاصل می‌شود) قرار دارند. همچنین در [۶] به منظور

تاریخ وصول: ۹۱/۱/۳۱

تاریخ تصویب: ۹۲/۷/۲

علیرضا بیرجندی: دانش آموخته دانشکده مهندسی صنایع، دانشگاه تفرش،
*نویسنده مسئول مقاله: دکتر حمیدرضا گل‌مکانی: دانشیار، دانشکده مهندسی صنایع، دانشگاه تفرش، golmakni@mie.utoronto.ca

یافتن حل‌های بهینه و یا نزدیک به بهینه در مسائل زمانبندی کارگاه‌های، یک الگوریتم ابتکاری بر مبنای الگوریتم ژنتیک ارائه شده است. الگوریتم جستجوی ممنوع، به عنوان یک الگوریتم فرابتکاری دیگر، برای حل بسیاری از مسائل بهینه سازی، خصوصاً در زمینه زمانبندی تولید، مورد توجه محققین بوده است. به عنوان مثال، در [۷] از الگوریتم جستجوی ممنوع برای حل مسئله زمانبندی کارگاه‌های، با هدف کمینه سازی زمان اتمام کارها استفاده شده است. در این مسئله کارها به وسیله یک ربات از ماشین به ماشین دیگر انتقال میابند. بنابراین علاوه بر زمان پردازش کارها روی ماشین‌ها، زمان حرکت و انتقال کارها توسط ربات نیز در نظر گرفته می‌شود. از آنجایی که اثربخشی و راندمان این الگوریتم در مسائل کارگاه‌های، وابسته به یک استراتژی مناسب برای ارزیابی تعویض (جابجایی) عملیات‌های همسایه می‌باشد، در [۸]، یک استراتژی برای اینگونه تعویض‌ها طراحی و به کار گرفته شده است. در [۹] نیز یک الگوریتم ابتکاری بر مبنای الگوریتم جستجوی ممنوع برای حل مسئله زمانبندی کارگاه‌های ارائه شده است. الگوریتم بهینه سازی مورچگان با الهام از رفتار مورچه‌ها برای یافتن کوتاه‌ترین مسیر بین لانه و غذا، به عنوان تکنیکی برای جستجوی بهترین جواب در مسائل بهینه سازی، مورد توجه محققین بوده است. در [۱۰] الگوریتم مورچگان برای حل مسائل زمانبندی کارگاه‌های به کار رفته است. الگوریتم با روشی بر مبنای جستجوی محلی، به منظور افزایش سرعت در رسیدن به جواب نهایی و بالا بردن کیفیت آن، ترکیب شده است. در [۱۱] نیز یک الگوریتم ابتکاری از ترکیب الگوریتم مورچگان و جستجوی ممنوع، برای حل مسئله زمانبندی کارگاه‌های با هدف کمینه سازی زمان اتمام کارها، ارائه شده است. الگوریتم شبیه سازی تبرید نیز، به عنوان یکی دیگر از روش‌های جستجوی تصادفی، برای حل مسائل بهینه سازی، توسط محققین مورد استفاده قرار گرفته است. به عنوان مثال در [۱۲] استفاده از این الگوریتم برای حل مسئله کارگاه‌های، با هدف کمینه کردن دیرکرد وزنی کارها ارائه شده است. در [۱۳] نیز با ترکیب این الگوریتم با الگوریتم ژنتیک مسئله کارگاه‌های مورد بررسی قرار گرفته است. در [۱۴] الگوریتم شبیه سازی تبرید برای حل مسائل زمانبندی در سیستم‌های تولیدی کارگاه‌های سلولی و با هدف کمینه سازی زمان اتمام کارها ارائه شده است. در این مقاله فرض بر این است که هر کار در مسیر تولید خود ممکن است یک یا چند ماشین را بیش از یک بار ملاقات کند. الگوریتم انتقال گلوگاه از دیگر الگوریتم‌های ابتکاری است که برای حل مسائل زمانبندی کارگاه‌های مورد استفاده زیاد محققان قرار گرفته است. به عنوان مثال در [۱۵] از ترکیب الگوریتم انتقال گلوگاه و جستجوی ممنوع برای حل مسائل کارگاه‌های، با هدف کمینه سازی زمان اتمام کارها استفاده شده است. هدف از ترکیب دو الگوریتم، تولید

جواب‌های اولیه مناسب و بهبود بهتر آن‌ها گزارش شده است. در [۱۶] یک الگوریتم ابتکاری بر مبنای الگوریتم انتقال گلوگاه، برای حل مسائل کارگاه‌های، با هدف کمینه سازی زمان اتمام کارها، ارائه شده است. در [۱۷] الگوریتم انتقال گلوگاه در ترکیب با الگوریتمی بر مبنای جستجوی محلی برای حل مسائل کارگاه‌های ارائه شده است. الگوریتم بهینه سازی گروه ذرات با الهام از رفتار اجتماعی پرندگان برای یافتن غذا و به عنوان یک تکنیک محاسبه‌ای تکاملی برای حل مسائل بهینه سازی، مورد استفاده محققین قرار گرفته است. به عنوان مثال در [۱۸] از الگوریتم بهینه‌سازی گروه ذرات برای حل مسائل کارگاه‌های استفاده شده است. همچنین به منظور بالا بردن کیفیت جواب‌های تولید شده از الگوریتم جستجوی ممنوع نیز استفاده شده است. در [۱۹] الگوریتم بهینه سازی گروه ذرات برای حل مسائل زمانبندی کارگاه‌های با فرض وجود زمان آماده سازی برای شروع هر عملیات روی ماشین و با هدف کمینه سازی زمان اتمام کارها ارائه شده است. در این تحقیق با توجه به اثرگذاری جواب‌های اولیه بر کیفیت جواب نهایی، به جای تولید جواب‌های تصادفی از یک دسته جواب‌های با کیفیت نسبتاً خوب و با پراکندگی مناسب استفاده شده است. جواب‌های نهایی بدست آمده از این الگوریتم با استفاده از الگوریتم آنیلینگ شبیه سازی شده بهبود داده شده است. از دیگر نمونه‌های کاربرد الگوریتم بهینه سازی گروه ذرات برای حل انواع مختلف مسائل زمانبندی کارگاه‌های توسط محققین، میتوان به [۲۰-۲۲] اشاره کرد. مسئله کارگاه‌های انعطاف‌پذیر، تعمیم یافته مسئله کارگاه‌های است. این ساختار شامل تعدادی ایستگاه کاری است که در آن، حداقل یک ایستگاه دارای بیش از یک ماشین موازی یکسان است. کارهایی که وارد هر ایستگاه می‌شوند، امکان پردازش توسط هر یک از ماشین‌های موجود در آن ایستگاه را دارند. در حقیقت این ساختار ترکیبی از مسئله کارگاه‌های ساده و مسئله ماشین‌های موازی است. تاکنون الگوریتم‌های ابتکاری و فرابتکاری مختلفی برای حل مسائل زمانبندی کارگاه‌های انعطاف‌پذیر ارائه شده است. به عنوان مثال در [۲۳] الگوریتم ابتکاری جدیدی بر مبنای الگوریتم ژنتیک، به منظور حل مسئله زمانبندی کارگاه‌های انعطاف‌پذیر طراحی شده است. در [۲۴] و [۲۵] برای حل مسائل زمانبندی کارگاه‌های انعطاف‌پذیر، با اهداف کمینه کردن زمان اتمام کارها، کمینه کردن بیشترین حجم کاری هر ماشین (برای ایجاد تعادل بین مدت زمان کارکردن ماشین‌ها مؤثر است) و کمینه کردن حجم کاری کل ماشین‌ها (برای مواقعی که راندمان ماشین‌ها متفاوت باشند مؤثر است)، به ترتیب یک الگوریتم ابتکاری بر مبنای الگوریتم ژنتیک و انتقال گلوگاه و یک الگوریتم ابتکاری بر مبنای الگوریتم بهینه سازی گروه ذرات ارائه شده است. در [۲۶] الگوریتم ابتکاری بر مبنای الگوریتم جستجوی ممنوع برای حل مسائل کارگاه‌های انعطاف‌پذیر، با اهداف کمینه

الگوریتم ایمنی مصنوعی حل شده اند، با الگوریتم پیشنهادی حل و نتایج با یکدیگر مقایسه شده اند.

در ادامه در بخش ۲ تعریف مسئله و نحوه مدل‌سازی آن شرح داده می‌شود. در بخش ۳، الگوریتم بهینه سازی گروه ذرات بطور مختصر مرور و سپس در بخش ۴ الگوریتم پیشنهادی برای حل مسئله مذکور ارائه می‌شود. در بخش ۵ نتایج استفاده از الگوریتم پیشنهادی در حل ۳۰ مسئله نمونه و مقایسه آن با نتایج حاصل از استفاده الگوریتم ایمنی مصنوعی مطرح خواهد شد. در پایان در بخش ۶ نتیجه‌گیری و پیشنهاداتی برای مطالعات آتی آورده شده است.

۲. تعریف و مدل‌سازی مسئله

۲-۱. تعریف مسئله

در این مقاله، مسئله زمانبندی کارگاه‌های چند مسیره، با هدف تعیین توالی انجام کارها به گونه‌ای که زمان اتمام کارها حداقل شود، مورد بررسی قرار می‌گیرد. در مسئله کارگاه‌های چند مسیره هرکار می‌تواند از مسیرهای متفاوتی عبور کرده و به اتمام برسد. تعداد ماشین‌هایی که یک کار در مسیرهای مختلف با آن مواجه می‌شود، لزوماً یکسان نیستند. به عبارت دیگر، ممکن است در یک مسیر، کار، با عبور از دو ماشین (هر ماشین یک عملیات) به اتمام برسد و در مسیر دیگری همان کار با عبور از پنج ماشین انجام شود. جمع زمان مورد نیاز برای اتمام یک کار، با توجه به مسیر انتخابی برای آن کار، تعیین و این زمان در مسیرهای مختلف برای آن کار لزوماً برابر نخواهد بود. پردازش هرکار بر روی هر ماشین یک فعالیت نامیده می‌شود و بین هر دو فعالیت از یک کار، رابطه پیش‌نیازی وجود دارد که از قبل معلوم می‌باشد. فرض بر این است که بین هر دو فعالیت از کارهای مختلف هیچگونه رابطه پیش‌نیازی وجود ندارد. هرکار در هر لحظه زمانی، تنها می‌تواند بر روی یک ماشین پردازش شود و هر ماشین نیز در هر لحظه زمانی، حداکثر می‌تواند یک کار را پردازش کند. فرض می‌شود که همه کارها در لحظه صفر در دسترس هستند.

در مسئله زمانبندی کارگاه‌های چند مسیره باید به دو سوال مهم جواب داد: اول اینکه هرکار چه مسیری را برای پردازش انتخاب کند و دوم آنکه چه توالی از عملیات‌های تخصیص داده شده به هر ماشین، مناسب تر است. هدف، تعیین زمانبندی عملیات کارها بگونه‌ای است که زمان اتمام کارها حداقل گردد. در ادامه ضمن تعریف نمادهای مورد نیاز، نحوه مدل‌سازی ریاضی مسئله زمانبندی کارگاه‌های چند مسیره ارائه خواهد شد.

۲-۲. نمادها و متغیرهای مورد استفاده:

فرض کنید n معرف تعداد کارها و m معرف تعداد ماشین‌ها باشد. برای انجام کار j ام، $n, 2, 1, \dots, n = j$ ، چند مسیر پردازش وجود دارد.

کردن زمان اتمام کارها، کمینه کردن حجم کاری کل ماشین‌ها و کمینه کردن حجم کاری ماشین بحرانی (ماشینی که بیشترین حجم کاری را نسبت به دیگر ماشین‌ها دارد) ارائه شده است. در [۲۷] الگوریتم ایمنی مصنوعی برای حل مسائل کارگاه‌های انعطاف‌پذیر، با هدف کمینه کردن زمان اتمام کارها، ارائه شده است. در این الگوریتم چند استراتژی جدید برای تولید جواب‌های اولیه مناسب، همچنین تکثیر (تولید مثل) و بهبود تعدادی از جواب‌های اولیه انتخاب شده، ارائه شده است. در [۲۸-۳۲] نمونه‌های دیگری از الگوریتم‌های ابتکاری برای حل مسائل زمانبندی کارگاه‌های انعطاف‌پذیر، با در نظر گرفتن فرضیات و اهداف مختلف زمانبندی، توسط محققین ارائه شده است. تعمیم وسیع‌تر مسئله کارگاه‌های، مسئله کارگاه‌های چند مسیره است که در آن، هرکار می‌تواند از مسیرهای متفاوتی عبور کرده و به اتمام برسد. لزوماً تعداد ماشین‌ها در مسیرهای مختلف برای یک کار برابر نیست. این ساختار در اغلب واحدهای تولیدی منعطف، که در آن‌ها ماشین‌های کنترل عددی قادر به انجام عملیات متفاوتی از یک کار هستند، مشاهده می‌شود. مسئله زمانبندی کارگاه‌های چند مسیره شامل دو زیر مسئله (۱) تخصیص هرکار به یک مسیر از میان مسیرهای موجود برای آن کار و (۲) تعیین توالی عملیات‌های تخصیص داده شده به ماشین‌ها، می‌باشد. باید توجه داشت یکی از تفاوت‌های اساسی محیط کارگاه‌های چند مسیره با محیط کارگاه‌های انعطاف‌پذیر این است که در محیط کارگاه‌های انعطاف‌پذیر برای انجام یک کار عملیات‌های مشخصی در نظر گرفته شده است و تصمیم‌گیری روی انتخاب یکی از ماشین‌های انجام دهنده هر یک از عملیات‌های کار می‌باشد. اما در محیط کارگاه‌های چند مسیره تعداد عملیات‌های هرکار، توالی انجام عملیات‌ها و ماشین‌های انجام دهنده عملیات‌های کار با توجه به مسیری که برای هرکار انتخاب می‌شود ممکن است متفاوت باشد. بنابراین، وجود مسیرهای مختلف برای انجام یک کار موجب پیچیدگی بیشتر مسئله نسبت به مسائل کارگاه‌های و کارگاه‌های انعطاف‌پذیر که در گروه مسائل هستند، خواهد شد. در [۳۳]

