

کمینه کردن جریمه دیرکرد و بیشینه کردن پاداش زودکرد انجام فعالیت‌ها در مسئله MRCPSP/max با استفاده از الگوریتم ژنتیک دو مرحله‌ای

جعفر باقری نژاد^{۱*}، فریبرز جولای^۲ و زهرا رفیعی مجد^۳

^۱ استادیار دانشکده مهندسی صنایع - دانشگاه الزهرا (س)

^۲ استاد دانشکده مهندسی صنایع - پردیس دانشکده‌های فنی - دانشگاه تهران

^۳ کارشناس ارشد مهندسی صنایع - دانشگاه الزهرا (س)

(تاریخ دریافت ۹۱/۲/۹، تاریخ دریافت روایت اصلاح‌شده ۹۱/۸/۶، تاریخ تصویب ۹۲/۱/۲۴)

چکیده

در این مقاله، برای اولین بار مسئله زمانبندی پروژه در شرایط محدود بودن منابع، امکان اجرای فعالیت‌ها در چندین مد و با در نظر گرفتن تأخیرات زمانی بیشینه و کمینه میان زمان‌های شروع فعالیت‌ها، MRCPSP/max، با هدف کمینه‌کردن جریمه دیرکرد و بیشینه کردن پاداش زودکرد تمام فعالیت‌ها، مطرح شده و مورد بررسی قرار گرفته است. پس از بررسی تاریخچه و روند استفاده از روش‌های مختلف در حل مسائل مشابه در سال‌های گذشته، الگوریتم ژنتیک به عنوان الگوریتم مورد استفاده در این تحقیق برگزیده شده است. در این مقاله، شیوه پیدا کردن جواب برای مسئله مورد نظر به این ترتیب است که با استفاده از یک الگوریتم ژنتیک، مسئله اصلی را به مسئله‌ای که در آن هر فعالیت فقط یک حالت اجرایی دارد، ساده کرده و سپس در فاز دوم حل، با استفاده از یک الگوریتم ژنتیک مستقل دیگر، بهترین جواب مسئله حاصل از فاز اول الگوریتم را می‌یابیم. عناصر اصلی و عملگرهای هر دو الگوریتم، از هم مستقل هستند. در انتها نیز نتایج عددی حاصل از الگوریتم‌های پیشنهادی که به وسیله زبان برنامه نویسی MATLAB نوشته شده است، با نتایج موجود در کتابخانه مسائل زمانبندی مقایسه شده‌اند و مشاهده می‌شود که الگوریتم ارائه شده در این تحقیق در چندین مورد جواب‌های موجود را بهبود داده است.

واژه‌های کلیدی: زمانبندی پروژه، تأخیرات زمانی بیشینه و کمینه، الگوریتم ژنتیک، فعالیت‌ها با چندین مد اجرایی

مقدمه

کرده‌اند. از جمله این محققان می‌توان به فرانک و همکاران (۲۰۰۱) [۶] و (۱۹۹۸) [۷]، سستا (۲۰۰۲) [۸]، اسمیت و پایل (۲۰۰۴) [۹]، ایفینوا و همکاران (۲۰۱۳) [۱۰]، فو و همکاران (۲۰۱۲) [۱۱]، شوت و همکاران (۲۰۱۳) [۱۲] و بیانکو و کارامیا (۲۰۱۱) [۱۳] اشاره کرد. بالستین و همکاران (۲۰۱۱) [۱۴] نیز از یک الگوریتم تکاملی برای بررسی این مسئله بهره برده‌اند.

حال اگر فعالیت‌ها امکان اجرا در بیش از یک مد را داشته باشند، مسئله MRCPSP/max^۱ (یا MRCPSP/GPR) پدید خواهد آمد.

مسئله MRCPSP/max، حالت تعمیم یافته بسیاری از مسائل زمانبندی پروژه است که از میان آنها می‌توان به RCPSP/max، RCPSP و MRCPSP اشاره کرد.

درباره مسئله MRCPSP/max مشاهده می‌شود که ادبیات MRCPSP/max به گسترده‌گی ادبیات MRCPSP نیست. در تحقیقات انجام شده توسط

بر اساس بیان دوردنورف و همکاران [۱]، تأخیرات زمانی کمینه^۱ و تأخیرات زمانی بیشینه^۲ امکان مدلسازی بسیاری از خصوصیات را که به طور کلی در مسائل زمانبندی کاربرد دارند، امکانپذیر می‌کنند [۲]، از این رو مسئله به مسائل دنیای واقعی نزدیک‌تر می‌شود. اگر مسئله زمانبندی پروژه را در شرایطی که منابع محدود هستند و تأخیرات زمانی بیشینه و کمینه میان فعالیت‌ها نیز وجود دارد، در نظر بگیریم، به یک مسئله زمانبندی پروژه با منابع محدود با روابط تقدمی کلی^۳ (محدودیت‌های زمانی^۴، تأخیرات زمانی^۵، پنجره‌های زمانی^۶) که به صورت خلاصه RCPSP/max^۲ (یا RCPSP/GPR) نامیده می‌شود، خواهیم رسید. برای حل مسئله RCPSP/max، پژوهشگرانی چون بارتوش و همکاران (۱۹۸۸) [۳]، دی ریک (۱۹۹۸) [۴]، فست (۱۹۹۸) [۵] و دوردنورف (۲۰۰۰) [۱] از رویکرد شاخه و کران بهره برده‌اند. بعضی دیگر از محققان نیز با کمک روش‌های ابتکاری و فرابابتکاری این نوع مسئله را بررسی

بخش ۵، نتیجه‌گیری و ارائه پیشنهاد برای تحقیقات آینده ارائه شده است.

مدل

پیش از پرداختن به مدل ریاضی مسئله مورد بحث در این مقاله، ابتدا جدول علائم استاندارد مورد استفاده، معرفی می‌شود.

جدول ۱: علائم استاندارد مورد استفاده در این مقاله

نماد	تعریف
$V = \{0, 1, \dots, n+1\}$	مجموعه فعالیت‌های شبکه با اندیس j
$N = \{(i, j); i, j \in V\}$	مجموعه روابط تقابلی بین فعالیت‌های شبکه
$M_j = \{1, \dots, M_j\}$	مجموعه حالت‌های اجرای فعالیت j با اندیس m
$R = \{1, \dots, R\}$	مجموعه منابع تجدیدپذیر با اندیس k
$NR = \{1, \dots, NR\}$	مجموعه منابع تجدیدناپذیر با اندیس l
R_k^p	سطح ثابت موجودی منبع تجدیدپذیر k در هر پریود زمانی
R_l^p	کل موجودی منبع تجدیدناپذیر l
d_{jm}	طول اجرای فعالیت j در حالت m
r_{jm}^p	میزان نیاز فعالیت j در حالت m به منبع تجدیدپذیر k
r_{jm}^l	میزان نیاز فعالیت j در حالت m به منبع تجدیدناپذیر l
$s = (s_0, \dots, s_{n+1})$	مجموعه زمان‌های شروع
$f = (f_0, \dots, f_{n+1})$	مجموعه زمان‌های پایان
T_{ij}^{max} و T_{ij}^{min}	حداقل و حداکثر فاصله زمانی بین شروع فعالیت‌های i و j
D	تاریخ تحویل پروژه
T	یک حد بالا بر طول زمان‌بندی پروژه
$t = 0, 1, 2, \dots, T$	اندیس زمان
e_{jm}	پاداش هر واحد زودکرد فعالیت j که در مد m اجرا می‌شود
t_{jm}	جریمه هر واحد دیرکرد فعالیت j که در مد m اجرا می‌شود
E_{jm}	مقدار زودکرد فعالیت j که در مد m اجرا می‌شود
T_{jm}	مقدار دیرکرد فعالیت j که در مد m اجرا می‌شود

مدل ریاضی مسئله زمان‌بندی پروژه با هدف کمینه‌کردن هزینه‌های دیرکرد و بیشینه‌کردن پاداش زودکرد انجام فعالیت‌ها:

