



# Solving A New Bi-Objective Open Shop Scheduling Problem with Machine Availability and Transportation Time by Two Multi-Objective Meta-Heuristic Algorithms

Abolfazl Shojaei-Barjouei & Reza Tavakkoli-Moghaddam\*

Abolfazl Shojaei-Barjouei, M.Sc. of Industrial Engineering, Mazandaran University of Science and Technology  
Reza Tavakkoli-Moghaddam, Professor of Industrial Engineering, College of Engineering, University of Tehran

## Keywords

Open shop scheduling,  
Machine availability,  
Transportation time,  
Makespan,  
Total tardiness,  
Multi-objective parallel  
meta-heuristics,  
Taguchi.

## ABSTRACT

This paper presents a new bi-objective mixed-integer linear programming (MOMILP) model for a non-preemptive open shop scheduling problem, in which machines are not available to process jobs on known periodic interval times. Transportation time between machines is considered that can be different from one job to another. The objectives of the presented model are to minimize the makespan and the total tardiness, simultaneously. To validate the presented model, a number of test problems are generated randomly and solved globally for small-sized instances. In order to solve large-sized problems, which often occur in real world, two meta-heuristic algorithms, namely multi-objective parallel genetic algorithm (MOPGA) and multi-objective parallel simulated annealing (MOPSA) are proposed. The results show that the MOPSA has a high convergence rate (i.e., speed); however, MOPGA is better in terms of efficiency in solving large-sized problems. The parameters of these proposed algorithms are tuned by the Taguchi design of experiments. Finally, the performance of the algorithms is validated by some large-sized instances and the related results are illustrated and discussed.

© 2017 IUST Publication, IJIEPM Vol. 28, No. 3, All Rights Reserved



## حل مسأله زمان بندی کارگاه باز دو هدفه جدید با در دسترس بودن ماشین ها و زمان های حمل و نقل با استفاده از دو الگوریتم فراابتکاری موازی چند هدفه

ابوالفضل شجاعی برجویی و رضا توکلی مقدم\*

### چکیده:

در این مقاله، یک مدل برنامه ریزی ریاضی جدید دو هدفه خطی و عدد صحیح مختلط برای مسأله زمان بندی کارگاه باز بدون انقطاع با در نظر گرفتن محدودیت در دسترس بودن ماشین ها ارائه می شود. محدودیت در دسترس بودن در زمان های مشخص روی ماشین ها اتفاق می افتد و طول مدت آن برای ماشین های مختلف متفاوت ولی ثابت است. بعلاوه، زمان های حمل و نقل نامتقارن و وابسته به کار بین ماشین ها در نظر گرفته می شود. حداکثر زمان تکمیل کارها و کل دیرکرد به عنوان توابع هدف همزمان کمینه می شوند. برای ارزیابی مدل ریاضی ارائه شده، چند مسأله با ابعاد کوچک که به صورت تصادفی تولید شده اند، بطور دقیق حل می شوند. به علاوه حساسیت این مدل ریاضی به پارامترهای ورودی مورد بررسی قرار می گیرد. برای حل مسائل با ابعاد بزرگ، دو الگوریتم فراابتکاری شامل الگوریتم ژنتیک موازی چند هدفه (MOPGA) و الگوریتم شبیه سازی تبرید موازی چند هدفه (MOPSA) طراحی می شوند. نتایج نشان می دهد هر چند الگوریتم MOPSA سرعت بیشتری دارد، الگوریتم MOPGA کارایی بهتری نسبت به آن در برخورد با مسأله های با ابعاد بزرگ دارد. همچنین روش طراحی آزمایشات تاگوچی جهت تنظیم پارامترهای الگوریتم ها و پایا نمودن آنها بکار گرفته می شود. در خاتمه، جهت ارزیابی عملکرد الگوریتم های طراحی شده، مسأله هایی با ابعاد بزرگ که به صورت تصادفی تولید شده اند، حل می شوند و نتایج مربوطه ارائه و بحث می شود.

### کلمات کلیدی

زمان بندی کارگاه باز،  
در دسترس بودن ماشین ها،  
زمان حمل و نقل،  
حداکثر زمان تکمیل کارها،  
کل دیرکرد،  
الگوریتم های فراابتکاری  
موازی چند هدفه،  
تاگوچی .

### ۱. مقدمه

مسأله زمان بندی کارگاه باز (OSSP<sup>1</sup>) شامل  $m$  ماشین  $(M_1, M_2, \dots, M_m)$  برای انجام  $n$  کار  $(J_1, J_2, \dots, J_n)$  است که هر کار در برگیرنده  $m$  عملیات<sup>۱</sup> است [۱].  $j$ -امین عملیات کار  $J_j$  با  $O_{ij}$  نشان داده می شود که باید بمدت  $p_{ij} \geq 0$  واحد زمانی روی ماشین  $M_i$  پردازش شود. بنابراین، در مجموع تعداد  $n \times m$  عملیات خواهیم داشت  $(O_{11}, O_{12}, \dots, O_{nm})$ . خدمات پیش گیرانه و داروخانه ها را به صورت متمرکز ارائه می دهند.

تاریخ وصول: ۹۳/۰۴/۲۸

تاریخ تصویب: ۹۴/۰۸/۱۰

ابوالفضل شجاعی برجویی، دانش آموخته کارشناسی ارشد گروه مهندسی صنایع،  
دانشگاه علوم و فنون مازندران، [shojaei.b@gmail.com](mailto:shojaei.b@gmail.com)

\*نویسنده مسئول مقاله: رضا توکلی مقدم، استاد دانشکده مهندسی صنایع،  
پردیس دانشکده های فنی، دانشگاه تهران، [tavakoli@ut.ac.ir](mailto:tavakoli@ut.ac.ir)

تفاوت اصلی بین مسأله های جریان کارگاهی و کارگاه باز اینست که در جریان کارگاهی عملیات های هر کار باید بترتیب پردازش شوند. یعنی، امکان شروع پردازش عملیات  $O_{i,j}$  تا زمانی که عملیات  $O_{i-1,j}$  تمام نشود وجود ندارد ( $1 < i < m$ ). این در حالی است که ترتیب پردازش عملیات ها در کارگاه باز بصورت دلخواه است [۲]. در مسأله کار کارگاهی، تعداد عملیات های هر کار به  $m$  محدود نبوده و هر تعدادی می تواند باشد. برخلاف جریان کارگاهی و کارگاه باز که  $i$ -امین عملیات هر کار باید توسط ماشین  $M_i$  پردازش شود، در کار کارگاهی چنین محدودیتی وجود نداشته و هر عملیات از هر کار به یکی از ماشین ها اختصاص می یابد. اما همانند جریان کارگاهی، در کار کارگاهی نیز ترتیب پردازش عملیات ها مشخص است. بنابراین، می توان اینگونه برداشت نمود که مسأله کارگاه باز همانند مسأله جریان کارگاهی است با افزودن این انعطاف که ترتیب پردازش عملیات ها دلخواه است.

دیگر مواردی است که در مسأله‌های زمان‌بندی مورد توجه محققین قرار گرفته‌است. استراسویچ [۱۰] در مسأله کارگاه باز دو ماشینه یک فاصله زمانی بین اتمام پردازش فعالیت مربوط به یک کار روی ماشینی و شروع فعالیت بعدی همان کار روی ماشین دیگر متصور شد که بدلیل وجود حمل و نقلی که در شرایط واقعی اتفاق می‌افتد آن را زمان‌های حمل و نقل نامید. همچنین، وی اشاره می‌کند که در کاربردهای شیمیایی و متالورژیک این زمان‌های حمل و نقل معادل زمان‌های فرآیندهای گرم کردن و یا سرد کردن هستند.

معیارهای اندازه‌گیری مختلفی جهت بهینه شدن بصورت یک تابع هدف منفرد و یا چند تابع هدف همزمان در مسأله‌های زمان‌بندی توسط محققان مورد بررسی قرار گرفته‌اند. از آن جمله می‌توان به کار پناهی و همکاران [۱۱] اشاره نمود که مجموع تأخیرها و طولانی‌ترین زمان تکمیل را بعنوان توابع هدف کمینه‌سازی در محیط کارگاه باز بررسی کردند. جانگواناکیوت و همکاران [۱۲] ترکیب محدب دو هدف طولانی‌ترین زمان تکمیل کارها و تعداد کارهای تأخیری را بعنوان توابع کمینه‌سازی در مسأله جریان کارگاهی ترکیبی در نظر گرفتند. پناهی و توکلی مقدم [۱۳] کمینه سازی همزمان دو هدف شامل طولانی‌ترین زمان تکمیل کارها و مجموع تأخیرها را در مسأله کارگاه باز در نظر گرفته و از الگوریتم بهینه‌سازی جمعیت مورچگان ترکیبی<sup>۳</sup> جهت حل آن در زمان قابل قبول استفاده کردند. امیری و همکاران [۱۴] حداقل نمودن حداکثر زمان تکمیل کارها، مجموع زمان دیرکرد و زودکرد کارها و مجموع هزینه راه‌اندازی کارها را در یک کارگاه باز مورد نظر قرار دادند.

مسأله زمان‌بندی کارگاه باز بعنوان یک مسأله NP-hard شناخته شده است [۱۷-۱۵] و امکان حل دقیق مسأله‌های آن در زمان چندجمله‌ای مگر در ابعاد کوچک وجود ندارد. بنابراین، دستاوردهای حل تقریبی شامل روش‌های ابتکاری و فرا ابتکاری می‌توانند کارایی بیشتری نسبت به روش‌های دقیق داشته باشند. پناهی و همکاران [۱۱] الگوریتم شبیه‌سازی تبرید چند هدفه و نیز الگوریتم بهینه‌سازی جمعیت مورچگان (ACO<sup>۴</sup>) را در مسأله کارگاه باز چند هدفه با هدف‌های مجموع تأخیرها و طولانی‌ترین زمان تکمیل بکار گرفتند. پناهی و توکلی مقدم [۱۳] الگوریتم‌های شبیه‌سازی تبرید عمومی و نیز شبیه‌سازی تبرید چند هدفه را در بهینه‌سازی جمعیت مورچگان ترکیب کرده و آن را در بهینه‌سازی همزمان دو تابع هدف شامل طولانی‌ترین زمان تکمیل کارها و مجموع تأخیرها در مسأله کارگاه باز بکاربردند. فیلی و همکاران [۱۸] از الگوریتم‌های ژنتیک و شبیه‌سازی تبرید در مسأله زمان بندی دسته‌ای در حالت ماشین‌های موازی چند کاره بهره بردند. در این مقاله یک مسأله کارگاه باز مورد بررسی قرار می‌گیرد که در آن توقفات در زمان‌های مختلفی روی ماشین‌ها اتفاق می‌افتد و طول مدت آن برای ماشین‌های مختلف متفاوت ولی مشخص است.

