



SOLVING AN OPEN SHOP SCHEDULING PROBLEM WITH MACHINE FLEXIBILITY BY A GENETIC ALGORITHM

N. Kamali, R. Tavakkoli-Moghaddam*, A. Shojaei-Barjouei & B. Naderi

N. Kamali, MSc Department of Industrial Engineering, Mazandaran University of Science and Technology.

R. Tavakkoli-Moghaddam, Professor Department of Industrial Engineering, College of Engineering, University of Tehran.

A. Shojaei-Barjouei, MSc Department of Industrial Engineering, Mazandaran University of Science and Technology.

B. Naderi, Assistance Professor Department of Industrial Engineering, College of Engineering, University of Tehran.

Keywords

Open shop scheduling,
Machine flexibility,
Genetic algorithm,
Taguchi method,
Lower bound

ABSTRACT

This paper presents an open shop scheduling problem with machine flexibility. Machine flexibility means that each operation can be processed on all or some of the machines in the shop. The travel time between machines is also considered. The aim of this problem is to minimize the mean completion time. In order to solve small-sized instances a new mixed-integer linear programming (MILP) model is presented. Due to the complexity of solving such a NP-hard problem, a genetic algorithm (GA) is proposed and designed. Some individuals of an initial population in the proposed GA are generated by a heuristic method, named OSSF-PopGen. By this method, the machine with the minimum processing time is not necessarily considered as the selected machine to process; however, the processing time on each machine and the travel time are used to select the appropriate one. Furthermore, the effect of different parameters and operators on the performance of the proposed GA is explored by the use of the Taguchi method. To the algorithm, the results of a lower bound are computed and compared with the associated results obtained by the proposed GA.

© 2016 IUST Publication, IJIEPM. Vol. 27, No. 2, All Rights Reserved



حل مسأله زمان بندی در محیط کارگاه باز با ماشین های انعطاف پذیر با الگوریتم ژنتیک

نرجس کمالی، رضا توکلی مقدم*، ابوالفضل شجایی برجویی و بهمن نادری

چکیده:

در این مقاله، مسأله زمانبندی در محیط کارگاه باز با ماشین های انعطاف پذیر ارائه می شود. انعطاف پذیری ماشین ها بدان معناست که هر عملیات می تواند بر روی همه و یا برخی از ماشین های کارگاه پردازش شود. همچنین زمان حمل و نقل بین ماشین ها در نظر گرفته می شود. هدف این مسأله، کمینه سازی میانگین زمان تکمیل کارها است. به منظور حل مسأله های با ابعاد کوچک یک مدل برنامه ریزی ریاضی خطی مختلط ارائه می گردد. این مسأله با توجه به ماهیت پیچیده آن، جز خانواده مسایل NP-hard است، بنابراین یک الگوریتم ژنتیک برای حل آن ارائه و طراحی می شود. برای تولید بخشی از جواب اولیه آن از یک روش ابتکاری استفاده می شود. با بکارگیری این روش، ماشین با کمترین زمان پردازش، لزوماً ماشین منتخب برای پردازش هر عملیات نیست، بلکه زمان پردازش عملیات روی هر ماشین و زمان حمل و نقل بین ماشین ها ملاک انتخاب قرار می گیرد. تاثیر پارامترها و عملگرهای مختلف بر روی عملکرد الگوریتم ژنتیک پیشنهادی با روش تاگوچی صورت می گیرد. همچنین، یک کران پایین تقریبی برای مسأله محاسبه شده و نتایج مربوطه با الگوریتم ژنتیک پیشنهادی مقایسه می شود.

کلمات کلیدی

زمانبندی کارگاه باز،
انعطاف پذیری ماشین،
الگوریتم ژنتیک،
روش تاگوچی،
کران پایین

۱. مقدمه

زمانبندی یک فرآیند تصمیم گیری است که در بسیاری از صنایع تولیدی و خدماتی به کار می رود. تخصیص منابع به فعالیت ها در یک دوره زمانی به منظور بهینه سازی یک یا چند هدف، وظیفه ای است که زمانبندی بر عهده دارد [۱]. یک مسأله زمانبندی در محیط کارگاه باز، با مجموعه ای از عملیات مواجه است که در آنها ترتیب خاصی رعایت نمی شود.

بنابراین فضای حل این مسأله به طور قابل ملاحظه ای از فضای

تاریخ وصول: ۹۲/۱۱/۰۵

تاریخ تصویب: ۹۳/۰۴/۲۴

نرجس کمالی، گروه مهندسی صنایع، دانشگاه علوم و فنون مازندران،
n_kamali28@yahoo.com
ابوالفضل شجایی برجویی، گروه مهندسی صنایع، دانشگاه علوم و فنون مازندران،
shojaei.b@gmail.com
بهمن نادری، گروه مهندسی صنایع، دانشکده فنی و مهندسی، دانشگاه خوارزمی،
کرج، bahman.naderi@aut.ac.ir

*نویسنده مسئول مقاله: رضا توکلی مقدم، دانشکده مهندسی صنایع، پردیس
دانشکده های فنی، دانشگاه تهران، tavakoli@ut.ac.ir

جواب مسایل زمانبندی نظیر زمانبندی کار کارگاهی، و زمانبندی جریان کارگاهی بزرگتر است. در این محیط m ماشین وجود دارد که هر کار باید روی هر یک از این m ماشین برود، البته زمان پردازش کارها روی برخی از ماشین ها صفر است، محدودیتی در ارتباط با مسیر پردازش هر کار روی ماشین های کارگاه وجود ندارد. در زمانبندی این کارگاه هر کار مجاز است که مسیر پردازش خود را داشته باشد و مسیر پردازش کارهای مختلف می تواند متفاوت از هم باشد.

حوزه مورد بررسی در این مقاله، مسأله زمانبندی در محیط کارگاه باز با ماشین های انعطاف پذیر می باشد. در این مسأله n کار داریم که هر کار شامل p عملیات است که باید روی ماشین ها پردازش شوند. ماشین ها انعطاف پذیرند بدین معنی که توانایی انجام چندین کار در کارگاه را دارند.

مسأله با ۲ کار، ۳ عملیات و ۲ ماشین را مطابق جدول شماره (۱) در نظر بگیرید. عملیات اول از کار اول، می تواند بر روی ماشین اول و دوم انجام شود و اینکه نهایتاً ماشین بهینه برای پردازش این عملیات کدامیک از این دو ماشین است تصمیمی است که باید گرفته شود تا هدف مورد نظر بهینه گردد. زمان حمل و نقل بین

ماشین‌ها در نظر گرفته شده است. برش و قطع عملیات‌ها مجاز نیست. بنابراین در مسأله زمانبندی در محیط کارگاه باز با ماشین‌های انعطاف‌پذیر سه تصمیم با در نظر گرفتن تابع هدف، باید به صورت همزمان گرفته شود: (۱) تخصیص ماشین به هر یک از عملیات‌ها، با انتخاب ماشین مناسب از بین ماشین‌هایی که توان پردازش عملیات فوق را دارند. (۲) تعیین توالی عملیات‌های هر کار بر روی ماشین‌ها. (۳) تعیین توالی ماشین‌ها برای پردازش عملیات‌های هر کار.

در ادبیات مسایل زمانبندی، مسأله کارگاه باز با ماشین‌های انعطاف‌پذیر بندرت مورد مطالعه قرار گرفته است. البته حالت‌هایی در ادبیات در نظر گرفته شده که در آن ماشین‌ها به صورت موازی و چند پردازشی هستند. نادری و همکاران [۲] مسأله کارگاه باز با ماشین‌های موازی را با هدف کمینه سازی کل زمان تکمیل کارها، مورد بررسی قرار دادند. که در آن n کار باید بر روی مجموعه‌ای از m مرحله کاری پردازش شوند، که هر مرحله شامل تعدادی ماشین موازی یکسان است. شوارمن و وگینگر [۳] مسأله کارگاه باز چند مرحله‌ای چند پردازنده را مطالعه کردند. در این مدل از کارگاه باز، s مرحله وجود دارد که هر مرحله شامل تعدادی ماشین موازی است. هر کاری شامل s عملیات متناسب با s مرحله موجود است که با هر توالی دلخواه می‌توانند پردازش شوند. جانسن و اسوریدنکو [۴]، سواستیانوف و وگینگر [۵] و کنونف و سویریدنکو [۶] نیز مسأله کارگاه باز و جریان کارگاهی چند مرحله‌ای چند پردازنده را مطالعه کردند. ماتا [۷] حالت خاصی از مسأله کارگاه باز چند پردازنده را برای مرکز پزشکی دانشگاه بکار گرفت که ترکیبی از محیط انعطاف‌پذیر جریان کارگاهی و محیط کارگاه باز چند پردازنده ای است. ماتا و المغربی [۸] دو کلاس خاص از مسأله کارگاه باز چند پردازنده‌ای را برای یک مرکز تست تشخیص پزشکی بررسی کردند که صدها بیمار را در محیط کارگاه باز با تسهیلات چند پردازنده‌ای، زمانبندی می‌کند. کیس و همکاران [۹] گسترشی از مسأله کارگاه باز چند پردازنده را در حالتی که برش مجاز نباشد بررسی کردند که در آن مجموعه پردازنده‌ها به گروه‌های پردازشی تقسیم شده‌اند.

توجه به حمل و نقل و لحاظ نمودن آن در برنامه‌ریزی‌ها علاوه بر روان تر کردن جریان مواد و کاهش اختلالات ناشی از آن، تأثیر بسزایی در کاهش هزینه‌ها خواهد داشت. استراسویچ [۱۰] زمان‌های حمل و نقل را در مسأله کارگاه باز دو ماشین در نظر گرفته است. به‌علاوه، وی این زمان‌های حمل و نقل را معادل زمان‌های فرآیندهای گرم کردن و یا سرد کردن در کاربردهای شیمیایی و متالورژیکی می‌داند. آلاوی و آرتیبا [۱۱] زمان‌های حمل و نقل را در مسأله زمانبندی جریان کارگاهی تلفیقی^۱ در نظر گرفته و با توجه به محدود بودن وسایل جابجایی، شروع زمان حمل و نقل را مشروط به وجود وسایل دانسته‌اند.