مدلی که در این مقاله معرفی می‌شود، به این شکل است که در آن به دنبال یافتن یک زمان شروع و یک مد اجرایی برای هر فعالیت هستیم، به گونه‌ای که محدودیت‌های مربوط به تأخیرات زمانی بیشینه و کمینه و نیز محدودیت‌های منابع تجدیدپذیر و تجدیدناپذیر برقرار باشند و هزینه دیرکرد فعالیت‌ها کمینه شود و پاداش تحویل دادن فعالیت‌ها پیش از موعد مقرر نیز بیشینه شود. پیش از ارائه مدل ریاضی مسئله مورد بحث،

محققانی نظیر دی ریک (۱۹۹۸) [۱۵] و ایباراکی و نونوبه (۲۰۰۳) [۱۶] از رویکرد فراابتکاری جستجوی ممنوع برای حل مسئله MRCPSP/max استفاده شده است. هیلمن [۱۷] نیز در سال ۲۰۰۱ از قانون اولویت چندگذری کمک گرفته است و در سال ۲۰۰۳ [۱۸] با استفاده از روش حل دقیق (روش شاخه و کران) به بررسی مسئله مورد نظر پرداخته است (در روش اولویت چندگذری، که از جمله روش‌های ابتکاری است، با استفاده از طرح‌های تولید زمانبندی و قواعد اولویت، به تولید جواب پرداخته می‌شود [۱۹]). سید حسینی و سبزه‌پرور نیز با کمک الگوریتم‌های فراابتکاری (۲۰۰۷) [۲۰] و (۲۰۰۸) [۲۱] اقدام به حل مسئله مذکور کرده‌اند. جیاویچ نیز در سال ۲۰۱۱ از روش‌های ابتکاری برای حل این مسئله استفاده کرده است [۲۲ و ۲۳]. پژوهشگران دیگری نظیر باریوس و همکاران (۲۰۱۱) [۲] از الگوریتم ژنتیک دوبل^۹ برای حل مسئله MRCPSP/max بهره برده‌اند. بدیهی است هنگامی که فعالیت‌های یک پروژه فقط در یک حالت اجرا می‌شوند، حل مسئله زمانبندی مرتبط بسیار ساده‌تر از زمانی است که هر یک از فعالیت‌ها چندین مد اجرایی دارند. اگر بتوان با کمک روش‌هایی، مسئله پیچیده اولی را به مسئله دوم تبدیل کرد، در زمان حل صرفه‌جویی خواهد شد. یکی از شیوه‌های انجام این کار، استفاده از الگوریتم‌های فراابتکاری است که در این مقاله مورد توجه قرار گرفته است.

از آنجایی که الگوریتم ژنتیک، یکی از پرکاربردترین روش فراابتکاری در حل مسائل زمانبندی بوده است [۲۴] و [۲۵]، در این مقاله نیز مورد استفاده قرار می‌گیرد. الگوریتم ژنتیک در حقیقت یک شیوه تکاملی است که روی یک جمعیت اولیه، عملیاتی را انجام می‌دهد، والدین را بر می‌گزیند، عملگرهای تقاطع و جهش را اجرایی می‌کند و فرزندان به وجود آمده را ارزشیابی می‌کند. هدف این الگوریتم این است که به طور پی در پی با احتمال بیشتری، بهترین پاسخ‌های موجود را برای ترکیب انتخاب کند تا در نهایت جواب‌های بهتری ایجاد شود.

ادامه مقاله بدین ترتیب است: در بخش ۲، مدل ریاضی مسئله مورد بحث در این مقاله ارائه می‌شود. در بخش ۳، راه حل پیشنهادی برای مدل ارائه‌شده در بخش ۲، شرح داده شده است. بخش چهارم به بررسی نتایج محاسباتی الگوریتم‌های ارائه‌شده می‌پردازد و در نهایت در

تحویل فعالیت است. رابطه (۵) محدودیت روابط
تقدمی کلی میان فعالیت‌ها را نشان می‌دهد و رابطه‌های
(۶) و (۷) نیز به ترتیب بیانگر محدودیت منابع تجدیدپذیر
و منابع تجدیدناپذیر هستند. زمان‌های شروع فعالیت‌ها
غیر منفی در نظر گرفته شده است. پروژه نیز در زمان
صفر آغاز می‌شود (روابط ۸ و ۹).

روش حل مدل با کمک الگوریتم ژنتیک دو مرحله‌ای

هیلمن در پژوهشی در سال ۲۰۰۱ [۱۷]، یک مسئله
MRCPSP/max را به دو زیر مسئله: ۱- مسئله
تخصیص بهترین مد ۲- مسئله RCPSPP/max تقسیم
کرده است. به عبارت دیگر برای حل یک مسئله
MRCPSP/max باید ابتدا در مورد هر فعالیت تعیین
کرد که بهترین حالت انجام آن فعالیت، کدام حالت است؟
پاسخ به این سؤال، مدهای انجام هر فعالیت را به ۱ مد
تقلیل داده و مسئله را به یک مسئله زمانبندی
RCPSPP/max تبدیل می‌کند. حل مسئله
RCPSPP/max حاصل، در حقیقت به حل مسئله اصلی
MRCPSP/max منجر خواهد شد.

در این پژوهش نیز از ایده هیلمن بهره می‌بریم و
مسئله مورد بحث را که می‌توان آن را از مشتقات مسئله
MRCPSP/max به حساب آورد به دو زیر مسئله ۱-
تخصیص بهترین مد و ۲- مسئله حاصل از تخصیص
بهترین مد به فعالیت‌ها، تفکیک می‌کنیم و هر یک از این
زیر مسائل را به کمک یک الگوریتم ژنتیک مستقل که در
محیط نرم‌افزار MATLAB نسخه ۲۰۰۵ با کمک یک
کامپیوتر Core(TM)2 Duo P7350 2.00GHz با
سیستم عامل ویستا کدنویسی کرده‌ایم، حل می‌کنیم. هر
الگوریتم ژنتیک در یک الگوریتم ژنتیک دومرحله‌ای،
ساختار کروموزوم^{۱۰}، تابع برازش^{۱۱}، عملگر تقاطع^{۱۲} و
عملگر جهش^{۱۳} مربوط به خود را دارد.

مسئله یافتن بهترین حالت‌های انجام فعالیت‌ها

در تعیین بهترین مد انجام هر فعالیت، از برخی از
نکات مطرح شده در پژوهش باریوس و همکاران در سال

لازم است درباره مفاهیم کلیدی این مدل توضیحاتی ارائه
شود.

در این مدل h_{jm_j} نمایانگر حداکثر زمان مطلوب اتمام
فعالیت j که در مد m_j اجرا می‌شود، است و f_{jm_j} نیز
زمان اتمام فعالیت مورد نظر را نشان می‌دهد. بدیهی
است که رابطه زیر همواره برقرار است:

$$f_{jm_j} = s_{jm_j} + d_{jm_j} \quad (۱)$$

دیرکرد فعالیت j که در مد m_j اجرا می‌شود، از معادله
زیر به دست می‌آید:

$$T_{jm_j} = \max(0, f_{jm_j} - h_{jm_j}) \quad (۲)$$

t_{jm_j} نیز جریمه هر واحد دیرکرد فعالیت j که در مد
اجرا می‌شود، است.

زودکرد فعالیت j که در مد m_j اجرا می‌شود، از معادله
زیر به دست می‌آید:

$$E_{jm_j} = \max(0, h_{jm_j} - f_{jm_j}) \quad (۳)$$

e_{jm_j} نیز پاداش هر واحد زودکرد فعالیت j که در مد
اجرا می‌شود، است.

