مهندسی صنایع و مدیریت شریف، (تابستان ۱۳۹۵)
دوره‌ی ۱ - ۳۲، شماره‌ی ۱/۱، ص. ۳۱-۳۹

Original Article

حل مسئله‌ی اجاره‌ی منابع با جریان‌های نقدی تنزیل شده توسط الگوریتم ژنتیک

مهران سپهری* **(دانشیار)**
دانشکده‌ی اقتصاد و مدیریت، دانشگاه صنعتی شریف
شهرام شادرخ (دانشیار)
امیر عصرزاد (کارشناس ارشد)
دانشکده‌ی مهندسی صنایع، دانشگاه صنعتی شریف

مسئله‌ی زمان‌بندی پروژه یکی از مسائل کلاسیک در مدیریت پروژه است. فعالیت‌های پروژه برای اجرایی شدن نیاز به منابع دارد. در مسئله‌ی سرمایه‌گذاری منابع[1] هدف، خرید بهینه از منابع است به‌طوری که در هر دوره منبع کافی در اختیار باشد. در مسئله‌ی دیگری که شبیه این مسئله است و اجاره‌ی منابع[2] نامیده می‌شود، فرض بر آن است که منابع مورد نیاز پروژه را اجاره می‌کنند و به این‌ترتیب هزینه‌های اجاره وابسته به زمانِ در اختیار داشتن آن‌ها می‌شود. در مسئله‌ی اجاره‌ی منابع استاندارد که سعی می‌شود هزینه‌ی در اختیار بودن منابع را کمینه کند، تابع هدف تنها شامل هزینه است و محاسبات زمانی در آن دیده نشده است. در مسئله‌یی که در این نوشتار تشریح شده است و «مسئله‌ی اجاره‌ی منابع با جریان نقدی تنزیل شده» نام دارد، تابع هدف «بیشینه‌سازی ارزش خالص فعلی پول» در نظر گرفته شده است.

sepehri@sharif.edu
shadrokh@sharif.edu
asrzad@gmail.com

واژگان کلیدی: زمان‌بندی پروژه با منابع محدود، اجاره‌ی منابع، الگوریتم ژنتیک.

۱. مقدمه

مسائل زمان‌بندی پروژه طیف بسیار زیادی از مسائل بهینه‌سازی را شامل می‌شود. در برخی از این مسائل با توجه به محدودیت منابع و روابط تقدم و تأخر به‌دنبال کمینه‌کردن زمان اتمام پروژه، و در برخی دیگر به دنبال کاهش هزینه‌های انجام پروژه هستیم. علاوه بر تفاوت در تابع هدف این نوع مسائل با تغییر نوع متغیرها و نوع محدودیت‌های مسئله می‌توان مسائل جدیدتری ساخت که هرکدام نمودی در دنیای واقعی دارند و بدین‌ترتیب رفته‌رفته مدل‌سازی مسائل را به دنیای واقعی نزدیک‌تر کرد. همچنین منابع مختلفی برای انجام فعالیت‌ها در پروژه مورد بررسی قرار می‌گیرد که موجب تقسیم‌بندی ماهیت منابع مورد استفاده در پروژه می‌شود.[1] در برخی از منابع موجود انواع مسائل مربوط به زمان‌بندی پروژه به‌لحاظ انواع توابع هدف، انواع محدودیت‌های مسئله، متغیرهای مختلف و... بررسی شده است.[2-4] در مسئله‌ی زمان‌بندی پروژه که مقالات زیادی به آن پرداخته‌اند فرض بر آن است که هزینه‌ی دسترسی منابع مستقل از زمان است.[5] درنتیجه برای انجام پروژه‌یی که به ظرفیت x واحد از یک منبع نیاز دارد، صرف‌نظر از این که این x واحد برای یک واحد زمانی استفاده می‌شود یا برای کل زمان اجرای پروژه، هزینه‌ی آن یکسان در نظر گرفته شده است.[6] بنابراین در اینجا تهیه‌ی واحدهای منابع به‌معنای خریداری آن‌هاست. برای خیلی از پروژه‌های واقعی استفاده از منابع هزینه‌های وابسته به زمان دارد، مثلاً هزینه‌ی ماشین‌های سنگین یا نیروی کار در بسیاری از پروژه‌ها با افزایش زمان استفاده از آن‌ها افزایش خواهد یافت.

بدین‌ترتیب مسئله‌ی اجاره‌ی منابع یکی از انواع مسائل شناخته شده در حوزه‌ی برنامه‌ریزی پروژه است. نوبل[3] با معرفی و مدل‌سازی این مسئله،[7] در مدل خود کمینه‌کردن هزینه‌ی در دسترس بودن منابع را به‌عنوان تابع هدف این مسئله تعریف کرده است. اما چنان‌که اشاره شد وجه تمایز این مسئله با مسائل اولیه‌ی مطرح شده در برنامه‌ریزی پروژه رعایت هزینه‌های وابسته به زمان برای منابع است و بدین‌ترتیب زمان وارد محاسبات خواهد شد. در مسائلی که هزینه‌ها تدریجی و در گذر زمان منظور می‌شود، بدون شک واردکردن محاسبات زمانی پول و نرخ بهره به مسئله، مدل را هرچه بیشتر به واقعیت نزدیک‌تر خواهد کرد.[8] لذا در نظر نگرفتن نرخ بهره در مدل‌سازی این مسئله توسط مطالعات پیشین، یکی از دلایل ورود به حل این مسئله است.

مسئله‌ی زمان‌بندی پروژه با منابع محدود از سری مسائل بسیار سخت محسوب می‌شود.[9] مسئله‌ی اجاره‌ی منابع طبق مطالعات پیشین[7] از مسائل سخت محسوب می‌شود و روش‌های حل متنوعی دارد. انواع روش‌های برنامه‌ریزی خطی، برنامه‌ریزی عدد صحیح، برنامه‌ریزی صفر و ۱، برنامه‌ریزی پویا، برنامه‌ریزی غیر خطی و روش شاخه و کران و.... از جمله روش‌های حل دقیق‌اند. اما این روش‌ها معمولاً در زمان معقول جواب بهینه را ارائه نخواهند داد و برای حل این‌گونه مسائل به الگوریتم‌های دیگری متوسل خواهیم شد.

* نویسنده مسئول
تاریخ: دریافت ۱۳۹۲/۸/۲۰، اصلاحیه ۱۳۹۳/۲/۱۲، پذیرش ۱۳۹۳/۵/۲۷.

روش‌های فرابتکاری نظیر الگوریتم‌های بهینه‌سازی تکاملی ازجمله روش‌های نادقیق برای حل این مسائل‌اند که جواب نزدیک به بهینه را ارائه می‌دهند.

ایجاد یک الگوریتم مناسب برای حل مسئله بیشتر به‌عنوان یک ابتکار مطرح می‌شود. روش الگوریتم ژنتیک پارامترهای زیادی برای تنظیم کردن دارد و از آن می‌توان به روش‌های بسیار متفاوت استفاده کرد، و لذا کاربرد آن می‌تواند انعطاف‌پذیری زیادی به این الگوریتم بدهد. این روش تنها زمانی به‌عنوان یک روش مناسب برای حل یک مسئله‌ی خاص مطرح می‌شود که نتایج آن مشاهده شده باشد و در محک با دیگرروش‌های موجود قابلیت‌های اثبات شده داشته باشد. در زمینه‌ی کاربرد الگوریتم ژنتیک در مسائل برنامه‌ریزی و تخصیص منابع نتایج خوبی تجربه شده است. برای مثال می‌توان به الگوریتم ژنتیک هارتمن[۱۰و۱۱] اشاره کرد که برای تخصیص منابع محدود ــ در حالتی که برای هر فعالیت چند روش اجرا وجود دارد ــ طرح شده است. او با استفاده از یک طرح مناسب در پارامترهای الگوریتم ژنتیک و با استفاده از یک روش جست‌وجوی موضعی به نتایجی دست یافت که در این نوع مسائل روش او را ممتاز ساخت.[۱۲]

باتوجه به تجربیات مثبت استفاده از الگوریتم ژنتیک در حل مسائل برنامه‌ریزی پروژه[۱۳و۱۴] این چالش مطرح می‌شود که با استفاده از یک الگوریتم مناسب برای مسئله‌ی اجاره‌ی منابع می‌توان گامی بزرگ در یافتن جواب بهینه برای مسائل بزرگ در این زمینه برداشت.

