

ارائه سه الگوریتم فرا ابتکاری توسعه یافته به منظور حل مسئله هزینه دسترس پذیری منابع با اهداف کمینه سازی زمان اتمام پروژه و مجموع هزینه های منابع به صورت همزمان

امیرعباس نجفی^{۱*}، مسعود ارجمند^۲

۱. دانشیار دانشکده مهندسی صنایع، دانشگاه صنعتی خواجه نصیرالدین طوسی

۲. کارشناس ارشد مهندسی صنایع دانشکده مهندسی صنایع و مکانیک، دانشگاه آزاد اسلامی

(تاریخ دریافت: ۹۵/۰۳/۱۶، تاریخ دریافت روایت اصلاح شده: ۹۵/۰۴/۱۷، تاریخ تصویب: ۹۵/۰۹/۰۹)

چکیده

در این پژوهش، مسئله هزینه دسترس پذیری منابع با در نظر گرفتن دو هدف بررسی شده است. در این مسئله، روابط پیش نیازی از نوع پایان-شروع با تأخیر زمانی صفر و منابع تجدیدپذیر در نظر گرفته شده اند. برای این مسئله ابتدا یک مدل برنامه ریزی مختلط صحیح ارائه شده و سپس سه الگوریتم توسعه یافته ژنتیک چندهدفه مرتب سازی نامغلوب، الگوریتم ژنتیک رتبه بندی نامغلوب و الگوریتم چندهدفه اجتماع ذرات برای حل این مسئله ارائه شده است. به منظور ارزیابی کارایی الگوریتم ها، یک مجموعه از مسائل با ابعاد متنوع از کتابخانه دیجیتال مسائل زمان بندی پروژه انتخاب شده و نتایج محاسباتی با روش های مختلفی بررسی شده است. در انتها، از یک روش تصمیم گیری چندمعیاره به نام روش تاپسیس برای رتبه بندی الگوریتم ها و تحلیل حساسیت آن ها با توجه به میزان اهمیت هر یک از معیارها استفاده شده است.

واژه های کلیدی: الگوریتم بهینه سازی چندهدفه اجتماع ذرات، الگوریتم ژنتیک مرتب سازی نامغلوب، بهینه سازی چندهدفه، زمان بندی پروژه با محدودیت منابع، هزینه دسترس پذیری منابع.

مقدمه

مسئله زمان بندی پروژه با محدودیت منابع یکی از مسائل مهم مورد مطالعه در حوزه زمان بندی پروژه با درجه پیچیدگی NP-hard است [۱]. متغیرهای تصمیم و تابع هدف این مسئله در حالت استاندارد، به ترتیب زمان شروع فعالیت های پروژه و کمینه سازی زمان اتمام پروژه هستند. در مسئله زمان بندی پروژه با محدودیت منابع، برای اجرای فعالیت های پروژه به منابعی نیاز است که به سقف مشخصی محدودند. همچنین، فعالیت ها باید به نحوی برنامه ریزی شوند که در هیچ مقطع زمانی بیش از سطح در دسترس نیاز نباشد. در بسیاری از تحقیقات، روش های حل دقیق و ابتکاری برای حل این مسئله ارائه شده است [۱۰-۲].

در مدل های ارائه شده در سال های اخیر، سطح مورد نیاز منابع به صورت متغیر تصمیم در نظر گرفته شده است که به آن مسئله هزینه دسترس پذیری منابع (RACP)^۱ می گویند. در این مسائل، تعیین سطح منابع

یکی از متغیرهای تصمیم مسئله است که باید با مشخص شدن زمان بندی پروژه تعیین شود. مورینگ [۱۱] مسئله سرمایه گذاری در منابع را معرفی کرد. او برای حل این مسئله یک روش حل دقیق پیشنهاد داد و نیز اثبات کرد که از لحاظ پیچیدگی یک مسئله NP-hard است. همچنین، وی روش حل دقیق بر پایه الگوریتم های گراف توسعه داد و مثال هایی را با شانزده فعالیت و چهار منبع با روش ارائه شده حل کرد. نوبل [۱۲] یک الگوریتم شاخه و کران برای مسئله سرمایه گذاری در منابع با فرض وجود محدودیت به وجود آورد که جواب بهینه را به دست می آورد. در مورد نوع تأخیری RIP/max^۲ در دامنه الگوریتم های ابتکاری می توان به کار نیومن و زیمرمن [۱۳] اشاره کرد. در زمینه وجود داشتن حداقل و حداکثر در مسئله در اتمام پروژه می توان به پژوهش های اکپان [۱۴] و کیمز [۱۵] اشاره کرد. کیمز روشی را برای به دست آوردن کران های بالا و پایین برای این مسئله ایجاد کرد که یکی روش

فعالیت‌های پروژه به k نوع منبع تجدیدپذیر نیاز است و هزینه استخدام منبع نوع k برابر C_k می‌باشد. همچنین، میزان مورد نیاز فعالیت i به منبع نوع k در هر واحد زمانی برابر r_{ik} است. EF_i و LF_i به ترتیب بیانگر زودترین زمان پایان و دیرترین زمان پایان فعالیت i است. توالی انجام دادن فعالیت‌ها و سطح تدارک منابع به نحوی تعیین می‌شود که ضمن رعایت محدودیت پیش‌نیازی و محدودیت منابع، توابع هدف مسئله بهینه شوند. به منظور مدل‌سازی ریاضی از متغیرهای تصمیم زیر استفاده شده است.

$$\left. \begin{array}{l} \text{فعالیت } i \text{ در زمان } t \text{ به اتمام رسیده است} \\ \text{در غیر این صورت} \end{array} \right\} X_{it}$$

R_k : میزان در دسترس منبع نوع

مدل ریاضی

در این مدل، مسئله زمان‌بندی پروژه با دو هدف کمینه‌سازی زمان اتمام پروژه و همچنین کمینه‌سازی مجموع هزینه منابع در نظر گرفته شده است.

$$\text{Min } Z_1 = \sum_{t=EF_n}^{EF_n} t \cdot x_{nt} \quad (1)$$

$$\text{Min } Z_2 = \sum_{k=1}^{\rho} C_k \cdot R_k \quad (2)$$

S.T.