بیشتر مطالعه‌های صورت گرفته بر روی مسأله‌های زمان‌بندی تحت این فرض اساسی که ماشین‌ها در تمام طول افق برنامه‌ریزی در دسترس هستند انجام شده‌است در حالی که در بسیاری موارد در دنیای واقعی این چنین نیست. ماشین‌ها ممکن است بدلیلی همچون نگهداری و تعمیرات پیشگیرانه، خرابی، دوره استراحت، کارهای باقیمانده از دوره برنامه‌ریزی قبلی که در ابتدای دوره جدید باید پردازش شوند و غیره، در دسترس نباشند [۱]. در بسیاری موارد، این زمان‌های عدم دسترسی از قبل معلوم هستند و بنابراین تصمیم‌گیرنده با در نظر گرفتن آن‌ها می‌تواند تصمیم کاراتری اتخاذ نماید. کلو و همکاران [۳] یک زمان از پیش تعیین شده در دسترس نبودن ماشین در مسأله تک ماشینه در نظر گرفته و دو روش ابتکاری برای حل آن ارائه دادند. بریت و همکاران [۴] مسأله کارگاه باز دو ماشینه را با شرط وجود یک زمان توقف از پیش تعیین شده روی یکی از ماشین‌ها بررسی کرده و یک الگوریتم ابتکاری ارائه دادند. آن‌ها همچنین در کار بعدی خود [۵] چند زمان توقف از پیش تعیین شده را برای هر دو ماشین در نظر گرفته و مجدداً از الگوریتم‌های ابتکاری جهت حل تقریبی مسأله بهره بردند. هسو و همکاران [۶] دو رویکرد متفاوت نگهداری و تعمیرات دوره‌ای را بطور همزمان در مسأله زمان‌بندی تک ماشینه مورد بررسی قرار دادند. رویکردهای نگهداری و تعمیرات دوره‌ای در مسأله مورد نظر آن‌ها بدین ترتیب است که ماشین پس از گذشت مدت زمان مشخص ( $T$ ) و یا پس از پردازش تعداد مشخصی کار ( $K$ ) بمنظور انجام فرآیند نگهداری و تعمیرات متوقف می‌شود. بدیهی است هر یک از دو شرط مذکور زودتر رخ دهد موجب توقف ماشین خواهد شد. سان و لی [۷] دو ماشین موازی یکسان را که در آن بدلیل کاهش احتمال خرابی ماشین‌ها یک حداکثر زمان مشخص برای پردازش پیوسته آن‌ها در نظر گرفته شده، بررسی کردند. گل‌مکانی و نمازی [۸] محدودیت در دسترس نبودن ماشین‌ها را در یک محیط کار کارگاهی چند مسیره بررسی نمودند. ما و همکاران [۹] خلاصه‌ای از مسأله‌های زمان‌بندی با محدودیت عدم دسترسی ماشین‌ها ناشی از فعالیت‌های نگهداری و تعمیرات پیشگیرانه ارائه دادند. بعلاوه، آن‌ها بطور مختصر به تشریح پیچیدگی مسأله‌ها، روش‌های حل دقیق و تقریبی در محیط‌های زمان‌بندی مختلف شامل تک ماشینه، ماشین‌های موازی، جریان کارگاهی، کارگاه باز و کار کارگاهی با معیارهای اندازه‌گیری متفاوت پرداختند.

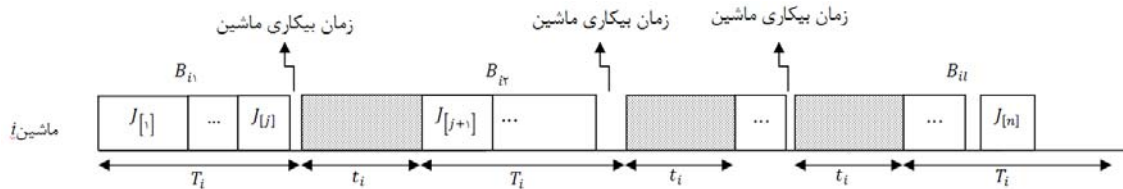
از دیگر شرایطی که در صورت وجود محدودیت در دسترس بودن ماشین‌ها مطرح می‌شود، مجاز نبودن بریدگی کارهاست که به سبب آن در صورت نبود زمان کافی برای انجام یک عملیات و مواجه با توقف ماشین، عملیات مورد نظر روی ماشین بارگذاری نشده و پردازش آن به بعد از اتمام توقف ماشین موکول می‌شود. در نظر گرفتن زمان‌های حمل و نقل و جابجایی بین ماشین‌ها از

مسأله‌هایی که بصورت تصادفی تولید شده‌اند جهت ارزیابی عملکرد مدل ریاضی و الگوریتم‌های طراحی شده استفاده می‌شود. در ادامه، در بخش ۲ مدل برنامه‌ریزی ریاضی خطی آمیخته ارائه می‌شود. الگوریتم‌های فرا ابتکاری پیشنهادی و جزئیات آن در بخش ۳ تشریح می‌شود. بخش ۴ طراحی آزمایشات و ارزیابی محاسباتی را توضیح می‌دهد. بخش ۵ جمع‌بندی و مطالعات آتی را ارائه می‌دهد.

## ۲. مدل ریاضی

در این بخش، مسأله موردنظر با استفاده از فرض‌های ساده‌ساز و نیز نمادهای ریاضی معرفی و یک مدل برنامه‌ریزی خطی آمیخته چند هدفه جهت حل دقیق مسأله‌های با ابعاد کوچک ارائه و تحلیل می‌شود.

بمنظور دستیابی به مدل ریاضی، زمان در دسترس هر ماشین بعنوان یک بسته با طول مشخص  $T_i$  در نظر گرفته می‌شود که کارها باید طوری در آن قرار بگیرند که مجموع زمان پردازش آن‌ها از طول بسته تجاوز نکنند. برای روشن شدن موضوع به شکل (۱) نگاه کنید که در آن  $J[j]$  نشان‌دهنده  $j$ -امین کار در توالی و  $B_{ij}$  بیانگر  $i$ -امین بسته ماشین  $i$  است.



شکل ۱. توالی کارها روی ماشین

- طول زمان‌های در دسترس بودن (و عدم دسترسی) ماشین‌ها، مشخص و به ماشین بستگی داشته و در طی دوره برنامه‌ریزی ثابت است.

### ۲-۲. اندیس‌ها و پارامترها

- $j, k$ : اندیس‌های مربوط به کار ( $j, k = 1, 2, \dots, n$ ).
- $i, h$ : اندیس‌های مربوط به ماشین ( $i, h = 1, 2, \dots, m$ ).
- $l, l'$ : اندیس‌های مربوط به بسته ( $l, l' = 1, 2, \dots, b$ ).
- $p_{ij}$ : زمان پردازش کار  $j$  روی ماشین  $i$ .
- $Tr_{ihj}$ : زمان جابجایی کار  $j$  از ماشین  $i$  به ماشین  $h$ .
- $T_i$ : طول زمان‌های در دسترس بودن ماشین  $i$ .
- $t_i$ : طول زمان‌های عدم دسترسی ماشین  $i$ .
- $d_j$ : موعد تحویل کار  $j$ .
- $M$ : یک عدد مثبت بزرگ.

### ۲-۳. متغیرهای تصمیم

- $st_{ij}$ : زمان شروع آماده‌سازی کار  $j$  روی ماشین  $i$ .
- $C_{ij}$ : زمان تکمیل کار  $j$  روی ماشین  $i$ .

توابع هدف در نظر گرفته شده شامل کمینه‌سازی طولانی‌ترین زمان تکمیل کارها ( $C_{max}$ ) و مجموع تأخیرها که بصورت ترکیب وزنی در یک تابع منفرد ارائه می‌شوند، است. همچنین، بدلیل امکان استفاده از وسایل مختلف جهت حمل و نقل اقلام مختلف در یک کارگاه، زمان جابجایی در یک مسیر ثابت برای هر کار متفاوت است. همچنین، مسیر رفت کاری از ماشینی به ماشین دیگر متفاوت از مسیر برگشت آن در نظر گرفته شده که علت آن به حداقل رساندن تداخل جریان مواد است. بنابراین، ماتریس حمل و نقل کارها بین ایستگاه‌های کاری مختلف نامتقارن است. بمنظور حل مسأله‌های با ابعاد کوچک و در شرایط معلوم بودن وزن هر یک از هدف‌ها، رویکرد دقیق شامل یک مدل ریاضی دو هدفه خطی آمیخته و رویکردهای تقریبی الگوریتم‌های فرا ابتکاری تک هدفه ژنتیک و شبیه‌سازی تبرید بکار گرفته می‌شوند. بعلاوه، در شرایط نامعلوم بودن وزن هر یک از هدف‌ها، از دستاورد جواب‌های بهینه پارتو با استفاده از الگوریتم‌های فرا ابتکاری شامل الگوریتم ژنتیک موازی چند هدفه و الگوریتم شبیه‌سازی تبرید موازی چند هدفه جهت حل مسائل با ابعاد بزرگ استفاده می‌شود. پارامترهای مربوط به الگوریتم‌های فرا ابتکاری با استفاده از روش طراحی آزمایشات تاگوشی تنظیم می‌شوند که در نتیجه آن، الگوریتم‌ها حالت پایداری در مواجهه با مسأله‌های مختلف نشان می‌دهند. در پایان از

### ۲-۱. فرض‌های مسأله

فرض‌های ساده‌ساز بشرح زیر در نظر گرفته می‌شوند:

- هر ماشین در آن واحد حداکثر یک کار را می‌تواند پردازش کند.
- هر کار در آن واحد حداکثر روی یک ماشین می‌تواند پردازش شود.
- کارها به هر توالی دلخواهی می‌توانند توسط همه یا برخی از ماشین‌ها پردازش شوند.
- تمامی کارها از ابتدای افق برنامه‌ریزی در کارگاه موجود هستند.
- بریدگی کارها در حین پردازش مجاز نیست.
- تنها یک ماشین از هر نوع در کارگاه موجود است (هر عملیات مربوط به کارها توسط تنها یک ماشین خاص انجام می‌شود).
- زمان‌های عدم دسترسی در دوره‌های زمانی مشخصی روی هر ماشین صورت می‌گیرد.

$$\begin{aligned}
 C_j: \text{ زمان تکمیل کار } j \text{ روی آخرین ماشین مطابق با توالی.} \\
 Tard_j: \text{ مدت تأخیر زمان تکمیل کار } j \text{ از موعد تحویل.} \\
 \left. \begin{aligned}
 C_{ij} - M(1 - A_{ijl}) &\leq (l-1)t_i + l.T_i & \forall i, j, l & (6) \\
 C_{ij} + M(1 - A_{ijl}) &\geq (l-1)(t_i + T_i) & \forall i, j, l & (7) \\
 M(1 - A_{ijl}) + M(1 - A_{ikl}') &+ X_{ijk} \geq 1 & \begin{aligned} &\forall i, j, k, l, l' \\ &j \neq k, \\ &l < b, l' \\ &> l \end{aligned} & (8) \\
 X_{ijk} + X_{ikj} &= 1 & \forall i, k, j; & j < n, k \\ & & & > j \\
 Y_{ihj} + Y_{hij} &= 1 & \forall i, j, h; & i < m, \\ & & & h > i \\
 \sum_{l=1}^b A_{ijl} &= 1 & \forall i, j & (11) \\
 \sum_{j=1}^n p_{ij} \cdot A_{ijl} &\leq T_i & \forall i, l & (12) \\
 C_i &\geq C_{ij} & \forall i, j & (13) \\
 C_{max} &\geq C_i & \forall i & (14) \\
 Tard_j &\geq C_j - d_j & \forall j & (15) \\
 st_{ij}, C_{ij}, C_j, C_{max}, Tard_j &\geq 0 & \forall i, j & (16) \\
 X_{ijk}, Y_{ihj}, A_{ijl} &\in \{0, 1\} & \begin{aligned} &\forall i, h, j, k, l; \\ &\neq h, j \\ &\neq k \end{aligned} & (17) \\
 st_{ij} &\geq (l-1)(T_i + t_i)A_{ijl} & \forall i, j, l & (5)
 \end{aligned}
 \right\}
 \end{aligned}$$

۲-۴. مدل برنامه‌ریزی خطی آمیخته چند هدفه

در این قسمت، با استفاده از فرض‌ها و نمادهای فوق یک مدل برنامه‌ریزی خطی آمیخته چند هدفه (MOMILP<sup>5</sup>) برای مسأله کارگاه باز با در نظر گرفتن محدودیت عدم دسترسی ماشین‌ها، زمان‌های آماده‌سازی و جداسازی وابسته به کار و ماشین و زمان‌های جابجایی کارها بین ماشین‌ها جهت کمینه کردن همزمان معیارهای طولانی‌ترین زمان تکمیل و مجموع تأخیرها بصورت یک تابع منفرد ارائه می‌کنیم. مدل MOMILP موردنظر بصورت زیر است:

$$\begin{aligned}
 \min \quad Z &= \theta_1 C_{max} + \theta_2 \sum_{j=1}^n Tard_j & (1) \\
 \text{s.t.} & \\
 st_{ij} + p_{ij} &= C_{ij} & \forall i, j & (2) \\
 st_{ij} + p_{ij} + Tr_{ihj} - & & \forall i, h, j; & \\
 M(1 - Y_{ihj}) &\leq st_{hj} & i \neq h & (3) \\
 (st_{ij} + p_{ij}) - M(1 - A_{ijl}) & & \forall i, j, k, l; & \\
 -M(1 - A_{ikl}') - M(1 - X_{ijk}) & & j \neq k & (4) \\
 & & & \leq st_{ik} \\
 st_{ij} &\geq (l-1)(T_i + t_i)A_{ijl} & \forall i, j, l & (5)
 \end{aligned}$$

معادله (۱) تابع هدف مسأله است که شامل ترکیب خطی محدب دو معیار طولانی‌ترین زمان تکمیل کارها و مجموع تأخیرها است.