۲. معرفی مسئله و مدل‌سازی ریاضی

همانطورکه اشاره شد درمسئله‌ی سرمایه‌گذاری منابع، تهیه‌ی واحدهای منابع به‌معنای خریداری آن‌هاست در صورتی که برای خیلی از پروژه‌های واقعی استفاده از منابع ــ نظیر تجهیزات هزینه‌بر در پروژه‌های بزرگ صنعتی ــ دربردارنده‌ی هزینه‌های وابسته به زمان است. در نظرگرفتن هزینه‌های وابسته به زمان، شبیه‌سازی مسئله‌ی اجاره‌ی منابع را ممکن می‌سازد. به‌همین علت برای مدل‌سازی مسئله کلیتی از مسئله‌ی سرمایه‌گذاری منابع (RIP) را ارائه می‌دهیم که در آن درکنار هزینه‌های تدارکات[۴] (مستقل از زمان)، هزینه‌ی اجاره[۵] (وابسته به زمان) نیز برای منابع در نظر گرفته شده است. این مسئله‌ی موسوم به اجاره‌ی منابع[۶] اولین‌بار توسط نوبل معرفی شد. در مسئله‌ی برنامه‌ریزی پروژه با منابع محدود اولین محدودیت دسترسی کافی به منابع مورد نیاز برای هر دوره است. حال منابع تعریف شده برای مسئله را می‌توان از طریق اجاره‌ی آن‌ها در اختیار داشت.

مسئله‌ی اجاره‌ی منابع بسیار شبیه به مسئله‌ی سرمایه‌گذاری منابع است با این تفاوت که در اینجا با اجاره‌ی منابع سروکار داریم و مدت زمان استفاده از منابع اهمیت دارد، اما در مسئله‌ی سرمایه‌گذاری منابع فرض می‌شود برای منبعی که در اختیار می‌گیریم هزینه‌یی بابت نگه‌داری آن نمی‌پردازیم.

هدف از حل هر مسئله‌ی زمان‌بندی در درجه اول یافتن یک زمان‌بندی موجه برای مسئله است. اگر در زمان‌بندی شرایط کمینه و بیشینه فواصل زمانی رعایت شود یک برنامه‌ریزی موجه[۷] خواهیم داشت.[۱۵] مجموعه جواب‌های موجه مسئله را F می‌نامیم. اگر زمان‌بندی مفروض S عضو مجموعه‌ی F باشد در این صورت S را موجه می‌گوییم. محدودیت منابع که در مسئله‌ی استاندارد زمان‌بندی پروژه مهم‌ترین محدودیت است در اینجا وجود ندارد، اما محدودیت زمانی[۸] برای اتمام پروژه در زمان تعیین شده (d) وجود دارد.

هر پروژه از تعدادی فعالیت لازم‌الاجرا تشکیل شده که انجام هریک از این فعالیت‌ها مستلزم منابع است. مجموعه‌یی به‌نام R در نظر می‌گیریم و منابع پروژه را در آن تعریف می‌کنیم. مقدار مصرف هر منبع توسط فعالیت i را با r_{ik} نشان می‌دهیم و آن مقدار منبعی از نوع k است که توسط فعالیت i در دوره‌یی که فعالیت در حال انجام است مصرف می‌شود.

تا اینجا مدل‌سازی این مسئله هیچ تفاوتی با مسئله‌ی استاندارد زمان‌بندی پروژه با منابع محدود ندارد. اما تفاوت بارز این مسئله با سایر مسائل، به‌ویژه مسئله‌ی سرمایه‌گذاری منابع، این است که در این مسئله فرض پیامد استفاده از منابع دو نوع هزینه فرض می‌شود: ۱. هزینه‌ی تدارک (C_p) شامل هزینه‌ی لازم برای سفارش و دیگر هزینه‌های ثابتی که کاملاً مستقل از زمان هستند و علاوه بر اجاره‌ی منابع باید بپردازیم؛ ۲. هزینه‌ی اجاره (C_r) و آن هزینه‌یی است که پس از تدارک منبع برای اجاره، به‌ازای هر واحد زمانی که از این منبع استفاده شود باید بپردازیم ــ زمان مصرف بیشتر، هزینه‌ی بیشتر. در نظرگرفتن هزینه‌ی اجاره نقطه‌ی تمایز مسئله‌ی سرمایه‌گذاری منابع و مسئله‌ی اجاره‌ی منابع است؛ اگر این هزینه صفر شود این مسئله به یک مسئله‌ی سرمایه‌گذاری منابع تبدیل خواهد شد.

تعریف دقیق این هزینه‌ها عبارت است از: C_k^p هزینه‌ی تدارک اولیه‌ی منبع kام که وابسته به زمان نیست و هر موقع که منبع را اضافه کنیم باید این هزینه پرداخت شود. C_k^r هزینه‌ی اجاره‌ی منبع kام که وابسته به زمان است و مادامی که از منبع استفاده می‌کنیم و منبع را کم نکرده‌ایم باید آن را بپردازیم.

برای مدل‌سازی بهتر و راحت‌تر، بدون این که از کلیت مسئله کم شود فرض می‌کنیم α نماینده‌ی افزایش منبع و ω نماینده‌ی کاهش منبع است؛ یعنی در هر لحظه از زمان این دو متغیر برای هرکدام از منابع تعریف می‌شود: α_{kt} میزانی که از منبع kام در زمان t اضافه می‌شود؛ ω_{kt} میزانی که از منبع kام در زمان t کم می‌شود.

با در نظرگرفتن مفروضات بالا مدل ریاضی مسئله‌ی اجاره‌ی منابع عبارت خواهد بود از:

$$\min\Big(\sum_{k\in R} C_k^p \sum_{t=0}^{d} \alpha_{kt} + \sum_{k\in R} C_k^r \sum_{t=0}^{d} t(\omega_{kt}-\alpha_{kt})\Big)$$

s.t. :

$$S \in F \qquad (۱)$$

$$r_k(S,t) \le \sum_{\tau=0}^{t} (\alpha k\tau - \omega k\tau) \qquad (۲)$$

$$\sum_{t=0}^{d} \alpha_{kt} - \sum_{t=0}^{d} \omega_{kt} = 0 \qquad (۳)$$

$$\alpha kt, \omega kt \in \mathbb{Z} \ge 0 \qquad (۴)$$

محدودیت ۱ تضمین‌گر موجه بودن برنامه‌ریزی به دست آمده است؛ یعنی روابط تقدمی و تأخری و فواصل زمانی موجود بین فعالیت‌ها رعایت شده است.

محدودیت ۲ تأمین‌کننده‌ی کم‌ترین منبع مورد نیاز برای هر دوره است. سمت چپ این عبارت مقدار مورد نیاز از هر منبع در دوره زمانی t است و سمت راست عبارت مقدار منبع از همان نوعی است که تأمین شده و این مقدار باید حداقل به اندازه مقدار مورد نیاز در دوره‌ی مورد نظر باشد.

محدودیت ۳ نشان‌گر منابع اجاره شده است و هر منبعی که اضافه شده بالاخره در یک دوره زمانی پس داده خواهد شد.

محدودیت ۴ بیان می‌کند که متغیرهای سیاست‌گذاری اعداد صحیح‌اند و مقادیر اعشاری نمی‌گیرند.