$$\sum_{t=EF_j}^{LF_j} x_{jt} = 1 \quad ; \quad j = 1, \dots, n \quad (3)$$

$$\sum_{t=EF_j}^{LF_j} x_{jt} \cdot (t - d_j) \geq \sum_{t=EF_i}^{LF_j} t \cdot x_{it} \quad ; \quad j = 1, \dots, n, \quad i \in p_j \quad (4)$$

$$\sum_{j=1}^n \sum_{q=t}^{t+d_j-1} r_{jk} \cdot x_{jq} \leq R_k \quad ; \quad k = 1, \dots, \rho, \quad t = 1, \dots, T \quad (5)$$

$$x_{jt} \in \{0,1\} \quad ; \quad j = 1, \dots, n, \quad t = ES_j, \dots, LS_j \quad (6)$$

$$R_k \geq 0, \quad ; \quad k = 1, \dots, \rho \quad (7)$$

ساده‌سازی لاگرانژ و دیگری روش ایجاد ستون است. البته این روش‌ها جواب‌های موجه نیز تولید می‌کنند. شادرخ و کیانفر [۱۶] نیز مسئله سرمایه‌گذاری در منابع را بررسی کردند و الگوریتم ژنتیکی برای حل آن ارائه دادند. نتایج محاسباتی شادرخ و کیانفر نشان داد الگوریتم ژنتیک توسعه داده شده به جواب‌های بسیار مناسبی منجر می‌شود. نجفی و نیاکی [۱۷] مسئله سرمایه‌گذاری در منابع را تحت وجود جریان‌های نقدی مثبت منفی و باهدف حداکثر کردن ارزش خالص فعلی نقدینگی پروژه معرفی کردند و الگوریتم ژنتیک را برای حل این مسئله پیشنهاد دادند.

در مجموع، تمام تحقیقات صورت گرفته در این زمینه با یک تابع هدف بررسی شده‌اند و به بهینه‌سازی در نظر گرفتن چند تابع هدف به صورت همزمان توجه نشده است. در این پژوهش، مسئله هزینه دسترس‌پذیری منابع با هدف کمینه‌سازی مدت زمان اتمام پروژه و کمینه‌سازی مجموع هزینه‌های تدارک منابع بررسی می‌شود. با توجه به اینکه دو هدف این مسئله ناسازگار هستند، مسئله یک مجموعه جواب دارد؛ بنابراین، برای ارزیابی جواب‌ها از معیارهای ارزیابی مسائل چندهدفه استفاده می‌شود. برای حل این مدل از سه الگوریتم توسعه یافته NSGA-II^۳، NRGA^۴ و MOPSO^۵ استفاده شده است.

در این راستا، در بخش بعد به بیان مسئله پرداخته می‌شود. سپس الگوریتم‌های فرا ابتکاری توسعه داده شده برای حل مسئله معرفی می‌شوند و جزئیات طراحی آن‌ها تشریح می‌شود. در ادامه، مسائل با ابعاد متفاوت از طریق الگوریتم‌ها حل می‌شود و عملکرد الگوریتم‌ها با استفاده از معیارهای ارزیابی و همچنین روش تاپسیس ارزیابی می‌شود. در انتها، جمع‌بندی مطالب ارائه می‌شود.

بیان مسئله

پروژه‌ای متشکل از n فعالیت را در نظر بگیرید که در آن فعالیت‌ها از ۱ تا n شماره‌گذاری شده‌اند. فعالیت‌های ۱ و n مجازی بوده و به ترتیب شروع و پایان پروژه را نشان می‌دهند. مدت زمان فعالیت i به صورت d_i تعریف می‌شود. روابط پیش‌نیازی فعالیت‌ها از نوع روابط پایان شروع با تأخیر زمانی صفر در نظر گرفته شده و مجموعه فعالیت‌های پیش‌نیاز فعالیت i با p_i نشان داده می‌شود. برای اجرای

لیست فعالیت‌ها یک جایگشت موجه از فعالیت‌ها با در نظر گرفتن روابط پیش‌نیازی آن‌هاست. در این لیست، ترتیب فعالیت‌ها از چپ به راست به نحوی است که تمام فعالیت‌های پیش‌نیاز یک فعالیت در سمت چپ آن فعالیت قرار بگیرند؛ یعنی هر فعالیت مانند J_a^1 باید بعد از مجموعه پیش‌نیازهای خود $P_{j_a^1}^A$ زمان بندی شود. لیست منابع نیز موجه خواهد بود؛ به عبارت دیگر، سطح هر کدام از منابع تعیین شده باید در رابطه ۹ صدق کند.

$$R_k^I \geq \max_{i=1, \dots, n} \{r_{ik}\} \quad ; \quad k = 1, \dots, \rho \quad (9)$$

برای تبدیل کروموزوم I به یک زمان بندی پروژه، از روش برنامه ریزی تولید زمان بندی موازی (SSS)^۹ استفاده شده است. این روش شامل n مرحله است که در هر مرحله یک فعالیت با توجه به روابط پیش‌نیازی و در دسترس بودن منابع در زودترین زمان ممکن زمان بندی می‌شود.

تولید کروموزوم‌ها

برای تولید سطوح تدارک منابع R_k^I برای هر کروموزوم I، اعداد تصادفی با تابع توزیع یکنواخت بین کران پایین \underline{R}_k و کران بالا \overline{R}_k تولید می‌شود و کران‌های بالا و پایین با استفاده از روابط ۱۰ و ۱۱ محاسبه می‌شوند:

i_{ik} : میزان نیاز فعالیت i به منبع نوع k در هر واحد زمانی

d_i : مدت زمان انجام دادن فعالیت i

T: مدت زمان تکمیل پروژه

$$\underline{R}_k = \max \left\{ \sum_{i=1}^n \frac{r_{ik} \cdot d_i}{T}, \max_{i=1, \dots, n} \{r_{ik}\} \right\} \quad (10)$$

$$\overline{R}_k = \sum_{i=1}^n r_{ik} \quad (11)$$

برای تولید بخش اول، لیست فعالیت‌های موجه، ابتدا یک جایگشت تصادفی از فعالیت‌ها ایجاد می‌شود. سپس یک سازوکار اصلاحی تمامی روابط پیش‌نیازی را برای تمامی فعالیت‌ها به ترتیب قرار گرفتن آن‌ها در لیست فعالیت‌ها بررسی می‌کند و لیست فعالیت‌های موجه تولید می‌شود.

رابطه ۱ بیانگر کمینه سازی مدت زمان انجام دادن پروژه است. رابطه ۲ نیز کمینه سازی مجموع هزینه استفاده از منابع را نشان می‌دهد. رابطه ۳ بیان می‌کند هر یک از فعالیت‌ها فقط یک بار ختم می‌شود و با توجه به اینکه فعالیت‌ها انقطاع پذیر نیستند، یک فعالیت فقط یک بار آغاز می‌شود. رابطه ۴ نشان دهنده روابط پیش‌نیازی است و بیان می‌کند در این مدل یک فعالیت تنها زمانی شروع می‌شود که تمام پیش‌نیازهای مربوط به آن فعالیت اجرا شده باشند. رابطه ۵ مربوط به وجود محدودیت منابع است و نشان می‌دهد میزان منابع در هر واحد زمان یک مقدار مشخص است و هر فعالیت زمانی آغاز می‌شود که به اندازه زمان آن، منابع مورد نیاز موجود باشد. رابطه ۶ نشان دهنده متغیر تصمیم این مسئله است که از نوع دوتایی است. رابطه ۷ دیگر متغیر تصمیم مسئله (میزان تعیین شده از هر یک از منابع) را نشان می‌دهد که مقداری صحیح و مثبت است.

