

Presenting an Integrated Model for Production Planning and Preventive Maintenance Scheduling Considering Uncertainty of Parameters and Disruption of Facilities

Fariba Salahi^{*}, Amir Daneshvar^{}, Mehdi Homayounfar^{***},
Adel Pourghader Chobar^{****}**

Abstract

The scheduling of parallel machines and preventive maintenance is one of the key issues in the field of production processes, and has always been a topic of interest for researchers. This research aims to design an integrated model for scheduling and preventive maintenance for parallel machines considering the probability of disruption in facilities and uncertainty in parameters of the model. In this regard, a mathematical scheduling model has been proposed with two objective functions of minimizing the weighted completion time of products and maximizing the reliability of the production line. The NP-hard nature of the studied problem from a computational perspective, meta-heuristic algorithms such as NSGA-II and MOPSO were utilized to solve numerical problems in medium and large scales. Therefore, numerical problems were designed in different size and solved by the proposed algorithms. The results showed that the NSGA-II compared to the MOPSO algorithm provide better solutions. However, MOPSO has better efficiency than NSGA-II in term of computation time, this superiority is not considerable and it can not be considered as a definitive basis for comparing two algorithms.

Keywords: Preventive Maintenance; Scheduling; Disruption; NSGA-II; MOPSO.

Received: Des. 19, 2021; Accepted: Oct. 17, 2022.

^{*} Assistant Professor of Industrial Management, Faculty of Management, Electronic Unit, Islamic Azad University, Tehran, Iran.

^{**} Assistant Professor of Information Technology Management, Faculty of Management, Electronic Unit, Islamic Azad University, Tehran, Iran (Corresponding Author).

Email: daneshvar.amir@gmail.com

^{***} Assistant Professor of Industrial Management, Faculty of Management and Accounting, Rasht Branch, Islamic Azad University, Rasht, Iran.

^{****} Ph.D, Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.

چشم‌انداز مدیریت صنعتی

شاپای چاپی: ۹۸۷۴-۲۲۵۱، شاپای الکترونیکی: ۴۱۶۵-۲۶۴۵

سال سیزدهم، شماره ۴۹، بهار ۱۴۰۲، صص ۱۰۵ - ۱۳۹ (نوع مقاله: پژوهشی)

DOI: [10.52547/JIMP.13.1.105](https://doi.org/10.52547/JIMP.13.1.105)

ارائه مدل یکپارچه برنامه‌ریزی تولید و زمان‌بندی نگهداری و تعمیرات پیشگیرانه با در نظر گرفتن عدم قطعیت پارامترها و اختلال در تسهیلات

فریبا صلاحی*، امیر دانشور**، مهدی همایون فر***، عادل پورقادر چوبر****

چکیده

مسئله زمان‌بندی ماشین‌های موازی و نگهداری و تعمیرات پیشگیرانه این دسته از ماشین‌ها از جمله مسائل کلیدی در حوزه فرآیندهای تولیدی است که همواره مورد توجه پژوهشگران بوده است. این پژوهش به دنبال طراحی مدل یکپارچه‌ای برای زمان‌بندی تولید و برنامه‌ریزی نگهداری و تعمیرات ماشین‌های موازی با در نظر گرفتن احتمال اختلال در عملکرد تسهیلات و عدم قطعیت در پارامترهای مسئله است. در این راستا یک مدل برنامه‌ریزی ریاضی با دو هدف حداقل سازی زمان تکمیل وزنی محصولات و حداکثر سازی قابلیت اطمینان در خط تولید ارائه شده است. با توجه به ماهیت NP-hard مسئله مورد بررسی از جنبه محاسباتی، از الگوریتم‌های حل فراابتکاری NSGA-II و MOPSO به منظور حل مسائل عددی در ابعاد متوسط و بزرگ استفاده شده است. بر این اساس، مسائل عددی در ابعاد مختلف طراحی شده و از الگوریتم‌های مورد نظر به منظور حل این مسائل استفاده شد. نتایج نشان می‌دهند که الگوریتم NSGA-II در مقایسه با الگوریتم MOPSO جواب‌های مناسب‌تری را ارائه می‌کند. هرچند الگوریتم MOPSO نسبت به الگوریتم NSGA-II از نظر زمان حل مسئله از کارایی بیشتری برخوردار است، مقدار این برتری قابل ملاحظه نیست و نمی‌توان آن به عنوان مبنای قطعی مقایسه دو الگوریتم در نظر گرفت.

کلیدواژه‌ها: نگهداری و تعمیرات پیشگیرانه؛ زمان‌بندی؛ اختلال؛ NSGA-II؛ MOPSO.

تاریخ دریافت مقاله: ۱۴۰۰/۰۹/۲۸، تاریخ پذیرش مقاله: ۱۴۰۱/۰۷/۲۵.

* استادیار مدیریت صنعتی، دانشکده مدیریت، واحد الکترونیکی، دانشگاه آزاد اسلامی، تهران، ایران.

** استادیار مدیریت فناوری اطلاعات، دانشکده مدیریت، واحد الکترونیکی، دانشگاه آزاد اسلامی، تهران، ایران (نویسنده مسئول).

Email: daneshvar.amir@gmail.com

*** استادیار مدیریت صنعتی، دانشکده مدیریت و حسابداری، واحد رشت، دانشگاه آزاد اسلامی، رشت، ایران.

**** دانش‌آموخته دکتری، گروه مهندسی صنایع، دانشکده مهندسی صنایع و مکانیک، واحد قزوین، دانشگاه آزاد اسلامی، قزوین، ایران.

۱. مقدمه

نتیجه یک برنامه زمان‌بندی و توالی فعالیت‌های بهینه این خواهد بود که کالاها در موعد مقرر و با کیفیت مطلوب به مشتری تحویل داده شود و در نتیجه بهره‌وری شرکت بهبود یابد. در واقع حتی در شرایط مشابهت فناوری مورد استفاده و قطعه‌های یکسان ماشین‌آلات موازی، مواردی نظیر سن، مقدار کارکرد و استهلاک باعث می‌شوند که کارکرد آن‌ها کاملاً با هم یکسان نباشد؛ از این رو در برنامه‌ریزی ماشین‌آلات موازی باید به این نکات توجه لازم صورت گیرد. به همین دلیل نرخ تولید و هزینه تمام‌شده انجام فعالیت‌ها، متناسب با ماشینی است که فعالیت مورد نظر را اجرا می‌کند [۲۱]. با توسعه و پیشرفت فناوری و افزایش تعداد ماشین‌آلات در سیستم‌های تولیدی، مدیران در جست‌وجوی راهکارهایی برای کاهش خرابی‌های ناگهانی ماشین‌آلات و افزایش دوره کارایی آن‌ها هستند تا با پرداخت کمترین هزینه بتوانند کارایی حداکثری را از این ماشین‌آلات کسب کنند. نگهداری و تعمیرات از ابزارهای استراتژیک در این زمینه است که به شرکت‌های تولیدی برای حضور پر قدرت در فضای رقابتی تولید کمک می‌کند [۱]؛ همچنین به دلیل عدم زمان‌بندی صحیح فعالیت‌های نگهداری و تعمیرات موجب می‌شود که ماشین‌آلات در انتظار انجام بازرسی و نگهداری و تعمیرات باشند که بر این اساس برخی از سفارش‌ها در صف انتظار برای پردازش توسط این ماشین‌آلات قرار می‌گیرند [۲۵].

نگاه جزیره‌ای و جداگانه به مسائل زمان‌بندی تولید و نگهداری و تعمیرات باعث ازدست‌رفتن کیفیت، اعتبار و غیر کاربردی بودن خروجی‌های حاصل از مدل می‌شود؛ چراکه مباحث زمان‌بندی تولید و نگهداری و تعمیرات ارتباط تنگاتنگی با یکدیگر دارد و نمی‌توان آن‌ها را به صورت مجزا بررسی کرد [۳]. هدف از مسئله نگهداری و تعمیرات، ارائه یک برنامه زمان‌بندی است که فعالیت‌های کاری حتی‌الامکان نزدیک به زمان تعیین‌شده خود انجام شوند و انحراف کل از برنامه «ایده‌آل» تعریف‌شده بر اساس تاریخ‌های سررسید حداقل شود. مسائل زمان‌بندی تعمیرات و نگهداری، به دلیل اهمیت بالایی آن در عملکرد سازمانی به تعدد در مبانی نظری بررسی شده است [۱۷].

با وجود پژوهش‌های صورت‌گرفته در مبانی نظری موضوع، وجود برخی از شکاف‌های پژوهشی ضرورت مطالعه بیشتر در این حوزه را بیش‌ازپیش نمایان می‌سازد. از جمله شکاف‌های پژوهشی این است که در بیشتر پژوهش‌های صورت‌گرفته، صرفاً به یکی از مباحث زمان‌بندی تولید یا نگهداری و تعمیرات پرداخته شده است که به دلیل درهم‌تنیدگی این مفاهیم در حوزه مدیریت عملیات، نتایج به‌دست‌آمده از این نگرش جزیره‌ای در دنیای واقعی چندان کاربردی نیست؛ بنابراین ضرورت انجام پژوهش‌هایی که به دو مقوله زمان‌بندی تولید و نگهداری و تعمیرات به صورت هم‌زمان بپردازند، به وضوح احساس می‌شود. علاوه بر این در نظر گرفتن محدودیت‌ها و مفروضاتی که مسئله پژوهش را در تطابق هر چه بیشتر با شرایط دنیای واقعی

قرار داده و مدلی ارائه کند که از قابلیت کاربرد در شرایط واقعی برخوردار باشد، حائز اهمیت است. مباحثی نظیر احتمال اختلال در تسهیلات و عدم قطعیت در پارامترها از جمله مسائلی موجود در دنیای واقعی هستند که نادیده گرفتن آنها می‌تواند مدل ارائه‌شده را از شرایط دنیای واقعی دور سازد.

با توجه به دلایل بالا، هدف این پژوهش، طراحی مدلی یکپارچه به‌منظور زمان‌بندی و برنامه‌ریزی نگهداری و تعمیرات ماشین‌های موازی است. بدین منظور با طراحی یک مدل ریاضی برای زمانی که چند ماشین به‌صورت موازی قرار دارند، زمان‌بندی بهینه انجام فعالیت‌ها، تخصیص فعالیت‌ها به ماشین‌آلات و همچنین برنامه‌ریزی نگهداری و تعمیرات ماشین‌آلات انجام می‌شود. هدف کلی این پژوهش، برنامه‌ریزی زمان‌بندی و نگهداری و تعمیرات ماشین‌های موازی انعطاف‌پذیر برای حداقل‌سازی حداکثر زمان تکمیل فعالیت‌ها است. ضمن اینکه احتمال اختلال در عملکرد تسهیلات و عدم قطعیت در پارامترهای مسئله نیز به‌منظور افزایش اعتبار مدل و کاربردی‌تر شدن نتایج مدل در مدل‌سازی موردتوجه قرار گرفته است. درنهایت در این پژوهش از یک الگوریتم فراابتکاری و جدید برای حل مسائل عددی در ابعاد مختلف استفاده شده است.

۲. مبانی نظری و پیشینه پژوهش

در رابطه با زمان‌بندی تولید و تعمیرات و نگهداری پژوهش‌های بسیاری صورت گرفته است. فیتوچی و همکاران^۱ (۲۰۱۴) به ترکیب برنامه‌ریزی تولید و زمان‌بندی نت پیشگیرانه برای ماشین‌آلات چندحالتی پرداختند. در مدل آنها حداقل‌سازی هزینه‌های راه‌اندازی، تعمیرات اصلاحی، نگهداری، سفارش تحویل داده نشده و هزینه‌های عملیاتی موردتوجه قرار گرفت [۶]. رویز و ماروتو^۲ (۲۰۱۶)، همین مسئله را با رویکرد فراابتکاری مبتنی بر الگوریتم ژنتیک برای مسئله جریان کارگاهی هیبریدی با ماشین‌های موازی مستقل و با در نظر گرفتن زمان راه‌اندازی ماشین‌آلات طراحی کردند [۱۴]. یانگ و همکاران^۳ (۲۰۱۷)، یک سیاست نگهداری و تعمیرات پیشگیرانه جدید با در نظر گرفتن شوک‌های خارجی بر ماشین‌آلات ارائه کردند [۲۴]. لیو و همکاران^۴ (۲۰۱۷)، یک مدل برنامه‌ریزی یکپارچه نت و برنامه‌ریزی تولید را با لحاظ کردن هزینه‌ها و زمان‌های مربوط به آماده‌سازی مبتنی بر توالی ماشین‌ها ارائه دادند [۱۰]. شن و همکاران^۵ (۲۰۱۸)، به حل مسئله جریان کارگاهی انعطاف‌پذیر با لحاظ کردن زمان‌های آماده‌سازی وابسته به توالی پرداختند. این پژوهش زمان‌بندی و توالی فعالیت‌ها را در مسئله

1. Fitouhi, et al.
2. Ruiz & Maroto
3. Yang, et al.
4. Liu, et al.
5. Shen, et al.

جریان کارگاهی مختلط طوری برنامه‌ریزی می‌کند که در انتها مدت‌زمان انجام فعالیت‌ها حداقل شود [۱۹]. ساپر و همکاران^۱ (۲۰۱۹)، یک مدل زمان‌بندی ماشین‌های موازی با هدف حداقل-سازی مدت‌زمان انجام فعالیت‌ها ارائه کردند [۲۱]. پیچیدگی محاسباتی مدل با بهره‌گیری از روابطی تعیین و سپس حد پایین جواب بهینه مشخص شد. وانگ و همکاران^۲ (۲۰۱۸)، یک الگوریتم شاخه و قیمت برای برنامه‌ریزی فعالیت‌های یک ماشین در یک سیستم منعطف ارائه دادند. مسائل عددی در اندازه‌های مختلف، کارکرد رویکرد حل پیشنهادی آن‌ها را در شرایط مختلف تأیید کرد [۲۲]. امیری و همکاران (۲۰۱۸)، در پژوهشی به ارائه یک مدل یکپارچه برای برنامه‌ریزی زمان‌بندی و نت‌هاب‌های انرژی ارائه کردند. بدین منظور، مدلی یکپارچه برای زمان‌بندی و برنامه‌ریزی نت با در نظر گرفتن ریسک شکست‌های تصادفی تسهیلات در نظر گرفته شد و یک رویکرد سناریویی دومرحله‌ای برای در نظر گرفتن عدم قطعیت پارامترهای مدل به کار رفت [۳].