الگوریتم ایمنی مصنوعی برای حل مسئله زمانبندی کارگاه‌های چند مسیره، با هدف کمینه کردن زمان اتمام کارها، ارائه شده است. در تحقیق مذکور، ۳۰ مسئله در ابعاد کوچک، متوسط و بزرگ طراحی و توسط الگوریتم ارائه شده و همچنین توسط نرم افزار بهینه سازی ۸، حل شده‌اند. در مقاله حاضر، یک مدل برنامه‌ریزی غیرخطی عدد صحیح مختلط برای حل مسئله زمانبندی کارگاه‌های چند مسیره، ارائه می‌شود. همچنین یک الگوریتم فراابتکاری، بر مبنای الگوریتم بهینه سازی گروه ذرات با هدف کمینه سازی زمان اتمام کارها، برای حل مسئله مذکور ارائه خواهد شد. و نهایتاً جهت بررسی و مقایسه عملکرد الگوریتم پیشنهادی، مسائل نمونه در [۳۳] که با استفاده از

ملاقات می‌کند، می‌باشد. جهت مدل‌سازی مسئله، دو نوع متغیر صفر و یک به شرح زیر تعریف می‌شود:

اگر کار z ام به مسیر پردازش k اختصاص یابد

$$Z_{kj} = \begin{cases} 1, \\ 0, \end{cases}$$

در غیر اینصورت

برای انجام عملیات، زودتر i با ماشین j امین دفعه‌ی ملاقات کار s اگر صورت گیرد i با ماشین j' امین دفعه‌ی ملاقات کار s' از

$$X_{ijsj's'} = \begin{cases} 1, \\ 0, \end{cases}$$

در غیر اینصورت

۳-۲. مدل ریاضی مسئله:

مدل ریاضی مسئله زمانبندی کارگاه‌های چند مسیره، یک مدل از نوع برنامه‌ریزی غیرخطی عدد صحیح مختلط است و به صورت زیر بیان می‌شود:

$$\text{Minimize } C_{max} \quad C_{max} \geq \forall j = 1, 2, \dots, n; \forall k = 1, 2, \dots, K_j \quad (1)$$

$$F_{i_1 j k s_{1 i_1}} \geq P_{i_1 j k s_{1 i_1}} \times Z_{kj} \quad \forall j = 1, 2, \dots, n; \forall k = 1, 2, \dots, K_j; S_{1 i_1} = 1 \quad (2)$$

$$F_{i_{p'} j k s_{p' i_{p'}}} \geq \left(F_{i_{p'} j k s_{p i p}} + P_{i_{p'} j k s_{p' i_{p'}}} \right) \times Z_{kj} \quad (3)$$

$$\forall p = 1, 2, \dots, N_{jk} - 1; p' = p + 1; \forall j = 1, 2, \dots, n; \forall k = 1, 2, \dots, K_j \quad (4)$$

$$\left\{ \begin{array}{l} F_{ijks} \geq F_{ij'k's'} + P_{ijks} + L \times (Z_{kj} + Z_{k'j'} + X_{ij's'js} - 3) \\ F_{ij'k's'} \geq F_{ijks} + P_{ijks} + L \times (Z_{ks} + Z_{k's'} + X_{ijsj's'} - 3) \\ X_{ij's'js} + X_{ijsj's'} = 1 \end{array} \right\} X_{ij's'js}, X_{ijsj's'} \in \{0, 1\}$$

$$\forall i = M_1, M_2, \dots, M_m; \forall j, j' \in \{1, 2, \dots, n\}; \forall k(k') \in \{1, 2, \dots, K_j(j')\}; \forall s(s') = 1, 2, \dots, S_{N_{jk(j'k')i}} \quad (5)$$

$$\sum_k Z_{kj} = 1 \quad \forall j = 1, 2, \dots, n; \forall k = 1, 2, \dots, K_j; Z_{kj} \in \{0, 1\}$$

بزرگ است و با بی‌اثر کردن تعداد زیادی از محدودیت‌های مسئله، از طریق مقادیری که به متغیرهای تصمیم‌گیری تخصیص داده می‌شود، برای ساده کردن مدل ارائه شده به کار برده می‌شود. مجموعه معادلات (۵)، صرفاً یک مسیر از بین مسیرهای مختلف را به هر کار تخصیص می‌دهد.

در ضمیمه ۱، جهت شرح بهتر هریک از معادلات و یا نامعادلات موجود در مدل ارائه شده، از یک مثال استفاده و مدل فوق برای آن به کار برده شده است.

۳. الگوریتم بهینه سازی گروه ذرات

۳-۱. الگوریتم بهینه سازی گروه ذرات

کندی و ابره‌ارت در سال ۱۹۹۵ با الهام گرفتن از هوش جمعی پرندگان، الگوریتم بهینه سازی گروه ذرات، که یک تکنیک بهینه

مسیره‌های پردازش برای هر کار توسط توالی ماشین‌ها مشخص می‌شود. تعداد کل مسیرهای پردازش کار z ام را با K_j و تعداد کل عملیات‌های کار z ام، در مسیر k ام، را با N_{jk} نشان می‌دهیم. توالی عملیات‌های کار z ام، در مسیر k ام را با مجموعه‌ای به نام $i_p \in \{1, 2, \dots, m\}, \forall p = 1, 2, \dots, N_{jk}$ که در آن $I_{jk} = \{i_1, i_2, \dots, i_{N_{jk}}\}$ نشان می‌دهیم. i_p از مجموعه‌ی I_{jk} نشان دهنده‌ی ماشین اجراکننده‌ی عملیات p ام از کار z ام در مسیر k ام می‌باشد. از آنجایی که ممکن است کار z ام در مسیر k ام ماشین خاصی را بیش از یک بار ملاقات کند، برای نمایش تعداد ملاقات‌های کار با آن ماشین (مثلاً ماشین i_p)، در طول اجرای p عملیات از مسیر پردازش k ام، از $S_{p i p}$ استفاده می‌کنیم. $P_{i p j k s}$ معرف زمان مورد نیاز برای انجام p امین عملیات کار z ام در مسیر k ام که توسط ماشین i_p انجام می‌شود و S امین $(S \in \{1, 2, \dots, S_{N_{jk} i p})$ دفعه‌ای باشد که کار z ام ماشین i_p را ملاقات می‌کند، می‌باشد. L نماد یک عدد ثابت و بزرگ است. متغیر تصمیم‌گیری F_{ijks} بیان‌کننده زمان خروج کار z ام در مسیر k ام از ماشین $i \in \{1, 2, \dots, m\}$ وقتی S امین باری باشد که کار z ام ماشین i را

مجموعه نامعادلات (۱)، موجب می‌شود تا زمان اتمام آخرین عملیات از کار z ام در مسیر k ام توسط ماشین $i_{N_{jk}}$ (ماشینی که آخرین عملیات کار z ام در مسیر k ام را انجام می‌دهد)، کمتر یا مساوی با زمان انجام کلیه کارها، C_{max} ، باشد. مجموعه نامعادلات (۲) بیان می‌کند که زمان اتمام اولین عملیات کار z ام در مسیر k ام که توسط ماشین i_1 (ماشینی که اولین عملیات کار z ام در مسیر k ام را انجام می‌دهد)، لزوماً با زمان خروج آن کار از ماشین و شروع عملیات جدید روی ماشین دیگر برابر نخواهد بود. مجموعه نامعادلات (۳) موجب می‌شوند تا روابط پیش‌نیازی بین عملیات‌های هر کار، با توجه به مسیر انتخابی برای کارها، رعایت شود؛ به عبارت دیگر، موجب می‌شوند تا هر یک از عملیات‌های یک کار تنها وقتی شروع شود که عملیات‌های قبلی آن انجام شده باشد. مجموعه نامعادلات (۴)، تعیین‌کننده توالی کارها روی هر ماشین است. باید توجه داشت که پارامتر L یک مقدار مثبت و

که W از یک مقدار نزدیک به ۱، شروع شود و بتدریج در هر مرحله تکرار الگوریتم کم شده تا به یک مقدار کوچک برسد. مقادیر بزرگتر از یک ممکن است باعث ناپایدار شدن سیستم شود در حالی که مقادیر کوچکتر از یک به منزله جستجوی بیشتر در کل فضای جستجو است و مقادیر کوچکتر به منزله جستجوی دقیق‌تر در اطراف نقاط بهینه محلی است. ضریب وزن اینرسی از رابطه (۸) محاسبه می‌شود [۴۰].

$$w = W_{max} - \left(\frac{W_{max} - W_{min}}{N} \right) \times k \quad (۸)$$

در رابطه (۸)، k اندیس تکرار جاری و N تعداد کل تکرارها در الگوریتم است. W_{max} و W_{min} ثابت بوده و توسط کاربر تعیین می‌شود. در هر تکرار الگوریتم، سرعت را در ضریبی کوچکتر از یک (ضریب محدودسازی سرعت یا ضریب انقباض) ضرب می‌کنند تا از سرعت پراکنده شدن بیش از حد ذرات کاسته شود. این ضریب δ نامیده می‌شود و از رابطه (۹) بدست می‌آید [۴۰].

$$\left\{ \begin{array}{l} \delta = \frac{2}{\left| \varphi - 2 + \sqrt{\varphi^2 - 4\varphi} \right|} \\ \varphi = c_1 + c_2, \quad \varphi > 4 \end{array} \right. \quad (۹)$$

۳-۲. الگوریتم بهینه سازی گروه ذرات دودویی

کندی و ابرهارت در سال ۱۹۹۷، الگوریتم بهینه سازی گروه ذرات دودویی را ابداع کردند [۴۱]. این الگوریتم دارای یک فضای حل گسسته است که می‌تواند مربوط به یک مسئله گسسته خالص باشد (مانند مسئله‌های مسیریابی که نیاز به مرتب کردن یا ردیف کردن عناصر گسسته دارند) یا مربوط به یک مسئله پیوسته باشد که فضای حل آن به کدهای دودویی تبدیل می‌شود. در الگوریتم بهینه سازی گروه ذرات دودویی، موقعیت هر ذره، یک رشته از بیت‌های دودویی است که مقادیر ۰ یا ۱ را به خود می‌گیرند. برای هر ذره یک بردار سرعت تعریف می‌شود. عناصر تشکیل دهنده بردار سرعت، نشان دهنده امکان تغییر وضعیت رشته بیت‌های بردار موقعیت، از مقدار ۰ به ۱ یا بالعکس می‌باشند. بنابراین، هر ذره به کمک بردار سرعت نسبت داده شده به آن، می‌تواند در یک فضای جستجوی محدود شده به ۰ و ۱ در هر بُعد، حرکت کند [۴۱].

معادله‌های موقعیت و سرعت هر ذره در الگوریتم بهینه سازی گروه ذرات دودویی بصورت زیر می‌باشند:

$$v_{id}(t+1) = v_{id}(t) + \varphi_1 \cdot (P_{id}(t) - x_{id}(t)) + \varphi_2 \cdot (P_{gd}(t) - x_{id}(t))$$

$$x_{id}(t+1) = \begin{cases} 1 & \rho_{id} < \text{sigmoid}(v_{id}(t+1)) \\ 0 & \text{درغیراینصورت} \end{cases} \quad (۱۰)$$

سازی تکاملی است و ماهیت پیوسته دارد، را ابداع کردند [۳۴-۳۵]. الگوریتم از تعداد مشخصی ذره (جواب اولیه)، تشکیل می‌شود که به طور تصادفی، مقدار می‌گیرند. برای هر ذره یک بردار مکان و یک بردار سرعت تعریف می‌شود. ذرات، مکرراً در فضای n بعدی مسئله و با توجه به بردار سرعت در هر تکرار، حرکت می‌کنند و با محاسبه مقدار بهینگی به عنوان یک ملاک سنجش، گزینه‌های ممکن جدید را بررسی می‌کنند. بُعد فضای مسئله، برابر تعداد پارامترهای موجود در تابع مورد نظر برای بهینه سازی می‌باشد. یک حافظه برای ذخیره کردن بهترین موقعیت هر ذره در گذشته (بهترین جواب محلی)، و یک حافظه برای ذخیره کردن بهترین موقعیت پیش آمده در میان همه ذرات (بهترین جواب کلی)، اختصاص می‌یابد. با تجربه حاصل از این حافظه‌ها، ذرات تصمیم می‌گیرند که در تکرار بعدی، چگونه حرکت کنند. در هر بار تکرار، همه ذرات در فضای n بعدی مسئله حرکت می‌کنند تا بالاخره با لحاظ شرایط توقف الگوریتم بهترین جواب کلی به عنوان جواب حاصل از الگوریتم تعیین گردد. در هر تکرار، سرعت‌ها و موقعیت‌های هر ذره بر حسب بهترین جواب‌های مطلق و محلی به روز می‌شوند [۳۶-۳۷]. بردارهای سرعت و موقعیت هر ذره بصورت زیر بهنگام می‌شوند [۳۸]:

$$\vec{v}_k^i = \delta \cdot (w \cdot \vec{v}_{k-1}^i + c_1 \cdot \vec{r}_1 \cdot (\vec{p}_k^i - \vec{x}_k^i) + c_2 \cdot \vec{r}_2 \cdot (\vec{p}_k^q - \vec{x}_k^i)) \quad (۶)$$

$$\vec{x}_{k+1}^i = \lfloor \vec{x}_k^i + \vec{v}_k^i \rfloor \quad (۷)$$

که در آن، برای ذره i ام، \vec{x}_k^i بردار موقعیت ذره، \vec{v}_k^i بردار سرعت ذره، \vec{p}_k^i بردار بهترین موقعیت بدست آمده توسط ذره و \vec{p}_k^q بردار بهترین موقعیت بدست آمده توسط تمام ذرات در مرحله k ام الگوریتم را نشان می‌دهد. برای جلوگیری از یکنواختی حرکت ذره‌ها در فضای جواب، از اعداد تصادفی در هر تکرار الگوریتم، استفاده می‌شود [۳۹]. بردارهای \vec{r}_1, \vec{r}_2 اعداد تصادفی در بازه $[0, 1]$ می‌باشند. c_1, c_2 ثوابت شتاب هستند و توسط کاربر تعیین می‌گردند. هر چه مقدار این ثوابت بیشتر باشد شتاب و حرکت‌های ناگهانی ذرات در فضای جواب بیشتر خواهد بود، (برای مسائلی با فضای حل بزرگ مناسب‌تر است) و هر چه مقدار آن‌ها کمتر باشد حرکت ذره‌ها هموارتر خواهد بود (برای مسائلی با فضای حل کوچک مناسب‌تر است). W ضریب وزن اینرسی بوده و مقادیر آن در هر مرحله از تکرار الگوریتم متفاوت است. معمولاً توصیه می‌شود

پیش‌نیازها رعایت شده باشد)، نشان می‌دهد. همچنین بطور مشابه با [۴۲]، برای ایجاد تطبیق بین شیوه نمایش بر مبنای فهرست اولویت که ماهیت گسسته دارد، با شیوه مورد نیاز در الگوریتم که ماهیتی پیوسته دارد، از روش تبدیل مبنای اعداد به مبنای فاکتوریل استفاده می‌شود. این تبدیل مبنای فاکتوریلی، به طور غیر مستقیم، عمل تبدیل یک جایگشت به یک عدد در مبنای ۱۰ و بالعکس را با یک واسطه انجام می‌دهد.

