

یک الگوریتم PSO اصلاح شده برای به حداقل رساندن کل هزینه های منابع در

MRCPSP

ما یک مسئله برنامه ریزی پروژه محدود در-منبع چندحالتی را با روابط اولویت پایان-به-آغاز در میان فعالیت های پروژه، با در نظر گرفتن هزینه های منبع تجدیدپذیر و تجدیدناپذیر معرفی می نماییم. فرض می کنیم که منابع تجدیدپذیر، کرایه می شوند و در تمام دوره های زمانی پروژه در دسترس نیستند. به بیانی دیگر، یک تاریخ آماده اجباری و تاریخ مقرر برای هر نوع منبع تجدیدپذیر وجود دارد به طوری که هیچ منبعی قبل از تاریخ آماده آن استفاده نمی شود. هرچند به واسطه پرداخت هزینه جریمه، بعد از تواریخ وظیفه، منابع مجاز برای استفاده نیستند. این مسئله، یک مسئله برنامه ریزی پروژه محدود در-منبع چندحالتی با مینیمم سازی کلی هزینه جریمه دیرکرد منبع وزندهی شده نامیده می شود (MRCPSP-TWRTPC)، که در آن، برای هر فعالیت، هر دو الزامات منبع تجدیدپذیر و تجدیدناپذیر وابسته به حالت فعالیت هستند. برای این مسئله، ما یک الگوریتم فراابتکاری مبتنی بر رویکرد بهینه سازی اصلاح شده ازدحام ذرات را ارائه می دهیم که توسط Tchomt'e and Gourgand معرفی شده است و از یک قاعده اصلاح شده برای جابجایی ذرات استفاده می کند. ما یک قاعده اولویت بندی را برای فعالیت ها و چندین بهبود و روش های جستجوی موضعی ارائه می دهیم. نتایج آزمایشی، اثربخشی و کارایی الگوریتم پیشنهادی را برای مسئله مورد نظر نشان می دهند.

۱. مقدمه

مسئله برنامه ریزی پروژه با محدود در منابع (RCPSP)، برنامه ریزی زمانبندی فعالیت های پروژه تحت روابط تقدم و همچنین محدودیت های منبع تجدید پذیر با هدف به حداقل رساندن طول زمان این پروژه است. هر فعالیت غیرپیشگیرانه در RCPSP را می توان در یک حالت واحد انجام داد. برای کسب اطلاعات بیشتر در مورد RCPSP و روش های راه حل، به کار Demeulemeester و Herroelen [1] اشاره می کنیم. در RCPSP چند حالتی

(MRCPSP)، مجموعه ای از حالات مجاز را می توان برای هر فعالیت تعریف نمود که توسط یک مدت زمان ثابت و منابع مورد نیاز مرتبط مشخص می شوند. در این مقاله، ما MRCPSP را با هدف به حداقل رساندن هزینه های کل تمام منابع در نظر می گیریم. دو نوع از منابع، تجدید پذیر و تجدید ناپذیر، در نظر گرفته می شوند. هزینه منبع تجدید ناپذیر از یک فعالیت، تابعی از الزامات منبع آن است که توسط حالت آن تعیین می شود. منابع تجدید پذیر محدود اجاره ای هستند و هر منبع تجدید پذیر در یک دوره زمانی از پیش تعیین شده پی در پی توسط زمان آماده و تاریخ مقرر در دسترس است و قبل از زمان آماده در دسترس نیست. با این حال، هر منبع تجدید پذیر می تواند پس از تاریخ مقرر آن با هزینه ی جریمه ی تأخیر استفاده شود. زمانی که هزینه اجاره برای هر منبع قابل تجدید تثبیت می شود، نیازی به گنجاندن آن در تابع هدف وجود ندارد و تنها هزینه جریمه دیرکرد برای هر منبع تجدید پذیر در نظر گرفته می شود. MRCPSP تحت به حداقل رساندن کل هزینه های منابع (RCPSPTWRTPC)، یک مسئله قابل کاربرد و یک نسخه اصلاح شده از MRCPSP است که در آن همه مفروضات و محدودیت های MRCPSP حفظ می شوند، اما تابع هدف متفاوت است. ما فرض می کنیم که چند منبع تجدید پذیر وجود دارند مانند منابع بسیار متخصص با سطوح مهارت های بالا، انواع خاصی از جرثقیل، و ماشین آلات حفاری تونل که باید از شرکت های دیگر ارائه دهنده این نوع از منابع اجاره شوند. از آنجا که این منابع تجدید پذیر محدود در پروژه های دیگر به کار گرفته می شوند، یک تاریخ آماده دیکته شده و نیز یک تاریخ مقرر برای هر یک از آنها وجود دارد به طوری که هیچ منبعی نمی تواند قبل از تاریخ آماده خود در دسترس باشد، اما این منابع، بعد از تواریخ مقرر خود، با پرداخت هزینه ی جریمه، بسته به نوع منبع، قابل استفاده هستند. همچنین، ما فرض می کنیم که چند منبع غیرقابل تجدید مانند بودجه، مواد، انرژی، و یا منابع دیگری وجود دارند که در طول پروژه مصرف می شوند.

Ranjbar و همکاران (۲) این مسئله را با حالت تک برای هر فعالیت و در دسترس بودن یک واحد برای هر نوع از منابع تجدید پذیر، بدون در نظر گرفتن منابع غیر قابل تجدید مورد مطالعه قرار دادند. آنها این مسئله را مسئله برنامه ریزی پروژه با محدود در منابع، به حداقل رساندن هزینه کلی جریمه دیرکرد منبع وزندهی شده (RCPSPTWRTPC)

(TWRTPC) نامیدند که یک فرم بسط یافته از مسئله برنامه ریزی پروژه با محدود در منابع (RCPSP) است. آنها یک الگوریتم GRASP فوق ابتکاری را با یک روش شاخه و مرز برای حل این مسئله توسعه دادند.