قلمی و گروسو^۳ (۲۰۱۹)، یک الگوریتم تقریبی موازی برای حل مسئله زمان‌بندی ماشین‌های موازی ارائه کردند. بدین منظور، مدل کلاسیک زمان‌بندی ماشین‌های موازی ارائه و سپس الگوریتم جدید تقریب موازی برای حل مسائل طراحی شد. نتایج حاکی از کاهش قابل توجه زمان حل با استفاده از رویکرد ارائه شده است [۷]. ناتف و همکاران^۴ (۲۰۱۹)، یک مدل برنامه‌ریزی ریاضی برای مسئله زمان‌بندی ماشین‌های موازی با محدودیت‌های زمانی تنظیم ماشین‌ها ارائه کردند. در این پژوهش فرض شد که فعالیت‌های مربوط به خانواده‌های مختلف باید روی ماشین‌های موازی انجام شود. مدل ارائه‌شده دارای دو هدف اصلی کمینه‌سازی زمان فعالیت‌ها و خانواده کارها برای ماشین‌ها است. پس از طراحی مدل، رویکردی فراابتکاری برای حل مسائل عددی در اندازه‌های متنوع ارائه و کارکرد مدل با بهره‌گیری از آزمون‌های عددی تأیید شد [۱۳]. میاتا و همکاران^۵ (۲۰۱۹)، یک مدل ریاضی برای برنامه‌ریزی هم‌زمان زمان‌بندی فعالیت‌ها و نگهداری و تعمیرات در یک مسئله تولید کارگاهی با در نظر داشتن توالی وابسته به زمان و زمان-های راه‌اندازی ارائه کردند. مدل ارائه‌شده با استفاده از الگوریتم هیوریستیک حل شد که یافته‌های عددی حاکی از کارایی مناسب رویکرد حل در مسائل عددی مختلف بود [۱۱].

ژانگ و همکاران (۲۰۱۹)، یک مدل فازی مبتنی بر قابلیت اطمینان و هزینه برای بهینه‌سازی برنامه نگهداری و تعمیرات تجهیزات نیروگاه بادی مستقر در دریا ارائه کردند. در مطالعه آن‌ها یک مدل برنامه‌ریزی ریاضی با دو هدف کمینه‌سازی هزینه‌های نت و بیشینه‌سازی

1. Soper, et al.
 2. Wang, et al.
 3. Ghalami & Grosso.
 4. Nattaf, et al.
 5. Miyata, et al.

قابلیت اطمینان تجهیزات ارائه شد. آن‌ها برای حل مسائل عددی در اندازه‌های گوناگون نیز از رویکرد

NSGA-II استفاده کردند [۲۵]. وانگ و یه^۱ (۲۰۱۹)، از رویکرد برنامه‌ریزی عدد صحیح مختلط برای حل مسئله تعیین توالی و زمان‌بندی ماشین‌های موازی مستقل استفاده کردند که هدف از آن حداقل‌سازی زمان اجرای کلیه فعالیت‌ها بر روی ماشین‌ها بود. در راستای حل مسئله‌های عددی در اندازه‌های مختلف، رویکرد شاخه و قیمت استفاده شد [۲۳].

ارتم^۲ و همکاران (۲۰۲۲)، مدل ریاضی مختلط عدد صحیح را همراه با الگوریتم‌های حل برای مسئله زمان‌بندی منابع چندمهارتی انعطاف‌پذیر ارائه کردند. با توجه به در دسترس بودن محدود خدمه تعمیر و نگهداری چندمهارتی و نیاز مبرم به به حداقل رساندن زمان تکمیل SM، مدل‌های بهینه‌سازی پیشنهادی این امکان را فراهم می‌آورند که در صورت نیاز، یک کارگر ماهر وظیفه‌ای را انجام دهد که به سطوح مهارت کمتری نیاز دارد. نتایج عددی حاکی از کاهش متوسط ۱۶/۵۲ درصدی در زمان اتمام تعطیلی زمانی است که از کارگران انعطاف‌پذیر استفاده می‌شود [۵].

جعفرزنجانی و همکاران (۲۰۲۲)، در مطالعه‌ای یک مدل بهینه‌سازی دوهدفه را برای برنامه‌ریزی و زمان‌بندی نگهداری در یک شبکه MFP پیشنهاد کردند [۸]. در مدل پیشنهادی نحوه انجام عملیات تعمیر و نگهداری و دوره‌های نگهداری به ترتیب بر اساس عملکرد خرابی در مراحل برنامه‌ریزی و زمان‌بندی تعیین شد. علاوه بر این، دو استراتژی برای انعطاف‌پذیری شبکه MFP در صورت وقوع اختلال پیشنهاد شده بود. توابع هدف به حداقل رساندن هزینه‌های تعمیر و نگهداری و به حداکثر رساندن قابلیت اطمینان بود. روش محدودیت اسیلون تقویت‌شده برای به دست آوردن جبهه پارتو و مبادله اهداف مورد استفاده قرار گرفت. به دلیل عدم قطعیت ذاتی پارامترها، یک رویکرد برنامه‌ریزی قوی مؤثر برای کنترل عدم قطعیت پارامترهای ورودی و سطح محافظه‌کاری تصمیمات خروجی به کار رفت.

عباسی و همکاران (۲۰۲۲)، موضوع برنامه‌ریزی تعمیر و نگهداری مبتنی بر ریسک و پیش-بینی زیرساخت‌های مهندسی را بررسی کرد. برای اطمینان از سطح قابل قبولی از ایمنی و در دسترس بودن سیستم، اتخاذ برنامه‌های بهینه و رویه‌های عملی در دارایی‌های مهندسی برنامه‌ریزی تعمیر و نگهداری ضروری است. هدف این پژوهش، شناسایی شکاف‌ها، نیازها و چالش‌های برنامه‌ریزی نگهداری در تأسیسات مهندسی بود. نتایج تجربی و نظری را با ادغام کرده و قابلیت‌ها و معایب پیشرفته‌ترین فناوری‌ها را برجسته می‌کند و فرصت‌ها و چالش‌های پژوهشی را توضیح می‌دهد [۶].

1. Wang & Ye

2 Ertem et al

ساجدی‌نژاد و لطفی (۲۰۱۹)، مدل بهینه زمان‌بندی نت پیشگیرانه غیرادواری برای سیستم‌های چندجزیی (سری - موازی) را بر مبنای حداکثر قابلیت دسترسی اجزای سیستم ارائه کردند. در مطالعه آن‌ها علاوه بر تأمین سطح قابلیت اطمینان موردنیاز سیستم و ارضای سایر محدودیت‌های سیستمی، کل هزینه‌های مرتبط با نت حداقل‌سازی شده و برخی از فعالیت‌های نت شامل بازرسی و سرویس ساده، تعمیرات پیشگیرانه و تعویض پیشگیرانه برای هر جزء پیشنهاد شده است [۱۶]. نادى‌زاده اردکانی و همکاران (۲۰۲۰)، مدلی برای برنامه‌ریزی تعمیرات و نگهداری سیستم‌های چندمؤلفه‌ای که در معرض خرابی سخت و نرم قرار دارند، ارائه دادند که در آن با استفاده از معیار حداقل‌سازی متوسط مجموع هزینه‌های بازرسی، تعمیر و نگهداری و جریمه ناشی از تأخیر در شناسایی خرابی‌های نرم، فاصله زمانی بهینه بین بازرسی‌های متوالی سیستم در یک افق زمانی محدود تعیین می‌شود [۱۲]. شایانیا و بهنامیان (۲۰۱۷)، یک الگوریتم بهینه‌سازی حرکت پرندگان به‌منظور حل مدل مسئله زمان‌بندی تولید کارگاهی منعطف با ماشین‌های موازی ارائه کردند. در این پژوهش، نوع خاصی از انعطاف‌پذیری در مسئله سیستم زمان‌بندی تولید کارگاهی تعریف شد که در هر ایستگاه کاری ممکن است چندین ماشین موازی برای انجام فعالیت‌ها موجود باشد [۱۸]. صفرزاده و اخوان (۲۰۱۹)، به زمان‌بندی ماشین‌های موازی یکنواخت با در نظر گرفتن شاخص‌های زیست‌محیطی پرداختند. در این پژوهش یک مدل زمان‌بندی در محیط ماشین‌های موازی یکنواخت با هدف حداقل‌سازی مجموع هزینه‌های عملیات ماشین‌ها ارائه شد که در آن مسئله زمان تکمیل فعالیت‌ها به‌عنوان یک محدودیت در نظر گرفته شده است. به‌منظور حل مسئله، یک رویکرد ابتکاری مبتنی بر حالت آزاد شده مدل پیشنهاد شد و کارکرد آن با استفاده از مسائل عددی به اثبات رسید [۱۵]. خلاصه‌ای از پژوهش‌های انجام‌شده در سال‌های اخیر در جدول ۱، قابل مشاهده است.

جدول ۱. پژوهش‌های انجام‌شده در مبنای نظری موضوع

| پژوهشگر/ سال | نوع مسئله | پارامترها | روش حل |
|-------------------------------------|---|-----------|--------------------------|
| نادی‌زاده اردکانی و همکاران، (۲۰۲۰) | برنامه‌ریزی نت سیستم‌های چندمؤلفه‌ای | قطعی | برنامه‌ریزی ریاضی |
| عباسی و همکاران، (۲۰۲۲) | برنامه‌ریزی تعمیر و نگهداری مبتنی بر ریسک | قطعی | برنامه‌ریزی ریاضی |
| ارتم و همکاران (۲۰۲۲) | زمان‌بندی تعمیر و نگهداری | قطعی | مدل ریاضی مختلط عدد صحیح |
| وانگ و یه، (۲۰۱۹) | زمان‌بندی ماشین‌های موازی مستقل از یکدیگر | قطعی | روش شاخه و قیمت |
| میاتا و همکاران، (۲۰۱۹) | زمان‌بندی ماشین‌آلات در سیستم تولید کارگاهی | قطعی | رویکرد ابتکاری |
| نتاف و همکاران، | زمان‌بندی ماشین‌های موازی با در نظر گرفتن | قطعی | رویکرد ابتکاری |

| پژوهشگر/ سال | نوع مسئله | پارامترها | روش حل |
|-------------------------------|--|---------------------|---|
| (۲۰۱۹) | محدودیت‌های زمانی | | |
| قلمی و گروسو، (۲۰۱۹) | زمان‌بندی ماشین‌های موازی | قطعی | رویکرد تقریبی موازی |
| وانگ و همکاران (۲۰۱۸) | زمان‌بندی ماشین‌های منعطف | قطعی | الگوریتم شاخه و قیمت |
| سایر و همکاران، (۲۰۱۹) | زمان‌بندی ماشین‌های موازی | قطعی | رویکرد ابتکاری |
| شن و همکاران، (۲۰۱۸) | زمان‌بندی جریان کارگاهی منعطف | قطعی | جست‌وجوی ممنوع |
| رویز و ماروتو، (۲۰۱۶) | برنامه‌ریزی ماشین‌های موازی مستقل | قطعی | رویکرد فراابتکاری |
| فیتوچی و همکاران، (۲۰۱۴) | برنامه‌ریزی نت پیشگیرانه | قطعی | رویکرد ابتکاری |
| شایانیان و بهنامیان، (۲۰۱۷) | برنامه‌ریزی ماشین‌آلات منعطف | قطعی | رویکرد فراابتکاری |
| ساجدی‌نژاد و لطفی، (۲۰۱۹) | زمان‌بندی نت پیشگیرانه برای سیستم‌های چندجزیی | قطعی | رویکرد فراابتکاری |
| صفرزاده و اخوان‌نیاکی، (۲۰۱۹) | برنامه‌ریزی ماشین‌آلات با در نظر گرفتن مسائل زیست‌محیطی | قطعی | رویکردهای ابتکاری و محدودیت اسپیلون |
| جعفرزنجانی و همکاران، (۲۰۲۲) | برنامه‌ریزی و زمان‌بندی نگهداری در یک شبکه MFP | قطعی | رویکردهای ابتکاری و محدودیت اسپیلون |
| پژوهش حاضر | زمان‌بندی تولید و برنامه‌ریزی نگهداری و تعمیرات برای ماشین‌های موازی | احتمالی و عدم قطعیت | رویکرد فراابتکاری، مدت زمان انجام فعالیت‌های فازی |

با مرور پژوهش‌های پیشین ملاحظه می‌شود که تاکنون پژوهشی با رویکرد پژوهش حاضر انجام نشده است. به‌طور کلی نوآوری‌های این پژوهش را می‌توان در این موارد بیان کرد:

۱. تحلیل هم‌زمان دو مسئله زمان‌بندی و نگهداری و تعمیرات در یک قالب یکپارچه؛ ۲. بررسی احتمال اختلال در عملکرد ماشین‌های موازی؛ ۳. در نظر گرفتن عدم قطعیت در پارامتر مدت‌زمان پردازش و ۴. طراحی رویکرد جدیدی برای حل مسائل عددی در اندازه‌های مختلف. نظر به ساختار سخت مسئله مورد بررسی در این پژوهش، زمان حل آن در ابعاد بزرگ بسیار قابل توجه است. از این نظر، حل مسائلی در ابعاد بزرگ با بهره‌گیری از نرم‌افزارهای تجاری مثل GAMS و LINGO بسیار دشوار و در برخی موارد غیرممکن است؛ از این رو در پژوهش حاضر به منظور حل مسائل عددی در ابعاد بزرگ از رویکرد فراابتکاری جدیدی استفاده می‌شود که تاکنون در مبانی نظری موضوع به کار نرفته است و یکی از نوآوری‌های پژوهش حاضر را می‌توان در رویکرد حل آن دانست.

