

زمان بندی ماشین های موازی غیر مرتبط با محدودیت های پردازش و

زمان های آماده سازی وابسته به توالی

فردین احمدی زر^{۱*}، کسری مهدوی^۲، جمال ارکات^۱

۱. دانشیار گروه مهندسی صنایع دانشگاه کردستان

۲. کارشناسی ارشد مهندسی صنایع دانشگاه کردستان

(تاریخ دریافت: ۹۶/۰۶/۰۱، تاریخ دریافت روایت اصلاح شده: ۹۷/۰۷/۰۷، تاریخ تصویب: ۹۸/۰۱/۱۷)

چکیده

در مسائل دنیای واقعی، ماشین ها به دلایلی از قبیل خرابی های ناگهانی، فعالیت های نگهداری و تعمیرات، یا عملیات از قبل برنامه ریزی شده در بعضی دوره های زمانی در دسترس نیستند. در این پژوهش، مسئله زمان بندی ماشین های موازی غیر مرتبط با این فرض که هر ماشین در بعضی زمان ها طی افق برنامه ریزی در دسترس قرار ندارد و همچنین ممکن است توانایی پردازش بعضی کارها را نداشته باشد، بررسی شده است. این محدودیت ها، محدودیت های پردازش نامیده می شوند. از سوی دیگر، فرض می شود زمان های آماده سازی به توالی کارها و نیز ماشین وابسته است. تابع هدف مسئله مورد بررسی کمینه کردن مجموع زود کردها و دیر کردهاست. ابتدا مسئله به صورت مدل برنامه ریزی خطی عدد صحیح مختلط ارائه شد. سپس به دلیل NP-hard بودن آن، الگوریتم رقابت استعماری که از رویکردی جدید برای رمزگشایی بهره می برد، برای حل مسائل با ابعاد بزرگ ارائه شد. همچنین به منظور ارزیابی عملکرد الگوریتم پیشنهادی، تولید و حل تعدادی مسئله نمونه صورت گرفت.

واژه های کلیدی: آماده سازی وابسته به توالی، الگوریتم رقابت استعماری، زمان بندی، ماشین های موازی غیر مرتبط، محدودیت های پردازش.

مقدمه

سوی دیگر، بررسی دسته ای از ماشین ها که به صورت موازی قرار گرفته اند، از هر دو دیدگاه نظری و عملی اهمیت دارد. از دیدگاه نظری، ماشین های موازی تعمیمی از حالت تک ماشین و حالت خاصی از کارگاه جریان انعطاف پذیر^۱ است. از دیدگاه عملی این مسئله به این دلیل اهمیت دارد که قرار گرفتن منابع به طور موازی در جهان واقعی امری متداول است. در ادامه، پژوهش های انجام شده در حوزه زمان بندی ماشین های موازی، که محدودیت دسترسی به ماشین ها را در نظر گرفته اند، بررسی می شود.

سان و لی [۱] مسئله زمان بندی دو ماشین موازی را با این فرض که هر یک از ماشین ها در بازه هایی از زمان برای فعالیت های نگهداری و تعمیرات در دسترس نیست و نیز زمان کارکرد متوالی هر ماشین نمی تواند بیشتر از یک مدت از پیش تعیین شده باشد، بررسی کردند. آن ها دو مدل متفاوت را در نظر گرفتند. در مدل اول، فرض این است که فعالیت های نگهداری و تعمیرات به صورت دوره ای انجام می شود و هدف

زمان بندی فرایند تصمیم سازی است که در بسیاری از صنایع خدماتی و تولیدی کاربرد دارد. زمان بندی با تخصیص منابع به کارها در بازه های زمانی سروکار داشته و هدف آن بهینه سازی یک یا چند معیار است. به طور کلی، روش های هوشمند زمان بندی برای تخصیص کارها به ماشین ها هنگام مواجهه با محدودیت زمان و منابع مورد نیاز کاربرد دارد. در مسائل کلاسیک زمان بندی فرض می شود ماشین ها به طور پیوسته در تمام مدت برنامه تولید در دسترس هستند. این فرض ممکن است در مواردی توجیه پذیر باشد، اما در بسیاری از موقعیت های عملی در دنیای واقعی توجیه پذیر نیست. عملیات یک ماشین در دوره هایی از زمان می تواند به دلایل گوناگون از قبیل بازرسی یا تعمیرات پیشگیرانه قطع شود؛ بنابراین بررسی روش هایی برای زمان بندی کارها با توجه به از دسترس خارج شدن ماشین ها در بازه هایی از زمان مهم است. از

کمینه کردن مجموع وزنی زمان‌های تکمیل و با این فرض که ماشین‌ها در بازه‌هایی از زمان به‌دلیل تعمیرات از دسترس خارج می‌شوند بررسی کرد و الگوریتم شاخه و کران را برای حل آن ارائه داد. لی و چن [۹] مسئله کمینه کردن دامنه عملیات را با این فرض که حداقل یک ماشین همواره در دسترس است و سایر ماشین‌ها در بازه‌ای از زمان در دسترس نیستند، بررسی کردند. براساس فرضیه آن‌ها، پردازش کار می‌تواند متوقف شود و پس از در دسترس قرار گرفتن ماشین ادامه یابد. همچنین نشان دادند مسئله کمینه کردن مجموع وزنی زمان‌های تکمیل NP-hard بوده و الگوریتم برنامه‌ریزی پویا برای حل آن پیشنهاد دادند. قربی و هائوری [۱۰] یک الگوریتم شاخه و کران را برای مسئله کمینه کردن دامنه عملیات ارائه دادند؛ با این فرض که کارها در زمان‌های مختلفی در دسترس قرار می‌گیرند. شین و همکاران [۱۱] یک الگوریتم چندجمله‌ای برای مسئله کمینه کردن دامنه عملیات را درحالی‌که هر ماشین تنها توانایی پردازش بعضی از کارها را دارد، ارائه کردند. آن‌ها همچنین یک الگوریتم شاخه و کران را برای کمینه کردن بیشینه تأخیر^۴ با این فرض که موعد تحویل کارها مستقل از یکدیگر است، توسعه دادند. شن و همکاران [۱۲] مسئله را با این فرض که بعضی از ماشین‌ها ممکن است از ابتدا در دسترس نبوده است و به‌مرور در دسترس قرار می‌گیرند و دو تابع هدف را بررسی کردند؛ تابع اول کمینه کردن مجموع زمان‌های تکمیل و مجموع قدر مطلق اختلاف این زمان‌ها و تابع دوم کمینه کردن مجموع زمان‌های انتظار کارها و مجموع قدر مطلق اختلاف زمان‌های انتظار. آن‌ها نشان دادند الگوریتم چندجمله‌ای برای یافتن جواب بهینه هر یک از مسائل وجود دارد. هو و ژائو [۱۳] مسئله کمینه کردن مجموع زمان‌های تکمیل را به‌گونه‌ای که دامنه عملیات از حد مشخصی تجاوز نکند بررسی کردند. همچنین با این فرض که در هر دوره زمانی حداقل ۲ ماشین در دسترس است و پردازش یک کار می‌تواند متوقف شود و بعد از در دسترس قرار گرفتن ماشین ادامه یابد، الگوریتمی را برای یافتن جواب بهینه ارائه کردند. لی و همکاران [۱۴] مسئله را با هدف کمینه کردن مجموع دیرکردها و با این فرض که ماشین‌ها نیازمند فعالیت‌های نگهداری و تعمیرات هستند، بررسی کردند.

