مدیریت تولید و عملیات، دوره سوم، پیاپی(٤)، شماره (١)، بهار و تابستان ١٣٩١
وصول: ٨٩/١٠/١٥ پذیرش: ٩٠/٥/١٢
ص٧٨ - ٦٥

الگوریتم ژنتیک برای مساله زمان‌بندی تک ماشین با جرایم زودکرد خطی، و دیرکرد توان دوم و با در نظر گرفتن زمان بیکاری و شکست کار

محمد باقر فخرزاد*[1]، مهدی عظیم زاده[2]
١- استادیار دانشکده مهندسی، گروه مهندسی صنایع، دانشگاه یزد
٢- دانشجوی کارشناسی ارشد مهندسی صنایع، دانشگاه یزد

چکیده

در این مقاله، مسئله زمانبندی تک ماشین با هزینه‌های زودکرد خطی و دیرکرد توان دوم، با در نظر گرفتن شکست کار و بیکاری مجاز مورد بررسی قرار گرفته و یک مدل ریاضی غیرخطی جدیدی برای مسئله زمان‌بندی تک ماشین ارائه شده است. با در نظر داشتن پیچیدگی در حل، این مسئله به عنوان مسائل NP-hard تلقی می‌گردد. بنابراین استفاده از روش‌هایی که نتایج بهینه تولید می‌کنند، تنها برای مسئله های با اندازه کوچک مناسب است. براین اساس یک الگوریتم ژنتیک برای حل این مسئله در اندازه‌های متوسط و بزرگ ارائه شده است به طوری که زمان حل به میزان بهینه یا نزدیک به آن کاهش پیدا کرده است. نمونه‌های عددی نشان می‌دهد که الگوریتم ارائه شده کارا و مؤثر می‌باشد.

واژه‌های کلیدی: تک ماشین، الگوریتم ژنتیک، زودکرد خطی، دیرکرد توان دوم، زمان بیکاری ماشین و شکست کار

* نویسنده مسؤول: Mfakhrzad@yazduni.ac.ir

١- مقدمه

محیط‌های زمان بندی تک ماشین به‌طور واقعی در صنایع مختلف مانند صنعت شیمی استفاده می‌گردد (واکنر و همکاران، ٢٠٠٠). نگارش مقاله‌های زمانبندی با موضوع مدل‌های زمان‌بندی زودکرد/ دیرکرد، به علت اهمیت و ارتباط کاربردی آنها به طور قابل ملاحظه‌ای روبه افزایش است (هندل و همکاران ٢٠٠٩، والنته و همکاران ٢٠٠٨، والنته ٢٠٠٩). مدیران برنامه‌ریزی تولید, در چنین سازمان‌هایی زودکرد و دیرکرد را نامطلوب در نظر می‌گیرند. در حقیقت، مسائل زمان‌بندی با جرایم زودکرد و دیرکرد، با مفاهیمی مانند تولید بهنگام (JIT)[1] و مدیریت زنجیره تأمین سازگار هستند. تولید JIT تأکید می کند که هزینه زودکرد را، همانند دیرکرد (تأخیر) می‌توان کاهش داد. این کار به این علت است که وظایف زودکرد ممکن است به هزینه‌های انبار، مانند فرصت از دست‌رفته از سرمایه گذاری در انبار، هزینه‌های بیمه و ضایعات انبار منجر گردد. در عوض، کاری که دیرتر از موعد تحویل آماده می شود، ممکن است به نارضایتی مشتری، جرایم قراردادی، از دست دادن فروش و اعتبار منجر شود (لیائو و چنج٢٠٠٧).

مسائل زمان‌بندی تک ماشین با هزینه های زودکرد و دیرکرد بدون مجاز بودن وقفه به عنوان مساله NP-hard به طور وسیعی توسط لنسترا و همکاران در سال ١٩٧٧ مطالعه شده. همچنین، چندین الگوریتم دقیق و ابتکاری در این خصوص ارائه شده است (داویس و کانت١٩٩٣ ، بالبال و همکاران ١٩٩٣، هندل و سورد ٢٠٠٦، سورد و سایدهوم ٢٠٠٣، جفوسکا ٢٠٠٧).

جریمه دیرکرد توان دوم در صنایع مختلف، کاربرد مناسبی دارد، و برای کارهای با تأخیر استفاده می‌شود. در حقیقت، تأخیر صفت مهمی در کیفیت سرویس است، و نارضایتی مشتری تمایل به افزایش توان دیرکرد مانند تابع هزینه را دارد (تاگوچی ١٩٨٦). همچنین، جریمه دیرکرد در برخی موقعیت‌ها می تواند نسبت به توابع تأخیر بیشینه یا تأخیر خطی معمولی ترجیح داده شود؛ چنانکه سان و همکاران در سال ١٩٩٩ عنوان نموده‌اند. والنته در سال ٢٠٠٨ رویه کران پایین را همراه با یک الگوریتم شاخه و حد ارائه کرده است. در میان الگوریتم‌های ابتکاری الگوریتم‌های جدیدی از سوی وی در سال ٢٠٠٦ و ٢٠٠٧ ارائه شده است.