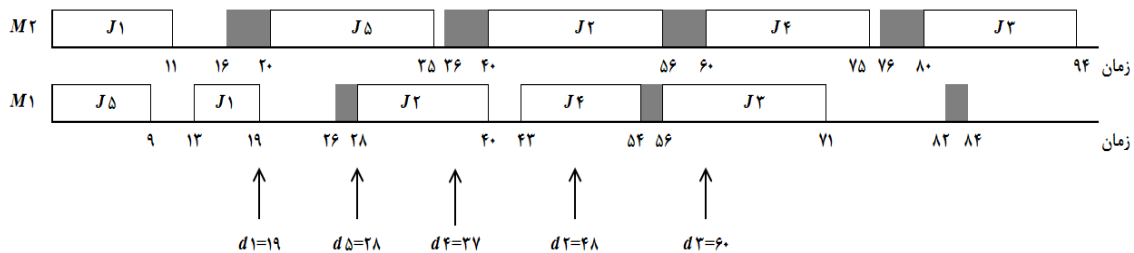
$(n = 5)$ ، ۲ ماشین  $(m = 2)$  و ۵ بسته برای هر ماشین  $(b = 5)$  است. مابقی پارامترها بقرار زیرند:

$$p_{ij} = \begin{bmatrix} 6 & 11 \\ 12 & 16 \\ 15 & 14 \\ 11 & 15 \\ 9 & 15 \end{bmatrix}, Tr_{ih_1} = \begin{bmatrix} \cdot & \cdot \\ 2 & \cdot \end{bmatrix}, Tr_{ih_2} = \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \end{bmatrix},$$

$$Tr_{ih_3} = \begin{bmatrix} \cdot & 2 \\ 2 & \cdot \end{bmatrix}, Tr_{ih_4} = \begin{bmatrix} \cdot & \cdot \\ 2 & \cdot \end{bmatrix}, Tr_{ih_5} = \begin{bmatrix} \cdot & \cdot \\ 2 & \cdot \end{bmatrix}$$

$$T_i = (26 \quad 16), t_i = (2 \quad 4), d_j = (19 \quad 48 \quad 60 \quad 37 \quad 28)$$

با در نظر گرفتن وزن هدف‌ها بصورت  $\theta_1 = \theta_2 = 0.5$ ، جواب بهینه سراسری معادل  $Z = 90.5$  از طریق حل مدل MOMILP توسط نرم‌افزار لینگو ۸ بدست می‌آید که نمودار گانت این جواب در شکل (۲) آمده است. همانگونه که در شکل پیداست، هیچ عملیاتی در زمان‌های عدم دسترسی ماشین‌ها که با رنگ خاکستری نشان داده شده‌اند واقع نگردیده است. بدین ترتیب، هدف اصلی ما از طراحی مدل برآورده شده‌است. بعلاوه، مجموع کل زمان‌های پردازش مربوط به عملیات‌های قرار گرفته در هر یک از بسته‌ها از زمان آن بسته متجاوز نیست. همچنین، توالی بدست آمده حاکی از رعایت موعد تحویل بوده بگونه‌ای که کارهای با زمان تحویل زودتر در ابتدای توالی و کارهای با زمان تحویل دیرتر در پایان توالی قرار گرفته‌اند. نکته قابل توجه دیگر رعایت شدن زمان‌های جابجایی است که بعنوان نمونه می‌توان به فاصله بین اتمام کار  $J_1$  روی ماشین  $M_2$  و شروع این کار روی ماشین بعدی اشاره کرد  $(Tr_{211} = 2)$ .



شکل ۲. نمودار گانت مربوط به حل بهینه مثال

منظور، تمامی اندیس‌های مربوط به متغیرها و محدودیت‌ها شمرده و به ترتیب در جدول‌های (۱) و (۲) نمایش داده شده است. تأثیر افزایش تعداد کارها، ماشین‌ها و بسته‌ها روی تعداد متغیرها و محدودیت‌ها نیز در جدول (۳) قابل مشاهده است.

$\theta_1$  و  $\theta_2$  در تابع هدف ضرایب مثبت و یا عبارتی وزن هر یک از اهداف بوده و بصورت  $\theta_1 + \theta_2 = 1$  در نظر گرفته می‌شوند. زمان تکمیل هر یک از عملیات‌ها  $(C_{ij})$  از طریق رابطه (۲) بدست می‌آید. اگر کار  $J$  روی ماشین  $i$  قبل از ماشین  $h$  پردازش شود  $(Y_{ihj} = 1)$ ، آنگاه زمان شروع آماده‌سازی کار  $J$  روی ماشین  $h$   $(st_{hj})$  بزرگتر از زمان تکمیل آن کار روی ماشین  $i$  خواهد بود که رابطه (۳) این مطلب را نشان می‌دهد. اگر کار  $J$  قبل از کار  $k$  روی ماشین  $i$  پردازش شود  $(X_{ijk} = 1)$  و هر دو کار در یک بسته  $l$  واقع شده باشند  $(A_{ijl} = A_{ikl} = 1)$ ، آنگاه زمان شروع آماده‌سازی کار  $k$  روی ماشین  $i$   $(st_{ik})$  بزرگتر از زمان تکمیل کار  $J$  روی آن ماشین خواهد بود که رابطه (۴) بیانگر این مطلب است. رابطه (۵) تضمین می‌کند که هیچ آماده‌سازی در زمان‌های عدم دسترسی ماشین‌ها واقع نشود. رابطه‌های (۶) و (۷) بطور توأم از قرار گرفتن زمان تکمیل عملیات‌ها در زمان‌های عدم دسترسی ماشین‌ها جلوگیری می‌کنند. رابطه‌های (۸) و (۹) بطور همزمان توالی کارها روی هر ماشین و بسته مربوطه را مشخص می‌کنند. رابطه (۱۰) ترتیب ماشین‌ها را برای عملیات‌های هر کار تعیین می‌کند. رابطه (۱۱) تضمین می‌کند که هر عملیات دقیقاً در یک بسته از هر ماشین واقع شود. رابطه (۱۲) مجموع کل زمان‌های پردازش مربوط به عملیات‌های قرار گرفته در یک بسته را با حداکثر زمان آن بسته که در حقیقت همان زمان دسترسی ماشین مربوطه  $(T_i)$  است کنترل می‌کند. زمان تکمیل هر کار از طریق رابطه (۱۳) و با توجه به تابع هدف محاسبه می‌شود. رابطه‌های (۱۴) و (۱۵) به ترتیب طولانی‌ترین زمان تکمیل و زمان تأخیر هر عملیات را محاسبه می‌کنند. سرانجام، رابطه‌های (۱۶) و (۱۷) نشان‌دهنده متغیرهای نامنفی و متغیرهای صفرویک هستند.

## ۵-۲. یک مثال

در اینجا، جهت روشن‌تر شدن مفهوم مسأله، یک مثال ارائه و آن را با مدل MOMILP فوق حل می‌کنیم. این مثال شامل ۵ کار

## ۶-۲. تحلیل مدل

در این بخش، حساسیت مدل MOMILP نسبت به تعداد کارها  $(m)$ ، ماشین‌ها  $(m)$  و بسته‌ها  $(b)$  مورد بررسی قرار می‌گیرد. بدین-

حل مسأله‌های بهینه‌سازی [۱۹] و نیز الگوریتم شبیه‌سازی تبرید بعنوان یک الگوریتم سریع استفاده می‌کنیم.

### ۳-۱. الگوریتم ژنتیک

الگوریتم ژنتیک یکی از الگوریتم‌های تکاملی<sup>۷</sup> بوده که در دسته تکنیک‌های جستجوی تصادفی هدایت شده<sup>۸</sup> قرار می‌گیرد [۲۰]. الگوریتم‌های تکاملی درحقیقت تکامل موجود در طبیعت را بگونه-ای غیرجنسی برای جستجوی فضای حل شبیه‌سازی می‌کنند [۲۱]. هولاند و همکاران [۲۲] از جهش تصادفی ژن‌ها که در تولیدمثل اتفاق می‌افتد الهام گرفته و آن را در جهت جستجوی فضای حل توسعه داد. در دستاورد هولاند، هر فرزند از ترکیب دو والد ایجاد شده و عملگر تقاطع بعنوان یکی از اجزای پایه‌ای این تکنیک جستجو در نظر گرفته می‌شود. پارامترهای الگوریتم ژنتیک عبارتند از: احتمال تقاطع ( $p_c$ )، احتمال جهش ( $p_m$ )، تعداد افراد در هر نسل ( $popsize$ ) و یک معیار توقف. در ادامه، برخی عبارات پایه‌ای الگوریتم ژنتیک توضیح داده شده و سپس ساختار الگوریتم موردنظر تشریح می‌شود.

### ۳-۱-۱. نمایش کروموزوم

کروموزوم‌ها<sup>۹</sup> در حقیقت همان جواب‌های کد شده و نقاط فضای حل هستند. در این مقاله، هر عملیات بعنوان یک ژن در نظر گرفته شده و هر کروموزوم (یعنی؛ جواب) بصورت یک جایگشت از ژن‌ها (یعنی؛ عملیات‌ها) نشان داده می‌شود. برای مثال، مسأله‌ای با ۵ کار و ۲ ماشین و در نتیجه ۱۰ عملیات را در نظر بگیرید. یک کروموزوم (یا یک جواب کد شده) می‌تواند بصورت زیر باشد:

$O_{11}$	$O_{12}$	$O_{13}$	$O_{14}$	$O_{15}$	$O_{21}$	$O_{22}$	$O_{23}$	$O_{24}$	$O_{25}$
۳	۶	۸	۲	۵	۱	۹	۱۰	۴	۷

که در آن  $O_{ij}$  عملیات مربوط به کار  $j$  روی ماشین  $i$  است. مطابق با این کروموزوم، عملیات  $O_{21}$  قبل از عملیات  $O_{11}$  (توجه شود که این دو عملیات مربوط به یک کار هستند) و نیز عملیات  $O_{14}$  قبل از عملیات  $O_{11}$  (توجه شود که این دو عملیات مربوط به یک ماشین هستند) پردازش خواهند شد.

### ۳-۲. جمعیت اولیه

وجود پراکندگی در نسل‌ها از همگرایی سریع و هدایت شدن به بهینه محلی جلوگیری می‌کند. بنابراین، ما جمعیت اولیه<sup>۱۰</sup> (نسل آغازین) را بصورت تصادفی تولید می‌کنیم تا پراکندگی جواب‌ها در فضای حل در حد امکان حفظ شود. بدین معنی که یک جایگشت تصادفی از مجموعه  $\{1, 2, \dots, m.n\}$  بعنوان یک فرد در نظر گرفته می‌شود که در آن  $m.n$  بیانگر تعداد عملیات‌هاست.