بروکر و همکاران [۱۲] نشان دادند که تقریباً همه مسایل زمانبندی کارگاهی با دو کار و با هر تابع هدفی می‌توانند در زمان چند جمله‌ای حل شوند. این در حالی است که با بیش از دو کار مسأله NP-hard می‌شود. با توجه به این پیچیدگی برای مسأله زمانبندی کارگاه باز، ضرورت بکارگیری روش‌های ابتکاری و فراابتکاری مشهود است. نادری و همکاران [۱۳] دو الگوریتم فراابتکاری بر اساس الگوریتم ژنتیک و جستجوی همسایگی برای مسأله زمانبندی کارگاه باز طراحی کردند. لو و یه [۱۴] الگوریتم ژنتیک ترکیب شده با جستجوی ممنوعه را بر مبنای روش‌های ابتکاری برای مسأله زمانبندی کارگاه باز ارائه نمودند. آنها الگوریتم پیشنهادی خود را با روش‌های فراابتکاری کلاسیک مانند الگوریتم ژنتیک، شبیه سازی تبرید و جستجوی ممنوعه مقایسه کردند. سراج و توکلی مقدم [۱۵] یک مدل ریاضی برای مسأله زمانبندی در محیط کارگاه باز ارائه کردند. برای حل مسأله با اندازه‌ی متوسط و بزرگ در زمان منطقی، یک روش جستجوی ممنوع پیشنهاد شده است. اندرسون و همکاران [۱۶] الگوریتمی بر مبنای شبیه سازی تبرید و الگوریتم ژنتیک برای مسأله زمانبندی کارگاه باز ارائه نمودند. شا و سو [۱۷] الگوریتمی را بر مبنای الگوریتم ذرات انبوه ارائه دادند و آن را با جستجوی بیم ترکیب نمودند. لیاو [۱۸] الگوریتمی را بر مبنای جستجوی ممنوع برای حل مسأله زمانبندی کارگاه باز دو ماشین با فرض مجاز بودن بریدگی کارها ارائه داده است. لیاو [۱۹] یک الگوریتم ژنتیک ترکیبی با جستجوی ممنوعه را برای مسأله زمانبندی کارگاه باز ارائه نموده است. لیاو [۲۰] الگوریتمی بر مبنای جستجوی ممنوع برای مسأله زمانبندی کارگاه باز ارائه کرده است. وی یک روش تقریبی برای یافتن کمترین مقدار حداکثر زمان تکمیل کارها در یک مسأله کارگاه باز با فرض غیر مجاز بودن بریدگی ارائه نموده است. بلوم [۲۱] از الگوریتم فراابتکاری کلونی مورچگان برای بهینه سازی مسأله کارگاه باز استفاده کرد. این الگوریتم با جستجوی بیم که یکی از روش‌های مشهور جستجوی درختی^۲ است ترکیب شده و نتایج مطلوبی را ایجاد کرده است.

معیارهای اندازه‌گیری متفاوتی برای بهینه سازی در ادبیات زمانبندی بکار می‌رود، کمینه‌سازی، حداکثر زمان تکمیل کارها، معیاری است که توسط بسیاری از محققان به کار گرفته شده است [۱۹-۲۰]، [۳-۹]. کمینه سازی میزان تاخیر کل معیار دیگری است که در برخی از پژوهش‌ها بکار گرفته شده است [۱۳-۱۸].

لذا تحقیقات بسیار اندکی روی مسأله کارگاه باز با ماشین‌های انعطاف‌پذیر صورت گرفته است. از آن جمله می‌توان به ویتکوفسکی و همکاران [۲۲] اشاره نمود که در آن ترکیب الگوریتم شبیه‌سازی تبرید^۳ با دستاورد چند عاملی^۴ جهت کمینه سازی بیشترین زمان تکمیل در محیط کارگاه باز با ماشین‌های انعطاف‌پذیر بکار گرفته شده است.

ژنتیک پیشنهادی با روش تاگوچی تنظیم پارامتر شده و نتایج حل آن با کران پایین ارایه شده برای مسأله مقایسه شده است.

در این مقاله، برای حل مسأله زمان بندی کارگاه باز با ماشین‌های انعطاف پذیر یک الگوریتم ژنتیک طراحی شده است که بخشی از جمعیت اولیه آن با یک الگوریتم ابتکاری تولید می‌شود. الگوریتم

جدول ۱. کارگاه باز با ماشین‌های انعطاف پذیر

عملیات‌ها					
کارها	۱	۲	۳		
شماره ماشین	زمان عملیات	شماره ماشین	زمان عملیات	شماره ماشین	زمان عملیات
کار ۱	۱	۲	۴	۱	۸
	۲			۲	۲۷
کار ۲	۱	۱	۵	۱	۷
				۲	۹

۲-۲. اندیس‌ها و پارامترها

i, j	اندیس‌های کار ($i, j=1, 2, \dots, n$)
k, l	اندیس‌های ماشین ($k, l=1, 2, \dots, m$)
h, e	اندیس‌های عملیات ($h, e=1, 2, \dots, p$)
M	یک عدد مثبت بزرگ
O_{jh}	h امین عملیات کار j
$\alpha_{jnk} = \begin{cases} 1 & \text{اگر } O_{jh} \text{ بتواند روی ماشین } k \text{ پردازش شود.} \\ 0 & \text{در غیر این صورت} \end{cases}$	
Tr_{kl}	زمان جابجایی از ماشین k به ماشین l
β_{jnk}	زمان پردازش O_{jh} روی ماشین k

۲-۳. متغیرهای تصمیم مسأله

st_{jhk}	زمان شروع پردازش O_{jh} روی ماشین k
$Y_{jnk} = \begin{cases} 1 & \text{اگر ماشین } k \text{ برای پردازش } O_{jh} \text{ انتخاب شود.} \\ 0 & \text{در غیر این صورت} \end{cases}$	
$Z_{jhiek} = \begin{cases} 1 & \text{اگر عملیات } O_{jh} \text{ قبل از عملیات } O_{ie} \text{ روی ماشین } k \text{ پردازش شود.} \\ 0 & \text{در غیر این صورت} \end{cases}$	
$V_{jhe} = \begin{cases} 1 & \text{اگر عملیات } O_{jh} \text{ قبل از عملیات } O_{je} \text{ پردازش شود.} \\ 0 & \text{در غیر این صورت} \end{cases}$	
C_j	j زمان تکمیل کار

۲-۴. مدل برنامه ریزی خطی مختلط عدد صحیح

با در نظر گرفتن فرضیات، اندیس‌ها، پارامترها و متغیرهای تصمیم ذکر شده، مسأله مورد نظر به صورت برنامه ریزی ریاضی خطی مختلط، با تابع هدف حداقل میانگین زمان تکمیل کارها مدل سازی شده است.

در ادامه، در بخش ۲ مدل ریاضی برنامه ریزی خطی مختلط ارائه می‌شود. الگوریتم ژنتیک پیشنهادی و جزئیات آن در بخش ۳ تشریح می‌شود. بخش ۴ تنظیم پارامتر الگوریتم با روش تاگوچی را توضیح می‌دهد. نتایج الگوریتم پیشنهادی بر روی مثال‌های تولید شده، در بخش ۵ آمده است. بخش ۶ نتیجه‌گیری را ارایه می‌دهد.

۲. مدل ریاضی

در این بخش بعد از بیان فرضیات مسأله و معرفی اندیس‌ها، پارامترها و متغیرهای تصمیم، یک مدل برنامه ریزی ریاضی خطی مختلط برای مسأله زمان بندی در محیط کارگاه باز با ماشین‌های انعطاف پذیر ارائه خواهد شد.

۲-۱. مفروضات مسأله

اهم مفروضات مسأله به شرح زیر است:

- هر ماشین در هر لحظه از زمان حداکثر یک کار را می‌تواند پردازش کند.
- همپوشانی عملیات‌های یک کار غیر مجاز است (هر کار در هر لحظه از زمان حداکثر روی یک ماشین می‌تواند پردازش شود).
- هر کار دارای تعدادی عملیات است که هر کدام از آنها قابلیت انجام بر روی همه و یا برخی از ماشین‌ها را دارند و باید تنها یک ماشین برای پردازش هر عملیات انتخاب شود.
- عملیات‌های یک کار در هر لحظه از زمان حداکثر روی یک ماشین می‌توانند پردازش شوند.
- ترتیب انجام عملیات‌های یک کار مهم نیست.
- همه کارها در ابتدای دوره برنامه ریزی (در زمان صفر) در کارگاه در دسترس هستند.
- بریدگی کارها و خرابی ماشین‌ها مجاز نیست.
- زمان‌های پردازش برای هر عملیات از هر کار روی هر ماشین، وابسته به ماشین تخصیص یافته هستند.

روی یک ماشین می‌تواند پردازش شود. توالی عملیات‌های یک کار با محدودیت (۹) محاسبه می‌شود. محدودیت (۱۰) نشان می‌دهد که اگر دو عملیات از یک کار روی یک ماشین متوالی‌اند، در توالی مربوط به کار خود نیز این دو عملیات متوالی خواهند بود. محدودیت (۱۱) نشان می‌دهد که هر عملیات فقط روی ماشین تخصیص یافته خود، دارای زمان شروع خواهد بود. محدودیت (۱۲) زمان تکمیل هر عملیات را محاسبه می‌کند. در نهایت، محدودیت‌های (۱۳) و (۱۴) متغیرهای عدد صحیح و مثبت را ارایه می‌کنند.

۳. روش الگوریتم ژنتیک پیشنهادی

الگوریتم ژنتیک روشی است فراابتکاری که با ایده گرفتن از سیستم‌های طبیعی، به جستجو برای یافتن راه حل‌های نزدیک به بهینه برای مسایل بهینه سازی می‌پردازد. این روش جستجو با یک جمعیت از جواب‌ها که جمعیت اولیه نام دارند، آغاز می‌گردد و در این جمعیت هر جواب از طریق یک کروموزوم نمایش داده می‌شود. در این الگوریتم، جواب‌های بدست آمده از یک جمعیت برای تولید جمعیت بعد بکار می‌رود. در هر نسل اپراتورهای الگوریتم روی کروموزوم‌ها عمل نموده و نسل بعد را تولید می‌کنند. با پیشرفت روند الگوریتم نسل‌ها بهبود می‌یابند و جواب‌های بهتری بدست می‌آید.

روند کار الگوریتم ژنتیک طراحی شده در این پژوهش بدین صورت است که، نیمی از جمعیت اولیه به طور تصادفی و نیم دیگر آن با یک الگوریتم ابتکاری تولید می‌شود. سپس تابع شایستگی برای همه کروموزوم‌های جمعیت محاسبه می‌شود و درصد ثابتی از بهترین کروموزوم‌ها مستقیماً به نسل بعد منتقل می‌شوند (اپراتور نخبه گرایی). در ادامه استراتژی چرخ رولت^۵ برای نسل فعلی بکار گرفته می‌شود تا کروموزوم‌هایی را که شایستگی بیشتری دارند انتخاب کند. بعد از این مرحله اپراتور تقاطع با احتمال P_c بر روی جمعیت اعمال می‌شود و کروموزوم‌هایی که تقاطع یافته‌اند جایگزین والد‌های خود می‌شوند تا با احتمال P_m تحت عمل جهش قرار گیرند. با محاسبه تابع شایستگی کروموزوم‌های نسل اول، بهترین کروموزوم نسل معرفی می‌شود تا با بهترین کروموزومی که تاکنون داشته‌ایم مقایسه شود که اگر شایستگی بیشتری دارد کران بالای جواب به روز شود. این روند ادامه می‌یابد تا شرط توقف برقرار شود که در اینجا شرط توقف زمان است و از ضرب تعداد کار، عملیات و ماشین در ضریب ثابتی بدست می‌آید. مقدار این ضریب ثابت در بخش تنظیم پارامتر الگوریتم مورد بررسی قرار گرفته است. جزئیات الگوریتم ژنتیک طراحی شده به صورت زیر هستند.