کمینه کردن دامنه عملیات^۲ است. در مدل دوم نیز فرض شده است زمان این فعالیت‌ها در ابتدای زمان بندی مشخص، و هدف کمینه کردن مجموع زمان‌های تکمیل است. الگوریتم چندجمله‌ای برای مدل اول معرفی شد و برای مدل دوم نیز قاعده کلاسیک کوتاه‌ترین زمان پردازش مطرح شد. ژائو و همکاران [۲] مسئله دوماشینه را با هدف کمینه کردن مجموع وزنی زمان‌های تکمیل و با این فرض که تنها یکی از ماشین‌ها در بازه‌های مشخصی از زمان در دسترس نیست، بررسی کردند. آن‌ها پس از اینکه نشان دادند این مسئله NP-hard است، روشی تقریبی را برای حل آن ارائه کردند و سپس نتایج را برای حالات بیش از دو ماشین گسترش دادند. تان و همکاران [۳] مسئله دوماشینه را با هدف کمینه کردن مجموع زمان‌های تکمیل و با این فرض که هر یک از ماشین‌ها در بازه‌های مشخصی از زمان در دسترس نیستند، بررسی کردند. آن‌ها نشان دادند قاعده کوتاه‌ترین زمان پردازش برای حالتی که یکی از ماشین‌ها یا هر یک از ماشین‌ها در بازه‌ای از زمان از دسترس خارج شوند، به‌طوری‌که این بازه‌ها هم‌پوشانی نداشته باشند، منجر به جوابی خواهد شد که انحراف آن از جواب بهینه در بدترین حالت مشخص است. ژو و یانگ [۴] مسئله دوماشینه را با هدف کمینه کردن دامنه عملیات و با این فرض که یکی از ماشین‌ها به‌صورت دوره‌ای از دسترس خارج شده و ماشین دیگر همواره در دسترس است، بررسی کردند. وانگ و چنگ [۵] مسئله دوماشینه را با هدف بیشینه کردن تعداد کارهایی که پردازش آن‌ها بدون تأخیر پایان می‌یابد، بررسی کردند و با این فرض که یکی از ماشین‌ها در بازه زمانی مشخص در دسترس نیست، روشی ابتکاری برای حل آن پیشنهاد کردند. لی و کیم [۶] مسئله دوماشینه را با هدف کمینه کردن مجموع دیرکرد^۳ کارها و با این فرض که هر فعالیت‌های نگهداری و تعمیرات از دسترس خارج شود، بررسی کردند و به ارائه الگوریتم شاخه و کران برای حل آن پرداختند. هی و همکاران [۷] نیز مسئله دوماشینه را با هدف کمینه کردن دامنه عملیات بررسی کردند. همچنین با این فرض که هر یک از دو ماشین به‌صورت دوره‌ای در بازه‌هایی از زمان با حداکثر فاصله مشخص از دسترس خارج می‌شوند، چندین الگوریتم ابتکاری برای حل آن ارائه کردند. لی [۸] مسئله زمان بندی ماشین‌های موازی را با هدف

پژوهش‌هایی که تاکنون به زمان‌بندی ماشین‌های موازی (یکسان و یا غیرمرتبط) با فرض وابسته‌بودن زمان‌های آماده‌سازی به توالی کارها و محدودیت قابلیت پردازش ماشین‌ها پرداخته‌اند، فرض شده است که ماشین‌ها در تمام افق برنامه‌ریزی در دسترس هستند. این فرض ممکن است در مواردی توجیه‌پذیر باشد، اما در بسیاری از موقعیت‌های عملی در دنیای واقعی توجیه‌پذیر نیست. فعالیت یک ماشین در دوره‌هایی از زمان به‌دلیل خرابی‌های اتفاقی، تعمیرات پیشگیرانه، تعمیرات دوره‌ای، کارهای از قبل برنامه‌ریزی‌شده یا دلایل دیگر باید قطع شود؛ بنابراین بررسی مدل‌هایی که کارها را با توجه به محدودیت دسترسی به ماشین‌ها زمان‌بندی می‌کنند، اهمیت دارد. از سوی دیگر، در این پژوهش علاوه بر محدودیت‌های ذکرشده، کمینه‌کردن مجموع زودکردها و دیرکردها، که امروزه در مسائل زمان‌بندی کاربرد وسیعی یافته است و تابع هدف بی‌قاعده^۷ به‌شمار می‌آید، بررسی می‌شود؛ اجرای موفقیت‌آمیز مفهوم تولید به‌موقع در مدیریت تولید و موجودی واحدهای تولیدی ایجاب می‌کند پردازش کارها تا حد امکان نزدیک به موعد تحویلشان پایان یابد. مسئله مورد بررسی ابتدا به‌صورت مدل برنامه‌ریزی خطی عدد صحیح مختلط ارائه می‌شود. سپس، به‌منظور حل مسائل با ابعاد بزرگ، الگوریتم رقابت استعماری^۸ که از رویکردی حریصانه^۹ برای رمزگشایی بهره می‌برد، توسعه می‌یابد.

در بخش بعد، مسئله مدنظر تشریح و به‌صورت مدل ریاضی ارائه می‌شود. در بخش سوم، الگوریتم رقابت استعماری برای حل مسئله در ابعاد بزرگ توسعه قابل مشاهده است. عملکرد الگوریتم پیشنهادی نیز در بخش چهارم ارزیابی می‌شود. در بخش پایانی نیز جمع‌بندی و ارائه پیشنهادهایی مفید برای مطالعات آتی صورت می‌گیرد.

بیان مسئله و ارائه مدل ریاضی

مسئله مورد نظر به این صورت است: تعداد n کار که همه در زمان صفر در دسترس هستند، باید به‌کمک m ماشین موازی غیرمرتبط پردازش شوند. در واقع، ماشین‌ها سرعت‌های متفاوتی دارند؛ به‌گونه‌ای که یک ماشین ممکن است کاری را سریع‌تر و کار دیگری را کندتر از سایر

آن‌ها یک الگوریتم شاخه و کران و یک الگوریتم ژنتیک را برای حل مسئله ارائه کردند. به‌تازگی نیز یین و همکاران [۱۵] به بررسی مسئله با هدف کمینه‌کردن مجموع زمان‌های تکمیل مورد انتظار پرداختند و روش‌های تقریبی حل را توسعه دادند؛ با این فرض که بعضی ماشین‌ها ممکن است در بازه‌ای از زمان با احتمال مشخصی از دسترس خارج شوند و مدت‌زمان پردازش هر کار با تعویق پردازش افزایش یابد.

در حوزه زمان‌بندی ماشین‌های موازی غیرمرتبط^۵ پژوهش‌های اندکی با فرض محدودیت دسترسی به ماشین‌ها انجام شده است. سورش و قادهوری [۱۶] مسئله کمینه‌کردن دامنه عملیات را درحالی‌که ماشین‌ها در بازه‌هایی از زمان از دسترس خارج می‌شوند، بررسی کردند و یک الگوریتم ژنتیک برای حل آن ارائه دادند. بایزیدی [۱۷] این مسئله را با این فرض بررسی کرد که کارها در زمان‌های گوناگون در دسترس قرار می‌گیرند، زمان‌های آماده‌سازی وابسته به توالی است و پردازش یک کار می‌تواند متوقف شود و پس از در دسترس قرارگرفتن ماشین ادامه یابد. وی یک مدل برنامه‌ریزی عدد صحیح مختلط و یک الگوریتم جست‌وجوی ممنوعه را برای حل مسئله پیشنهاد کرد. همچنین جیانگ و تان [۱۸] مسئله را در حالتی که ماشین‌ها از ابتدای برنامه در دسترس نیستند، در زمان‌های مختلفی در دسترس قرار می‌گیرند و امکان رد سفارش تولید هر کار میسر است، درنظر گرفتند و الگوریتمی ابتکاری برای کمینه‌کردن مجموع دامنه عملیات و هزینه‌های رد یا پردازش کارها ارائه دادند که جواب آن از دو برابر جواب بهینه بدتر نخواهد بود.