۳. روش‌شناسی پژوهش

پژوهش حاضر از منظر روش، توصیفی است؛ زیرا وضعیت و شرایط فعلی یک سیستم موجود را بررسی می‌کند. از منظر هدف نیز پژوهشی کاربردی به‌شمار می‌رود.

مدل‌سازی پژوهش

مفروضات مدل. مفروضاتی که در این پژوهش برای مدل‌سازی مسئله در نظر گرفته شده است را می‌توان در چند بند به شرح زیر خلاصه کرد:

- ماشین‌آلات با یکدیگر به‌صورت موازی چیدمان شده‌اند؛
- مدت‌زمان انجام فعالیت‌های مربوط به نگهداری و تعمیرات در ماشین‌آلات مختلف دارای عدم قطعیت فازی است؛
- مسئله از جمله مسائل چندمحصولی است؛
- احتمال اختلال در تسهیلات مورد استفاده وجود دارد و باعث کاسته شدن از قابلیت اطمینان تجهیزات می‌شود؛
- سیستم نگهداری و تعمیرات از نوع پیشگیرانه است.

اجزای مدل

مجموعه‌ها

M : مجموعه ماشین‌های موجود

I : مجموعه محصولات شرکت

اندیس‌ها

$m \in M$: اندیس مربوط به ماشین‌آلات

$i, j \in I$: اندیس مربوط به محصولات

پارامترها

t_{im} : مدت‌زمان پردازش مورد نیاز کالای i در ماشین m

w_{im} : وزن هر محصول در جهت پردازش در ماشین m

قابلیت اعتماد ماشین m : λ_m

$\bar{P}t_m$: مدت‌زمان مورد نیاز برای انجام نگهداری و تعمیرات پیشگیرانه روی ماشین m

متغیرهای باینری (صفر و یکی)

x_{jm} : اگر کالای j به ماشین m تخصیص یابد، ۱ و در غیر این صورت صفر است.

z_{jm} اگر ماشین m پیش از پردازش محصول j احتیاج به نگهداری و تعمیرات پیشگیرانه داشته باشد، ۱ و در غیر این صورت صفر است.
 y_{ijm} اگر محصول j پس از محصول i در ماشین m پردازش شود، ۱ و در غیر این صورت صفر است.

متغیرهای مثبت

x_{im} زمان پایان پردازش محصول i ام در ماشین m

مدل پژوهش

توابع هدف

$$\text{Min}(Z1) = \sum_{i \in I} \sum_{m \in M} w_{im} \cdot c_{im} \quad (1)$$

$$\text{Max}(Z2) = \sum_{j \in J} \sum_{m \in M} \lambda_m \cdot x_{jm} \quad (2)$$

تابع هدف نخست، کمینه‌سازی زمان تکمیل وزنی محصولات و تابع هدف دوم، بیشینه‌سازی قابلیت اطمینان در خط تولید است.

محدودیت‌ها

$$\sum_m x_{jm} = 1 \quad \forall j \in I \quad (3)$$

$$\sum_{i \in I} y_{ijm} = x_{jm} \quad \forall j \in I, m \in M \quad (4)$$

$$\sum_{i \in I; i \neq j} y_{ijm} \leq x_{jm} \quad \forall j \in I, m \in M \quad (5)$$

$$\sum_{j \in I} y_{ijm} = x_{jm} \quad \forall m \in M \quad (6)$$

$$\sum_j z_{jm} = 1 \quad \forall j \in I \quad (7)$$

$$z_{jm} \leq x_{jm} \quad \forall j \in I, m \in M \quad (8)$$

$$c_{im} \geq t_{im} + \tilde{P}t_m \cdot z_{jm} \quad \forall j \in I, i \in I, m \in M; j = i \quad (9)$$

$$c_{jm} \geq t_{im} + c_{im} + \tilde{P}t_m \cdot z_{jm} \quad \forall j \in I, i \in I, m \in M; j \neq i \quad (10)$$

محدودیت ۳، مربوط به تخصیص محصولات به ماشین‌ها است. محدودیت‌های ۴، ۵ و ۶، توالی هر محصول در ماشین m را نشان می‌دهند. محدودیت‌های ۷ و ۸، نگهداری و تعمیرات پیشگیرانه را بر اساس اینکه پیش از پردازش کدام محصول باید اجرا شود، مشخص می‌سازند. محدودیت‌های ۹ و ۱۰، مدت‌زمان تکمیل هر محصول را مشخص می‌کنند.

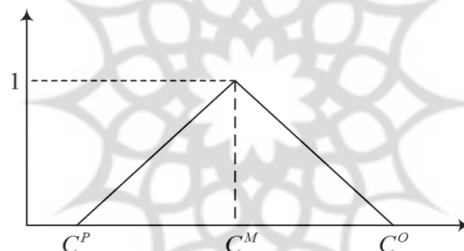
فازی‌زدایی مدل با استفاده از رویکرد جیمنز

معرفی رویکرد جیمنز. روش فازی‌زدایی مدل‌های برنامه‌ریزی ریاضی جیمنز و همکاران، از نظر محاسباتی شیوه‌ای مؤثر و کارآمد است؛ چراکه خطی بودن مدل را حفظ می‌کند و تعداد توابع هدف و محدودیت‌هایی که از جنس غیرمساوی هستند را افزایش نمی‌دهد [۶]. در راستای توضیح این روش، ابتدا بازه موردانتظار^۱ یک عدد فازی (EI) تعریف می‌شود. چنانچه \tilde{C} عددی فازی مثلی باشد، با توجه به شکل ۱، بازه موردانتظار این عدد فازی به شکل زیر تعریف می‌شود:

$$EI(\tilde{c}) = [E_1^C, E_2^C]$$

$$E_1^C = \frac{C^M + C^P}{2}$$

$$E_2^C = \frac{C^M + C^O}{2}$$
(۱۱)



شکل ۱. نحوه نشان‌دادن یک عدد فازی مثلی

در راستای فازی‌زدایی از تابع هدف، در رویکرد جیمنز از مقدار موردانتظار استفاده می‌شود؛ به‌طوری‌که به‌جای پارامترهای فازی در تابع هدف، مقادیر موردانتظار آن‌ها جایگزین می‌شود. مقدار موردانتظار یک عدد فازی از رابطه زیر به‌دست می‌آید:

$$EV(Z) = \frac{E_1^Z + E_2^Z}{2}$$
(۱۲)

1. Expected Interval

برای فازی‌زدایی محدودیت‌ها، اعداد \tilde{a} و \tilde{b} را به‌عنوان دو عدد فازی مثلثی در نظر بگیرید. با بهره‌گیری از رویکرد جیمنز، درجه ارضای رابطه $\tilde{a} \leq \tilde{b}$ به‌صورت زیر به‌دست می‌آید:

$$\mu_M(\tilde{a} \leq \tilde{b}) = \begin{cases} 0 & \text{if } E_2^b - E_1^a \leq 0 \\ \frac{E_2^b - E_1^a}{E_2^a - E_1^a + E_2^b - E_1^b} & \text{if } 0 \in [E_2^b - E_1^a, E_2^a - E_1^b] \\ 1 & \text{if } E_1^b - E_2^a \geq 0 \end{cases} \quad (13)$$

اگر $\mu_M(\tilde{a} \leq \tilde{b}) \geq \alpha$ برقرار باشد، این‌گونه گفته می‌شود که حداقل در درجه α ، \tilde{b} بزرگ‌تر یا مساوی \tilde{a} است؛ از این‌رو به‌منظور فازی‌زدایی از محدودیت $\sum_j \tilde{a}_{ij} x \leq \tilde{b}_i$ به‌صورت زیر عمل می‌شود:

$$\mu_M(\sum_j \tilde{a}_{ij} x \leq \tilde{b}_i) \geq \alpha \rightarrow (1-\alpha)E_2^{a_{ij}x} + \alpha E_1^{a_{ij}x} \leq \alpha E_2^{b_i} + (1-\alpha)E_1^{b_i} \quad (14)$$

در رابطه بالا، حداقل در درجه α ، $\sum_j \tilde{a}_{ij} x$ بزرگ‌تر از \tilde{b}_i است.

فازی‌زدایی از مدل. در این بخش، مدل پژوهش با بهره‌گیری از رویکرد جیمنز و همکاران (۲۰۰۷)، فازی‌زدایی می‌شود. در این پژوهش پارامترهای دارای عدم قطعیت با رویکرد فازی مثلثی مدنظر قرار گرفته‌اند. هر پارامتر فازی \tilde{C} به‌صورت $[C^P, C^M, C^O]$ مدنظر قرار می‌گیرد. همان‌طور که قبلاً بیان شد، محدودیت‌های ۹ و ۱۰، عدم قطعیت فازی دارند؛ از این‌رو در این بخش معادل قطعی برای این دو محدودیت ذکر می‌شود. در این راستا، محدودیت ۹، به‌صورت محدودیت ۱۵، نوشته می‌شود:

$$c_{im} \geq t_{im} + [(1-\alpha)\frac{1}{4}(Pt_m^o + 2Pt_m^m + Pt_m^p)].z_{jm} + [(\alpha)\frac{1}{4}(Pt_m^o + 2Pt_m^m + Pt_m^p)].z_{jm} - M.(1-y_{ijm}) \quad (15)$$

محدودیت ۱۰، نیز به شکل محدودیت ۱۶، نوشته می‌شود:

$$c_{jm} \geq t_{im} + c_{im} + [(1-\alpha)\frac{1}{4}(Pt_m^o + 2Pt_m^m + Pt_m^p)].z_{jm} + [(\alpha)\frac{1}{4}(Pt_m^o + 2Pt_m^m + Pt_m^p)].z_{jm} - M.(1-y_{ijm}) \quad (16)$$

۴. تحلیل داده‌ها و یافته‌های پژوهش

این بخش به حل مدل ریاضی مسئله و همچنین تحلیل نتایج عددی اختصاص دارد. بر این اساس ابتدا روش‌های به‌کاررفته برای حل مسئله ارائه می‌شوند و در ادامه، نتایج عددی که با بهره‌گیری از این روش‌های حل به‌دست آمده‌اند، تشریح خواهد شد.

معرفی روش‌های حل مدل. با توجه به ساختار مدل‌های مسائل زمان‌بندی تولید، نیاز به استفاده از رویکردهای فراابتکاری یا متاهیوریستیک برای حل مسائل عددی در ابعاد مختلف احساس می‌شود؛ چراکه حل مدل توسط نرم‌افزارهای محاسباتی تجاری مثل GAMS و LINGO در اندازه‌های بزرگ به زمان زیادی نیاز دارد. الگوریتم‌های فراابتکاری پژوهش حاضر شامل MOPSO و NSGA-II است که در این بخش به معرفی آن‌ها پرداخته خواهد شد.