توابع هدفی که نوبل[۷] برای مسئله‌ی اجاره‌ی منابع معرفی کرده براساس کمینه‌سازی هزینه‌ی دسترسی به منابع است و برابر است با جمع دو هزینه‌ی تدارک و اجاره‌ی منبع:

$$\sum_{k\in R} C_k^p \sum_{t=0}^{d} \alpha_{kt} + \sum_{k\in R} C_k^r \sum_{t=0}^{d} t(\omega_{kt} - \alpha_{kt}) \qquad (۵)$$

با توجه به وابستگی متغیرهای سیاست‌گذاری موجود در مسئله به زمان، باید به این نکته‌ی کلیدی توجه داشت که در صورت وارد کردن نرخ بهره و محاسبات زمانی پول در این مسئله، آن را به واقعیت نزدیک‌تر خواهیم کرد. در واقع در سیاست‌گذاری برای منابع تنها هزینه‌ی منابع نیست که باید در نظر گرفت بلکه ارزش زمانی پول را نیز باید وارد محاسبات کرد. برای در نظر گرفتن ارزش زمانی پول نیاز به ورود نرخ بهره به محاسبات داریم.

برای مدل‌سازی جریان نقدینگی پروژه نیازمند دانستن نحوه‌ی دریافت‌ها و پرداخت‌ها یا همان درآمدها و هزینه‌ها هستیم. از آنجا که بحث اصلی مسئله درمورد هزینه‌ها و مدیریت هزینه‌هاست، برای سادگی مدل فرض می‌کنیم دریافت‌ها از مدل پرداخت در آخر پروژه پیروی می‌کنند.

پروژه‌یی را با n فعالیت در نظر می‌گیریم که این فعالیت‌ها بدون هیچ وقفه[۹]یی انجام می‌شود. دو فعالیت موهومی[۱۰] صفر و $n+1$ را در نظر می‌گیریم که بیان‌گر آغاز و پایان فعالیت‌اند. برای ادامه یک شبکه به این پروژه اختصاص می‌دهیم که به‌صورت $N = \{V, E; \delta\}$ تعریف می‌شود و در آن V مجموعه‌ی گره‌ها و E مجموعه‌ی کمان‌ها و δ وزن کمان‌ها خواهد بود. هر فعالیت i با گره i شناخته می‌شود و برای کم‌ترین فاصله‌ی زمانی[۱۱] تعریف شده $\min d_{ij}$ بین شروع دو فعالیت i و j یک کمان $\{i,j\}$ با وزن $\delta = d_{ij} \min$ تعریف می‌شود. یک کمان $\{i,j\}$ با وزن $\delta = d_{ij} \max$ نیز تعریف می‌شود که نشان‌گر بیشترین فاصله‌ی زمانی[۱۲] بین دو فعالیت i و j است. همچنین d نشان‌گر زمان پایان[۱۳] این پروژه و S_i نشان‌گر زمان شروع فعالیت iام است.

بردار $S = (S_0, ..., S_{n+1})$ با شرط $S_0 = 0$ و $S_i \geq 0$ یک زمان‌بندی برای این پروژه نامیده می‌شود. اگر در یک زمان‌بندی شرایط کمینه و بیشینه فواصل زمانی رعایت شده باشد آن را یک برنامه‌ریزی موجه می‌خوانند. برای آسان‌تر شدن مدل و بدون کاستن از کلیات مسئله در اینجا یک متغیر دیگر علاوه بر متغیرهای تعریف شده در بخش قبل تعریف می‌کنیم. متغیری تجمعی که نشان دهد در هر لحظه از زمان چه مقدار منبع و از چه نوعی اجاره شده و در اختیار است. a_{kt} نشان می‌دهد در هر لحظه از زمان چه مقدار از منبع kام در اختیار است.

درمورد مسئله‌ی سرمایه‌گذاری منابع با جریان‌های نقدی تنزیل‌یافته،[۱۶] برای تنزیل هزینه‌های اجاره و تدارکات نیز عبارت $e^{i(d-t)}$ را در هر دوره زمانی در هزینه‌ها ضرب، و آن را از نظر زمانی با سایر هزینه‌ها هم‌ارزش می‌کنیم؛ در اینجا d بیشینه زمانی است که پروژه می‌تواند طی آن پایان پذیرد. با اضافه شدن این ضریب تمام هزینه‌ها به آخر پروژه تنزیل می‌شود.

برای پرداخت‌های پروژه نیز از مدل پرداخت یکجا در آخر پروژه یا همان روش LSP[۱۴] استفاده می‌کنیم. برای این منظور فرض می‌کنیم پس از پایان موفقیت‌آمیز پروژه، پرداخت P به‌صورت یکجا انجام خواهد شد.

جواب مسئله در واقع سه‌گانه (S, α, ω) است. یافتن یک زمان‌بندی بهینه و به‌تبع آن ω و α بهینه هدف نهایی حل این مسئله است. از آنجا که به‌ازای هر بردار S می‌توان ω و α بهینه‌ی مختص این زمان‌بندی را پیدا کرد، هدف اصلی مسئله یافتن هم‌زمان سه متغیر (S, α, ω) است به‌طوری که تابع هدف به‌ازای این توابع بهترین مقدار خود را بگیرد.

در ادامه مدل برنامه‌ریزی خطی مسئله‌ی اجاره‌ی منابع با هدف بیشینه‌سازی ارزش خالص فعلی ارائه می‌شود. از آنجا که در تابع هدف مسئله فقط هزینه‌ها در نظر گرفته شده و جریان نقدی کاملاً منفی است، به‌جای بیشینه‌سازی ارزش خالص فعلی جریان نقدی تنزیل شده کمینه شده است.

$$\min[(\sum_{k\in K} C_k^p \sum_{t=0}^{d} \alpha_{kt}.e^{i(d-t)} + \sum_{k\in K} C_k^r \sum_{t=0}^{d} e^{i(d-t)}.a_{kt})] \qquad (۶)$$

s.t. :

$$a_{kt} = \sum_{\tau=0}^{t} (\alpha_{k\tau} - \omega_{k\tau}), \qquad \forall k \in K \qquad (۷)$$

$$\sum_{t=0}^{d} t.x_{it} + d_{ij}^{\min} \leq \sum_{t=0}^{d} t.x_{jt}, \qquad j \in Sec_i, \;\; i = 1, ..., n+1 \qquad (۸)$$

$$\sum_{i=1}^{n} \sum_{u=t-D_i+1}^{t} r_{ik} \times x_{iu} \leq a_{kt}, \qquad t = 0, 1, ..., d, \;\; k \in K \qquad (۹)$$

$$\sum_{t=0}^{d} \alpha_{kt} - \sum_{t=0}^{d} \omega_{kt} = 0 \qquad (۱۰)$$

$$\sum_{t=0}^{d} x_{it} = 1, \qquad i = 1, ..., n+1 \qquad (۱۱)$$

$$x_{01} = 1 \qquad (۱۲)$$

$$\sum_{t=0}^{d} t.x_{(n+1)t} \leq d \qquad (۱۳)$$

$$\sum_{t=0}^{d} t.x_{it} = S_i \qquad (۱۴)$$

$$\alpha_{k\tau}, \omega_{k\tau} \in \mathbb{Z} \geq 0 \qquad (۱۵)$$

متغیرهای جدید استفاده شده در مدل عبارت‌اند از: X_{it} متغیر صفر و ۱ است که اگر فعالیت i در زمان t شروع شود برابر ۱، و در غیر اینصورت صفر است؛ S_i زمان شروع فعالیت iام؛ $d_{ij}^{\min}$ کم‌ترین فاصله‌ی زمانی بین شروع دو فعالیت که پیش‌نیاز هم هستند؛ D_i زمان لازم برای انجام فعالیت iام؛ Sec_i مجموعه فعالیت‌های پس‌نیاز فعالیت iام؛ d بیشینه زمان مجاز که پروژه می‌تواند طی آن تمام شود.