مسئله ای که ما در اینجا مورد مطالعه قرار می دهیم، تعمیم مسئله معرفی شده توسط Ranjbar و همکاران (۲) با دیدگاه واقع بینانه تر از هزینه های منابع، توسط در نظر گرفتن هر دو هزینه منابع تجدید پذیر و تجدید ناپذیر است. ما این مسئله را مسئله برنامه ریزی پروژه چند حالتی با محدود در منابع، به حداقل رساندن هزینه کلی جریمه دیرکرد منبع وزندهی شده (MRCPSPTWRTPC) می نامیم.

چند روش دقیق و ابتکاری برای MRCPSP ارائه شده است. به عنوان مثال، ما می توانیم به روش شاخه و برش معرفی شده توسط Heilmann (3) و روش شاخه و حد توسعه یافته توسط Zhu و همکاران (۴) به عنوان دو روش قدرتمند دقیق اشاره نماییم. Zhang و همکاران. (۵) روش های بهینه سازی ازدحام ذرات (PSO) کلاسیک را برای مسائل برنامه ریزی پروژه محدود در منابع چند حالتی ارائه برای به حداقل رساندن مدت زمان پروژه های ساختمانی ارائه نمودند. Lova و همکاران (۶)، یک الگوریتم ابتکاری را بر اساس قوانین اولویت و پس از آن الگوریتم ژنتیک ترکیبی (۷) برای حل MRCPSP پیشنهاد دادند. Jarboui و همکاران (۸) یک ذره بهینه سازی ازدحام ترکیبی برای حل مساله برنامه ریزی پروژه با محدود در منابع چند حالتی استفاده می شود. Ranjbar و همکاران (۹) الگوریتم جستجوی پراکنده را برای مقابله با این مسئله، با استفاده از روش شناسی پیوند دادن دوباره مسیر روش به عنوان روش جستجوی موضعی آن توسعه داد. _10_ Van Peteghem and Vanhoucke الگوریتم ژنتیک را برای حل مسائل چندحالتی منابع-محدود برنامه ریزی پروژه پیشگیرانه و غیرپیشگیرانه ارائه نمودند. Kazemi and Tavakkoli-Moghaddam (۱۱) یک بهینه سازی ازدحام ذرات چند حالتی را توسعه دادند که با اپراتور ژنتیکی برای حل یک مسئله برنامه ریزی پروژه چند حالتی با محدود در منابع دو-هدفی با جریان های نقدی مثبت و منفی ترکیب می شود. تفاوت بین این مطالعه و مقالات قبلی اینست که همه آنها به حداقل رساندن طول زمانی پروژه را به عنوان تابع هدف در نظر گرفتند، اما، در مسئله ما، تابع هدف، حداقل سازی کل هزینه های منابع تجدید پذیر و تجدید ناپذیر است.

MRCPSp-TWRTPC یک کلیت از مسئله RCPSP است، و با توجه به سختی-NP از RCPSP (12، 13)، مسئله MRCPSp-TWRTPC، به خوبی سخت-NP است، و از این رو، روش فوق ابتکاری، رویکرد عملی است. در ادامه این مقاله، یک الگوریتم PSO مبتنی بر فوق ابتکار را برای حل این مسئله معرفی می کنیم. PSO، در شکل فعلی آن، به دهه ۱۹۹۰ باز می گردد؛ با این حال، در این مدت کوتاه، PSO عملکرد خوبی را در انواع مختلف از حوزه های کاربردی، به خصوص در مسائل بهینه سازی مقید نشان داده است. بسیاری از محققان، PSO را به طور گسترده ای مورد مطالعه قرار دادند و چندین تغییر را پیشنهاد دادند. در این مقاله، ما از الگوریتم PSO اصلاح شده توسعه یافته و مورد استفاده توسط Tchomt'e و Gourgand برای حل موثر RCPSP (۱۴) استفاده نمودیم.

بقیه این مقاله به شرح زیر است. در بخش بعدی، MRCPSp-TWRTPC با جزئیات توصیف شده است و در یک مدل ریاضی فرموله شده است. در بخش ۳، یک توصیف از الگوریتم PSO و تغییرات آن ارائه شده است، و در بخش ۴ یک الگوریتم بر اساس PSO اصلاح شده معرفی شده توسط Tchomt'e و Gourgand (14) توضیح داده شده است. بخش ۵ تجزیه و تحلیل تجربی است. در نهایت، بخش ۶ نتیجه گیری کار است.

۲. توصیف مسئله

در MRCPSp-TWRTPC، یک پروژه به منظور به حداقل رساندن هزینه های کل آن باید برنامه ریزی شود. منابع در دسترس برای تکمیل فعالیت های پروژه را می توان به صورت تجدید پذیر و یا غیر قابل تجدید طبقه بندی نمود. فعالیت J می تواند تعداد حالات اجرای M_j را داشته باشد. هر حالت فعالیت، طول مدت فعالیت و الزامات فعالیت برای مقدار مشخصی از منابع تجدید پذیر و تجدید ناپذیر را مشخص می کند. هر نوع منبع تجدید پذیر محدود برای یک فاصله زمانی ثابت، با شروع از زمان آماده و با پایان دادن در تاریخ مقرر آن اجاره داده می شود، و قبل از زمان آماده، در دسترس نیست اما می تواند بعد از تاریخ مقرر آن با هزینه ی جریمه ی تأخیر استفاده شود. منابع غیر قابل تجدید محدود نمی شوند. تمام فعالیت ها، در آغاز این پروژه آماده هستند، و هیچ پیشدستی مجاز نیست. اگر یک فعالیت تحت یک حالت خاص آغاز شود، حالت فعالیت قابل تغییر نیست. فعالیت J اجرا شده در حالت m دارای