علت انتخاب NSGA-II به‌عنوان الگوریتم اصلی حل. علت انتخاب NSGA-II برای حل

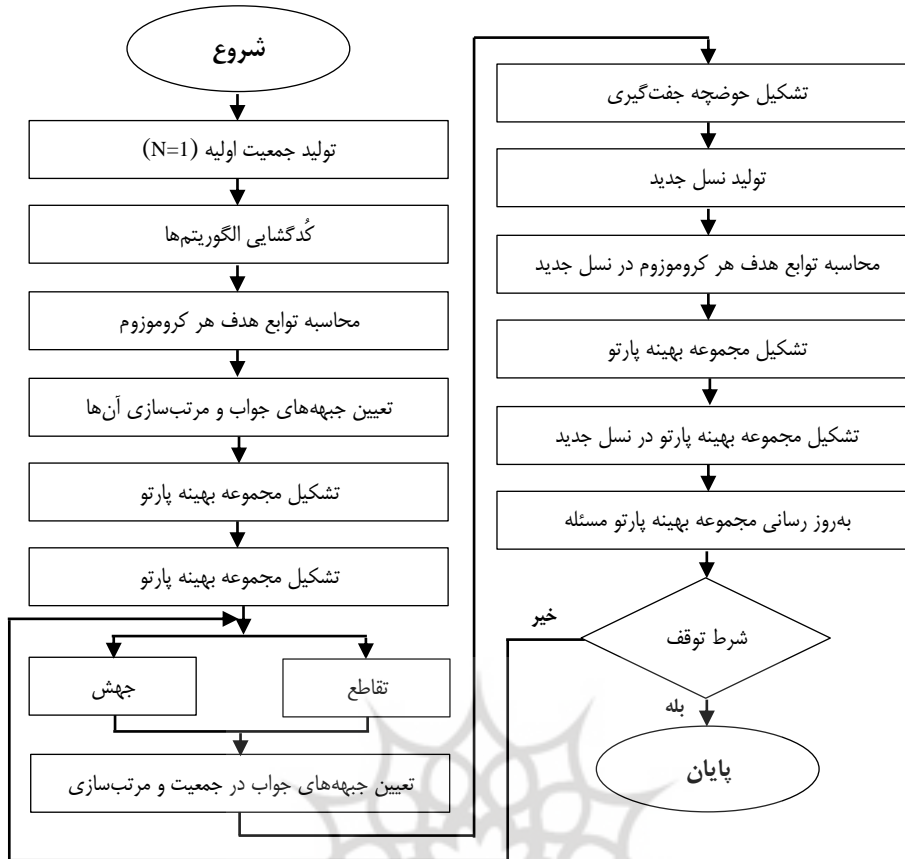
مسائل در اندازه‌های گوناگون در این پژوهش را می‌توان به شرح زیر ذکر کرد:

- یافته‌های محاسباتی و نظری در مبنای نظری موضوع پژوهش حاکی از آن است که الگوریتم ژنتیک در حل مسائل عددی که متغیرهای عددی صحیح باینری زیادی دارند، عملکرد بسیار مناسب‌تری نسبت به سایر رویکردها دارد؛
- این الگوریتم رویکرد جست‌وجوی ساده‌ای دارد و زمان پردازش نسبتاً مناسبی در غالب مسائل از خود نشان می‌دهد؛
- این الگوریتم انعطاف بالایی دارد و با هر نوعی از تابع هدف و محدودیت در فضای جست‌وجو فعالیت می‌کند؛
- صرفاً از مقادیر تابع هدف به‌منظور اجرای بهینه‌سازی استفاده کرده و به همین دلیل حافظه کمتری را اشغال می‌کند [۳].

الگوریتم NSGA-II. به دلیل انتقاداتی از جمله پیچیدگی محاسباتی بالا، رویکرد غیرنخبه‌گرایانه و مواردی از این قبیل که به الگوریتم‌های تکاملی چندهدفه معمولی وارد است، دب و همکاران (۲۰۰۲)، یک الگوریتم سریع و نخبه‌گرا به نام NSGA-II ارائه کردند [۳]. این الگوریتم به این دلیل در پژوهش حاضر استفاده می‌شود که مختص مدل‌های برنامه‌ریزی چندهدفه است و همچنین مشاهدات تجربی نشان داده است که این الگوریتم برای حل مسائل برنامه‌ریزی عدد صحیح و به‌خصوص مسائل عدد صحیح از نوع صفر و ۱ (باینری) عملکرد بهتری در مقایسه با سایر الگوریتم‌ها دارد.

گام‌های الگوریتم NSGA-II

- گام‌های این الگوریتم به شرح زیر است [۳]:
- گام اول: جمعیت اولیه P_0 به اندازه N با جواب‌های تصادفی ایجاد می‌شود و $t=0$ قرار می‌گیرد.
- گام دوم: در صورت برقرار نبودن شرط توقف به P_t بازگشته می‌شود.
- گام سوم: به کمک عملگر انتخاب مسابقه‌ای دودویی تعداد N والد از جمعیت P_0 انتخاب می‌شود.
- گام چهارم: با استفاده از عملگرهای تقاطع و جهش برای جمعیت P_t جمعیت فرزندان Q_t به اندازه N ایجاد می‌شود.
- گام پنجم: جمعیت جدید برابر با مجموع جمعیت P_t و جمعیت فرزندان Q_t ایجاد می‌شود که با R_t مشخص می‌شود.
- گام ششم: برای مشخص کردن مجموعه‌های پارتو F_i ، $i=1,2,3,\dots$ در جمعیت R_t از روش رتبه‌بندی نامغلوب استفاده می‌شود.
- گام هفتم: جمعیت P_{t+1} برابر با صفر قرار گرفته و $i=1$ در نظر گرفته می‌شود.
- گام هشتم: تا زمانی که رابطه $|P_{t+1}| + |F_i| < N$ برقرار است؛
۱. جواب‌های مجموعه F_i به جمعیت P_t اضافه می‌شود.
 ۲. $i=i+1$ در نظر گرفته می‌شود.
- گام نهم: مجموعه جواب‌های به دست آمده طبق فاصله ازدحام و نزولی مرتب می‌شود.
- گام دهم: به اندازه $N - |P_{t+1}|$ از اولین جواب‌های F_i به جمعیت P_{t+1} انتقال داده می‌شود.
- گام یازدهم: $t=t+1$ در نظر گرفته شده و به گام دوم بازگشته می‌شود.



شکل ۲. فلوجارت نمایش مراحل الگوریتم فراابتکاری NSGA-II

رویکرد رتبه‌بندی نامغلوب سریع. در یک رویکرد ساده برای تعیین جواب‌های نخستین جبهه غیرمغلوب در یک جمعیت با اندازه N ، هر جواب می‌تواند با هر جواب دیگر در جمعیت به‌منظور یافتن اینکه آیا مغلوب است یا خیر، مقایسه شود. این رویکرد به تعداد $O(MN)$ مقایسه نیاز دارد که M تعداد اهداف است. زمانی که این فرایند ادامه می‌یابد تا تمامی اعضای نخستین سطح نامغلوب در جمعیت را بیابد، پیچیدگی کل برابر $O(MN^2)$ می‌شود. در این مرحله همه افراد در جبهه اول نامغلوب پیدا می‌شوند. برای یافتن افراد در جبهه نامغلوب بعدی، جواب‌های نخستین جبهه به‌طور موقتی کاسته می‌شوند. در بدترین شرایط، یافتن دومین جبهه به تعداد $O(MN^2)$ مقایسه نیاز دارد، به‌خصوص زمانی که $O(N)$ تعداد جواب‌های متعلق به دومین و بالاترین سطوح نامغلوب است؛ بنابراین در بدترین شرایط تعداد N جبهه و تنها یک جواب در هر جبهه وجود دارد که به تعداد $O(MN^2)$ مقایسه نیاز است.

ابتدا برای هر جواب، دو نهاد محاسبه می‌شود: ۱. شمارش غلبه (n_p) تعداد جواب‌هایی که بر جواب p غلبه می‌کنند؛ ۲. S_p ، مجموعه‌ای از جواب‌هایی که جواب p بر آن‌ها غلبه می‌کند که این به $O(MN^2)$ مقایسه نیاز دارد.

شمار غلبه برای تمامی جواب‌ها در نخستین جبهه نامغلوب برابر صفر است. برای هر جواب P با $n_p=0$ هر عضو q از مجموعه S_p نظارت و شمارش غلبه آن به اندازه ۱ کاهش می‌یابد. به این ترتیب اگر برای هر عضو q شمارش غلبه برابر صفر شود، در مجموعه Q قرار داده می‌شود. این اعضا به جبهه نامغلوب دوم تعلق دارند؛ سپس رویه بالا با هر عضو مجموعه Q ادامه می‌یابد و جبهه سوم مشخص می‌شود. این رویکرد ادامه می‌یابد تا وقتی که تمامی جبهه‌ها مشخص شوند.

عملگر مقایسه جمعیتی. این عملگر، رویه انتخاب در چندین مرحله مختلف الگوریتم را به سمت یک جبهه بهینه پارتو هدایت می‌کند. فرض کنید که هر فرد i در جمعیت دو ویژگی رتبه نامغلوبی و فاصله ازدحامی دارد. در اینجا یک ترتیب نسبی از عملگر مقایسه جمعیتی مشخص می‌شود:

$$i <_n j \text{ if } (i_{rank} < j_{rank}) \text{ or } ((i_{rank} = j_{rank}) \text{ and } (i_{distance} > j_{distance})) \quad (14)$$

بین دو جواب با رتبه‌های نامغلوب متفاوت، انتخاب جواب با رتبه کمتر ترجیح داده می‌شود؛ در غیر این صورت، اگر هر دو جواب متعلق به یک جبهه باشند، جوابی که در یک ناحیه جمعیتی کمتر باشد، انتخاب می‌شود.

الگوریتم فراابتکاری MOPSO. الگوریتم PSO^۱ یک الگوریتم جست‌وجوی اجتماعی است که از روی رفتار اجتماعی دسته‌های پرندگان مدل شده است. در ابتدا این الگوریتم به منظور کشف الگوهای حاکم بر پرواز هم‌زمان پرندگان و تغییر ناگهانی مسیر آن‌ها و تغییر شکل بهینه‌ی دسته به کار رفت. در الگوریتم PSO، ذرات در فضای جست‌وجو جاری می‌شوند. تغییر مکان ذرات در فضای جست‌وجو تحت تأثیر تجربه و دانش خودشان و همسایگانشان است؛ بنابراین موقعیت دیگر ازدحام ذرات روی چگونگی جست‌وجوی یک ذره اثر می‌گذارد. نتیجه مدل‌سازی این رفتار اجتماعی فرایند جست‌وجویی است که ذرات به سمت نواحی موفق میل می‌کنند. ذرات از یکدیگر می‌آموزند و بر مبنای دانش به دست آمده به سمت بهترین همسایگان خود می‌روند. اساس کار PSO بر این اصل استوار است که در هر لحظه هر ذره مکان خود را در فضای جست‌وجو با توجه به بهترین مکانی که تاکنون در آن قرار گرفته است و بهترین مکانی که در کل همسایگی‌اش وجود دارد، تنظیم می‌کند.

نخستین نسخه MOPSO در سال ۱۹۹۹ ارائه شد. این الگوریتم از عملگری به نام «عملگر جهش واحد» استفاده می‌کند که یک عضو از جمعیت را انتخاب کرده و مقدار یک بعد آن را به عددی در محدوده مقادیر معتبر تغییر می‌دهد؛ همچنین یک سیاست نخبه‌گرایی به منظور نگه داشتن نتایج برتر و غالب در تکرارهای الگوریتم تعریف شده است. پاسخ‌های غالب در آرشیو خارجی که ساختار گرید دارد، ذخیره می‌شوند. دستاورد مدیریت صحیح آرشیو، تنوع جمعیت و توزیع مناسب پاسخ‌ها است. انتخاب pbest و gbest طبق سازوکار خاصی انجام می‌شود. الگوریتم چندهدفه نمی‌تواند از رابطه ۳ برای شناسایی آن‌ها استفاده کند؛ زیرا همه پاسخ‌های غالب به یک میزان برتری دارند. تنها زمانی pbest به‌روز می‌شود که ذره جدیدی مقدار قبلی آن را مغلوب کند. gbest نیز در هر تکرار از میان پاسخ‌های غالب موجود در آرشیو انتخاب می‌شود.

الگوریتم بهینه‌سازی ازدحام ذرات به‌عنوان یک روش جست‌وجوی غیرقطعی برای بهینه‌سازی تابعی مطرح شد. این الگوریتم از حرکت دسته‌جمعی پرندگان در جست‌وجوی غذا الهام گرفته شده است. گروهی از پرندگان در فضایی به‌صورت تصادفی دنبال غذا می‌گردند و تنها یک تکه غذا در فضای موردبحث وجود دارد؛ درحالی‌که هیچ‌یک از پرندگان محل غذا را نمی‌داند. یکی از بهترین استراتژی‌ها می‌تواند دنبال کردن پرنده‌ای باشد که کمترین فاصله را تا غذا داشته باشد. این استراتژی در واقع جانمایه الگوریتم است. هر راه‌حل که به آن یک ذره گفته می‌شود. در الگوریتم بهینه‌سازی ازدحام ذرات معادل یک پرنده در الگوریتم حرکت جمعی پرندگان است. هر ذره یک مقدار شایستگی دارد که توسط یک تابع شایستگی محاسبه می‌شود. هر چه ذره در

فضای جست‌وجو به هدف - غذا در مدل حرکت پرندگان نزدیک‌تر باشد، شایستگی بیشتری دارد؛ همچنین هر ذره دارای یک سرعت است که هدایت حرکت ذره را بر عهده دارد. هر ذره با دنبال کردن ذرات بهینه در حالت فعلی به حرکت خود در فضای مسئله ادامه می‌دهد. به این شکل که گروهی از ذرات در آغاز کار به صورت تصادفی به وجود می‌آیند و با به‌روزرسانی نسل‌ها سعی در یافتن راه‌حل بهینه می‌کنند. در هر گام، هر ذره با استفاده از دو بهترین مقدار به‌روز می‌شود. نخستین مورد $pbest$ بهترین موقعیتی است که تاکنون ذره موفق به رسیدن به آن شده است. موقعیت یادشده شناخته و نگهداری می‌شود. بهترین مقدار دیگری که با نام $gbest$ توسط الگوریتم مورد استفاده قرار می‌گیرد، بهترین موقعیتی است که تاکنون توسط جمعیت ذرات به‌دست آمده است. در ادامه شبه‌کد الگوریتم MOPSO آورده شده است:

```

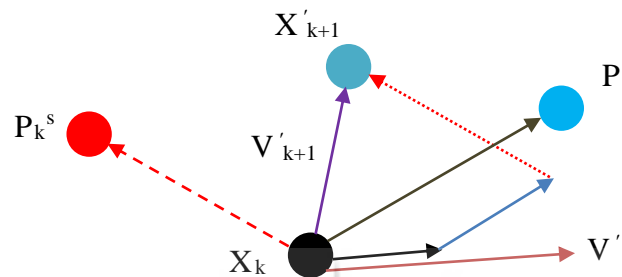
گام ۱: شروع MOPSO }
گام ۲: جمعیت اولیه را ایجاد کنید ();
گام ۳: سرعت اولیه را تعیین کنید ();
گام ۴: جمعیت اولیه را ارزیابی کنید ();
گام ۵:  $Fbest$  را به‌روزرسانی کنید ();
گام ۶:  $pbest$  را به‌روزرسانی کنید ();
گام ۷:  $nodom$  را وارد کنید؛
گام ۸:  $Gbestpos = rnd(0, nodomfileSize)$  {
گام ۹: for (i=1 to MAXCYCLES) {
گام ۱۰: for (j=0 to MAXPARTICLES) {
گام ۱۱: سرعت را به‌روزرسانی کنید ();
گام ۱۲: ذره را به‌روزرسانی کنید (); {
گام ۱۳: ذخیره‌سازی کنید.
گام ۱۴: جمعیت اولیه را ارزیابی کنید ();
گام ۱۵:  $Fbest$  را به‌روزرسانی کنید ();
گام ۱۶:  $pbest$  را به‌روزرسانی کنید ();
گام ۱۷:  $nodom$  را وارد کنید ();
گام ۱۸:  $Gbestpos = rnd(0, nodomfileSize)$  {
گام ۱۹: خروجی را چاپ کنید ();