حال با توجه به مفاهیم یاد شده، شکل ریاضی مدل
زمانبندی پروژه با هدف کمینه‌کردن هزینه‌های دیرکرد و
بیشینه‌کردن پاداش زودکرد انجام فعالیت‌ها را می‌توان به
صورت زیر نمایش داد:

$$\min \sum_{j=1}^n \sum_{m_j \in M_j} (t_{jm_j} T_{jm_j} - e_{jm_j} E_{jm_j}) \quad (۴)$$

subject to:

$$s_i - s_j \geq \delta_{jm_j i} m_i \quad ((j, i) \in N) \quad (۵)$$

$$\sum_{j \in A(M, S, t)} r_{jm_j k}^p \leq R_k^p, \quad (k \in R^p; t \geq 0) \quad (۶)$$

$$\sum_{j \in V} r_{jm_j k}^g \leq R_k^g, \quad (k \in R^g) \quad (۷)$$

$$m_j \in M_j \quad (j \in V), \quad (۸)$$

$$s_j \geq 0 \quad (j \in V), \quad (۹)$$

$$s_0 = 0$$

تابع هدف در مدل بالا (رابطه ۴)، در پی
کمینه‌کردن هزینه دیرکرد و بیشینه‌کردن پاداش زودکرد

که در آن کروموزوم برای فعالیت‌ها تعریف شده است منجر به مصرف بیش از مقدار موجود منبع تجدیدنشده خواهد شد و در نتیجه، آن کروموزوم از نظر منبع تجدیدنشده، ناموجه و نشدنی خواهد بود و باید حذف شود.

برای ساخت جمعیت اولیه، ابتدا به تعداد از پیش تعیین شده (که "اندازه جمعیت" 16 و در اصطلاح pop-size نامیده می‌شود)، به شیوه‌ای که بیان شد، کروموزوم‌های از نظر منبع موجه، تولید می‌کنیم. گام بعدی این است که کروموزوم‌های جمعیت اولیه، از نظر زمان نیز شدند باشند. برای این منظور کافی است در پروژه انجام‌شده طبق بردار مد هر کروموزوم، "سیکل‌هایی با طول مثبت" 17 ایجاد نشود. یعنی جمع تأخیرات زمانی روی آن سیکل، مثبت نباشد. پس از اعمال شرط شدند بودن از نظر زمان، بار دیگر باید کروموزوم‌ها را بررسی کرد تا در طول فرآیند شدند شدن از نظر زمان، از منظر منبع، ناموجه نشده باشند؛ حال، جمعیت اولیه ساخته شده است. به عبارت دیگر تعداد مورد نظر کروموزوم وجود دارد که از نظر منبع تجدیدنشده و زمان، موجه هستند.

عملگر تقاطع فاز اول

با توجه به ماهیت مسئله تخصیص بهترین مد به هر فعالیت، "عملگر تقاطع مبتنی بر ساختار حلقه" 18 [۲] (تقاطع مبتنی بر ساختار حلقه، تضمین می‌کند که مدهای ساخته شده به وسیله آن از نظر زمان موجه هستند) را با عملگر تقاطع دونقطه‌ای 19 (در این روش دو نقطه به طور تصادفی به عنوان نقاط برش انتخاب شده و کروموزوم‌های والد از آن دو نقطه به سه قسمت تقسیم می‌شوند، آنگاه قسمت وسط ثابت مانده و دو قسمت کناری از دو کروموزوم با یکدیگر تعویض می‌شوند). ترکیب کرده‌ایم. با مشخص بودن احتمال تقاطع، pc، ماکزیمم درصد کروموزوم‌هایی که مشمول اجرای عملگر تقاطع می‌شوند مشخص خواهد شد. اگر این تعداد زوج نبود، از آنجا که امکان جفت کردن کروموزوم‌ها میسر نمی‌شود، می‌توان یک کروموزوم اضافه یا کم کرد. حال به تعداد مشخص شده، کروموزوم‌هایی را که میزان آرایه‌بردار باقیمانده بیشتری از سایرین دارند را دو به دو برای اعمال عملگر تقاطع انتخاب می‌کنیم. سپس از آنجایی که در مورد همه کروموزوم‌ها ژن آخر، مربوط به میزان باقیمانده

۲۰۱۱ [۲]، بهره‌گرفته‌ایم. طرح کلی فاز اول الگوریتم را می‌توان به این شرح بیان کرد:

- تولید جمعیت اولیه
- تا زمانی که شرط پایان الگوریتم (تکرار به تعداد از پیش تعیین شده / معیار همگرایی الگوریتم) محقق نشده است:
- کروموزوم‌هایی را به عنوان والدین از میان جمعیت انتخاب کنید (بدیهی است که هر چه کروموزومی بهتر باشد، با احتمال بیشتری به عنوان پدر یا مادر انتخاب می‌شود).
- به وسیله عملگر تقاطع فرزندان والدین انتخاب شده را تولید کنید. (شدنی بودن فرزندان از نظر منبع بررسی شود).
- عملگر جهش را روی فرزند اعمال کنید.
- ارزیابی کنید که آیا فرزندان از نظر منبع شدند هستند یا خیر؟
- اگر فرزندان بهتر از والدین بودند، جایگزین والدین شوند.
- بهترین بردار مد، بردار باقیمانده متناظر با این بردار مد و نیز مقدار برازش این بردار مد را گزارش کنید.

ساختار کروموزوم و جمعیت اولیه در فاز اول

در الگوریتم ژنتیک فاز اول، هر کروموزوم به وسیله یک بردار مد 14 M نشان داده می‌شود. برازش هر کروموزوم نیز برابر با طول مسیر بحرانی (یا به عبارتی دیگر، جمع تأخیرات زمانی در مسیر بحرانی) پروژه متناظر با آن کروموزوم خواهد بود و با نماد CP(M) نشان داده می‌شود.

برای هر بردار مد، یک بردار به نام بردار باقیمانده 15 تعریف می‌شود که تعداد آرایه‌های این بردار، به تعداد انواع منابع تجدیدنشده است. هر آرایه، نمایانگر تعداد واحدهای آن نوع منبع تجدیدنشده است که با فرض اجرای فعالیت‌ها طبق بردار مد متناظر، هنوز در دسترس هستند. بدیهی است که اگر آرایه‌های یک بردار باقیمانده نامنفی باشند، این موضوع به معنی این است که بردار مد متناظر با این بردار باقیمانده، از نظر منبع، شدند خواهد بود و چنانچه مقدار این آرایه‌ها، عددی منفی بود به این معنا است که اجرا شدن فعالیت‌های پروژه طبق مدهایی

طور تصادفی مقدار مجاز دیگری می‌گیرد. پس از اعمال عملگر جهش، بار دیگر شرط شدنی بودن از نظر منبع تجدیدنشدهی مورد بازبینی قرار می‌گیرد و در صورت برقرار نبودن این شرط، ژن‌های مجاز آنقدر تغییر می‌کنند تا شرط شدنی بودن از نظر منبع برقرار شود.

ارزیابی مقدار برازش هر کروموزوم در فاز اول

برای محاسبه مطلوبیت هر کروموزوم، مقدار برازش هر کروموزوم را برابر با طول مسیر بحرانی پروژه، در حالتی که فعالیت‌ها از مدهای موجود در آن کروموزوم پیروی کنند، قرار می‌دهیم. طول مسیر بحرانی را می‌توان به صورت حاصل جمع تأخیرات زمانی در مسیر انجام پروژه متناظر تعریف کرد.

پس از محاسبه مقدار برازش برای همه کروموزوم‌های موجود در جمعیت، کروموزوم‌ها را به ترتیب از بهترین برازش به بدترین، مرتب می‌کنیم. بهترین کروموزوم (که بهترین برازش / کمترین طول مسیر بحرانی را دارد)، انتخاب می‌شود. در اینجا یک نسل از الگوریتم ژنتیک به پایان می‌رسد.

زمانی که نسل‌ها به تعداد از پیش تعیین شده (معیار همگرایی الگوریتم) تکرار شدند، بهترین کروموزوم همه نسل‌ها (کروموزومی که بهترین برازش را در میان بهترین کروموزوم‌های همه نسل‌ها دارد)، به عنوان جواب نهایی مسئله معرفی می‌شود و می‌توان از کروموزوم حاصله استفاده کرد و فعالیت‌ها را طبق مدهای موجود در این کروموزوم اجرا کرد.