مدت زمان d_{jm} است و به r_{jmk} واحد k منبع تجدید پذیر و n_{jmk} واحد k منبع تجدید ناپذیر نیاز دارد. شبکه پروژه توسط یک فعالیت روی ارائه گره (AON) با روابط تقدم پایان به شروع و زمان تاخیر صفر به تصویر کشیده می شود. فعالیت های ساختگی ۱ و n مربوط به شروع و اتمام پروژه می شوند. لیستی از فعالیت ها از نظر توپولوژیکی شماره گذاری می شود؛ یعنی، هر ماقبل هر فعالیت دارای تعداد کمتر از تعداد فعالیت است. همچنین، اولین و آخرین زمان شروع فعالیت J را توسط EST_J و LST_J تعریف می کنیم. EST_{jS} و LST_{jS} توسط CPM عبور به جلو و عقب با استفاده از حالت با کوتاه ترین مدت زمان برای هر فعالیت و اختصاص دادن $LST_n = LFT_n = T$ ، محاسبه می شوند که در آن T یک کران بالا برای طول مدت پروژه است که توسط هر روش معتبر، مانند مجموع طولانی ترین مدت زمان فعالیت کل پروژه به علاوه زمان های آماده از منابع تجدید پذیر تعیین میشود. در نتیجه، هر فعالیت J را تنها می توان در مدت زمان $[EST_j, LST_j]$ داد.

پارامترهای مسئله را به صورت زیر تعریف می کنیم:

n : تعداد فعالیت های پروژه،

NR : تعداد منابع غیر قابل تجدید،

ck : واحد هزینه K منبع تجدید ناپذیر،

R : تعداد منابع تجدید پذیر،

Rk : در دسترس بودن منبع تجدید پذیر k ،

RK : زمان آماده منبع تجدید پذیر k ،

DK : تاریخ مقرر منبع تجدید پذیر k ،

PK : هزینه جریمه دیرکرد منبع تجدید پذیر k ، برای هر دوره،

MJ : تعداد حالات فعالیت J ،

PJ : مجموعه پیشینیان فعالیت J ،

DJM : طول مدت فعالیت J تحت حالت M ،

rjmk: نیاز به منبع تجدید پذیر k, برای اجرای فعالیت j تحت حالت M,

njmk: نیاز به منبع تجدید پذیر k, برای اجرای فعالیت j تحت حالت M

ESTj: اولین زمان شروع فعالیت j

LSTj: آخرین زمان شروع فعالیت j

T: حد بالای طول مدت پروژه.

ما همچنین متغیرهای تصمیم گیری را به شرح زیر تعریف می کنیم:

$$x_{jm\tau} = \begin{cases} 1, & \text{اگر فعالیت j تحت حالت m در دوره t آغاز شود} \\ 0, & \text{در غیراینصورت} \end{cases}$$

$$y_{k\tau} = \begin{cases} 1, & \text{اگر منبع تجدید پذیر k, در دوره t استفاده شود.} \\ 0, & \text{در غیراینصورت} \end{cases}$$

تعیین می شود که در آن CPk, زمان $l_k = \max\{0, CP_k -$

.....

آزادسازی منبع k به واسطه پروژه است.

مدل برنامه نویسی عددی ترکیبی برای این مسئله را می توان به صورت زیر فرموله نمود:

$$\text{Min} \sum_{k=1}^{NR} c_k \left(\sum_{j=1}^n \sum_{m=1}^{M_j} n_{jmk} \sum_{\tau=EST_j}^{LST_j} x_{jm\tau} \right) + \sum_{k=1}^R p_k \cdot l_k \quad (2.2)$$

$$\text{S.t.} \sum_{m=1}^{M_j} \sum_{\tau=EST_j}^{LST_j} x_{jm\tau} = 1, \quad j = 1, 2, \dots, n, \quad (2.3)$$

$$\sum_{m=1}^{M_j} \sum_{\tau=EST_j}^{LST_j} (\tau + d_{im}) x_{im\tau} \leq \sum_{m=1}^{M_j} \sum_{\tau=EST_j}^{LST_j} \tau x_{jmt}, \quad j = 1, 2, \dots, n, \quad i \in P_j, \quad (2.4)$$

$$\sum_{j=1}^n \sum_{m=1}^{M_j} r_{jmk} \sum_{z=\tau-d_{jm}+1}^{\tau} x_{jmz} \leq R_k \cdot y_{k\tau}, \quad k = 1, 2, \dots, R, \quad \tau = 1, 2, \dots, T, \quad (2.5)$$

$$\sum_{\tau=1}^{r_k-1} y_{k\tau} = 0, \quad k = 1, 2, \dots, R, \quad (2.6)$$

$$\tau \cdot y_{k\tau} - d_k \leq l_k, \quad k = 1, 2, \dots, R, \quad \tau = d_k, d_k + 1, \dots, LST_n, \quad (2.7)$$

$$x_{jm\tau} \in \{0, 1\}, \quad j = 1, 2, \dots, n, \quad m = 1, 2, \dots, M_j, \quad \tau = EST_j, \dots, LST_j, \quad (2.8)$$

$$y_{k\tau} \in \{0, 1\}, \quad k = 1, 2, \dots, R, \quad \tau = 0, \dots, LST_n, \quad (2.9)$$

$$l_k \geq 0, \quad k = 1, 2, \dots, R. \quad (2.10)$$

در مدل فوق، تابع هدف (۲،۲)، به حداقل رساندن هزینه پروژه است که در آن شرایط اولیه و ثانویه، به ترتیب کل هزینه های استفاده از منابع تجدید ناپذیر و کل هزینه های جریمه تأخیر منابع تجدیدپذیر می باشند. مجموعه محدودیت (۲،۳) تضمین می کند که هر فعالیت j تحت یکی از حالات آن در دوره های زمانی مشخص شده شروع، آغاز می شود، یعنی $[EST_j, LST_j]$. مجموعه محدودیت (۲،۴)، رابطه اولویت بین فعالیت ها را اجباری می سازد. محدودیت (۲،۵)، استفاده از منابع تجدید پذیر را محدود می کند. با توجه به محدودیت (۲،۶)، منابع تجدید پذیر قبل از زمان های آماده قابل استفاده نیستند و دوره تأخیر آنها توسط محدودیت (۲،۷) تعیین می شود. در نهایت، مجموعه محدودیت های (۲،۸)، (۲،۹) و (۲،۱۰)، محدودیت های غیر کارکردی هستند.