### ۳-۱-۳. تابع هدف

جهت محاسبه تابع هدف ۱۱ مربوط به هر کروموزوم (یعنی؛ نقاط فضای حل) می‌بایست این جواب‌های کد شده را اصطلاحاً

### جدول ۱. تعداد متغیرها

متغیر	تعداد
$Z, C_{max}$	۱
$C_j, Tard_j$	$n$
$st_{ij}, C_{ij}$	$mn$
$X_{ijk}$	$mn(n-1)/2$
$Y_{ihj}$	$mn(m-1)/2$
$A_{ijt}$	$mnb$

### جدول ۲. تعداد محدودیت‌ها

تعداد	محدودیت
۱	(۱)
$mn$	(۲)، (۱۱) و (۱۳)
$mn(m-1)$	(۳)
$mnb(n-1)$	(۴)
$mnb$	(۵)، (۶) و (۷)
$mn(n-1)b(b-1)/2$	(۸)
$mn(n-1)/2$	(۹)
$mn(m-1)/2$	(۱۰)
$mb$	(۱۲)
$n$	(۱۴) و (۱۵)

### جدول ۳. تعداد متغیرها و محدودیت‌ها مطابق با مدل

MOMILP		
تعداد محدودیت‌ها	تعداد متغیرها	اندازه مسأله $m.n.b$
۳۹۳	۶۱	۲,۴,۴
۶۰۷	۹۵	۳,۴,۴
۸۲۶	۹۱	۲,۵,۵
۱۲۶۱	۱۴۱	۳,۵,۵
۲۰۳۹۹۲۶	۷۱۵۶	۵,۳۰,۳۰
۴۰۸۲۱۰۱	۱۵۰۳۱	۱۰,۳۰,۳۰
۶۴۲۵۹۰۱	۱۲۵۴۱	۵,۴۰,۴۰
۱۲۸۵۴۸۰۱	۲۶۰۴۱	۱۰,۴۰,۴۰

### ۳. الگوریتم‌های فرا ابتکاری

همانگونه که در مقدمه عنوان شد، مسأله زمان‌بندی کارگاه باز بعنوان یک مسأله NP-hard شناخته شده است و تنها مسأله‌های با ابعاد کوچک را می‌توان در زمان منطقی بطور دقیق حل نمود. بنابراین، گرچه مدل ریاضی ارائه شده به جواب دقیق دست می‌یابد ولی با بزرگ شدن ابعاد مسأله از کارایی آن کم می‌شود. از اینرو، اغلب روش‌های ابتکاری و یا الگوریتم‌های تقریبی برای حل مسأله-هایی با ابعاد متوسط و بزرگ که عموماً در دنیای واقعی رخ می‌دهند مؤثرترند. بنابراین، ما از الگوریتم ژنتیک بعنوان یک الگوریتم تقریبی خیلی قوی و از قوی‌ترین روش‌های فرا ابتکاری مرسوم در

پس از انتخاب یک جفت والد با یکی از روش‌های انتخاب، عملگر ژنتیک تقاطع با احتمال  $p_c$  جهت ترکیب دو والد و تولید دو فرزند بکار گرفته می‌شود. روش‌های مختلفی برای تقاطع و انجام عمل ترکیب جفت والد وجود دارد که برخی از آن‌ها عبارتند از: تقاطع تک نقطه‌ای<sup>۱۵</sup>، دونقطه‌ای<sup>۱۶</sup>، چندنقطه‌ای<sup>۱۷</sup>، تقاطع یکنواخت<sup>۱۸</sup> و غیره [۲۴]. بهرحال، مشکل اصلی عملگر تقاطع ممکن است عدم تضمین شدنی بودن جواب‌های (یعنی؛ نوزادها) جدید تولید شده باشد. در مواردی که کروموزوم‌های جدید یا همان نوزادهای تولید شده جواب‌های نشدنی را نتیجه دهند، معمولاً یک اقدام اصلاحی جهت تبدیل آن‌ها به جواب‌های شدنی صورت می‌گیرد که طولانی شدن زمان حل را در پی خواهد داشت. ما در این مقاله از یک عملگر تقاطع دونقطه‌ای با حفظ شدنی بودن کروموزوم‌های جدید هر نسل و عدم نیاز به فرآیند اصلاحی استفاده می‌کنیم [۲۵]. در ادامه عملگر تقاطع مورد نظر طی چهار مرحله تشریح می‌شود:

قدم ۱. دو عملیات را بطور تصادفی از کروموزوم (والد) ۱ انتخاب کن.

قدم ۲. ژن‌های مربوط به دو عملیات انتخاب شده را به‌همراه تمامی ژن‌های بین آن دو با حفظ محل قرارگیری از والد ۱ به نوزاد ۱ منتقل کن.

قدم ۳. مابقی ژن‌های مورد نیاز نوزاد ۱ تا تبدیل آن به یک کروموزوم کامل را به ترتیب از چپ به راست از والد ۲ انتخاب و به همان ترتیب از چپ به راست در محل‌های خالی نوزاد ۱ قرار بده.

قدم ۴. جهت تولید نوزاد دوم، جای دو والد ۱ و ۲ را تعویض و قدم‌های ۲ و ۳ را با ژن‌های همان عملیات‌هایی که در قدم ۱ انتخاب شده بودند تکرار کن.

برای مثال، دو والد زیر را در نظر بگیرید:

والد ۱	۳	۶	۸	۲	۵	۱	۹	۱۰	۴	۷
والد ۲	۴	۹	۵	۶	۲	۱	۷	۱۰	۸	۳

قدم ۱. فرض کنید دو عملیات انتخاب شده والد ۱ عملیات‌های  $O_{۱۴}$  و  $O_{۲۳}$  یعنی به ترتیب ژن‌های ۲ و ۹ باشند.

قدم ۲. ژن‌های ۲، ۵، ۱ و ۹ را با حفظ محل قرارگیری از والد ۱ به نوزاد ۱ منتقل می‌کنیم.

والد ۱	۳	۶	۸	۲	۵	۱	۹	۱۰	۴	۷
والد ۲	۴	۹	۵	۶	۲	۱	۷	۱۰	۸	۳

نوزاد ۱				۲	۵	۱	۹			
---------	--	--	--	---	---	---	---	--	--	--

رمزگشایی<sup>۱۱</sup> نمود. بدین‌منظور، عملیات‌ها را به ترتیب از اولین عملیات (عملیات  $O_{۲۱}$  در مثال فوق) تا آخرین عملیات (عملیات  $O_{۳۳}$  در مثال فوق) با در نظر گرفتن زمان‌های عدم دسترسی، روی ماشین‌ها قرار داده و توأمأً زمان تکمیل آن‌ها ( $C_{ij}$ ) را نیز محاسبه می‌کنیم. سپس، با در دست داشتن زمان تکمیل همه عملیات‌ها، طولانی‌ترین زمان تکمیل بصورت  $C_{max} = \max_{i,j} \{C_{ij}\}$  محاسبه می‌شود. همچنین، جهت محاسبه عبارت دوم تابع هدف یعنی مجموع تأخیرها می‌بایست ابتدا زمان تکمیل هر کار بصورت  $C_j = \max_i \{C_{ij}\}$  محاسبه و سپس زمان تأخیر آن کار را از رابطه  $Tard_j = \max\{0, C_j - d_j\}$  بدست آورد. در نهایت مجموع تأخیرها بصورت  $\sum_{j=1}^n Tard_j$  محاسبه می‌شود. در ارتباط با نحوه تعیین مقادیر مربوط به ضرایب  $\theta_1$  و  $\theta_2$  بخش-های بعدی توضیح داده خواهد شد. بنابراین، تابع هدف مربوط به هر کروموزوم بصورت  $Z = \theta_1 C_{max} + \theta_2 \sum_{j=1}^n Tard_j$  محاسبه می‌شود.

### ۳-۱-۴. تابع برازندگی

اگرچه تابع هدف بصورت کمینه‌سازی است، ولی بمنظور ساده-سازی ما آن را به فرم بیشینه‌سازی تبدیل می‌کنیم. این کار از طریق کم کردن مقدار تابع هدف از عدد مثبت بزرگ  $M$  انجام می‌شود و به مقدار بدست آمده مقدار برازندگی می‌گوییم. بنابراین، تابع برازندگی<sup>۱۲</sup> فرد  $i$  بصورت  $f_i = M - Z_i$  بدست می‌آید. که در آن  $Z_i$  بیانگر مقدار تابع هدف فرد  $i$  است.

### ۳-۱-۵. انتخاب

انتخاب<sup>۱۳</sup>، فرآیندی است که براساس آن افراد حاضر در یک نسل دو به دو برای جفت‌گیری انتخاب می‌شوند. رویکردهای متنوعی برای انتخاب در ادبیات وجود دارد [۲۳]. ما در این مقاله از برنامه چرخ رولت<sup>۱۴</sup> استفاده می‌کنیم که طبق آن افراد با برازندگی بالاتر شانس بیشتری برای جفت‌گیری دارند و این در حقیقت اساس نظریه داروین است. جهت بکارگیری رویکرد چرخ رولت، دو پارامتر شامل: احتمال انتخاب فرد  $i$  ( $PS_i$ ) و احتمال تجمعی فرد  $i$  ( $CP_i$ ) مورد نیاز است که به ترتیب از رابطه‌های (۱۸) و (۱۹) بدست می‌آیند:

$$PS_i = \frac{f_i}{\sum_{j=1}^{popsize} f_j} \quad (18)$$

$$CP_i = \sum_{j=1}^i PS_j \quad (19)$$

براساس رویکرد چرخ رولت، یک عدد  $r$  بطور تصادفی در بازه  $[0, 1]$  تولید و سپس فرد  $i$  -م که حائز شرط  $CP_{i-1} < r \leq CP_i$  شده‌است بعنوان یکی از جفت‌ها انتخاب می‌گردد و این فرآیند جهت انتخاب فرد دوم تکرار می‌شود.

### ۳-۱-۶. تقاطع



قدم ۵. با احتمال  $p_m$  عملگر جهش را روی هر یک از نوزادها اعمال کن.

قدم ۶. برازندگی نوزادها را محاسبه کن.

قدم ۷. قدم‌های ۳ تا ۶ را تا تولید تعداد  $popsiz$  نوزاد جدید ادامه بده.

قدم ۸. تمامی افراد موجود اعم از قدیمی (یعنی؛ همان جمعیت اولیه) و جدید (یعنی؛ نوزادان تولید شده) که هم‌اکنون مجموع جمعیت آن‌ها به تعداد  $2 \times popsiz$  رسیده است را براساس میزان برازندگی مرتب کن.

قدم ۹. تعداد  $popsiz$  فرد از بهترین‌ها (برازنده‌ترین‌ها) را به نسل جدید منتقل کن.

قدم ۱۰. شرط توقف را کنترل، در صورت عدم برقراری آن به قدم ۳ و در غیر این صورت به قدم ۱۱ برو.

قدم ۱۱. بهترین جواب (برازنده‌ترین کروموزوم) مربوط به آخرین نسل را بعنوان جواب الگوریتم انتخاب کن.

قدم ۱۲. پایان.

مقادیر مربوط به پارامترهای الگوریتم ژنتیک شامل:  $p_m$ ،  $p_c$ ،  $popsiz$  و  $\epsilon$  از طریق طراحی آزمایشات آماری در بخش‌های آتی تعیین خواهد شد.

### ۳-۱-۱۰. الگوریتم ژنتیک موازی چند هدفه

تحقیقات زیادی در ادبیات وجود دارد که در آن‌ها از الگوریتم ژنتیک برای حل مسأله‌های چند هدفه استفاده شده است که علت آن، توانایی بالای این الگوریتم در برخورد با بهینه‌سازی چند هدفه است. از آنجایی که الگوریتم‌های ژنتیک با جمعیتی از نقاط (جواب‌ها) سر و کار دارد، جواب‌های چندگانه بهینه پارتو را می‌توان در یک جمعیت از الگوریتم ژنتیک یافت [۲۷]. در ادامه، نحوه برخورد با یک مسأله چند هدفه بیشتر تشریح می‌شود.

در مدل ریاضی MOMILP،  $\theta_1$  و  $\theta_2$  را به ترتیب وزن‌های هر یک از هدف‌های  $C_{max}$  و  $Tard_j$  معرفی کرده و تابع هدف را بصورت یک تابع منفرد بشکل عبارت (۱) نوشتیم. حال، مسأله دو هدفه ما به یک مسأله تک هدفه تبدیل شده است. اما، تعیین وزن‌ها کار ظریف و حساسی است. برای مرتفع نمودن این مشکل، می‌توان جمعیت را به چندین گروه تقسیم کرده که افراد هر گروه در جستجوی یکی از ترکیبات ممکن ضرایب هدف‌ها بوده و درعین حال همه گروه‌ها بطور موازی جواب‌های بهینه پارتو را جستجو کنند. با این روش می‌توانیم مسأله چند هدفه را به یک مسأله تک هدفه با وزن‌های متنوع تبدیل کنیم. بنابراین، ما مجموعه‌ای از  $\lambda$  وزن مختلف را طوری در نظر می‌گیریم که اختلاف اندکی بین هر دو وزن متوالی وجود داشته باشد. رابطه (۲۰) مجموعه ترکیبات ممکن را نشان می‌دهد.

قدم ۳. مابقی ژن‌های مورد نیاز نوزاد ۱ را به ترتیب از چپ به راست از والد ۲ انتخاب کرده و به همان ترتیب چپ به راست در محل‌های خالی قرار می‌دهیم.