۳-۱. نحوه نمایش جواب‌ها

در مسأله زمان‌بندی در محیط کارگاه باز با ماشین‌های انعطاف‌پذیر توالی عملیات‌ها و تخصیص ماشین، دو تصمیمی است که باید

$$\text{Min } Z = \frac{\sum_{j=1}^n C_j}{n} \quad (۱)$$

s.t.

$$Y_{jhk} \leq \alpha_{jhk} \quad \forall j, h, k \quad (۲)$$

$$\sum_{k=1}^m Y_{jhk} = 1 \quad \forall j, h, k \quad (۳)$$

$$st_{iek} - st_{jhk} + M(1 - Z_{jhiek}) + M(1 - Y_{jnk}) + M(1 - Y_{iek}) \geq Y_{jnk} \beta_{jnk} \quad \forall i, j, h, e, k; i \neq j \quad (۴)$$

$$Z_{jhiek} + Z_{iejhk} + M(1 - Y_{jnk}) + M(1 - Y_{iek}) \geq 1 \quad \forall i, j, h, e, k; i \neq j \quad (۵)$$

$$Z_{jhiek} + Z_{iejhk} - M(1 - Y_{jnk}) - M(1 - Y_{iek}) \leq 1 \quad \forall i, j, h, e, k; i \neq j \quad (۶)$$

$$Z_{jhiek} - M \cdot Y_{jnk} \leq 0 \quad \forall i, j, h, e, k; i \neq j \quad (۷)$$

$$st_{jel} - (Y_{jnk} \beta_{jnk} + Tr_{kl}) + M(1 - V_{jhe}) + M(1 - Y_{jnk}) + M(1 - Y_{jel}) \geq st_{jnk} \quad \forall j, h, e, k, l; e \neq h \quad (۸)$$

$$V_{jhe} + V_{jeh} = 1 \quad \forall j, h, e; e \neq h \quad (۹)$$

$$Z_{jhjek} \leq V_{jhe} \quad \forall j, h, e, k; e \neq h \quad (۱۰)$$

$$st_{jnk} - M \cdot Y_{jnk} \leq 0 \quad \forall j, h, k \quad (۱۱)$$

$$C_j \geq st_{jnk} + Y_{jnk} \beta_{jnk} \quad \forall j, h, k \quad (۱۲)$$

$$Y_{jnk}, Z_{jhiek}, V_{jhe} \in \{0, 1\} \quad \forall i, j, h, e, k \quad (۱۳)$$

$$st_{jnk}, C_j \geq 0 \quad \forall j, h, k \quad (۱۴)$$

تابع هدف در نظر گرفته شده یعنی حداقل میانگین زمان تکمیل کارها با تساوی (۱) مشخص شده است. محدودیت (۲) نشان می‌دهد که هر عملیات می‌تواند به یک ماشین از میان ماشین‌هایی که توان پردازش آن را دارند، تخصیص یابد. محدودیت (۳) مشخص می‌کند که هر عملیات فقط روی یک ماشین می‌تواند انجام گیرد (انتخاب تنها یک ماشین برای پردازش هر عملیات). محدودیت (۴) علاوه بر محاسبه زمان شروع پردازش عملیات‌های کارهای مختلف را روی یک ماشین خاص، تضمین می‌کند که هر ماشین حداکثر یک عملیات را در هر زمان می‌تواند پردازش کند. محدودیت‌های (۵) و (۶) توالی عملیات‌های کارهای مختلف را روی یک ماشین تعیین می‌کنند. محدودیت (۷) تضمین می‌کند که یک عملیات فقط روی ماشین تخصیص یافته خود پردازش شود. محدودیت (۸) علاوه بر محاسبه زمان شروع پردازش عملیات‌های یک کار روی ماشین‌های مختلف، تضمین می‌کند که هر کار در هر زمان حداکثر

۳-۲. الگوریتم ابتکاری طراحی شده برای تولید جمعیت اولیه

برای تولید جمعیت اولیه، الگوریتم ابتکاری متناسب با مسأله مورد نظر طراحی شده است. این الگوریتم بر اساس توالی تصادفی اولیه‌ای که تولید شده است، ماشین مناسب را به هر عملیات تخصیص می‌دهد. با بکارگیری این روش ماشین با حداقل زمان پردازش لزوماً انتخاب نمی‌شود، بلکه زمان پردازش روی هر ماشین، زمان حمل و نقل بین ماشین‌ها و بیکار بودن ماشین‌ها مد نظر قرار می‌گیرد تا میانگین زمان تکمیل کل کارها حداقل گردد. الگوریتم هر عملیات را به ترتیبی که در توالی تصادفی تولید شده است، در نظر می‌گیرد و ماشین متناسب را به آن اختصاص می‌دهد و این کار ادامه می‌یابد تا کلیه عملیات‌ها به ماشین مناسبی اختصاص یابند. گام‌های الگوریتم به شرح زیر است.

گام صفر: شروع

۱. یک جایگشت تصادفی از عملیات‌ها تولید کنید.
۲. ماشین‌هایی را که توانایی پردازش اولین عملیات از جایگشت تصادفی تولید شده را دارند، مشخص کنید و این عملیات را به ماشینی که دارای کمترین زمان پردازش است تخصیص دهید. (در صورت وجود بیش از یک ماشین با کمترین زمان پردازش، یکی را بطور تصادفی انتخاب کنید).
۳. K را برابر ۲ قرار دهید.

گام یک: تخصیص ماشین

۱. ماشین‌هایی را که توان پردازش K امین عملیات از جایگشت تصادفی تولید شده را دارند (ماشین‌های واجد شرایط) انتخاب و در مجموعه‌ای بنام U قرار دهید $U = \{U_1, U_2, \dots, U_u\}; u = 1, 2, \dots, m$.
۲. a را زمان اتمام آخرین عملیات پردازش شده از کار فعلی (کار مربوط به K امین عملیات از جایگشت تصادفی) و μ را شماره ماشینی که آن را پردازش کرده قرار دهید. اگر عملیاتی از این کار تاکنون انجام نگرفته است مقدار a را صفر قرار دهید.
۳. زمان اتمام آخرین عملیات روی هر یک از ماشین‌های مجموعه U مشخص کنید و آن را S_u بنامید (که در آن، u شماره اعضای مجموعه U است).
۴. اگر $a > 0$ تمام ماشین‌های مجموعه U را چک کنید؛ اگر شماره ماشین با μ برابر نیست مقدار حمل و نقل بین این ماشین و μ را به a اضافه کنید و آن را bu بنامید. اگر شماره ماشین با μ برابر است قرار دهید $bu = a$. همچنین، اگر $a = 0$ باشد آنگاه، $bu = 0$ است.
۵. حداکثر مقدار بین bu و S_u را برای همه ماشین‌های مجموعه U بدست آورید و این مقدار را du بنامید.

گرفته شود. با تخصیص عملیات‌ها به ماشین‌ها و تعیین توالی آنها روی هر ماشین، یک جواب موجه ایجاد می‌شود. بنابراین کروموزوم طراحی شده دارای دو بخش است: بخش توالی عملیات‌ها و بخش انتخاب ماشین‌ها. برای بخش توالی عملیات کدگذاری جایگشتی به کار می‌رود [۲]. در مسأله زمانبندی در محیط کارگاه باز توالی که در آن عملیات‌ها پردازش می‌شوند بی اهمیت است بدین معنی که عملیات‌ها می‌توانند در هر توالی مورد پردازش قرار گیرند. بنابراین تمامی کدگذاری‌های جایگشتی تولید شده، موجه هستند. برای بخش انتخاب ماشین، شماره ماشین‌ها در هر ژن به کار می‌رود. شکل شماره (۱) ساختار یک کروموزوم را نشان می‌دهد.

$$\text{کروموزوم} = \left[\begin{array}{|c|c|} \hline \text{انتخاب ماشین‌ها} & \text{توالی عملیات‌ها} \\ \hline \end{array} \right]$$

شکل ۱. ساختار کروموزوم

مسأله‌ای با دو کار، سه عملیات و دو ماشین را در نظر بگیرید که در آن هر دو ماشین قابلیت پردازش هر یک از سه عملیات را دارند. همان‌طور که در شکل شماره (۲) نشان داده شده است به هر یک از عملیات‌های کار اول و دوم یک عدد تخصیص می‌یابد تا نماینده آن عملیات در جایگشت تولیدی باشد. همچنین، O_{ih} بیانگر عملیات h -ام کار i است.

یک زمانبندی شدنی، می‌تواند به صورت $O_{11}, O_{12}, O_{21}, O_{22}, O_{31}, O_{32}, O_{33}$ باشد. که در آن O_{33} اولین عملیاتی است که باید با ماشین مناسب انتخاب شده پردازش شود، O_{12} دومین عملیات و O_{11} آخرین عملیات است. بنابراین توالی این جواب شدنی در بخش توالی عملیات به صورت $1 > 3 > 4 > 2 > 5 > 6$ خواهد بود. یک جواب شدنی با خواندن جایگشت از چپ به راست همیشه ایجاد خواهد شد. برای بخش انتخاب ماشین، شماره ماشین‌ها در هر ژن به کار می‌رود. همان‌طور که در شکل شماره (۳) نشان داده شده است، ماشینی که هر عملیات را پردازش می‌کند در ژن متناظر با آن آمده است. مثلاً عملیات شماره ۵ (O_{33}) باید با ماشین شماره ۱ پردازش شود.

$$\left[\begin{array}{|c|c|c|c|c|c|} \hline O_{33} & O_{22} & O_{21} & O_{13} & O_{12} & O_{11} \\ \hline \end{array} \right]$$

$$\left[\begin{array}{|c|c|} \hline \text{کار ۱} & \text{کار ۲} \\ \hline \end{array} \right]$$

$$\left[\begin{array}{|c|c|c|c|c|c|} \hline ۱ & ۲ & ۳ & ۴ & ۵ & ۶ \\ \hline \end{array} \right]$$

شکل ۲. عملیات‌های هر کار و عدد جایگشت آن

$$\left[\begin{array}{|c|c|c|c|c|c|} \hline \text{توالی عملیات} & ۶ & ۲ & ۵ & ۴ & ۳ & ۱ \\ \hline \text{انتخاب ماشین} & ۱ & ۲ & ۱ & ۱ & ۱ & ۲ \\ \hline \end{array} \right]$$

شکل ۳. بخش انتخاب ماشین

۶. زمان انجام هر عملیات روی ماشین‌های مجموعه U را به مقدار du متناظر با آن بیافزایید مقادیر حاصل را eu بنامید. در این مرحله از بین مقادیر موجود eu حداقل مقدار را انتخاب کنید. ماشینی که حداقل مقدار را داشته است برای پردازش عملیات K ام انتخاب می‌شود.