۳. بهینه سازی ازدحام ذرات

الگوریتم PSO نسبتاً جدید، تکاملی، و یک فوق ابتکار مبتنی بر جمعیت است که در اصل توسط Kennedy و Eberhart (۱۵) توسعه یافت و دوباره توسط shi و Eberhart (۱۶) تعریف شد. با وجود مراحل اولیه روش PSO، کاربردهای گسترده ای در حوزه های بهینه سازی ترکیبی و محدود یافته است و در حال حاضر به یک موضوع تحقیق اصلی تبدیل شده است. PSO با الهام از رفتار اجتماعی ازدحامات طبیعی، از یک ازدحام ذرات برای فضای جستجو استفاده می شوند که از یک تکرار به تکرار دیگر به روزسازی می شوند. هر ذره متناظر با یک راه حل ارزیابی

شده توسط تابع هدف مسئله مورد نظر است و به عنوان یک نقطه در فضای N بعدی در نظر گرفته می شود. وضعیت یک ذره به وسیله موقعیت و سرعت (۱۵) آن نشان داده می شود. موقعیت n بعدی برای ذره i در تکرار t را می توان به صورت $X_{i,t} = (x_{i1}^t, x_{i2}^t, \dots, x_{in}^t)$ نشان داد. به طور مشابه، سرعت، نیز یک بردار n -بعدی برای ذره i در تکرار t است که می توان به صورت $V_{i,t} = (v_{i1}^t, v_{i2}^t, \dots, v_{in}^t)$ نشان داد با استفاده از ارزیابی تناسب، هر ذره بهترین موقعیتی را که تاکنون به دست آورده است به یاد می آورد که $P_{i,t}$ نامیده می شود و بهترین موقعیت همه ذرات، بهترین موقعیت بهترین ذره در کل ازدحام، که G_t نامیده می شود. بردار موقعیت هر ذره در تکرار t با استفاده از (۳،۱) به روزسازی می شود و ذرات به موقعیت جدید خود حرکت می کنند:

$$X_{i,t+1} = X_{i,t} + V_{i,t}. \quad (3,1)$$

بردار سرعت نیز توسط (3.2) به روز سازی می شود

$$V_{i,t+1} = c_1 \cdot V_{i,t} + c_2 \cdot r_2 \cdot (P_{i,t} - X_{i,t}) + c_3 \cdot r_3 \cdot (G_t - X_{i,t}). \quad (3,2)$$

این بردار یک تابع از سه مولفه است: سرعت قبلی ذره، بهترین تجربه ذره، و همچنین، بهترین تجارب ازدحام کلی تا تکرار فعلی که به ترتیب اینرسی، بخشی شناخت، و بخش اجتماعی (۱۶) نامیده می شود. فرایند به روز رسانی تا زمانی ادامه می یابد که معیار خاتمه برآورده شود که معمولاً حداکثر تعداد نسل ها، زمان پردازش، و یا بهترین موقعیت ذره در طیف گروهی است که نمی تواند پس از یک تعداد از پیش تعریف شده از نسل ها بهبود یابد.

در (۳،۲)، r_2 و r_3 اعداد حقیقی تصادفی با توزیع یکنواخت است که معمولاً از بازه $[0, 1]$ انتخاب می شوند. c_2 و c_3 ثوابت شناخته به عنوان عوامل یادگیری هستند که اهمیت بهترین تجربه های کلی و موضعی را نشان می دهند. همچنین، c_1 به عنوان یک وزن اینرسی مثبت تعریف می شود که برای اولین بار توسط Shi و Eberhart (۱۷) معرفی شد. این پارامتر خاص را می توان برای هر ذره [18] معرفی نمود. Liu et al. _19_ and Shi and Eberhart _20_ وزن اینرسی کاهش دهنده-زمان را معرفی نمود.

تجزیه و تحلیل های پارامترهای PSO, موضوع مطالعات مختلف بوده است. به عنوان مثال، Tchomt'e و Gourgand (14) برخی از شرایط را برای پارامترها تعیین نمودند تا اطمینان حاصل شود که پس از تعداد کافی تکرار، هر ذره به برخی از نقاط تعادل همگرا می شود.

اگر چه PSO در اصل برای حل مسائل پیوسته طراحی شده است، می توان آن را برای حل مسائل گسسته نیز استفاده نمود. تکنیک های مختلف برای استفاده از این الگوریتم در مسائل ترکیبی بهینه سازی طراحی شده اند مانند تکنیک های معرفی شده توسط Jarbouï و همکاران (۸)، Clerc (21)، و Kennedy و Eberhart (۲۲).