الگوریتم‌های فرا ابتکاری

در این قسمت، ابتدا ساختار کروموزوم به کاررفته در الگوریتم‌های فرا ابتکاری و سپس جزئیات مورداستفاده برای هر یک از الگوریتم‌ها ارائه می‌شود.

ساختار جواب

طراحی یک کروموزوم مناسب، مهم‌ترین مرحله در پیاده سازی یک الگوریتم در فرایند حل مسئله است. در این تحقیق، هر کروموزوم به صورت یک سطر که شامل دو بخش است، در نظر گرفته شده است. بخش اول لیست فعالیت‌هاست. این لیست ترتیب انجام دادن فعالیت‌ها را نشان می‌دهد. بخش دوم آن نشان دهنده میزان تعیین شده سطح تدارک منابع است که در اینجا لیست منابع نامیده می‌شود. ساختار کروموزوم I به صورت رابطه ۸ است.

$$I = \left((j_1^I, \dots, j_n^I), (R_1^I, \dots, R_\rho^I) \right) \quad (8)$$

در رابطه ۸، j_n^I نشان دهنده یک فعالیت است که در جایگاه n قرار گرفته است و R_ρ^I نشان دهنده سطح تدارک منبع نوع ρ است که باید در زمان بندی به آن توجه شود.

الگوریتم ژنتیک چندهدفه بر مبنای مرتب‌سازی نامغلوب

یکی از بهترین الگوریتم‌ها بر مبنای جمعیت که برای حل مسائل چندهدفه ارائه شده است، الگوریتم NSGA-II است. دب [۱۸] این الگوریتم را ارائه داد. عملکرد این الگوریتم براساس دو مفهوم رتبه‌بندی و فاصله ازدحامی استوار است. انتخاب جمعیت جدید در هر گام این روش بر اصل غلبگی استوار است و با استفاده از نخبه‌گرایی و رتبه‌بندی جمعیت در هر گام حل، بهترین جواب‌های نامغلوب را انتخاب می‌کند و به گام بعد می‌رود.

مراحل کلی اجرای این الگوریتم به صورت زیر است:

گام ۱: جمعیت اولیه P_0 به اندازه N با جواب‌های تصادفی ایجاد کنید و $t = 0$ قرار دهید.

گام ۲: در صورت عدم برقراری شرط توقف، به P_t بازگردید.

گام ۳: با استفاده از عملگر انتخاب مسابقه‌ای دودویی، N والد از جمعیت P_t انتخاب کنید.

گام ۴: با اعمال عملگرهای تقاطع، جهش، جست‌وجوی محلی و حرکت روی جمعیت P_t ، جمعیت فرزندان Q_t به اندازه N ایجاد کنید.

گام ۵: قرار دهید $R_t = P_t \cup Q_t$.

گام ۶: از روش رتبه‌بندی نامغلوب برای تعیین مجموعه‌های پارتو $F_i, i = 1, 2, \dots$ در جمعیت R_t استفاده کنید.

گام ۷: قرار دهید $P_{t+1} = \emptyset$ و $i = 1$.

گام ۸: تا زمانی که $|P_{t+1}| + |F_i| < N$:

الف) جواب‌های مجموعه F_i را به جمعیت P_{t+1} اضافه کنید.

ب) قرار دهید $i = i + 1$.

گام ۹: جواب‌های مجموعه F_i را براساس فاصله ازدحام و به صورت نزولی مرتب کنید.

گام ۱۰: به اندازه $N - |P_{t+1}|$ از اولین جواب‌های F_i به جمعیت P_{t+1} انتقال دهید.

گام ۱۱: قرار دهید $t = t + 1$ و به گام ۲ بازگردید.

انتخاب والد

از عملگر انتخاب مسابقه‌ای دودویی برای انتخاب والدین و ایجاد فرزندان استفاده می‌شود. بدین صورت که ابتدا دو جواب به طور تصادفی از جمعیت انتخاب می‌شوند. جوابی که رتبه کمتری دارد، والد محسوب می‌شود. اگر دو جواب انتخاب شده رتبه یکسان داشته باشند، جوابی که فاصله ازدحام بیشتری دارد والد در نظر گرفته می‌شود. اگر فاصله ازدحامی این دو جواب نیز با هم برابر باشد، از بین آن‌ها یکی به صورت تصادفی انتخاب می‌شود.

عملگر تقاطع

عملگر تقاطع یک نقطه‌ای فقط روی بخش اول کروموزوم (لیست فعالیت‌ها) اعمال می‌شود. در این عملگر، ابتدا دو والد Pa_1 و Pa_2 انتخاب می‌شود. سپس یک عدد r در بازه $[0, 1]$ تا n (تعداد فعالیت‌ها) به تصادف انتخاب می‌شود. برای تولید فرزند اول CH_1 ، عناصر 1 تا فعالیت r از لیست فعالیت‌های والد اول و عناصر $r + 1$ تا n از والد دوم انتخاب می‌شود. برای تولید فرزند دوم CH_2 ، عکس این رویه استفاده می‌شود. در شکل ۱، یک مثال از نحوه کار این عملگر نشان داده می‌شود. در این مثال، $r = 2$ است. در انتها، توالی‌های جدید به توالی موجه تبدیل می‌شوند.

Pa_1	۱۴	۳۲۵۶	۱	CH	۱۴۲۵۶۳
Pa_2	۳۱	۲۵۶۴	۲	CH	۳۱۴۲۵۶

شکل ۱. نحوه عملگر تقاطع یک نقطه‌ای

عملگر جهش

عملگر جهش روی یک جواب که با استفاده از عملگر انتخاب مسابقه‌ای دودویی انتخاب شده است، اعمال می‌شود. در این پژوهش، سه عملگر مختلف استفاده شده است. شایان ذکر است یکی از آن‌ها به تصادف روی جوابی که قرار است این جهش روی آن اعمال شود، انتخاب می‌شود.

جابه‌جایی

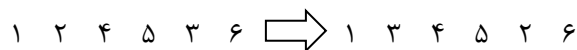
این عملگر، دو ژنی را که به طور تصادفی انتخاب شده‌اند با یکدیگر جابه‌جا می‌کند. شایان ذکر است این عملگر نیز فقط روی بخش اول کروموزوم (لیست فعالیت‌ها) اعمال می‌شود. پس از اعمال عملگر جهش، باید شرط موجه‌بودن

آن کران به جای آن مقدار قرار داده می شود. با در نظر گرفتن محدودیت منابع جدید، دوباره مسئله زمان بندی می شود. با توجه به تصادفی بودن ماهیت این عملگر و امکان تولید جواب با برازندگی کم، از یک عملگر انتخاب مسابقه ای استفاده شده است. در این روش به یک اندازه از قبل تعیین شده، عملگر جست و جوی محلی روی کروموزوم انتخاب شده اعمال می شود و جواب های متفاوتی ایجاد می شود. سپس این جواب ها با استفاده از مفهوم غلبه مقایسه می شوند و یک جواب نامغلوب از بین آن ها به عنوان خروجی این عملگر تعیین می شود. در صورتی که چند جواب نامغلوب موجود باشد، یکی از آن ها به تصادف انتخاب می شود.