حال با مشخص شدن بهترین حالت اجرای فعالیت‌ها، در حقیقت مسئله $MRCPSp/\max$ به مسئله $RCPSp/\max$ تبدیل شده است و از پیچیدگی آن تا حد زیادی کاسته شده است. لازم به ذکر است که به دلیل جامعیت مسئله $MRCPSp/\max$ نسبت به مسئله $MRCPSp$ ، از این شیوه می‌توان برای کاستن از پیچیدگی‌های مسئله $MRCPSp$ و تبدیل آن به مسئله $RCPSp$ نیز بهره برد.

حل مسئله $RCPSp/\max$ با هدف کمینه کردن هزینه دیرکرد و بیشینه کردن پاداش زودکرد انجام فعالیت‌ها

از منبع تجدیدنپذیر است و نمایانگر مد نیست. بنابراین در محاسبات مربوط به عملگر تقاطع، این ژن را در نظر نمی‌گیریم. حال با ترکیب دو عملگر یاد شده، به این صورت عمل می‌کنیم که فرزند دختر، ژن‌های واقع بر سیکل‌ها (حلقه‌ها) را از مادر و باقی ژن‌ها را از پدر می‌گیرد و فرزند پسر، ژن‌های واقع بر سیکل‌ها را از پدر و باقی ژن‌ها را از مادر می‌گیرد. به این شیوه، فرزندان دختر و پسر را ایجاد کرده و مقدار باقیمانده از منبع تجدیدنشدهی را برای هر یک از کروموزوم‌های دختر و پسر محاسبه می‌کنیم. اگر کروموزوم‌های فرزند، مقدار باقیمانده بیشتری از والدین داشته باشند، جایگزین والدینشان در جمعیت می‌شوند و در غیر این صورت والدین کماکان در جمعیت باقی خواهند ماند. با استفاده از این شیوه جایگزینی، شرط شدنی بودن از نظر منبع نیز در مورد کروموزوم‌های جمعیت همچنان برقرار خواهد ماند. با اعمال عملگر تقاطع، جمعیت حاصله، حداقل به خوبی جمعیت قبل از اعمال این عملگر خواهد بود.

عملگر جهش فاز اول

عملگر جهش که در این الگوریتم از آن استفاده می‌کنیم، به این ترتیب عمل می‌کند که با ژن‌هایی که روی حلقه‌ها واقع هستند، کاری ندارد. به عبارتی دیگر فقط مدهای فعالیت‌هایی ممکن است از سوی این عملگر دچار جهش شوند که در شبکه پروژه روی حلقه قرار ندارند (لازم به ذکر است که چون فعالیت‌های موهومی نیز فقط یک مد اجرایی دارند، در این عملگر لحاظ نمی‌شوند). با این تمهید شرط موجه بودن از نظر زمان نقض نمی‌شود. بنابراین به صورت بالقوه این عملگر فقط روی بعضی از ژن‌های هر کروموزوم قابل اعمال شدن است که این تعداد، پس از کسر ژن‌های مربوط به فعالیت‌های روی حلقه‌ها به دست می‌آید. از ضرب تعداد ژن‌های مجاز برای جهش، در احتمال جهش، می‌توان حداکثر تعداد ژن‌هایی که عملگر جهش روی آنها پیاده می‌شود را به دست آورد.

در ادامه به این ترتیب عمل می‌شود که به تعداد ژن‌های مجاز، اعداد تصادفی بین صفر و یک تولید می‌کنیم. در صورتی که هر یک از اعداد تصادفی تولیدشده کمتر از احتمال جهش باشد، آنگاه ژن متناظر با عدد تصادفی دچار جهش می‌شود. یعنی مد واقع در آن ژن به

از نظر روابط تقدمی موجه می‌شود و می‌توان آن را "لیست فعالیت موجه تقدمی"^{۲۲} نامید. حال اگر به تعداد مشخصی، کروموزوم موجه تولید کنیم، می‌توانیم یک جمعیت اولیه داشته باشیم.

عملگر تقاطع فاز دوم

در این قسمت، از تقاطع بخش-نگاشته (نگاشت جزئی)^{۲۳} استفاده می‌کنیم. در این روش دو عدد به صورت تصادفی به عنوان نقاط برش به دست می‌آیند. سپس قسمت بین دو نقطه برش را در دو لیست فعالیت تعویض کرده، آنگاه قسمت‌های دو طرف طوری مقدارگذاری می‌شوند که در هیچ یک از دو لیست، فعالیت تکراری انجام نگیرد. در حین انجام این کار، روابط پیشینازی میان فعالیت‌ها نیز مورد توجه قرار می‌گیرد. در نتیجه کروموزوم‌های حاصل از نظر روابط تقدمی نیز شدنی خواهند بود.

تعداد کروموزوم‌هایی که مشمول اجرای این عملگر می‌شوند را می‌توان از حاصلضرب نرخ تقاطع در تعداد اعضای جمعیت اولیه به دست آورد. به تعداد کروموزوم‌ها، عدد تصادفی بین صفر و یک تولید می‌کنیم. کروموزوم‌هایی که عدد تصادفی متناظر با آن‌ها، از احتمال تقاطع، کمتر باشد، برای عمل تقاطع انتخاب می‌شوند. حال دو عدد تصادفی بین صفر و یک بر می‌گزینیم. این دو عدد، موقعیت ژن‌هایی که در کروموزوم‌های انتخاب شده محل برش خواهند بود را تعیین می‌کنند. کروموزوم‌های دختر، ژن‌های بین این دو عدد تصادفی را از مادر و باقی را با رعایت شرط تکراری نبودن ژن و نیز شرط شدنی بودن از نظر روابط پیشینازی بین فعالیت‌ها، از پدر می‌گیرند و کروموزوم‌های پسر، ژن‌های بین آن دو عدد تصادفی را از پدر و باقی را با رعایت دو شرط پیش گفته، از مادر می‌گیرند. پس از ساخت کروموزوم‌های فرزند، این کروموزوم‌ها جایگزین والدینشان می‌شوند.

عملگر جهش فاز دوم

در اینجا از عملگر جهش جابه‌جایی^{۲۴}، استفاده خواهیم کرد. در این روش، دو ژن به طور تصادفی انتخاب شده و با یکدیگر تعویض می‌شوند. پس از اعمال عملگر جهش، شرط شدنی بودن کروموزوم‌ها بار دیگر مورد

پس از اجرای فاز اول، حال باید مسئله حاصل از فاز اول را در فاز دوم، مورد بررسی قرار داده و بهترین پاسخ آن را بیابیم. بهترین پاسخ عبارت است از تخصیص زمان‌های پایان (یا معادل آن، زمان‌های شروع) به فعالیت‌های پروژه، به طوری که محدودیت‌های روابط تقدمی میان فعالیت‌ها و نیز منابع تجدیدشدنی برآورده شوند و بهترین میزان تابع هدف به دست آید.

طرح کلی فاز دوم الگوریتم را می‌توان به شرح زیر بیان کرد:

- تولید جمعیت اولیه
- تا زمانی که شرط پایان الگوریتم (تکرار به تعداد از پیش تعیین شده / معیار همگرایی الگوریتم) محقق نشده است:
- کروموزوم‌هایی را به عنوان والدین از میان جمعیت انتخاب کنید (بدیهی است که هر چه کروموزومی بهتر باشد، با احتمال بیشتری به عنوان پدر یا مادر انتخاب می‌شود).
- به وسیله عملگر تقاطع فرزندان والدین انتخاب شده را تولید کنید. (شدنی بودن فرزندان از نظر روابط تقدمی میان فعالیت‌ها بررسی شود).
- عملگر جهش را روی فرزندان اعمال کنید.
- ارزیابی کنید که آیا فرزندان از نظر روابط تقدمی، شدنی هستند یا خیر؟
- بهترین ترتیب انجام فعالیت‌ها (بهترین لیست فعالیت)، برادر زمان‌های شروع / پایان و نیز مقدار برازش متناظر با بهترین لیست فعالیت را گزارش کنید.