گام دو: کنترل شرط توقف

۱. در صورت تخصیص همه عملیات‌ها ($K = n, p$) به گام ۳ برو. در غیر این صورت، قرار دهید $K = K + 1$ و $U = \emptyset$ و به گام ۱ بروید.

گام سه: ذخیره جواب و توقف

جهت درک بهتر موضوع مثال مربوط به جدول شماره (۱) را که شامل دو کار، سه عملیات و دو ماشین است با زمان‌های حمل و نقل بشرح زیر در نظر بگیرید:

$$Tr_{kl} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

گام‌های الگوریتم به صورت زیر خواهد بود:

گام صفر

۱. فرض کنید جایگشت تصادفی ۱، ۳، ۴، ۵، ۲، ۶، تولید شده است.

۲. اولین عملیات توالی، عملیات ۶ یا همان $O_{۳۳}$ است که ماشین‌های ۱ و ۲ بترتیب با زمان‌های ۷ و ۹ توان پردازش آن را دارند. بنابراین، ماشین ۱ با زمان پردازش کمتر انتخاب می‌شود.

$$K = 2$$

گام یک

۱. عملیات بعدی، عملیات ۲ ($O_{۱۲}$) است که تنها ماشین ۲ قابلیت پردازش آن را دارد. بنابراین، $U = \{2\}$.

۲. عملیات ۲ اولین عملیاتی از کار ۱ است که پردازش می‌شود. بنابراین، $a = 0$.

۳. قبلاً عملیاتی توسط ماشین ۲ پردازش نشده است. بنابراین، $S_1 = 0$.

۴. از آنجایی که $a = 0$ است داریم: $b_1 = 0$.

۵. $d_1 = \max\{0, 0\} = 0$ بنابراین داریم: $d_u = \max\{S_u, b_u\}$

$$e_u = d_u + \beta_{jhu} \Rightarrow e_1 = 0 + 4 = 4$$

۷. بنابراین، ماشین ۲ (که البته در اینجا تنها گزینه است) برای پردازش عملیات ۲ انتخاب می‌شود.

گام دو

۱. قرار دهید $K = 2 + 1 = 3$ و $U = \emptyset$ و به گام ۱ می‌رویم.

گام یک

۱. عملیات بعدی، عملیات ۵ ($O_{۲۲}$) است که تنها ماشین ۱ قابلیت پردازش آن را دارد. بنابراین، $U = \{1\}$.

۲. آخرین عملیات کار فعلی (کار ۲)، عملیات ۶ بوده که در زمان توسط ماشین ۱ تکمیل شده است. بنابراین، $a = 7$ و $\mu = 1$.

۳. ماشین ۱ قبلاً عملیات ۶ را پردازش کرده است. بنابراین، $S_1 = 7$.

۴. از آنجایی که $a > 0$ و $\mu = 1$ است داریم: $b_1 = a = 7$.

$$d_1 = \max\{7, 7\} = 7$$

$$e_1 = 7 + 5 = 12$$

۷. بنابراین، ماشین ۱ (که البته در اینجا تنها گزینه است) برای پردازش عملیات ۵ انتخاب می‌شود.

گام دو

۱. قرار دهید $K = 3 + 1 = 4$ و $U = \emptyset$ و به گام ۱ می‌رویم.

گام یک

۱. عملیات بعدی، عملیات ۴ ($O_{۲۱}$) است که تنها ماشین ۱ قابلیت پردازش آن را دارد. بنابراین، $U = \{1\}$.

۲. آخرین عملیات کار فعلی (کار ۲)، عملیات ۵ بوده که در زمان توسط ماشین ۱ تکمیل شده است. بنابراین، $a = 12$ و $\mu = 1$.

۳. ماشین ۱ قبلاً عملیات ۵ را پردازش کرده است. بنابراین، $S_1 = 12$.

۴. از آنجایی که $a > 0$ و $\mu = 1$ است داریم: $b_1 = a = 12$.

$$d_1 = \max\{12, 12\} = 12$$

$$e_1 = 12 + 10 = 22$$

۷. بنابراین، ماشین ۱ (که البته در اینجا تنها گزینه است) برای پردازش عملیات ۴ انتخاب می‌شود.

گام دو

۱. قرار دهید $K = 4 + 1 = 5$ و $U = \emptyset$ و به گام ۱ می‌رویم.

گام یک

۱. عملیات بعدی، عملیات ۳ ($O_{۱۳}$) است که ماشین‌های ۱ و ۲ قابلیت پردازش آن را دارد. بنابراین، $U = \{1, 2\}$.

۲. آخرین عملیات کار فعلی (کار ۱)، عملیات ۲ بوده که در زمان توسط ماشین ۲ تکمیل شده است. بنابراین، $a = 4$ و $\mu = 2$.

۳. ماشین ۱ قبلاً عملیات ۴ و ماشین ۲ قبلاً عملیات ۲ را پردازش کرده است. بنابراین، $S_1 = 22$ و $S_2 = 4$.

۴. از آنجایی که $a > 0$ است داریم:

$$b_1 = a + Tr_{r1} \Rightarrow b_u = 4 + 1 = 5; \mu = 1$$

$$b_2 = a = 4; \mu = 2$$

$$d_1 = \max\{22, 5\} = 22 \text{ و } d_2 = \max\{4, 4\} = 4$$

کروموزوم‌ها اعمال شده است. عملگری که دارای بهترین عملکرد است، در بخش تنظیم اپراتورها مورد بررسی قرار گرفته است. این اپراتورها به شرح زیر هستند.

تقاطع SBOX: این عملگر در محیط جریان کارگاهی در مقایسه با بسیاری از اپراتورهای تقاطع دارای عملکرد بالایی است. این عملگر تقاطع دو جایگشت را به این صورت با هم ترکیب می‌کند که، ابتدا هر دو والد موقعیت به موقعیت چک می‌شوند و عملیات-هایی که در هر دو والد دار ای موقعیت مشابهی هستند به همان موقعیت‌ها در فرزندان کپی می‌شوند. سپس هر فرزندی همه عملیات‌های والد خود را تا یک نقطه برش به طور مستقیم دریافت می‌کند. عملیات‌های خالی هر فرزند با عملیات‌های والد دیگر پر می‌شوند البته عملیات‌هایی که در این فرزند نباشد [۲۳]. شکل شماره (۴) مثالی از این اپراتور را نشان می‌دهد.

تقاطع UPC: این اپراتور با اعداد تصادفی کار می‌کند و به صورت زیر عمل می‌کند که ابتدا برای هر والد تعداد $n.p$ عدد تصادفی در بازه صفر و یک تولید می‌شود. سپس هر عدد تصادفی به یک عملیات تخصیص می‌یابد. برای هر عملیات یک عدد تصادفی تولید می‌شود، اگر این مقدار از ۰.۷ کمتر بود، عدد تصادفی از والد ۱ به فرزند ۱ کپی می‌شود. در غیر این صورت عدد تصادفی از والد ۲ به فرزند ۱ کپی می‌شود. نهایتاً، عملیات‌ها به صورت صعودی بر اساس اعداد تصادفی مرتب می‌شوند. روند مشابهی برای تولید فرزند دوم نیز بکار گرفته می‌شود [۲۴]. شکل شماره (۵) روند کار را در مثالی نشان می‌دهد.

تقاطع MTPC: این اپراتور از ادبیات کارگاه باز گرفته شده است و روند کار آن بدین صورت است که دو نقطه تصادفی k_1 و k_2 برای برش انتخاب می‌شوند بطوریکه $L < k_1 < k_2 < L$ (طول بخش توالی عملیات‌ها است). سپس عملیات‌های والد ۱ به بردار ۱ بر اساس عملیات‌های بین دو نقطه برش از والد ۲ کپی می‌شود. سایر موقعیت‌ها خالی هستند. در ادامه موقعیت‌های پر بردار ۱ به فرزند ۱ کپی می‌شود، بدین صورت که برای کپی کردن از موقعیت $k_2 + 1$ ام فرزند ۱ شروع می‌کنیم و تا موقعیت k_1 پیش می‌رویم، عملیات‌ها با توالی موجود در بردار ۱ کپی می‌شوند. موقعیت‌های خالی فرزند ۱ را با عملیات‌های والد ۲ با توجه به عملیات‌های بین دو نقطه برش پر می‌کنیم. روند مشابهی برای تولید فرزند ۲ بکار می‌رود [۲۵]. شکل شماره (۶) این روند را با مثالی توضیح می‌دهد که در آن $k_1 = 2$ و $k_2 = 4$.

تقاطع دو نقطه‌ای: گام‌های این روش به صورت زیر است:
گام ۱: دو نقطه تصادفی k_1 و k_2 را برای برش انتخاب کنید بطوریکه $L < k_1 < k_2 < L$ (طول بخش توالی عملیات‌هاست).
گام ۲: رشته بین این دو نقطه برش از والد ۱ را به فرزند ۲ کپی کنید. سپس این عملیات‌های کپی شده را در والد ۲ پیدا کرده و حذف می‌کنیم.

$$e_1 = 22 + 8 = 30 \text{ و } e_2 = 4 + 27 = 31 \quad ۸.$$

۹. بنابراین، ماشین ۱ برای پردازش عملیات ۳ انتخاب می‌شود.

گام دو

۱. قرار دهید $U = \emptyset$ و $K = 5 + 1 = 6$ و به گام ۱ می‌رویم.

گام یک

۱. عملیات بعدی، عملیات ۱ (O_{11}) است که ماشین‌های ۱ و ۲ قابلیت پردازش آن را دارد. بنابراین، $U = \{2, 1\}$.
۲. آخرین عملیات کار فعلی (کار ۱)، عملیات ۳ بوده که در زمان ۳۰ توسط ماشین ۱ تکمیل شده است. بنابراین، $a = 30$ و $\mu = 1$.
۳. ماشین ۱ قبلاً عملیات ۳ و ماشین ۲ قبلاً عملیات ۲ را پردازش کرده است. بنابراین، $S_1 = 30$ و $S_2 = 4$.
۴. از آنجایی که $a > 0$ است داریم:

$$1\mu = b_1 = a = 30$$

$$2\mu = b_2 = a + Tr_{1,2} \Rightarrow b_2 = 30 + 1 = 31$$

۵.

$$d_2 = \max\{4, 31\} = 31$$

$$d_1 = \max\{30, 30\} = 30$$

۶.

$$e_1 = 30 + 9 = 39 \text{ و } e_2 = 31 + 6 = 37$$

۷. بنابراین، ماشین ۲ برای پردازش عملیات ۱ انتخاب می‌شود.

گام دو

۱. $K = n.p = 2 \times 3 = 6$ بنابراین همه عملیات‌ها

تخصیص یافته‌اند. در نتیجه، به گام ۳ می‌رویم.

گام سه: جواب را که در شکل شماره (۳) نیز نشان داده شده است ذخیره کرده و متوقف می‌شویم.