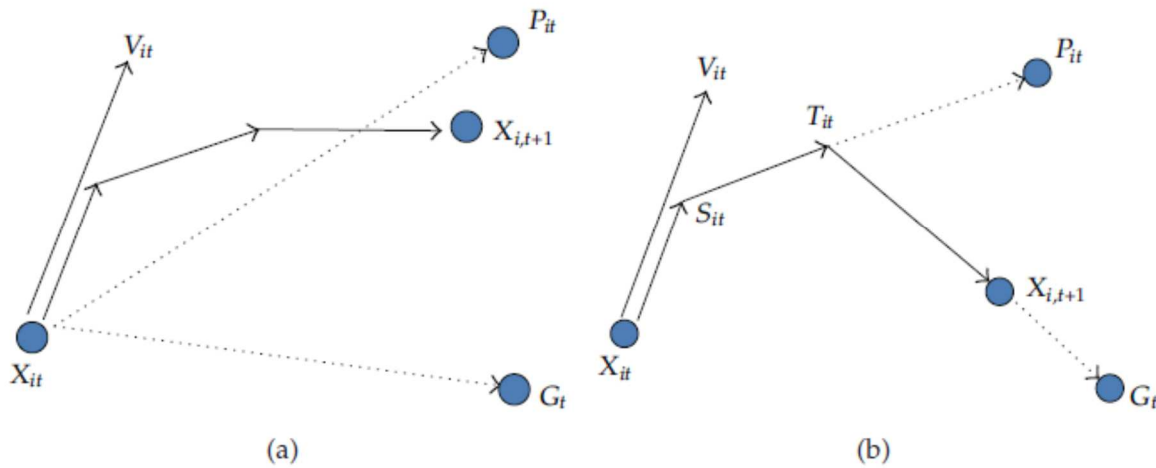
در این مقاله، ما از روش PSO اصلاح شده استفاده می کنیم که توسط Tchomt'e و Gourgand (14) به عنوان یک فرمت از PSO معرفی شد که یک قاعده جابجایی جدید ذره را ادغام می کند. نتایج محاسباتی الگوریتم آنها نشان داد که الگوریتم PSO بهتر از تمام الگوریتم های مدرن در حل RCSP عمل می کند و این دلیل انتخاب روش آنها برای مسئله ماست. ما این روش PSO اصلاح شده را در زیر توصیف می کنیم.

یک الگوریتم فوق ابتکاری باید به طور موثر و کارآمد، قادر به اکتشاف فضای جستجو باشد. الگوریتم PSO باید به اندازه کافی هوشمند باشد تا به شدت نواحی فضای جستجو را با راه حل هایی با کیفیت بالا کاوش نماید و به طور متنوع به مناطق ناشناخته از فضای جستجو حرکت کند. این دو تکنیک که توسط Glover and Laguna _23_

معرفی شدند، به عنوان روش تشدید و تنوع شناخته شده اند. Tchomt'e و Gourgand (14) مسیرهای ذرات را مورد تجزیه و تحلیل قرار داد و قوانین به روز رسانی جایگاه ذره را اصلاح نمود. ایده نشات گرفته از کاربردهای PSO که در آن ذرات اساساً از موقعیت فعلی خود به سوی بهترین موقعیت های موضعی و کلی، $(P_{i,t}, G_t)$ حرکت می

کنند، اما این ذرات به اندازه کافی به $P_{i,t}$ و G_t نزدیک نمی شوند. در نتیجه، تنوع به خوبی انجام می شود، اما تشدید به خوبی انجام نمی شود. در نتیجه، Tchomt'e و Gourgand (14) یک قانون جدید جابجایی ذره را به منظور بهبود فرآیند های تشدید پیشنهاد داد و این کار را با بازدید دو موقعیت ذره $S_{i,t}$ و $T_{i,t}$ قبل از حرکت از موقعیت فعلی، $X_{i,t}$ ، به موقعیت بعدی $X_{i,t+1}$ پیشنهاد نمود. اولاً، اینرسی در این موقعیت با حرکت دادن ذره از $X_{i,t}$ به $S_{i,t}$ تحت تاثیر قرار می گیرد. سپس، بخش شناخت، ذره را به $T_{i,t}$ حرکت می دهد و در نهایت تحت بخش

اجتماعی، ذره برای رسیدن به موقعیت جدید خود $X_{i,t+1}$ در تکرار بعدی را تحت تاثیر قرار می گیرد. با اقتباس از Gourgand و Tchomt'e (14)، جابجایی ذره در PSO کلاسیک و این PSO اصلاح شده در شکل 1(a) و 1(b) نشان داده شده است.



شکل ۱: جابجایی ذره در ازدحام: (a) PSO کلاسیک، (b) PSO اصلاح شده.

برای این منظور، قاعده به روزرسانی موقعیت ذره بدین صورت است

$$\begin{aligned}
 S_{i,t} &= X_{i,t} + c_1 \cdot V_{i,t}, \\
 T_{i,t} &= S_{i,t} + c_2 \cdot r_2 \cdot (P_{i,t} - S_{i,t}), \\
 X_{i,t+1} &= T_{i,t} + c_3 \cdot r_3 \cdot (G_t - T_{i,t}), \\
 X_{i,t+1} &= X_{i,t} + c_1 \cdot (1 - c_2 \cdot r_2) \cdot (1 - c_3 \cdot r_3) \cdot V_{i,t} \\
 &\quad + c_2 \cdot r_2 \cdot (1 - c_3 \cdot r_3) \cdot (P_{i,t} - X_{i,t}) + c_3 \cdot r_3 \cdot (G_t - X_{i,t}).
 \end{aligned} \tag{۳,۳}$$

پس از آن، در هر گام زمانی t ، بردار سرعت به شرح زیر به روز می شود:

$$V_{i,t+1} = \alpha \cdot V_{i,t} + \beta \cdot (P_{i,t} - X_{i,t}) + \gamma \cdot (G_t - X_{i,t}), \quad (3.4)$$

$$\alpha = c_1 \cdot (1 - r_2 \cdot c_2) \cdot (1 - r_3 \cdot c_3),$$

$$\beta = c_2 \cdot (1 - r_3 \cdot c_3), \quad (3.5)$$

$$\gamma = c_3.$$

Tchomt'e و Gourgard (14) نشان داد که شرایط لازم برای ضرایب به طوری که ذرات به تعادل همگرا شوند،