ساختار کروموزوم و جمعیت اولیه فاز دوم

در یک دسته‌بندی کلی می‌توان بیان کرد که در یک الگوریتم ژنتیک، کروموزوم‌ها را می‌توان به سه حالت کلی بر مبنای جایگشت (لیست فعالیت)، بر مبنای روش "کلید تصادفی"^{۲۰} و بر مبنای "قواعد اولویت"^{۲۱} تعریف کرد [۲۶]. برخی محققان مانند کولیش و هارتمن [۲۶] و [۲۷]، نشان داده‌اند که در مسائل RCPSP نمایش کروموزوم‌ها به صورت لیست فعالیت نسبت به سایر روش‌های نمایش، باعث ایجاد نتایج بهتری می‌شود. بنابراین در این مقاله از این شیوه استفاده می‌کنیم. از آنجا که در یک لیست فعالیت، هر فعالیت بعد از تمام پیش‌نیازهایش قرار می‌گیرد، بنابراین لیست فعالیت

الگوریتم در چه مدت زمانی مسئله دشوار اولیه را به یک مسئله بسیار ساده‌تر که در آن هر فعالیت فقط یک حالت اجرا دارد تبدیل می‌کند؟ بنابراین مدت زمان سپری‌شده برای حل مسئله تخصیص بهترین مد، پارامتری است که با استفاده از آن، کارآمدی الگوریتم ژنتیک فاز اول را می‌سنجیم.

برای بررسی کارایی الگوریتم ژنتیک ارائه‌شده در فاز اول، از مسئله mm-j10-ppsp104 موجود در کتابخانه مسائل زمانبندی (از دسته مسائل MRCSP/max) استفاده می‌کنیم. این مسئله توسط نرم‌افزار ProGen/Max تولید شده است.

احتمال جهش در دو سطح ۰.۱ (پایین) و ۰.۵ (بالا) و احتمال تقاطع نیز در دو سطح ۰.۱ (پایین) و ۰.۸ (بالا) مورد بررسی قرار گرفته‌اند.

با یک "طراحی عاملی کامل" ^{۲۶} (۲)، از پارامترهای ذکر شده، تعداد $4 = 2 \times 2$ حالت (اجرا) برای بررسی تأثیر هر یک از عوامل احتمال جهش و احتمال تقاطع بر مدت زمان حصول نتیجه توسط الگوریتم وجود خواهد داشت. هر یک از حالت‌های آزمایش، ۵ بار تکرار شده است و نتایج حاصله که زمان دستیابی نتیجه را برحسب ثانیه نشان می‌دهند، در جدول (۲) آورده شده‌اند. معیار توقف الگوریتم، تکرار تا ۵۰۰ نسل در نظر گرفته شده است.

جدول ۲: آزمایشات انجام شده در ۴ سطح جدول (۲)

	نرخ تقاطع ۰.۱ (-)	نرخ تقاطع ۰.۸ (+)
نرخ جهش ۰.۱ (-)	۵,۲۴۱۶	۵,۳۶۶۴
	۵,۳۰۴۰	۵,۲۲۶۰
	۵,۲۷۲۸	۵,۲۱۰۴
	۵,۱۹۴۸	۵,۱۳۲۴
	۵,۰۳۸۸	۵,۲۴۱۶
نرخ جهش ۰.۵ (+)	۵,۵۵۳۶	۵,۸۰۳۲
	۵,۶۱۶۰	۵,۵۸۴۸
	۵,۶۱۶۰	۵,۶۱۶۰
	۵,۵۳۸۰	۵,۵۲۲۴
	۵,۷۸۷۶	۵,۷۷۲۰

میانگین و واریانس آزمایشات انجام‌شده در هر یک از حالات، در جدول (۳) محاسبه شده است. اثرات (تفاوت

بازبینی قرار گرفته و در صورت نقض شدن، به وسیله ایجاد حداقل تغییرات در ترتیب ژن‌ها، نسبت به اصلاح آن اقدام می‌شود.

ارزیابی کروموزوم‌ها در فاز دوم

برای ارزیابی هر کروموزوم ابتدا باید آن را به یک زمانبندی موجه تبدیل کرد. می‌دانیم که برای یک لیست فعالیت، یک طرح تولید زمانبندی سری (SGS سری) می‌تواند به طور مستقیم به عنوان شیوه کدشکنی به کار رود و از این طریق زمانبندی به دست آید [۲۸].

استفاده از طرح تولید زمانبندی سریال، بدین صورت است که ابتدا فعالیتی که در لیست فعالیت (کروموزوم) مورد نظر در اولین مکان قرار گرفته و بیشترین اولویت را دارد، انتخاب و با در نظر گرفتن بیشترین تأخیر زمانی میان زمان‌های شروع، کمترین تأخیر زمانی میان زمان‌های شروع و میزان در دسترس منبع تجدیدشده در هر واحد زمانی، در زودترین زمان شروع آن، زمانبندی می‌کنیم و این کار را تا زمانبندی شدن همه فعالیت‌ها ادامه می‌دهیم. سپس هزینه کل ناشی از زودکرد و دیرکرد اتمام فعالیت‌ها نیز که نمایانگر تابع برازش الگوریتم خواهد بود، به راحتی و برای هر زمانبندی به دست آمده، قابل محاسبه خواهد بود. بدیهی است هر زمانبندی که کمترین میزان تابع برازش (هزینه کل) را داشته باشد، زمانبندی بهتری است.

نتایج محاسباتی

پس از طراحی یک الگوریتم، نوبت به آزمودن کارایی آن در حیظه مورد نظر می‌رسد. از آنجا که محققان مسائل زمانبندی با محدودیت منابع، اغلب برای این منظور از مسائل نمونه کتابخانه مسائل زمانبندی پروژه که به اختصار PSpLib^{۲۵} نامیده می‌شود استفاده می‌کنند، بنابراین در این مقاله نیز از همین شیوه استفاده می‌شود. به این صورت که نتایج حاصل از الگوریتم با نتایج موجود در کتابخانه مسائل زمانبندی با منابع محدود، مقایسه شده است.

نتایج محاسباتی مربوط به الگوریتم فاز اول

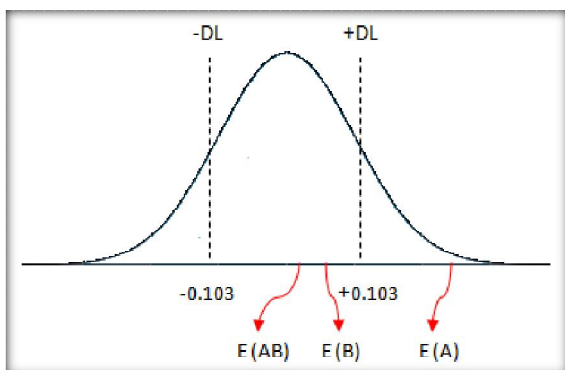
از آنجا که الگوریتم پیشنهادی فاز اول، مسئله تخصیص بهترین مد به هر فعالیت را مورد بررسی قرار می‌دهد. بنابراین آنچه برای ما مهم است، این است که بدانیم این

زمان پاسخ‌دهی الگوریتم، از آماره ای با توزیع t و با درجه آزادی ۱۶ (۱۶) = (۱) - (تعداد مشاهدات در هر ترکیب آزمایش)) × (تعداد ترکیب های آزمایش)) استفاده می‌کنیم. سطح معناداری این آزمون را ۰,۰۱ در نظر می‌گیریم. برای تعیین حدود تصمیم‌گیری از رابطه‌های (۱۱) و (۱۲) استفاده می‌کنیم:

$$t_{0.01,16} = 2.583 \quad (11)$$

$$DL = \pm t \cdot S_{effect} = \pm 2.583 \times 0.04 = \pm 0.103 \quad (12)$$

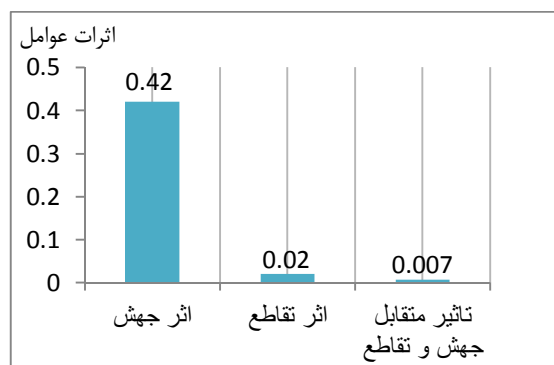
پس از رسم حدود تصمیم‌گیری در شکل (۲)، مشاهده می‌شود که تنها اثر عامل نرخ جهش، خارج از حدود تصمیم‌گیری می‌افتد، در نتیجه تنها عامل نرخ جهش، تأثیر معناداری بر زمان حصول نتیجه در الگوریتم ژنتیک فاز اول دارد.