۳-۳. تابع برازندگی، مکانیزم انتخاب و نخبه‌گرایی

تابع هدف در این الگوریتم میانگین زمان تکمیل کارها و تابع برازندگی "میانگین زمان تکمیل کارها/۱" در نظر گرفته شده است. چرخ رولت به عنوان مکانیزم انتخاب کروموزوم‌های شایسته‌تر استفاده شده است و برای استراتژی نخبه‌گرایی درصد ثابتی از بهترین کروموزوم‌ها مستقیماً به نسل بعد منتقل می‌شوند.

۳-۴. اپراتور تقاطع

عملگر تقاطع، نقش مهمی در تولید فرزندان بهتر بازی می‌کند. این عملگر دو والد از حوضچه تولید مثل را در نظر گرفته و با یک احتمال از قبل تعیین شده P_c ، بر روی کروموزوم‌های والد عمل می‌کند و بر اساس آنها دو فرزند جدید تولید می‌نماید. این عملگر با این هدف انجام می‌شود که کروموزوم‌های جدید در بردارنده بخش‌های خوب کروموزوم‌های قبلی خواهند بود و شاید این کروموزوم‌های جدید عملکرد بهتری داشته باشند. در این پژوهش چهار تا از بهترین اپراتورهای تقاطع در ادبیات زمانبندی روی

محلی جلوگیری می‌نماید. اپراتور جهش روی هر دو بخش کروموزوم طراحی شده اعمال می‌شود. برای بخش توالی عملیات چهار روش مختلف به کار رفته است. عملگری که دارای بهترین عملکرد است، در بخش تنظیم اپراتورها مورد بررسی قرار گرفته است. کروموزوم شکل شماره (۸) را در نظر بگیرید، هر روش با مثالی برای درک بهتر موضوع توضیح داده خواهد شد.

گام ۳: موقعیت‌های خالی فرزند ۲ با عملیات‌های والد ۲ پر می‌شود. بدین صورت همیشه جواب شدنی تولید می‌شود [۷]. شکل شماره (۷) مثالی از این اپراتور است. در این مثال دو نقطه ۳ و ۵ به عنوان نقاط برش استفاده شده‌اند.

۳-۵. اپراتور جهشی

بعد از تقاطع، کروموزوم‌ها تحت اپراتور جهش قرار می‌گیرند. عملگر جهش با حفظ گوناگونی جمعیت از افتادن الگوریتم در بهینه

والد ۱	۶	۲	۵	۴	۳	۱	۱	۲	۱	۱	۱	۲
والد ۲	۴	۶	۵	۱	۳	۲	۲	۲	۱	۲	۱	۱
فرزند ۱	-	-	۵	-	۳	-	-	-	۱	-	۱	-
فرزند ۲	-	-	۵	-	۳	-	-	-	۱	-	۱	-

↓

فرزند ۱	۶	۲	۵	-	۳	-	۱	۲	۱	-	۱	-
فرزند ۲	۴	۶	۵	-	۳	-	۲	۲	۱	-	۱	-
فرزند ۱	۶	۲	۵	۴	۳	۱	۱	۲	۱	۲	۱	۲
فرزند ۲	۴	۶	۵	۲	۳	۱	۲	۲	۱	۲	۱	۲

شکل ۴. مثالی از SBOX

والد ۱	۶	۲	۵	۴	۳	۱
	۱	۲	۱	۱	۱	۲
والد ۲	۴	۶	۵	۱	۳	۲
	۲	۲	۱	۲	۱	۱
عملیات‌ها	O_{11}	O_{12}	O_{21}	O_{22}	O_{31}	O_{32}
اعداد تصادفی برای والد ۱	.۵۵	.۶۹	.۶۹	.۴۳	.۱۶	.۶۹
اعداد تصادفی برای والد ۲	.۶۶	.۲۵	.۲۱	.۸۷	.۹۸	.۶۳
عملیات‌ها	O_{11}	O_{12}	O_{21}	O_{22}	O_{31}	O_{32}
اعداد تصادفی برای هر عملیات	.۰۴	.۶۸	.۹۴	.۰۱	.۰۵	.۸۷
عملیات‌ها	O_{11}	O_{12}	O_{21}	O_{22}	O_{31}	O_{32}
اعداد تصادفی برای فرزند ۱	.۵۵	.۶۹	.۲۱	.۴۳	.۱۶	.۶۳

فرزند ۱	O_{11}	O_{12}	O_{21}	O_{22}	O_{31}	O_{32}
مرتب سازی	.۱۶	.۲۱	.۴۳	.۵۵	.۶۳	.۶۹

فرزند ۱	۵	۳	۴	۱	۶	۲
	۱	۱	۱	۲	۲	۲

شکل ۵. مثالی از UPC

جهش پرش بزرگ^{۱۳}: موقعیت دو عملیات تصادفی انتخاب شده تغییر می‌کند. دو عدد تصادفی انتخابی نشان دهنده دو عملیات‌اند که می‌خواهیم موقعیت‌شان را تغییر دهیم [۷]. شکل شماره (۱۲) این حالت را نشان می‌دهد در حالتی که دو عملیات ۳ و ۶ انتخاب شده‌اند. برای اعمال اپراتور جهش روی بخش انتخاب ماشین، یک کروموزوم با احتمال P_m به صورت تصادفی انتخاب می‌شود. سپس برای هر ژن در بخش انتخاب ماشین (هر عملیات)، یک عدد تصادفی $r \in [0,1]$ تولید می‌شود. اگر r کوچکتر از ۵ باشد، ماشین‌هایی که توان پردازش عملیات فوق را دارند معرفی می‌شود و یکی از ماشین‌های معرفی شده برای پردازش این عملیات به طور تصادفی انتخاب می‌شود. شکل شماره (۱۳) مثالی از این حالت را نشان می‌دهد که در آن اعداد تصادفی تولید شده برای ژن‌های ۱ و ۵ از ۵ کمتر بوده است.

جهش شیفت^{۱۴}: یک عملیات بطور تصادفی انتخاب شده و بطور تصادفی در یک موقعیت در توالی قرار داده می‌شود. همچنین، تغییری در مابقی کروموزوم داده نمی‌شود [۲۶]. اگر فرض کنیم عملیات اول و موقعیت چهارم بطور تصادفی انتخاب شده باشند، شکل شماره (۹) مثالی از این اپراتور را نشان می‌دهد. جهش معاوضه^{۱۱}: دو موقعیت تصادفی در کروموزوم انتخاب شده و جایگاه آن دو عملیات معاوضه می‌شود [۲۶]. برای این عمل دو عدد به صورت تصادفی تولید می‌شود که نشان‌دهنده دو موقعیت است. اگر این دو موقعیت تصادفی ۲ و ۴ باشند، شکل شماره (۱۰) این حالت را نشان می‌دهد. جهش معکوس^{۱۲}: دو عدد تصادفی به عنوان نقاط برش تولید می‌شود و عملیات‌های ما بین این دو نقطه معکوس می‌شوند [۲۶]. مثالی از این حالت در شکل شماره (۱۱) نشان داده شده است در این مثال اعداد تصادفی ۳ و ۶ هستند.

والد ۱	۶	۲	۵	۴	۳	۱	۱	۲	۱	۱	۱	۲
والد ۲	۴	۶	۵	۱	۳	۲	۲	۲	۱	۲	۱	۱
بردار ۱	-	۲	-	۴	۳	-	-	۲	-	۱	۱	-
فرزند ۱	۳	-	-	-	۲	۴	۱	-	-	-	۲	۱
فرزند ۱	۳	۶	۵	۱	۲	۴	۱	۲	۱	۲	۲	۱

شکل ۶. مثالی از MTPC

والد ۱	۶	۲	۵	۴	۳	۱	۱	۲	۱	۱	۱	۲
والد ۲	۴	۶	۵	۱	۳	۲	۲	۲	۱	۲	۱	۱
فرزند ۱	۶	۲	۵	۱	۳	۴	۱	۲	۱	۲	۱	۱
فرزند ۲	۶	۱	۵	۴	۳	۲	۲	۱	۱	۱	۱	۱

شکل ۷. مثالی از تقاطع دو نقطه‌ای

۶	۲	۵	۴	۳	۱	۱	۲	۱	۱	۱	۲
---	---	---	---	---	---	---	---	---	---	---	---

شکل ۸. کروموزوم

۶	۲	۵	۱	۴	۳	۱	۲	۱	۲	۱	۱
---	---	---	---	---	---	---	---	---	---	---	---

شکل ۹. اپراتور جهش شیفت

۶	۴	۵	۲	۳	۱	۱	۱	۱	۱	۱	۲
---	---	---	---	---	---	---	---	---	---	---	---

شکل ۱۰. اپراتور جهش معاوضه

۶	۲	۵	۱	۳	۴	۱	۲	۱	۲	۱	۱
---	---	---	---	---	---	---	---	---	---	---	---

شکل ۱۱. اپراتور جهش معکوس کردن

۳	۲	۵	۴	۶	۱	۱	۲	۱	۱	۱	۲
---	---	---	---	---	---	---	---	---	---	---	---

شکل ۱۲. اپراتور جهش پرش بزرگ

۶	۲	۵	۴	۳	۱	۲	۲	۱	۲	۱	۲
---	---	---	---	---	---	---	---	---	---	---	---

شکل ۱۳. مثالی برای اعمال اپراتور جهشی روی بخش ماشین

۴. کالیبره کردن الگوریتم ژنتیک

پس از طراحی یک الگوریتم فراابتکاری برای مسأله‌ای خاص می‌بایست حساسیت الگوریتم نسبت به پارامترهای ورودی آن مسأله (مانند مقادیر پارامترها و یا ابعاد مسأله) تحلیل گردد. این تحلیل با بررسی برخی فاکتورهای تاثیرگذار و حساس الگوریتم صورت می‌گیرد که نتیجه آن یافتن وضعیتی نسبتاً پایدار برای الگوریتم در مواجهه با مسأله‌های متنوع از آن دست است.

برای یک الگوریتم طراحی شده مشخصاتی تعریف می‌شود که با تنظیم آنها می‌توانیم عملکرد الگوریتم را بهبود دهیم مثلاً تعداد جمعیتی که در هر نسل تولید می‌شود، احتمالی که کروموزوم‌ها تحت عمل تقاطع یا جهش قرار می‌گیرند، زمانی که الگوریتم برای شرط توقف سپری می‌کند و تعیین بهترین نوع از هر اپراتور و غیره. ابزاری که برای این منظور بکار می‌رود طراحی آزمایشات است. هر الگوریتم دارای ورودی‌هایی است که بر عملکرد آن تاثیر می‌گذارد و یک خروجی به عنوان متغیر پاسخ تولید می‌کند. در طراحی آزمایشات می‌خواهیم تاثیر عوامل به گونه‌ای باشد که خروجی مطلوبی تامین کند این خروجی در مسأله‌ای که در این پژوهش تعریف شده کمینه سازی میانگین زمان تکمیل کل کارهاست. این کار در واقع بهبود کیفیت الگوریتم طراحی شده به منظور دستیابی به بهترین مقدار تابع هدف است.