نوزاد ۱	۴	۶	۷	۲	۵	۱	۹	۱۰	۸	۳
---------	---	---	---	---	---	---	---	----	---	---

قدم ۴. جای دو والد ۱ و ۲ را عوض کرده و قدم‌های ۲ و ۳ را تکرار می‌کنیم تا نوزاد ۲ بشکل زیر تولید شود:

نوزاد ۲	۳	۸	۵	۶	۲	۱	۷	۹	۱۰	۴
---------	---	---	---	---	---	---	---	---	----	---

### ۳-۱-۷. جهش

بمنظور جلوگیری از هدایت نسل‌های تولید شده به سمت جواب بهینه محلی، عملگری بنام جهش با احتمال  $p_m$  روی هر یک از نوزادان تولید شده اعمال می‌شود. نحوه کار این عملگر بدین ترتیب است که دو عملیات بطور تصادفی انتخاب شده و ژن‌های آن‌ها با هم تعویض می‌شود [۲۵]. بعنوان مثال، فرض کنید در نوزاد ۱ که در بالا تولید شد، عملیات‌های  $O_{13}$  و  $O_{24}$  انتخاب شده باشند. بنابراین، ژن‌های ۶ و ۸ جابجا شده و نوزاد جهش یافته بصورت زیر خواهد بود:

نوزاد ۱	۴	۶	۷	۲	۵	۱	۹	۱۰	۸	۳
جهش یافته	۴	۸	۷	۲	۵	۱	۹	۱۰	۶	۳
نوزاد ۱										

### ۳-۱-۸. معیار توقف

الگوریتم ژنتیک تا برقراری شرط توقف که عبارت از گذشت مدت زمان  $m.n.\epsilon$  ثانیه از شروع آن است ادامه یافته و نسل‌های جدیدی تولید می‌کند [۲۶]. که در آن،  $\epsilon$  ضریبی است که مقدار آن از طریق آزمایشات تنظیم می‌شود. در بخش‌های بعدی در مورد نحوه تنظیم آن بحث خواهد شد. همچنین، علت انتخاب این معیار توقف آنست که برای حل مسأله‌های بزرگتر زمان بیشتری به الگوریتم می‌دهد.

### ۳-۱-۹. الگوریتم ژنتیک اولیه

قبل از آنکه الگوریتم ژنتیک مورد نظر برای حل مسأله‌های چند هدفه را ارائه دهیم، در اینجا ساختار و نحوه عملکرد الگوریتم ژنتیک اولیه را با استفاده از توضیحات فوق تشریح می‌کنیم. در زیر نحوه عملکرد الگوریتم ژنتیک اولیه در ۱۲ قدم آمده است:

قدم ۱. تعداد  $popsiz$  فرد (کروموزوم) را بعنوان جمعیت اولیه بطور تصادفی تولید کن.

قدم ۲. برازندگی هر فرد را محاسبه کن.

قدم ۳. یک جفت والد را از افراد بر اساس استراتژی انتخاب برگزین.

قدم ۴. با احتمال  $p_c$  عملگر تقاطع را روی والد‌ها بکار گرفته و دو نوزاد جدید تولید کن.

$$\theta = \left\{ (0.0, 1.0), (0.05, 0.95), \dots, (1.0, 0.0) \right\} \quad (22)$$

قدم ۴. فرآیند الگوریتم ژنتیک اولیه را از قدم ۲ به بعد برای همه زیر-جمعیت‌ها انجام بده و جواب‌های غالب را ذخیره کن.  
قدم ۵. بهترین جواب را از بین جواب‌های غالب پیدا کن.  
قدم ۶. پایان.

### ۳-۲. الگوریتم شبیه‌سازی تبرید

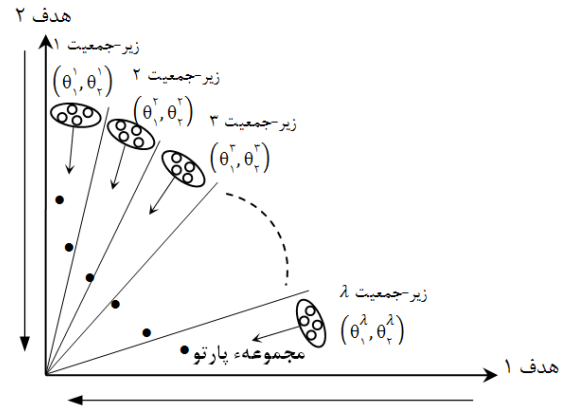
یکی دیگر از الگوریتم‌های فرا ابتکاری رایج برای حل مسأله‌های NP-hard زمان‌بندی شبیه‌سازی تبرید (SA) است. این الگوریتم، الهام گرفته از سیستم‌های فیزیکی است که در آن‌ها انرژی اتم‌ها از طریق فرآیند سردسازی تا رسیدن به یک سطح حداقلی کاهش می‌یابد [۲۹، ۲۸، ۲۶]. علت استفاده ما از این الگوریتم سرعت آن در رسیدن به جواب است که از سادگی ساختار محاسباتی آن ناشی می‌شود [۲۶]. ما در این بخش ابتدا یک نمای کلی از الگوریتم شبیه‌سازی تبرید ارائه داده و در ادامه الگوریتم شبیه‌سازی تبرید موازی چند هدفه را تشریح می‌کنیم.

### ۳-۲-۱. الگوریتم شبیه‌سازی تبرید اولیه

شبیه‌سازی تبرید از یک جواب (یعنی؛ سطح انرژی) بعنوان پایه در یک دمای اولیه ( $temp_0$ ) شروع کرده و فضای حل را در همسایگی آن جواب جستجو می‌کند. جستجو تا برقراری یک شرط توقف که ما آن را بصورت تعداد  $m.n.\phi$  تکرار در نظر می‌گیریم ادامه می‌یابد. پارامتر  $\phi$ ، ضریب ثابتی است که از بازه (۱ ۰) انتخاب می‌شود. ما در این الگوریتم نیز از روشی که در الگوریتم ژنتیک جهت کد کردن جواب‌های مسأله و معرفی کروموزوم‌ها استفاده شد بهره می‌بریم. همچنین، یک جایگشت تصادفی از عملیات‌ها را بعنوان جواب پایه برای شروع در نظر گرفته می‌شود. بمنظور حرکت از یک جواب به جوابی دیگر، روش جستجوی همسایگی که در شکل (۴) تشریح شده‌است بکار گرفته می‌شود. پس از اتمام فرآیند جستجوی همسایگی، دمای فعلی از طریق ضرب آن در پارامتر  $\mu \in (0, 1)$  کاهش یافته و مجدداً فرآیند جستجوی همسایگی تکرار می‌شود. الگوریتم تا رسیدن به دمای نهایی ( $temp_f$ ) ادامه می‌یابد. در طی تکرارها و جستجوهای الگوریتم چنانچه جواب جدید مقدار تابع هدف را بهبود دهد پذیرفته می‌شود. همچنین، در صورت عدم بهبود مقدار تابع هدف، جواب جدید با احتمال  $exp(-\Delta/)$  ( $temp_i$ ) پذیرفته می‌شود. که در آن،  $temp_i$  بیانگر دمای فعلی بوده و  $\Delta = 100 \times (Z_{new} - Z_{old})/Z_{new}$  شبیه‌سازی تبرید موردنظر بطور خلاصه در شکل (۵) آمده‌است. ما دمای نهایی را ۱ در نظر گرفته و در مورد مقدار پارامترهای دیگر الگوریتم شامل:  $temp_0$  و  $\mu$  و  $\phi$  که با روش طراحی آزمایشات آماری تنظیم می‌شوند در بخش ۴ بحث می‌کنیم.

$$\theta = \left\{ (\theta_1^1, \theta_1^1), (\theta_1^2, \theta_1^2), \dots, (\theta_1^\lambda, \theta_1^\lambda) \right\} \quad (20)$$

بطریق مشابه، جمعیت را به  $\lambda$  زیر-جمعیت تقسیم کرده و هر یک از وزن‌های مجموعه فوق را به یکی از آن‌ها اختصاص می‌دهیم. بدین ترتیب، هر زیر-جمعیت در مسیر جداگانه‌ای جواب‌های بهینه پارتو را جستجو می‌کند. زیر-جمعیت‌ها بطور مستقل و نامرتبط از یکدیگر بعنوان الگوریتم‌های ژنتیک موازی عمل می‌کنند ولی پارامترها و عملگرهای دیگر آن‌ها شامل عملگرهای تقاطع و جهش و نیز اندازه زیر-جمعیت و شرط توقف همگی یکسان هستند. همچنین، مطابق با آنچه در شکل (۳) پیداست، هر زیر-جمعیت ناحیه مربوط به خود را جستجو می‌کند.



شکل ۳. نحوه تقسیم‌بندی جمعیت و عملکرد موازی زیر-جمعیت‌ها

با این روش، پراکندگی جواب‌ها بطور مؤثری تقویت می‌شود. همچنین، مجموعه جواب‌های پارتو جهت نگهداری جواب‌های غالب ذخیره می‌شوند. بدین ترتیب، از مفقود شدن جواب‌های غالب در طول فرآیند بهینه‌سازی جلوگیری می‌شود. این مجموعه به منظور نزدیک شدن به جواب بهینه پارتو بطور مکرر به روز آوری می‌شود. فرآیند الگوریتم ژنتیک موازی چند هدفه ( $MOPGA^{19}$ ) موردنظر بشرح زیر است:

قدم ۱. جمعیت اولیه را بطور تصادفی و به تعداد  $\lambda \times popsize$  تولید کن.

قدم ۲. جمعیت اولیه را به  $\lambda$  زیر-جمعیت تقسیم کن.

قدم ۳. وزن‌های مربوط به تابع هدف منفرد را بشکل زیر تولید و به زیر-جمعیت‌ها اختصاص بده.

$$\theta = \left\{ (sub \times \rho, 1 - sub \times \rho) : sub = 0, 1, \dots, \lambda - 1 \right\} \quad (21)$$

ما در این مقاله مقدار پارامترهای  $\lambda$  و  $\rho$  را بصورت  $\lambda = 21$  و  $\rho = 0.05$  در نظر می‌گیریم. بنابراین، داریم:

جمعیت اولیه به چند زیر-جمعیت تقسیم می‌شد ولی از آنجایی که الگوریتم SA تنها با یک تک کروموزوم کار را شروع می‌کند نمی‌توان چنین تقسیم‌بندی انجام داد. بنابراین، تعداد  $\lambda$  کروموزوم تولید شده و هر یک ناحیه مربوط به خود را مطابق آنچه پیشتر در شکل (۳) مشاهده کردید جستجو می‌کند. فرآیند الگوریتم شبیه-سازی تبرید موازی چند هدفه موردنظر بشرح زیر است:

قدم ۱. تعداد  $\lambda$  کروموزوم را بطور تصادفی تولید کن.

قدم ۲. وزن‌های مربوط به تابع هدف منفرد را بشکل عبارت (۲۲) تولید و به کروموزوم‌ها اختصاص بده.

قدم ۳. فرآیند الگوریتم شبیه‌سازی تبرید اولیه را مطابق شکل (۵) و از قدم ۲ به بعد برای همه کروموزوم‌ها انجام بده و جواب‌های غالب را ذخیره کن.

قدم ۴. بهترین جواب را از بین جواب‌های غالب پیدا کن.

قدم ۵. پایان.