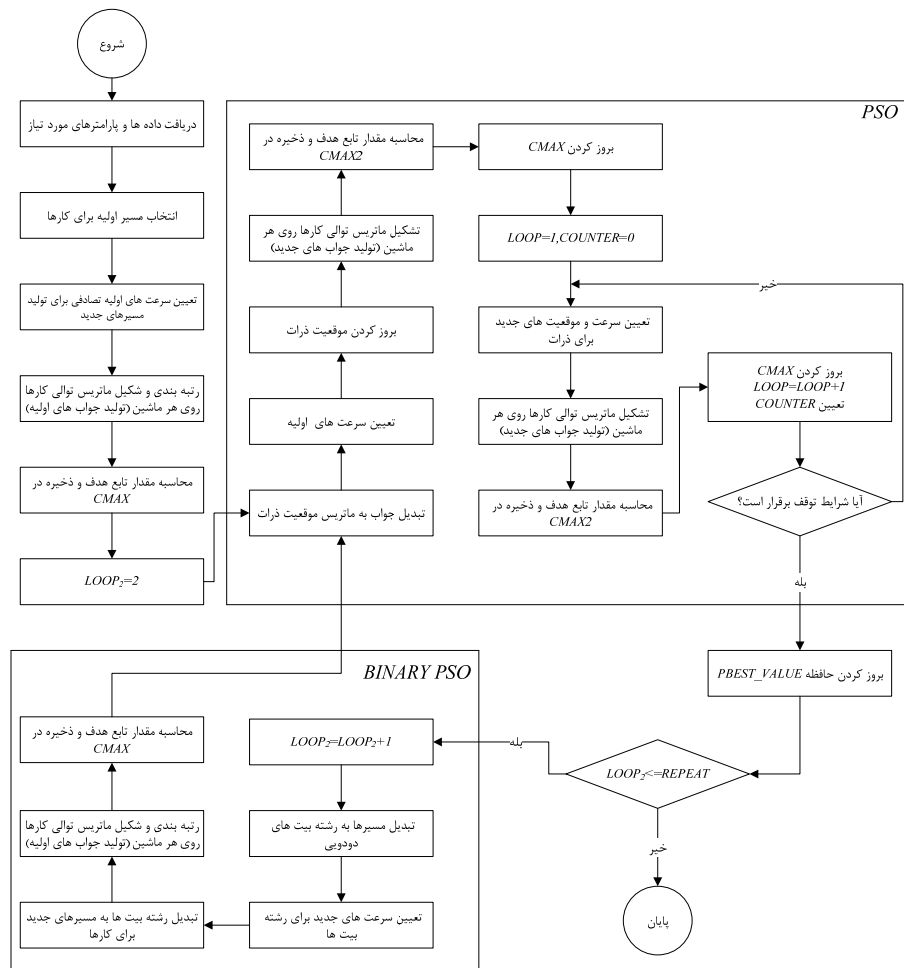
فلوچارت الگوریتم ارائه شده برای مسئله‌ی زمانبندی کارگاه‌های چند مسیره، در شکل ۱ نشان داده شده است. همان‌طور که در شکل ۱ مشاهده می‌شود، الگوریتم پیشنهادی به دو بخش کلی تقسیم می‌گردد: (۱) با استفاده از الگوریتم بهینه سازی گروه ذرات دودویی، مسیرهای مناسب برای کارها و در نهایت جواب‌های اولیه (ذرات اولیه)، تولید می‌شوند و (۲) با استفاده از الگوریتم بهینه سازی گروه ذرات، جواب‌های اولیه تولید شده (ذرات اولیه)، در فضای n بُعدی مسئله و با توجه به بردار سرعت در هر تکرار، به سمت نقطه بهینه حرکت می‌کنند. الگوریتم دو بخش فوق را مکرراً تکرار می‌کند و پس از وقوع شرایط توقف، بهترین جواب برای مسئله مورد نظر بدست خواهد آمد. لازم به ذکر است که مقادیر پارامترهای به کار برده شده در الگوریتم پیشنهادی، به صورت تجربی و بر اساس ارزیابی نتایج به دست آمده از به کار بردن الگوریتم پیشنهادی برای حل تعداد زیادی از مسائل تحت مقادیر مختلف آن پارامترها، در نظر گرفته شده است.

که در آن، برای هر ذره i ، $x_{id}(t)$ حالت کنونی بیت d ام ذره، $v_{id}(t)$ سرعت نسبت داده شده به بیت d ام ذره، $P_{id}(t)$ بهترین حالت یافت شده برای بیت d ام ذره، $P_{gd}(t)$ بهترین حالت یافت شده برای بیت d ام توسط تمام ذرات در مرحله‌ی t ام تکرار الگوریتم می‌باشد. عدد تصادفی تولید شده با توزیع یکنواخت در بازه $[0, 1]$ است. مقادیر φ_1 و φ_2 نیز اعداد تصادفی تولید شده با توزیع یکنواخت هستند. معمولاً در الگوریتم بهینه سازی گروه ذرات دودویی مقادیر φ_1 و φ_2 را بگونه‌ای انتخاب می‌کنند که در رابطه $\varphi_1 + \varphi_2 \leq 4$ صدق کند. بردار سرعت، که امکان اتخاذ تصمیم ۰ یا ۱ را نشان می‌دهد، ممکن است در طی مراحل تکرار الگوریتم خارج از بازه $(0, 1)$ قرار گیرد؛ بنابراین از تابع *sigmoid*، رابطه (۱۱)، برای تغییر مقیاس استفاده می‌شود.

$$\text{sigmoid}(v_{id}(t)) = \frac{1}{1 + \exp(-v_{id}(t))} \quad (11)$$

۴. الگوریتم پیشنهادی

الگوریتم پیشنهادی در این مقاله، بر مبنای الگوریتم بهینه سازی گروه ذرات و با هدف کمینه‌سازی زمان اتمام کارها، برای حل مسئله زمانبندی کارگاه‌های چند مسیره می‌باشد. در الگوریتم پیشنهادی، به منظور حفظ موجه بودن جواب‌ها، در هر تکرار الگوریتم، از شیوه نمایش بر مبنای فهرست اولویت کارها برای هر ماشین استفاده می‌شود. این فهرست، اولویت انجام یک کار بر کار دیگر را در صورتیکه قابل انجام باشد (محدودیت مسیر پردازش و



شکل ۱. فلوچارت الگوریتم ارائه شده

گام‌های الگوریتم پیشنهادی برای مسئله زمانبندی کارگاه‌های چند مسیره در مراحل زیر خلاصه می‌شود:

۱. داده‌های مورد نیاز برای تولید جواب‌های اولیه را بگیرید. این داده‌ها شامل تعداد کارها ($Njob$)، تعداد ماشین‌ها ($Nmachine$)، تعداد مسیره‌هایی که برای انجام هر کار وجود دارد (ماتریس $Npath$)، زمان پردازش هر کار بر روی ماشین‌های مختلف در هر مسیر (حافظه $Processing\ time$) و مسیرهای پردازش برای کارها (حافظه $Sequence$) می‌باشد.

۲. برای هر کار به صورت تصادفی مسیری انتخاب کنید. انتخاب مسیر برای کارها به این صورت است که هر یک از اعداد ماتریس $Npath$ را به یک رشته از بیت‌های دودویی که مقادیر ۰ یا ۱ را به خود می‌گیرند، تبدیل کنید؛ سپس با قرار دادن مقادیر ۰ یا ۱ بصورت تصادفی در هر بیت و تبدیل رشته

بیت‌های جدید به عددی در مبنای ۱۰، برای هر کار مسیری انتخاب می‌شود. مجموعه مسیرهای انتخابی برای کارها در هر تکرار الگوریتم با هم متفاوت خواهند بود.

۳. ماتریس سرعت v را با توجه به ماتریس $Npath$ تشکیل دهید. در ماتریس v ، سطرها نشان دهنده کارها و ستون‌ها نشان دهنده رشته بیت‌های دودویی هستند. بنابراین تعداد سطرها ماتریس v برابر $Njob$ خواهد بود. برای پیدا کردن تعداد ستون‌های ماتریس v ، بیشترین مقدار ماتریس $Npath$ را به رشته بیت‌های دودویی تبدیل کنید. طول رشته بیت تولید شده، تعداد ستون‌های ماتریس v را نشان می‌دهد. با استفاده از رابطه (۱۲)، برای هر یک از درایه‌های این ماتریس، v_{id} ، مقداری تولید کنید. مجموعه مقادیر تولید شده در هر سطر، بردار سرعت برای مسیر هر کار را تعیین می‌کند. مقدار Bin_v_{max} برابر با ۴ در نظر گرفته می‌شود [۴۳].

۱. داده‌های مورد نیاز برای تولید جواب‌های اولیه را بگیرید. این داده‌ها شامل تعداد کارها ($Njob$)، تعداد ماشین‌ها ($Nmachine$)، تعداد مسیره‌هایی که برای انجام هر کار وجود دارد (ماتریس $Npath$)، زمان پردازش هر کار بر روی ماشین‌های مختلف در هر مسیر (حافظه $Processing\ time$) و مسیرهای پردازش برای کارها (حافظه $Sequence$) می‌باشد.

۲. برای هر کار به صورت تصادفی مسیری انتخاب کنید. انتخاب مسیر برای کارها به این صورت است که هر یک از اعداد ماتریس $Npath$ را به یک رشته از بیت‌های دودویی که مقادیر ۰ یا ۱ را به خود می‌گیرند، تبدیل کنید؛ سپس با قرار دادن مقادیر ۰ یا ۱ بصورت تصادفی در هر بیت و تبدیل رشته

$$v_{id} = (Bin_v_{max} \times rand) - 0.5 \quad \forall i = 1, 2, \dots, Njob; \quad \forall d = 1, 2, \dots, length((\max(Npath))_2) \quad (12)$$

$$\vec{v}_i(1) = [v_{i1}, v_{i2}, \dots, v_{id}] \forall i = 1, 2, \dots, N_{job}; \forall d = 1, 2, \dots, \text{length}((\max(N_{path}))_2) \quad (13)$$

$$v = [\vec{v}_1(1); \vec{v}_2(1); \dots; \vec{v}_{N_{job}}(1)] \quad (14)$$

$$\vec{x}_1 = [x_1^1, x_1^2, x_1^3, \dots, x_1^{N_{machine}}] \quad (15)$$

۸. اولین ماتریس سرعت برای هر یک از ذرات را به صورت اعداد صحیح تصادفی در بازه زیر تولید کنید:

$$\vec{v}_1 = [v_1^1, v_1^2, v_1^3, \dots, v_1^{N_{machine}}] \quad (16)$$

$$v_1^i \in [-x_1^i, x_1^i] \forall i = 1, 2, \dots, N_{machine} \quad (17)$$

۹. موقعیت ذرات را با استفاده از ماتریس سرعت بدست آمده بروز کنید.

$$x_2^i = [x_1^i + v_1^i] \quad \forall i = 1, 2, \dots, N_{machine} \quad (18)$$

۱۰. موقعیت جدید ذرات را با روش تبدیل مبنای فاکتوریلی به یک جایگشت تبدیل کنید و در ماتریسی به نام *Permutation* ذخیره کنید. مقدار تابع هدف را محاسبه کرده و در حافظه *CMAX2* ذخیره کنید. مقدار *CMAX2* ذخیره شده را با *CMAX* مقایسه کنید و از این طریق *CMAX* و حافظه *T_O* را بروز کنید.

۱۱. قرار دهید؛ $LOOP = 2, COUNTER = 0$

۱۲. سرعت و موقعیت جدید ذرات را از روابط زیر تعیین کنید:

$$\left\{ \begin{array}{l} v_{Loop}^i = \delta \cdot (w \cdot v_{Loop-1}^i + c_1 \cdot \text{rand} \cdot (p_{Loop}^i - x_{Loop}^i)) \\ \text{if} : v_{Loop}^i < (0 - x_{Loop}^i) \text{ then} : v_{Loop}^i = \text{rand} \times (0 - x_{Loop}^i) \\ \text{if} : v_{Loop}^i > x_{Loop}^i \text{ then} : v_{Loop}^i = \text{rand} \times (x_{Loop}^i) \\ \forall i = 1, 2, \dots, N_{machine} \end{array} \right\} \quad (19)$$

۲. شروط توقف زیر را بررسی کنید.

شروط اول : تعداد تکرارهای الگوریتم از مقدار تعیین شده N بیشتر باشد. مقدار N برابر ۲۰۰ در نظر گرفته می‌شود.

شروط دوم : مقدار $COUNTER > REPEAT_2$ باشد. $REPEAT_2$ برابر ۱۰ در نظر گرفته می‌شود.

۳. در صورتی که هیچ یک از شروط برقرار نباشد قرار دهید؛ $LOOP = LOOP + 1$ ، به گام ۱۲ برگردید. در صورتی که شروط برقرار بود، مقدار *CMAX* و ماتریس *T_O* را به ترتیب در حافظه‌های *PBEST_VALUE* و *PBEST_T* ذخیره کنید. مجموعه مسیر انتخابی برای کارها را در حافظه *PBEST* ذخیره کنید.

۴. ماتریس‌های O, P را با توجه به مسیری که برای هر کار انتخاب شده است، به ترتیب از حافظه‌های *Processing time* و *Sequence* تشکیل دهید. در این دو ماتریس، سطرها نشان دهنده کارها و ستون‌ها نشان دهنده ماشین‌ها هستند. درایه‌های ماتریس P ، زمان پردازش کارها روی هر ماشین را نشان می‌دهد. همچنین درایه‌های ماتریس O ، مسیر پردازش هر کار را نشان می‌دهند.

۵. با توجه به توالی قرارگیری هر کار در ماشین‌های مختلف (توالی‌ها در ماتریس O مشخص شده است)، رتبه هر کار بر روی هر ماشین را محاسبه کنید (کارها ممکن است رتبه یکسان داشته باشند). بنابراین اولویت کارهای ۱ تا N_{job} برای هر ماشین مشخص می‌شود. اولویت‌های مشخص شده را در ماتریس T_O بریزید.