عملگر حرکت

این عملگر روی زمان های شروع فعالیت ها اعمال می شود. ابتدا یک کروموزوم با استفاده از عملگر انتخاب مسابقه ای دودویی انتخاب می شود. سپس یکی از انواع منابع به تصادف انتخاب می شود و تعیین می شود که در چه دوره زمانی بیشترین استفاده از این منبع صورت گرفته است. به فعالیت هایی که در این بازه شروع شده اند، فعالیت های مؤثر گفته می شود. سپس یکی از فعالیت های مؤثر به تصادف انتخاب می شود. در این مرحله از یک عملگر انتخاب مسابقه ای استفاده می شود و این فعالیت در هر تکرار به اندازه تصادفی جابه جا می شود و زمان بندی دوباره انجام می گیرد و مقادیر برازندگی آن ها ثبت می شود. سپس جواب ها مقایسه می شوند و آن جوابی که برازندگی بهتری دارد، خروجی در نظر گرفته می شود؛ برای مثال، یک پروژه با شش فعالیت و یک منبع در نظر گرفته می شود و توالی ۱۲۳۴۵۶ یک جواب موجه برای این مسئله است. شکل ۵ نشان دهنده میزان استفاده از منبع برای این جواب است. همان طور که مشاهده می کنید، در بازه زمانی ۲ بیشترین استفاده از منبع صورت گرفته است. فعالیت هایی که در این بازه زمانی آغاز شده اند، فعالیت های ۲ و ۳، فعالیت های مؤثر هستند.

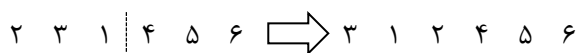
جواب دوباره بررسی شود و در صورت موجه نبودن، جواب اصلاح شود. نحوه انجام دادن این کار در شکل ۲ مشاهده می شود.



شکل ۲. نحوه عملکرد عملگر جابه جایی

الحاق

این عملگر ابتدا یک ژن را به طور تصادفی انتخاب می کند. سپس یک موقعیت در لیست فعالیت ها به صورت تصادفی انتخاب می شود و ژن انتخاب شده به موقعیت جدید فرستاده می شود. در انتها، شرط موجه بودن جواب از طریق سازوکار اصلاحی بررسی می شود و در صورت غیرموجه بودن، جواب اصلاح می شود. شکل ۳ نحوه کار عملگر الحاق را روی توالی یک جواب نشان می دهد.



شکل ۳. نحوه عملکرد عملگر الحاق

معکوس

این عملگر نیز همانند عملگر جابه جایی دو ژن را به صورت تصادفی انتخاب می کند و تمامی ژن های بین آن دو ژن را به صورت وارونه قرار می دهد. سپس شرط موجه بودن بررسی می شود و در صورت لزوم سازوکار اصلاحی، جواب موجه را ایجاد می کند. نحوه کار این عملگر در شکل ۴ نشان داده می شود.



شکل ۴. نحوه عملکرد عملگر معکوس

عملگر جست و جوی محلی

این عملگر روی بخش دوم کروموزوم (میزان سطح منابع) اعمال می شود. در این روش، یک کروموزوم با استفاده از عملگر انتخاب مسابقه ای دودویی انتخاب می شود. از بخش دوم این کروموزوم یک ژن به تصادف انتخاب R_K^I می شود. در مرحله بعد، ۱ واحد از سطح منبع انتخاب شده کسر $R_K^I - 1$ می شود. در نهایت، سطح منبع مربوطه مشخص می شود. اگر سطح تعیین شده از کران پایین بگذرد، مقدار

الگوریتم ژنتیک رتبه‌بندی نامغلوب

NRGA نیز یکی از الگوریتم‌های ژنتیک چندهدفه است که براساس رتبه‌بندی نامغلوب به بهینه‌سازی مسائل چندهدفه می‌پردازد. استفاده از چرخ رولت در فرایند انتخاب والد موجب شده است این الگوریتم نسبت به سایر الگوریتم‌های مشابه سرعت همگرایی بیشتری داشته باشد. مراحل این الگوریتم مشابه با NSGA-II است، با این تفاوت که ابتدا برای هر مجموعه پارتو، یک مقدار احتمال براساس رابطه ۱۲ تخصیص داده می‌شود. با در نظر گرفتن احتمالات تخصیص داده شده، مجموعه‌ای انتخاب می‌شود. مجموعه‌ای که رتبه کمتری دارد، احتمال انتخاب بیشتری دارد. سپس از مجموعه انتخاب شده، والدین انتخاب می‌شوند.

$$p_k = \frac{1}{\sum_k \frac{1}{Rank_k}} \quad (12)$$

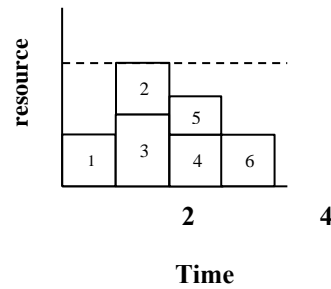
در رابطه ۱۲، p_k و $Rank_k$ به ترتیب بیانگر احتمال انتخاب و رتبه مجموعه k است. فرض کنید مجموعه F انتخاب شده است، حال باید یک والد از این مجموعه انتخاب کرد. به هر جواب مجموعه F ، یک مقدار احتمال براساس رابطه ۱۲ تخصیص داده می‌شود. براین اساس، جوابی که فاصله ازدحام بیشتری دارد، احتمال انتخاب شدن آن نیز بیشتر است.

$$p_i = \frac{CD_i}{\sum_i CD_i} \quad (13)$$

در رابطه ۱۳، p_i بیانگر احتمال انتخاب جواب نامغلوب i است. شایان ذکر است نحوه نمایش جواب در این الگوریتم مانند نحوه نمایش جواب برای NSGA-II است.

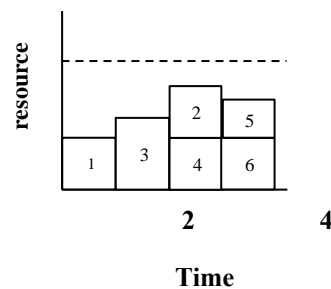
عملگر تقاطع

در این الگوریتم، از عملگر تقاطع دونقطه‌ای استفاده شده است. سازوکار به کاررفته در این عملگر مشابه عملگر تقاطع الگوریتم NSGA-II است و تنها تفاوت آن تعداد نقاط انتخابی است.