## جستجوی همسایگی

یک  $\lambda$  را بطور تصادفی انتخاب کن.  
اگر  $\lambda$  انتخاب شده در اولین محل از کروموزوم واقع گردیده است آنگاه آن را با  $\lambda$  کناری (بعدی) جابجا کن.  
در غیر این صورت اگر  $\lambda$  انتخاب شده در آخرین محل از کروموزوم واقع گردیده است آنگاه آن را با  $\lambda$  کناری (قبلی) جابجا کن.  
در غیر این صورت جابجایی با دو  $\lambda$  کناری (قبلی و بعدی) را ارزیابی کن.  
 $\lambda$  انتخاب شده را با آن  $\lambda$  کناری که بهبود بیشتری حاصل می‌کند جابجا کن.  
پایان شرط

## شکل ۴. جستجوی همسایگی الگوریتم شبیه‌سازی تبرید

## شبیه‌سازی تبرید اولیه

۱. یک جایگشت تصادفی از عملیات‌ها تولید کن.
۲. مقدار تابع هدف را محاسبه کن.
۳. پارامترهای  $\mu$ ،  $\varphi$  و  $temp_i$ ،  $temp$  را تعیین کن.
۴. معیار توقف را کنترل، در صورت عدم برقراری به گام ۵ و در غیر این صورت به گام ۱۵ برو.
۵. قرار بده  $repeat = 0$ .
۶. اگر  $repeat < Int(m.n.\varphi)$  آنگاه به گام ۷ و در غیر این صورت به گام ۱۴ برو.
۷. فرآیند جستجوی همسایگی را فراخوانی کن.
۸. اگر در مقدار تابع هدف بهبود حاصل شد آنگاه به گام ۹ و در غیر این صورت به گام ۱۰ برو.
۹. جواب جدید را بپذیر، مقدار تابع هدف را بروز کن و به گام ۱۳ برو.
۱۰. یک عدد تصادفی از بازه (۰، ۱) انتخاب و در  $rand$  قرار بده.
۱۱. اگر  $exp(-\Delta/temp_i) < rand$  آنگاه به قدم ۱۲ و در غیر این صورت به گام ۱۳ برو.
۱۲. جواب جدید را بپذیر، مقدار تابع هدف را بروز کن.
۱۳. قرار بده  $repeat = repeat + 1$  و به گام ۶ برو.
۱۴. دمای جدید را بصورت  $temp_{new} = \mu \cdot temp_{old}$  اصلاح کن و به گام ۴ برو.
۱۵. پایان.

## شکل ۵. قدم‌های الگوریتم شبیه‌سازی تبرید اولیه

## ۲-۲-۲ الگوریتم شبیه‌سازی تبرید موازی چند هدفه

فرآیند الگوریتم شبیه‌سازی تبرید موازی چند هدفه (MOPSA)<sup>20</sup> بسیار شبیه فرآیند الگوریتم ژنتیک موازی چند هدفه بوده و هر دو از یک رویکرد در برخورد با بهینه‌سازی چند هدفه بهره می‌برند. در اینجا نیز تابع هدف بصورت تابع منفرد عبارت (۱) در نظر گرفته شده و چند الگوریتم SA بطور موازی و نامرتب ترکیب‌های مختلف وزن‌های هر یک از اهداف را بررسی می‌کنند. تفاوت شروع الگوریتم MOPSA با MOPGA در آنست که در MOPGA

## ۴. طراحی آزمایشات و ارزیابی محاسباتی

در این بخش، ما بدنبال پایا نمودن الگوریتم‌های پیشنهادی از طریق انجام آزمایشات و تحلیل‌های آماری هستیم. بدین منظور، می‌بایست سطوح مختلفی برای فاکتورهای مربوطه در نظر گرفته شده و ترکیبات مختلف آن‌ها مورد بررسی قرار گیرند. بنابراین، رویکرد تاگوچی بعنوان روش طراحی آزمایشات در این مقاله تشریح شده و مورد استفاده قرار می‌گیرد.

بعلاوه، پس از تنظیم پارامترهای مربوط به الگوریتم‌های پیشنهادی با استفاده از روش تاگوچی و از طریق تولید داده‌های تصادفی، دو گروه مسأله یکی با ابعاد کوچک و دیگری با ابعاد بزرگ نیز بطور تصادفی تولید و نتایج روش‌های حل مقایسه می‌شوند.

## ۴-۱-۴ طراحی آزمایشات تاگوچی

روش تاگوچی یکی از روش‌های تحلیل آماری است که در آن با استفاده از آرایه‌های متعامد تعداد آزمایشات نسبت به طرح‌های فاکتوریلی کامل بطور چشمگیری کاهش می‌یابد [۲۶]. برای مثال، جهت طراحی آزمایشی با ۴ فاکتور که هر فاکتور آن شامل ۳ سطح است، مطابق با طرح فاکتوریلی کامل تعداد ۸۱ (۳<sup>۴</sup>) آزمایش باید انجام شود در حالی که تاگوچی آن‌ها را به تنها ۹ آزمایش کاهش می‌دهد.

تاگوچی معتقد بود که کلید بهبود کیفیت کاهش انحرافات است نه فقط کنترل قرارگیری در بازه تعیین شده. بنابراین، او تلاش کرد انحرافات را به مقدار صفر نزدیک کند و توجه کمتری به قرارگیری در یک دامنه ثابت نمود. و این دلیل اصلی آنست که از روش تاگوچی بعنوان یک روش پایا یاد می‌شود [۳۰]. طبق روش تاگوچی، فاکتورها به دو دسته فاکتورهای قابل کنترل یا سیگنال و فاکتورهای غیرقابل کنترل یا نویز تقسیم‌بندی می‌شوند. بمنظور

که در آن  $a$  بطور تصادفی از مجموعه  $\left\{\frac{1}{5}, \frac{1}{4}, \frac{1}{3}\right\}$  انتخاب می‌شود.

• زمان تحویل کارها بصورت زیر محاسبه می‌شود:  
مجموع مدت زمانی که کار روی همه ماشین‌ها قرار می‌گیرد

$$d_j = \text{میانگین زمان‌های جابجایی کار بین همه ماشین‌ها} + \text{میانگین زمان‌های عدم دسترسی همه ماشین‌ها} \\ \text{ها.} (n-1) \cdot U(0,1) + U(0,1) \cdot (n-1) \\ \text{بنابراین، داریم:}$$

$$d_j = \sum_{i=1}^m p_{ij} + \frac{\left(\sum_{i=1}^m \sum_{h=1}^m Tr_{ihj}\right)}{m} + \frac{\sum_{i=1}^m t_i}{m} \cdot U(0,1) \cdot (n-1) \quad (26)$$

#### ۴-۱-۲. تنظیم پارامترهای الگوریتم MOPGA

فاکتورهایی که باید در الگوریتم MOPGA تنظیم شوند عبارتند از: احتمال تقاطع ( $p_c$ )، احتمال جهش ( $p_m$ )، اندازه جمعیت ( $popsiz$ ) و ضریب معیار توقف ( $\epsilon$ ). سه سطح برای هر فاکتور مطابق جدول (۴) [۲۶، ۲۵] در نظر گرفته می‌شود. بنابراین، طرح آرایه متعامد  $L_9$  مناسب‌ترین حالت است چراکه دقیقاً تعداد فاکتورها و سطوح مربوطه یکسانی با آزمایش موردنظر داشته و تمامی اهداف را برآورده می‌کند. ترکیبات سطوح مختلف فاکتورها در هر آزمایش از طرح  $L_9$  مربوط به الگوریتم MOPGA در جدول (۵) آمده‌است. الگوریتم MOPGA و طرح آزمایشات همگی با زبان ویژوال بیسیک نت ۲۰۰۸<sup>۳۳</sup> برنامه‌نویسی و در رایانه‌ای با پردازنده (CPU) Intel® Core™ i3 ۲،۲۶GHz و ۴GB حافظه موقت (RAM) اجرا شدند. متغیر پاسخ (تابع هدف) به کسر  $S/N$  تبدیل و نتایج در شکل (۶) نمایش داده شده‌اند. با توجه به نتایج، سطوح بهینه فاکتورها عبارتند از:  $p_c=0,8$ ،  $popsiz=30$  و  $\epsilon=0,4$ .

#### جدول ۴. فاکتورهای الگوریتم MOPGA و سطوح آن‌ها

فاکتور	سطح		
	(۱)	(۲)	(۳)
$p_c$	۰،۴	۰،۶	۰،۸
$p_m$	۰،۰	۰،۱	۰،۲
$popsiz$	۲۰	۳۰	۴۰
$\epsilon$	۰،۲	۰،۳	۰،۴

#### جدول ۵. آزمایشات مربوط به آرایه $L_9$ در الگوریتم

آزمایش	فاکتور		
	$popsiz$	$p_m$	$p_c$
۱	۲۰	۰،۰	۰،۴
۲	۴۰	۰،۱	۰،۴

کاهش انحرافات، کسری بنام سیگنال به نویز  $(S/N)^{21}$  تعریف می‌شود که حساسیت متغیر پاسخ را نسبت به فاکتورهای غیرقابل کنترل نشان داده و بیانگر انحراف حول یک مقدار سیگنال است. مقادیر بالای  $S/N$  معرف انحراف پایین بوده و این بدان معناست که فاکتورهای قابل کنترل تأثیر بیشتری از فاکتورهای غیرقابل کنترل دارند. با استفاده از سطوح بهینه فاکتورها که با این روش بدست می‌آیند، نزدیکی به مقدار میانی با کمترین انحراف مورد انتظار است [۲۶].

جهت تبدیل تابع هدف به کسر  $S/N$ ، سه نوع معیار در روش تاگوچی قابل تصور است که عبارتند از: "هرچه کمتر بهتر"، "هرچه بیشتر بهتر" و "هرچه نزدیکتر به مقدار اسمی بهتر" [۲۶]. با توجه به معیار اندازه‌گیری مسأله مورد بحث این مقاله که بصورت یک عبارت کمینه‌سازی بیان می‌شود، حالت "هرچه کمتر بهتر" مناسب است که فرمول مربوطه  $S/N$  آن بصورت زیر است:

$$S/N = -10 \log_{10} (Z^2) \quad (23)$$

همچنین، یک معیار عملکرد معروف بنام  $RPD^{22}$  [۳۰-۳۲] استفاده می‌شود که از رابطه زیر بدست می‌آید:

$$RPD = \frac{Alg_{sol} - Min_{sol}}{Min_{sol}} \cdot 100 \quad (24)$$

که در آن،  $Alg_{sol}$  مقدار تابع هدف مربوط به یک مثال است که از یک الگوریتم در آزمایش طراحی شده بدست می‌آید و  $Min_{sol}$  کمترین مقدار برای تابع هدف آن مثال است که از تمامی الگوریتم‌های آزمایشات بدست آمده‌است. در نتیجه، واضح است که مقادیر کم  $RPD$  ارزش بیشتری دارند.

#### ۴-۱-۱. تولید داده‌ها

جهت انجام آزمایشات تاگوچی مسأله‌ای با ابعاد  $(m, n, b) = (10, 5, 20)$  در نظر گرفته و ۵ تکرار بطور تصادفی از آن تولید می‌شود. همچنین، پارامترهای دیگر مسأله با الهام از ادبیات [۳۵-۳۳] بقرار زیرند:

- زمان‌های پردازش دارای توزیع یکنواخت پیوسته در بازه (۱، ۹۹) است.
- زمان‌های جابجایی بین ماشین‌ها دارای توزیع یکنواخت پیوسته در بازه (۱، ۲۰) است.
- طول زمان‌های متوالی عدم دسترسی هر ماشین دارای توزیع یکنواخت در بازه (۱، ۵۰) است.
- طول زمان‌های متوالی دسترسی ماشین‌ها (اندازه بسته‌های هر ماشین) بصورت زیر محاسبه می‌شود:

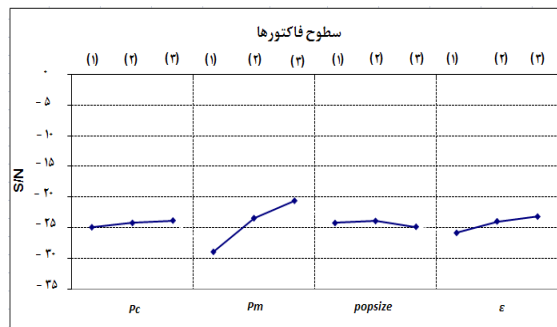
$$T_i = \max \left\{ a \sum_{j=1}^n p_{ij}, \max_i p_{ij} \right\} \quad (25)$$

شکل ۶. نمودار کسر S/N مربوط به RPD در فاکتورهای

### الگوریتم MOPGA

بمنظور یافتن میزان تأثیر هر یک از فاکتورها روی متغیر پاسخ، یک تحلیل واریانس (ANOVA) انجام می‌دهیم. توجه به این نکته ضروری است که در آزمایش طراحی شده، درجه آزادی خطا صفر است. برای برخورد با این مشکل، فاکتورهایی که کمترین میانگین مربعات را دارند، آمیخته و بعنوان خطا در نظر گرفته می‌شوند. در آزمایش فوق، فاکتورهای  $p_c$  و  $popsiz$  با میانگین مربعات به ترتیب ۰،۹۲ و ۰،۷۰ کمترین میانگین مربعات را داشته و در نتیجه کمترین تأثیر را روی متغیر پاسخ خواهد داشت. بنابراین، دو فاکتور مذکور بعنوان خطا در نظر گرفته می‌شوند. تحلیل واریانس مربوط به کسر S/N در جدول (۶) آمده‌است. بنابراین، مؤثرترین فاکتورها عبارتند از  $p_m$  و  $\varepsilon$  با درصد تأثیر به ترتیب ۸۶،۹۳٪ و ۷،۸۱٪.