(۳,۶) به همراه (۳,۷) و یا (۳,۶) به همراه (۳,۸) را برآورده می سازد:

$$\phi > 0, \quad \phi - 2 * (c_1 + 1) < 0, \quad c_1 < 1, \quad (3.6)$$

$$\phi = (c_2 + c_3) / 2, \quad \text{که در آن}$$

$$0 < c_1 < 0.9, \quad 0 < c_2 < 2, \quad 0 < c_3 < 2, \quad (3.7)$$

$$0 < c_1 < 0.9, \quad 2 \leq c_2 < 4, \quad 2 \leq c_3 < 4. \quad (3.8)$$

۴. PSO اصلاح شده برای MRCPSP-TWRTPC

در این بخش، ما یک الگوریتم PSO اصلاح شده را با استفاده از رویکرد Tchomt'e و Gourgard (14)، برای

حل MRCPSP-TWRTPC. الگوریتم ۱، شبه کد را نشان می دهد. در این الگوریتم، موقعیت ذره i ام در تکرار t

بوسیله بردار n بعدی $X_{i,t} = (x_{i1}^t, x_{i2}^t, \dots, x_{in}^t)$ نشان داده می شود که در آن $x_{ij}^t, j = 1, 2, 3, \dots, n$

انتساب حالت به فعالیت i ام در تکرار t است و یک عدد صحیح در بازه $[1, M_i]$ می باشد. یک برنامه عملی برای

این پروژه از هر $X_{i,t}$ ساخته می شود. برای این منظور، ابتدا فعالیت ها اولویت بندی می شوند، بخش ۴,۳ را ببینید.

سپس، با استفاده از طرح تولید برنامه زمانبندی موازی تک-پاس، فعالیت ها زمانبندی می شوند، بخش ۴,۴ را

ببینید. رویه های جستجوی موضعی معین و بهبود در این راه حل برای رسیدن به برنامه زمانبندی بهتر به کار برده

می شوند، بخش ۴,۵, ۴,۶, ۴,۷ و ۴,۸ را ببینید.

- (1) Do Preprocessing
- (2) Generate initial particle swarm
- (3) While termination criterion is met do
- (4) While all particles have been evaluated do
- (5) Determine activities priorities
- (6) Schedule activities based on their modes and priorities using the parallel schedule generation and delay local search
- (7) While schedule is improved do
- (8) Improve schedule by Mode Assignment Modification—Part I
- (9) Improve schedule by Local Left Shift
- (10) End while
- (11) Improve schedule by Mode Assignment Modification—Part II
- (12) Compute corresponding cost of the generated schedule
- (13) End while
- (14) Update the local and global best solutions if necessary
- (15) Update position and velocity of each particle according to (3.3) and (3.4), respectively
- (16) End while
- (17) Report the global best solution

الگوریتم ۱: شبه الگوریتم PSO اصلاح شده برای MRCPSP-TWRTPC.

مقدار تناسب هر ذره توسط محاسبه هزینه کل برنامه نهایی تعیین می شود. سپس، در صورت لزوم، بهترین موقعیت های موضعی و / یا کلی به روز نمی شوند، بخش ۴،۹ را ببینید. اگر معیار خاتمه برآورده نشود، موقعیت ذرات و بردارهای سرعت به ترتیب توسط (۳،۳) و (۳،۴) برای تکرار بعدی به روز می شوند، به بخش ۴،۱۰ مراجعه کنید. قسمت های مختلف این الگوریتم با جزئیات بیشتر به شرح زیر است.

۴،۱. پیش پردازش

Sprecher و همکاران (۲۴) چند قانون پردازش را به منظور کاهش فضای عملی MRCPSP معرفی نمودند. پس از آن، این قوانین در مقالات دیگر مانند Lova و همکاران. (۶)، Peteghem و Vanhouck van (10)، و Hartman و Briskorn (25) استفاده شده است. با توجه به شباهت های بین MRCPSP-TWRTPC و MRCPSP، ما دو تا از این قوانین را برای مسئله پیشنهادی خود اعمال می کنیم. یکی، قاعده حذف حالت غیرقابل اجرا برای یک فعالیت است. برای حالت غیرقابل اجرا، مقدار منبع مورد نیاز برای اجرای فعالیت بیش از در دسترس بودن منبع است. روش دیگر استفاده از روش حذف حالت ناکارآمد است. حالت داده ناکارآمد برای یک فعالیت در

صورتی وجود دارد که حالت دیگری وجود داشته باشد که مدت زمان فعالیت برای آن کمتر باشد و این فعالیت را می توان با مقدار کلی کمتر از هر دو منابع تجدید پذیر و تجدید ناپذیر انجام داد. بنابراین، حالات فعالیت ها یک به یک تجزیه تحلیل می شوند و حالات غیرقابل اجرا و ناکارآمد حذف می شوند.

۴.۲. تولید ازدحام اولیه ذره

ازدحام اولیه ذرات به طور تصادفی تولید می شود. در اینجا، مولفه j ، $j = 1, \dots, n$ برای هر دو بردار موقعیت یا سرعت هر ذره به طور تصادفی و به ترتیب از محدوده های $[1, M_j]$ و $[-M_j, M_j]$ تولید می شود، زیرا هیچ محدودیت منبع تجدید ناپذیر در این مسئله وجود ندارد؛ علاوه بر این، تمام حالات غیرقابل اجرا حذف می شوند، تکالیف حالت اولیه امکان پذیر هستند، و هیچ تغییری مورد نیاز نیست.

۴.۳. اولویت فعالیت برای برنامه ریزی

به منظور تولید یک راه حل در MRCPSPTWRTPC، دو موضوع باید تعیین شوند: انتساب حالت فعالیت ها و برنامه ریزی فعالیت ها. با مشخص نمودن انتساب حالت برای یک راه حل، هزینه منابع غیر قابل تجدید تعیین و تثبیت می شود. پس از آن، برنامه ریزی فعالیت ها با هدف به حداقل رساندن هزینه کل جریمه های تاخیر منابع تجدید پذیر انجام می شود. بنابراین، اولویت های فعالیت ها برای برنامه ریزی توسط منابع تجدیدپذیر تعیین می شود. روش ما برای این منظور به شرح زیر است.