سورد و سیدهانم در سال ٢٠٠٣ یک نسخه عمومی از مساله وزن‌دار را در مساله تک ماشین بررسی کرده‌اند. در اکثر کارهای انجام شده در مساله تک ماشین، زمان‌بندی بدون در نظر گرفتن شکست کار فرض شده است (جفوسکا٢٠٠٧، تاگوچی١٩٨٦، سان و همکاران١٩٩٩، والنته٢٠٠٨). این مساله با در نظر گرفتن مفهوم شکست کار در مساله زمان‌بندی تک ماشین با مجاز بودن زمان بیکاری ماشین $\sum_{i=1}^{n} \alpha_i E_i + \beta_i T_i$ توسط هندل و همکاران در سال ٢٠٠٩ بررسی شده است. این مدل توسط والنته در سال ٢٠٠٨ که زودکرد و دیرکرد فعالیت را به صورت توان دوم در نظر گرفته و چندین الگوریتم ابتکاری را برای حل طراحی و عملکرد آنها را با یکدیگر مقایسه کرده، ارائه شده است. والنته در سال ٢٠٠٩ در مقاله دیگری، تابع هدف مساله را توسعه داده است، وی در مدل خود بیکاری و شکست در کار را لحاظ نکرده است و بیشتر الگوریتم‌های فرا ابتکاری، از قبیل GA، GA-IN،

MA و MA-IN و تفاوت بین الگوریتم‌ها را بیان کرده است. این مدل، همچنین توسط خورشیدیان و همکاران در سال۲۰۱۱ توسعه یافته است. آنها با استفاده از مدل هندل، با در نظر گرفتن بیکاری مجاز و وقفه در کار و افزودن محدودیت به مدل مربوطه، مدل جدیدی را ارائه کرده‌اند. وانگ و همکاران در سال ۲۰۱۱ در مدل خود فقط به دنبال کمینه کردن زودکرد را هدف قرار داده و الگوریتم های مختلفی برای حل ارائه کرده‌اند.

از آن جایی که در مدل ارائه شده در این مقاله شکست کار مجاز فرض شده است، لازم است تابع هدف بر اساس واقعیت و بر اساس ایجاد جریمه وقفه در کار مورد بررسی شود. در حقیقت، اگر کاری وقفه داشته باشد، زمانی که این کار صرف می کند، افزایش می‌یابد، و مقدار (WIP) [1] افزایش می یابد. یک هدف عمده در محیط‌های صنعتی نگه داشتن مقدار WIP به کمترین مقدار ممکن است، به طور واضح، هر کار j_i حداقل به مدت P_i بر روی ماشین می ماند. در مدل ارائه شده مدت زمانی را که ماشین بر روی کار فعالیتی انجام نمی دهد، جریمه می‌شود. این زمان بیکاری (برای کار j) برابر $C_i - S_i - P_i$ است. بر این اساس، در این مقاله با در نظر گرفتن توان دوم برای زمان دیرکرد، وزن‌دار کردن زمان زودکرد و دیرکرد، و اضافه کردن WIP در تابع هدف مدل جدیدی ارائه شده است. هدف از توان دوم در زمان دیرکرد، اهمیت بیشتر آن نسبت به زمان زودکرد از دیدگاه مشتری بوده است. همچنین، وزن‌دار کردن زمان زودکرد و دیرکرد، اهمیت تولید هر کدام از محصول را از دیدگاه تولید کننده نشان می دهد. با توجه به اینکه، حل مدل ریاضی به روش‌های دقیق در واقعیت امکان پذیر نیست، لذا از الگوریتم ژنتیک برای یافتن راه حل‌های خوب در زمان‌های محاسباتی منطقی استفاده شده است؛ به‌طوری که از شیوه‌های کلاسیک مؤثرتر و کاراتر است.

۲-تعریف مساله

مجموعه ای از n کار مستقل ($\{J_1, J_2, ..., J_n\}$) باید روی یک ماشین که در هر لحظه تنها یک کار را انجام می دهد، زمان‌بندی گردد. فرض کنید این ماشین به صورت پیوسته از لحظه صفر در دسترس بوده، زمان بیکاری و شکست کار هم مجاز باشند. کار $J_i, J = 1, 2, ..., n,$ به زمان پردازش p_i نیاز دارد و باید به صورت ایده آل در موعد مقرر خود d_i کامل شود. همچنین فرض کنید، α_i، β_i به ترتیب نشان دهنده جرایم زودکرد، دیرکرد و γ_iجریمه کار نیمه ساخته j_i باشند. x_i متغیر صفر و یک بوده، انتخاب کار j_i در لحظه k را نشان می دهد. همچنین، فرض شده است که تمامی زمان‌های پردازش و موعدهای تحویل عدد صحیح هستند و اندیس‌ها، پارامترها و متغیر های مساله به صورت زیر نمایش داده شده‌اند:

اندیس ها

i	*اندیس برای کارها*
k	*اندیس برای دوره های زمانی*
H	تعداد دوره های زمانی

پارامترها :

d_i	موعد تحویل برای کار j_i
p_i	زمان محاسباتی برای کار j_i
α_i	جریمه زودکرد برای کار j_i
β_i	جریمه دیرکرد برای کار j_i
γ_i	جریمه WIP به ازای هر واحد زمانی
M	مقدار بسیار زیاد

متغیر ها :

S_i زمان شروع برای کار j_i

C_i زمان تکمیل کار j_i

E_i زودکرد کار j_i

T_i دیرکرد کار j_i

$S_i^{'}$ زمان شروع برای کار j_i در لحظه $k = 2, 3, \dots, H$

$S_i^{''}$ زمان شروع برای کار j_i در لحظه $k = 1$

$$x_{ik} = \begin{cases} 1 & \text{اگرکار } j_i \text{ در زمان } k \text{ انتخاب شود} \\ 0 & \text{در غیر این صورت} \end{cases}$$

$$\delta_{ik} = \begin{cases} 1 & \text{if} \quad x_{ik} = 1 \\ M & \text{if} \quad x_{ik} = 0 \end{cases}$$

۳-مدل ریاضی

با توجه به تعریف پارامترها، تابع هدف و محدودیت ها را می توان به صورت زیر بیان نمود:

$$Min\ z = \sum_{i=1}^{n} \left((\alpha_i E_i + \beta_i T_i^2) + \gamma_i (C_i - S_i - P_i) \right) \quad (1)$$