۶. مقدار تابع هدف، زمان تکمیل تمام کارها، را محاسبه کنید و مقدار حاصل را در حافظه‌ای به نام *CMAX* ذخیره کنید. قرار دهید؛ $LOOP_2 = 2$

۷. ماتریس T_O را به عنوان یک جایگشت فرض کرده و با روش تبدیل مبنای فاکتوریلی، توالی کارها بر روی هر ماشین را به یک عدد در مبنای ۱۰ تبدیل کنید. به این ترتیب بردار \vec{x}_1 تشکیل می‌شود. این بردار نشان دهنده موقعیت ذره مورد استفاده در الگوریتم است و به صورت زیر نشان داده می‌شود.

که در آن، rand یک عدد تصادفی در بازه $[0, 1]$ است. p_{Loop}^i درایه‌های مربوط به ماتریس بهترین موقعیت بدست آمده توسط ذره در مرحله *Loop* ام الگوریتم می‌باشد. پارامترهای w و δ اعداد ثابت و به ترتیب برابر 0.9 و 0.1 در نظر گرفته می‌شود.

$$x_{Loop+1}^i = [x_{Loop}^i + v_{Loop}^i] \quad (20)$$

۱. موقعیت جدید ذرات را به یک جایگشت تبدیل کنید و در ماتریس *Permutation* ذخیره کنید. مقدار تابع هدف را بدست آورید و در حافظه *CMAX2* ذخیره کنید. مقدار *CMAX2* ذخیره شده را با *CMAX* مقایسه کنید و از این طریق *CMAX* و حافظه *T_O* را بروز کنید. در صورت تغییر مقدار *CMAX*، قرار دهید؛ $COUNTER = 0$. در غیر این صورت قرار دهید؛ $COUNTER = COUNTER + 1$.

کارگاه‌های چند مسیره، طراحی و توسط نرم افزار بهینه‌سازی LINGO ۸ و همچنین توسط الگوریتم ایمنی مصنوعی حل شده اند. همانطور که در جدول ۱ نشان داده شده است، مسائل با ابعاد کوچک، S1-S10، شامل ۲ تا ۳ کار با ۳ تا ۴ ماشین و هر کار با ۱ تا ۳ مسیر انتخابی می‌باشند. مسائل طراحی شده با ابعاد متوسط، M1-M10، شامل ۵ تا ۶ کار با ۴ تا ۶ ماشین و ۲ تا ۳ مسیر برای هر کار می‌باشند. در مسائل با ابعاد بزرگ، L1-L10، تعداد ۷ تا ۹ کار و تعداد ۷ تا ۸ ماشین و هر کار با ۲ تا ۴ مسیر برای پردازش طراحی شده است.

تمامی ۳۰ مسئله طراحی شده در [۳۳]، با استفاده از روش مدل‌سازی مطرح شده در قسمت ۲، مدل‌سازی و با استفاده از نرم افزار بهینه‌سازی LINGO ۸ حل شده‌اند. لازم به ذکر است که در بعضی از مسائل با ابعاد بزرگ، با توجه به اینکه نرم افزار بهینه سازی LINGO برای پیدا کردن بهترین جواب به زمان بسیار زیادی نیاز داشت، زمان ۱۰۸۰۰ ثانیه به عنوان حداکثر زمان برای نرم افزار در نظر گرفته شده است. همچنین با استفاده از الگوریتم بهینه سازی گروه ذرات پیشنهادی، تمام ۳۰ مسئله حل و بهترین نتایج انتخاب بدست آمده است. الگوریتم پیشنهادی در محیط MATLAB2008 کد شده و بر روی یک دستگاه رایانه با مشخصات (CPU)Pentium IV، 1.8GHZ، (RAM) 1GB، اجرا شده است.

در ادامه، در قسمت ۵-۱، یکی از مسائل بزرگ، مسئله L6، مورد بررسی قرار خواهد گرفت. مشخصات کامل این مسئله به همراه برنامه زمانبندی کارها، بدست آمده از الگوریتم پیشنهادی و با استفاده از نرم افزار بهینه سازی LINGO ۸، ارائه خواهد شد. سپس در قسمت ۵-۲، نتایج بدست آمده از حل کلیه ۳۰ مسئله مذکور به تفصیل آورده شده و مقایسات لازم صورت گرفته است.

۴. در صورتی که $LOOP_2 \leq REPEAT$ برقرار نباشد اجرای الگوریتم را متوقف کنید و در غیر این صورت به گام ۱۷ بروید. $REPEAT$ توسط کاربر تعیین می‌شود.

۵. مسیر فعلی تخصیص داده شده به هر کار را به رشته بیت‌های دودویی تبدیل کرده و در ماتریس X ذخیره کنید (ابعاد ماتریس X با ابعاد ماتریس v برابر است). بهترین مجموعه مسیر ذخیره شده در $PBEST$ را به رشته بیت‌های دودویی تبدیل و در ماتریس BP ذخیره کنید. با استفاده از رابطه (۲۱)، ماتریس سرعت را بروز کنید. با استفاده از رابطه (۲۲)، ماتریس X را بروز کنید. با تبدیل عناصر هر سطر ماتریس X به عددی در مبنای ۱۰، مسیر جدیدی برای هر کار به دست آورید. به گام ۱۸ بروید.

$$v_{id} = v_{id} + rand.Bin_{C_1}.(P_{id} - x_{id}) \quad (21)$$

$$x_{id} = \begin{cases} 1 & \rho_{id} < sigmoid(v_{id}) \\ 0 & other \end{cases} \quad (22)$$

که در آن Bin_{C_1} ، ثابت شتاب است و توسط کاربر تعیین می‌شود. $rand$ عدد تصادفی با توزیع یکنواخت در بازه [۰ ۱] می‌باشد. ۶. گام‌ها ۴-۱۶ را تکرار کنید.

۵. نتایج محاسباتی

مناسب‌ترین روش برای مقایسه و بررسی اثربخش بودن الگوریتم پیشنهادی به کار بردن آن برای حل تعدادی از مسائل مشهور که قبلاً توسط محققین حل شده‌اند، می‌باشد. از آنجایی که برای مسائل زمانبندی کارگاه‌های چند مسیره مثال‌های مشهوری یافت نشد، در این مقاله به منظور آزمایش عملکرد الگوریتم پیشنهادی، از مسائل طراحی شده در [۳۳] استفاده خواهیم کرد (جدول ۱). در [۳۳]، ۳۰ مسئله در ابعاد مختلف (۱۰ مسئله‌ی کوچک، ۱۰ مسئله‌ی متوسط و ۱۰ مسئله‌ی بزرگ) برای مسئله زمانبندی

جدول ۱. مشخصات کلی مسائل کارگاه‌های چند مسیره طراحی شده در [۳۳]

| تعداد کل مسیره‌ها | تعداد ماشین‌ها | تعداد کارها | تعداد مسائل بزرگ | تعداد مسیره‌ها | تعداد ماشین‌ها | تعداد کارها | تعداد مسائل متوسط | تعداد مسیره‌ها | تعداد ماشین‌ها | تعداد کارها | تعداد مسائل کوچک |
|-------------------|----------------|-------------|------------------|----------------|----------------|-------------|-------------------|----------------|----------------|-------------|------------------|
| ۱۸ | ۷ | ۸ | L1 | ۱۲ | ۶ | ۶ | M1 | ۴ | ۳ | ۲ | S1 |
| ۲۱ | ۶ | ۸ | L2 | ۱۲ | ۶ | ۶ | M2 | ۴ | ۳ | ۲ | S2 |
| ۲۰ | ۷ | ۸ | L3 | ۱۰ | ۶ | ۵ | M3 | ۴ | ۴ | ۲ | S3 |
| ۱۹ | ۷ | ۹ | L4 | ۱۲ | ۶ | ۶ | M4 | ۷ | ۴ | ۳ | S4 |
| ۱۸ | ۷ | ۸ | L5 | ۱۰ | ۴ | ۵ | M5 | ۴ | ۴ | ۲ | S5 |
| ۲۱ | ۷ | ۹ | L6 | ۱۰ | ۶ | ۵ | M6 | ۷ | ۳ | ۳ | S6 |
| ۱۸ | ۷ | ۹ | L7 | ۱۲ | ۶ | ۵ | M7 | ۴ | ۳ | ۲ | S7 |
| ۱۸ | ۶ | ۹ | L8 | ۱۰ | ۶ | ۵ | M8 | ۷ | ۳ | ۳ | S8 |
| ۱۸ | ۶ | ۸ | L9 | ۱۰ | ۶ | ۵ | M9 | ۴ | ۴ | ۲ | S9 |
| ۱۷ | ۷ | ۷ | L10 | ۱۲ | ۵ | ۵ | M10 | ۶ | ۳ | ۳ | S10 |

۱-۵. یک مثال عددی

دو مسیر قابل انجام هستند. در جدول ۲، برای هر یک از کارهای مذکور، ماشین‌های مورد نیاز در هر مسیر به همراه زمان مورد نیاز جهت هر عملیات آورده شده است.

ساختاری را در نظر بگیرید که شامل هفت ماشین و نه کار است. کارهای ۳ و ۹ به ترتیب از سه مسیر و چهار مسیر و مابقی کارها از

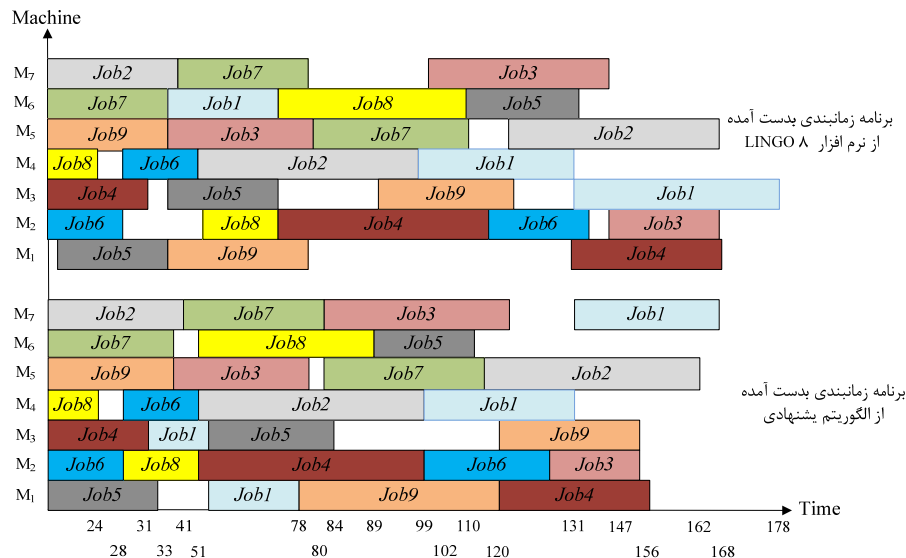
جدول ۲. مسیرهای مختلف، ماشینها و زمان مورد نیاز

| | | | | | | | | | |
|----------------|----------------|----------------|----------------|--------|----------------|----------------|----------------|----------------|--------|
| M ₂ | M ₄ | M ₂ | مسیر ۱ | کار | M ₃ | M ₄ | M ₆ | مسیر ۱ | کار ۱ |
| ۲۹ | ۲۳ | ۲۸ | | | ۴۵ | ۳۴ | ۳۷ | | |
| M ₄ | M ₅ | M ₃ | مسیر ۲ | ۶ | M ₇ | M ₄ | M ₁ | M ₃ | مسیر ۲ |
| ۲۲ | ۳۴ | ۳۱ | | | ۳۷ | ۳۲ | ۲۶ | ۲۱ | |
| M ₅ | M ₇ | M ₆ | مسیر ۱ | کار ۷ | M ₅ | M ₄ | M ₇ | مسیر ۱ | کار ۲ |
| ۳۶ | ۳۹ | ۳۷ | | | | ۴۵ | ۴۸ | ۴۱ | |
| M ₃ | M ₇ | M ₆ | M ₇ | مسیر ۲ | M ₅ | M ₄ | M ₃ | M ₁ | مسیر ۲ |
| ۲۵ | ۳۵ | ۲۲ | ۳۲ | | ۳۱ | ۳۵ | ۳۹ | ۳۰ | |
| M ₆ | M ₂ | M ₄ | مسیر ۱ | کار | M ₁ | M ₆ | M ₂ | مسیر ۱ | کار ۳ |
| ۳۸ | ۲۳ | ۲۴ | | | | ۴۹ | ۳۳ | ۳۶ | |
| M ₄ | M ₅ | M ₂ | M ₄ | مسیر ۲ | M ₄ | M ₆ | M ₂ | M ₄ | مسیر ۲ |
| ۱۸ | ۲۰ | ۲۵ | ۲۲ | ۸ | ۲۴ | ۲۲ | ۲۴ | ۲۹ | |
| | | | | | M ₂ | M ₇ | M ₅ | مسیر ۳ | |
| | | | | | ۲۱ | ۴۵ | ۴۳ | | |
| M ₃ | M ₁ | M ₅ | مسیر ۱ | کار | M ₆ | M ₃ | M ₅ | M ₂ | مسیر ۱ |
| ۳۲ | ۴۲ | ۳۸ | | | | ۲۵ | ۲۶ | ۳۷ | ۳۲ |
| M ₁ | M ₃ | M ₂ | مسیر ۲ | | M ₁ | M ₂ | M ₃ | مسیر ۲ | کار ۴ |
| ۳۵ | ۳۱ | ۴۱ | | | ۳۶ | ۴۸ | ۳۱ | | |
| M ₂ | M ₄ | M ₅ | M ₆ | مسیر ۳ | ۹ | M ₆ | M ₃ | M ₁ | مسیر ۱ |
| ۳۰ | ۳۱ | ۲۰ | ۲۸ | | | ۲۱ | ۳۲ | ۳۳ | کار ۵ |
| M ₁ | M ₂ | M ₆ | M ₇ | مسیر ۴ | | M ₄ | M ₅ | مسیر ۲ | |
| ۲۴ | ۲۶ | ۳۶ | ۲۶ | | | ۴۵ | ۴۲ | | |

تابع هدف، زمان اتمام کلیه کارها، برابر با ۱۶۸ می‌باشد. الگوریتم پیشنهادی برای کارهای ۱، ۲، ۵، ۶، ۷، ۸ و ۹ مسیر اول، برای کار ۱ و ۴، مسیر دوم و برای کار ۳ مسیر سوم را انتخاب می‌کند. از آنجایی که حل این مسئله توسط نرم افزار بهینه سازی ۸ LINGO نیازمند به زمان زیادی بود، با در نظر گرفتن سقف زمانی ۱۰۸۰۰ ثانیه معادل ۳ ساعت، بهترین مقدر به دست آمده برابر با ۱۷۸ بدست آمد (شکل ۲). نرم افزار بهینه سازی LINGO برای کارهای ۱، ۲، ۵، ۶، ۷، ۸، ۹ مسیر اول، برای کار ۴ مسیر دوم و برای کار ۳ مسیر سوم را انتخاب می‌کند. با مشاهده شکل ۲، دیده میشود که در حل این مسئله، استفاده از الگوریتم پیشنهادی موجب حصول جواب بهتر در زمانی بسیار کمتر در مقایسه با نرم افزار LINGO میگردد.