شکل ۲: بررسی موقعیت اثرات عوامل نسبت به حدود تصمیم‌گیری

میانگین آزمایشات در سطح بالا و میانگین آزمایشات در سطح پایین) هر یک از عوامل نرخ جهش، نرخ تقاطع و نیز تأثیر متقابل نرخ های جهش و تقاطع شده‌اند.

همان گونه که مشاهده می‌شود، تأثیر نرخ جهش (۰,۴۲) بر زمان حصول نتیجه در الگوریتم فاز اول بسیار بیشتر از تأثیر عامل نرخ تقاطع (۰,۰۲) و نیز تأثیر متقابل نرخ جهش و نرخ تقاطع (۰,۰۰۷) است. نمودار پارتو اثرات، شکل (۱)، تفاوت این اثرات را با وضوح بیشتری نشان می‌دهد:



شکل ۱: نمودار پارتو اثرات

پس از محاسبه میانگین واریانس‌های آزمایشات $\bar{S}^2 = 0.01125$ و انحراف معیار آزمایشات $\bar{S} = 0.106$ ، انحراف معیار اثرات با استفاده از رابطه زیر حاصل می‌شود:

$$S_{effect} = \bar{S} \sqrt{\frac{4}{N}} = 0.106 \sqrt{\frac{4}{20}} = 0.04 \quad (10)$$

در آزمون تشخیص، معنادار بودن تأثیر عوامل نرخ جهش، نرخ تقاطع و اثر متقابل نرخ جهش و نرخ تقاطع بر

جدول ۳: محاسبات مربوط به اندازه گیری اثرات هر عامل

اجرا	جهش	تقاطع	AB	نتایج آزمایشات					S ²	
				1	2	3	4	میانگین		
۱	-	-	+	۵,۲۴۱۶	۵,۳۰۴۰	۵,۲۷۲۸	۵,۱۹۴۸	۵,۰۳۸۸	۵,۲۱۰۴	۰,۰۱۴
۲	-	+	-	۵,۳۶۶۴	۵,۲۲۶۰	۵,۲۱۰۴	۵,۱۳۲۴	۵,۲۴۱۶	۵,۳۳۵۳	۰,۰۰۷
۳	+	-	-	۵,۵۵۲۶	۵,۶۱۶۰	۵,۶۱۶۰	۵,۵۳۸۰	۵,۷۸۷۶	۵,۶۲۲۲	۰,۰۰۹
۴	+	+	+	۵,۸۰۳۲	۵,۵۸۴۸	۵,۶۱۶۰	۵,۵۲۲۴	۵,۷۷۲۰	۵,۶۵۹۶	۰,۰۱۵
+	۵,۶۴۴۰	۵,۴۴۷۴	۵,۴۳۵							
-	۵,۲۲۲۸	۵,۴۱۶۳	۵,۴۲۸							
Δ	۰,۴۲	۰,۰۲	۰,۰۰۷							

جدول ۴: نتایج محاسباتی مربوط به مسئله‌هایی با اندازه 10j

شماره مثال	حد پایین	جواب حاصل از الگوریتم پیشنهادی		درصد انحراف از حد پایین		زمان محاسبه (ثانیه)		نتایج مربوط به جریمه دیرکرد و پاداش زودکرد	
		تعداد تکرار ۱۰۰۰	تعداد تکرار ۵۰۰	تعداد تکرار ۱۰۰۰	تعداد تکرار ۵۰۰	تعداد تکرار ۱۰۰۰	تعداد تکرار ۵۰۰	واحد جریمه (تعداد تکرار ۱۰۰۰)	زمان محاسبه (ثانیه)
Psp13	۴۰°	۴۴	۴۴	۰,۱۰۰۰	۰,۱۰۰۰	۴۰,۷۳۰۵	۳۱,۳۴۱۹	-۵۳۳	۴۶,۲۳۸۷
Psp107	۴۶°	۵۰	۵۰	۰,۰۸۶۹	۰,۰۸۶۹	۴۹,۵۴۵۹	۲۵,۰۲۲۶	۱۷۱	۴۹,۹۶۷۱
Psp50	۶۸°	۷۱	۷۱	۰,۰۴۴۱	۰,۰۴۴۱	۵۴,۰۶۹۹	۲۶,۱۷۷۰	۳۸۰	۵۲,۳۲۲۷
Psp64	۴۷°	۴۷	۴۷	۰,۰۰۰۰	۰,۰۰۰۰	۵۷,۴۳۹۶	۲۸,۶۷۳۰	۲۲۰	۵۸,۱۲۶۰
Psp26	۵۴	۷۰	۷۰	۰,۲۲۶۹	۰,۲۲۶۹	۵۳,۲۲۷۵	۲۶,۲۷۰۶	۳۹۲	۴۸,۸۱۲۷
Psp92	۸۱	۸۵	۸۵	۰,۰۴۹۳	۰,۰۴۹۳	۴۸,۶۰۹۹	۲۴,۵۰۷۸	۱۳۳	۴۴,۱۲۴۶
Psp237	۴۹	۶۱	۶۳	۰,۲۴۴۸	۰,۲۸۵۸	۴۹,۸۱۷۳۵	۲۵,۱۱۶۲	۲۲۹	۵۰,۳۵۷۱
Psp125	۴۲	۵۹	۵۹	۰,۴۰۴۷	۰,۴۰۴۷	۵۲,۵۵۶۷	۲۶,۴۴۲۲	۳۰۰	۵۱,۸۳۹۱
Psp100	۴۴°	۶۱	۶۱	۰,۳۸۶۳	۰,۳۸۶۳	۴۹,۸۵۷۹	۲۵,۹۱۱۸	۴۹۵	۴۹,۷۹۵۵
Psp17	۳۶	۵۲	۵۳	۰,۴۴۴۴	۰,۴۷۲۲	۵۱,۷۱۴۳	۲۶,۰۰۵۴	۴۱۷	۵۱,۳۸۶۷

* - پاسخ بهینه

جدول ۵: نتایج محاسباتی مربوط به مسئله‌هایی با اندازه 20j

شماره مثال	حد پایین حد بالا - درصد انحراف	جواب حاصل از الگوریتم پیشنهادی		درصد انحراف از حد پایین		زمان محاسبه (ثانیه)		نتایج مربوط به جریمه دیرکرد و پاداش زودکرد	
		تعداد نسل ۱۰۰۰	تعداد نسل ۵۰۰	تعداد نسل ۱۰۰۰	تعداد نسل ۵۰۰	تعداد نسل ۱۰۰۰	تعداد نسل ۵۰۰	واحد جریمه (تعداد تکرار ۵۰۰)	زمان محاسبه (ثانیه)
Psp65 (GA)	۶۸ ٪۳۵,۲۹-۹۲	۸۴	۸۷	۰,۲۳۵۲	۰,۲۷۹۴	۳۲۵,۰۹۰	۱۶۳,۵۶۷	۱۴۷۲	۱۶۵,۷۶۶۷
Psp70 (BB)	۷۹ ٪۴۸,۱۰-۱۱۷	۱۲۴	۱۲۴	۰,۵۶۹۶	۰,۵۶۹۶	۳۳۳,۲۰۲	۱۶۲,۳۶۵۸	۲۳۵۹	۱۶۴,۸۳۰۷
Psp224	۸۳°	۱۰۶	۱۰۶	۰,۲۷۷۱	۰,۲۷۷۱	۲۹۴,۸۴۱۹	۱۴۸,۵۲۸۶	۲۵۵۶	۱۵۱,۲۱۳۴
Psp154 (BB)	۸۵ ٪۴۰,۰۰-۱۱۹	۱۳۱	۱۳۱	۰,۵۴۱۱	۰,۵۴۱۱	۳۱۸,۹۷۵	۱۵۷,۹۳۵	۲۴۴۶	۱۵۹,۰۷۴۲
Psp220 (BB)	۷۴ ٪۵۲,۷۰-۱۱۳	۱۲۴	۱۲۴	۰,۶۷۵۶	۰,۶۷۵۶	۳۱۰,۱۶۱۲	۱۵۷,۳۸۹۴	۲۳۴۸	۱۵۶,۰۱۶۶