شکل ۵. جواب موجه برای مسئله با شش فعالیت

بنابراین، عملگر جابه‌جایی یکی از این دو فعالیت را به تصادف انتخاب می‌کند و به میزانی تصادفی جابه‌جا می‌کند؛ برای مثال، اگر فعالیت ۲ انتخاب شود و به میزان یک واحد زمان شروع آن تغییر پیدا کند، جواب زیر حاصل می‌شود. همان‌طور که در شکل ۶ نشان داده می‌شود، میزان منبع استفاده شده کاهش پیدا کرده است.



شکل ۶. استفاده از عملگر حرکت

شرط توقف

ممکن است شرایط توقف متفاوتی برای یک الگوریتم به کار رود. یک آستانه از پیش تعیین شده برای مقدار تابع هدف و شرط رسیدن به آن مقدار آستانه ممکن است شرط خاتمه در نظر گرفته شود، اما با توجه به اینکه الگوریتم‌های بر پایه ژنتیک احتمالی هستند و معمولاً هیچ تضمینی برای رسیدن به آستانه از پیش تعیین شده وجود ندارد، این شرط ممکن است هیچ‌گاه تأمین نشود. در نتیجه، شرط خاتمه‌ای که برای الگوریتم‌ها در این تحقیق در نظر گرفته شده است، رسیدن به تعداد تکرارهای از قبل تعیین شده است.

عملگر جهش

است منعکس می کند و در کل با استفاده از رابطه ۱۵ تعریف می شود:

$$\begin{aligned} \bar{v}_1(t) = & W\bar{v}_1(t-1) \\ & + C_1 r_1 (\bar{x}_{pbest_1} - \bar{x}_1(t)) \\ & + C_2 r_2 (\bar{x}_{gbest_1} - \bar{x}_1(t)) \end{aligned} \quad (15)$$

تعدد انواع منابع K انتخاب می شود. سپس به تعداد r در بازه ۱ تا r در این عملگر یک عدد تصادفی r در بازه ۱ تا r تا تعداد انواع منابع K انتخاب می شود. سپس به تعداد r به صورت تصادفی از منابع انتخاب می شود و سطح تعیین شده آن ها ۱ واحد کاهش می یابد و زمان بندی مجدداً با سطح منابع جدید انجام می گیرد. تفاوت دوم این است که در الگوریتم NPGA از عملگر رقابت مسابقه ای استفاده نشده است.

عملگر جست و جوی محلی

عملکرد این عملگر مشابه عملگر به کاررفته در الگوریتم NSGA-II است و فقط دو تفاوت دارد؛ تفاوت اول در این است که در این عملگر یک عدد تصادفی r در بازه ۱ تا r تا تعداد انواع منابع K انتخاب می شود. سپس به تعداد r به صورت تصادفی از منابع انتخاب می شود و سطح تعیین شده آن ها ۱ واحد کاهش می یابد و زمان بندی مجدداً با سطح منابع جدید انجام می گیرد. تفاوت دوم این است که در الگوریتم NPGA از عملگر رقابت مسابقه ای استفاده نشده است.

عملگر حرکت

این عملگر نیز مانند عملگر جابه جایی NSGA-II است، با این تفاوت که در این عملگر یک عدد تصادفی x در بازه ۱ تا x تا تعداد انواع منابع K انتخاب می شود. سپس به تعداد x به صورت تصادفی از منابع انتخاب می شود. سپس یک فعالیت مؤثر مربوط به هر کدام از منابع انتخاب می شود و در یک انتخاب مسابقه ای بهترین حالت جابه جایی آن تعیین می شود و مسئله مجدداً زمان بندی می شود.

الگوریتم بهینه سازی چندهدفه اجتماع ذرات

این الگوریتم بر مبنای الگوریتم اجتماع ذرات PSO^{۱۰} نوشته شده است و اصطلاحات به کاررفته در الگوریتم های PSO چندهدفه مانند الگوریتم PSO تک هدفه است. در این الگوریتم، موقعیت هر ذره طبق تجربه شخصی و تجربه همسایگی اش تغییر می کند. فرض کنید $\bar{x}_i(t)$ موقعیت ذره p_i را در هر مرحله زمانی t مشخص می کنند. سپس موقعیت ذره این p_i توسط افزودن یک سرعت $v_i(t)$ به موقعیت فعلی، با استفاده از رابطه ۱۴ تغییر می کند.

$$\bar{x}_i(t) = \bar{x}_i(t-1) + \bar{v}_i(t) \quad (14)$$

بردار سرعت، اطلاعاتی را که به صورت جمعی مبادله شده

رابطه ۱۵ از سه بخش تشکیل شده است: در بخش اول، سرعت فعلی از تغییر سرعت قبلی به دست می آید. بخش دوم قسمت شناختی است که نمایانگر بهترین خاطره شخصی ذره است که از تجربیات پروازهای گذشته نگهداری می شود. بخش سوم قسمت اجتماعی است که نمایانگر بهترین خاطره جمعی ذرات است که از تجربیات پروازهای دسته جمعی ذرات به دست می آید. کوئلو و همکاران [۱۹] این الگوریتم را پیشنهاد داده اند و براساس ایده داشتن آرشو خارجی است که در آن هر ذره تجربیات پروازش را بعد از هر سیکل پرواز ذخیره می کند. به روزرسانی های آرشو خارجی با در نظر گرفتن خطوط شبکه (Grid lines) فضای فونوتیپ که بر حسب مقادیر تابع هدف هر ذره تعریف شده است انجام می گیرد. در این رویکرد، از یک عملکرد جهش نیز استفاده می شود که هم روی ذرات درون جمعیت و هم روی دامنه متغیرهای تصمیم مسئله انجام می گیرد.

عملگر جهش

همان طور که در بخش قبل بیان شد، هنگامی که یک ذره، موقعیتش را به روز می کند، جهشی هوشمند صورت می گیرد. با وجود این، در بعضی مواقع به هوشمند نبودن که کندی و ابرهات [۲۰] آن را حماقت نامیده اند- نیاز است. در کل، وقتی جمعیت را کم می شود؛ یعنی سرعت ذرات تقریباً صفر می شود، جمعیت توانایی تولید جواب های جدید را برای خارج شدن از این وضعیت ندارد. این رفتار نشان می دهد کل جمعیت در بهینه محلی ای گیر کرده است و توانایی فرار از آن را ندارد. این عملگر ابتدا روی تمام

$$S = \sqrt{\frac{1}{|Q|} \sum_{i=1}^{|Q|} (d_i - \bar{d})^2} \quad (18)$$

در رابطه ۱۸، $|Q|$ بیانگر اندازه آرشیو پارتو است و مقادیر d_i و \bar{d} را می‌توان به ترتیب از روابط ۱۹ و ۲۰ محاسبه کرد. الگوریتمی که میزان این معیار کمتر باشد، مطلوب‌تر است.