۳	۰،۴	۰،۲	۳۰	۰،۴
۴	۰،۶	۰،۰	۴۰	۰،۴
۵	۰،۶	۰،۱	۳۰	۰،۲
۶	۰،۶	۰،۲	۲۰	۰،۳
۷	۰،۸	۰،۰	۳۰	۰،۳
۸	۰،۸	۰،۱	۲۰	۰،۴
۹	۰،۸	۰،۲	۴۰	۰،۲



جدول ۶. جدول تحلیل واریانس کسر S/N مربوط به فاکتورهای الگوریتم MOPGA

فاکتور	df	SS	MS	F	درصد تأثیر	درصد تجمعی	مقدار P
$popsiz$	۲	۱۰۸،۵۶	۵۴،۲۸	۶۷،۰۷	٪۸۶،۹۳	٪۸۶،۹۳	۰،۰۰
$\varepsilon$	۲	۱۱،۲۳	۵،۶۱	۶،۹۴	٪۷،۸۱	٪۹۴،۷۴	۰،۰۵
خطا ( $p_c + popsiz$ )	۴	۳،۲۴	۰،۸۱		٪۵،۲۶	٪۱۰۰،۰۰	
مجموع	۸	۱۲۳،۰۲					

جدول ۷. فاکتورهای الگوریتم MOPSA و سطوح آن‌ها

فاکتور	سطح (۱)	سطح (۲)
$temp.$	۵۰	۱۰۰
$\mu$	۰،۷	۰،۸
$\varphi$	۰،۳	۰،۶

جدول ۸. آزمایشات مربوط به آرایه  $L_4$  در الگوریتم

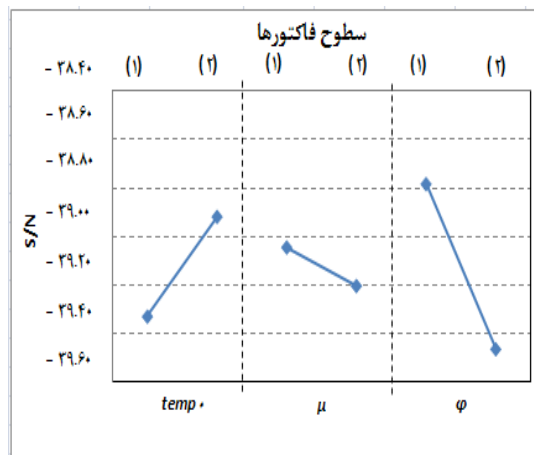
### MOPSA

آزمایش	فاکتور	$temp.$	$\mu$	$\varphi$
۱		۵۰	۰،۷	۰،۳
۲		۵۰	۰،۸	۰،۶
۳		۱۰۰	۰،۷	۰،۶
۴		۱۰۰	۰،۸	۰،۳

### ۴-۳. تنظیم پارامترهای الگوریتم MOPSA

مطابق با آنچه در تشریح الگوریتم شبیه‌سازی تبرید آمد، سه فاکتور باید تنظیم شوند که عبارتند از: دمای اولیه ( $temp.$ )، ضریب کاهش دما ( $\mu$ ) و ضریب تعداد تکرارها در هر دما ( $\varphi$ ). با در نظر گرفتن ۲ سطح برای هر فاکتور، آرایه متعامد  $L_4$  مناسب می‌نماید که فاکتورها و سطوح آن‌ها در جدول (۷) آمده‌اند. همچنین، ترکیبات سطوح مختلف فاکتورها در هر آزمایش از طرح  $L_4$  مربوط به الگوریتم MOPSA در جدول (۸) آمده‌است. الگوریتم MOPSA و طرح آزمایشات همگی با زبان ویژوال بیسیک نت ۲۰۰۸ برنامه‌نویسی و در رایانه‌ای با پردازنده (CPU) Intel® Core™ i3 ۲،۲۶GHz و ۴GB حافظه موقت (RAM) اجرا شدند. متغیر پاسخ به کسر S/N تبدیل و نتایج در شکل (۷) آمده‌اند. در نتیجه، سطوح بهینه بدست آمده عبارتند از:  $temp. = 100$ ،  $\mu = 0.7$  و  $\varphi = 0.3$ .

یک تحلیل واریانس جهت یافتن میزان تأثیر مربوط به هر یک از فاکتورها انجام گرفته است. همچنین، برای جلوگیری از صفر شدن درجه آزادی خطا، فاکتوری که کمترین میانگین مربعات را دارد را بعنوان خطا در نظر می گیریم. نتایج حاکی از آنست که فاکتور  $\mu$  با میانگین مربعات  $0.03$  کمترین تأثیر را روی متغیر پاسخ دارد. تحلیل واریانس کسر S/N مربوط به فاکتورهای الگوریتم MOPSA در جدول (۹) آمده است که در آن فاکتور  $\varphi$  با درصد تأثیر  $66.62\%$  بیشترین تأثیر را روی متغیر پاسخ دارد. همچنین،  $temp$  دومین فاکتور تأثیرگذار با درصد تأثیر  $21.89\%$  است.



شکل ۷. نمودار کسر S/N مربوط به RPD در فاکتورهای الگوریتم MOPSA

جدول ۹. جدول تحلیل واریانس کسر S/N مربوط به فاکتورهای الگوریتم MOPSA

فاکتور	df	SS	MS	F	درصد تأثیر	درصد تجمعی	مقدار p
temp	۱	۰,۱۷	۰,۱۷	۶,۷۲	21.89%	21.89%	۰,۲۳
$\varphi$	۱	۰,۴۷	۰,۴۷	۱۸,۴۰	66.62%	88.51%	۰,۱۵
خطا ( $\mu$ )	۱	۰,۰۳	۰,۰۳		11.49%	100.00%	
مجموع	۳	۰,۶۶					

کوچک است ولی با افزایش زمان محاسباتی مدل MOMILP حین بزرگ تر شدن مسأله، روش GA با انحراف  $4.85\%$  از بهینگی کارا تر می نماید. همچنین، هرچند که الگوریتم SA انحراف نسبتاً زیادی از جواب بهینه نشان می دهد ولی صرفه زمانی آن در مقایسه با GA روشن است.

جهت حل مسأله های با ابعاد بزرگ از الگوریتم های MOPGA و MOPSA استفاده کرده و هر مسأله ۵ بار با هر یک از الگوریتم ها حل می شود. بعلاوه، بمنظور مقایسه الگوریتم ها، نتایج بدست آمده به معیار RPD تبدیل و در جدول (۱۱) نشان داده شده اند. بنابراین، الگوریتم MOPGA با میانگین RPD معادل  $0.05\%$  بسیار کارا تر و پایا تر از الگوریتم MOPSA در برخورد با مسأله های بزرگ عمل می کند.

الگوریتم های MOPGA و MOPSA نیز با زبان ویژوال بیسیک نت ۲۰۰۸ برنامه نویسی شده اند. همچنین، تمامی اجراها اعم از مدل MOMILP و الگوریتم های GA، SA، MOPGA و MOPSA همگی در رایانه ای با پردازنده (CPU) ۳ GHz و Intel® Core™ i ۲,۲۶ و ۴GB حافظه موقت (RAM) اجرا شده اند.

#### ۴-۲. ارزیابی محاسباتی

بمنظور بررسی کارایی مدل MOMILP و الگوریتم های پیشنهادی، دو مجموعه مثال شامل مسأله های با ابعاد کوچک ( $n=4,56$  و  $m=2,3$ ) و مسأله های با ابعاد بزرگ ( $n=10,20$  و  $m=5,10$ ) تولید شده اند [۲۶]. با در نظر گرفتن یک تکرار برای هر یک از مسأله های با ابعاد کوچک و ۵ تکرار برای هر یک از مسأله های با ابعاد بزرگ، در مجموع ۲۶ مسأله خواهیم داشت. مابقی پارامترها دقیقاً مانند آنچه در بخش تولید داده ها آمد در نظر گرفته می شوند. با در نظر گرفتن وزن ها بصورت  $\theta_p = \theta_1 = 0.5$ ، مسأله های با کوچک ابعاد با الگوریتم های GA و SA اولیه حل و نتایج بدست آمده با جواب بهینه حاصل از مدل MOMILP با وزن های مذکور مقایسه می شوند. مدل MOMILP در نرم افزار لینگو ۸ کدنویسی شده و از زبان ویژوال بیسیک نت ۲۰۰۸ برای کدنویسی الگوریتم های GA و SA اولیه استفاده شده است. عملکرد مدل MOMILP و الگوریتم های GA و SA اولیه در جدول (۱۰) نشان داده شده است. همانگونه که از جدول پیداست، گرچه مدل MOMILP روش مناسبی در برخورد با مسأله های

جدول ۱۰. عملکرد مدل MOMILP و الگوریتم‌های GA و SA اولیه در برخورد با مسأله‌های با ابعاد کوچک ( $\theta_1 = \theta_2 = 0.5$ )

مسأله	Z			انحراف از بهینگی (%)		زمان حل (ثانیه)			m. n. b
	مدل MOMILP	GA	SA	GA	SA	مدل MOMILP	GA	SA	
۱	۴۴۹,۷۴	۴۴۹,۷۴	۵۱۰,۵۳	%۰,۰۰	%۱۳,۵۲	۱۱	۸۴	۱	
۲	۳۴۷,۳۱	۳۵۰,۸۳	۶۶۸,۹۰	%۱,۰۲	%۹۲,۶۰	۶۰	۱۰۵	۱	
۳	۴۹۲,۹۵	۴۹۲,۹۵	۷۲۶,۷۱	%۰,۰۰	%۴۷,۴۲	۱۴۵	۱۰۶	۱	
۴	۳۸۲,۶۵	۴۶۶,۳۸	۷۳۷,۰۰	%۲۱,۸۸	%۹۲,۶۱	۱۹۵۱	۱۴۷	۲	
۵	۸۴۸,۰۹	۸۵۹,۴۳	۱۴۱۵,۳۳	%۱,۳۴	%۶۶,۸۷	۱۳۸۶	۱۰۵	۳	
۶	قطع اجرا	۱۰۲۶,۹۷	۲۳۷۵,۳۶	-	-	۳۸۷۴۴	۱۶۸	۴	
میانگین				%۴,۸۵	%۶۲,۶۰				

جدول ۱۱. میانگین RPD برای الگوریتم‌های MOPGA و MOPSA در حل مسأله‌های با ابعاد بزرگ

مسأله	m. n. b	MOPGA	MOPSA	مسأله	m. n. b	MOPGA	MOPSA
۷	۵,۱۰,۲۰	۰,۰۳	۰,۰۶۹	۱۷	۵,۲۰,۴۰	۰,۰۵	۰,۰۳۹
۸	۵,۱۰,۲۰	۰,۰۳	۰,۰۶۲	۱۸	۵,۲۰,۴۰	۰,۰۱	۰,۰۴۷
۹	۵,۱۰,۲۰	۰,۰۷	۱,۱۰	۱۹	۵,۲۰,۴۰	۰,۰۵	۰,۰۴۸
۱۰	۵,۱۰,۲۰	۰,۰۲	۰,۰۷۶	۲۰	۵,۲۰,۴۰	۰,۰۴	۰,۰۶۵
۱۱	۵,۱۰,۲۰	۰,۰۴	۰,۰۸۰	۲۱	۵,۲۰,۴۰	۰,۰۴	۰,۰۵۴
۱۲	۱۰,۱۰,۲۰	۰,۰۸	۰,۰۷۳	۲۲	۱۰,۲۰,۴۰	۰,۰۱	۰,۰۳۰
۱۳	۱۰,۱۰,۲۰	۰,۱۱	۰,۰۸۰	۲۳	۱۰,۲۰,۴۰	۰,۰۳	۰,۰۲۴
۱۴	۱۰,۱۰,۲۰	۰,۰۷	۰,۰۷۳	۲۴	۱۰,۲۰,۴۰	۰,۰۴	۰,۰۲۸
۱۵	۱۰,۱۰,۲۰	۰,۰۶	۰,۰۸۰	۲۵	۱۰,۲۰,۴۰	۰,۰۷	۰,۰۴۵
۱۶	۱۰,۱۰,۲۰	۰,۰۸	۰,۰۶۱	۲۶	۱۰,۲۰,۴۰	۰,۰۶	۰,۰۴۱