بطوری‌که:

$$\sum_{k=1}^{H} x_{ik} = P_i \quad (2)$$

$$\sum_{i=1}^{n} x_{ik} \le 1 \quad (3)$$

$$C_i = max_{k=1}^{H}(k * x_{ik}) \quad (4)$$

$$S_i^{'} = min_{k=2}^{H}\left((k-1) * \delta_{ik}\right) \quad (5)$$

$i = 1,2, \dots, n \ , k = 2, 3, \dots, H$

$$S_i^{''} = (k * \delta_{ik}) - 1 \quad i = 1,2, \dots, n \ , k = 1 \quad (6)$$

$$S_i = min(S_i^{'}, S_i^{''}) \quad i = 1,2, \dots, n \quad (7)$$

$$T_i = max(0, C_i - d_i) \quad i = 1,2, \dots, n \quad (8)$$

$$E_i = max(0, d_i - C_i) \quad i = 1,2, \dots, n\, n \quad (9)$$

$$\delta_{ik} = M(1 - x_{ik}) + 1 \quad (10)$$

$i = 1,2, \dots, n \ , \ k = 1,2, \dots, H$

هدف، یافتن برنامه زمانی است که مجموع هزینه‌های زودکرد خطی و دیرکرد توان دوم را کمینه می نماید و میزان WIP را تا حد ممکن در سطح پایین‌تری نگه می‌داردکه در محدودیت (۱) نشان داده شده است. محدودیت (۲) تضمین می کند که هر کار j_i به p_i عملیات تقسیم گردد بطوریکه دارای زمان‌های اجرای خاص باشند. این عملیات باید به H مقطع زمانی تولید مجزا اختصاص داده شوند (t-1,t)، که در آن $1 \le t \le H$ و H طول زمان‌بندی می‌باشد. محدودیت (۳) تضمین می کند که در هر مقطع زمانی (t-1,t)، تنها یک بخش از کار می‌تواند اجرا شود و این

جدول۱: مقادیر پارامتر مساله

Job j_i	p_i	d_i	α_i	β_i	γ_i
j_1	۲۰	۸۲	۹	۱۷	۱۶
j_2	۶۵	۱۲۶	۶	۸	۱۱
j_3	۱۵	۱۶۲	۱۲	۳	۶
j_4	۵	۱۳۵	۱	۱۹	۷
j_5	۵۰	۱۱۲	۱۸	۱۱	۱۰

محدودیت قادر است زمان بیکاری ماشین را در نظر بگیرد. محدودیت (۴) در ارتباط با زمان تکمیل هر کار می‌باشد. برای محاسبه زمان شروع هر کار j_i بعد از NP-hard. قرار دارد که الگوریتم ژنتیک می‌تواند به‌عنوان یکی از راه حل‌ها مورد توجه قرار گیرد

دوره زمانی دوم محدودیت (۵) ارائه شده است. محدودیت (۶) در ارتباط با محاسبه زمان شروع هر کار j_i در لحظه اول می باشد. محدودیت (۷) با استفاده از محاسبه حداقل محدودیت های(۴) و (۵) زمان شروع هر کار j_i در هر لحظه از زمان را محاسبه می کند. هزینه دیرکرد و زودکرد هر کار را به ترتیب در محدودیت (۸) و (۹) نشان داده شده است. محدودیت (۱۰) جهت برقرار کردن محدودیت‌های (۵، ۶ و۷) می‌باشد.

٤-پیچیدگی مساله

نرم افزار Lingo می تواند راه حل بهینه مسائل دارای اندازه کوچک را در زمانی کم محاسبه کند. اما مسائل واقعی و با اندازه بزرگ را به دلیل تعداد بیش از اندازه متغیرها و محدودیت‌ها نمی‌توان با Lingo در زمان معقولی حل نمود. به عنوان مثال مسئله‌ای با ۵ کار را در Lingo و بر اساس پارامترهای لیست شده در جدول ۱، پس از گذشت ۲۰ ساعت و با ۳۷۲, ۶۶۵ و ۴۹۵ تکرار حل نشده است. به همین دلیل اینگونه مسائل با حل در اندازه‌های بزرگ در ردیف مسائل

بر اساس نتایج بدست آمده، مشاهده گردید که راه‌حل‌های این دو روش (دقیق، و فرا ابتکاری) برای مسائل با اندازه کوچک شبیه هم هستند.

برای پیش‌گیری از افزایش بی‌حد و مرز در فضای راه حل، یک مجموعه‌برتر ارائه شده است.

فضای راه حل این مجموعه برتر با اعمال طول فرآیند مطرح شده (H) ، به صورت رابطه (۱۱) به‌دست می‌آید

$$H = max\left\{\sum_{i=1}^{n} P_i, d_{max}\right\} + n$$

$$where$$

$$d_{max} = max\{d_i\} \quad ,$$

$$i = 1,2,\dots,n \qquad (11)$$

زمانی که طول فرآیند برابر با مقدار:

$$\max\{\sum_{i=1}^{n} p_i, d_{\max}\} + n$$

در نظر گرفته شود، و وقفه در کار برابر با یک واحد زمانی باشد، فضای راه‌حل دارای $(\max\{\sum_{i=1}^{n} p_i, d_{\max}\} + n)!$ حالت خواهد بود. عبارت $\max\{\sum_{i=1}^{n} p_i, d_{\max}\}$، تضمین می‌کند که ضای راه حل همیشه شدنی و امکان پذیر می‌ماند،

چون طول فرآیند همیشه بزرگتر یا مساوی مجموع زمان‌های پردازش است.