در این مقاله، مسئله زمان‌بندی ماشین‌های موازی غیرمرتبط با درنظرگرفتن زمان‌های آماده‌سازی وابسته به توالی و محدودیت‌های پردازش بررسی شد. منظور از محدودیت‌های پردازش، محدودیت دسترسی به ماشین‌ها و قابلیت هریک از آن‌ها در پردازش مجموعه مشخصی از کارهاست. در مدل مورد بررسی، تخصیص کارها به ماشین‌ها و تعیین زمان‌های شروع پردازش کارهای تخصیص‌یافته به هر ماشین به‌گونه‌ای انجام می‌شود که مجموع زودکردها و دیرکردها^۶ کمینه شود. ضرورت انجام این پژوهش از آنجا ناشی می‌شود که در بیشتر

پارامترها

P_{ij} : زمان پردازش کار j بر روی ماشین i

d_j : موعد تحویل کار j

s_{jki} : زمان لازم برای آماده‌سازی ماشین i برای

پردازش کار k اگر بعد از کار j پردازش شود.

s_{0ji} : زمان لازم برای آماده‌سازی ماشین i برای

پردازش کار j اگر به‌عنوان اولین کار با این ماشین پردازش شود.

u_{ij} : برابر ۱ است؛ اگر ماشین i قابلیت پردازش کار j را

داشته باشد. در غیر این صورت صفر است

$[0, R_i]$: بازه اولیه عدم دسترسی به ماشین i (در واقع،

پس از سپری شدن R_i واحد زمانی از ابتدای برنامه، این ماشین در دسترس قرار می‌گیرد).

$[SM_i, FM_i]$: بازه عدم دسترسی به ماشین i پس از

دسترسی اولیه ($FM_i > SM_i > R_i$)

L : یک عدد بزرگ

متغیرهای تصمیم

x_j : زمان آغاز آماده‌سازی ماشین برای پردازش کار j

C_j : زمان تکمیل کار j

E_j : زودکرد کار j ($E_j = \max\{0, d_j - C_j\}$)

T_j : دیرکرد کار j ($T_j = \max\{0, C_j - d_j\}$)

A_{ij} : برابر ۱ است، اگر کار j به ماشین i تخصیص یابد.

در غیر این صورت صفر است

Z_{jki} : برابر ۱ است، اگر کار k بعد از کار j به‌کمک

ماشین i پردازش شود. در غیر این صورت صفر است

Z_{0ji} : برابر ۱ است، اگر کار j به‌عنوان اولین کار توسط

ماشین i پردازش شود. در غیر این صورت صفر است.

y_j : برابر ۱ است، اگر کار j بعد از دومین بازه عدم

دسترسی یک ماشین پردازش شود. در غیر این صورت (اگر

بین دو بازه عدم دسترسی پردازش شود) برابر صفر است.

N_{jki} : یک متغیر صفر و ۱ که به‌منظور خطی‌سازی

مدل، جایگزین عبارت غیرخطی $A_{ij}A_{ik}$ شده است.

مدل ریاضی مسئله بررسی شده به‌صورت زیر ارائه می‌شود:

$$\text{Min} \sum_{j=1}^n (E_j + T_j) \quad (1)$$

s.t.

ماشین‌ها پردازش کند. هر کار باید به‌کمک یکی از ماشین‌ها پردازش شود؛ به‌طوری‌که در موعدهی ازپیش تعیین‌شده (مستقل از دیگر کارها) تکمیل شود. در غیر این صورت، جریمه زودکرد یا دیرکرد تحمیل می‌شود. همچنین هر ماشین در یک زمان می‌تواند حداکثر یک کار را پردازش کند. سایر فرضیه‌ها نیز به شرح زیر است:

- زمان لازم برای آماده‌سازی یک ماشین برای پردازش یک کار به کار پردازش‌شده قبلی و نیز نوع ماشین وابسته است.

- هر یک از ماشین‌ها قابلیت پردازش مجموعه مشخصی از کارها را دارد (و نه لزوماً همه آن‌ها).

- ماشین‌ها از ابتدای برنامه در دسترس نیستند، بلکه در زمان‌های گوناگونی در دسترس قرار می‌گیرند (برای مثال، به‌دلیل درگیربودن در برنامه تولید قبلی). همچنین، هر یک از آن‌ها در بازه زمانی مشخص از دسترس خارج می‌شوند (برای مثال، به‌منظور انجام فعالیت‌های نگهداری و تعمیرات برنامه‌ریزی‌شده).

- ماهیت کارها به‌گونه‌ای است که وقفه^۱ در پردازش آن‌ها مجاز نیست. به عبارت دیگر، چنانچه پردازش یک کار به‌منظور پردازش کاری دیگر، یا به‌دلیل فرارسیدن بازه عدم دسترسی به ماشین قطع شود، پردازش مجدد مستلزم تکرار فرایند صورت می‌گیرد (و نه ادامه از نقطه توقف)؛ بنابراین پردازش هر کار باید زمانی شروع شود که بتوان تا تکمیل‌شدن آن از ماشین استفاده کرد.

با توجه به فرضیه‌های فوق، تخصیص کارها به ماشین‌ها و تعیین زمان‌های شروع پردازش کارهای تخصیص‌یافته به هر ماشین باید به‌گونه‌ای صورت بگیرد که مجموع زودکردها و دیرکردهای کارها کمینه شود. در ادامه، مسئله در قالب مدل برنامه‌ریزی خطی عدد صحیح مختلط ارائه می‌شود. نمادگذاری استفاده‌شده در مدل ریاضی به شرح زیر است:

اندیس‌ها

j و k : اندیس‌های مربوط به کارها ($j, k = 1, \dots, n$)

i : اندیس مربوط به ماشین‌ها ($i = 1, \dots, m$)

آماده‌سازی و پردازش آن کار فاصله داشته باشد. محدودیت‌های ۱۰ تا ۱۲ تضمین می‌کنند که کارها در بازه‌هایی از زمان که ماشین‌ها در دسترس هستند پردازش شوند. محدودیت ۱۳ توالی کارهای تخصیص یافته به یک ماشین را تضمین می‌کند. محدودیت‌های ۱۴ تا ۱۶ زودکرد و دیرکرد کارها را تعیین می‌کنند (واضح است که همواره $E_j T_j = 0$). همچنین، محدودیت ۱۷ متغیرهای صفر و ۱ مدل را تعریف می‌کند.

در مدل ارائه شده، فرض بر این است که $SM_i = FM_i$ و $R_i = 0$ اما $0 < R_i < SM_i < FM_i$ بیانگر حالتی هستند که ماشین i در طول برنامه‌ریزی همواره در دسترس باشد. $0 < SM_i < FM_i$ و $R_i = 0$ حالتی را نشان می‌دهند که ماشین i از ابتدای برنامه در دسترس بوده است و تنها در بازه $[SM_i, FM_i]$ از دسترس خارج می‌شود. همچنین $R_i > 0$ و $SM_i = FM_i$ حالتی را نشان می‌دهند که ماشین i فقط در بازه $[0, R_i]$ در دسترس نیست.

الگوریتم رقابت استعماری

به دلیل NP-hard بودن مسئله مورد نظر، الگوریتم رقابت استعماری به منظور حل مسائل با ابعاد بزرگ توسعه داده می‌شود. این الگوریتم یکی از روش‌های فراابتکاری است که در سال ۲۰۰۷ آتش‌پزگرگری و لوکاس [۱۹] برای حل مسائل بهینه‌سازی پیوسته ارائه شد و به تدریج پژوهشگران مختلف برای حل مسائل بهینه‌سازی ترکیباتی توسعه دادند (برای مثال، مراجع [۲۴-۲۰]).