تابع هدف و محدودیت‌های مدل نیز چنین تشریح می‌شود: عبارت ۶ تابع هدف شامل هزینه‌های اجاره و تدارک منبع است که به زمان T برده شده است. در اینجا وجود نرخ بهره در مدل باعث می‌شود که هزینه‌ها به دیرترین زمان ممکن برده شود و پروژه در دیرترین زمان ممکن زمان‌بندی شود. وجود هزینه‌ی اجاره‌ی منابع باعث تسطیح منابع در زمان‌بندی خواهد شد.

محدودیت ۷ نشان‌گر منابع فراهم شده تا دوره‌ی حاضر است و برابر مقدار تجمعی منابع اضافه شده منهای منابع کم شده تا دوره‌ی مورد نظر است.

محدودیت ۸ بیان‌گر روابط تقدمی و تأخری به‌کمک یک متغیر صفر و ۱ است و فعالیت‌هایی که جزو مجموعه‌ی پس‌نیاز یک فعالیت‌اند به‌صورت رابطه‌ی SS_{ij} یا همان شروع فعالیت مورد نظر لحاظ شده است. دلیل لحاظ کردن تمامی روابط بدین صورت آن است که این روابط در مسائل استانداردی که برای حل‌شان مورد

استفاده قرار می‌گیرند به‌صورت استاندارد شده و فقط به‌صورت رابطه‌ی SS_{ij} در نظر گرفته می‌شود. در اینجا شروع دو فعالیت باید به‌اندازه‌ی $\min d_{ij}$ از هم فاصله داشته باشد.

محدودیت ۹ تأمین‌کننده‌ی حداقل منبع مورد نیاز برای هر دوره است. سمت چپ این عبارت مقدار مورد نیاز از هر منبع در دوره زمانی t، و سمت راست آن مقدار منبع از همان نوع تأمین‌شده است و این مقدار باید دست کم به‌اندازه مقدار مورد نیاز در دوره مورد نظر باشد.

محدودیت ۱۰ نشان‌گر منابع اجاره شده است و هر منبعی که اضافه شده بالاخره در یک دوره زمانی پس داده خواهد شد.

محدودیت ۱۱ هریک از فعالیت‌ها را مجاب می‌کند که در یک دوره زمانی تا انتهای پروژه شروع شوند.

محدودیت ۱۲ بیان می‌دارد که فعالیت موهومی آغازی قطعاً در زمان ۱ شروع می‌شود.

محدودیت ۱۳ نشان‌گر فعالیت موهومی پایانی است، و در واقع پایان پروژه را محدود به زمان پایان از پیش تعیین شده می‌کند.

محدودیت ۱۴ نشان‌گر رابطه‌ی بین متغیرهای صفر و ۱ گنجانده شده در مدل و شروع فعالیت است.

محدودیت ۱۵ بیان می‌کند که متغیرهای سیاست‌گذاری اعداد صحیح‌اند و مقادیر اعشاری نمی‌گیرند.

۳. الگوریتم ژنتیک

الگوریتم ژنتیک کار جست‌وجو را از یک دسته جواب اولیه برای رسیدن به جواب بهینه انجام می‌دهد.[۱۷و۱۸] هریک از جواب‌های تولیدشده کروموزوم نامیده می‌شود. اولین گام در برنامه‌نویسی به‌وسیله‌ی الگوریتم ژنتیک طراحی کروموزوم‌های جواب است.[۱۹] در کدگذاری جواب‌هایی که به مسئله‌ی زمان‌بندی پروژه شبیه‌اند می‌توان اولویت انجام کارها را به‌عنوان ژن‌های کروموزوم در نظر گرفت و سپس این اولویت‌ها را به زمان‌بندی تبدیل کرد.

کدگذاری کروموزوم‌های این مسئله به این صورت انجام شده که ابتدا اولویتی از تقدم و تأخر فعالیت‌ها تهیه و سپس برای هر فعالیت یک متغیر دیگر تعریف می‌شود که مشخص می‌کند این فعالیت درکدام یک از سه حالت زیر زمان‌بندی شده است:

۱. در زودترین زمان (ES)[۱۵] ممکن؛

۲. در دیرترین زمان (LS)[۱۶] ممکن؛

۳. بلافاصله بعد از اتمام فعالیت زمان‌بندی شده (AF)[۱۷] قبلی خود در اولویت زمانی ذکر شده.

براساس مطالعات انجام شده،[۱۵] در مسئله‌ی اجاره‌ی منابع برای یافتن جواب بهینه جست‌وجوی تمام فضای جواب لازم نیست و جواب بهینه با زمان‌بندی فعالیت‌ها در یکی از سه نقطه‌ی ذکر شده به دست می‌آید. برای مثال در شکل ۱ هر کروموزوم شامل ۵ فعالیت است که در قسمت اول اولویت‌ها و در قسمت دوم نحوه‌ی زمان‌بندی نشان داده شده است.

اولویت انجام فعالیت ها					نحوه انجام زمانبندی				
۱	۲	۳	۴	۵	ES	AF	LS	ES	AF

شکل ۱. ساختار کروموزوم‌های مسئله.

در واقع جوابی که این کروموزوم ارائه می‌دهد بدین معناست که فعالیت اول در زودترین زمان برنامه‌ریزی شروع شود و فعالیت دوم و سوم بلافاصله بعد از اتمام فعالیت اول و فعالیت چهارم در دیرترین زمان ممکن و... پس از کدگذاری کروموزوم باید برنامه‌ی زمان‌بندی مسئله از کروموزوم‌ها استخراج شود. در بسیاری از روش‌های ابتکاری برای ایجاد زمان‌بندی، یکی از روش‌های سری یا موازی کاربرد دارد.[۱۲] روش سری که در آن در هر مرحله یکی از فعالیت‌هایی که از نظر محدودیت تأخر و تقدم موجه است انتخاب، و هنگامی زمان‌بندی می‌شود که محدودیت منابع را راضی کند.

این روش‌ها زمان‌بندی‌هایی فعال[۱۸] ایجاد می‌کنند که در حالتی که محدودیت منابع نداریم، زمان‌بندی بهینه را ارائه می‌کنند. در زمان‌بندی فعال حتی اگر فعالیت‌ها را قابل شکسته شدن در نظر بگیریم نمی‌توان هیچ فعالیتی را بدون این که یک یا بیشتر از یک فعالیت دیگر را به تأخیر بیندازیم، زودتر از موعد آغاز کرد. حالت خاصی از SGS سری به SGS فهرست معروف است. فعالیت‌های پروژه را ابتدا در یک فهرست قرار می‌دهند طوری که محدودیت تقدم و تأخر در آنها رعایت شده باشد، یعنی اگر فعالیتی در جایی از فهرست قرار گرفت باید تمامی فعالیت‌های پیش‌نیاز آن جلوتر در فهرست قرار گرفته باشد. درمورد فهرست نیز خاصیت ایجاد مجموعه‌جواب‌های فعال وجود دارد. یک نوع دیگر از نحوه‌ی ایجاد زمان‌بندی SGS موازی است که در آن برخلاف SGS سری ــ که در آن در هر مرحله یک فعالیت به جلو می‌رویم ــ در هر مرحله یک پریود زمانی به جلو حرکت می‌کنیم. در هر پریود زمانی از بین فعالیت‌هایی که محدودیت تقدم و تأخر و محدودیت‌های منابع را ارضا کرده‌اند یک فعالیت انتخاب و زمان‌بندی می‌شود. SGS موازی موجد زمان‌بندی‌های غیرتأخیری است که در نبود محدودیت منابع، زمان‌بندی بهینه است. زمان‌بندی غیرتأخیری نوعی زمان‌بندی است که در صورت امکان تخصیص منابع به یک فعالیت، آن فعالیت را رها نکنیم. مجموعه زمان‌بندی‌های غیرتأخیری زیرمجموعه‌یی از مجموعه زمان‌بندی‌های فعال است ولی ممکن است جواب بهینه را شامل نشود. نحوه‌ی ایجاد برنامه از کروموزوم (یا همان روش SGS) در این مسئله روش موازی است. به این صورت که در هر مرحله یا پریود زمانی، به فعالیتی که می‌تواند برنامه‌ریزی شود زمان شروع تخصیص می‌یابد و در نهایت یک زمان‌بندی برای کل پروژه به دست می‌آید.[۲۰] اما جواب مسئله تنها زمان‌بندی نیست و باید براساس این زمان‌بندی سیاست بهینه‌ی اجاره‌ی منبع اتخاذ شود و در نهایت تابع هدف به دست آید.