در طراحی آزمایشات میزان تاثیر هر یک از عوامل ورودی محاسبه می‌شود و بدین طریق عواملی که تاثیر چندانی نداشته‌اند حذف می‌شوند و با عوامل تاثیر گذار آنقدر آزمایشات تکرار می‌شود تا به

بهترین جواب برسیم. طراحی آزمایشات روش‌های مختلفی دارد که صورت زیر می‌باشد:

- روش‌های آزمایشات سنتی^{۱۴}
- آزمایشات کسری^{۱۵}
- روش سطح پاسخ^{۱۶}

در روش آزمایشات سنتی ابتدا پارامترهای موثر بر مسأله را حدس زده و سپس سطوح آنها را به صورت گسسته تعیین می‌کنیم. این سطوح به صورت سعی و خطا مشخص می‌شوند که البته کار دشواری نیست. سختی کار تعیین بهترین ترکیب از بین ترکیب‌های موجود بعد از تعیین فاکتورها و سطوح آنها است. مثلاً اگر فرض کنیم الگوریتم مسأله‌ای دارای ۱۰ فاکتور موثر است که هر یک دارای ۲ سطح باشند، آنگاه ۲^{۱۰} سناریو داریم که باید آزمایش شوند. در روش آزمایشات سنتی کلیه حالات ممکن که برای یک مسأله وجود دارد مورد آزمایش قرار می‌گیرد، طبیعی است زمانی که تعداد حالات زیاد شود یعنی تعداد فاکتورهای موثر و سطوح آنها بالا باشد، این روش سخت و وقت‌گیر خواهد بود و استفاده از آن منطقی به نظر نمی‌رسد. در اینگونه از مواقع که تعداد حالات زیاد می‌شود به جای روش آزمایشات سنتی از آزمایشات کسری استفاده می‌کنیم. بدین ترتیب، به جای انجام ۲^{۱۰} سناریو تعداد کمتری از سناریوها اجرای می‌شوند. یکی از ضعف‌های اصلی این روش آنست که ممکن است هر آزمایشگر جهت انجام یک آزمایش با هدف مشخص طرح متفاوتی ارائه دهد که این امر منجر به اختلاف در نتایج خواهد شد. بدین منظور، ناگوچی طراحی‌های ساده و استاندارد برای روش آزمایشات کسری

و روی کامپیوتری با پردازنده ۲.۵۳ گیگا هرتز و ۴ گیگا بایت حافظه اصلی کد شده است. زمان پردازش‌ها به صورت تصادفی یکنواخت در بازه (۱،۹۹) و زمان‌های حمل‌ونقل بین ماشین‌ها نیز به صورت یکنواخت در بازه‌های (۰،۲۰)، (۰،۵۰) تولید شده است.

۵. محاسبه آماره S/N ^{۱۸}: برای تعیین سطوح بهینه تاگوچی از نسبت S/N استفاده می‌کنیم. در روش تاگوچی برای تنظیم پایداری الگوریتم از این نسبت استفاده می‌شود که در واقع نسبت علامت به خطا را تعیین می‌کند و به صورت معادله (۱۶) محاسبه می‌شود.

$$S/N \text{ ratio} = -10 \log_{10}(\text{objective function})^2 \quad (16)$$

با محاسبه این نسبت نموداری با نام "نمودار اثرات عوامل" رسم می‌شود. این نمودار، در واقع نمودار فاکتورها بر حسب این نسبت را نشان می‌دهد. برای مقایسه روش‌ها از RPD ^{۱۹} به عنوان یک معیار استفاده می‌کنیم که به صورت معادله (۱۷) محاسبه می‌شود.

$$RPD = \frac{Alg_{sol} - LB}{LB} \quad (17)$$

در این معادله Alg_{sol} میانگین زمان تکمیل کلیه کارها است که برای هر الگوریتم و مثال بدست می‌آید. LB کران پایینی است که به صورت معادله (۱۸) محاسبه می‌شود.

در این معادله β_{jhk} زمان پردازش h امین عملیات کار z روی ماشین k است، n تعداد کار و p تعداد ماشین‌هاست. این کران پایین در واقع حداکثر حجم کار روی ماشین‌ها برای عملیات‌های هر کار است که در ادبیات برای مسایل به کار رفته است. از آنجایی که برای مسایل این پژوهش در فضای کارگاه باز، کران پایینی موجود نبود از این کران استفاده شده است تا معیاری برای مقایسه عملکرد الگوریتم به کار رفته برای حل مسأله باشد.

$$LB \rightarrow \frac{\sum_{j=1}^n \sum_{h=1}^p \min(\beta_{jhk})}{n} \quad (18)$$

برای نتایج هر سناریو اجرا شده، مقدار میانگین زمان تکمیل کلیه کارها محاسبه و سپس RPD آن با توجه به مقدار خروجی و کران پایین محاسبه شده برای آن مسأله، بدست آمده است. و نهایتاً نرخ S/N برای RPD ، محاسبه و نمودار آن رسم شده است.

۶. نمودار اثرات عوامل: این نمودار در شکل شماره (۱۴) نشان داده شده است.

۷. انتخاب سطوح بهینه برای پارامترها: با توجه به نمودار رسم شده سطوح بهینه برای فاکتورها به شرح زیر است:

ارائه داده است. روش تاگوچی حالت خاصی از روش آزمایشات کسری است که از طرح‌های متعامد استفاده می‌کند، سناریوهایی که تاگوچی استفاده می‌کند را اصطلاحاً عمود بر هم می‌نامیم. در این حالت می‌توان نتیجه سایر سناریوها را با کمک سناریوهای موجود بدست آورد [۲۸].

روش طراحی آزمایشات تاگوچی دارای ویژگی‌های زیر است:

- فقط با استفاده از کسری از کل سناریوها کار می‌کند.
- برخلاف طراحی آزمایشات سنتی، که در آن اولویت روی تنظیم فاکتورهای موثر بر میانگین متغیر پاسخ است، تاگوچی تاثیر فاکتورها را بر واریانس متغیر پاسخ بررسی می‌کند.

در دو روش بیان شده سطوحی که برای فاکتورها در نظر گرفته می‌شود گسسته است، و این نکته اشکال این روش‌ها است. در روش سطح پاسخ، تمام فاکتورها به صورت پیوسته در نظر گرفته می‌شود و حداقل و حداکثری برای هر پارامتر مشخص می‌کند و سپس با مدل ریاضی سطح بهینه را در کران مشخص شده تعیین می‌کند.

برای تنظیم الگوریتم ژنتیک طراحی شده روش تاگوچی بکار رفته است که گام‌های آن به صورت زیر است:

۱. تعیین فاکتورهای موثر و سطوح آنها: اپراتورهای تقاطع، اپراتورهای جهشی، احتمال جهش، اندازه جمعیت، شرط توقف فاکتورهایی هستند که به عنوان عوامل موثر در نظر گرفته شده‌اند و سطوح آنها در جدول شماره (۲) آمده است.

۲. محاسبه درجه آزادی: برای تخمین متوسط یک متغیر پاسخ به یک درجه آزادی نیاز داریم که با درجه آزادی که برای هر فاکتور محاسبه می‌شود جمع می‌گردد. بنابراین درجه آزادی به صورت معادله (۱۵) است:

$$\text{درجه آزادی} = 1 + \text{تعداد فاکتورها} \times (\text{تعداد سطوح} - 1) \quad (15)$$

با توجه به جدول شماره (۲)، ۵ فاکتور ۴ سطحی داریم، بنابراین درجه آزادی برابر با ۱۶ می‌شود و آرایه متعامدی که برای این الگوریتم می‌تواند استفاده شود باید حداقل ۱۶ سطر و ۵ ستون داشته باشد. از جدول استاندارد، آرایه L_{16} انتخاب می‌شود.

۳. انتخاب تعداد مسأله‌ها برای اجرا با سناریوهای داده شده: برای اجرای تاگوچی ۵ مثال به صورت تصادفی تولید شده است. که بدین صورت است، ۵ نوع مسأله که در آنها تعداد کار و عملیات و ماشین با هم برابر است و در سایزهای ۴، ۵، ۷، ۸، ۱۰ است مثلاً $4 \times 4 \times 4$. از هر سایز ۱۰ مسأله تولید شده است بنابراین ۵۰ مسأله داریم که هر یک نیز ۵ بار حل شده است. بنابراین هر سناریو بایستی ۲۵۰ بار اجرا شود.

۴. اجرای آزمایشات و جمع‌آوری داده‌ها: برای اجرای آزمایشات، کد الگوریتم در نرم افزار متلب^{۱۷} ۲۰۰۸ نوشته شده

- اپراتور تقاطع: *SBOX*
- اپراتور جهش: *Shift*
- اندازه جمعیت: ۲۰۰
- ضریب شرط توقف: ۰.۲
- احتمال جهش: ۰.۲

تحلیل واریانس برای آزمایش انجام شده در جدول شماره (۳) آمده است. هر فاکتور دارای ۴ سطح است بنابراین درجه آزادی هر یک ۳ می‌شود و درجه آزادی کل ۱۵ خواهد بود این امر موجب می‌شود که درجه آزادی خطا صفر باشد و تحلیل واریانس را غیر ممکن سازد [۲۸].

با محاسبه مقادیر میانگین مربعات برای هر فاکتور مشاهده می‌شود که فاکتورهای B و D (اپراتور جهش و شرط توقف) مقدار کوچکی دارند و تاثیر زیادی نداشته‌اند، بنابراین از تحلیل واریانس ادغام شده استفاده می‌کنیم و مقدار درجه آزادی خطا را برابر جمع درجه آزادی این دو فاکتور قرار می‌دهیم، یعنی درجه آزادی خطا برابر ۶ و مقدار میانگین مربعات آن نیز جمع این دو مقدار می‌شود.

همان طور که در جدول مشخص است فاکتور C یعنی اندازه جمعیت با ۶۹.۳۱۳٪ اهمیت، دارای بالاترین تاثیر از میان همه فاکتورهاست. و احتمال جهش در مرتبه بعد با ۱۴.۹۱۷٪ دارای اهمیت است. علاوه بر آن احتمال جهش از نوع اپراتورهای جهش مهم‌تر بوده است.

۵. نتایج محاسباتی

برای ارزیابی الگوریتم ژنتیک ارائه شده دو دسته مثال در ابعاد کوچک و بزرگ تولید شده است. مثال‌های با ابعاد بزرگ بر اساس بنچمارک‌های تیلارد^{۲۰} هستند [۲۷]. در این تحقیق، مسایلی که دارای تعداد کار و ماشین برابرند را در نظر می‌گیریم بر این اساس ۸ مسأله در اندازه‌های ۴، ۳، ۵، ۶، ۷، ۸، ۱۰، ۹ و از هر سایز ۵ مسأله تولید شده است بنابراین ۴۰ مسأله داریم. مثال‌های با ابعاد کوچک با تعداد کار، عملیات و ماشین‌های متفاوت تولید شده‌اند. زمان پردازش عملیات‌ها در بازه (۱،۹۹) و زمان‌های حمل و نقل در بازه (۰،۲۰) و (۰،۵۰) به صورت یکنواخت است. الگوریتم ارائه شده، توسط نرم افزار متلب و بر روی کامپیوتری با پردازنده ۲.۵۳ گیگا هرتز و ۴ گیگا بایت حافظه اصلی کد شده است. نتایج در جدول شماره‌های (۲) و (۳) گزارش شده است.