برای اولین بار، مجموعه ای از فعالیت ها را تعریف می کنیم که به عنوان مجموعه فعالیت K منبع (ASR_k) ، نیاز به K منبع تجدیدپذیر دارد. هر فعالیت در این مجموعه ممکن است دارای پیشینیان فوری و یا غیرفوری باشند که ممکن است یک عضو از این مجموعه نباشند. ما مجموعه ای از این پیشینیان را تعریف می کنیم که به عنوان فعالیت های پیشینیان k منبع (APR_k) اعضای ASR_k نیستند. پس از آن، جفت های ASR_k و APR_k ، $k = 1, \dots, R_n$ توسط شاخص k با استفاده از اکتشافی اولویت بندی می شوند که فعالیت ها در ASR_k و APR_k برای منبع که

دارای پتانسیل بیشتر ایجاد جریمه تأخیر است. باید اولویت بالاتری را در برنامه ریزی دریافت کند. برای دسترسی به پتانسیل هزینه جریمه دیرکرد منبع k ام، توجه داشته باشید که این هزینه جریمه با $P_k \times \max\{0, CP_k - d_k\}$ برابرست که در آن CP_k زمان انتشار منبع k در این پروژه است، و از این رو $P_k \times (CP_k - d_k)$ یک معیار خوب برای اولویت بندی منبع برای این منظور است حال سوال این است که نحوه دسترسی به CP_k بدون دانستن برنامه چیست. ما از روش زیر برای دسترسی به $CP_k, k = 1, \dots, R$ استفاده می کنیم.

از آنجا که هیچ فعالیتی نمی تواند زودتر از زمان آماده برای تمام منبع مورد نیاز آن آغاز شود، به منظور به حساب آوردن زمان آماده منبع، ما یک گره k ساختگی را برای هر منبع $k, k = 1, \dots, R$ به شبکه پروژه اضافه می کنیم. هر فعالیت ساختگی R ، حالت تک بدون نیاز به منبع، و مدت زمان r_k است. همه این فعالیت های ساختگی را می توان در زمان صفر برنامه ریزی نمود. پس از آن، برای هر فعالیت l که به k منبع تجدیدپذیر نیاز دارد، ما K فعالیت ساختگی را به عنوان یکی از پیشینیان آن تنظیم می کنیم. بنابراین اجرای فعالیت l قبل از زمان RK ممکن نیست. توجه داشته باشید که فعالیت ساختگی K ، عضوی از APR_k است. طول مسیر حیاتی زیر شبکه K با CP_k نشان داده می شود و به عنوان یک معیار از CP_k برای $k = 1, \dots, R$ در نظر گرفته می شود. CP_k با استفاده از روش CPM محاسبه می شود. پس از محاسبه CP_k برای همه منابع، بخش هایی از مجموعه های فعالیت ASR_k و $APR_k, k = 1, \dots, R$ ، با استفاده از مقدار $P_k \times (CP_k - d_k)$ اولویت بندی می شوند.

در روش خود برای اولویت بندی فعالیت ها در برنامه ریزی، ما اولویت بیشتر را به فعالیت در APR_k اختصاص می دهیم و نه فعالیت در ASR_k ، زیرا فعالیت ها در ASR_k نمی تواند برنامه ریزی شود مگر اینکه فعالیتها در APR_k زمانبندی و برنامه ریزی شوند.

در نهایت، برای هر منبع k ، اولویت بندی فعالیت های متعلق به هر مجموعه ای از ASR_k و APR_k لازم است. ما خاطرنشان می کنیم زمانی که مجموعه ای از فعالیت ها با استفاده از یک منبع تجدید پذیر برنامه ریزی شوند، ما در واقع با $RCPS$ سرو کار داریم، زیرا فعالیت ها تحت حالات مشخص به منظور اجرا در ظرف کوتاهترین زمان ممکن برنامه ریزی می شوند. بنابراین، به منظور اولویت بندی فعالیت های مجموعه ای از ASR_k یا APR_k ، ما یکی از

کارآمد ترین ابتکارات را برای برنامه ریزی فعالیت ها در RCPSA اعمال می کنیم. Lova و همکاران (۶)، تعدادی از کارآمد ترین فن آوری هوشمند را برای اولویت بندی فعالیت ها در RCPSA مقایسه کردند و دریافتند که اولویت بندی فعالیت ها به صورت مرتبه غیرکاهشی مجموع زمان شروع و پایان (LSTLFT)، بهترین نتایج را در میان ابتکارات تک پاس حاصل می کند. این مورد در میان بهترین روش های چند-پاس بوده است. روش های چند-پاس به زمان های محاسبات بسیار بیشتر از روش های تک پاس نیاز دارد، اما آنها معمولاً به بهبود ناچیز راه حل منجر می شوند؛ از این رو، ما از روش LSTLFT با برنامه ریزی تک پاس به منظور اولویت بندی فعالیت های ASRk و APRk استفاده می کنیم.

۴.۴. برنامه ریزی فعالیت ها

ما از طرح تولید برنامه موازی برای برنامه ریزی فعالیت ها استفاده می کنیم، زیرا به خوبی متناسب با جستجوی موضعی تاخیر است که ما از آن استفاده خواهیم کرد. برای جزئیات بیشتر در مورد طرح تولید برنامه ریزی موازی، بخش ۲، ۱، ۲، ۳، ۶ از Herroelen و Demeulemeester را ببینید (۱).

طرح تولید برنامه ریزی موازی، نقاط تصمیم گیری جداگانه را تکرار می کند که در آن فعالیت ها می توانند به برنامه اضافه شوند. یک نقطه تصمیم در افق زمانی متناظر با زمان های اتمام فعالیت های قبلاً برنامه ریزی شده هستند؛ بنابراین، حداکثر n نقطه تصمیم باید در نظر گرفته شوند. در هر نقطه تصمیم گیری، فعالیت های برنامه ریزی نشده که پیشینیان آن انجام شده اند در مرتبه فهرست اولویت در نظر گرفته می شوند و در صورتی که هیچ تناقض منبع در آن لحظه زمانی وجود نداشته باشد، برنامه ریزی می شوند.