* - پاسخ بهینه

جدول ۶: نتایج محاسباتی مربوط به مسئله‌هایی با اندازه 30j

شماره مثال	حد پایین حد بالا - درصد انحراف	جواب حاصل از الگوریتم پیشنهادی		درصد انحراف از حد پایین		زمان محاسبه (ثانیه)		نتایج مربوط به جریمه دیرکرد و پاداش زودکرد	
		تعداد نسل ۱۰۰۰	تعداد نسل ۵۰۰	تعداد نسل ۱۰۰۰	تعداد نسل ۵۰۰	تعداد نسل ۱۰۰۰	تعداد نسل ۵۰۰	واحد جریمه (تعداد تکرار ۲۵۰)	زمان محاسبه (ثانیه)
Psp129 (BB)	۸۳ ٪۷۴,۷۰-۱۴۵	۱۶۵	۱۶۵	۰,۹۸۷۹	۰,۹۸۷۹	۹۹۳,۸۱۲۴	۵۰۸,۵۷۸۹	۵۴۶۴	۲۴۳,۱۷۴۴
Psp247 (GA)	۱۰۶ ٪۶۵,۰۹-۱۷۵	۱۶۷	۱۶۷	۰,۵۷۵۴	۰,۵۷۵۴	۹۷۲,۰۲۶۶	۵۰۲,۴۷۹۲	۴۵۵۴	۲۴۱,۰۰۵۹
Psp261	۱۸۴°	۱۸۴	۱۸۷	۰,۰۰۰۰	۰,۰۱۶۳	۹۵۴,۵۵۴۵	۴۹۰,۷۷۹۱	۵۳۴۳	۲۳۸,۱۰۴۳
Psp157	۱۱۰	۱۷۰	۱۷۲	۰,۵۴۵۵	۰,۵۶۳۶	۹۹۶,۰۰۴۰	۴۹۹,۸۴۲۸	۷۴۰۲	۲۴۳,۰۳۴۰
Psp216	۱۴۵	۱۶۷	۱۶۹	۰,۱۵۱۷	۰,۱۶۵۵	۹۸۵,۰۰۵۹	۴۸۶,۵۵۱۵	۵۰۵۴	۳۷۶,۰۰۵۹

* - پاسخ بهینه

نتایج محاسباتی مربوط به الگوریتم فاز دوم

برای بررسی الگوریتم ژنتیک پیشنهادی فاز دوم، تعداد ۱۰ مسئله نمونه با ۱۲ فعالیت (که از این تعداد ۱۰ فعالیت، غیرمجازی هستند)، ۵ مسئله نمونه با ۲۰ فعالیت غیر مجازی و ۵ مسئله نمونه دیگر با ۳۰ فعالیت غیر مجازی به طور تصادفی از کتابخانه مسائل زمانبندی و از دسته مسائل RCPSP/max انتخاب شده‌اند که این مسئله‌ها نیز توسط نرم‌افزار ProGen/Max تولید شده‌اند (جدول ۴، ۵ و ۶). همانگونه که مشاهده می‌شود، الگوریتم پیشنهادی در چندین مورد بهترین جواب موجود در کتابخانه مسائل زمانبندی را بهبود بخشیده است و جواب‌های به دست آمده قابلیت ارسال به سایت کتابخانه مسائل زمانبندی را دارند.

در ستون‌های آخر از سمت چپ در هر یک از جداول ۴، ۵ و ۶، نتایج مربوط به جریمه دیرکرد و پاداش زودکرد تحویل فعالیت‌ها آورده شده است. از آنجا که حداکثر زمان مطلوب اتمام هر فعالیت، جریمه هر واحد دیرکرد و پاداش هر واحد زودکرد فعالیت‌ها در هر پروژه‌ای بر اساس شرایط آن پروژه و نظر کارفرما تعیین می‌شود، بنابراین برای اینکه مبنای مقایسه‌ای میان الگوریتم ارائه‌شده در این مقاله و تحقیقات بعدی فراهم باشد، به این ترتیب عمل کرده‌ایم که جریمه‌های دیرکرد و پاداش‌های زودکرد را اعدادی تصادفی بین ۱ تا ۵ مقدار داده‌ایم و مقادیر h_z را نیز به طور تصادفی در حد فاصل میان ۱ تا حد پایین متناظر با مسئله (در مورد مسائل جدول ۴) و حد بالای متناظر با هر مسئله (در باره مسائل جدول ۵ و ۶) انتخاب کرده‌ایم. سپس نتایج حاصله را در ستون ذکر شده در هر یک از سه جدول، آورده‌ایم.

نتیجه‌گیری و ارائه پیشنهادهایی برای تحقیقات بعدی

در حیطه مسائل زمانبندی پروژه، مسئله MRCPSp/max از جمله کلی‌ترین و دشوارترین مسائل به شمار می‌رود. می‌توان یک مسئله MRCPSp/max را با کمک الگوریتم ژنتیک به مسئله بسیار ساده‌تری که در آن، فعالیت‌ها فقط در یک مد اجرا می‌شوند، تبدیل کرد. این کار سرعت رسیدن به جواب

بهینه مسئله را بالا خواهد برد و از پیچیدگی‌های حل مستقیم MRCPSp/max خواهد کاست. در این مقاله، مدلی که از مشتقات مسئله MRCPSp/max به شمار می‌آید، مورد بررسی قرار گرفت. برای حل این مسئله از الگوریتم ژنتیکی با کروموزوم‌های نمایانگر بردار مدهای تخصیص یافته و بردار مقدار باقیمانده منابع تجدیدناپذیر کمک گرفتیم و با اعمال عملگرهای تقاطع و جهش، سعی در بهبود کروموزوم‌ها داشتیم و در نهایت، پس از تکرار نسل‌های کروموزوم به تعداد معین، بهترین بردار مد که کمترین طول مسیر بحرانی را ایجاد می‌کرد را به دست آوردیم. پس از کسب این نتیجه، مسئله MRCPSp/max به یک مسئله RCPSP/max تبدیل شده است و با استفاده از الگوریتم ژنتیک دیگری، اقدام به یافتن بهترین جواب برای این مسئله خواهیم کرد. در تحقیقات آینده می‌توان با ایجاد تغییراتی در مدل مطرح شده در این مقاله، نزدیکی بیشتری با دنیای واقعی ایجاد کرد. برای مثال می‌توان با توجه به خواست کارفرما و نیز شرایط پروژه، معیارهای ارزیابی عملکرد دیگری را نیز در نظر گرفت و توازنی بین آن معیارها و تابع هدف مطرح‌شده در این مقاله برقرار کرد. همچنین می‌توان برای حل مدل، از الگوریتم‌های ترکیبی استفاده و نتایج حاصله را با نتایج الگوریتم پیشنهادی در این مقاله مقایسه کرد. الگوریتم پیشنهادشده در این پژوهش را نیز می‌توان با اعمال تغییراتی در ساختار کروموزوم‌ها و بهبود شیوه نمایش جواب‌ها، همچنین ایجاد تغییراتی در عملگرهای تقاطع و جهش، ارتقاء بخشید. برای تسریع حصول جواب نیز، می‌توان الگوریتم را به گونه‌ای طراحی کرد که در ابتدا و برای ساده‌تر شدن مراحل انجام الگوریتم، منبع/منابع تجدیدنشده را حذف کند. (یک منبع تجدیدنشده را زائد گویند، اگر جمع بیشترین نیازهای فعالیت‌ها به این منبع، در هیچ صورتی از میزان در دسترس و مجاز آن منبع تجاوز نکند).