$$d_i = \min_{k \in Q \wedge k \neq i} \sum_{m=1}^M |f_m^i - f_m^k|, \quad (19)$$

$$\bar{d} = \sum_{i=1}^{|Q|} \frac{d_i}{|Q|}, \quad (20)$$

ه) بیشترین گسترش (D)^{۱۴}: این معیار نشان می‌دهد چه میزان از جواب‌های یک مجموعه پارتو در فضای جواب توزیع شده است که از رابطه ۲۱ محاسبه می‌شود. بزرگ‌تر بودن مقدار این معیار، نشان‌دهنده تنوع مناسب جواب‌های مجموعه پارتو است.

$$D = \sqrt{\sum_{m=1}^M (\max_{i=1:|Q|} f_m^i - \min_{i=1:|Q|} f_m^i)^2} \quad (21)$$

و) عدم یکنواختی سطح پارتو (NPF)^{۱۵}: این معیار برای اندازه‌گیری غیریکنواختی توزیع منحنی پارتو استفاده می‌شود و مقدار آن از رابطه ۲۲ محاسبه می‌شود.

$$NPF = \sqrt{\frac{\sum_i (\frac{d_i}{\bar{d}} - 1)^2}{|A| - 1}} \quad (22)$$

در رابطه ۲۲، مقدار d_i و \bar{d} از روابط ۲۳ و ۲۴ به دست می‌آید.

$$d_i = \min_{k \in \Omega \wedge k \neq i} \sqrt{\sum_{m=1}^2 (f_m^i - f_m^k)^2} \quad (23)$$

$$\bar{d} = \sum_{i=1}^A \frac{d_i}{|A|} \quad (24)$$

مقدار کمتر این معیار همانند معیار فاصله‌گذاری نشان‌دهنده برازندگی بیشتر آن است.

ذرات اعمال می‌شود و با افزایش تعداد تکرارها، تعداد ذراتی که این عملگر روی آن‌ها اعمال می‌شود، کاهش می‌یابد. شایان ذکر است این عملگر در هر مرحله یکی از دو بخش کروموزوم را به‌طور تصادفی انتخاب می‌کنند و آن را تغییر می‌دهد. سپس سازوکار اصلاحی اعمال می‌شود.

معیارهای ارزیابی

در مسائل چندهدفه در پایان حل، مجموعه‌ای از جواب‌ها ایجاد می‌شود که باید با توجه به این مجموعه در خصوص عملکرد الگوریتم اظهار نظر شود. همگرایی به جواب‌های بهینه پارتو و فراهم کردن چگالی و تنوع در میان مجموعه جواب‌های به دست آمده دو هدف اصلی هر الگوریتم تکاملی چندهدفه است، ولی این دو معیار تا حدودی در تضاد با یکدیگرند و معیاری که بتواند به تنهایی و به صورت مطلق در مورد عملکرد الگوریتم‌ها تصمیم بگیرد، وجود ندارد؛ بنابراین، برای ارزیابی عملکرد الگوریتم‌ها، از معیارهای زیر استفاده می‌شود:

الف) زمان محاسبات (CPU Time): الگوریتمی که زمان محاسبات کمتری داشته باشد، مطلوب‌تر است.

ب) تعداد جواب‌ها در پارتو (NPS)^{۱۱}: تعداد جواب‌های نامغلوب در مجموعه پارتو به دست آمده برای هر مسئله را نشان می‌دهد و هر مقدار تعداد این نقاط بیشتر باشد به این معنی است که الگوریتم کارایی بیشتری دارد.

ج) فاصله از نقطه ایده‌آل (MID)^{۱۲}: این معیار برای اندازه‌گیری میزان نزدیکی به سطح بهینه پارتو واقعی به کار می‌رود که از رابطه ۱۶ محاسبه می‌شود:

$$MID = \frac{\sum_{i=1}^n c_i}{n} \quad (16)$$

در این رابطه، n تعداد جواب‌ها در مجموعه بهینه پارتو است و c_i فاصله اقلیدسی هر عضو از مجموعه پارتو از نقطه ایده‌آل است که از رابطه ۱۷ محاسبه می‌شود:

$$c_i = \sqrt{(f_{1i} - f_1^*)^2 + (f_{2i} - f_2^*)^2 + \dots + (f_{mi} - f_m^*)^2} \quad (17)$$

د) فاصله‌گذاری (S)^{۱۳}: نشان‌دهنده میزان قرار گرفتن جواب‌ها به‌طور یکنواخت در کنار هم است که از رابطه ۱۸ محاسبه می‌شود.

یکدیگر مقایسه کند، برای مقایسه نتایج این سه الگوریتم، آن‌ها دوبه دو باهم مقایسه شده‌اند. شایان ذکر است در قسمت‌هایی که از خط تیره استفاده شده است، دو الگوریتم اختلاف معناداری نداشتند.

تحلیل آماری نتایج الگوریتم‌های فرا ابتکاری

در این بخش، تحلیل آماری نتایج هر سه الگوریتم صورت گرفته است که در جدول ۱ به آن اشاره می‌شود. تحلیل آماری نتایج در نرم‌افزار Minitab16 پیاده‌سازی شده است. با توجه به اینکه آزمون تی زوجی می‌تواند دو جامعه را با

جدول ۱. مقایسه الگوریتم‌ها با استفاده از آزمون تی زوجی

	NPS	MID	S	D	NPF	CPU Time
NSGA-II/ NPGA	NPGA	NSGA-II	-	NPGA	-	NPGA
NSGA-II/ MOPSO	NSGA-II	NSGA-II	-	-	-	MOPSO
NPGA/ MOPSO	NPGA	-	-	NPGA	NPGA	-

جدول ۲. متوسط درصد انحراف از معیارها به ازای تمام مسائل

	NPS (%)	MID (%)	S (%)	D (%)	NPF (%)	CPU Time (%)
NSGA-II	۱۱/۶۲۶۹	۲/۰۰۵۳	۳۵/۲۴۳۳	۱۵/۴۱۳۳	۳۲/۶۰۳۱	۳۲۸/۹۷۷۴
NPGA	۱/۶۱۸۲	۶/۶۲۱۷	۲۸/۷۴	۴/۲۸۰۴	۱۷/۰۳۸۴	۳۲/۵۵۴۵
MOPSO	۱۹/۶۹۷۲	۷/۴۶۹۸	۴۲/۳۰۶۱	۲۱/۸۲۴۸	۴۰/۱۴۶۶	۱۷/۹۸۵۸

به دست می‌آید و از این طریق الگوریتم‌ها بررسی می‌شود. نتایج مربوطه در جدول ۲ بیان شده است؛ برای مثال، الگوریتم NSGA-II در شاخص NPS به طور متوسط ۱۱/۶۲۶۹ درصد از بهترین جواب به ازای مسائل حل شده فاصله دارد. جدول ۲ نشان می‌دهد الگوریتم NPGA به طور متوسط در معیارهای NPS, S, D, NPF کمترین انحراف را از بهترین جواب دارد. الگوریتم‌های NSGA-II و MOPSO نیز به ترتیب در معیارهای MID و CPU Time کمترین انحراف نسبی را دارند.