۵- رویکرد الگوریتم ژنتیک

ایده اصلی الگوریتم‌های تکاملی در سال ۱۹٦۰ از سوی ریچنبرگ مطرح شد. الگوریتم ژنتیک که منشعب از این الگوریتم‌هاست، در حقیقت روش جستجوی رایانه‌ای بر پایه الگوریتم بهینه‌سازی و بر اساس ساختار ژن و کروموزم است که به وسیله پرفسورهولند در دانشگاه میشیگان مطرح و پس از وی توسط جمعی از دانشجویان، مانند گلدبرگ در سال ۱۹۸۹ توسعه یافت. الگوریتم ژنتیک با الهام از تئوری داروین و بر اساس اصل ادامه حیات بهترین‌ها پی‌ریزی شده است. می‌توان‌گفت یکی از مزیت های اصلی الگوریتم ژنتیک نسبت به روش‌های قدیم بهینه‌سازی در این است که GA با جمعیت یا مجموعه‌ای از نقاط در یک لحظه خاص کار دارد، در حالی‌که در روش‌های قدیم بهینه‌سازی تنها برروی یک نقطه خاص عمل می گردد. این بدان معناست که GA تعداد زیادی از طرح‌ها را در یک زمان پردازش می‌کند. برخی مراجع رویکرد الگوریتم ژنتیک را به صورت جزیی بررسی می کنند (ریوز۱۹۹۷، ریوز و همکاران ۲۰۰۳، بین۱۹۹٤). این الگوریتم در ابتدا با مجموعه‌ای از جواب‌های تصادفی (کروموزم) که به آن جمعیت اولیه گفته می شود، آغاز و سپس مقدار شایستگی هر کروموزم با توجه به تابع شایستگی تعیین می‌گردد. کروموزوم‌های با شایستگی بالاتر شانس بیشتری برای تولید فرزندان دارند. بر همین اساس، عمل انتخاب والدین انجام می گیرد و سپس فرزند به وسیله عمل تقاطع روی والدین به‌وجود می آید، سرانجام بعضی از ژن‌های فرزند با عمل جهش تغییر می یابد و بعد فرزند جدید جانشین ضعیفترین کروموزم در مجموعه اولیه می‌شود.

گام‌های اصلی در پیاده‌سازی الگوریتم ژنتیک به صورت زیر هستند:

۱. شکل کروموزم
۲. جمعیت اولیه
۳. شایستگی تابع برای محاسبه تناسب جمعیت
٤. استراتژی های انتخاب
۵. عملگرهای ژنتیک

۵-۱- الگوریتم ژنتیک ارائه شده

برای حل مساله زمان‌بندی یک ماشین با در نظر گرفتن وقفه و زمان بیکاری ماشین، الگوریتم ژنتیکی ارائه شده است. مؤلفه‌های اصلی پیاده سازی GA شامل موارد زیر هستند:

۵-۱-۱ شکل کروموزم

برای هر پیاده‌سازی الگوریتم ژنتیک، نخستین گام نگاشت خصوصیات ویژه راه‌حل در قالب یک رشته کروموزوم است. هر کروموزوم از ترکیبی از ژن‌ها، از الفبایی خاص ساخته شده است. این الفبا می تواند مجموعه‌ای از اعداد دودویی، اعداد حقیقی، اعداد طبیعی، نمادها، یا ماتریس ها باشد. طرح ارائه دهنده نه‌تنها تعیین کننده میزان مؤثر طراحی مساله است، بلکه حتی تعیین می‌نماید که از عملگرهای ژنتیک تا چه میزان مؤثری می‌توان استفاده نمود. برای نمایش آن، دو کار در نظر گرفته شده است، و زمان پردازش، موعد پایان، جریمه زودکرد و دیرکرد آنها در جدول ۲ آمده است. همچنین، نمایش کدگذاری کروموزوم به صورت گرافیکی در شکل ۱ نشان داده شده است.

جدول ۲ : پارامترهای دو وظیفه داده شده.

$Job\ j_i$	p_i	d_i	α_i	β_i
j_1	4	5	2	4
j_2	2	7	1	3

شکل (۱) : نحوه کدگذاری سیستم

در ابتدا، یک ترتیب اولیه تولید می‌شود، به طوری که طول کروموزم برابر H بوده مقدار ژن‌ها در آن برای کارj_1 به تعداد P_1 (در اینجا ٤) عدد ۱ و برای کار j_2 به تعداد P_2 (در اینجا ٤) عدد ۲ و مابقی عدد

بنابراین، هر کروموزوم یک بردار رمزگشایی تصادفی بین ۰ و ۱ است. برای این‌که تناسب هر کدام اندازه‌گیری شود، لازم است که کروموزوم آن از یک راه حل متناظر با مساله تغییر داده شود. رمزگشایی یا نگاشت یک کروموزوم به یک رشته به وسیله مرتب‌سازی کار انجام می شود. اگر تعداد وظیفه‌ها بیشتر باشد، رویه را به همین صورت ادامه می‌دهیم.

یک ویژگی مهم الفبای کلید تصادفی این است که تمامی زاد و ولد انجام شده توسط عمل تقاطع جزو راه حل‌های امکان‌پذیر به شمار می‌آیند. اگر هر بردار تصادفی را بتوان به یک راه حل امکان‌پذیر تغییر داد، آنگاه هر کروموزوم گرفته شده توسط تقاطع نیز با یک راه حل امکان‌پذیر ارتباط دارد. بر این اساس، الگوریتم ژنتیک رابطه بین بردارهای اعداد تصادفی و صفر که نشان دهنده بیکاری است، تولید گردد. سپس از یک الفبای کلید تصادفی استفاده شده $U(0,1)$ تا کروموزوم‌ها را رمزگشایی کند(بین،۱۹۹٤). در این الفبا هر عضو یک مقدار تصادفی بین ۰ و ۱ است. راه حل‌های با تابع شایستگی و ارزش تابع هدف خوب را جستجو می کند (والنته،۲۰۰۹).