دلیل این که در این کدگذاری از منابع استفاده نشده این است که براساس زمان‌بندی مطلوب فعالیت‌ها می‌توان سیاست اجاره‌ی بهینه‌ی منابع متناظر با آن زمان‌بندی را اتخاذ کرد. به این ترتیب نیازی به استفاده از منابع در کدگذاری نیست و با تولید هرکروموزوم جدید توسط الگوریتم ژنتیک، در واقع اجاره‌ی بهینه‌ی منابع نیز برای آن زمان‌بندی به دست خواهد آمد.

در تحقیق حاضر روش چرخه‌ی رولت[۲۱] به‌عنوان فرایند انتخاب والد مورد استفاده قرار گرفته است. پس از انتخاب والدین، فرایند تزویج به‌منظور خلق فرزندان جدید از طریق ترکیب والدین انجام می‌شود. هدف این فرایند تولید فرزندان بهتر از طریق ترکیب خواص والدین است. در این تحقیق دو نوع عملگر تزویج مد نظر است: ۱. تزویج تک‌نقطه‌یی مرتب[۱۹] (OX)؛ ۲. تزویج دونقطه‌یی[۲۰] مرتب (2OX).[۲۱]

عملگر جهشی که در این تحقیق در نظر گرفته می‌شود یک نوع عملگر جهش مبادله[۲۱] است[۲۱] که چنین عمل می‌کند: نقطه‌ی تصادفی p انتخاب، و اولویت موجود در مکان p با اولویت موجود در مکان $p+1$ تعویض می‌شود. اگر این تعویض محدودیت‌های پیش‌نیازی را بر هم بریزد تعویض انجام نمی‌شود و نقطه‌ی

راهکار	۱	۴	۲	۳	۵	۶	۷
نتیجه	۱	۴	۲	۳	۵	۶	۷

شکل ۲. عملگر جهش تعویض اصلاح شده.

جدول ۱. عوامل و سطوح در نظرگرفته شده برای انجام آزمایش.

عامل	سطوح
اندازه جمعیت (Pop_size)	{۶۰, ۸۰, ۱۰۰}
احتمال انجام عملگر جهش (mut_rate)	{۰/۱, ۰/۲, ۰/۳}
تعداد تکرارهای بهبودنیافته برای طرح شروع مجدد (GR)	{۳۰, ۴۰, ۵۰}
تعداد کل تکرارها	{۸۰, ۱۰۰, ۱۲۰}

تصادفی دیگری انتخاب خواهد شد. این عملگر را جهش تعویض اصلاح‌شده[۲۲] می‌نامند. در شکل ۲ مثالی برای این عملگر جهشی نشان داده شده است. در این مثال فعالیت ۵ به‌طور تصادفی انتخاب شده و فرض شده که هیچ رابطه‌ی پیش‌نیازی با فعالیت ۳ ندارد؛ درنتیجه جای این دو فعالیت با هم عوض می‌شود.

عملگر جهش دیگری که در این تحقیق به کار رفته به این صورت است که اولویتی به‌طور تصادفی انتخاب می‌شود و نحوه‌ی انجام زمان‌بندی آن (قسمت دوم کروموزوم) در صورت امکان تعویض می‌شود. برای مثال اگر قسمت دوم کروموزوم مربوط به فعالیت AF باشد آن را تبدیل به LS می‌کنیم؛ با فرض شدنی بودن کروموزوم جدید تولید شده است.

فرایند نخبه‌گرایی در الگوریتم پیشنهادی این تحقیق نیز بدین صورت است که ۱۰ درصد از بهترین جواب‌های جمعیت جاری مستقیماً به جمعیت بعدی منتقل می‌شود. پارامترهای الگوریتم ژنتیک در این تحقیق به‌کمک طراحی آزمایشات انتخاب شده است. جدول ۱ روش‌های زیادی برای طراحی آزمایش‌های تصادفی وجود دارد. بهترین و در عین حال زمان‌برترین روش، روش طرح عاملی کامل است. این روش به‌دلیل تعداد زیاد عامل‌ها و سطوح متناظر آن‌ها ناکارآمد است. برای برطرف کردن این مشکل، طرح‌های عاملی کسری مورد استفاده قرارگرفته که تعداد آزمایش‌های لازمه را به کسری از کل ترکیبات ممکنه محدود می‌کند. تاگوچی خانواده‌یی از ماتریس‌های کاهنده‌ی تعداد آزمایش‌ها را توسعه داد.[۲۲] در روش او از آرایه‌های متعامد برای مطالعه‌ی تعداد زیادی متغیر تصمیم با استفاده از تعداد کمی آزمایش استفاده می‌شود. برای این تحقیق ۴ عامل در نظرگرفته شده که سطوح آن باید مشخص و از مقدار مناسب آن استفاده شود. این عوامل عبارت‌اند از: جمعیت اولیه، تعداد تکرارهای بهبودنیافته، احتمال جهش و تعداد کل تکرارها.

مثال‌های مورد نظر برای انجام آزمایش از کتابخانه‌ی مسائل استاندارد ProGen انتخاب شده[۲۳] که در بخش بعد توضیح داده شده است. همچنین متوسط هزینه‌ی کل (تابع هدف مسئله) به‌عنوان متغیر پاسخ در نظرگرفته شده است. پس از به دست آوردن نتایج آزمایش تاگوچی مقادیر بهینه‌ی هریک از عوامل مورد نظر به دست می‌آید.

پس از تولید یک کروموزوم ــ کروموزوم زمان‌بندی فعالیت‌ها ــ باید سیاست اجاره‌ی بهینه‌ی منبع برای زمان‌بندی استخراج شده به دست آید. در ادامه برای یافتن سیاست اجاره‌ی بهینه‌ی منابع الگوریتمی ارائه، و اثبات می‌شود که این سیاست بهینه است.[۷]

جدول ۲. سطوح بهینه‌ی حاصل از انجام آزمایش.