آنالیز حساسیت براساس تعداد فعالیت‌ها

در این بخش، میزان حساسیت معیارهای ارزیابی نسبت به تغییر فعالیت‌ها در هر یک از الگوریتم‌های ابتکاری ارائه شده ارزیابی شده است.

مقایسه نتایج الگوریتم‌ها براساس درصد

انحراف نسبی

از آنجاکه مقدار تابع هدف در هر مسئله متفاوت است و به طور مستقیم قابل استفاده نیست، از درصد انحراف نسبی (RPD)^{۱۶} برای ارزیابی بهتر جواب‌ها استفاده می‌شود. مقدار آن با استفاده از رابطه ۲۵ محاسبه می‌شود.

$$RPD = \left| \frac{ALG_{sol} - best_{sol}}{best_{sol}} \right| * 100 \quad (25)$$

در رابطه ۲۵، ALG_{sol} جواب الگوریتم و $best_{sol}$ بهترین جواب در بین جواب‌های به دست آمده از سه الگوریتم است. جواب این رابطه به صورت درصد بیان می‌شود. هرچقدر میزان این درصد انحراف کمتر باشد، الگوریتم عملکرد بهتری داشته است. این شاخص برای تمامی مسائل به دست می‌آید. سپس به ازای هر معیار ارزیابی، متوسط درصد انحراف نسبی برای تمام الگوریتم‌ها

جدول ۳. مقایسه معیارهای ارزیابی به تفکیک تعداد فعالیت‌ها

		NPS (%)	MID (%)	S (%)	D (%)	NPF (%)	CPU Time (%)
J10s	NSGA-II	۸/۶۸۵۹	۲/۴۷۲۴	۴۲/۳۰۱۵	۱۲/۴۱۲۸	۴۱/۶۲۶۰	۵۰۸/۰۳۵۶
	NRGA	۵/۰۰۶۹	۴/۲۳۴۴	۵۶/۱۰۲۲	۴/۳۶۹۹	۳۷/۳۸۶۵	۹۸/۰۱۸۸
	MOPSO	۳۰/۳۱۳۹	۲/۷۱۴۱	۵۵/۱۰۱۳	۲۳/۵۶۵۷	۲۹/۵۸۱۹	.
J20s	NSGA-II	۱۰/۱۴۹۶	۳/۰۴۱۲	۳۴/۲۸۶۲	۸/۸۱۸۷	۱۸/۴۸۶۵	۲۸۰/۳۱۸۵
	NRGA	.	۳/۵۵۲۹	۱۴/۳۶۱۱	۶/۶۵۸۹	۶/۳۲۹۸	۹/۴۱۰۹
	MOPSO	۲۲/۶۰۶۸	۲/۴۷۱۲	۱۷/۲۸۲۶	۳۴/۹۳۳۷	۲۴/۳۹۲۵	۱/۳۶۵۴
J30s	NSGA-II	۱۷/۵۲۱۸	.	۳۵/۹۷۲۱	۲۶/۶۸۶۱	۴۳/۹۵۰۵	۲۰۷/۹۹۳۳
	NRGA	.	۱۲/۹۱۵۶	۱۵/۹۵۰۷	۱/۵۹۲۰	۴/۹۹۵۱	.
	MOPSO	۱۲/۰۷۹۹	۱۷/۶۲۵۹	۴۵/۶۵۱۰	۱۳/۴۷۳۶	۵۶/۱۹۸۵	۱۷/۳۵۴۵

جواب ایده‌آل است که هوانگ و یون [۲۱] برای اولین بار ارائه دادند. در این روش، گزینه انتخاب شده باید کوتاه‌ترین فاصله را از جواب ایده‌آل و دورترین فاصله را از ناکارآمدترین جواب داشته باشد. از محاسن روش یادشده این است که می‌تواند شاخص‌های کمی و کیفی را برای تصمیم‌سازی تلفیق کند.

به‌طور خلاصه در روش تاپسیس، ماتریس $m \times n$ تصمیم‌گیری ارزیابی می‌شود که دارای m گزینه و n معیار در ماتریس سنجش است. در این الگوریتم فرض می‌شود که هر شاخص و معیار در ماتریس تصمیم‌گیری مطلوبیت افزایشی یا کاهشی یکنواخت دارد؛ به بیان دیگر، مقادیر زیادتری که معیارها در این ماتریس کسب می‌کنند اگر از نوع سود باشد، هرچه مقدارش بیشتر باشد مطلوبیت بالاتری دارد و اگر از نوع هزینه باشد، مطلوبیت پایین‌تری دارد.

به‌منظور محاسبات ریاضی باید تمام مقادیر نسبت داده‌شده به معیارها از نوع کمی باشد و در صورت کیفی بودن معیارها، باید آن‌ها را به مقادیر کمی تبدیل کرد. در این پژوهش، مدل ارائه‌شده توسط ۳ الگوریتم فرا ابتکاری حل شده است. همچنین، ۶ معیار برای مقایسه کارایی الگوریتم‌ها معرفی شده است؛ بنابراین، ماتریس تصمیم‌گیری ما به ازای هر مدل یک ماتریس ۳×۶ است.

با توجه به نتایج روش تاپسیس، الگوریتم NRGA با مطلوبیت ۰/۹۱۵۰ بهترین عملکرد را داشته است و بعد از آن به‌ترتیب الگوریتم‌های MOPSO و NSGA-II با مطلوبیت ۰/۵۵۶۶ و ۰/۲۷۷۰ قرار می‌گیرند.

با توجه به جدول ۳، می‌توان به این نتیجه رسید که در معیار NPS، الگوریتم NSGA-II با افزایش تعداد فعالیت‌ها به میزان بیشتری از بهترین جواب منحرف می‌شود و عملکرد بدتری دارد، اما الگوریتم‌های NRGA و MOPSO با افزایش تعداد فعالیت‌ها عملکرد بهتری از خود نشان می‌دهند. در معیار MID، در تمام الگوریتم‌ها نمی‌توان روند مشخصی را شناسایی کرد. در نتیجه، می‌توان نتیجه گرفت این شاخص نسبت به تعداد فعالیت‌ها مستقل است. در معیار S, D و NPF نیز همین شرایط برقرار است، اما در مورد معیار CPU Time می‌توان گفت وابسته به تعداد فعالیت‌هاست. در الگوریتم‌های NRGA و NSGA-II با افزایش تعداد فعالیت‌ها، میزان انحراف جواب‌ها از بهترین جواب کاهش پیدا می‌کند، اما در الگوریتم MOPSO این روند برعکس است و با افزایش فعالیت‌ها، جواب‌ها بدتر می‌شوند.