الگوریتم رقابت استعماری با جمعیت اولیه‌ای از جواب‌ها که هر کدام از آن‌ها یک کشور نامیده می‌شوند، شروع می‌شود. تعدادی از بهترین کشورها به عنوان استعمارگر انتخاب می‌شوند و باقیمانده جمعیت نیز به عنوان مستعمرات به این استعمارگرها تخصیص می‌یابند. قدرت کل هر امپراتوری به هر دو بخش تشکیل دهنده آن، یعنی استعمارگر (به منزله هسته مرکزی) و مستعمرات آن بستگی دارد. هر استعمارگر به تدریج سعی خواهد کرد مستعمراتش را به خود جذب کند که این کار سبب هدایت جست‌وجو به ناحیه‌های خوب فضای جواب خواهد شد. همچنین وقوع انقلاب در یک مستعمره می‌تواند سبب

$$\sum_{i=1}^m u_{ij} A_{ij} = 1; \quad \forall j \quad (2)$$

$$\sum_{\substack{j=1 \\ j \neq k}}^n Z_{jki} + Z_{0ki} = A_{ik}; \quad \forall i, k \quad (3)$$

$$\sum_{\substack{j=1 \\ j \neq k}}^n Z_{kji} \leq A_{ik}; \quad \forall i, k \quad (4)$$

$$\sum_{j=1}^n Z_{0ji} \leq 1; \quad \forall i \quad (5)$$

$$Z_{jki} \leq N_{jki}; \quad \forall i, j, k; k \neq j \quad (6)$$

$$N_{jki} \leq \frac{A_{ij} + A_{ik}}{2}; \quad \forall i, j, k; k \neq j \quad (7)$$

$$A_{ij} + A_{ik} - 2N_{jki} \leq 1; \quad \forall i, j, k; k \neq j \quad (8)$$

$$C_j \geq x_j + \sum_{i=1}^m \sum_{\substack{k=1 \\ k \neq j}}^n Z_{kji} S_{kji} + \sum_{i=1}^m Z_{0ji} S_{0ji} + \sum_{i=1}^m A_{ij} P_{ij}; \quad \forall j \quad (9)$$

$$x_j \geq \sum_{i=1}^m A_{ij} R_i; \quad \forall j \quad (10)$$

$$C_j \leq \sum_{i=1}^m A_{ij} SM_i + Ly_j; \quad \forall j \quad (11)$$

$$x_j \geq \sum_{i=1}^m A_{ij} FM_i - L(1 - y_j); \quad \forall j \quad (12)$$

$$C_j \leq x_k + L(1 - \sum_{i=1}^m Z_{jki}); \quad \forall j, k; k \neq j \quad (13)$$

$$E_j \geq d_j - C_j; \quad \forall j \quad (14)$$

$$T_j \geq C_j - d_j; \quad \forall j \quad (15)$$

$$E_j, T_j \geq 0; \quad \forall j \quad (16)$$

$$A_{ij}, Z_{jki}, Z_{0ji}, y_j, N_{jki} \in \{0, 1\}; \quad \forall i, j, k; k \neq j \quad (17)$$

رابطه ۱ تابع هدف مسئله است که مجموع زودکردها و دیرکردهای کارها را کمینه می‌کند. براساس محدودیت ۲ تضمین می‌شود که هر کار دقیقاً به یکی از ماشین‌هایی که قابلیت پردازش آن را دارند تخصیص می‌یابد. محدودیت‌های ۳ تا ۸ تضمین می‌کنند که هر یک از کارهای تخصیص یافته به یک ماشین، تنها یک بار در توالی کارهای آن ماشین ظاهر می‌شوند. محدودیت‌های ۶ تا ۸ جایگزین محدودیت غیرخطی $Z_{jki} \leq A_{ij} A_{ik}$ شده‌اند. محدودیت ۹ نشان می‌دهد زمان تکمیل هر کار باید از زمان آغاز آماده‌سازی ماشین به اندازه مجموع زمان‌های

به منظور محاسبه مقدار تابع هدف یک جواب، ابتدا باید آن را رمزگشایی کرد. به عبارت دیگر، باید جایگشت مربوط را به برنامه زمان بندی نگاشت کرد. در ادبیات مرتبط با مسائل زمان بندی ماشین های موازی، رویکردهای مختلفی برای رمزگشایی ارائه می شود که حریصانه است و کارها را با در نظر گرفتن تابع هدف زمان بندی می کند. در این پژوهش نیز رویکرد حریصانه ای برای رمزگشایی توسعه داده می شود که روند موجه سازی جوابها را نیز در بر می گیرد. رویکرد ارائه شده به این صورت است که با شروع از سمت چپ جایگشت، هر کار از میان ماشین هایی که می توانند آن را پردازش کنند و با توجه به زمان های آماده سازی و محدودیت های دسترسی، به ماشینی تخصیص داده می شود که بتواند پردازش آن را تا حد امکان نزدیک به موعد تحویلش تکمیل کند (درواقع، منجر به کمترین زودکرد یا دیرکرد آن کار شود)، این روند تا زمان بندی همه کارها ادامه می یابد. برای تشریح بهتر رویکرد رمزگشایی، یک مثال با ۴ کار و ۲ ماشین را در نظر بگیرید. فرض کنید کار ۱ تنها به کمک ماشین ۱ و بقیه کارها با استفاده از هر دو ماشین قابل پردازش هستند. موعد تحویل کارها نیز به ترتیب ۰، ۱، ۲، ۱۱، ۱۷ و ۲۴ است، اما ماشین ۱ در بازه های [۰، ۱] و [۱۲، ۱۵] و ماشین ۲ در بازه های [۰، ۳] و [۱۸، ۲۱] در دسترس نیستند. زمان های پردازش و آماده سازی نیز در جدول ۱ و ۲ ارائه شده است. در جدول ۲، زمان های s_{0ji} در سطر کار '۰' آمده است.

جواب رمزگذاری شده در شکل ۱ را در نظر بگیرید. ابتدا کار ۲ که در موقعیت اول قرار دارد، به ماشین ۱ تخصیص داده می شود؛ به طوری که در زمان ۱۲ تکمیل شود. ماشین ۲ می تواند این کار را در زمان ۱۸ تکمیل کند. سپس، کار ۳ به ماشین ۲ به گونه ای تخصیص داده می شود که به موقع تکمیل شود؛ ماشین ۱ این کار را با تأخیر می تواند تکمیل کند. با ادامه این روند، بقیه کارها نیز آن گونه که در شکل ۲ آمده است زمان بندی می شوند. مقدار تابع هدف این زمان بندی ۱۴ است. همان گونه که در این مثال نیز نشان داده شد، رویکرد ارائه شده برای رمزگشایی می تواند هر جایگشت مفروض را از میان برنامه های متعدد، به یک زمان بندی خیلی خوب که همواره موجه خواهد بود نگاشت کند.

تغییراتی در آن شود که این امر منجر به جست و جوی ناحیه های جدید فضای جواب می شود. به مرور زمان، چنانچه یک مستعمره به وضعیت بهتری (با توجه به تابع هدف مسئله) از استعمارگرش دست یابد، جای آن را خواهد گرفت.

پس از شکل گیری امپراتوری های اولیه، رقابت استعماری میان آن ها برای تملک مستعمرات یکدیگر شروع می شود. به این صورت که در هر تکرار الگوریتم، رقابتی میان امپراتوری ها برای تصاحب ضعیف ترین مستعمره ضعیف ترین امپراتوری شکل می گیرد. هر امپراتوری که نتواند بر قدرت خود بیفزاید، در جریان رقابت به تدریج مستعمراتش را از دست می دهد و در نهایت حذف خواهد شد. این فرایند تا زمانی که همه امپراتوری ها سقوط می کند و تنها یک امپراتوری که دیگر کشورها را تحت کنترل خود دارد باقی بماند، ادامه می یابد. از آنجا که رسیدن به چنین وضعیتی ممکن است بسیار زمان بر باشد، حد بالایی برای تعداد تکرارهای الگوریتم ارائه شده به عنوان شرط توقف منظور می شود. در صورتی که تکرارها به مقدار مشخصی برسد، حتی در صورت وجود بیش از یک امپراتوری، الگوریتم خاتمه داده می شود. در ادامه این بخش، مؤلفه های مختلف الگوریتم ارائه شده بررسی می شوند.