عامل	سطح بهینه
اندازه جمعیت (Pop_size)	۸۰
احتمال انجام عملگر جهش (mut_rate)	۰/۲
تعداد تکرارهای بهبودنیافته	۵۰
تعداد کل تکرارها	۱۲۰

فرض کنیم زمان‌بندی موجه S و به تبع آن ω و α تعیین شود؛ در این صورت اگر در جواب موجه به دست آمده یک ω_{kt1} بزرگ‌تر از صفر، و نیز یک α_{kt2} بزرگ‌تر از صفر با شرط $t_2 > t_1$ وجود داشته باشد و همچنین معادله ۱۶ برقرار باشد اثبات می‌شود که (S, α', ω') بهتر از سیاست کنونی است.

$$\alpha'_{kt} = \left\{ \begin{array}{l} \alpha_{kt} - \min(\omega_{kt1}, \alpha_{kt2}) \\ \alpha_{kt} \end{array} \right\}$$

$$\omega'_{kt} = \left\{ \begin{array}{l} \omega_{kt} - \min(\omega_{kt1}, \alpha_{kt2}) \\ \omega_{kt} \end{array} \right\} \qquad (۱۶)$$

$$C^k_r(t_2 - t_1) < C^p_k \qquad (۱۷)$$

نحوه‌ی کار الگوریتم به این صورت است که مادامی که شرط ذکر شده در بالا (α_{kt2} بزرگ‌تر از صفر و نیز ω_{kt1} بزرگ‌تر از صفر با شرط $t_2 > t_1$ در این جواب وجود داشته باشد) در اجاره‌ی منابع صورت گرفته وجود نداشته باشد منابع به اندازه مصرف فعالیت‌های پروژه اجاره خواهد شد و در صورت احراز شرط بالا جواب بهتر معرفی، و جایگزین خواهد شد.(جدول ۲)

۴. نتایج محاسباتی

برای حل این مسئله از دو روش دقیق و فرابتکاری (ژنتیک) استفاده شده است. برای به دست آوردن جواب بهینه‌ی دقیق از نرم‌افزار قدرت‌مند سیپلکس[۲۳] استفاده شده است. سیپلکس از ترکیب روش‌هایی مثل سیمپلکس و شاخه و کران و سایر روش‌های بهینه‌سازی برای یافتن جواب استفاده می‌کند.[۲۴]

خروجی سیپلکس علاوه بر جواب نهایی، در هر زمان یک کران پایین برای مسئله‌ی کمینه‌سازی نیز می‌دهد که می‌تواند برای انجام مقایسه با جواب به دست آمده از الگوریتم ژنتیک بسیار مفید باشد. کدنویسی الگوریتم ژنتیک در زبان برنامه‌نویسی C# انجام شده و تمامی محاسبات در یک رایانه‌ی شخصی با سرعت پردازنده‌ی G.H ۱ و GB RAM ۲ انجام شده است.

پژوهشگران کتابخانه‌یی از مسائل استاندارد به وجود آوردند[۲۲] که نسبت به کتابخانه‌های قبلی دارای دامنه‌ی وسیع‌تری بود و مسائل سرمایه‌گذاری منابع و تسطیح منابع را نیز پوشش می‌داد. امتیاز این‌گونه مسائل استاندارد این است که اولاً از طریق اینترنت قابل دسترس‌اند، ثانیاً دائماً درحال کامل شدن هستند، و در عین حال دائماً مدل‌های جدید به آنها اضافه می‌شود و جواب‌های بهینه یا جواب‌های نزدیک‌تر به بهینه که تاکنون برای آنها به دست آمده است در آنها ثبت شده و از طریق اینترنت قابل دسترس است. از طرفی برای مسائلی که جواب بهینه آنها به دست نیامده است هر فردی می‌تواند درصورتی که جواب بهینه یا بهتر برای آن مسائل یافت به این مجموعه اضافه کند تا مورد استفاده دیگران قرارگیرد. هر روش ابتکاری جدید نیز از این طریق قابل مقایسه با دیگر روش‌هاست.

مسائل استفاده شده از کتابخانه‌ی مسائل استاندارد ProGen، و مجموعه مسائل شناخته شده تحت نام $j۱۲۰, j۹۰, j۶۰, j۳۰, j۲۰, j۱۰$ انتخاب شده است.

برای سنجش میزان پیچیدگی مسائل زمان‌بندی پروژه شاخص‌هایی وجود دارد که از مسئله‌یی به مسئله‌ی دیگر تغییر می‌کند. یکی از این پارامترها که تقریبی از بزرگی یک مسئله را ارائه می‌دهد تعداد فعالیت‌های پروژه است که هرچه بیشتر باشد حل مسئله زمان‌برتر خواهد شد. یکی دیگر از این پارامترها، «تعداد منبع[۲۴]» (RF) است که متوسط تعداد روابط پیش‌نیاز برای هر فعالیت را نشان می‌دهد؛ پارامتر «پیچیدگی شبکه[۲۵]» (NC) معرف متوسط درصد مقدار منابعی است که هر فعالیت لازم دارد؛ پارامتر قدرت منبع (RS)[۲۶] مقدار صفر می‌گیرد اگر منبع مورد نظر به‌اندازه‌ی استفاده فعالیتی باشد که بیشترین استفاده را از آن می‌برد، و نیز مقدار ۱ می‌گیرد اگر مقدار منبع مورد نظر به‌اندازه‌یی باشد که معادل حالت منبع نامحدود باشد.

برای مسئله‌ی اجاره‌ی منابع پارامترهای دیگری نیز مطرح است. اولین پارامتر d یا همان بیشینه زمان پایان پروژه است. این زمان می‌تواند برابر یا بیشتر از بیشینه زمان پایانی باشد که از روابط پیش‌نیاز به دست می‌آید. این زمان که از روابط پیش‌نیاز بین فعالیت‌ها به دست می‌آید $d_{۰,n+۱}$ نام‌گذاری شده است. هرچه d بیشتر از $d_{۰,n+۱}$ باشد فضای بیشتری به مسئله می‌دهد و جست‌وجو را زمان‌برتر می‌کند[۷] و این باعث سخت‌تر شدن مسئله خواهد شد. پارامتر دیگر نسبت هزینه‌ی اجاره به تدارک (CQ)[۲۷] برای منابع است که چنین تعریف می‌شود:

$$CQ = \frac{C_r}{C_p}$$

با افزایش این نسبت به پیچیدگی این مسئله افزوده می‌شود[۷] به‌طوری که وقتی این نسبت صفر است این مسئله به مسئله‌ی ساده‌تر سرمایه‌گذاری منابع تبدیل می‌شود.

با حل مثال‌های ذکرشده توسط نرم‌افزار سیپلکس مشاهده می‌شود مسائل کوچک با پیچیدگی پایین در زمان معقول حل می‌شود و جواب بهینه به دست می‌آید. درخصوص مسائل بزرگ نیز پس از گذشت زمانی نسبتاً طولانی نرم‌افزار حد بالا و پایین مسئله را با اختلاف حداکثر ۲ درصد نسبت به هم ارائه می‌دهد که حد پایین آن برای مقایسه با جواب‌های الگوریتم ژنتیک استفاده شده است.

در جدول ۳ زمان مورد نیاز برحسب ثانیه برای حل این مسئله توسط نرم‌افزار Cplex به‌ازای تعداد فعالیت‌های مختلف پروژه نشان ثبت شده است.

این جدول نشان می‌دهد با افزایش تعداد فعالیت‌ها زمان حل افزایش می‌یابد به‌طوری که نرم‌افزار برای حل مثال‌های با ۶۰ فعالیت زمانی به‌مراتب بیشتر از مسائل کوچک صرف می‌کند. از آنجا که درمورد مسائل کوچک با ۱۰ و ۲۰ فعالیت، هر دو روش دقیق و ابتکاری در زمانی محدود قادر به حل مسئله‌اند، درصد موارد رسیدن به جواب بهینه در زمان محدود به‌عنوان معیاری برای بررسی عملکرد الگوریتم ژنتیک در نظر گرفته شده است.

در جدول ۴ درصد مواردی که الگوریتم ژنتیک طراحی شده براساس فواصل

جدول ۳. زمان مورد نیاز برای حل مسئله با استفاده از نرم‌افزار سیپلکس.

تعداد فعالیت	زمان مورد نیاز (s)
۱۰	۱٫۱۷
۲۰	۱٫۵۸
۳۰	۲٫۲۸
۶۰	۴۵۹

مختلف زمان پایان به جواب بهینه رسیده نشان می‌دهد. در اینجا با افزایش فاصله بین d و $d_{۰,n+۱}$ فضای جست‌وجو افزایش یافته و برای رسیدن به جواب بهینه زمان بیشتری لازم است. در جداول ۴ و ۵ نتایج محاسباتی برای زمان اجرای ۱۰ ثانیه ثبت شده است. عدد CQ نسبت هزینه‌ی اجاره به هزینه‌ی تدارک را نشان می‌دهد.