```

شکل ۳. شبه‌کد الگوریتم فراابتکاری MOPSO

هر عضو در این گروه توسط بردار سرعت و بردار موقعیت در فضای جست‌وجو تعریف می‌شود. در هر تکرار زمانی، موقعیت جدید ذرات با توجه به بردار سرعت فعلی، بهترین موقعیت یافت‌شده توسط آن ذره و بهترین موقعیت یافت‌شده توسط بهترین ذره موجود در گروه

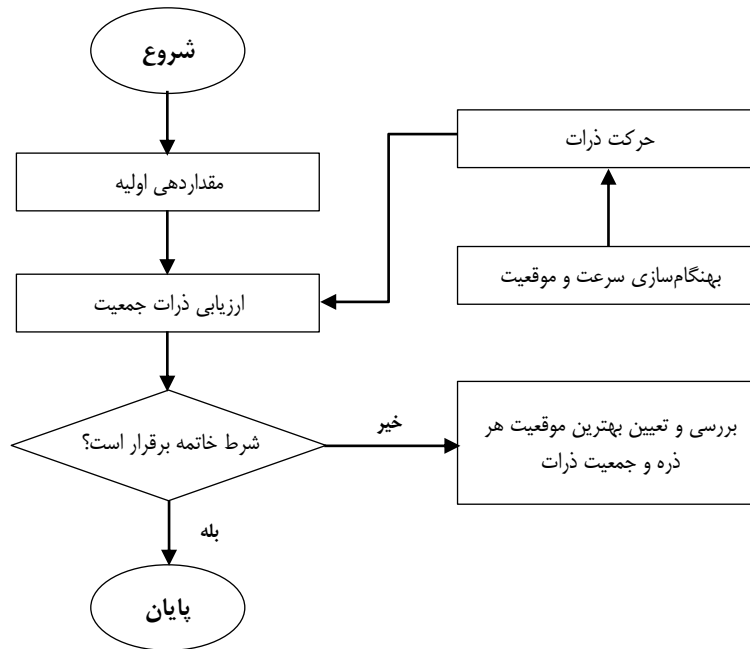
به‌روزرسانی می‌شود. همان‌طور که در شکل ۴، مشاهده می‌شود؛ گره قرمز رنگ (p_k^g) معرف بهترین موقعیت یافت‌شده توسط بهترین ذره موجود در گروه در تکرار k م اجرای الگوریتم است. گره‌های آبی و مشکی به ترتیب p^i ، x_k^i و به ترتیب بهترین موقعیت یافت‌شده برای ذره k م و موقعیت‌های این ذره در تکرارهای k م و $k+1$ م آن ذره است؛ همچنین بردارهای سبز و آبی و قرمز توپر به ترتیب بردارهای انتقال ذره در راستای سرعت قبلی، رسیدن به بهترین موقعیت پیدا شده برای ذره تا به الآن و بردار حرکت در راستای رسیدن به بهترین ذره موجود در گروه است.



شکل ۴. روند حرکت ذره‌ها در الگوریتم شبیه‌سازی حرکت پرندگان

در این الگوریتم، ایجاد جواب‌های اولیه به‌نحوی تصادفی تولید می‌شود. گام‌های این الگوریتم در شکل ۳، ارائه شده است. در این الگوریتم، آزمون همگرایی در حقیقت به‌روزرسانی موقعیت بهینه به‌دست‌آمده به‌وسیله هر ذره و موقعیت بهینه به‌دست‌آمده به‌وسیله بهترین ذره گروه ذرات است و تا زمانی که شرط توقف (انجام همه تکرارهای الگوریتم، گذر از مدت‌زمان مشخص‌شده برای اجرای الگوریتم، رسیدن به نوسانات مشخص‌شده در بهترین مقدار هدف ذرات و غیره) محقق شود، اجرای الگوریتم ادامه خواهد یافت.

آماده‌سازی الگوریتم‌ها. از جمله اقداماتی که باید برای استفاده از الگوریتم‌های هیوریستیک انجام داد، آماده‌کردن و به‌روزرسانی آن‌ها مطابق با مدل و مسئله خاصی است که قصد استفاده از آن وجود دارد؛ بنابراین در این بخش نحوه آماده‌سازی و به‌روزرسانی الگوریتم منطبق بر مدل مسئله ارائه می‌شود.



شکل ۵. فلوجارت الگوریتم بهینه‌سازی MOPSO

ژن‌های بخش اول: این ژن‌ها به منظور مشخص کردن متغیر X_{jm} استفاده می‌شوند. فرض کنید ۲ عدد تصادفی بین صفر و ۱ تولید و سپس با بهره‌گیری از دستور sort مرتب‌سازی و از بزرگ به کوچک نوشته شوند؛ آنگاه بر مبنای عناصر مرتب‌شده، اندیس‌های مربوط به هر کدام از این درایه‌ها را خارج می‌کنیم، مثلاً:

اگر ژن‌های مربوطه به صورت روبه‌رو در نظر گرفته شوند $a = [0.1 \ 0.3]$ باشد؛ آنگاه بعد از عملیات sort به صورت $a' = [0.3 \ 0.1]$ درآمده و سپس درایه مربوط به a' به صورت $b = [2 \ 1]$ می‌شود که برای ادامه کار به b احتیاج است و به جای اندیس j قرار داده می‌شود.

ژن‌های بخش دوم: از این ژن‌ها همانند رویکرد ژن‌های بخش یک و با استفاده از متغیر به‌دست‌آمده از بخش قبلی یعنی X_{jm} برای تعیین متغیر Z_{jm} ، Y_{ijm} استفاده می‌شود. در انتها با استفاده از متغیرهای به‌دست‌آمده و متغیرهای دیگر زمان تکمیل هر یک از محصولات محاسبه شده و سپس توابع هدف تعیین می‌شود.

نحوه نمایش کروموزوم. نحوه نمایش کروموزوم‌ها در جدول ۲، نشان داده شده است.

جدول ۲. نحوه نمایش کروموزمها

| | | | |
|--------|------|-----|------|
| بخش یک | ۰/۱ | ... | ۰/۶ |
| | ⋮ | ... | ⋮ |
| | ۰/۵ | .. | ۰/۱ |
| بخش دو | ۰/۳۱ | ... | ۰/۷۶ |
| | ⋮ | ... | ⋮ |
| | ۰/۴۵ | .. | ۰/۶۱ |
| بخش سه | ۰/۴۱ | ... | ۰/۸۵ |
| | ⋮ | ... | ⋮ |
| | ۰/۹۵ | .. | ۰/۰۱ |

بخش ۱: یک ماتریس به اندازه $|T| \times |D|$ با مقادیر بین صفر و یک ایجاد می‌شود؛

بخش ۲: یک ماتریس به اندازه $|T| \times |H|$ با مقادیر بین صفر و یک است؛

بخش ۳: یک ماتریس به اندازه $|T| \times |V|$ با مقادیر بین صفر و یک است.

تکنیک مرتب‌سازی. فرض کنید یک ماتریس با اعداد تصادفی زیر تولید شده است. می‌خواهیم از آن‌ها اعداد گسسته خارج کنیم:

| | | | |
|-----|-----|-----|-----|
| ۰/۳ | ۰/۹ | ۰/۶ | ۰/۱ |
|-----|-----|-----|-----|

پس عدد ۰/۳ در درایه یک و عدد ۰/۹ در درایه دو و عدد ۰/۶ در درایه ۳ و عدد ۰/۱ در درایه چهارم قرار دارد. حال با استفاده از دستور SORT به صورت صعودی مرتب می‌شوند؛ یعنی:

| | | | |
|-----|-----|-----|-----|
| ۰/۱ | ۰/۳ | ۰/۶ | ۰/۹ |
|-----|-----|-----|-----|

که اگر شماره درایه‌های اولیه آن را با دستور SORT جدا کنیم، به صورت زیر خواهد بود:

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

که یک توالی از اعداد گسسته به دست می‌آید.

قاعده نرمال‌سازی. یک ماتریس با اعداد تصادفی مثال بالا را در نظر بگیرید:

| | | | |
|-----|-----|-----|-----|
| ۰/۳ | ۰/۹ | ۰/۶ | ۰/۱ |
|-----|-----|-----|-----|

می‌خواهیم عدد ۱۰۰ را به سه قسمت مساوی تقسیم کنیم:

ابتدا ۳ ژن از این کروموزوم جدا می‌کنیم؛ برای مثال، [۰/۹ ۰/۶ ۰/۱] و در ادامه آن را نرمالایز کرده و در عدد ۱۰۰ ضرب می‌کنیم؛ یعنی:

$$\frac{[0.9, 0.6, 0.1]}{0.9 + 0.6 + 0.1} \cdot 100 = 56.25 \quad 37.5 \quad 6.25$$

عملگر تقاطع. برای عملگر تقاطع روش‌های متعددی تعریف شده است؛ اما روش کاربردی در این پژوهش، روش تقاطع یکنواخت است. در این روش، یک ژن از هر دو والد به‌طور مستقل از سایر ژن‌ها، شانس برابر برای حضور در کروموزوم یک فرزند را دارند. مزیت این روش نسبت به سایر روش‌ها، ایجاد تنوع ژنتیک بالاتر پس از ترکیب است.

$$X_1 = \{X_{11}, X_{12}, \dots, X_{1n}\}$$

$$X_2 = \{X_{21}, X_{22}, \dots, X_{2n}\}$$

$$\alpha \in \{0, 1\}$$

$$Y_1 = \{Y_{11}, Y_{12}, \dots, Y_{1n}\}$$

$$Y_2 = \{Y_{21}, Y_{22}, \dots, Y_{2n}\}$$

$$Y_1 = \alpha \cdot X_1 + (1 - \alpha) \cdot X_2 \quad 0 \leq \alpha \leq 1$$

$$Y_2 = \alpha \cdot X_2 + (1 - \alpha) \cdot X_1$$

$$N_c = 2 \lceil (P_c \cdot Npop) / 2 \rceil$$

$$0 \leq P_c \leq 1$$

عملگر جهش

۱. ژن‌هایی را که می‌خواهیم بر روی آن‌ها جهش انجام دهیم، بر اساس پارامترهای احتمال جهش و تعداد ژن‌ها و طبق مثال زیر به‌دست می‌آوریم.
۱-۱. فرض می‌کنیم کروموزومی به شکل زیر داریم.

$$y = [\quad 0/1 \quad 0/25 \quad 0/12 \quad 0/45 \quad 0/8 \quad 0/6]$$

ژن ۶ ژن ۵ ژن ۳ ژن ۴ ژن ۲ ژن ۱

۱-۲. تعداد ژن‌هایی که می‌خواهند جهش پیدا کنند را تعیین می‌کنیم:

احتمال جهش * تعداد ژن‌ها = nmu

۲. عدد ۰/۱ را در اختلاف بیشترین مقدار از کمترین مقدار ژن‌ها ضرب می‌کنیم.

$$\sigma = 0.1 * (\text{VarMax} - \text{VarMin}) = 0.1(1 - 0) = 0.1$$

۳. برای ژن‌های انتخاب‌شده، اعداد تصادفی بین صفر و یک تولید می‌شود و آن را در عدد به‌دست‌آمده از مرحله ۲ ضرب می‌کنیم.

$$y(j)=y(j)+\sigma*(randn)$$

۴. در انتها اگر ژن‌های جهش‌یافته از بازه تعیین‌شده تخطی کردند، اصلاح لازم اعمال می‌شود؛ یعنی اگر بالاتر از کران بالای خود بودند، کران بالا و اگر پایین‌تر از کران پایین خود بودند، کران پایین را قرار می‌دهیم.

ارزیابی کارایی الگوریتم. برای ارزیابی کارایی الگوریتم‌ها در حل مدل‌های برنامه‌ریزی ریاضی چندهدفه، شاخص‌های متعدد و متنوعی استفاده می‌شود. چند مورد از این معیارها که کاربرد بیشتری در مبانی نظری موضوع دارند در ادامه تشریح می‌شوند:

۱. **تعداد پاسخ‌های پارتو^۱ (NPS).** این شاخص تعداد جواب‌های نامغلوب که در هر سری اجرای الگوریتم به‌دست می‌آید را در نظر می‌گیرد. در حقیقت با توجه به این معیار، هر اندازه که تعداد پاسخ‌های ناچیره بیشتر باشد نشان‌دهنده آن است که الگوریتم عملکرد بهتری داشته است.