با توجه به نتایج، نمی‌توان با قطعیت گفت کدام الگوریتم بهترین عملکرد را داشته است. پس برای مقایسه الگوریتم‌ها باید از یک روش تصمیم‌گیری بر مبنای چند شاخص (MADM)^{۱۷} استفاده کرد. برای این کار از روش تاپسیس استفاده شده است.

مقایسه کارایی الگوریتم‌ها با استفاده از روش تاپسیس

روش تاپسیس یکی از روش‌های تصمیم‌سازی تکنیکی و قوی برای اولویت‌بندی گزینه‌ها از طریق شبیه‌کردن به

نتیجه گیری

اینکه هیچ کدام از الگوریتم‌ها در تمامی معیارها بهترین عملکرد را نداشته‌اند، از روش تاپسیس برای مقایسه نتایج استفاده شده است. نتایج این روش نشان می‌دهد الگوریتم NSGA-II و سپس به ترتیب MOPSO و NPGA بهترین عملکرد را داشته‌اند. همچنین، نتایج آنالیز حساسیت نشان می‌دهد الگوریتم NSGA-II در مسائل بزرگ‌تر عملکرد بهتری دارد، اما الگوریتم MOPSO در مسائل کوچک‌تر عملکرد بهتری دارد

در پژوهش حاضر، مسئله هزینه دسترسی پذیری منابع با دو هدف کمینه سازی همزمان مدت زمان اتمام پروژه و هزینه تدارک منابع مدل سازی شده است. برای حل مسئله، سه الگوریتم فرا ابتکاری توسعه یافته ارائه شده و سپس مسائل با ابعاد مختلف از کتابخانه دیجیتال انتخاب و حل شده است. برای مقایسه عملکرد الگوریتم‌ها از شش معیار ارزیابی استفاده شده است. نتایج نشان می‌دهد الگوریتم NPGA در معیارهای NPS, S, D, NPF بهترین عملکرد را دارد و الگوریتم‌های MOPSO و NSGA-II به ترتیب در معیارهای

مراجع

1. Stalk, G., Evans, P. and Shulman, L. (1992). "Competing on capabilities: The new rules of corporate strategy", *Harvard Business Review*, Vol. 70, No. 2, PP. 57-66
2. Blazewicz, J., Lenstra, J. K. and Rinnooy Kan A. H. G. (1983). "Scheduling subject to resource constraints", *Discrete Appl Math*, Vol. 1, No. 5, PP. 11-24.
3. Paraskevopoulos, D. C., Tarantilis, C. D. and Ioannou, G. (2012). "Solving project scheduling problems with resource constraints via an event list-based evolutionary algorithm", *Expert. Syst. Appl*, Vol. 39, No. 4, PP. 3983-3994.
4. Hartmann, S. and Kolisch, R. (2000). "Experimental evaluation of state-of-the-art heuristics for the resource constrained project scheduling problem", *Eur. J. Oper. Res.*, Vol. 127, No. 2, PP. 394-407.
5. Afshar-Nadjafi, B., Karimi, H., Rahimi, A. and Khalili, S. (2013). "Project scheduling with limited resources using an efficient differential evolution algorithm", *Journal of King Saud University*, In Press, Corrected Proof.
6. Zamani, R. (2013). "A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem", *Eur. J. Oper. Res.*, Vol. 229, No. 2, PP. 552-559.
7. Jairo, R., Torres, M. and Edgar, G. F. (2010). "Carolina Pirachicán-Mayorga, project scheduling with limited resources using a genetic algorithm", *Int. J. project. Manage.*, Vol. 28, No. 6, PP. 619-628.
8. Hartmann, S. and Briskorn, D. A. (2010). "Survey of variants and extensions of the resource constrained project scheduling problem", *Eur. J. Oper. Res.*, Vol. 207, No. 1, PP. 1-14.
9. Agarwal, R., Colak, S. and Erenguc, S. (2011). "A Neurogenetic approach for the resource constrained project scheduling problem", *Comput. Oper. Res.*, Vol. 38, No. 1, PP. 44- 50.
10. Fang, C. and Wang, L. (2012). "An effective shuffled frog-leaping algorithm for resource constrained project scheduling problem", *Comput. Oper. Res.*, Vol. 39, No. 5, PP. 890-901.
11. Koné, O. (2012). "New approaches for solving the resource constrained project scheduling problem", *4OR*, Vol. 10, No. 1, PP. 105-106.
12. Mohring, R. H. (1984). "Minimizing Costs of Resource Requirements in project Networks subject to a Fix Completion Time", *Op Research*, Vol. 32, No. 1, PP. 89-120.
13. Nubel, H. (1999). "A Branch and Bound procedure for Resource Investment problem subject to Temporal Constraints", Technical Report 574, University Karlsruhe, Germany.
14. Neumann, K. and Zimmermann, J. (1997). "Resource Leveling for project with schedule dependent time windows", Technical Report 508, Institut für wirtschaftstheorie and Operations Research, Universität Karlsruhe.

14. Akpan, E. O. P. (1997). "Optimal Resource Determination for project scheduling", *Project Planning and Control*, Vol. 8, No. 5, PP. 462-468.
15. Kimms, A. (2001). "Minimizing the Net Present Value of a project under Resource Constraints Using a Lagrangian Relaxation Based Heuristic with tight upper bound", *Annual of operation Research*, Vol. 102, No. 1, PP. 221-236.
16. Najafi, A. A. and Niaki, S. T. A. (2006). "A genetic algorithm for resource investment problem with discount cash flow", *Applied Mathematical and Computation*, Vol. 183, No. 2, PP. 1057-1070.
17. Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*, Wiley, Chichester.
18. Shadrokh, S. and Kianfar, F. (2007). "A genetic algorithm for resource investment project scheduling problem, tardiness permitted with penalty", *Eur J Oper Res*, Vol. 181, No. 1, PP. 86-101.
19. Coello-Coello, C. A., Van Veldhuizen, D. A. and Lamont, G. B. (2002). *Evolutionary algorithms for solving multi-objective problems*, Kluwer Academic Publishers.
20. Kennedy, J. and Eberhart, R. C. (1995). "Particle swarm optimization", *Proceeding of IEEE International Conference on Neural Networks*, Piscataway, NJ, Seoul, Korea, Vol. IV, PP. 1942-1948
21. Hwang, C. L. and Yoon, K. (1981). *Multiple attribute decision making*, Springer.9.

واژه‌های لاتین به ترتیب استفاده در متن

1. Resource Availability Cost Problem
2. Resource Investment Problem
3. Non-dominated Sorting Genetic Algorithm
4. Non-dominated Ranking Genetic Algorithm
5. Multi-objective Particle Swarm Optimization
6. Earliest Finish Time
7. Latest Finish Time
8. Predecessor
9. Serial Scheduling Scheme
10. Particle Swarm Optimization
11. Number of Pareto Solution
12. Mean Ideal Distance
13. Spacing
14. Diversity
15. Non-uniform Pareto Front
16. Relative Percentage Deviation
17. Multi Attribute Decision Making