نمایش جواب

یکی از مهم ترین مراحل طراحی الگوریتم فراابتکاری که تأثیر فراوانی بر عملکرد آن خواهد داشت، رمزگذاری هر جواب مسئله در محیط الگوریتم و رمزگشایی آن است که به اختصار نمایش جواب^{۱۱} نام دارد. در واقع، شیوه ای که برای نمایش جواب استفاده می شود، تعیین کننده فضای است که باید به کمک الگوریتم جست و جو شود. برای نمایش هر جواب مسئله بررسی شده در محیط الگوریتم پیشنهادی از یک جایگشت از اعداد ۱ تا n استفاده می شود. در این شیوه، الگوریتم باید در فضایی شامل $n!$ جواب جست و جو کند؛ فضایی که تعداد ماشین ها در حجم آن تأثیری نخواهد گذاشت. در شکل ۱، جوابی برای یک مسئله با ۴ کار آمده است.

۲	۳	۱	۴
---	---	---	---

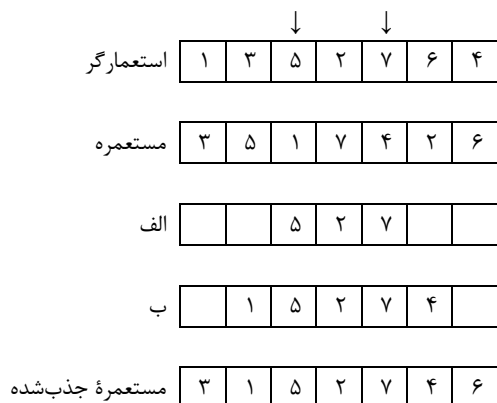
شکل ۱. جواب رمزگذاری شده مسئله به همراه ۴ کار

نسبی هر استعمارگر به صورت زیر محاسبه شده است و کشورهای مستعمره بر مبنای آن میان استعمارگرها توزیع می‌شوند.

$$pw(imp) = \frac{f'(imp)}{\sum_{\tau=1}^{N_{imp}} f'(\tau)} \quad (19)$$

عملگر جذب

براساس آنچه سیاست جذب^{۱۲} نامیده می‌شود، هر استعمارگر مستعمراتش را از ابعاد گوناگون اجتماعی و سیاسی به سمت خود متمایل می‌کند تا به تدریج سبب تکامل آن‌ها شود. در الگوریتم ارائه شده، برای اجرای سیاست جذب از عملگر مطابقت جزئی^{۱۳} استفاده می‌شود. به منظور تشریح این عملگر، مثالی با ۷ کار را در نظر بگیرید. ابتدا دو خانه به تصادف انتخاب و اعداد میان آن‌ها از استعمارگر در مستعمره جذب شده کپی می‌شود (شکل ۳-الف). سپس اعدادی که در مستعمره اولیه میان دو خانه منتخب وجود داشته‌اند و کپی نشده‌اند، پس از تطبیق دادن با اعدادی که محل آن‌ها را اشغال کرده‌اند، در مستعمره جذب شده کپی می‌شوند. با توجه به شکل ۳-ب، عدد ۱ در محل عدد ۵ و عدد ۴ در محل عدد ۲ کپی شده است. در نهایت، بقیه اعداد مستعمره جذب شده مانند مستعمره اولیه تخصیص می‌یابند.



شکل ۳. فرایند جذب

عملگر انقلاب

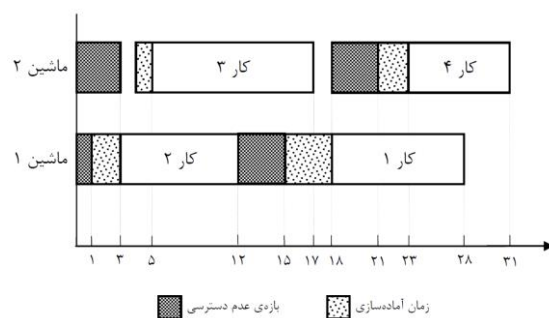
در برخی مواقع، بروز انقلاب اجتماعی-سیاسی می‌تواند به طور ناگهانی سبب تغییر ویژگی‌های یک کشور شود. در

جدول ۱. زمان‌های پردازش

ماشین		
کار	۱	۲
۱	۱۰	-
۲	۹	۱۲
۳	۸	۱۲
۴	۱۰	۸

جدول ۲. زمان‌های آماده‌سازی

کار	ماشین ۱			ماشین ۲		
	۱	۲	۳	۴	۳	۲
۰	۱	۲	۳	۲	۱	۳
۱	-	۲	۱	۳	-	-
۲	۳	-	۳	۲	۳	۲
۳	۱	۲	-	۲	-	۳
۴	۲	۱	۱	-	۳	۱



شکل ۲. زمان‌بندی کارها برای مثال ارائه شده

شکل‌دهی امپراتوری‌های اولیه

پس از آنکه جمعیت اولیه به صورت تصادفی تولید شد، تعداد N_{imp} از بهترین اعضا به عنوان استعمارگر انتخاب می‌شوند. به منظور تخصیص بقیه اعضا به استعمارگرها، ابتدا مقدار نرمال شده تابع هدف هر استعمارگر به صورت زیر محاسبه می‌شود:

$$f'(imp) = f_{\max} - f(imp) \quad (18)$$

که در آن $f(imp)$ مقدار تابع هدف استعمارگر imp و f_{\max} بیشترین مقدار تابع هدف میان استعمارگرهاست. باید توجه داشت که نرمال‌سازی به دلیل آنکه کمینه کردن تابع هدف مسئله مورد نظر است، انجام می‌شود. اکنون کمینه کردن $f(imp)$ معادل بیشینه کردن $f'(imp)$ است. سپس قدرت

امپراتوری‌های قوی‌تر با شانس بیشتری به رقابت خواهند پرداخت.

نتایج محاسباتی

از آنجا که مسئله تعریف شده در این مقاله برای اولین بار بررسی می‌شود، مسائل عددی استاندارد^{۱۴} در ادبیات موضوع مشاهده نمی‌شود؛ بنابراین به منظور ارزیابی عملکرد مدل ریاضی و الگوریتم پیشنهادی، تعدادی مسئله عددی در اندازه‌های گوناگون به طور تصادفی تولید شده‌اند. مشابه با مرجع [۲۵]، زمان‌های پردازش و آماده‌سازی به ترتیب از توزیع‌های یکنواخت گسسته در بازه‌های [۰، ۳۰] و [۰، ۱]، [۷] و همچنین u_{ij} ها به تصادف از میان مقادیر ۰ و ۱ طوری انتخاب می‌شوند که هر کار حداقل با نیمی از ماشین‌ها قابل پردازش باشد. موعدهای تحویل کارها نیز مشابه با مرجع [۲۶] از توزیع یکنواخت گسسته در بازه زیر تولید می‌شوند:

$$\left[\frac{Sum^p}{2} \left(1 - Fac_1 - \frac{Fac_2}{2} \right), \frac{Sum^p}{2} \left(1 - Fac_1 + \frac{Fac_2}{2} \right) \right] \quad (۲۳)$$

مقادیر Fac_1 و Fac_2 در همه مسائل به ترتیب ۰/۵ و ۰/۱ فرض شده‌اند؛ همچنین، Sum^p با توجه به زمان‌های پردازش کارها به صورت زیر محاسبه می‌شود:

$$Sum^p = \sum_j \bar{p}_j \quad \& \quad \bar{p}_j = \frac{\sum_i P_{ij}}{m} \quad (۲۴)$$

همچنین، زمان‌های R_i و SM_i به ترتیب از توزیع‌های یکنواخت گسسته در بازه‌های [۰، ۱۰] و [۰، ۲۰۰] و طول بازه $FM_i - SM_i$ از توزیع یکنواخت گسسته در بازه [۰، ۱۵۰] انتخاب می‌شوند. به ازای هر بعد از مسئله که با تعداد کارها و ماشین‌ها مشخص می‌شود، پنج مسئله نمونه تولید شده است.