۲. **فاصله از نقطه ایده آل^۲ (MID).** این معیار میزان قرابت مجموعه کارایی به‌دست‌آمده توسط الگوریتم را نسبت به مرز پارتو مشخص می‌کند. به دلیل آنکه رسیدن به مرز کارایی برای بسیاری مسائل مقدور نیست، فاصله میان نقاط پارتو و نقطه ایده‌آل (همان صفر و صفر) را محاسبه می‌کنند. این معیار از رابطه ۱۷، محاسبه می‌شود.

$$MID = \frac{\sum_{i=1}^n c_i}{n} \quad c_i = \sqrt{f_{1i}^2 + f_{2i}^2 + \dots} \quad (17)$$

در رابطه ۱۷، n تعداد جواب‌های ناچیره (نامغلوب) به‌دست‌آمده و $i = 1, 2, \dots, n$. با توجه به این شاخص، هر میزان که مقدار MID یک الگوریتم بالاتر باشد، نشان‌دهنده فاصله بیشتر این الگوریتم نسبت به مرز کارایی است.

۳. **پراکندگی حل‌های نامغلوب^۳ (SNS).** این شاخص مقدار پراکندگی میان جواب‌های موجود در مجموعه پاسخ‌های نامغلوب حاصله به‌وسیله الگوریتم را حساب می‌کند. این معیار با استفاده از رابطه ۱۸، به‌دست می‌آید.

$$SNS = \sqrt{\frac{\sum_{i=1}^n (MID - c_i)^2}{n - 1}} \quad (18)$$

-
1. Number of Pareto Solutions
 2. Mean Ideal Distance
 3. Spread of Non-dominance Solutions

هرچه مقدار این شاخص برای یک الگوریتم بالاتر باشد، پراکندگی حل آن بالاتر و مطلوب‌تر است.

تنظیم پارامترها. نوع عملکرد الگوریتم‌های متاهوریستیک وابستگی زیادی به مقدار مؤلفه‌های الگوریتم دارد. در همین راستا، رویکردهای متعددی به منظور تنظیم پارامترهای الگوریتم‌های متاهوریستیک به منظور داشتن یک رویکرد حل قدرتمند وجود دارد. در این بخش از رویکرد تاگوچی سه‌سطحی، به دلیل ویژگی‌های متمایزی مثل کاهش قابل توجه تعداد آزمایش‌ها و همچنین حفظ اطلاعات موردنیاز، استفاده می‌شود.

مقادیر متوسط و واریانس در پایان یکپارچه و معیاری واحد به نام «نرخ سیگنال نسبت به نویز» و یا S/N ارائه می‌شود. در اینجا سیگنال و نویز نشان‌دهنده مقادیر مناسب (متغیرهای پاسخ) و نامناسب (انحراف استاندارد) هستند.

با حفظ چارچوب‌های اصل روش، یک نسخه تاگوچی چندهدفه در رابطه ۱۹، در راستای محاسبه مقدار هدف y که «متغیر کیفیت» نام دارد ارائه شده است؛ به نحوی که y_i و n نشان‌دهنده ارزش پاسخ آزمون آزمون و تعداد ارائه‌های متعامد هستند. مقادیر جواب‌های آزمایش‌ها توسط رابطه ۲۰، حساب می‌شوند. مهم‌ترین مزیت بهره‌گیری از این رابطه آن است که این معیار به‌طور هم‌زمان واگرایی و همگرایی را در بطن خود در نظر می‌گیرد.

$$S/N = -10 \log \left[\frac{\sum_{i=1}^n y_i^2}{n} \right] \quad (19)$$

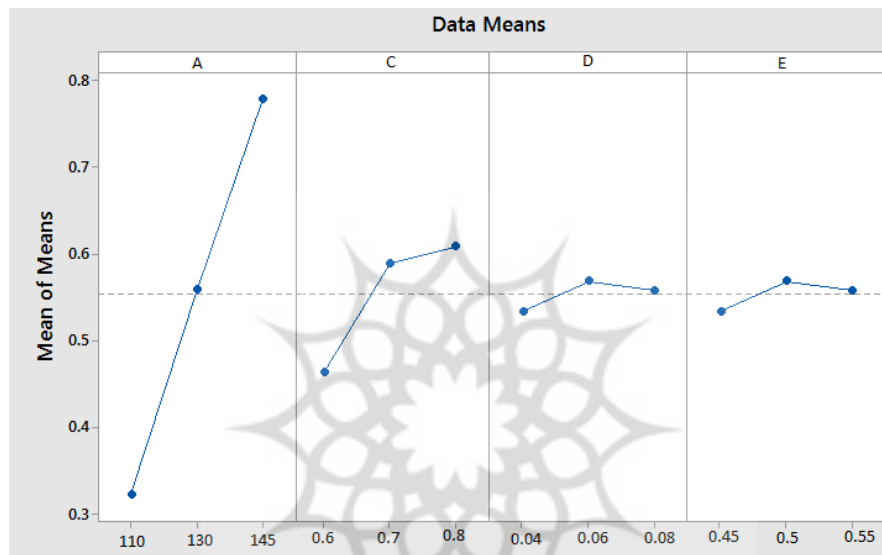
$$= \frac{MID}{DM}; i = 1, 2, \dots, ny_i \quad (20)$$

جدول ۳، ارائه‌های طراحی شده برای الگوریتم NSGA-II است. هر ارائه پنج بار برای محاسبه نرخ S/N تکرار شده است. پارامترهای جمعیت اولیه، تعداد تکرار، نرخ تقاطع، نرخ جهش و نرخ بازتولید به‌عنوان پارامترهای اصلی در الگوریتم موردنظر بوده‌اند. در جدول ۳، A نشان‌دهنده اندازه جمعیت، C نماد نرخ تقاطع، D نرخ بازتولید و E نمایانگر نرخ جهش است. برای عملیات تنظیم پارامترهای گفته‌شده از یک طرح سه‌سطحی تاگوچی استفاده می‌شود؛ بنابراین ۳ مقدار مختلف برای هر پارامتر بر اساس مبانی نظری و تخصص پژوهشگران در نظر گرفته شده است که در جدول ۳، نشان داده شده است.

جدول ۳. نمایش سطوح پارامترهای الگوریتم ژنتیک طبق روش تاگوچی

| نام پارامتر | A | C | D | E |
|----------------|---------------|---------------|-----------------|--------------------|
| مقادیر سه‌سطحی | ۱۱۰، ۱۳۰، ۱۴۵ | ۰/۸، ۰/۷، ۰/۶ | ۰/۸، ۰/۰۶، ۰/۰۴ | ۰/۵، ۰/۴۵، ۰/۵۵/۵۵ |

برای انجام آزمایش‌های عددی به منظور تنظیم پارامترهای بالا، با استفاده از نرم‌افزار MiniTab به اجرای آزمون‌های تاگوچی پرداخته می‌شود. به این صورت که هر آزمون ۵ مرتبه در نرم‌افزار اجرا می‌شود تا اثر تصادفی بودن آن‌ها کاهش یابد. در نهایت مقدار به دست آمده از میانگین اجرای آزمایش‌ها معادل مقدار سطح پاسخ در الگوریتم است. نتایج تجزیه و تحلیل داده‌ها که از نرم‌افزار MiniTab به دست آمده در شکل ۶ نشان داده شده است.



شکل ۶. نمودار تجزیه و تحلیل پارامترها

بهترین مقدار برای هر پارامتر طبق محاسبات صورت گرفته در نرم‌افزار Minitab و همچنین مقادیر سه‌سطحی تعریف شده در جدول ۴، ارائه شده است.

جدول ۴. پارامترها و مقادیر بهینه آن‌ها در NSGA-II

| نام پارامتر | A | B | C | D | E |
|-------------|-----|-----|-----|------|-----|
| مقدار بهینه | ۱۴۵ | ۱۷۰ | ۰/۸ | ۰/۰۶ | ۰/۵ |

نتایج عددی. بعد از آماده‌سازی الگوریتم و همچنین تنظیم سطوح پارامترها، می‌توان نسبت به انجام آزمون‌های عددی اقدام کرد. با حل نمونه‌های عددی در اندازه‌های مختلف می‌توان به اعتبارسنجی مدل ریاضی مسئله و تضمین موجه بودن محدودیت‌های آن پرداخت. در این بخش به طراحی مسائل عددی در اندازه‌های مختلف و سپس حل و تحلیل نتایج پرداخته می‌شود؛ بنابراین سه دسته مسائل در ابعاد کوچک، متوسط و بزرگ طراحی شده و سپس با استفاده از الگوریتم‌های طراحی شده حل می‌شوند. در مسائل عددی طراحی شده، F_1 مقادیر مربوط به تابع هدف نخست و F_2 مقادیر مربوط به تابع هدف دوم هستند.

مسائل کوچک. جدول ۵، شیوه تولید داده‌های مربوط به مسائل عددی را نشان می‌دهد. توابع استفاده شده در جدول ۶، مستخرج از مثال‌های عددی در مبانی نظری موضوع پژوهش به همراه مقداری تغییر طراحی شده است.

جدول ۵. منبع ایجاد مسائل عددی در اندازه‌های مختلف

| پارامتر | تابع توزیع تصادفی |
|-------------|-------------------|
| w_{im} | U [0.001,0.90] |
| pt_m | Fuzzy (3,5,7) |
| t_{im} | U [1,5] |
| λ_m | U [0.5,9] |

جدول ۶ اندیس مربوط به تولید مثال‌های کوچک به صورت زیر است:

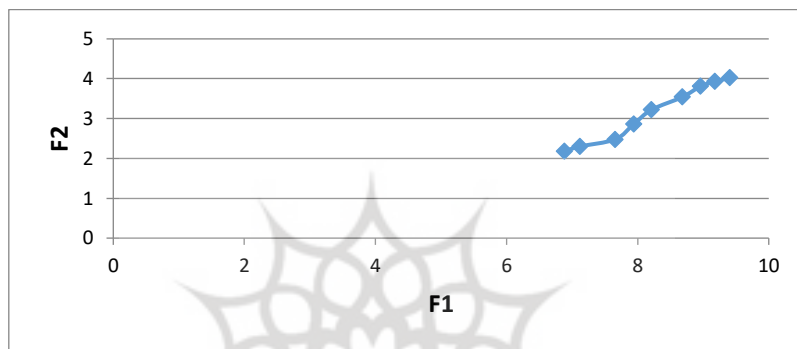
جدول ۶ ابعاد مسائل کوچک عددی طراحی شده

| مثال | اندیس‌ها | |
|------|----------|-------|
| | $ M $ | $ I $ |
| ۱ | ۱ | ۲ |
| ۲ | ۳ | ۳ |
| ۳ | ۳ | ۵ |
| ۴ | ۴ | ۵ |
| ۵ | ۴ | ۶ |

جدول ۷ و شکل ۸، مرز کارایی به دست آمده برای مسئله شماره ۲ را نشان می‌دهد.

جدول ۷. مرز کارا برای مسئله شماره ۲

| شماره پاسخ | F1 | F2 |
|------------|------|------|
| ۰ | ۶/۸۸ | ۲/۱۸ |
| ۱ | ۷/۱۲ | ۲/۳۰ |
| ۲ | ۷/۶۶ | ۲/۴۸ |
| ۳ | ۷/۹۴ | ۲/۸۶ |
| ۴ | ۸/۲۱ | ۳/۲۲ |
| ۵ | ۸/۶۸ | ۳/۵۴ |
| ۶ | ۸/۹۶ | ۳/۸۱ |
| ۷ | ۹/۱۸ | ۳/۹۳ |
| ۸ | ۹/۴۱ | ۴/۰۲ |



شکل ۷. مرز کارا برای مسئله شماره ۲

با در نظر گرفتن مقدار $0/5$ برای α ، مقادیر تابع هدف برای مسائل ۱ تا ۳ به شرح جدول ۸، است.

جدول ۸. نتایج حل مسائل عددی در ابعاد کوچک

| مثال | مقدار تابع هدف ۱ | | | مقدار تابع هدف ۲ | | | زمان (ثانیه) | | |
|------|------------------|-----------|------------|------------------|-----------|------------|--------------|-----------|------------|
| | GAM S | MOPS O | NSGAI I | GAM S | MOPS O | NSGAI I | GAMS | MOPS O | NSGAI I |
| ۱ | ۵/۰۸ | ۵/۰۸ | ۵/۰۸ | ۰/۷۴ | ۰/۷۴ | ۰/۷۴ | ۸/۷۹ | ۲/۱۶۶ | ۲/۹۹۱۲ |
| ۲ | ۸/۲۱ | ۸/۲۱ | ۸/۲۱ | ۳/۲۲ | ۳/۲۲ | ۳/۲۲ | ۱۰/۲۸۵ | ۲/۶۳۴ | ۳/۱۷۳۸ |
| ۳ | ۷/۹۷۳ | ۷/۹۷۳ | ۷/۹۷۳ | ۲/۸۶ | ۲/۸۶ | ۲/۸۶ | ۱۱/۸۸۸ | ۲/۷۳۴ | ۳/۱۱۴۸ |
| ۴ | ۸/۲۵۶ | ۸/۲۵۶ | ۷/۸۵۲ | ۳/۶۱ | ۳/۵۱ | ۳/۶۶ | ۱۲/۶۹۸ | ۲/۳۶۹ | ۳/۲۱۸۱ |
| ۵ | ۸/۴۱۸ | ۸/۴۱۸ | ۸/۶۵۲ | ۴/۱۲ | ۳/۸۷ | ۳/۸۱ | ۱۳/۷۸۹ | ۲/۵۸۷ | ۳/۴۱۵۵ |

مسائل متوسط. جدول ۹، ابعاد مسائل عددی در ابعاد متوسط را نشان می‌دهد.