هدف، تعیین یک مسیر از بین مسیرهای موجود برای هر کار و سپس تعیین توالی انجام کارها به گونه‌ای که زمان اتمام کارها حداقل شود، می‌باشد. این مسئله توسط الگوریتم پیشنهادی حل و نتایج در شکل ۲ آورده شده است. همچنین، این مسئله، با استفاده از مدل برنامه ریزی غیر خطی عدد صحیح مختلط ارائه شده در این مقاله، مدل‌سازی و توسط نرم افزار بهینه سازی ۸ LINGO حل و نتایج حاصل در شکل ۲ آورده شده است. لازم به ذکر است که رایانه مورد استفاده مشابه با مشخصات رایانه مورد استفاده در [۳۳]، (Pentium IV, 1.8GHZ (CPU) ، 1GB (RAM)) بوده است.

حل این مسئله توسط الگوریتم پیشنهادی با صرف زمان ۳/۲۳۴۰ ثانیه به طول انجامید. همچنانکه در شکل ۲ دیده میشود مقدار



شکل ۲. برنامه زمانبندی کارها

۵-۲. نتایج به دست آمده از حل ۳۰ مسئله نمونه

تمامی ۳۰ مسئله طراحی شده در [۳۳]، به چهار روش، (۱) استفاده از الگوریتم پیشنهادی، (۲) استفاده از الگوریتم ایمنی مصنوعی پیشنهاد شده در [۳۳]، (۳) استفاده از نرم افزار بهینه سازی LINGO ۸ به روش مدل سازی پیشنهادی در این مقاله و (۴)

استفاده از نرم افزار بهینه سازی ۸ LINGO به روش مدل سازی پیشنهادی در [۳۳]، حل و نتایج در جدول ۳ آورده شده است. از آنجایی که نرم افزار بهینه سازی LINGO برای حل مسائل بزرگ به زمان زیادی نیاز دارد، سقف زمانی ۱۰۸۰۰ ثانیه، معادل ۳ ساعت، برای حل توسط نرم افزار در نظر گرفته شده است.

جدول ۳. مقایسه نتایج حاصل از الگوریتم ارائه شده با الگوریتم ایمنی مصنوعی و نرم افزار LINGO

| مسائل | مدل ریاضی ارائه شده در این مقاله، حل با LINGO | | | | مدل ریاضی ارائه شده در [۳۳]، حل با LINGO | | | | الگوریتم ایمنی مصنوعی | | الگوریتم پیشنهادی | |
|-------|--|---------------|-----------|------------|---|---------------|-----------|------------|-----------------------|-----------|-------------------|-----------|
| | تعداد متغیر | تعداد محدودیت | C_{max} | Time (sec) | تعداد متغیر | تعداد محدودیت | C_{max} | Time (sec) | C_{max} | Time(sec) | C_{max} | Time(sec) |
| S1 | ۵۱ | ۵۳ | ۱۱ | ۱ | ۲۰ | ۳۵ | ۱۱ | ۱ | ۱۱ | ۰/۷۹۹۳ | ۱۱ | ۰/۰۳۹۵ |
| S2 | ۴۶ | ۵۳ | ۸ | ۱ | ۲۰ | ۳۵ | ۸ | ۱ | ۸ | ۰/۷۹۳۱ | ۸ | ۰/۰۳۸۷ |
| S3 | ۵۳ | ۶۳ | ۱۸ | ۱ | ۲۶ | ۴۱ | ۱۸ | ۱ | ۱۸ | ۰/۷۰۰۵ | ۱۸ | ۰/۰۳۷۲ |
| S4 | ۸۳ | ۹۶ | ۱۲ | ۵ | ۴۳ | ۸۴ | ۱۲ | ۱ | ۱۲ | ۰/۶۹۵۷ | ۱۲ | ۰/۰۶۰۵ |
| S5 | ۴۹ | ۵۹ | ۱۸ | ۱ | ۲۲ | ۳۷ | ۱۸ | ۱ | ۱۸ | ۰/۷۲۰۴ | ۱۸ | ۰/۰۵۰۱ |
| S6 | ۶۳ | ۸۳ | ۱۰۹ | ۲ | ۳۳ | ۶۵ | ۱۰۹ | ۲ | ۱۰۹ | ۰/۷۲۴۶ | ۱۰۹ | ۰/۰۴۱۶ |
| S7 | ۳۹ | ۵۲ | ۱۳۱ | ۱ | ۱۹ | ۳۲ | ۱۳۱ | ۱ | ۱۳۱ | ۰/۶۵۲۹ | ۱۳۱ | ۰/۰۳۵۴ |
| S8 | ۱۰۲ | ۱۲۶ | ۳۳۷ | ۳۳ | ۴۳ | ۸۵ | ۳۳۷ | ۱۶ | ۳۳۷ | ۰/۷۵۶۱ | ۳۳۷ | ۰/۰۵۷۴ |
| S9 | ۳۸ | ۳۷ | ۴۹۳ | ۱ | ۱۹ | ۳۰ | ۴۹۳ | ۱ | ۴۹۳ | ۰/۶۶۴۴ | ۴۹۳ | ۰/۰۴۱۴ |
| S10 | ۸۰ | ۱۰۵ | ۲۲۷ | ۳ | ۵۳ | ۸۳ | ۲۲۷ | ۳ | ۲۲۷ | ۰/۸۲۵۳ | ۲۲۷ | ۰/۰۵۳۵ |
| AVE | ۶۰/۴ | ۷۲/۷ | ۱۳۶ | ۴/۹ | ۲۹/۸ | ۵۲/۷ | ۱۳۶ | ۲/۸ | ۱۳۶ | ۰/۷۳۲۶ | ۱۳۶ | ۰/۰۴۵۵۳ |
| M1 | ۲۲۱ | ۲۵۹ | ۵۴۷ | ۹۶ | ۱۲۸ | ۲۱۷ | ۵۴۷ | ۶۴ | ۵۵۰ | ۳/۵۴۶۷ | ۵۴۷ | ۱/۰۴۷۷ |
| M2 | ۱۶۲ | ۲۰۹ | ۳۶۱ | ۲۴۸ | ۱۴۹ | ۲۲۰ | ۳۶۱ | ۶۹۳ | ۳۶۱ | ۳/۴۰۵۲ | ۳۶۱ | ۱/۰۰۳۷ |
| M3 | ۱۳۸ | ۱۷۶ | ۳۳۶ | ۸۶ | ۱۰۹ | ۱۸۷ | ۳۳۶ | ۱۸ | ۳۳۶ | ۳/۲۲۹۵ | ۳۳۶ | ۰/۸۱۷۵ |
| M4 | ۲۵۲ | ۳۴۵ | ۳۷۸ | ۳۶۹ | ۱۹۷ | ۳۸۲ | ۳۷۸ | ۱۷۰ | ۳۷۸ | ۳/۸۲۴۱ | ۳۷۸ | ۱/۵۵۵ |
| M5 | ۱۶۵ | ۲۲۲ | ۹۰ | ۱۰۹ | ۱۱۰ | ۲۳۴ | ۹۰ | ۱۴۵ | ۹۰ | ۲/۸۵۹۴ | ۹۰ | ۰/۵۴۰۲ |
| M6 | ۱۹۳ | ۲۵۴ | ۴۳۶ | ۱۴۳ | ۱۳۴ | ۲۵۷ | ۴۳۶ | ۷۱ | ۴۳۶ | ۳/۲۶۰۰ | ۴۳۶ | ۱/۰۸۴۳ |

| | | | | | | | | | | | | |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|-------|--------|
| M7 | ۲۵۱ | ۳۳۸ | ۳۸۸ | ۶۱۴ | ۱۶۷ | ۳۴۱ | ۳۸۸ | ۱۶۵ | ۴۱۷ | ۳/۲۹۱۹ | ۳۸۸ | ۱/۱۲۱۵ |
| M8 | ۱۴۶ | ۱۸۴ | ۳۱۴ | ۲۲۱ | ۱۰۳ | ۱۷۸ | ۳۱۴ | ۴۵ | ۳۱۹ | ۲/۹۵۷۷ | ۳۱۹ | ۱/۰۲۸۷ |
| M9 | ۱۹۴ | ۲۵۷ | ۴۱۳ | ۵۰۱ | ۱۴۱ | ۲۶۰ | ۴۱۳ | ۷۱۵ | ۴۱۸ | ۳/۰۹۵۳ | ۴۱۳ | ۱/۲۷۳ |
| M10 | ۲۳۷ | ۳۲۴ | ۹۱ | ۴۲۶ | ۱۶۶ | ۳۴۲ | ۹۱ | ۶۶ | ۹۲ | ۳/۱۹۴۲ | ۹۷ | ۰/۷۶۲۷ |
| AVE | ۱۹۵/۹ | ۲۵۶/۸ | ۳۳۵/۴ | ۲۸۱/۳ | ۱۴۰/۴ | ۲۶۱/۸ | ۳۳۵/۴ | ۲۱۵/۲ | ۳۳۹/۷ | ۳/۲۶۶۴ | ۳۳۷/۵ | ۱/۰۲۳۴ |
| L1 | ۴۴۱ | ۶۲۹ | ۲۳۳ | ۱۰۸۰۰ | ۳۶۹ | ۶۹۲ | ۱۷۲ | ۱۰۸۰۰ | ۱۵۹ | ۴/۰۷۲ | ۱۵۷ | ۲/۷۱۶۳ |
| L2 | ۶۳۳ | ۹۹۸ | ۲۵۷ | ۱۰۸۰۰ | ۴۳۰ | ۱۰۴۹ | ۱۸۸ | ۱۰۸۰۰ | ۱۹۱ | ۴/۲۸۶۱ | ۱۸۲ | ۲/۳۰۵۷ |
| L3 | ۴۶۵ | ۶۷۷ | ۱۵۸ | ۱۰۸۰۰ | ۳۴۲ | ۷۲۰ | ۱۴۴ | ۱۰۸۰۰ | ۱۴۷ | ۳/۹۸۹۷ | ۱۴۱ | ۲/۶۶۲۵ |
| L4 | ۴۸۰ | ۷۰۴ | ۲۲۲ | ۱۰۸۰۰ | ۴۳۲ | ۷۸۹ | ۲۰۸ | ۱۰۸۰۰ | ۲۲۳ | ۴/۶۰۵۶ | ۲۱۸ | ۲/۹۴۱۷ |
| L5 | ۵۰۵ | ۷۴۳ | ۲۷۳ | ۱۰۸۰۰ | ۴۲۰ | ۸۲۷ | ۲۲۵ | ۱۰۸۰۰ | ۲۲۸ | ۴/۴۱۰۲ | ۲۲۵ | ۲/۸۱۵۳ |
| L6 | ۵۳۰ | ۷۸۸ | ۳۹۷ | ۱۰۸۰۰ | ۳۸۶ | ۸۶۰ | ۱۷۸ | ۱۰۸۰۰ | ۱۷۹ | ۴/۴۰۱۵ | ۱۶۸ | ۳/۲۳۴۰ |
| L7 | ۴۴۲ | ۶۲۸ | ۲۱۱ | ۱۰۸۰۰ | ۴۵۱ | ۷۳۸ | ۱۹۱ | ۱۰۸۰۰ | ۲۰۸ | ۴/۷۳۳۰ | ۲۰۵ | ۲/۸۴۸۹ |
| L8 | ۴۱۳ | ۶۱۰ | ۱۹۱ | ۱۰۸۰۰ | ۳۵۳ | ۶۷۳ | ۱۶۲ | ۱۰۸۰۰ | ۱۸۰ | ۴/۲۸۲۹ | ۱۵۶ | ۲/۵۰۴۵ |
| L9 | ۴۷۷ | ۷۱۳ | ۱۷۲ | ۱۰۸۰۰ | ۴۱۶ | ۷۹۷ | ۱۷۶ | ۱۰۸۰۰ | ۱۸۳ | ۴/۲۴۰۸ | ۱۹۴ | ۲/۳۲۳۶ |
| L10 | ۴۱۶ | ۵۹۸ | ۱۲۶ | ۱۰۸۰۰ | ۳۱۴ | ۶۳۹ | ۱۲۵ | ۱۰۸۰۰ | ۱۲۸ | ۳/۹۴۵۱ | ۱۲۸ | ۲/۱۳۳ |
| AVE | ۴۸۰/۲ | ۷۰۸/۸ | ۲۲۴ | ۱۰۸۰۰ | ۳۹۱/۳ | ۷۷۸/۴ | ۱۷۶/۹ | ۱۰۸۰۰ | ۱۸۲/۶ | ۴/۲۹۶۷ | ۱۷۷/۴ | ۲/۶۴۸۵ |

مصرف شده توسط الگوریتم پیشنهادی، $۹۸/۳۷\%$ کمتر از متوسط زمان مصرف شده توسط LINGO و همچنین $۹۳/۷۸\%$ کمتر از متوسط زمان مصرفی توسط الگوریتم ایمنی مصنوعی می‌باشد. در مسائل با ابعاد متوسط، میانگین جواب به دست آمده توسط الگوریتم پیشنهادی، $۶/۲۲\%$ بدتر از میانگین جواب به دست آمده توسط LINGO است و $۰/۶۵\%$ بهتر از میانگین جواب به دست آمده توسط الگوریتم ایمنی مصنوعی می‌باشد. این در حالی است که متوسط زمان مصرف شده توسط الگوریتم پیشنهادی، $۹۹/۵۲\%$ کاهش نسبت به متوسط زمان مصرف شده توسط LINGO و $۶۸/۶۷\%$ کاهش نسبت به متوسط زمان مصرف شده توسط الگوریتم ایمنی مصنوعی داشته است. در مسائل با ابعاد بزرگ، هر دو روش، الگوریتم پیشنهادی و الگوریتم ایمنی مصنوعی، از نقطه نظر زمان و جواب بدست آمده از نرم‌افزار LINGO نتایج بهتری را داشته‌اند. معذک بطور متوسط، الگوریتم پیشنهادی جواب‌های بهتری را در زمان کمتری در مقایسه با الگوریتم ایمنی مصنوعی بدست آورده است.