۵-۱-۲ جمعیت اولیه

مرحله دوم پیاده‌سازی الگوریتم ژنتیک، تولید یک سری راه‌حل‌های اولیه است، که به آن جمعیت می‌گویند. تعداد راه‌حل‌های اولیه‌ای که باید در این جمعیت قرار گیرند، اندازه جمعیت نامیده می شوند. جمعیت اولیه تنها یک بار در ابتدای اولین نسل الگوریتم ژنتیک تولید می شود. تعیین اندازه مناسب جمعیت، یک انتخاب مهم در پیاده‌سازی الگوریتم ژنتیک است. اگر میزان انتخاب شده کوچک باشد، ممکن است قادر به دستیابی به راه حل خوبی نباشیم. از سوی دیگر، اگر این عدد خیلی بزرگ باشد، ممکن

است برای رسیدن به جوابی بهتر، زمان زیادی از CPU را به خود اختصاص دهد.

۵-۱-۳ تابع شایستگی

در پیاده‌سازی GA یک تابع تناسب برای ارزیابی و بازتولید کروموزوم های جدید، به نام "اولاد نسل های بعدی"، استفاده می‌شود. هدف از این تابع شایستگی اندازه‌گیری کاندیداهای پیشنهاد شده در جمعیت، با در نظر داشتن توابع هدف و محدودیت در مدل موجود، است. با توجه به تابع هدف، تناسب کروموزوم ها باید با محاسبه دو پارامتر زمان شروع و زمان تکمیل انجام گیرد. با محاسبه این پارامترها، زودکرد و دیرکرد هرکدام از کارها با اعمال روابط $T_i = max(0, C_i - d_i)$ و $E_i =$ $max(0, d_i - C_i)$ به‌دست می آید. در نهایت تابع هدف هر کروموزوم با اعمال Ei و Ti در فرمول (۱۲) به‌دست می آید:

$$\sum_{i=1}^{n}\left((\alpha_i E_i + \beta_i T_i^2) + \gamma_i (C_i - S_i - P_i)\right) \qquad 12$$

٥-۱-٤ انتخاب استراتژی‌ها

رویه انتخاب چرخ رولت، که از سوی گلدبرگ در سال۱۹۸۹ ارائه شده است، استراتژی انتخابی است که در الگوریتم پیشنهادی استفاده شده است. هدف این استراتژی انتخاب، بیشتر در نظر داشتن مناسبترین موارد، برای تولید دوباره فرزندان برای نسل بعد است.

۵-۱-۵ عملگرهای ژنتیک

برای تولید نسل‌های بعدی عملگرهای تقاطع و جهش والدین را به صورت جفت از جمعیت انتخاب می‌کنیم؛ یعنی مکانیزمی که به وسیله آن جمعیت جدیدی از افراد موجود و به اندازه جمعیت فعلی را تشکیل دهد. اندازه جمعیت در کل روند کار ثابت باقی می‌ماند که در ادامه شرح داده شده است :

تقاطع : فرزندان یا راه‌حل‌های جدید با برخورد دو ترتیب یا والد دیگر از طریق عملگری به نام تقاطع حاصل می‌شوند. عملگرهای تقاطع باید از تولید راه‌حل‌های نشدنی کاملاً دوری کنند. هدف تولید اولاد بهتری است؛ یعنی ساختن ترتیب‌هایی بهتر پس از ترکیب والدین است (میچالویکس،۱۹۹٦). با توجه به تنوع گسترده‌ای از عملگرهای تقاطع پیشنهادی، تقاطع در نظر گرفته شده در این مقاله، به‌صورت زیر است. برای نشان دادن عملگر تقاطع دو ترتیب والد زیر را در نظر بگیرید:

والد ۱ : A B C | A A B | A C B A

والد ۲ : A A B | B B C | C A A A

نقاط برش به صورت تصادفی تعیین می شود. در ادامه با جابه‌جایی کاراکترهای پس از برش راستی به ابتدای ترتیب، ترتیبی جدید شکل می‌گیرد که نتایج آن به شکل زیر است:

توالی جدید ۱:

A C B | A A B | C A A B (A C A A A A B)

توالی جدید ۲ :

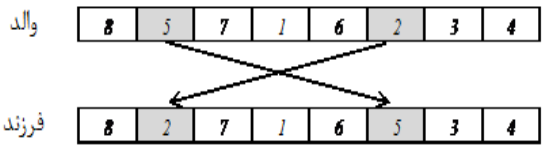
C A A | A A A| B B B C (C A A A B B C)

از ترتیب جدید، کاراکترهایی که با کاراکترهای بین نقاط برش والد دیگر مطابقت دارند، برداشته می‌شوند (در پرانتز نشان داده شده است). سپس ترتیب میان پرانتزهای والد و کاراکترهای باقی مانده از دیگر ترتیب جدید برای تولید فرزند مورد استفاده می‌شوند. با استفاده از این رویه، فرزندان زیر تولید می شوند:

Offspring 1: B B C | A A B | C A A A

Offspring 2: A A B | B B C | A C A A

جهش: عملگر جهش از ترتیب‌های اولیه برای تولید کروموزوم های تغییر یافته استفاده می‌نماید. در این رویه، یک مورد به صورت تصادفی انتخاب می‌گردد، و در ادامه، دو ژن تصادفی میان (۱ و H) انتخاب می‌شوند. عملگر جهش با تعویض دو ژن در کروموزوم انتخاب شده استفاده می شود. شکل ۲ نشان می دهد که دو عدد به دست آمده ۲ و ۵ هستند و فرزندان نیز با جایگزینی این دو ژن تولید شده‌اند.