جدول ۹. ابعاد مسائل متوسط عددی طراحی شده

| اندیس‌ها | | مثال |
|----------|-------|------|
| $ M $ | $ I $ | |
| ۴ | ۶ | ۶ |
| ۵ | ۸ | ۷ |
| ۶ | ۸ | ۸ |
| ۷ | ۹ | ۹ |
| ۸ | ۹ | ۱۰ |

جدول ۱۰، نتایج حاصل از حل مسائل عددی در ابعاد متوسط را نشان می‌دهد. لازم به ذکر است که حل مسائل عددی در ابعاد متوسط با استفاده از نرم‌افزار GAMS در مدت‌زمان مناسب مقدور نبود و بنابراین جدول ۱۰، صرفاً شامل نتایج حل مسائل عددی با الگوریتم‌های فراابتکاری است.

جدول ۱۰. نتایج حل مسائل عددی در ابعاد متوسط

| مثال | مقدار تابع هدف ۱ | | مقدار تابع هدف ۲ | | زمان |
|------|------------------|---------|------------------|---------|---------|
| | MOPSO | NSGA-II | MOPSO | NSGA-II | |
| ۶ | ۹/۷۱۸۷ | ۹/۷۱۸۷ | ۳/۱۲ | ۳/۱۲ | ۵/۷۵۵۶ |
| ۷ | ۹/۶۲۴ | ۹/۶۲۴ | ۳/۲۸ | ۳/۲۸ | ۶/۲۱۲۱ |
| ۸ | ۱۶/۸۲۵۱ | ۱۶/۸۲۵۱ | ۴/۱۱ | ۴/۱۱ | ۶/۱۹۴ |
| ۹ | ۱۳/۲۴۸۲ | ۱۳/۲۴۸۲ | ۳/۸۸ | ۳/۸۸ | ۱۳/۷۸۷۵ |
| ۱۰ | ۱۴/۳۳۳۶ | ۱۴/۳۳۳۶ | ۳/۹۸ | ۳/۹۸ | ۱۳/۹۵۴۷ |

مسائل بزرگ. جدول ۱۱، ابعاد مسائل عددی در ابعاد بزرگ را نشان می‌دهد.

جدول ۱۱. ابعاد مسائل بزرگ عددی طراحی شده

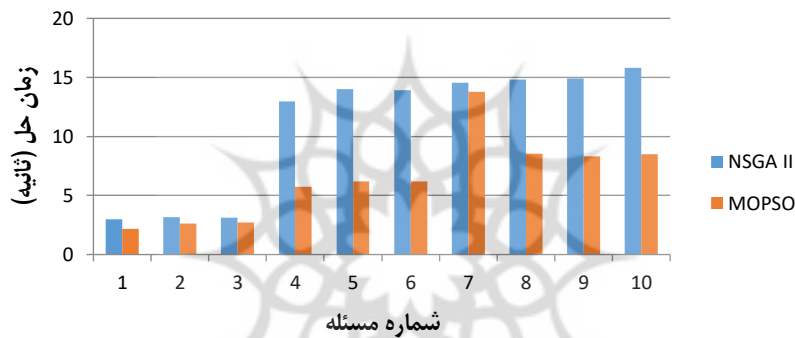
| اندیس‌ها | | مثال |
|----------|-------|------|
| $ M $ | $ I $ | |
| ۸ | ۱۰ | ۱۱ |
| ۹ | ۱۱ | ۱۲ |
| ۱۰ | ۱۳ | ۱۳ |
| ۱۱ | ۱۳ | ۱۴ |
| ۱۱ | ۱۴ | ۱۵ |

جدول ۱۲، نتایج حاصل از حل مسائل عددی در ابعاد بزرگ را نشان می‌دهد.

جدول ۱۲. نتایج حل مسائل عددی در ابعاد بزرگ

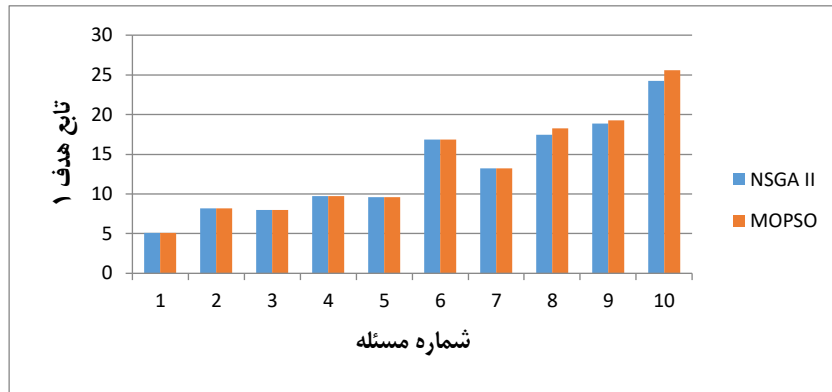
| مثال | مقدار تابع هدف ۱ | | مقدار تابع هدف ۲ | | زمان |
|------|------------------|---------|------------------|---------|--------|
| | MOPSO | NSGA-II | MOPSO | NSGA-II | |
| ۱۱ | ۱۸/۲۵۳۱ | ۱۷/۴۷۳۸ | ۴/۳۳ | ۴/۵۶ | ۸/۵۳۰۹ |
| ۱۲ | ۱۹/۲۵۱۲ | ۱۸/۸۶۴۱ | ۴/۶۲ | ۴/۸۸ | ۸/۳۰۶۹ |
| ۱۳ | ۲۵/۵۹۴۹ | ۲۴/۲۰۷۴ | ۵/۱۰ | ۵/۳۴ | ۸/۴۹۰۷ |
| ۱۴ | ۲۶/۱۴۷۷ | ۲۵/۱۱۷۸ | ۵/۸۴ | ۶/۱۴ | ۹/۰۱۴۷ |
| ۱۵ | ۲۶/۱۹۵۵ | ۲۶/۳۹۸۷ | ۶/۱۱ | ۶/۹۸ | ۹/۵۹۸۳ |

شکل ۸، مقایسه رویکردهای حل از نظر زمان حل موردنیاز را نشان می‌دهد. همان‌گونه که ملاحظه می‌شود، الگوریتم NSGA-II زمان حل بیشتری نسبت به الگوریتم MOPSO نیاز دارد؛ از این رو الگوریتم MOPSO از منظر زمان حل کارکرد مناسب‌تری را نسبت به الگوریتم NSGA-II از خود نشان می‌دهد.



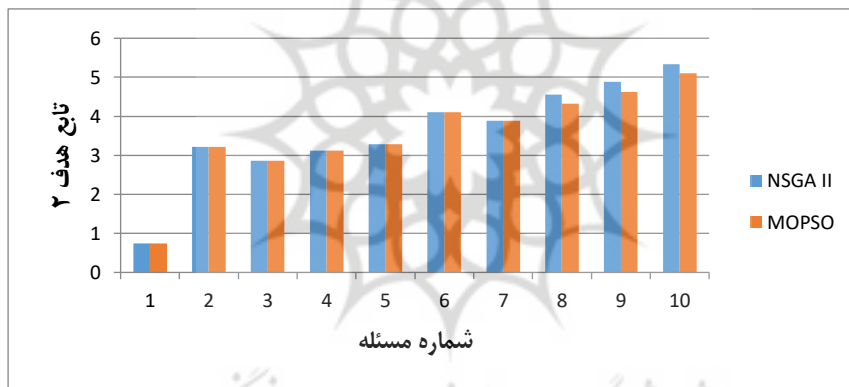
شکل ۸. مقایسه الگوریتم‌های حل از نظر زمان حل

شکل ۹، مقایسه الگوریتم‌ها از منظر تابع هدف اول را نشان می‌دهد. همان‌طور که ملاحظه می‌شود، الگوریتم NSGA-II نسبت به الگوریتم MOPSO مقدار تابع هدف شماره یک کمتری دارد. با توجه به اینکه تابع هدف ۱ از نوع کمینه‌سازی است، الگوریتم NSGA-II نسبت به الگوریتم MOPSO، عملکرد بهتر و نزدیک‌تر به مقدار بهینه را از خود نشان داده است.



شکل ۹. مقایسه الگوریتم‌های حل از نظر تابع هدف اول

در شکل ۱۰، مقایسه رویکردهای حل مسئله از منظر تابع هدف دوم نشان داده شده است. همان‌طور که ملاحظه می‌شود، الگوریتم NSGA-II نسبت به الگوریتم MOPSO مقدار تابع هدف ۲ بیشتری در غالب مسائل دارد. با توجه به اینکه تابع هدف ۲ از نوع بیشینه‌سازی است، الگوریتم NSGA-II نسبت به الگوریتم MOPSO، عملکرد بهتر و نزدیک‌تر به مقدار بهینه را از خود نشان داده است.



شکل ۱۰. مقایسه الگوریتم‌های حل از نظر تابع هدف دوم

از آنجاکه الگوریتم NSGA-II جواب‌هایی بهتر و نزدیک‌تر به پاسخ بهینه را نسبت به الگوریتم MOPSO ایجاد می‌کند، برای این مدل، الگوریتم NSGA-II الگوریتم حل مناسب‌تری است؛ البته این مسئله از قبل نیز قابل پیش‌بینی بود؛ چراکه این الگوریتم در حل مسائل عددی با متغیرهای باینری نسبت به سایر الگوریتم‌ها عملکرد مناسب‌تری را از خود نشان

داده بود. هرچند الگوریتم MOPSO از نظر زمان حل کمی برتر از الگوریتم NSGA-II است، این برتری قابل توجه نیست و نمی‌توان آن را برتری چشمگیری برای MOPSO تلقی کرد.

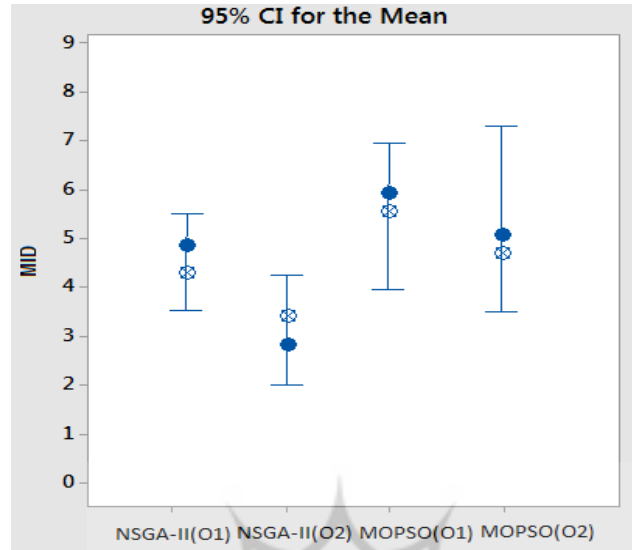
مقایسه عملکرد الگوریتم‌ها. در این بخش کارایی الگوریتم‌های ارائه شده در بخش قبل، یعنی الگوریتم NSGA-II و MOPSO مقایسه می‌شود. برای مقایسه عملکرد الگوریتم‌ها از معیار فاصله از نقطه ایده‌آل (MID) استفاده می‌شود. از آنجاکه در این پژوهش هدف رسیدن به بهترین مقدار تابع هدف و در نتیجه کمترین فاصله تا نقطه ایده‌آل است، از شاخص فاصله از نقطه ایده‌آل برای مقایسه عملکرد الگوریتم‌ها استفاده خواهد شد. طبق ۱۰ مسئله عددی طراحی شده در بخش قبل، به حل آن‌ها و مقایسه عملکرد الگوریتم‌ها در به دست آوردن توابع هدف اول و دوم پرداخته خواهد شد. در ادامه با استفاده از شاخص فاصله از نقطه ایده‌آل دو روش ارائه شده در پژوهش تحلیل و مقایسه خواهد شد.

جدول ۱۳. MID حاصل از بهترین جواب ۱۰ آزمون عددی

| مسئله | MOPSO(O2) | MOPSO(O1) | NSGA-II (O2) | NSGA-II (O1) |
|-------|-----------|-----------|--------------|--------------|
| ۱ | ۱/۵۲ | ۱/۷۸ | ۱/۲۵ | ۱/۱۵ |
| ۲ | ۱/۶۹ | ۱/۹۲ | ۱/۶۵ | ۱/۵۳ |
| ۳ | ۱/۸۲ | ۲/۲۶ | ۱/۸۴ | ۱/۹۵ |
| ۴ | ۱/۷۵ | ۲/۴۵ | ۱/۶۱ | ۲/۱۶ |
| ۵ | ۱/۵۱ | ۲/۴۷ | ۱/۴۲ | ۲/۱۷ |
| ۶ | ۱/۴۵ | ۲/۱۱ | ۱/۴۲ | ۲/۹۴ |
| ۷ | ۱/۶۹ | ۱/۳۶ | ۱/۵۵ | ۱/۰۶ |
| ۸ | ۱/۸۵ | ۱/۴۷ | ۱/۸۹ | ۱/۳۲ |
| ۹ | ۱/۹ | ۱/۱۵ | ۱/۶۶ | ۱/۰۸ |
| ۱۰ | ۱/۱۹ | ۱/۴۶ | ۱/۰۳ | ۱/۵۲ |

با توجه به جدول ۱۳، می‌توان نتیجه گرفت که مقدار شاخص فاصله از نقطه ایده‌آل در الگوریتم NSGA-II کمتر از مقدار شاخص حاصل از حل مسئله با روش MOPSO است؛ به استثنای برخی از مسئله‌های عددی که به مقدار اندکی بیشتر است. با این تفاسیر می‌توان نتیجه گرفت که الگوریتم NSGA-II دارای عملکرد بهتری نسبت به MOPSO برای بهینه‌سازی مسئله برنامه‌ریزی یکپارچه زمان‌بندی و نگهداری و تعمیرات با فرض عدم قطعیت و احتمال اختلال در تسهیلات است.