قدردانی

نویسندگان مقاله از آقای شهرام شادرخ و خانم نیلوفر هدایت به دلیل الهام‌بخشی ایده این تحقیق سپاسگزارند.

مراجع

- 1- Dorndorf, U., Pesch, E. and Phan-Huy, T. (2000). "A time oriented branch and bound algorithm for resource constrained project scheduling with generalised precedence constraints." *Management Science Informs*, Vol. 46, PP. 1365–84.
- 2- Barrios, A., Ballestín, F. and Valls, V. (2011). "A double genetic algorithm for the MRCPSp/max." *Computers & Operations Research*, Vol. 38, PP. 33– 43.
- 3- Bartusch, M., Möhring, R. H. and Radermacher, F.J. (1988) "Scheduling project networks with resource constraints and time windows." *Annals of Operations Research*, Vol. 16, PP. 199 -240.
- 4- De Reyck, B. and Herroelen, W. (1998). "A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence constraints." *European Operational Research*, Vol. 111, PP. 152–174.
- 5- Fest, A., Möhring, R. H., Stork, F. and Uetz, M. (1998). "Resource constrained project scheduling with time windows: a branching scheme based on dynamic release dates." Technical report 596, TU Berlin, Germany.
- 6- Franck, B., Neumann, K. and Schwindt, Ch. (2001). "Truncated branch and bound, schedule construction, and schedule improvement procedures for resource constrained project scheduling." *OR Spektrum-Berlin Springer*, Vol. 23, PP. 297–324.
- 7- Franck, B. and Selle, T. (1998). "Metaheuristics for the resource-constrained project scheduling problem with schedule-dependent time windows." Technical Report WIOR-546, Universitat Karlsruhe, Germany.
- 8- Cesta, A., Oddi, A. and Smith, S.F. (2002). "A Constraint-Based Method for Project Scheduling with Time Windows." *Journal of Heuristics*, Vol. 8(1), PP. 109-136.
- 9- Smith, T.B. and Pyle, J.M. (2004). "An Effective Algorithm For Project Scheduling With Arbitrary Temporal Constraints." *Proc., 19th national conf. on Artificial intelligence*, AAAI Press, PP. 544-549.
- 10- Ifeyinwa, M.J.O. and Wei, S. (2013). "Project Scheduling Under Resource Constraints: A Recent Survey." *IJERT, International Journal of Engineering Research & Technology*, Vol. 2, Issue. 2, PP. 1-20.
- 11- Fu, N., Chu Lau, H., Varakantham, P. and Xiao, F. (2012). "Robust Local Search for Solving RCPSP/max with Durational Uncertainty." *Journal of Artificial Intelligence Research*, Vol. 43, PP.43-86.
- 12- Schutt, A., Feydy, T., Stuckey, P.J. and Wallace, M.G. (2013). "Solving RCPSP/max by lazy clause generation." *Journal of Scheduling*, Vol. 16, Issue. 3, PP. 273-289.
- 13- Bianco, L. and Caramia, M. (2011). "A new lower bound for the resource-constrained project scheduling problem with generalized precedence relations." *Computers & Operations Research*, Vol. 38, Issue. 1, Pages 14–20.
- 14- Ballestín, F., Barrios, A. and Valls, V. (2011). "An evolutionary algorithm for the resource-constrained project scheduling problem with minimum and maximum time lags." *Journal of Scheduling*, Vol. 14, Issue. 4, PP. 391-406.
- 15- De Reyck, B. (1998). "Scheduling projects with generalized precedence relations: exact and heuristic procedures." PhD thesis, Katholieke Universiteit Leuven, Belgium.
- 16- Nonobe, K. and Ibaraki, T. (2003). "A tabu search algorithm for a generalized resource constrained project scheduling problem." *Proc., 5th Int. Conf. on meta-heuristics*, MIC2003, PP. 55-1–55-6.
- 17- Heilmann, R. (2001). "Resource-constrained project scheduling: a heuristic for the multi-mode case." *OR Spektrum*, Vol. 23, PP. 335–57.

- 18- Heilmann, R. (2003). "A branch-and-bound procedure for the multi-mode resource- constrained project scheduling problem with minimum and maximum time lags." *European Journal of Operational Research*, Vol. 144, PP. 348–65.
- 19- Artigues, C., Lopez, P. and Ayache, P.D. (2005). "Schedule generation schemes for the job-shop problem with sequence dependent setup times: dominance properties and computational analysis." *Annals of Operations Research*, Vol. 138, PP. 21-52.
- 20- Seyed-Hosseini, S. M., Sabzehparvar, M. and Nouri, S. (2007). "A Genetic algorithm and a model for the resource constrained project scheduling problem with multiple crushable modes." *IUST, International Journal of Engineering Science*, Vol. 18, No.3-4, PP. 7-18.
- 21- Sabzehparvar, M. and S. M Seyed-Hosseini. (2008). "A mathematical model for the multi-mode resource-constrained project scheduling problem with mode dependent time lags." *The Journal of Supercomputing*, Vol. 44, No. 3, PP. 257–273.
- 22- Jędrzejowicz, P. and Ratajczak-Ropel, E. (2011). "A-Team for solving MRCPSP/max problem." *Proc., 5th Int. conf. on Agent and multi-agent systems, technologies and application*, PP. 466-475.
- 23- Jędrzejowicz, P. and Ratajczak-Ropel, E. (2011). "Double-Action Agents Solving the MRCPSP/Max Problem." *Computer Science*, Vol. 6923, PP. 311-321.
- 24- Alcaraz, J., Maroto, C. and Ruiz, R. (2003). "Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms." *Journal of the Operational Research*, Vol. 54, PP. 614–626.
- 25- Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. 1st. Ed. Oxford Univ. Press.
- 26- Hartmann, S. (1998). "A competitive genetic algorithm for resource-constrained project scheduling." *Naval Research Logistics*, Vol. 45, PP. 733–750.
- 27- Kolisch, R. and Hartmann, S. (1999). "Heuristic algorithms for solving the resource-constrained project scheduling problem: classification and computational analysis, Project Scheduling: Recent Models-algorithms and Applications." Kluwer Academic Publishers, Berlin, PP. 147–178.
- 28- Chen, W., Shi, Y., Teng, H., Lan, X. and Hu, L. (2010). "An efficient hybrid algorithm for resource-constrained project scheduling." *Information Sciences*, Vol. 180, Issue. 6, PP.1031-1039.

واژه‌های انگلیسی به ترتیب استفاده در متن

- 1 - Minimum time lags
- 2 - Maximum time lags
- 3 - Generalized Precedence Relations
- 4 - Temporal Constraints
- 5 - Time Lags
- 6 - Time windows
- 7 - Resource-constrained Project Scheduling Problem With Minimum and Maximum time lags (RCPSP/MAX)
- 8- Multi-mode Resource-constrained Project Scheduling Problem With Minimum and Maximum time lags (MRCPSP/MAX)
- 9- Double Genetic Algorithm(DGA)
- 10 - Chromosome
- 11 - Fitness Function
- 12 - Crossover
- 13 - Mutation
- 14 - Mode Vector
- 15 - Remaining Vector
- 16- Population size (pop-size)

-
- 17- Cycle Of Positive Length
 - 18- The Cycle Structure-based Crossover Operator
 - 19 - Two point Crossover
 - 20 - Random Key Representation
 - 21 - Priority Rule Representation
 - 22 - Precedence Feasible Priority List
 - 23 - Partially- mapped Crossover (PMX)
 - 24 - Shift Mutation
 - 25 - <http://129.187.106.231/psplib>
 - 26 - Complete Factorial Design
 - 27 - Redundant
-