شکل (۲): تولید فرزند از طریق عملگر جهش

٦- طراحی آزمایش‌ها و تنظیم داده‌ها

در این بخش، کارایی GA پیشنهاد شده برای این مساله ارائه شده است. هدف، آزمایش کارایی GA پیشنهاد شده برای مساله زمان‌بندی مسئله تک ماشین با بیکاری مجاز و هزینه‌های زودکرد خطی و دیرکرد توان دوم و شناسایی نمونه‌های پیچیده است. برای نشان دادن کارایی الگوریتم پیشنهاد شده، مسائل به صورت تصادفی به صورت زیر تولید شدند. مسائل با اندازه کوچک با ۳، ٤ و ۱۰ کار، اندازه متوسط با ۲۰ و ٤۰ کار، و اندازه بزرگ با ۸۰ تا ۱۰۰ کار در نظر گرفته شده است. زمان‌های پردازش از توزیع یکنواخت گسسته (۲۰ و ۱) و زمان‌های زودکرد و دیرکرد از توزیع یکنواخت گسسته (۱۰ و ۱) تولید شده‌اند. زمان‌های تحویل هر کار از توزیع یکنواخت به صورت روابط زیر تعریف شده‌اند:

$$[d_{min} - \lambda, d_{min} + \lambda\,] \qquad (13)$$

$$d_{min} = P \times (1 - TEF) \qquad (14)$$

$$P = \sum_{i=1}^{n} p_i \qquad (15)$$

$$\lambda = P \times (\frac{RDD}{2}) \qquad (16)$$

دو پارامتر TEF و RDD محدوده مناسبی از موعد تحویل هستند. مقادیر ۰/۵ و ۰/۸ برای RDD و ۰/۲ و ۰/۵ برای TEF در نظر گرفته شده است.

٦-۱ تنظیم پارامترها

به منظور درجه‌بندی الگوریتم، چندین روش برای طراحی آزمایش‌ها وجود دارد، اما رویکردی که بیشترین استفاده را دارد، یک آزمایش فاکتوریل کامل است (مونتگومری،۲۰۰۰). این رویکرد همیشه کارآمد نیست، زیرا اجرای بازرسی‌ها زمانی که تعداد فاکتورها به میزان قابل ملاحظه‌ای زیاد می‌شود، حل مساله سختتر خواهد شد. برای کاهش تعداد آزمایش‌های لازم، آزمایش فاکتوریل کسری (FFE) توسعه داده شد (کوچران و کوکس،۱۹۹۲). تاگوچی در سال۱۹۸٦خانواده‌ای از ماتریس های FFE را

توسعه داد که نهایتاً تعداد آزمایش‌ها را کاهش می‌دهند، اما هنوز هم اطلاعاتی کافی فراهم می‌نماید. در روش تاگوچی، برای مطالعه تعداد زیادی از متغیرهای تصمیم‌گیری با آزمایش‌های گوناگون، آرایه‌های متعامد استفاده شده‌اند. در اینجا مقادیر زیر برای پارامترهای لازم در الگوریتم ژنتیک در نظر گرفته شده است : احتمال تقاطع (pc) : سه سطح (۰/۹۰ ، ۰/۸۵ ، ۰/۸۰) احتمال جهش (pm) : سه سطح (۰/۰۳ ، ۰/۰٦ و ۰/۰۹) تعداد جمعیت اولیه (np) : سه سطح (۲۰۰ و ۱۰۰) تعداد نسل ها (ng) : یک سطح (۱۰۰). بهترین مقدار آزمایش‌های محاسباتی برای مساله زمان‌بندی تک ماشین، با مجموع زودکرد و دیرکرد اعمال شده، از طریق مقایسه نتایج آزمایش‌ها با تغییر پارامترها که به عنوان مقادیر پیش فرض در نظر گرفته شده بود، به صورت زیر به‌دست آمد: pc = 0.80، pm = 0.06، np = 100 و ng = 100. جدول ۳ نشان می‌دهد که الگوریتم ژنتیک طراحی شده در مقابل حل Lingo، تعداد کارهای متوسط و بزرگ را در زمان مناسبی حل کرده است. همچنین Lingo تعداد کارهای کوچک را در زمان قابل قبولی حل نموده است.

آخرین ستون جدول ۳، درصد اختلاف بین جواب بهینه و جواب GA را نشان می‌دهد که با فرمول ۱۷ که در زیر نشان داده می شود. تمامی جواب‌ها در سطوح کارهای کوچک مقدار Gap برابر صفر بوده، نشان دهنده این است که الگوریتم ژنتیک در سطوح کارهای کوچک در بهترین جواب به جواب بهینه Lingo دست یافته است