۶. نتیجه گیری و پیشنهادات آتی

در این مقاله، مسئله زمانبندی کارگاه‌های چند مسیره، مورد بررسی قرار گرفت. مدل ریاضی این مسئله به صورت یک مدل برنامه‌ریزی غیر خطی عدد صحیح مختلط ارائه شد. سپس یک الگوریتم ابتکاری بر مبنای الگوریتم بهینه سازی گروه ذرات برای حل مسئله پیشنهاد گردید. الگوریتم پیشنهادی شامل دو بخش کلی است که تکرار پی در پی این دو بخش جواب‌ها را به سمت جواب‌های نسبتاً بهینه و مناسب هدایت می‌کند. در بخش نخست، با استفاده از الگوریتم بهینه سازی گروه ذرات دودویی، مسیرهای مناسب برای کارها و در نهایت جواب‌های اولیه تولید می‌شود.

مقایسه‌ی نتایج بدست آمده از حل مسائل با استفاده از نرم‌افزار LINGO نشان می‌دهد که نحوه مدل‌سازی پیشنهادی در این مقاله بطور متوسط موجب کاهش زمان مورد نیاز برای بدست آوردن جواب‌ها توسط نرم‌افزار بوده است. در مسائل کوچک و متوسط این زمان برای مسائل کوچک $۲/۸$ ثانیه و برای مسائل متوسط $۲۱۵/۲$ می‌باشد در حالیکه متوسط زمان صرف شده به روش مدل‌سازی در [۳۳]، در مسائل کوچک $۴/۹$ ثانیه و در مسائل متوسط $۲۸۱/۳$ ثانیه بوده است. از نقطه نظر کیفیت جواب‌ها، در مسائل کوچک و متوسط هر دو نوع مدل‌سازی نتیجه یکسانی داشته‌اند. بنابراین، در مسائل کوچک و متوسط، روش مدل‌سازی در این مقاله نسبت به مدل ارائه شده در [۳۳]، صرفاً باعث کاهش متوسط زمان رسیدن به جواب بهینه می‌گردد. در مسائل بزرگ، با توجه به سقف زمانی اعمال شده، مقایسه متوسط جواب‌های بدست آمده نشان می‌دهد که این متوسط با استفاده از روش مدل‌سازی در این مقاله، $۱۷۶/۹$ ، بهتر از متوسط جواب‌های بدست آمده با استفاده از روش مدل‌سازی در [۳۳]، یعنی ۲۲۴ می‌باشد. این اختلاف ناشی از این است که نرم‌افزار، به دلیل کمتر بودن متوسط تعداد متغیرها در روش مدل‌سازی در این مقاله، استفاده موثرتری را از زمان تخصیص داده شده برای جستجو داشته و لذا قادر به کسب جواب‌های بهتر بوده است.

همچنین، الگوریتم پیشنهادی بطور متوسط، در مقایسه با الگوریتم ایمنی مصنوعی در [۳۳]، قادر به تولید جواب‌های خوب و در اکثر موارد بهتر، در زمانی کمتر از زمان صرف شده توسط الگوریتم ایمنی مصنوعی، می‌باشد. در مسائل با ابعاد کوچک، میانگین جواب‌های به دست آمده توسط الگوریتم پیشنهادی با میانگین جواب‌های الگوریتم ایمنی مصنوعی و همچنین میانگین جواب‌های بدست آمده توسط LINGO برابر است. معذک متوسط زمان

است. در جدول ۱.۱، برای هر یک از دو کار مذکور، ماشین‌های مورد نیاز در هر مسیر به همراه زمان مورد نیاز جهت هر یک از عملیات آورده شده است.

جدول ۱.۱. مسیرهای مختلف، ماشینها و زمان مورد نیاز

| | | | | |
|-------|-------|-------|--------|-------|
| M_2 | M_3 | M_1 | مسیر ۱ | کار ۱ |
| ۱۳۰ | ۶۰ | ۱۷ | | |
| M_3 | M_2 | M_1 | مسیر ۲ | کار ۲ |
| ۸۰ | ۶۰ | ۱۷ | | |
| M_3 | M_2 | M_1 | مسیر ۱ | کار ۳ |
| ۷۰ | ۱۶۰ | ۳۰ | | |
| | M_3 | M_1 | مسیر ۲ | |
| | ۷۰ | ۱۸۰ | | |
| | M_3 | M_1 | مسیر ۱ | |
| | ۲۸۰ | ۵۰ | | |
| | M_2 | M_1 | مسیر ۲ | |
| | ۲۶۰ | ۵۰ | | |
| M_2 | M_3 | M_1 | مسیر ۳ | |
| ۹۰ | ۱۸۰ | ۵۰ | | |

این مسئله با توجه به روش مدل‌سازی ارائه شده در بخش ۲-۳، مدل‌سازی و توسط نرم افزار ۸ LINGO، حل شده است. در ادامه به چگونگی مدل‌سازی برای این مسئله پرداخته می‌شود.

تابع هدف، زمان اتمام تمام کارها می‌باشد که باید کمینه گردد. $Minimize C_{max}$

مجموعه نامعادلات (۱)، موجب می‌شود تا زمان اتمام عملیات آخر، از هر یک از کارها در هر مسیر، کمتر یا مساوی با زمان انجام کلیه کارها، C_{max} باشد.

$$C_{max} \geq F_{iN_{jk}jkSN_{jk}iN_{jk}} \forall j \\ = 1, 2, \dots, n; \forall k \\ = 1, 2, \dots, K_j \quad (1)$$

در رابطه‌ی (۱)، اندیس j ، نشان دهنده‌ی کارها و اندیس k ، بیانگر مسیرهایی است که برای هر کار در نظر گرفته شده است. iN_{jk} ، نشان دهنده‌ی ماشین‌هایی است که آخرین عملیات کار j ام در مسیر k ام را انجام می‌دهد. اندیس $S_{N_{jk}iN_{jk}}$ معرف تعداد دفعات ملاقات کار j ام با ماشین iN_{jk} (ماشینی که آخرین عملیات کار j ام در مسیر k ام را انجام می‌دهد)، در طول کل مسیر پردازش کار می‌باشد. $F_{iN_{jk}jkSN_{jk}iN_{jk}}$ ، متغیری است که زمان خروج کار j ام از آخرین عملیات خود در مسیر k ام را نشان می‌دهد. بنابراین تمامی نامعادلات مورد نیاز در این خصوص به صورت زیر نوشته می‌شود:

بخش دیگر، با استفاده از الگوریتم بهینه سازی گروه ذرات، جواب‌های اولیه، طی تکرارهای مختلف به سمت نقطه بهینه هدایت می‌شوند. از آنجا که الگوریتم بهینه سازی گروه ذرات برای مسائل با ماهیت پیوسته طراحی شده است، برای تبدیل جواب‌های مسئله زمانبندی، که ماهیت گسسته دارد، به جواب‌های مورد پذیرش الگوریتم و بالعکس، از روش به کارگیری مبنای فاکتوریل استفاده شد.

به منظور بررسی کیفیت مدل‌سازی ریاضی مسئله در این مقاله و همچنین عملکرد الگوریتم پیشنهادی، ۳۰ مسئله در ابعاد کوچک، متوسط و بزرگ، که قبلاً در مقالات دیگر مورد استفاده قرار گرفته بود، با مدل ریاضی ارائه شده در این مقاله، مدل‌سازی و توسط نرم افزار ۸ LINGO و همچنین با الگوریتم پیشنهادی حل و نتایج آن مورد مقایسه قرار گرفت. نتایج نشان می‌دهد که روش مدل‌سازی ریاضی مسئله ارائه شده در این مقاله، در مقایسه با روش مدل‌سازی ریاضی مسئله ارائه شده در [۳۳] از کارایی بهتری (از نقطه نظر متغیرهای مورد نیاز) دارد. همچنین الگوریتم پیشنهادی در مقایسه با الگوریتم ایمنی مصنوعی، قادر است جواب‌های بهتری را در زمان کمتری تولید نماید.

برای حل اغلب مسائل زمانبندی، بسیاری از فاکتورها و محدودیت‌های موجود در محیط‌های تولیدی در نظر گرفته نمی‌شوند. بنابراین نتایج به دست آمده ممکن است تا حدی به دور از واقعیت باشند. این فاکتورها شامل در نظر گرفتن زمان‌های آماده سازی وابسته به توالی برای ماشین آلات، زمان‌های دسترسی به کارها، امکان وجود موعدهای تحویل برای کارها، ظرفیت‌های محدود انبارهای بین ماشین آلات که منجر به بلوکه شدن قطعه روی ماشین می‌شوند و محدودیت‌های مربوط به خرابی ماشین آلات، هستند. لحاظ هر یک از این محدودیت‌ها در مسائل زمانبندی مختلف از جمله مسئله زمانبندی کارگاه‌های چند مسیره می‌تواند به عنوان یک پیشنهاد برای تحقیقات آتی در نظر گرفته شود.

از دیگر پیشنهاداتی که می‌توان به عنوان تحقیقات آتی برای مسئله زمانبندی کارگاه‌های چند مسیره در نظر گرفت، ارائه روشی برای محاسبه مقادیر بهینه پارامترهای الگوریتم پیشنهادی به منظور یافتن جواب‌های با کیفیت‌تر در مدت زمان قابل قبول، اضافه کردن فرض وجود تعداد بیشتر از یک واحد از هر نوع کار، ارائه یک الگوریتم فراابتکاری بهتر از الگوریتم پیشنهادی و در نظر گرفتن استراتژی‌های دیگر برای نگهداری و تعمیرات پیشگیرانه می‌باشند.

ضمیمه ۱

مثال: ساختاری را در نظر بگیرید که شامل سه ماشین و سه کار است. کارهای ۱ و ۲ از دو مسیر و کار ۳ از سه مسیر قابل انجام

$$\begin{aligned}
C_{max} &\geq F_{i_{N_{11}}11S_{N_{11}}i_{N_{11}}} \rightarrow C_{max} \geq F_{2111}; & C_{max} &\geq F_{i_{N_{12}}12S_{N_{12}}i_{N_{12}}} \rightarrow C_{max} \geq F_{3121} \\
C_{max} &\geq F_{i_{N_{12}}12S_{N_{12}}i_{N_{12}}} \rightarrow C_{max} \geq F_{3121}; & C_{max} &\geq F_{i_{N_{21}}21S_{N_{21}}i_{N_{21}}} \rightarrow C_{max} \geq F_{3211} \\
C_{max} &\geq F_{i_{N_{22}}22S_{N_{22}}i_{N_{22}}} \rightarrow C_{max} \geq F_{3221}; & C_{max} &\geq F_{i_{N_{21}}21S_{N_{21}}i_{N_{21}}} \rightarrow C_{max} \geq F_{3211} \\
C_{max} &\geq F_{i_{N_{31}}31S_{N_{31}}i_{N_{31}}} \rightarrow C_{max} \geq F_{3311}; & C_{max} &\geq F_{i_{N_{32}}32S_{N_{32}}i_{N_{32}}} \rightarrow C_{max} \geq F_{2321} \\
C_{max} &\geq F_{i_{N_{33}}33S_{N_{33}}i_{N_{33}}} \rightarrow C_{max} \geq F_{2331}
\end{aligned}$$

مجموعه نامعادلات (۲) و (۳)، بیان می‌کنند که زمان اتمام پردازش کار توسط هر ماشین، لزوماً با زمان خروج کار از ماشین و شروع عملیات جدید کار روی ماشین دیگر برابر نخواهد بود. از طرفی مجموعه نامعادلات (۳) موجب می‌شوند تا روابط پیش‌نیازی بین فعالیت‌های هر کار رعایت گردد.

$$F_{i_1jkS_{1i_1}} \geq P_{i_1jkS_{1i_1}} \times Z_{kj} \quad \forall j = 1, 2, \dots, n; \forall k = 1, 2, \dots, K_j; S_{1i_1} = 1 \quad (۲)$$

$$\begin{aligned}
F_{i_p'jkS_{p'i_p'}} &\geq \left(F_{i_pjkS_{pi_p}} + P_{i_p'jkS_{p'i_p'}} \right) \times Z_{kj} & (۳) \\
\forall p &= 1, 2, \dots, N_{jk} - 1; p' = p + 1; \forall j = 1, 2, \dots, n; \forall k = 1, 2, \dots, K_j
\end{aligned}$$

می‌باشد. در نامعادلات (۳)، $P_{i_p'jkS_{p'i_p'}}$ زمان پردازش کار j ام در مسیر k ام هنگام اجرای p' عملیات را نشان می‌دهد. $F_{i_p'jkS_{p'i_p'}}$ متغیری است که زمان خروج کار j ام از p' عملیات خود در مسیر k ام را نشان می‌دهد. بنابراین مجموعه نامعادلات مورد نیاز بشرح زیر خواهد بود:

در نامعادله‌ی (۲)، رابطه‌ی $S_{1i_1} = 1$ با توجه به توضیحات گفته شده در متن کاملاً بدیهی به نظر می‌رسد. ماشینی که اولین عملیات کار j ام در مسیر k ام را انجام می‌دهد (ماشین i_1)، در طول اجرای اولین عملیات از مسیر پردازش کار، مشخصاً اولین بار است که کار j ام را ملاقات می‌کند ($S_{1i_1} = 1$). $P_{i_1jkS_{1i_1}}$ زمان پردازش کار j ام در مسیر k ام هنگام اجرای اولین عملیات را نشان می‌دهد. Z_{kj} متغیر صفر و یک برای تعیین مسیر هر کار

$$\begin{aligned}
F_{i_111S_{1i_1}} &\geq P_{i_111S_{1i_1}} \times Z_{11} \rightarrow F_{1111} \geq 17 \times Z_{11}; F_{i_211S_{2i_2}} \geq (F_{i_111S_{1i_1}} + P_{i_211S_{2i_2}}) \times Z_{11} \rightarrow F_{3111} \geq (F_{1111} + 60) \times Z_{11} \\
F_{i_311S_{3i_3}} &\geq (F_{i_211S_{2i_2}} + P_{i_311S_{3i_3}}) \times Z_{11} \rightarrow F_{2111} \geq (F_{3111} + 130) \times Z_{11} \\
F_{i_112S_{1i_1}} &\geq P_{i_112S_{1i_1}} \times Z_{21} \rightarrow F_{1121} \geq 17 \times Z_{21}; F_{i_212S_{2i_2}} \geq (F_{i_112S_{1i_1}} + P_{i_212S_{2i_2}}) \times Z_{21} \rightarrow F_{2121} \geq (F_{1121} + 60) \times Z_{21} \\
F_{i_312S_{3i_3}} &\geq (F_{i_212S_{2i_2}} + P_{i_312S_{3i_3}}) \times Z_{21} \rightarrow F_{2111} \geq (F_{3111} + 80) \times Z_{21} \\
F_{i_121S_{1i_1}} &\geq P_{i_121S_{1i_1}} \times Z_{12} \rightarrow F_{1211} \geq 30 \times Z_{12}; F_{i_221S_{2i_2}} \geq (F_{i_121S_{1i_1}} + P_{i_221S_{2i_2}}) \times Z_{12} \rightarrow F_{2211} \geq (F_{1211} + 160) \times Z_{12} \\
F_{i_321S_{3i_3}} &\geq (F_{i_221S_{2i_2}} + P_{i_321S_{3i_3}}) \times Z_{12} \rightarrow F_{3211} \geq (F_{2211} + 70) \times Z_{12} \\
F_{i_122S_{1i_1}} &\geq P_{i_122S_{1i_1}} \times Z_{22} \rightarrow F_{1221} \geq 180 \times Z_{22}; F_{i_222S_{2i_2}} \geq (F_{i_122S_{1i_1}} + P_{i_222S_{2i_2}}) \times Z_{22} \rightarrow F_{2211} \geq (F_{3221} + 70) \times Z_{22} \\
F_{i_131S_{1i_1}} &\geq P_{i_131S_{1i_1}} \times Z_{13} \rightarrow F_{1311} \geq 50 \times Z_{13}; F_{i_231S_{2i_2}} \geq (F_{i_131S_{1i_1}} + P_{i_231S_{2i_2}}) \times Z_{13} \rightarrow F_{3311} \geq (F_{1311} + 280) \times Z_{13} \\
F_{i_132S_{1i_1}} &\geq P_{i_132S_{1i_1}} \times Z_{23} \rightarrow F_{1321} \geq 50 \times Z_{23}; F_{i_232S_{2i_2}} \geq (F_{i_132S_{1i_1}} + P_{i_232S_{2i_2}}) \times Z_{23} \rightarrow F_{2321} \geq (F_{1321} + 260) \times Z_{23} \\
F_{i_321S_{3i_3}} &\geq (F_{i_221S_{2i_2}} + P_{i_321S_{3i_3}}) \times Z_{12} \rightarrow F_{3211} \geq (F_{2211} + 70) \times Z_{12} \\
F_{i_133S_{1i_1}} &\geq P_{i_133S_{1i_1}} \times Z_{33} \rightarrow F_{1331} \geq 50 \times Z_{33}; F_{i_233S_{2i_2}} \geq (F_{i_133S_{1i_1}} + P_{i_233S_{2i_2}}) \times Z_{33} \rightarrow F_{3331} \geq (F_{1331} + 180) \times Z_{33} \\
F_{i_333S_{3i_3}} &\geq (F_{i_233S_{2i_2}} + P_{i_333S_{3i_3}}) \times Z_{33} \rightarrow F_{2331} \geq (F_{3331} + 90) \times Z_{33}
\end{aligned}$$

مجموعه نامعادلات (۴)، تعیین کننده توالی کارها روی هر ماشین است:

$$\left\{ \begin{array}{l} F_{ijks} \geq F_{ij'k's'} + P_{ijks} + L \times (Z_{kj} + Z_{k'j'} + X_{ij's'js} - 3) \\ F_{ij'k's'} \geq F_{ijks} + P_{ijks} + L \times (Z_{ks} + Z_{k's'} + X_{ijsj's'} - 3) \\ X_{ij's'js} + X_{ijsj's'} = 1 \end{array} \right. \quad (۴)$$

$$\forall i = 1, 2, \dots, m; \forall k(k') \in \{1, 2, \dots, K_j(j')\}; \forall s(s') = 1, 2, \dots, S_{N_{jk(j'k')}} \\ j, j' \in \{1, 2, \dots, n\}; X_{ij's'js}, X_{ijsj's'} \in \{0, 1\}$$

زمان پردازش کار j ام روی ماشین i ام در مسیر k ام، وقتی S امین باری باشد که کار j ام ماشین i ام را ملاقات می‌کند، را نشان می‌دهد. F_{ijks} متغیر تصمیم‌گیری است که زمان خروج کار j ام از ماشین i ام در مسیر k ام، وقتی S امین باری باشد که کار j ام ماشین i ام را ملاقات می‌کند، را نشان می‌دهد. L ، یک مقدار ثابت و بزرگ است. در این مدل مقدار L را برابر ۱۰۰۰ در نظر می‌گیریم. بنابراین مجموعه معادلات و نامعادلات مورد نیاز به صورت زیر خواهد بود:

در مجموعه معادلات و نامعادلات (۴)، اندیس i ، نشان دهنده‌ی نوع ماشین است. اندیس‌های j, j' ، نشان دهنده‌ی کارهایی است که برای اجرای عملیات به ماشین i ، نیاز دارند. اندیس‌های k, k' ، نشان دهنده‌ی مسیره‌هایی است که کارهای j, j' در صورت انتخاب آن مسیره‌ها، ماشین i را ملاقات می‌کنند. اندیس‌های s, s' ، تعداد دفعات ملاقات کارهای j, j' با ماشین i را نشان می‌دهد. $X_{ij's'js}, X_{ijsj's'}$ متغیرهای صفر و یک هستند که تقدم و تأخر کارهای j, j' را نسبت به هم نشان می‌دهند. P_{ijks}

$$\begin{aligned} F_{1221} &\geq F_{1121} + P_{1221} + L \times (Z_{21} + Z_{22} + X_{11121} - 3) \rightarrow F_{1221} \geq F_{1121} + 180 + 1000 \times (Z_{21} + Z_{22} + X_{11121} - 3) \\ F_{1221} &\geq F_{1111} + P_{1221} + L \times (Z_{11} + Z_{22} + X_{11121} - 3) \rightarrow F_{1221} \geq F_{1111} + 180 + 1000 \times (Z_{11} + Z_{22} + X_{11121} - 3) \\ F_{1111} &\geq F_{1211} + P_{1111} + L \times (Z_{11} + Z_{12} + X_{12111} - 3) \rightarrow F_{1111} \geq F_{1211} + 17 + 1000 \times (Z_{11} + Z_{12} + X_{12111} - 3) \\ F_{1211} &\geq F_{1111} + P_{1211} + L \times (Z_{11} + Z_{12} + X_{11121} - 3) \rightarrow F_{1211} \geq F_{1111} + 30 + 1000 \times (Z_{11} + Z_{12} + X_{11121} - 3) \\ F_{1111} &\geq F_{1221} + P_{1111} + L \times (Z_{11} + Z_{22} + X_{12111} - 3) \rightarrow F_{1111} \geq F_{1221} + 17 + 1000 \times (Z_{11} + Z_{22} + X_{12111} - 3) \\ F_{1111} &\geq F_{1311} + P_{1111} + L \times (Z_{11} + Z_{13} + X_{13111} - 3) \rightarrow F_{1111} \geq F_{1311} + 17 + 1000 \times (Z_{11} + Z_{13} + X_{13111} - 3) \\ F_{1311} &\geq F_{1111} + P_{1311} + L \times (Z_{11} + Z_{13} + X_{13111} - 3) \rightarrow F_{1311} \geq F_{1111} + 50 + 1000 \times (Z_{11} + Z_{13} + X_{13111} - 3) \\ F_{1111} &\geq F_{1321} + P_{1111} + L \times (Z_{11} + Z_{23} + X_{13111} - 3) \rightarrow F_{1111} \geq F_{1321} + 17 + 1000 \times (Z_{11} + Z_{23} + X_{13111} - 3) \\ F_{1321} &\geq F_{1111} + P_{1321} + L \times (Z_{11} + Z_{23} + X_{11131} - 3) \rightarrow F_{1321} \geq F_{1111} + 50 + 1000 \times (Z_{11} + Z_{23} + X_{11131} - 3) \\ F_{1111} &\geq F_{1331} + P_{1111} + L \times (Z_{11} + Z_{33} + X_{13111} - 3) \rightarrow F_{1111} \geq F_{1331} + 17 + 1000 \times (Z_{11} + Z_{33} + X_{13111} - 3) \\ F_{1331} &\geq F_{1111} + P_{1331} + L \times (Z_{11} + Z_{33} + X_{11131} - 3) \rightarrow F_{1331} \geq F_{1111} + 50 + 1000 \times (Z_{11} + Z_{33} + X_{11131} - 3) \\ F_{1121} &\geq F_{1211} + P_{1121} + L \times (Z_{21} + Z_{12} + X_{12111} - 3) \rightarrow F_{1121} \geq F_{1211} + 17 + 1000 \times (Z_{21} + Z_{12} + X_{12111} - 3) \\ F_{1211} &\geq F_{1121} + P_{1211} + L \times (Z_{21} + Z_{12} + X_{11121} - 3) \rightarrow F_{1211} \geq F_{1121} + 30 + 1000 \times (Z_{21} + Z_{12} + X_{11121} - 3) \\ F_{1121} &\geq F_{1221} + P_{1121} + L \times (Z_{21} + Z_{22} + X_{12111} - 3) \rightarrow F_{1121} \geq F_{1221} + 17 + 1000 \times (Z_{21} + Z_{22} + X_{12111} - 3) \\ F_{1121} &\geq F_{1311} + P_{1121} + L \times (Z_{21} + Z_{13} + X_{13111} - 3) \rightarrow F_{1121} \geq F_{1311} + 17 + 1000 \times (Z_{21} + Z_{13} + X_{13111} - 3) \\ F_{1311} &\geq F_{1121} + P_{1311} + L \times (Z_{21} + Z_{13} + X_{13111} - 3) \rightarrow F_{1311} \geq F_{1121} + 50 + 1000 \times (Z_{21} + Z_{13} + X_{13111} - 3) \\ F_{1121} &\geq F_{1321} + P_{1121} + L \times (Z_{21} + Z_{23} + X_{13111} - 3) \rightarrow F_{1121} \geq F_{1321} + 17 + 1000 \times (Z_{21} + Z_{23} + X_{13111} - 3) \\ F_{1321} &\geq F_{1121} + P_{1321} + L \times (Z_{21} + Z_{23} + X_{11131} - 3) \rightarrow F_{1321} \geq F_{1121} + 50 + 1000 \times (Z_{21} + Z_{23} + X_{11131} - 3) \\ F_{1121} &\geq F_{1331} + P_{1121} + L \times (Z_{21} + Z_{33} + X_{13111} - 3) \rightarrow F_{1121} \geq F_{1331} + 17 + 1000 \times (Z_{21} + Z_{33} + X_{13111} - 3) \\ F_{1331} &\geq F_{1121} + P_{1331} + L \times (Z_{21} + Z_{33} + X_{11131} - 3) \rightarrow F_{1331} \geq F_{1121} + 50 + 1000 \times (Z_{21} + Z_{33} + X_{11131} - 3) \\ F_{1211} &\geq F_{1311} + 30 + 1000 \times (Z_{12} + Z_{13} + X_{13121} - 3); F_{1221} \geq F_{1311} + 180 + 1000 \times (Z_{22} + Z_{13} + X_{13121} - 3) \\ F_{1311} &\geq F_{1211} + 50 + 1000 \times (Z_{12} + Z_{13} + X_{12131} - 3); F_{1311} \geq F_{1221} + 50 + 1000 \times (Z_{22} + Z_{13} + X_{12131} - 3) \\ F_{1211} &\geq F_{1321} + 30 + 1000 \times (Z_{12} + Z_{23} + X_{13121} - 3); F_{1221} \geq F_{1321} + 180 + 1000 \times (Z_{22} + Z_{23} + X_{13121} - 3) \\ F_{1211} &\geq F_{1331} + 30 + 1000 \times (Z_{12} + Z_{33} + X_{13121} - 3); F_{1221} \geq F_{1331} + 180 + 1000 \times (Z_{22} + Z_{33} + X_{13121} - 3) \\ F_{1321} &\geq F_{1211} + 50 + 1000 \times (Z_{12} + Z_{23} + X_{12131} - 3); F_{1321} \geq F_{1221} + 50 + 1000 \times (Z_{22} + Z_{23} + X_{12131} - 3) \\ F_{1331} &\geq F_{1211} + 50 + 1000 \times (Z_{12} + Z_{33} + X_{12131} - 3); F_{1331} \geq F_{1221} + 50 + 1000 \times (Z_{22} + Z_{33} + X_{12131} - 3) \end{aligned}$$