جهت بررسی عملکرد الگوریتم ارائه شده، جواب‌های الگوریتم ژنتیک در ابعاد کوچک مسأله با جواب‌های بهینه این مسایل که از طریق حل مدل ریاضی بوسیله نرم افزار ILOG CPLEX 12.1 بدست آمده‌اند، مقایسه شده است و علاوه بر آن کران پایینی که در معادله (۱۸) معرفی شد برای هر مسأله محاسبه شده و اختلاف جواب‌های بهینه با این کران نیز آورده شده است.

برای مسایل با ابعاد بزرگ نیز کران پایین محاسبه شده و اختلاف آن با جواب‌های الگوریتم بدست آمده است و کارایی الگوریتم با توجه به اختلاف این کران از جواب‌های بهینه در مسایل با ابعاد کوچک تحلیل شده است.

در ادامه نتایج این محاسبات بر روی مسایل با ابعاد کوچک و بزرگ ارائه می‌شود.

۵-۱. مسائل با ابعاد کوچک

در جدول شماره (۴)، ۲۷ مسأله در ابعاد کوچک توسط الگوریتم ژنتیک طراحی شده و نیز مدل ریاضی ارائه شده، حل شده‌اند. مقدار جواب بهینه این مسایل نیز در این جدول آمده است. همان طور که مشخص است الگوریتم ژنتیک در ۱۹ مسأله به جواب بهینه ارائه شده رسیده است. بنابراین الگوریتم ژنتیک ۰.۳۷٪ از مسایل با ابعاد کوچک را به صورت بهینه حل کرده است.

کران پایین ارائه شده برای این مسایل محاسبه شده و اختلافی بین ۴.۵٪ تا ۷۴.۲٪ با میانگین ۲۷.۹٪ با جواب بهینه دارد. شکل شماره (۱۵) مقایسه کران پایین ارائه شده با جواب‌های بهینه ۲۷ مسأله جدول شماره (۴) را نشان می‌دهد.

الگوریتم ژنتیک اختلافی در حدود ۲۹.۸٪ با کران پایین ارائه شده دارد. با توجه به مقادیر ارائه شده می‌توان نتیجه گرفت که الگوریتم ژنتیک توانسته مسایل با ابعاد کوچک را با عملکرد خوبی حل کند.

۵-۲. مسائل با ابعاد بزرگ

جدول شماره (۵) مسایل با ابعاد بزرگ، بر اساس بنچمارک‌های تیلارد در ۸ سایز نشان می‌دهد. این مسایل نیز با الگوریتم ژنتیک حل شده‌اند و مقدار کران پایین برای هر یک محاسبه شده است. الگوریتم ژنتیک اختلافی در حدود ۶۹.۶۷٪ با کران پایین ارائه شده برای این مسایل دارد. این مقادیر حاکی از این است که الگوریتم ژنتیک عملکردی را که برای مسایل با ابعاد کوچک داشت در اینجا ندارد.

۶. نتیجه گیری

در دنیای واقعی در بسیاری از حالات ماشین‌ها انعطاف پذیرند، در این مقاله انعطاف‌پذیری ماشین‌ها در کارگاه باز مورد بررسی قرار گرفت. یک مدل برنامه‌ریزی ریاضی خطی مختلط ارائه گردید. مسأله‌های با ابعاد کوچک از طریق حل مدل ریاضی بوسیله نرم افزار ILOG CPLEX 12.1 به جواب بهینه رسیدند. از آنجایی که این مسأله NP-hard است، برای حل آن الگوریتم ژنتیک طراحی شد. در این الگوریتم برای تولید بخشی از جواب‌های اولیه، یک الگوریتم ابتکاری بکار گرفته شد. نتایج الگوریتم ژنتیک با کران پایین ارائه شده، مقایسه شد که حاکی از آن است که این الگوریتم برای مسایل با ابعاد کوچک بسیار خوب جواب می‌دهد، این در

مقایسه گردد. علاوه بر آن ارایه کران پایینی بهتری برای مسأله می‌تواند زمینه‌ای برای تحقیقات آتی باشد.

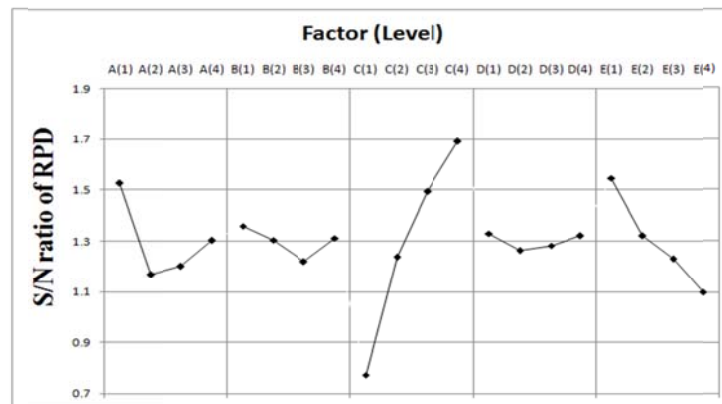
حالی است که با افزایش بعد مسأله الگوریتم جواب‌هایی تقریباً نزدیک به جواب‌های بهینه را تولید می‌کند. برای گسترش این تحقیق توصیه می‌شود که الگوریتم‌های ابتکاری و فراابتکاری دیگری برای مسأله بکار گرفته شود و با نتایج موجود

جدول ۲. فاکتورها و سطوح آنها

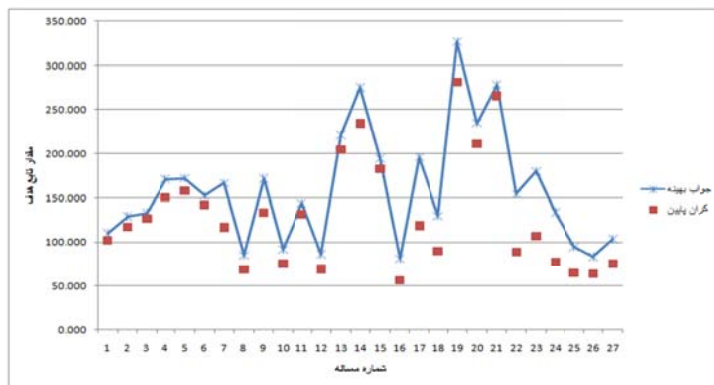
سطح ۴	سطح ۳	سطح ۲	سطح ۱	تعداد سطوح	نماد	فاکتورها
TPC	MTPC	UPC	SBOX	۴	A	اپراتور تقاطع
Giant leap	Inversion	Shift	Swap	۴	B	اپراتور جهش
۲۰۰	۱۵۰	۱۰۰	۵۰	۴	C	اندازه جمعیت
.۷	.۵	.۳۵	.۲	۴	D	ضریب شرط توقف
.۵	.۴	.۳	.۲	۴	E	احتمال جهش

جدول ۳. تحلیل واریانس برای نرخ S/N

S.V	D.F	S.S	M.S	F	Relative importance	Cumulative	P-Value
A	۳	.۳۱۷	.۱۰۶	۱۱.۸۴۷	۱۰.۷۹۴	۱۰.۷۹۴	.۰۰۶۲
C	۳	۱.۸۸۹	.۶۳	۷۰.۶۴۸	۶۹.۳۱۳	۸۰.۱۰۷	.۰۰۰۰
E	۳	.۴۲۷	.۱۴۲	۱۵.۹۸۹	۱۴.۹۱۷	۹۵.۰۲۴	.۰۰۲۹
Error	۶	.۰۵۳	.۰۰۹				
Total	۱۵	۲.۶۸۷					



شکل ۱۴. نمودار اثرات عوامل



شکل ۱۵. مقایسه جواب بهینه و کران پایین

جدول ۴. مقایسه نتایج الگوریتم ژنتیک با کران پایین در ابعاد کوچک مسأله

اندازه مسأله	شماره مسأله	جواب بهینه	مقدار کران پایین	مقدار الگوریتم ژنتیک	درصد انحراف از کران پایین	اندازه مسأله	شماره مسأله	جواب بهینه	مقدار کران پایین	مقدار الگوریتم ژنتیک	درصد انحراف از کران پایین
۲	۱	۱۱۰	۱۰۲	۱۱۰	۰.۰۷۸	۲	۱۶	۸۱.۲۵	۵۶.۲۵	۸۱.۲۵	۰.۴۴۴
۲	۲	۱۲۹	۱۱۷	۱۲۹	۰.۱۰۳	۲	۱۷	۱۹۵.۷۵	۱۱۸.۵	۱۹۵.۷۵	۰.۶۵۲
۲	۳	۱۳۳	۱۲۶.۵	۱۳۳	۰.۰۵۱	۳	۱۸	۱۲۹.۷۵	۸۹.۲۵	۱۲۹.۷۵	۰.۴۵۴
۲	۴	۱۷۱	۱۵۰	۱۷۱	۰.۱۴	۳	۱۹	۳۳۶.۵	۲۸۱.۵	۳۳۶.۵	۰.۱۶
۲	۵	۱۷۲	۱۵۸	۱۷۲	۰.۰۸۹	۳	۲۰	۲۳۴.۵	۲۱۱	۲۳۴.۵	۰.۱۱۱
۲	۶	۱۵۲.۵	۱۴۱.۵	۱۵۲.۵	۰.۰۷۸	۳	۲۱	۲۸۵.۵	۲۶۶	۲۷۸	۰.۰۴۵
۲	۷	۱۶۶.۶۶۷	۱۱۶.۶۶۷	۱۶۶.۶۶۷	۰.۴۲۹	۳	۲۲	۱۶۲.۲	۸۸.۶	۱۵۴	۰.۷۳۸
۲	۸	۸۴.۶۶۷	۶۸.۳۳۳	۸۴.۶۶۷	۰.۲۳۹	۳	۲۳	۱۸۷.۲	۱۰۶.۶	۱۸۰	۰.۶۸۹
۲	۹	۱۷۱.۶۶۷	۱۳۳	۱۷۱.۶۶۷	۰.۲۹۱	۳	۲۴	۱۳۹.۸	۷۶.۸	۱۳۳.۸	۰.۷۴۲
۲	۱۰	۹۷.۶۶۷	۷۵.۳۳۳	۹۷.۶۶۷	۰.۲۱۷	۳	۲۵	۹۶.۶	۶۴.۶	۹۴.۲	۰.۴۵۸
۲	۱۱	۱۴۳.۶۶۷	۱۳۱.۳۳۳	۱۴۳.۶۶۷	۰.۰۹۴	۳	۲۶	۸۹.۲	۶۳.۸	۸۳.۶	۰.۳۱
۲	۱۲	۸۶.۳۳۳	۶۹	۸۶.۳۳۳	۰.۲۵۱	۳	۲۷	۱۰۵.۲	۷۵	۱۰۳.۴	۰.۳۷۹
۲	۱۳	۲۲۱	۲۰۴.۵	۲۲۱	۰.۰۸۱	۳	۲۸	۲۲۱	۲۰۴.۵	۲۲۱	۰.۰۸۱
۲	۱۴	۲۷۵	۲۳۴.۵	۲۷۵	۰.۱۷۳	۳	۲۹	۲۷۵	۲۳۴.۵	۲۷۵	۰.۱۷۳
۲	۱۵	۱۹۴.۵	۱۸۳	۱۹۴.۵	۰.۰۶۳	۳	۳۰	۱۹۴.۵	۱۸۳	۱۹۴.۵	۰.۰۶۳