نتایج جدول ۱۳ را می‌توان در شکل ۱۱، مشاهده کرد که نشان‌دهنده برتری الگوریتم NSGA-II نسبت به روش MOPSO برای حداقل کردن زمان تکمیل وزنی محصولات و حداکثرسازی قابلیت اطمینان در خط تولید است.



شکل ۱۱. نمودارهای فاصله‌های الگوریتم NSGA-II و روش حل MOPSO برای شاخص MID

در ادامه با استفاده از شاخص تعداد جواب‌های پارتو به دو روش ارائه‌شده در پژوهش تحلیل و مقایسه می‌شود.

جدول ۱۴. NPS حاصل از بهترین جواب ۱۰ آزمون عددی

| NSGA-II (O1) | NSGA-II (O2) | MOPSO(O1) | MOPSO(O2) | مسئله عددی |
|--------------|--------------|-----------|-----------|------------|
| ۳ | ۵ | ۶ | ۷ | ۱ |
| ۷ | ۶ | ۳ | ۵ | ۲ |
| ۶ | ۹ | ۵ | ۷ | ۳ |
| ۵ | ۸ | ۷ | ۷ | ۴ |
| ۷ | ۴ | ۵ | ۶ | ۵ |
| ۶ | ۸ | ۴ | ۴ | ۶ |
| ۹ | ۷ | ۳ | ۶ | ۷ |
| ۸ | ۴ | ۶ | ۴ | ۸ |
| ۵ | ۵ | ۵ | ۴ | ۹ |
| ۶ | ۵ | ۴ | ۴ | ۱۰ |

با توجه به جدول ۱۴، می‌توان نتیجه گرفت که مقدار شاخص تعداد پاسخ‌های پارتو در الگوریتم NSGA-II بیشتر از مقدار شاخص حاصل از حل مسئله با روش MOPSO است؛ به‌استثنای برخی از مسئله‌های عددی که به مقدار اندکی کمتر است. بر این اساس می‌توان عنوان کرد که الگوریتم NSGA-II دارای عملکرد بهتری نسبت به MOPSO برای بهینه‌سازی مسئله برنامه‌ریزی یکپارچه زمان‌بندی و نگهداری و تعمیرات با فرض عدم قطعیت و احتمال اختلال در تسهیلات است.

با توجه به اینکه کارکرد ماشین‌ها و مقدار استهلاک آن‌ها با توجه به مقدار استفاده و عمرشان با هم متفاوت است، مدیران در جست‌وجوی راهکارهایی برای کاستن از خرابی‌های ناگهانی ماشین‌آلات و افزایش دوره‌ی کارایی آن‌ها هستند. در این پژوهش با طراحی مدلی یکپارچه برای زمان‌بندی و برنامه‌ریزی نگهداری و تعمیرات سعی شد تا زمان‌بندی بهینه‌ی انجام فعالیت‌ها، تخصیص فعالیت‌ها به ماشین‌ها و همچنین برنامه‌ریزی نگهداری و تعمیرات ماشین‌آلات انجام شود تا با پرداخت کمترین هزینه بتوان کارایی حداکثری را از این ماشین‌آلات کسب کرد. کاربرد این پژوهش به‌طور ویژه در راستای یاری مدیران و متصدیان در شرکت‌های تولیدی است که می‌توانند از یافته‌های این پژوهش به‌عنوان ابزاری کاربردی و مفید در تصمیم‌ها و برنامه‌ریزی‌های خود استفاده کنند.

۵. نتیجه‌گیری و پیشنهادها

هدف از این پژوهش، ارائه مدلی یکپارچه به‌منظور برنامه‌ریزی تولید و زمان‌بندی نگهداری و تعمیرات با فرض عدم قطعیت و احتمال اختلال در تسهیلات بود. مدل دارای دو تابع هدف اصلی شامل حداقل کردن زمان وزنی تکمیل کارها و حداکثرسازی قابلیت اطمینان ماشین‌آلات مورد استفاده است. پس‌ازاینکه مدل پژوهش ارائه شد، در گام فازی‌زدایی از مدل مسئله صورت گرفت. در این مرحله، مدل فازی‌زدایی‌شده پژوهش با استفاده از رویکرد جیمز و همکاران (۲۰۰۷) ارائه شد.

گام بعدی در راستای انجام پژوهش، انجام آزمون‌های عددی بود. در گام نخست، دو الگوریتم NSGA-II و MOPSO برای حل مسائل عددی در ابعاد مختلف تشریح شدند؛ سپس نحوه آماده‌سازی الگوریتم‌ها تشریح شد. در گام بعدی چند سری داده تعریف و نحوه تولید داده‌های تصادفی ارائه شد. مسائل در ابعاد مختلف حل شدند و نتایج حاکی از توانایی مدل و الگوریتم در حل مسائل مختلف بود؛ همچنین مشخص شد که الگوریتم NSGA-II نسبت به MOPSO دارای کارکرد بهتری از نظر نزدیکی به جواب بهینه است. نتایج ارائه‌شده حکایت از کارایی مدل و راحت حل شدن آن در ابعاد مختلف دارد؛ همچنین سعی شد تا محدودیت‌ها و پارامترهایی وارد مدل شوند که آن را هر چه بیشتر با شرایط دنیای واقعی تطابق دهند. هر

پژوهشی می‌تواند زمینه‌ساز پژوهش‌های جامع‌تر و قدرتمندتر شود. پژوهش حاضر نیز از این مسئله مستثنا نیست و می‌توان از آن به‌منظور انجام پژوهش‌هایی غنی‌تر استفاده کرد. در راستای انجام این پژوهش، برخی محدودیت‌ها وجود داشت که ضرورت دارد در پژوهش‌های آتی نسبت به رفع آن‌ها اقدام شود. یکی از این محدودیت‌ها، نبود داده‌های مربوط به نمونه‌ها و مسائل دنیای واقعی بود. به‌منظور بررسی هرچه‌بهتر نحوه کارکرد مدل در شرایط دنیای واقعی، وجود نمونه‌های موردی و اجرای مدل بر روی آن‌ها می‌توانست تحقیق حاضر را کامل‌تر کند که به‌دلیل نبود چنین مجموعه داده‌ای، امکان تحقق آن فراهم نشد. از سوی دیگر، به‌منظور ساده‌تر شدن مسئله و امکان مدل‌سازی و حل آن، برخی مفروضات ساده‌کننده در نظر گرفته شدند. در نظر گرفتن این مفروضات به‌منظور مدل‌سازی راحت‌تر مسئله را نیز می‌توان از جمله محدودیت‌های این پژوهش دانست.

با توجه به نتایج این پژوهش، موارد زیر می‌تواند زمینه مطالعات آتی را فراهم کند:

- از الگوریتم‌های شبکه‌های عصبی و شبیه‌سازی تبرید برای حل مسئله استفاده و نتایج مقایسه شود؛

- نتایج روی نمونه موردی اجرا شود و قابلیت کاربرد مدل با آن موردسنجش قرار گیرد؛

- از رویکردهای دیگر برای در نظر گرفتن عدم قطعیت در تسمیلات استفاده شود.

منابع

1. Abbassi, R., Arzaghi, E., Yazdi, M., Aryai, V., Garaniya, V., & Rahnamayiezekavat, P. (2022). Risk-based and predictive maintenance planning of engineering infrastructure: existing quantitative techniques and future directions. *Process Safety and Environmental Protection*, Volume 165, Pages 776-790.
2. Aghaee, M., & Fazli, S. (2012). Applying a Hybrid DEMATEL and ANP Approach for Suitable Maintenance Approach Selection (Case Study: Work Vehicle Industry). *The Journal of Industrial Manajement Perspective*, 6, 89-107. (In Persian)
3. Amiri, S., & Honarvar, M. (2018). Providing an integrated Model for Planning and Scheduling Energy Hubs and preventive maintenance. *Energy*, 163, 1093-1114.
4. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans Evolutionary Computing*, Indian, 6(2), 182-197.
5. Ertem, M., As' ad, R., Awad, M., & Al-Bar, A. (2022). Workers-constrained shutdown maintenance scheduling with skills flexibility: Models and solution algorithms. *Computers & Industrial Engineering*, 172, 108575.
6. Fitouhi, M. C., & Nourelfath, M. (2014). Integrating noncyclical preventive maintenance scheduling and production planning for multi-state systems. *Reliability Engineering & System Safety*, 121, 175-186.
7. Ghalami, L., & Grosu, D. (2019). Scheduling parallel identical machines to minimize makespan: A parallel approximation algorithm. *Journal of Parallel and Distributed Computing*, 133, 221-231.
8. Jafar-Zanjani, H., Zandieh, M., & Sharifi, M. (2022). Robust and resilient joint periodic maintenance planning and scheduling in a multi-factory network under uncertainty: A case study. *Reliability Engineering & System Safety*, 217, 108113.
9. Jiménez, M., Arenas, M., Bilbao, A., & Rodríguez, M.V. (2007). Linear programming with fuzzy parameters: An interactive method resolution. *European Journal of Operational Research*, 177(3), 1599-1609.
10. Liu, X., Wang, W., & Peng, R. (2017). An integrated preventive maintenance and production planning model with sequence-dependent setup costs and times. *Quality and Reliability Engineering International*, 33(8), 2451-2461.
11. Miyata, H. H., Nagano, M. S., & Gupta, J. N. (2019). Integrating preventive maintenance activities to the no-wait flow shop scheduling problem with dependent-sequence setup times and makespan minimization. *Computers & Industrial Engineering*, 135, 79-104.
12. Nadizadeh Ardakani, A., Ranjbar, H., & Moubed, M. (2020). Periodic Inspection Optimization for a TwoComponent System with Dependent Failures. *The Journal of Industrial Manajement Perspective*, 38, 83-110. (In Persian)
13. Nattaf, M., Dauzère-Pérès, S., Yugma, C., & Wu, C. H. (2019). Parallel machine scheduling with time constraints on machine qualifications. *Computers & Operations Research*, 107, 61-76.
14. Ruiz R., & Maroto C. (2016). A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *European Journal of Operational Research*, 169(3), 781-800.

15. Safarzadeh, H., & Akhavan Niaki, S. T. (2019). Bi-objective green scheduling in uniform parallel machine environments. *Journal of Cleaner Production*, 217, 559-572.
16. Sajedinejad, A., & Lotfi, M. (2019). Providing a model to optimize preventive maintenance schedules for multi-component systems using GA. *Industrial Management Studies*, 55, 137-160. (In Persian)
17. Shaukat, S., Katscher, M., Wu, C.L., Delgado, F., Larrain, H. (2020) Aircraft line maintenance scheduling and optimization. *Journal of Air Transport Management*, 89, 101914.
18. Shayanian, M., & Behnamian, J. (2017). Discrete Particle Swarm Optimization for Job Shop Scheduling Problem with Parallel Machine. *International Journal of Industrial Engineering & Production Management*, 28(1), 15-26. (In Persian)
19. Shen, L., Dauzère-Pérès, S., & Neufeld, J. S. (2018). Solving the flexible job shop scheduling problem with sequence-dependent setup times. *European Journal of Operational Research*, 265(2), 503-516.
20. Sherafat, A., Mohaghar, A., Karimi, F., & Davoodi, S.M.R. (2018). Designing the Mechanism for Choosing the Appropriate Maintenance Strategy. *The Journal of Industrial Manajement Perspective*, 6, 31-69. (In Persian)
21. Soper, A. J., & Strusevich, V. A. (2019). Schedules with a single preemption on uniform parallel machines. *Discrete Applied Mathematics*, 261, 332-343.
22. Wang, T., Baldacci, R., Lim, A., & Hu, Q. (2018). A branch-and-price algorithm for scheduling of deteriorating jobs and flexible periodic maintenance on a single machine. *European Journal of Operational Research*, 271(3), 826-838.
23. Wang, S., & Ye, B. (2019). Exact methods for order acceptance and scheduling on unrelated parallel machines. *Computers & Operations Research*, 104, 159-173.
24. Yang, L., Ma, X., Peng, R., Zhai, Q., & Zhao, Y. (2017). A preventive maintenance policy based on dependent two-stage deterioration and external shocks. *Reliability Engineering & System Safety*, 160, 201-211.
25. Zhong, S., Pantelous, A. A., Goh, M., & Zhou, J. (2019). A reliability-and-cost-based fuzzy approach to optimize preventive maintenance scheduling for offshore wind farms. *Mechanical Systems and Signal Processing*, 124, 643-663.



پروہشگاہ علوم انسانی و مطالعات فرہنگی
پرتال جامع علوم انسانی