$$\begin{aligned}
 &F_{2211} \geq F_{2111} + 160 + 1000 \times (Z_{11} + Z_{12} + X_{21121} - 3); F_{2211} \geq F_{2121} + 160 + 1000 \times (Z_{21} + Z_{12} + X_{21121} - 3) \\
 &F_{2321} \geq F_{2111} + 260 + 1000 \times (Z_{11} + Z_{23} + X_{21131} - 3); F_{2321} \geq F_{2121} + 260 + 1000 \times (Z_{21} + Z_{23} + X_{21131} - 3) \\
 &F_{2211} \geq F_{2321} + 160 + 1000 \times (Z_{12} + Z_{23} + X_{23121} - 3); F_{2321} \geq F_{2211} + 260 + 1000 \times (Z_{12} + Z_{23} + X_{22131} - 3) \\
 &F_{2111} \geq F_{2211} + 130 + 1000 \times (Z_{11} + Z_{12} + X_{22111} - 3); F_{2121} \geq F_{2211} + 60 + 1000 \times (Z_{21} + Z_{12} + X_{22111} - 3) \\
 &F_{2111} \geq F_{2321} + 130 + 1000 \times (Z_{11} + Z_{23} + X_{23111} - 3); F_{2121} \geq F_{2321} + 60 + 1000 \times (Z_{21} + Z_{23} + X_{23111} - 3) \\
 &F_{2111} \geq F_{2331} + 130 + 1000 \times (Z_{11} + Z_{33} + X_{23111} - 3); F_{2121} \geq F_{2331} + 60 + 1000 \times (Z_{21} + Z_{33} + X_{23111} - 3) \\
 &F_{2211} \geq F_{2331} + 160 + 1000 \times (Z_{12} + Z_{33} + X_{23121} - 3); F_{2331} \geq F_{2211} + 90 + 1000 \times (Z_{12} + Z_{33} + X_{22131} - 3) \\
 &F_{2331} \geq F_{2111} + 90 + 1000 \times (Z_{11} + Z_{33} + X_{21131} - 3); F_{2331} \geq F_{2121} + 90 + 1000 \times (Z_{21} + Z_{33} + X_{21131} - 3) \\
 &F_{3311} \geq F_{3111} + 280 + 1000 \times (Z_{11} + Z_{13} + X_{31131} - 3); F_{3311} \geq F_{3121} + 280 + 1000 \times (Z_{21} + Z_{13} + X_{31131} - 3) \\
 &F_{3331} \geq F_{3111} + 180 + 1000 \times (Z_{11} + Z_{33} + X_{31131} - 3); F_{3331} \geq F_{3121} + 180 + 1000 \times (Z_{21} + Z_{33} + X_{31131} - 3) \\
 &F_{3311} \geq F_{3211} + 280 + 1000 \times (Z_{12} + Z_{13} + X_{32131} - 3); F_{3311} \geq F_{3221} + 280 + 1000 \times (Z_{22} + Z_{13} + X_{32131} - 3) \\
 &F_{3331} \geq F_{3211} + 180 + 1000 \times (Z_{12} + Z_{33} + X_{32131} - 3); F_{3331} \geq F_{3221} + 180 + 1000 \times (Z_{22} + Z_{33} + X_{32131} - 3) \\
 &F_{3111} \geq F_{3211} + 60 + 1000 \times (Z_{11} + Z_{12} + X_{32111} - 3); F_{3121} \geq F_{3211} + 80 + 1000 \times (Z_{21} + Z_{12} + X_{32111} - 3) \\
 &F_{3211} \geq F_{3111} + 70 + 1000 \times (Z_{11} + Z_{12} + X_{31121} - 3); F_{3211} \geq F_{3121} + 70 + 1000 \times (Z_{21} + Z_{12} + X_{31121} - 3) \\
 &F_{3111} \geq F_{3221} + 60 + 1000 \times (Z_{11} + Z_{22} + X_{32111} - 3); F_{3121} \geq F_{3221} + 80 + 1000 \times (Z_{21} + Z_{22} + X_{32111} - 3) \\
 &F_{3221} \geq F_{3111} + 70 + 1000 \times (Z_{11} + Z_{22} + X_{31121} - 3); F_{3221} \geq F_{3121} + 70 + 1000 \times (Z_{21} + Z_{22} + X_{31121} - 3) \\
 &F_{3111} \geq F_{3311} + 60 + 1000 \times (Z_{11} + Z_{13} + X_{33111} - 3); F_{3121} \geq F_{3311} + 80 + 1000 \times (Z_{21} + Z_{13} + X_{33111} - 3) \\
 &F_{3211} \geq F_{3311} + 70 + 1000 \times (Z_{12} + Z_{13} + X_{33121} - 3); F_{3221} \geq F_{3311} + 70 + 1000 \times (Z_{22} + Z_{13} + X_{33121} - 3) \\
 &F_{3111} \geq F_{3331} + 60 + 1000 \times (Z_{11} + Z_{33} + X_{33111} - 3); F_{3121} \geq F_{3331} + 80 + 1000 \times (Z_{21} + Z_{33} + X_{33111} - 3) \\
 &F_{3211} \geq F_{3331} + 70 + 1000 \times (Z_{12} + Z_{33} + X_{33121} - 3); F_{3221} \geq F_{3331} + 70 + 1000 \times (Z_{22} + Z_{33} + X_{33121} - 3) \\
 &X_{13111} + X_{11131} = 1; X_{12111} + X_{11121} = 1; X_{13121} + X_{12131} = 1 \\
 &X_{23111} + X_{21131} = 1; X_{22111} + X_{21121} = 1; X_{23121} + X_{22131} = 1 \\
 &X_{32111} + X_{31121} = 1; X_{33111} + X_{31131} = 1
 \end{aligned}$$

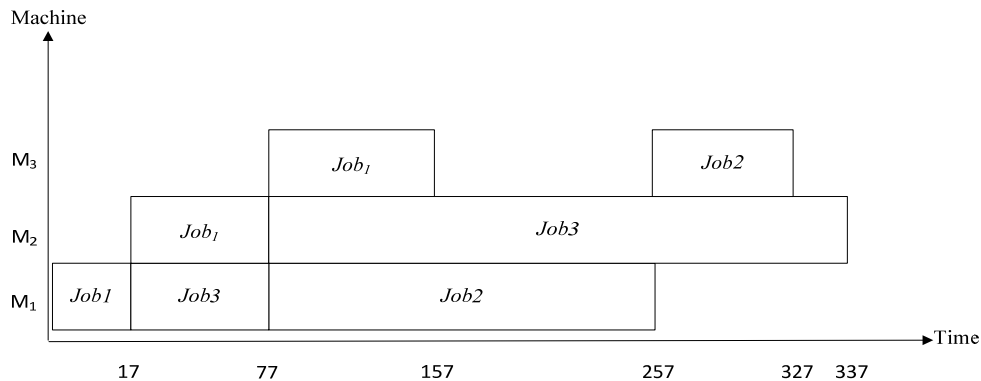
مجموعه معادلات (۵)، صرفاً یک مسیر از بین مسیرهای مختلف را به هر کار تخصیص می‌دهد.

$$\sum_k Z_{kj} = 1 \quad \forall j = 1, 2, \dots, n; \quad \forall k = 1, 2, \dots, K_j; \quad Z_{kj} \in \{0, 1\} \quad (5)$$

و لذا خواهیم داشت:

$$Z_{11} + Z_{21} = 1; \quad Z_{12} + Z_{22} = 1; \quad Z_{13} + Z_{23} + Z_{33} = 1$$

مدل فوق توسط نرم افزار ۸ LINGO با صرف زمان ۱۶ ثانیه حل و جواب بهینه در شکل ۱.۱ نشان داده شده است.



شکل ۱.۱. برنامه زمانبندی کارها

- and Computation, (2007), Vol. 187, No. 2, pp. 989-998.
- [11] Huang, K., Liao, C. Ant colony optimization combined with taboo search for the job shop scheduling problem. *Computers & Operations Research*, (2008), Vol. 35, No. 4, pp. 1030-1046.
- [12] Zhang, R., Wu, C. A simulated annealing algorithm based on block properties for the job shop scheduling problem with total weighted tardiness objective. *Computers & Operations Research*, (2011), Vol. 38, No. 5, pp. 854-867.
- [13] Wang, L., Zheng, D. An effective hybrid optimization strategy for job-shop scheduling problems. *Computers & Operations Research*, (2001), Vol. 28, No. 6, pp. 585-596.
- [14] Elmi, A., Solimanpur, M., Topaloglu, S., Elmi, A. A simulated annealing algorithm for the job shop cell scheduling problem with intercellular moves and reentrant parts. *Computers & Industrial Engineering*, (2011), Vol. 61, No. 1, pp. 171-178.
- [15] Pezzella, F., Merelli, E. A tabu search method guided by shifting bottleneck for the job shop scheduling problem. *European Journal of Operational Research*, (2000), Vol. 120, No. 2, pp. 297-310.
- [16] Wenqi, H., Aihua, Y. An improved shifting bottleneck procedure for the job shop scheduling problem. *Computers & Operations Research*, (2004), Vol. 31, No. 12, pp. 2093-2110.
- [17] Rego, C., Duarte, R. A filter-and-fan approach to the job shop scheduling problem. *European Journal of Operational Research*, (2009), Vol. 194, No. 3, pp. 650-662.
- [18] Sha, D.Y., Hsu, C. A hybrid particle swarm optimization for job shop scheduling problem. *Computers & Industrial Engineering*, (2006), Vol. 51, No. 4, pp. 791-808.
- [۱۹] بهروزی، مهدی؛ عشقی، کوروش. مدل‌سازی و حل مسئله زمانبندی کارگاه‌های با زمان‌های آماده سازی وابسته به توالی. نشریه بین المللی مهندس صنایع و مدیریت تولید، ۱۳۸۹، جلد ۲۱، شماره ۴.
- [20] Lin, T.L., Horng, S.J., Kao, T.W., Chen, Y.H., Run, R.S. and Chen, R.J. An efficient job shop scheduling algorithm based on particle swarm optimization. *Expert Systems with Applications*, (2010), Vol. 37, No. 3, pp. 2629-2636.
- [21] Kasemset, C. and Kachitvichyanukul, V. A PSO-based procedure for a bi-level multiobjective TOC-based job-shop scheduling problem. *International*
- بی‌نوشت**
1. Job Shop Scheduling
 2. Mixed Integer Nonlinear Programming
 3. Multiple Route
 4. Particle Swarm Optimization
 5. Make span
 6. Scheduling
- مراجع**
- [1] Pinedo, M. *Operations scheduling with applications in manufacturing and services*. Springer, New York (2005).
 - [2] Srinivasan, G., Srirangacharyulu, B. Minimizing variance of job completion times in a single machine. *International Journal of Operational Research*, (2012), Vol. 13, No. 1, pp. 110-127.
 - [3] Garey, M.R., Johnson, D.S., Sethi, R. The complexity of flowshop and job shop scheduling. *Mathematics of Operations Research*, (1976), Vol. 1, No. 2, pp. 117-129.
 - [4] Pan, J.C.H., Huang, H.C. A hybrid genetic algorithm for no-wait job shop scheduling problems. *Expert Systems with Applications*, (2009), Vol. 36, No. 3, pp. 5800-5806.
 - [5] Goncalves, J.F., Mendes, J.J.M., Resende, M.G.C. A hybrid genetic algorithm for the job shop scheduling problem. *European Journal of Operational Research*, (2005), Vol. 167, No. 1, pp. 77-95.
 - [6] Maqsood, S., Noor, S., Khan, M.K., Wood, A. Hybrid genetic algorithm (GA) for job shop scheduling problems and its sensitivity analysis. *International Journal of Intelligent Systems Technologies and Applications*, (2012), Vol. 11, No. 1, pp. 49-62.
 - [7] Hurink, J., Knust, S. Tabu search algorithms for job-shop problems with a single transport robot. *European Journal of Operational Research*, (2005), Vol. 162, No. 1, pp. 99-111.
 - [8] Zhang, C.Y., Li, P., Guan, Z., Rao, Y.Q. A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem. *Computers & Operations Research*, (2007), Vol. 34, No. 11, pp. 3229-3242.
 - [9] Nasiri, M.M., Kianfar, F. A GES/TS algorithm for the job shop scheduling. *Computers and Industrial Engineering*, (2012), Vol. 62, No. 4, pp. 946-952.
 - [10] Heinonen, J., Pettersson, F. Hybrid ant colony optimization and visibility studies applied to a job-shop scheduling problem. *Applied Mathematics*

- [31] Rahmati, S.H.A., Zandieh, M., Yazdani, M. Developing two multi-objective evolutionary algorithms for the multi-objective flexible job shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, (2013), Vol. 64, Nos. 5-8, pp. 915-932.
- [32] Wang, L., Zhou, G., Wang, S., Liu, M. An effective artificial bee colony algorithm for the flexible job-shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, (2012), Vol. 60, Nos. 1-4, pp. 303-315.
- [33] Golmakani, H.R., Namazi, A. An artificial immune algorithm for multiple-route job shop scheduling problem. *International Journal of Advanced Manufacturing Technology*, (2012), Vol. 63, Nos. 1-4, pp. 77-86.
- [34] Kennedy J., Eberhart R. Particle swarm optimization. *Proceedings of the IEEE, international conference on neural networks*, (1995), Vol. 4, pp. 1942-1948.
- [35] Eberhart, R.C. and Kennedy, J. A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, IEEE Service Center, Piscataway, (1995), pp. 39-43.
- [36] Li, J., Pan, Q., Xie, S., Jia, B., Wang, Y. A hybrid particle swarm optimization and tabu search algorithm for flexible job-shop scheduling problem. *International Journal of Computer Theory and Engineering*, (2010), Vol. 2, No. 2, pp. 1793-1820.
- [37] Song, X. Hybrid particle swarm algorithm for job shop scheduling problems. *School of Information and Control Engineering, Shenyang Jianzhu University*(2010).
- [38] [۳۸] بهروزی، مهدی؛ عشقی، کوروش. بکارگیری الگوریتم ترکیبی بهینه‌سازی دسته ذرات برای حل مسئله زمانبندی کارگاه‌های، نشریه بین‌المللی مهندسی صنایع و مدیریت تولید، ۱۳۸۸، جلد ۲۰، شماره ۲.
- [39] Golmakani, H.R., Fazel, M. Constrained portfolio selection using particle swarm optimization. *Expert Systems with Applications*, (2011), Vol. 38, No. 7, pp. 8327-8335.
- [40] Poli, R., Kennedy, J., Blackwell, T. Particle swarm optimization: An Overview. *Swarm Intelligence Journal*, (2007), Vol. 1, No. 1, pp. 33-57.
- [41] Kennedy J., Eberhart R.C. A discrete binary version of the particle swarm algorithm. *IEEE Journal of Operational Research*, (2012), Vol. 14, No. 1, pp. 50-69.
- [22] AitZai, A. and Boudhar, M. Parallel branch and bound and parallel PSO algorithms for job shop scheduling problem with blocking. *International Journal of Operational Research*, (2013), Vol. 16, No. 1, pp. 14-37.
- [23] Xie, S.M. A new genetic algorithms combined with learning strategy for flexible job-shop scheduling problem. *Advances in Intelligent and Soft Computing*, (2012), Vol. 117, pp. 153-161.
- [24] Gao, J., Gen, M., Sun, L., Zhao, X. A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems. *Computers & Industrial Engineering*, (2007), Vol. 53, No. 1, pp. 149-162.
- [25] Shao, X., Liu, W., Liu, Q., Zhang, C. Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, DOI.10.1007/s00170-012-4701-3(2013).
- [26] Li, J., Pan, Q., Liang, Y. An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering*, (2010), Vol. 59, No. 4, pp. 647-662.
- [27] Bagheri, A., Zandieh, M., Mahdavi, I., Yazdani, M. An artificial immune algorithm for the flexible job-shop scheduling problem. *Future Generation Computer Systems*, (2010), Vol. 26, No. 4, pp. 533-541.
- [28] Rahmati, S.H.A., Zandieh, M. A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, (2012), Vol. 58, Nos. 9-12, pp. 1115-1129.
- [29] Wang, Y.M., Yin, H.L., Qin, K.D. A novel genetic algorithm for flexible job shop scheduling problems with machine disruptions. *The International Journal of Advanced Manufacturing Technology*, DOI. 10.1007/S00170.013-4923-z(2013).
- [30] Li, J.Q., Pan, Q.K., Suganthan, P.N., Chua, T.J. A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, (2011), Vol. 52, Nos. 5-8, pp. 683-697.

-
- [43] Tasgetiren, M.F. and Liang, Y.C. A binary particle swarm optimization algorithm for lot sizing problem. *Journal of Economic and Social Research*, (2004), Vol. 5, No. 2, pp.1-20.
- [42] Knuth, D. *Semi numerical algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA(1997).
- International Conference on Systems, Man, and Cybernetics, (1997), Vol. 5, pp. 4104-4108.