جدول ۵. مقایسه نتایج الگوریتم ژنتیک با کران پایین در ابعاد بزرگ مسأله

اندازه مسأله	شماره مسأله	مقدار کران پایین	مقدار میانگین الگوریتم ژنتیک	درصد انحراف از کران پایین	اندازه مسأله	شماره مسأله	مقدار کران پایین	مقدار میانگین الگوریتم ژنتیک	درصد انحراف از کران پایین
۱	۱	۱۱۰	۱۲۴.۶۶۷	۰.۱۳۳	۱	۲۱	۱۰۶.۸۵۷	۱۹۱.۶۲۹	۰.۷۹۳
۲	۲	۱۱۱.۳۳۳	۱۳۷.۶۶۷	۰.۲۳۷	۲	۲۲	۱۲۵.۵۷۱	۲۱۹.۱۴۳	۰.۷۴۵
۳	۳	۱۱۵	۱۲۵.۹۳۳	۰.۰۹۵	۳	۲۳	۱۰۳	۲۰۰.۵۷۱	۰.۹۴۷
۴	۴	۶۴	۷۰.۲	۰.۰۹۷	۴	۲۴	۱۱۵.۸۵۷	۲۰۹.۳۴۳	۰.۸۰۷
۵	۵	۱۰۷	۱۲۲.۸	۰.۱۴۸	۵	۲۵	۱۲۰.۸۵۷	۲۱۶.۶۵۷	۰.۷۹۳
۶	۶	۸۹.۵	۱۲۷.۸۵	۰.۴۲۸	۶	۲۶	۱۱۸	۲۱۹.۲۷۵	۰.۸۵۸
۷	۷	۵۷.۷۵	۷۹.۷	۰.۳۸	۷	۲۷	۱۲۲.۳۷۵	۲۳۷.۵۵	۰.۹۴۱
۸	۸	۱۰۲.۷۵	۱۳۴.۴	۰.۳۰۸	۸	۲۸	۱۲۸.۳۷۵	۲۴۵.۲۲۵	۰.۹۱۰
۹	۹	۱۳۸	۱۷۲.۹	۰.۲۵۳	۹	۲۹	۱۱۱.۶۲۵	۲۲۸.۳۵	۱.۰۴۶
۱۰	۱۰	۱۳۸.۲۵	۱۸۵.۱۵	۰.۳۳۹	۱۰	۳۰	۱۲۵.۸۷۵	۲۴۲.۷	۰.۹۲۸
۱۱	۱۱	۱۳۰.۶	۱۸۵.۷۲	۰.۴۲۲	۱۱	۳۱	۱۵۵.۸۸۹	۲۹۳.۱۵۶	۰.۸۸۱

.۹۰۲	۲۳۸.۱۱۱	۱۲۵.۲۲۲	۳۲		.۴۴۲	۱۴۳.۳۶	۹۹.۴	۱۲
۱.۰۵۱	۲۱۵.۱۷۸	۱۰۴.۸۸۹	۳۳		.۵۴۳	۱۷۸.۳۶	۱۱۵.۶	۱۳
۱.۱۰۲	۲۵۶.۴۸۹	۱۲۲	۳۴		.۴۷۰	۱۳۳.۴۴	۹۰.۸	۱۴
۱.۰۹۲	۲۴۴.۳۵۶	۱۱۶.۷۷۸	۳۵		.۴۶۲	۱۹۴.۴۴	۱۳۳	۱۵
۱.۱۵۳	۲۹۴.۵۶	۱۳۶.۸	۳۶		.۶۲۹	۱۶۵.۶۳۳	۱۰۱.۶۶۷	۱۶
۱.۰۲۱	۲۶۷.۰۲	۱۳۲.۱	۳۷	۱	.۶۰۹	۲۱۲.۶۶۷	۱۳۲.۱۶۷	۱۷
۱.۰۹۵	۲۸۰.۱۶	۱۳۳.۷	۳۸	۱	.۸۲۹	۱۸۷.۸	۱۰۲.۶۶۷	۱۸
۱.۱۷۶	۲۸۹.۹۶	۱۳۱.۹	۳۹	۱	.۸۵	۱۸۷.۱۳۳	۱۰۱.۱۶۷	۱۹
۱.۱	۲۷۱.۰۶	۱۲۹.۱	۴۰		.۸۵۲	۱۶۰.۲	۸۶.۵	۲۰

- [5] Sevastianov SV, Woeginger GJ. Linear time approximation scheme for the multiprocessor open shop problem, *Discrete Applied mathematics*, 2001, Vol. 114, pp. 273-288.
- [6] Kononov A, Sviridenco M. A linear time approximation scheme for makespan minimization in an open shop with release date, *Operations Research Letters*, 2002, Vol. 30, pp. 276-280.
- [7] Matta M. A genetic algorithm for the proportionate multiprocessor open shop, *Computers & Operations Research*, 2009, Vol. 36, pp. 2601-2618.
- [8] Matta M, Elmaghraby SE. Polynomial time algorithms for two special classes of the proportionate Multiprocessor open shop, *European Journal of Operational Research*, 2010, Vol. 201, pp. 720-728.
- [9] Kis T, Werra D, Kubiak W. A projective algorithm for preemptive open shop scheduling with two multiprocessor groups, *Operations Research Letters*, 2010, Vol. 38, pp.129-132.
- [10] Strusevich VA. A heuristic for the two-machine open-shop scheduling problem with transportation times, *Discrete Applied Mathematics*, 1999, Vol. 93, pp. 287-304.
- [11] Allaoui H, Artiba A. Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints, *Computers & Industrial Engineering*, 2004, Vol. 47, pp. 431-450.
- [12] Brucker P, Sotskov Yu N, Werner F. Complexity of shop-scheduling problems with fixed number of jobs: a survey, *Math. Meth. Oper. Res*, 2007, Vol. 65, pp. 461-481.
- [13] Naderia B, Fatemi Ghomia SMT, Aminnayeria M, Zandieh M. A study on open shop scheduling to minimise total tardiness, *International Journal of Production Research*, 2010, pp. 1-22.

پی‌نوشت

- Hybrid Flow Shop
- Tree Search
- Simulated Annealing Algorithm
- Multi-Agent Approach
- Roulette Wheel
- Similar Block Order Crossover
- Uniform Parameterize Crossover
- Modification of the Two-Point Crossover
- Two Point Crossover
- Shift
- Swap
- Inversion Mutation
- Giant Leap
- Full Factorial DOE
- Fractional
- Response Surface Method (RSM)
- MATLAB
- Signal to Noise Ratio
- Relative Percentage Deviation (RPD)
- Taillard's Benchmark

مراجع

- Pinedo M. *Scheduling: Theory, Algorithms, and Systems*, original edition published by Prentice Hall, Stern School of Business New York University, Third Edition, 2008, pp. 1-25.
- Naderi B, Fatemi Ghomi SMT, Aminnayeri M, Zandieh M. Scheduling open shops with parallel machines to minimize total completion time, *Journal of Computational and Applied Mathematics*, 2011, Vol. 235, pp. 1275-1287.
- Schuurman PG, Woeginger J. Approximation algorithms for the multiprocessor open shop scheduling problem, *Operations Research Letters*, 1999, Vol. 24, pp. 157-163.
- Jansen K, Sviridenco M. *Polynomial Time Approximation Schemes for the Multiprocessor Open and Flow Shop Scheduling Problem*, Springer Verlag Berlin Heidelberg, 2000, Vol. 1770, pp. 455-465.

- open shop scheduling, *Computers & Operations Research*, 2005, Vol. 32, pp. 1565-1591.
- [22] Witkowski T, Antczak P, Antczak A. Hybrid method for solving flexible open shop scheduling problem with simulated annealing algorithm and multi-agent approach, *Advanced Materials Research*, 2012, Vols. 383-390, pp. 4612-4619.
- [23] Ruiz R, Maroto C, Alcaraz J. Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics, *European Journal of Operational Research*, 2005, Vol. 165, pp. 34-54.
- [24] Kurz ME, Askin RG. Scheduling flexible flow lines with sequence-dependent setup times, *European Journal of Operational Research*, 2004, Vol. 159, pp. 66-82.
- [25] Senthilkumar P, Shahabudeen P. GA based heuristic for the open job shop scheduling problem, *International Journal of Advanced Manufacturing Technology*, 2006, Vol. 30, pp. 297-301.
- [26] Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed., Springer, Berlin, 1996.
- [27] Taillard E. Benchmarks for basic scheduling problems, *European Journal of Operations Research*, 1993, Vol. 64, pp. 278-285.
- [۲۸] رانجیت روی، ترجمه دکتر داوود مراد خانی و مهندس فرشید تقوی. آشنایی با روش‌های طراحی آزمایشات تاگوچی، انتشارات دانشگاه زنجان، ۱۳۸۶.
- [14] 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*, 2009, Vol. 25, pp. 314-322.
- [15] Seraj O, Tavakkoli-Moghaddam R. A tabu search method for a new bi-objective open shop scheduling problem by a fuzzy multi objective decision making approach, *International Journal of Engineering, Transactions A: Basics*, 2009, Vol. 22, pp. 1-14.
- [16] Andresen M, Brasel H, Morig M, Tusch J, Werner F, Willenius P. Simulated annealing and genetic algorithms for minimizing mean flow time in an open shop, *Mathematical and Computer Modeling*, 2008, Vol. 48, pp. 1279-1293.
- [17] Sha DY, Hsu CY. A new particle swarm optimization for the open shop scheduling problem, *Computers and Operational Research*, 2008, Vol. 35, pp. 3243-3261.
- [18] Liaw CF. An efficient tabu search approach for the two-machine preemptive open shop scheduling problem, *Computers & Operations Research*, 2003, Vol. 30, pp. 2081-2095.
- [19] Liaw CF. A hybrid genetic algorithm for the open shop scheduling problem, *European Journal of Operational Research*, 2000, Vol. 124, pp. 28-42.
- [20] Liaw CF. A tabu search algorithm for the open shop scheduling problem, *Computers & Operations Research*, 1999, Vol. 26, pp. 109-126.
- [21] Blum C. Beam-ACO-hybridizing ant colony optimization with beam search: an application to