الگوریتم رقابت استعماری پیشنهادی به کمک زبان برنامه‌نویسی C# کد شده و با لپ‌تاپ ۲ هسته‌ای با پردازشگر مرکزی ۲/۶ گیگاهرتز و حافظه ۴ گیگابایت اجرا شده است. از مدل ریاضی ارائه شده نیز با نرم‌افزار GAMS و حل‌کننده CPLEX استفاده شده است. در این نرم‌افزار، حداکثر زمان حل ۷۲۰۰ ثانیه در نظر گرفته و پس از این مدت، بهترین جواب به دست آمده گزارش شد.

به منظور تنظیم پارامترهای الگوریتم رقابت استعماری،

الگوریتم رقابت استعماری، انقلاب با جابه‌جایی یک مستعمره به یک موقعیت تصادفی جدید مدل‌سازی می‌شود و به دلیل تنوع در مسیر جست‌وجو، از گیرافتادن در دام بهینه‌های محلی پیشگیری می‌کند. بدیهی است در برخی موارد نیز سبب بهبود موقعیت یک کشور می‌شود و آن را به محدوده بهتری می‌برد. در الگوریتم ارائه شده پس از فرایند جذب، با احتمال P_{rev} (که نرخ انقلاب نامیده می‌شود) در هر مستعمره انقلاب رخ می‌دهد. عملگر انقلاب به این صورت است که پس از انتخاب دو خانه به صورت تصادفی کارهای آن‌ها با یکدیگر جابه‌جا می‌شوند.

رقابت استعماری

امپراتوری‌ها برای تملک مستعمرات یکدیگر و افزایش قدرت خود به رقابت می‌پردازند. رقابت استعماری به تدریج سبب افزایش قدرت امپراتوری‌های قوی‌تر و کاهش قدرت ضعیف‌ترها می‌شود. چنانچه یک امپراتوری نتواند بر قدرت خود بیفزاید، به مرور از جریان رقابت حذف خواهد شد. به منظور رقابت در هر تکرار الگوریتم، قدرت کل هر امپراتوری متناسب با قدرت استعمارگر و مستعمرات آن به صورت زیر محاسبه می‌شود:

$$Tf (imp) = f (imp) + \alpha f_{avg}^{col} (imp) \quad (۲۰)$$

که در آن $f_{avg}^{col} (imp)$ میانگین مقادیر تابع هدف مستعمرات امپراتوری imp و α نیز عددی بین صفر و ۱ است. از آنجا که در بیشتر اجراها، α برابر با ۰/۰۵ به نتایج مطلوبی منجر شده است، در این پژوهش از همین مقدار استفاده شده است. مقدار نرمال‌شده قدرت کل هر امپراتوری به صورت زیر محاسبه می‌شود:

$$Tf' (imp) = Tf_{max} - Tf (imp) \quad (۲۱)$$

که در آن Tf_{max} قدرت کل ضعیف‌ترین امپراتوری است. در نهایت، احتمال تصاحب ضعیف‌ترین مستعمره ضعیف‌ترین امپراتوری به کمک هر امپراتوری imp به صورت زیر محاسبه می‌شود:

$$P_{pos} (imp) = \frac{Tf' (imp)}{\sum_{\tau=1}^{N_{imp}} Tf' (\tau)} \quad (۲۲)$$

بنابراین، ضعیف‌ترین مستعمره ضعیف‌ترین امپراتوری به کمک قوی‌ترین امپراتوری تصاحب نخواهد شد، بلکه

است جواب بهینه را حداقل یک‌بار در پنج بار اجرا به‌دست آورد. براساس نتایج، با افزایش ابعاد مسئله، مدل ریاضی توان رقابت با الگوریتم پیشنهادی را از دست می‌دهد. برای چهار مسئله از مسائل با اندازه ۴۰ کار و ۷ ماشین، نرم‌افزار مورد نظر در ۷۲۰۰ ثانیه قادر به یافتن حتی یک جواب موجه نبوده است. با توجه به نتایج می‌توان گفت الگوریتم ارائه‌شده عملکرد درخور توجهی دارد و قادر است جواب‌های خوبی را در زمان پذیرفته‌به‌ویژه برای مسائل با ابعاد بزرگ به‌دست آورد.

نتیجه‌گیری و پیشنهادها

در این پژوهش، مسئله زمان‌بندی ماشین‌های موازی غیرمرتبط با در نظر گرفتن زمان‌های آماده‌سازی وابسته به توالی کارها و نیز به نوع ماشین، محدودیت دسترسی به ماشین‌ها و همچنین قابلیت هر یک از آن‌ها در پردازش مجموعه مشخصی از کارها بررسی شد. تخصیص کارها به ماشین‌ها و تعیین زمان‌های شروع پردازش کارهای تخصیص‌یافته به هر ماشین، با هدف کمینه‌کردن مجموع زودکردها و دیرکردها انجام می‌شود. مسئله ابتدا به‌صورت مدل برنامه‌ریزی خطی عدد صحیح مختلط ارائه شد. به دلیل پیچیدگی مسئله مورد مطالعه، الگوریتم فراابتکاری رقابت استعماری به‌منظور حل مسائل با اندازه بزرگ طراحی شد. براساس نتایج محاسباتی، الگوریتم پیشنهادی رویکرد مؤثری برای یافتن جواب‌های بهینه یا نزدیک به بهینه است.

همان‌گونه که پیش از این بیان شد، کمینه‌کردن مجموع زودکردها و دیرکردها، تابع هدف بی‌قاعده محسوب می‌شود. به عبارت دیگر، بیکارگذاشتن ماشین‌ها و تأخیر پردازش کارها گاهی ممکن است به بهبود این تابع منجر شود. از سوی دیگر، بیکاری ماشین‌ها مطلوب تولیدکننده نیست و ممکن است در دنیای واقعی دربردارنده هزینه باشد؛ بنابراین کمینه‌کردن بیکاری ماشین‌ها علاوه‌بر کمینه کردن مجموع زودکردها و دیرکردها می‌تواند موضوع جالبی برای پژوهش‌های آتی باشد. در نظر گرفتن مسئله مورد نظر در این پژوهش با این فرض که محدودیت دسترسی به ماشین‌ها به بازه‌های زمانی بیشتری تعمیم یابد، می‌تواند موضوعی برای پژوهش‌های آتی باشد.

مقادیر گوناگونی برای هر پارامتر مدمظر قرار گرفت. سپس با حل برخی مسائل عددی در آزمایش‌های اولیه، مقادیر مناسبی به شرح جدول ۳ برای استفاده در آزمایش‌های بعدی انتخاب شدند:

جدول ۳. پارامترهای الگوریتم رقابت استعماری

پارامتر	مقدار تنظیم‌شده
اندازه جمعیت	$10n$
تعداد امپراتوری‌های اولیه	$0.3n$
حد بالای تعداد تکرارها	۱۰۰
نرخ انقلاب	0.4

به دلیل ماهیت تصادفی الگوریتم رقابت استعماری، هر یک از مسائل تولیدشده پنج بار حل شدند و کمترین، میانگین و بیشترین مقادیر تابع هدف جواب‌های به‌دست‌آمده گزارش شدند. همچنین به‌منظور تسهیل در مقایسه نتایج، از شاخص درصد انحراف نسبی ۱۵ زیر برای سنجش مقدار تابع هدف هر جواب sol برای یک مسئله مفروض استفاده شد:

$$RPD(sol) = \left(\frac{f(sol) - f(sol_{best})}{f(sol_{best})} \right) \times 100\% \quad (25)$$

که در آن sol_{best} بهترین جواب به‌دست‌آمده برای آن مسئله از میان جواب‌های حاصل از الگوریتم و مدل ریاضی است. شاخص RPD هر جواب را با بهترین جواب برای آن مسئله مقایسه و اختلاف آن‌ها را به‌صورت درصد اعلام می‌کند. هر چه مقدار این شاخص کمتر باشد، نشان‌دهنده کیفیت بهتر جواب است.