٦ -۲ نتایج آزمایش‌ها

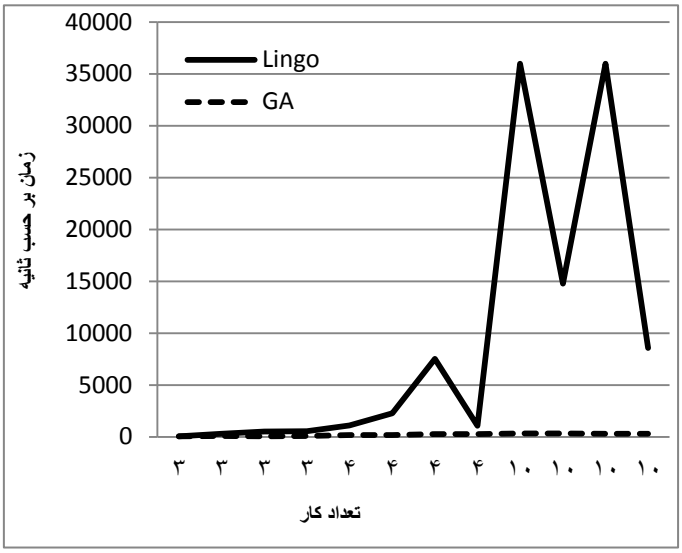
الگوریتم ژنتیک ارائه شده با استفاده از نرم افزار MATLAB 7.8 و روی یک PC Pentium IV با ۲/۸ گیگاهرتز سرعت و ۵۱۲ مگابایت RAM، اجرا شد. ۲۸ مساله آزمایش با تعداد کارهای ۳ تا ۱۰۰ و تکرار ۱۰ بار بررسی شد. جدول ۳ نشان می‌دهد که الگوریتم پیشنهاد شده در داشتن زمان محاسباتی قابل قبول، بسیار مناسب است. همچنین، نشان می‌دهد که Lingo می‌تواند راه حل بهینه را برای مسائل با اندازه کوچک در زمانی اندک محاسبه نماید. شکل ۳ نمودار مقایسه بین زمان پردازش GA با زمان پردازش لینگو در مسائل با ابعاد کوچک را نشان می‌دهد. این نمودار کارایی الگوریتم ژنتیک را در ابعاد کوچک از نظر زمانی نسبت به نرم افزار لینگو، نشان می‌دهد.

برای نشان دادن فرآیند الگوریتم ژنتیک، متوسط هزینه‌ها و بهترین آنها در هر نسل در شکل (٤) نشان داده شده است. از مزایای اصلی که می‌توان برای الگوریتم ژنتیک نام برد، سرعت بسیار بالای این الگوریتم است به‌طوری‌که کمترین هزینه در تکرار ۵۳ رخ داده است و از نسل ٦٨ متوسط هزینه‌ها و بهترین جواب در هر نسل همگرا شده‌اند و این نشان دهنده شرایط بهینه در همه کروموزوم‌هاست

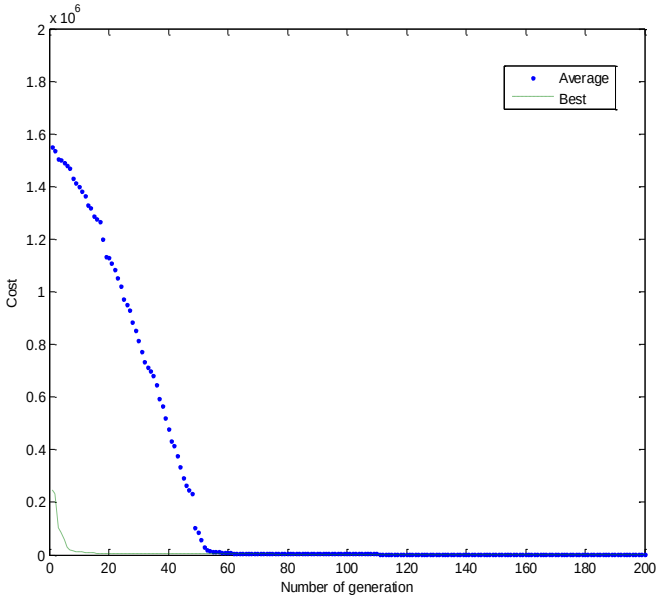
۷ – نتیجه‌گیری

در این مقاله، توسعه مساله زمان‌بندی تک ماشین با هزینه‌های زودکرد خطی و دیرکرد با مجاز بودن شکست کار و زمان بیکاری ماشین بررسی شد. برای این مساله مدلی ریاضی ارائه گردید که پیکربندی ترتیب کارها را با هدف کمینه ساختن هزینه‌های زمان‌بندی پیدا می‌کند. الگوریتمی کارا براساس GA

برای حل این مدل ریاضیاتی طرح‌ریزی شد. به آزمایش حل شدند. نتایج محاسباتی از لحاظ جواب‌های مساله مقایسه شد و از آنجایی که Lingo توانایی اجرای فعالیت‌ها با تعداد کارهای متوسط و بزرگ را ندارد، برتری الگوریتم پیشنهاد شده را در ترتیب‌دهی کارها نسبت به روش‌های دقیق نشان منظور تأیید کارایی GA ارائه شده، ۲۸ مساله می‌دهد. به علاوه، الگوریتم ژنتیک از لحاظ زمانی نیز مقایسه شد. با افزایش تعداد کارها قابلیت استفاده الگوریتم در داشتن زمان محاسباتی مناسب نیز اثبات شده است.