درمورد مسائل بزرگ‌تر میانگین انحراف از حد پایین ارائه‌شده توسط نرم‌افزار سیپلکس معیار ارزیابی و بررسی کارایی الگوریتم در نظرگرفته شده است (جدول ۵). در شکل ۳ زمان حل توسط الگوریتم ژنتیک به‌ازای تعداد فعالیت‌های مختلف پروژه نشان داده شده است.

در جدول ۵ ستون اول تعداد فعالیت‌ها و ستون دوم، میانگین انحراف از حد پایین ارائه شده توسط نرم‌افزار سیپلکس برای مثال‌های حل شده است. چنان که مشاهده می‌شود برای مثال‌های با تعداد فعالیت‌های بالا نیز الگوریتم ژنتیک طراحی شده قادر است جواب‌های نزدیک به بهینه که درصد انحراف قابل قبولی از حد پایین دارند ارائه دهد.

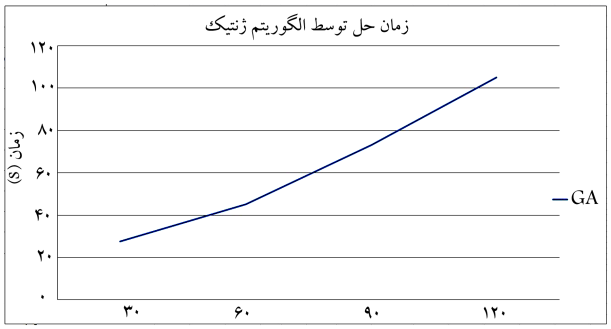
۱.۴. بررسی اثر نرخ بهره بر مسئله

در این بخش تأثیر نرخ بهره بر این مسئله مورد بررسی قرار می‌گیرد. فرض می‌کنیم پروژه‌یی با ۱۰ فعالیت و ۴ منبع وجود دارد و $d = ۱٫۵ \times d_{۰,n+۱}$ و $CQ = ۰٫۶$ در نظرگرفته شده است. روابط فعالیت‌ها و میزان مورد نیاز منبع برای آن‌ها مطابق جدول ۶ است.

در شکل ۴ زمان‌بندی و سیاست اجاره‌ی بهینه برای منبع اول این مسئله، در حالتی که نرخ بهره صفر است و این مسئله طبق معادله به مسئله‌ی ساده‌تر اجاره‌ی منابع تبدیل شده، نشان داده شده است.

حال نرخ بهره را ۳۰ درصد قرار می‌دهیم و زمان‌بندی بهینه را دوباره به دست می‌آوریم (شکل ۵). چنان که مشاهده می‌شود وجود نرخ بهره بالا باعث می‌شود فعالیت‌ها تا حد امکان در زمان بزرگ‌تری زمان‌بندی شوند. این موضوع نشان می‌دهد که درصورت لحاظ‌شدن نرخ بهره در مدل، هزینه‌های پروژه که همان مصرف منابع برای انجام فعالیت‌هاست به زمان عقب‌تری انتقال می‌یابد و در مواقعی که نرخ بهره صفر است منابع به سمت تسطیح بیشتر پیش خواهند رفت تا هزینه‌ی اجاره افزایش پیدا نکند. تسطیح منابع عبارت است از کمینه‌شدن تغییرات سطح استفاده از هریک از منابع،[۲۵و۲۶] و در ادبیات تسطیح منابع بیشتر تلاش در راستای کم‌شدن نوسان و تغییر سطح منبع است.

درصورت وجود نرخ بهره‌ی بالا، عمده‌ی فعالیت‌ها در دیرترین زمان ممکن زمان‌بندی می‌شوند و شناوری حاصل از تفاوت d و $d_{۰,n+۱}$ به ابتدای پروژه منتقل



شکل ۳. زمان حل مسائل مختلف با الگوریتم ژنتیک.

جدول ۴. درصد جواب‌های بهینه‌ی یافت شده توسط الگوریتم ژنتیک.

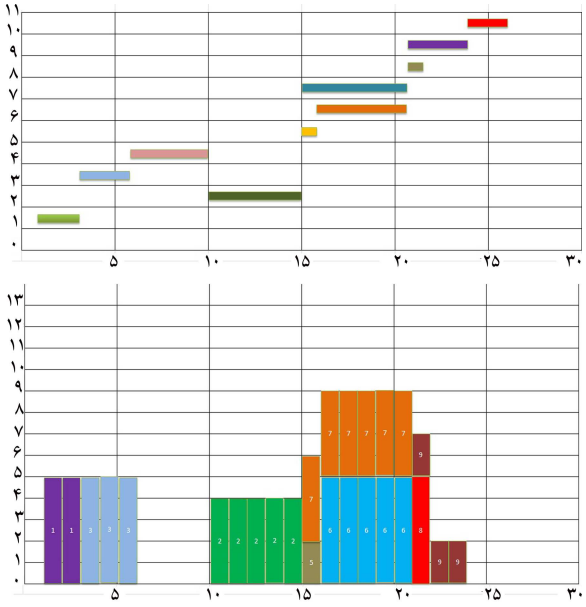
$CQ = ۰/۵$		$d = d_{۰,n+۱}$ (٪)	$d = ۱/۱d_{۰,n+۱}$ (٪)	$d = ۱/۲۵d_{۰,n+۱}$ (٪)	$d = ۱/۵d_{۰,n+۱}$ (٪)
Opt٪	$n = ۱۰$	۹۶/۰	۸۴/۰	۸۱/۰	۷۶/۰
	$n = ۲۰$	۳۷/۰	۲۵/۰	۱۹/۰	۱۱/۰

جدول ۵. زمان حل و انحراف از جواب بهینه به ازای تعداد فعالیت‌های مختلف.

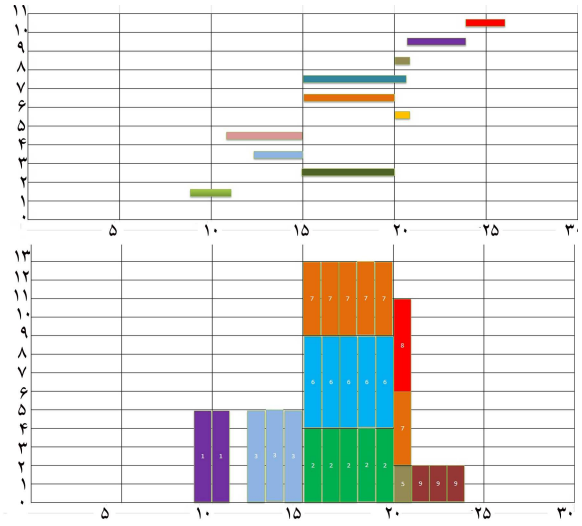
$CQ = ۰/۵$, $d = d_{۰,n+۱}$	Mean Deviation from lower bound (LB) (٪)
۳۰ Activity	۱/۳۲
۶۰ Activity	۲/۹۲
۹۰ Activity	۳/۴۲
۱۲۰ Activity	۳/۲۳

جدول ۶. روابط فعالیت‌ها و میزان مورد نیاز منبع برای مثال ذکر شده.

فعالیت	مدت زمان	پس‌نیازهای فعالیت	کاربرد منبع
Task۰	۰	(۱, ۲, ۳)	(۰, ۰, ۰, ۰)
Task۱	۲	۴	(۵, ۴, ۰, ۰)
Task۲	۵	۵	(۴, ۰, ۰, ۵)
Task۳	۳	(۵, ۷)	(۵, ۵, ۱, ۰)
Task۴	۴	(۶, ۷, ۸)	(۰, ۵, ۰, ۵)
Task۵	۱	۱۱	(۲, ۲, ۲, ۲)
Task۶	۵	(۸, ۹, ۱۰)	(۵, ۲, ۳, ۱)
Task۷	۶	۹	(۴, ۰, ۰, ۰)
Task۸	۱	۱۰	(۵, ۰, ۰, ۲)
Task۹	۳	(۱۰, ۱۱)	(۲, ۵, ۳, ۱)
Task۱۰	۲	۱۱	(۰, ۰, ۵, ۳)
Task۱۱	۰		(۰, ۰, ۰, ۰)



شکل ۴. زمان‌بندی و نحوه‌ی استفاده از منابع در حالت بدون نرخ بهره.