نتایج به‌دست‌آمده از حل مسائل با اندازه کوچک (شامل مسائلی با ۷ کار و ۲ ماشین تا ۱۰ کار و ۳ ماشین) و مسائل با اندازه متوسط و بزرگ (شامل مسائلی با ۱۱ کار و ۳ ماشین تا ۴۰ کار و ۷ ماشین) در جدول ۴ آمده است. برای نمونه‌هایی که اندازه یکسان دارند، میانگین نتایج نیز گزارش شده است. همچنین، مقادیر RPD مسائلی که مدل ریاضی قادر به یافتن جواب بهینه آن‌ها بوده، پررنگ شده است.

براساس جدول ۴، برای مسائلی که جواب بهینه آن‌ها با استفاده از مدل ریاضی به دست آمده است (۱۸ از ۲۵ مسئله با اندازه کوچک)، الگوریتم رقابت استعماری توانسته

جدول ۴. نتایج محاسباتی برای مسائل تولیدشده

الگوریتم رقابت استعماری				مدل ریاضی				
زمان (ثانیه)	RPD			زمان (ثانیه)	RPD	مسئله نمونه	تعداد ماشین	تعداد کار
	بیشترین	میانگین	کمترین					
۲۵	۰	۰	۰	۳	۰	۱		
۳۲	۰	۰	۰	۵	۰	۲		
۳۵	۰	۰	۰	۹	۰	۳	۲	۷
۲۷	۰	۰	۰	۵	۰	۴		
۳۲	۰	۰	۰	۶	۰	۵		
۳۰	۰	۰	۰	۶	۰			میانگین
۴۴	۰/۲۲	۰/۰۹	۰	۱۰۶	۰	۱		
۳۹	۰/۲۰	۰/۰۴	۰	۸۹	۰	۲		
۳۷	۰	۰	۰	۶۷	۰	۳	۲	۸
۳۹	۰	۰	۰	۷۴	۰	۴		
۴۳	۰/۱۸	۰/۰۴	۰	۹۷	۰	۵		
۴۰	۰/۱۲	۰/۰۳	۰	۸۷	۰			میانگین
۵۵	۰/۱۴	۰/۰۸	۰	۶۷۶	۰	۱		
۵۸	۰/۱۹	۰/۱۱	۰	۱۹۲۱	۰	۲		
۶۳	۰/۳۲	۰/۱۶	۰	۲۴۷۳	۰	۳	۲	۹
۶۲	۰/۳۹	۰/۲۴	۰	۷۲۰۰	۰	۴		
۵۹	۰/۲۹	۰/۱۸	۰	۷۲۰۰	۰	۵		
۵۹	۰/۲۶	۰/۱۵	۰	۳۸۹۴	۰			میانگین
۶۳	۰/۲۳	۰/۱۴	۰	۱۷۱۴	۰	۱		
۶۱	۰/۳۴	۰/۲۱	۰	۸۱۹	۰	۲		
۶۱	۰/۲۶	۰/۱۶	۰	۱۳۲۲	۰	۳	۳	۹
۶۸	۰/۳۵	۰/۲۱	۰	۲۹۲۶	۰	۴		
۶۶	۰/۵۸	۰/۳۵	۰	۷۲۰۰	۰	۵		
۶۴	۰/۳۵	۰/۲۱	۰	۲۷۹۶	۰			میانگین
۷۷	۰/۳۲	۰/۱۹	۰	۳۷۶۰	۰	۱		
۸۱	۰/۳۳	۰/۲۰	۰	۷۲۰۰	۰	۲		
۸۵	۰/۳۹	۰/۲۳	۰	۷۲۰۰	۰	۳	۳	۱۰
۹۱	۰/۴۵	۰/۲۷	۰	۷۲۰۰	۰	۴		
۸۳	۰/۵۱	۰/۳۱	۰	۷۲۰۰	۰	۵		
۸۳	۰/۴۰	۰/۲۴	۰	۶۵۱۲	۰			میانگین
۱۲۴	۰/۳۹	۰/۲۹	۰	۷۲۰۰	۰	۱		
۱۳۹	۰/۴۷	۰/۲۸	۰	۷۲۰۰	۰	۲		
۱۲۲	۰/۳۶	۰/۲۴	۰	۷۲۰۰	۰	۳	۳	۱۱
۱۴۳	۰/۷۳	۰/۵۱	۰	۷۲۰۰	۰	۴		
۱۵۲	۰/۵۰	۰/۳۵	۰	۷۲۰۰	۰	۵		
۱۳۶	۰/۴۹	۰/۳۳	۰	۷۲۰۰	۰			میانگین
۲۰۷	۰/۴۴	۰/۲۰	۰	۷۲۰۰	۰/۲۹	۱		
۱۸۸	۰/۳۹	۰/۲۴	۰	۷۲۰۰	۰/۲۶	۲		
۲۱۸	۰/۶۰	۰/۳۲	۰	۷۲۰۰	۰/۲۰	۳	۳	۱۲
۱۹۸	۰/۳۰	۰/۱۸	۰	۷۲۰۰	۰/۳۰	۴		
۲۰۲	۰/۵۳	۰/۳۶	۰	۷۲۰۰	۰/۳۶	۵		
۲۰۳	۰/۴۵	۰/۲۶	۰	۷۲۰۰	۰/۲۸			میانگین

ادامه جدول ۴. نتایج محاسباتی برای مسائل تولیدشده

۲۵۳	۰/۵۲	۰/۲۹	.	۷۲۰۰	۰/۹۴	۱		
۲۴۲	۰/۸۰	۰/۳۹	.	۷۲۰۰	۰/۹۱	۲		
۲۳۱	۰/۷۵	۰/۳۸	.	۷۲۰۰	۰/۹۱	۳	۳	۱۵
۲۷۶	۰/۸۰	۰/۳۷	.	۷۲۰۰	۱/۰۶	۴		
۲۸۰	۰/۸۵	۰/۵۶	.	۷۲۰۰	۱/۱۴	۵		
۲۵۶	۰/۷۴	۰/۳۹	.	۷۲۰۰	۰/۹۹			میانگین
۳۳۳	۱/۰۰	۰/۵۳	.	۷۲۰۰	۱/۷۵	۱		
۲۷۸	۱/۱۶	۰/۶۶	.	۷۲۰۰	۲/۳۲	۲		
۳۱۴	۱/۲۳	۰/۷۰	.	۷۲۰۰	۲/۴۶	۳	۴	۱۸
۲۷۱	۱/۴۷	۰/۶۹	.	۷۲۰۰	۱/۸۹	۴		
۳۲۶	۲/۲۱	۰/۸۹	.	۷۲۰۰	۲/۳۵	۵		
۳۰۴	۱/۴۱	۰/۶۹	.	۷۲۰۰	۲/۱۵			میانگین
۳۶۴	۱/۰۶	۰/۵۹	.	۷۲۰۰	۳/۰۶	۱		
۳۴۲	۱/۲۵	۰/۵۶	.	۷۲۰۰	۲/۶۱	۲		
۳۶۰	۱/۱۴	۰/۶۳	.	۷۲۰۰	۳/۸۰	۳	۵	۲۵
۳۸۱	۱/۰۵	۰/۵۳	.	۷۲۰۰	۲/۱۸	۴		
۳۵۹	۱/۴۰	۰/۷۰	.	۷۲۰۰	۲/۷۹	۵		
۳۶۱	۱/۱۸	۰/۶۰	.	۷۲۰۰	۲/۸۸			میانگین
۴۲۷	۱/۳۳	۰/۶۵	.	۷۲۰۰	۴/۱۸	۱		
۴۱۳	۱/۶۰	۰/۷۵	.	۷۲۰۰	۴/۸۰	۲		
۴۶۱	۱/۹۴	۰/۹۸	.	۷۲۰۰	۴/۴۵	۳	۶	۳۰
۴۷۶	۱/۳۹	۰/۶۱	.	۷۲۰۰	۳/۱۳	۴		
۴۴۵	۱/۵۷	۰/۸۲	.	۷۲۰۰	۳/۷۴	۵		
۴۴۴	۱/۵۶	۰/۷۶	.	۷۲۰۰	۴/۰۶			میانگین
۵۶۶	۲/۶۷	۱/۵۱	.	۷۲۰۰	-	۱		
۵۴۹	۲/۱۴	۱/۰۶	.	۷۲۰۰	-	۲		
۵۲۱	۱/۹۳	۰/۹۹	.	۷۲۰۰	-	۳	۷	۴۰
۵۹۰	۲/۷۱	۱/۶۶	.	۷۲۰۰	-	۴		
۵۷۳	۲/۵۳	۱/۴۱	.	۷۲۰۰	۵/۱۹	۵		
۵۶۰	۲/۳۹	۱/۳۲	.	۷۲۰۰	۵/۱۹			میانگین