میانگین RPD الگوریتم MOPGA: ۰,۰۵

میانگین RPD الگوریتم MOPSA: ۰,۵۹

بزرگ روشن می‌نمود. بنابراین، الگوریتم‌های فرا ابتکاری GA و SA برای مسأله‌های تک هدفه و نیز چند هدفه‌ای که وزن هر یک از هدف‌ها معلوم است ارائه شدند. علاوه، الگوریتم‌های MOPGA و MOPSA برای مسأله‌های چند هدفه‌ای که وزن هدف‌ها نامعلوم است معرفی شدند. همچنین، پایایی الگوریتم‌های چند هدفه از طریق طراحی آزمایشات تاکوچی بررسی شد. نتایج نشان داد که الگوریتم MOPGA کارایی بهتری نسبت به الگوریتم MOPSA در برخورد با مسأله‌های با ابعاد بزرگ دارد. در ادامه، به برخی زمینه‌های توسعه و گسترش مقاله حاضر اشاره می‌شود:

- توسعه روش‌های حل دیگر شامل روش‌های ابتکاری و فرا ابتکاری جدید جهت حل مسأله‌های با ابعاد بزرگ و مقایسه با الگوریتم‌های پیشنهادی این مقاله می‌تواند زمینه مناسبی در جهت هرچه کاربردی‌تر کردن مدل مورد نظر باشد.
- در نظر گرفتن معیارهای بهینه‌سازی متنوع‌تر با توجه به نیاز روز صنعت نیز می‌تواند زمینه مطالعات آتی را فراهم سازد.
- در این مقاله بهینه‌سازی فاکتورهای مربوط به الگوریتم‌ها در فضای گسسته صورت گرفت. بنابراین، از دیگر

## ۵. نتیجه‌گیری و مطالعات آتی

در این مقاله، یک مدل برنامه‌ریزی ریاضی جدید خطی و عدد صحیح مختلط چند هدفه (MOMILP) برای مسأله دو هدفه زمان‌بندی کارگاه باز با در نظر گرفتن زمان‌های عدم دسترسی ماشین‌ها ارائه شد. همچنین، "حداکثر زمان تکمیل کارها" و "کل دیرکردها" بطور همزمان بعنوان معیارهای بهینه‌سازی در نظر گرفته شدند. از دیگر شرایط و ویژگی‌های خاص محیط کارگاه باز مورد نظر به موارد زیر اشاره می‌شود:

- زمان‌های دسترسی و (عدم دسترسی) وابسته به ماشین هستند.
- زمان‌های جابجایی نامتقارن و وابسته به کار در نظر گرفته شد.
- بریدگی کارها مجاز نیست.

کارایی مدل MOMILP از طریق تعدادی مسأله با ابعاد کوچک بررسی و جهت حل مدل و رسیدن به جواب بهینه سراسری از نرم-افزار لینگو ۸ استفاده شد. گرچه نتایج حاکی از توانایی بالای مدل MOMILP در حل مسأله‌های با ابعاد کوچک بود، ولی تحلیل-های انجام گرفته روی مدل ضرورت توسعه روش‌های دیگر نظیر روش‌های ابتکاری و فرا ابتکاری را برای حل مسأله‌های با ابعاد

- makespan*”, Applied Mathematics and Computation, Vol. 215, 2010, pp. 3929–3935.
- [7] Sun, K., Li, H., “Scheduling problems with multiple maintenance activities and non-preemptive jobs on two identical parallel machines”, International Journal of Production Economics, Vol. 124, 2010, pp. 151-158.
- [۸] گل‌مکانی، حمیدرضا، نمازی، علی، «زمانبندی کار کارگاهی چندمسیره با در نظر گرفتن محدودیت نگهداری و تعمیرات (نت)»، نشریه بین المللی مهندسی صنایع و مدیریت تولید، جلد ۲۳، شماره ۴، صفحه ۴۷۰-۴۵۹، سال ۱۳۹۱.
- [9] Ma, Y., Chu, C., Zuo, C., “A survey of scheduling with deterministic machine availability constraints”, Computers & Industrial Engineering, Vol. 58, 2010, pp. 199-211.
- [10] Strusevich, V.A., “A heuristic for the two-machine open-shop scheduling problem with transportation times”, Discrete Applied Mathematics, Vol. 93, 1999, pp. 287-304.
- [11] Panahi, H., Rabbani, M., Tavakkoli-Moghaddam, R., “Solving an Open Shop Scheduling Problem by a Novel Hybrid Multi-Objective Ant Colony Optimization”, Eighth International Conference on Hybrid Intelligent Systems, 2008, pp. 320-325.
- [12] Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P., Werner, F., “Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria”, International Journal of Advanced Manufacturing Technology, Vol. 37, 2008, pp. 354-370.
- [13] Panahi, H., Tavakkoli-Moghaddam, R., “Solving a multi-objective open shop scheduling problem by a novel hybrid ant colony optimization”, Expert Systems with Applications, Vol. 38, 2011, pp. 2817–2822.
- زمینه‌های توسعه مقاله حاضر می‌تواند بهینه‌سازی فاکتورها در فضای پیوسته باشد.
- می‌توان زمان‌های آماده‌سازی و جداسازی وابسته به توالی در نظر گرفته و مدل ریاضی را توسعه داد.
  - تعداد وسایل نقلیه را می‌توان بعنوان محدودیت در مسأله اعمال کرد.
  - در نظر گرفتن سرعت متغیر برای ماشین‌ها امری کاربردی بنظر می‌رسد. بدین معنی که، سرعت پردازش ماشین‌ها پس از انجام تعداد مشخصی کار یا سپری شدن مدت زمان معین کاهش و پس از انجام فرآیند نگهداری و تعمیرات دوره‌ای مجدداً افزایش یابد.
- ### مراجع
- [1] Leung, J.Y.T., “Handbook of Scheduling: Algorithms, Models, and Performance Analysis”, Chapman & Hall/CRC Computer and Information Science Series, 2004, pp. 113-126.
- [2] Blazewicz, J., Ecker, K.H., Pesch, E., Schmidt, G., Weglarz, J., “Handbook on Scheduling: From Theory to Applications”, International Handbooks on Information Systems, Berlin: Springer, 2007, pp. 321-344.
- [3] Kellerer, H., Kubzin, M.A., Strusevich, V.A., “Two Simple Constant Ratio Approximation Algorithms for Minimizing the Total Weighted Completion Time on a Single Machine with a Fixed Non-Availability Interval”, European Journal of Operational Research, Vol. 199(1), 2009, pp. 111-116.
- [4] Breit, J., Schmidt, G., Strusevich, V.A., “Two-machine open shop scheduling with an availability constraint”, Operations Research Letters, Vol. 29, 2001, pp. 65-77.
- [5] Breit, J., Schmidt, G., Strusevich, V.A., “Non-preemptive two-machine open shop scheduling with non-availability constraints”, Mathematical Methods of Operations Research, Vol. 57, 2003, pp. 217-234.
- [6] Hsu, C.J., Low, C., Su, C.T., “A single-machine scheduling problem with maintenance activities to minimize



- European Journal of Operational Research, Vol. 124, 2000, pp. 28-42.
- [24] Matta, M.E., "A genetic algorithm for the proportionate multiprocessor open shop", Computers & Operations Research, Vol. 36, 2009, pp. 2601-2618.
- [25] Naderi, B., Fatemi Ghomi, S.M.T., Aminnayeri, M., Zandieh, M., "Scheduling open shops with parallel machines to minimize total completion time", Journal of Computational and Applied Mathematics, Vol. 235, 2011, pp. 1275-1287.
- [26] Tavakkoli-Moghaddam, R., Safaei, N., Sasani, F., "A memetic algorithm for the flexible flow line scheduling problem with processor blocking", Computers & Operations Research, Vol. 36(2), 2009, pp. 402-414.
- [27] Low, C., Yeh, Y., "Genetic algorithm-based heuristics for an open shop scheduling problem with setup, processing, and removal times separated", Robotics and Computer-Integrated Manufacturing, Vol. 25, 2009, pp. 314-322.
- [28] Tavakkoli-Moghaddam, R., Rahimi-Vahed, A.R., Ghodrathnama, A., Siadat, A., "A simulated annealing method for solving a new mathematical model of a multi-criteria cell formation problem with capital constraints", Advances in Engineering Software, Vol. 40, 2009, pp. 268-273.
- [29] Cheng, B.W., Chang, C.L., "A study on flowshop scheduling problem combining Taguchi experimental design and genetic algorithm", Expert Systems with Applications, Vol. 32, 2007, pp. 415-421.
- [30] Naderi, B., Zandieh, M., Fatemi Ghomi, S.M.T., "Scheduling sequence-dependent setup time job shops with preventive maintenance", International Journal of Advanced Manufacturing Technology, Vol. 43, 2009, pp. 170-181.
- [31] Ruiz, R., Garcia-Diaz, J.C., Maroto, C., "Considering scheduling and preventive maintenance in the flowshop sequencing problem", Computers & Operations Research, Vol. 34, 2007, pp. 3314-3330.
- [۱۴] امیری، نفیسه، توکلی مقدم، رضا، قلی‌پور کنعانی، یوسف، ترابی، سیدعلی، «مدل‌سازی یک مسأله زمان‌بندی کارگاه باز چند هدفه جدید و حل آن با استفاده از روش جستجوی پراکنده»، نشریه بین‌المللی مهندسی صنایع و مدیریت تولید، جلد ۲۳، شماره ۲، صفحه ۱۶۴-۱۵۳، سال ۱۳۹۱.
- [15] open shops to minimize total tardiness", European Journal of Operational Research, Vol. 162, 2005, pp. 173-183.
- [16] Sedeno-Noda, A., Alcaide, D., Gonzalez-Martin, C., "Network flow approaches to pre-emptive open-shop scheduling problems with time-windows", European Journal of Operational Research, Vol. 174, 2006, pp. 1501-1518.
- [۱۷] حمیدرضا، فیلی، کریمی نسب، مهدی، حداد، حمیدرضا، بابایی، محمد حسین، «حل مسئله زمان‌بندی تولید دسته ای در محیط ماشین‌های موازی چند کاره به کمک دو روش فراابتکاری»، نشریه بین‌المللی مهندسی صنایع و مدیریت تولید، جلد ۲۴، شماره ۲، صفحه ۲۳۷-۲۲۵، سال ۱۳۹۲.
- [18] Drezner, Z., "A distance based rule for removing population members in genetic algorithms", 4OR, Vol. 3, 2005, pp. 109-116.
- [19] Potvin, J.Y., "Genetic algorithm for the traveling salesman problem", Annals of Operations Research, Vol. 63, 1996, pp. 339-370.
- [20] Fogel, L.J., Owens, A.J., Walsh, M.J., "Artificial Intelligence through Simulated Evolution", Wiley, 1996.
- [21] Holland, J.H., "Adaptation in Natural and Artificial Systems", The university of Michigan press, Ann Arbor, 1975, reprinted by MIT press, 1992.
- [22] Gen, M., Cheng, R., "Genetic Algorithms & Engineering Design", Wiley, New York, 1997.
- [23] Liaw, C.F., "A hybrid genetic algorithm for the open shop scheduling problem",

- [32] Rajendran, C., Ziegler, H., “*Scheduling to minimize the sum of weighted flowtime and weighted tardiness of jobs in a flowshop with sequence-dependent setup times*”, European Journal of Operational Research, Vol. 149, 2003, pp. 513–522.
- [33] Kyparisis, G.J., Koulamas, C., “*Open shop scheduling with makespan and total completion time criteria*”, Computers & Operations Research, Vol. 27, 2000, pp. 15-27.
- [34] Rashidi, E., Jahandar, M., Zandieh, M., “*An improved hybrid multi-objective parallel genetic algorithm for hybrid flow shop scheduling with unrelated parallel machines*”, International Journal of Advanced Manufacturing Technology, Vol. 49, 2010, pp. 1129-1139.