شکل ۵. زمان‌بندی و نحوه‌ی استفاده از منابع در حالت با نرخ بهره ۳۰ درصد.

خواهد شد. اما وقتی نرخ بهره پایین است فعالیت‌ها از این شناوری برای تسطیح هرچه بیشتر منابع استفاده خواهند کرد. نحوه‌ی زمان‌بندی دو فعالیت ۳ و ۴ در مثال ارائه شده، به‌روشنی مؤید این مطلب است. در واقع بین تسطیح منابع و کاهش هزینه‌ی اجاره از یک سو، و انتقال زمان‌بندی به آخر پروژه و تجمیع فعالیت‌ها روی هم به‌دلیل وجود نرخ بهره از سوی دیگر، نوعی دوگانگی وجود دارد. این دوگانگی به‌نوبه‌ی خود در پیچیدگی مسئله تأثیرگذار است.

۵. نتیجه‌گیری

برای تأمین منابع مصرفی یک پروژه ــ چنان که در اکثر پروژه‌های واقعی دیده می‌شود ــ می‌توان منابع را اجاره کرد. در اجاره‌ی منابع نیز در نظرگرفتن هزینه‌های وابسته به زمان در کنار هزینه‌های مستقل از زمان باعث تصمیم‌گیری بهینه برای تأمین منابع می‌شود. لذا مدل‌سازی و حل این مسئله و تبدیل آن به مدلی که در پروژه‌های واقعی کاربرد داشته باشد بسیار حائز اهمیت است.

ارائه‌ی یک الگوریتم ژنتیک کارا و مناسب برای حل این مسئله منجر به کسب جواب‌های مرغوب و نزدیک به بهینه در زمانی معقول می‌شود. عملکرد این الگوریتم چنان که از نتایج محاسباتی حاصل شد، در به دست آوردن جواب‌های بهینه و نزدیک بهینه برای مسائل کوچک قابل توجه است؛ به همین دلیل می‌توان از آن برای به دست آوردن جواب‌های خوب برای مسائل واقعی و بزرگ استفاده کرد. همچنین در نظرگرفتن نرخ بهره در مسئله باعث نزدیک‌تر شدن فضای مدل‌سازی به واقعیت موجود در پروژه‌ها می‌شود. با ورود نرخ بهره به مدل مشاهده کردیم که چگونه تغییرات در نرخ بهره موجب واکنش مسئله به آن و تغییر زمان‌بندی انجام کارها و حتی زمان‌های اجاره‌ی منابع می‌شود.

پانوشت‌ها

1. resource investment
2. resource renting
3. Hartwig Nubel
4. procurement cost
5. renting cost
6. resource renting problem
7. feasible
8. deadline
9. interruption
10. fictitious
11. minimum time lag
12. maximum time lag
13. deadline
14. lump-sum payment
15. earliest start
16. latest start
17. after finished activity
18. active schedule
19. one point ordered crossover (OX)
20. two-point ordered crossover (2OX)
21. swap mutation
22. modified swap mutation
23. CPLEX
24. resource factor
25. network complexity
26. resource strength
27. cost quotient

منابع (References)

1. Słowiński, R. "Multiobjective network scheduling with efficient use of renewable and nonrenewable resources", *European Journal of Operational Research*, Elsevier Science Publishers, **7**(3), pp. 265-273 (1981).
2. Hartmann, S. and Briskorn, ."A survey of variants and extensions of the resource-constrained project scheduling problem", *European Journal of Operational Research*, **207**(1), pp. 1-14 (2010).
3. Kolisch, R. and Padman, R. "An integrated survey of deterministic project scheduling", *Omega*, **29**(3), pp. 249-272 (2001).
4. Brucker, P., Drexl, A., Möhring, R., Neumann, K. and Pesch, E. "Resource-constrained project scheduling: Notation, classification, models, and methods", *European Journal of Operational Research*, **112**(1), pp. 3-41 (1999).
5. Demeulemeester, E. "Minimizing resource availability costs in time-limited project network", *Management Science*, **41**(10), pp. 1590-1598 (1995).
6. Möhring, R.H. "Minimizing costs of resource requirements in project networks subject to a fixed completion time", *Operations Research*, **32**(1), pp. 89-120 (1984).
7. Nübel, H. "The resource renting problem subject to temporal constraints", *OR-Spektrum*, **23**(3), pp. 359-381 (2001).
8. Elmaghraby, S.E. and Herroelen, W.S. "The scheduling of activities to maximize the net present value of projects", *European Journal of Operational*, **49**(1), pp. 35-49 (1990).
9. Blazewicz, J., Lenstra, J.K. and Rinnooy Kan, A.H.G. "Scheduling subject to resource constraints: Classification and complexity", *Discrete Applied Mathematics*, **5**(1), pp. 11-23 (1983).
10. Hartmann, S. "Project scheduling with multiple modes: A genetic algorithm", *Annals of Operations Research*, **102**, pp. 111-135 (2001).
11. Hartmann, S. "A competitive genetic algorithm for resource-constrained project scheduling", *Naval Research Logistics*, **45**(7), pp. 733-750 (1998).
12. Kolisch, R. and Hartmann, S. "Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis", *International Series in Operations Research & Management Science*, **14**, pp. 147-178 (1999).
13. Zhu, J., Li, X. and Shen, W. "Effective genetic algorithm for resource-constrained project scheduling with limited preemptions", *International Journal of Machine Learning*, **2**, pp. 55-65 (2011).
14. Ballestın, F. "A genetic algorithm for the resource renting problem with minimum and maximum time lags", *Lecture Notes in Computer Science*, **4446**, pp. 25-35 (2003).
15. Neumann, K., Schwindt, C. and Zimmermann, J., *Project Scheduling with Time Windows and Scarce Resources*, Springer, Berlin (2003).
16. Najafi, A.A. and Niaki, S.T.A. "A genetic algorithm for resource investment problem with discounted cash flows", *Applied Mathematics and Computation*, **183**(2), pp. 1057-1070 (2006).
17. Eiben, A.E. and Smith, J.E., *Introduction to Evolutionary Computing*, Natural Computing Series, Berlin, Springer (2003).
18. Holland, J.H., *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, U Michigan Press (1975).
19. Koza, JR., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, 1 st Edition, MIT Press (1992).
20. Shadrokh, Sh. and Kianfar, F. "A genetic algorithm for resource investment project scheduling problem, tardiness permitted with penalty", *European Journal of Operational Research*, **181**(1), pp. 86-101 (2006).
21. Sivanandam, S.N. and Deepa, S.N., *Introduction to Genetic Algorithms*, Springer, New York (2008).
22. Taguchi, G., *Introduction to Quality Engineering: Designing Quality into Products and Processes*, White Plains: Asian Productivity Organization/UNIPUB (1986).
23. Kolisch, R., Schwindt, C. and Sprecher, A. "Benchmark instances for project scheduling problems", *International Series in Operations Research & Management Science*, Springer, **14**, pp. 197-212 (1999).
24. CPLEX, IBMI. International Business Machines Corporation (2009).

25. Neumann, K. and Zimmermann, J. "Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints", *European Journal of Operational Research*, **127**(2), pp. 425-443 (2000).

26. Nudtasomboon, N. and Randhawa, S.U. "Resource-constrained project scheduling with renewable and non-renewable resources and time-resource tradeoffs", *Computers & Industrial Engineering*, **32**(1), pp. 227-242 (1997).