منابع

1. Sun, K., and Li, H., (2010). "Scheduling Problems with Multiple Maintenance Activities and Non-Preemptive Jobs on Two Identical Parallel Machines", *International Journal of Production Economics*, Vol. 124, No. 1, PP. 151–158.
2. Zhao, C., Ji, M., and Tang, H. (2011). "Parallel-Machine Scheduling with an Availability Constraint", *Computers and Industrial Engineering*, Vol. 61, No. 3, PP. 778–781.
3. Tan, Z., Chen, Y., and Zhang, A. (2013). "On the Exact Bounds of SPT for Scheduling on Parallel Machines with Availability Constraints", *International Journal of Production Economics*, Vol. 146, No. 1, PP. 293–299.
4. Xu, D., and Yang, D. L. (2013). "Makespan Minimization for Two Parallel Machines Scheduling with a Periodic Availability Constraint: Mathematical Programming Model, Average-Case Analysis, and Anomalies", *Applied Mathematical Modelling*, Vol. 37, No. 14, PP. 7561–7567.
5. Wang, X., and Cheng, T. C. E. (2015). "A Heuristic for Scheduling Jobs on Two Identical Parallel Machines with a Machine Availability Constraint", *International Journal of Production Economics*, Vol. 161, PP. 74–82.
6. Lee, J. Y., and Kim, Y. D. (2015). "A Branch and Bound Algorithm to Minimize Total Tardiness of Jobs in a

- Two Identical-Parallel-Machine Scheduling Problem with a Machine Availability Constraint”, *Journal of the Operational Research Society*, Vol. 66, No. 9, PP. 1542–1554.
7. He, J., Li, Q., and Xu, D. (2016). “Scheduling Two Parallel Machines with Machine-Dependent Availabilities”, *Computers and Operations Research*, Vol. 72, PP. 31–42.
 8. Lee, C. Y. (1991). “Parallel Machines Scheduling with Nonsimultaneous Machine Available Time”, *Discrete Applied Mathematics*, Vol. 30, No. 1, PP. 53–61.
 9. Lee, C. Y., and Chen, Z. L. (2000). “Scheduling Jobs and Maintenance Activities on Parallel Machines”, *Naval Research Logistics*, Vol. 47, No. 2, PP. 145–165.
 10. Gharbi, A., and Haouari, M. (2005). “Optimal Parallel Machines Scheduling with Availability Constraints”, *Discrete Applied Mathematics*, Vol. 148, No. 1, PP. 63–87.
 11. Sheen, G. J., Liao, L. W., and Lin, C. F. (2008). “Optimal Parallel Machines Scheduling with Machine Availability and Eligibility Constraints”, *International Journal of Advanced Manufacturing Technology*, Vol. 36, No. 1 and 2, PP. 132–139.
 12. Shen, L., Wang, D., and Wang, X. Y. (2013). “Parallel-Machine Scheduling with Non-Simultaneous Machine Available Time”, *Applied Mathematical Modelling*, Vol. 37, No. 7, PP. 5227–5232.
 13. Huo, Y., and Zhao, H. (2015). “Total Completion Time Minimization on Multiple Machines Subject to Machine Availability and Makespan Constraints”, *European Journal of Operational Research*, Vol. 243, No. 2, PP. 547–554.
 14. Lee, W. C., Wang, J. Y., and Lee, L. Y. (2015). “A Hybrid Genetic Algorithm for an Identical Parallel-Machine Problem with Maintenance Activity”, *Journal of the Operational Research Society*, Vol. 66, No. 11, PP. 1906–1918.
 15. Yin, Y., Wang, Y., Cheng, T. C. E., Liu, W., and Li, J. (2017). “Parallel-Machine Scheduling of Deteriorating Jobs with Potential Machine Disruptions”, *Omega*, Vol. 69, PP. 17–28.
 16. Suresh V., and Ghauthuri, D. (1996). “Scheduling of Unrelated Parallel Machines When Machine Availability Is Specified”, *Production Planning and Control*, Vol. 7, No. 4, PP. 393–400.
 17. Baiazidi, M. (2014). “Group Scheduling Problem on Unrelated Parallel Machines with Availability Constraints and Sequence Dependent Setup Times”, *Msc Thesis, University of Kurdistan*.
 18. Jiang, D., and Tan, J. (2016). “Scheduling with Job Rejection and Nonsimultaneous Machine Available Time on Unrelated Parallel Machines”, *Theoretical Computer Science*, Vol. 616, PP. 94–99.
 19. Atashpaz-Gargari, E., and Lucas, C. (2007). “Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition”, *IEEE Congress on Evolutionary Computation*, PP. 4661–4667.
 20. Norouzi, N., Tavakkoli-Moghaddam, R., Sadegh-Amalnick, M., and Khaefi, S. (2015). “New Mathematical Modeling for a Facilities Location and Vehicle Routing Problem Solving by a Hybrid Imperialist Competitive Algorithm”, *Journal of Industrial Engineering (University of Tehran)*, Vol. 49, No. 1, PP. 129–137.
 21. Ahmadizar, F., and Farhadi, S. (2015). “Single-Machine Batch Delivery Scheduling with Job Release Dates, Due Windows and Earliness, Tardiness, Holding and Delivery Costs”, *Computers and Operations Research*, Vol. 53, PP. 194–205.
 22. Fallah Sanami, S., Ramezani, R., and Shafiei Nikabadi, M. (2016). “Simultaneous Lot-Sizing and Scheduling in Hybrid Flow Shop Production Environment with Resource Constraint”, *Journal of Industrial Engineering (University of Tehran)*, Vol. 50, No. 2, PP. 295–310.
 23. Yazdani, M., Aleti, A., Khalili, S. M., and Jolai, F. (2017). “Optimizing the Sum of Maximum Earliness and Tardiness of the Job Shop Scheduling Problem”, *Computers and Industrial Engineering*, Vol. 107, PP. 12–24.
 24. Panahi, I., and Nahavandi, N. (2017). “An Efficient Imperialist Competitive Algorithm for Resource Constrained Project Scheduling Problem”, *Journal of Industrial Engineering (University of Tehran)*, Vol. 51, No. 2, PP. 161–174.
 25. Afzalirad, M., and Rezaeian, J. (2017). “A Realistic Variant of Bi-Objective Unrelated Parallel Machine Scheduling Problem: NSGA-II And MOACO Approaches”, *Applied Soft Computing*, Vol. 50, PP. 109–123.
 26. Radhakrishnan, S., and Ventura, J. A. (2000). “Simulated Annealing for Parallel Machine Scheduling with Earliness-Tardiness Penalties and Sequence-Dependent Set-Up Times”, *International Journal of Production Research*, Vol. 38, No. 10, PP. 2233–2252.

واژه‌های انگلیسی به ترتیب استفاده در متن

1. Flexible Flow Shop
 2. Makespan
 3. Tardiness
 4. Maximum Lateness
 5. Unrelated
 6. Total Earliness And Tardiness
 7. Non-Regular
 8. Imperialist Competitive Algorithm
 9. Greedy
 10. Preemption
 11. Solution Representation
 12. Assimilation Policy
 13. Partially Mapped Operator
 14. Benchmark Problems
 15. Relative Percentage Deviation
-