شکل(۳): مقایسه بین نتایج زمان GA با زمان حل بهینه حاصل از Lingo



شکل (٤) : متوسط هزینه و بهترین جواب کروموزم

جدول ۳ : نتایج مدل حل شده توسط Lingo 9 و GA پیشنهادی

Number of jobs	RDD	TEF	Lingo		GA			Gap%
			Optimal solution	Computational time*	Average of solutions	Best solution	Computational time*	
3	0.5	0.2	45	0:00:56	48.2	45	0:00:40	۰
3	0.5	0.5	65	0:05:16	72	65	0:01:40	۰
3	0.8	0.2	0	0:08:26	0	0	0:00:50	۰
3	0.8	0.5	62	0:09:11	70.4	62	0:01:12	۰
4	0.5	0.2	40	0:18:17	52.8	40	0:03:05	۰
4	0.5	0.5	36	0:38:09	44.6	36	0:02:51	۰
4	0.8	0.2	44	2:05:28	49.2	44	0:04:16	۰
4	0.8	0.5	78	0:17:42	78	78	0:04:22	۰
10	0.5	0.2	---	10:00:00	140.7	126	0:05:43	---
10	0.5	0.5	114	4:06:32	123.2	114	0:05:25	۰
10	0.8	0.2	---	10:00:00	210.5	202	0:04:46	---
10	0.8	0.5	182	2:23:06	184.4	182	0:05:08	۰
20	0.5	0.2	---	10:00:00	5329	5166	8 min	---
20	0.5	0.5	---	10:00:00	5357	5025	8 min	---
20	0.8	0.2	---	10:00:00	3636	3443	8 min	---
20	0.8	0.5	---	10:00:00	3676	3375	8 min	---
40	0.5	0.2	---	10:00:00	11364	10931	10 min	---
40	0.5	0.5	---	10:00:00	13351	13016	10 min	---
40	0.8	0.2	---	10:00:00	13987	13847	10 min	---
40	0.8	0.5	---	10:00:00	10769	10364	10 min	---
80	0.5	0.2	---	10:00:00	52312	51430	14 min	---
80	0.5	0.5	---	10:00:00	55337	52183	14 min	---
80	0.8	0.2	---	10:00:00	51389	50008	14 min	---
80	0.8	0.5	---	10:00:00	48040	46512	14 min	---
100	0.5	0.2	---	10:00:00	83212	83001	16 min	---
100	0.5	0.5	---	10:00:00	91151	90366	16 min	---
100	0.8	0.2	---	10:00:00	78255	78352	16 min	---
100	0.8	0.5	---	10:00:00	84918	82892	16 min	---

منابع:

Bean, J.C., (1994). "Genetics and random keys for sequencing and optimization". *ORSA Journal on Computing*;6:154–60.

Bülbül, K., Kaminsky, P., Yano, C., (2007). "Preemption in single machine earliness/ tardiness scheduling". *Journal of Scheduling*, 10, 271–292.

Cochran, W.G., Cox, G.M., (1992). "Experimental designs:. *2nd ed. New York: Wiley*.

Davis, J.S., Kanet, J.J., (1993)." Single-machine scheduling with early and tardy completion costs". *Naval Research Logistics*, 40, 85–101

Goldberg, D.E., (1989)." Genetic algorithms in search, optimization, and machine learning. Reading*", MA: Addison-Wesley*

Hendel, Y., Sourd, F., (2006)." Efficient neighborhood search for the one-machine earliness–tardiness scheduling problem". *European Journal of Operational Research*, 173, 108–119.

Hendel, F., Runge, N., Sourd, F., (2009)." The one-machine just-in-time scheduling problem with preemption". *Discrete Optimization*, 6, 10-22.

Holland, J.H., "Adaptation in natural and artificial systems. Ann Arbor", *Michigan: University of Michigan Press.*

Jźefowska, J., 2007. Just-in-time scheduling. Berlin: Springer

Khorshidian, H., Javadian, N., Zandieh, M., Rezaeian, J., Rahmani, K., 2011. " A genetic algorithm for JIT single machine scheduling with preemption and machine idle time". *Expert Systems with Applications* 38, 7911-7918.

Lenstra, J.K., Rinnooy Kan, A.H., Brucker, G.P., 1977. "Complexity of machine scheduling problems". *Annals of Discrete Mathematics*, 1, 343–362.

Liao, C.J., Cheng, C.C., 2007. "A variable neighborhood search for minimizing single machine weighted earliness and tardiness with common due date". *Computers and Industrial Engineering*, 52, 404–413.

Michalewicz, Z., 1996. Genetic Algorithms + Data Structures = Evolution Programs. Springer, New York.

Montgomery, D.C., 2000. "Design and analysis of experiments". *5th ed. New York:*

Reeves, C.R., 1997. "Genetic algorithms for the operations researcher". *INFORMS Journal on Computing;* 9:231–50.

Reeves, C., Glover F, Kochenberger., 2003. "Handbook of metaheuristics." *Dordrecht: Kluwer Academic Publishers*; p. 55–82.

Sourd, F., Kedad-Sidhoum, S., 2003."The one machine problem with earliness and tardiness penalties". *Journal of Scheduling*; 6:533–49.

Sun, X., Noble, J.S., Klein C.M., 1999. "Single-machine scheduling with sequence dependent setup to minimize total weighted squared tardiness". *IIE Transactions*; 31:113–24.

Taguchi, G., 1986. "Introduction to quality engineering. White Plains": *Asian Productivity Organization/UNIPUB*

Valente, J.M.S., 2006. Heuristics for the single machine scheduling problem with early and quadratic tardy penalties.

Valente, J.M.S., 2007. "Beam search heuristics for the single machine scheduling problem with linear earliness and quadratic tardiness costs. Working Paper 250, Faculdade de Economia, Universidade do Porto,

Portugal";. *Asia-Pacific Journal of Operational Research, to appear*.

Valente, J.M.S., Alves, R.A.F.S., 2008. "Heuristics for the single machine scheduling problem with quadratic earliness and tardiness penalties". *Computers & Operations Research*. 35, 3696-3713.

Valente, J.M.S., 2008. "An exact approach for the single machine scheduling problem with linear early and quadratic tardy penalties. Asia-Pacific" *Journal of Operational Research;*

Valente, J.M.S. 2009. "A genetic algorithm approach for the single machischeduling problem with linear earliness and quadratic tardiness penalties ". *Computers & perations Research* 36, 2707-2715.

Wagner, B.J., Davis, D.J, Kher, H., 2002."The production of several items in a single facility with linearly changing demand rates". *Decision Sciences*; 33:317–46.

Wang, X.R., Hung, X., Wang, J.B., 2011. " Single-machine scheduling with linear decreasing deterioration to minimize earliness penalties". *Applied Mathematical Modeling* 35, 3509–3515..

پي‌نوشت:

1-Just In Time
2- Work In Process