در این مسئله، منابع تجدید پذیر دارای زمان آماده و در دسترس بودن هستند که قبل و بعد از این زمان متفاوت هستند و برخی از فعالیت های جدید را می توان بعد از این زمان ها برنامه ریزی نمود. بنابراین، برای انتخاب نقطه جدید در افق زمانی، ما زمان های آماده را علاوه بر زمان های تکمیل فعالیت ها در نظر می گیریم. یک نقطه جدید

در افق زمانی، نزدیک ترین نقطه به نقطه فعلی در میان زمان های آماده از منابع تجدید پذیر و زمان اتمام فعالیت های برنامه ریزی شده است.

۴,۵. جستجوی موضعی تاخیر

این روش جستجوی موضعی توسط Chen و همکاران (۲۶) برای مسئله RCPSP به منظور فرار از مینیمم موضعی مورد استفاده قرار گرفت. این روش مانند عملگر جهش در الگوریتم ژنتیک عمل می کند و می تواند برنامه ریزی هر فعالیت را به رغم اولویت آن برای مجاز نمودن برنامه ریزی فعالیت های دیگر به تاخیر بیاندازد و برخی از منابع درخواست شده توسط فعالیت انتخاب شده برای فعالیت های دیگر حفظ خواهند شد.

به منظور استفاده موثرتر از منابع، برنامه ریزی برخی از فعالیت ها با وجود اولویت آنها دچار تاخیر می شوند. بنابراین فعالیت های دیگر را می توان زودتر برنامه ریزی نمود. اگر این فعالیت های انتخاب شده با تاخیر همراه نشوند، آنها ممکن است دیگر فعالیت های پروژه را برای زمان نسبتاً طولانی به تاخیر بیاندازند. بنابراین، هر فعالیت در صورتی به تاخیر می افتد که $q \leq q_0$ ، که در آن q توزیع یکنواخت شکل تصادفی انتخاب شده $[0, 1]$ و $(0 < q_0 < 1)$ احتمال تاخیر است و "نرخ تاخیر" نامیده می شود.

با توجه به بهره وری این جستجوی موضعی تاخیر در کوتاه کردن مدت زمان پروژه که در chen و همکاران (۲۶) نشان داده شده است، ما این روش را برای فعالیت های برنامه ریزی اعمال می کنیم.

۴,۶. اصلاح انتساب حالت-قسمت اول

پس از برنامه ریزی فعالیت ها، برنامه زمانبندی فعلی می تواند دارای مجموعه ای از فعالیت ها با شناورهای آزاد مثبت باشد. ما این مورد را مجموعه FFA می نامیم. برای هر $j \in FFA$ ، ممکن است، تغییر حالت فعالیت j و برنامه ریزی مجدد آن، با استفاده از شناور آزاد آن ممکن است، به طوری که زمان پایان آن افزایش نمی یابد و از این رو هیچ فعالیت دیگری تحت تاثیر قرار نمی گیرد. تغییر حالت و زمانبندی دوباره فعالیت j می تواند هزینه های منبع

تجدید ناپذیر را کاهش دهد، اما می تواند CPK، زمان انتشار منبع $k = 1, \dots, R, K$ را تغییر دهد. تغییر CPK می تواند تاخیر منبع تجدید پذیر $k = \max\{0, CP_k - d_k\}$ ، و هزینه آن pk را تغییر دهد. اگر ما در دسترس بودن منبع $k = 1, \dots, R, K$ را برای دوره های پس از $\max\{CP_k, d_k\}$ برابر با صفر تنظیم نماییم و سپس فعالیت J را مجدداً برنامه ریزی نماییم، ما مطمئن می شویم که این برنامه ریزی جریمه های تاخیر از منبع تجدید پذیر را افزایش نمی دهد. با توجه به موارد فوق، ما "اصلاح انتساب حالت - قسمت اول" را به شرح زیر تعریف می کنیم و از آن به عنوان یک روش جستجوی موضعی در الگوریتم خود استفاده می نماییم.

(۱) برای برنامه ریزی فعلی، مجموعه FFA را با محاسبه پاس مستقیم بیابید.

(۲) در دسترس بودن منبع $k = 1, \dots, R, K$ ، را برای دوره های پس از $\max\{CP_k, d_k\}$ برابر با صفر قرار می دهیم.

(۳) برای هر $J \in FFA$ ، زمانی که ما یک پاس رو به جلو می رویم، حداقل حالت هزینه منبع تجدیدناپذیر فعالیت J را در نظر بگیرید. اگر این حالت فعلی آن نباشد، فعالیت J را در این حالت با استفاده از شناور آزاد آن و با توجه به محدودیت های منبع، بدون افزایش زمان پایان آن برنامه ریزی مجدد نمایید. اگر این تغییر زمانبندی ممکن نباشد، حداقل منبع تجدید ناپذیر بعدی حالت هزینه فعالیت J را بررسی نمایید.

۴.۷. بهبود موضعی شیفت به چپ

اصلاح انتساب حالت - قسمت یک، می تواند الزامات منبع تجدید پذیر پروژه را برای دوره های معین تحت برنامه زمانبندی حاصل کاهش دهد. این مورد انتظار می رود زیرا معمولاً این حالت با منابع کمتر غیر قابل تجدید دارای مدت زمان طولانی تر است و به منابع کمتر تجدید پذیر در هر دوره نیاز دارد. از این رو، امکان شیفت به چپ موضعی فعالیتهای خاص وجود دارد. بنابراین، ما روش شیفت به چپ استاندارد موضعی را پس از اصلاح انتساب حالت - قسمت اول انجام می دهیم.

پس از انجام شیفت به چپ موضعی، قادر به تغییر برخی از انتساب های حالت فعالیت دوباره توسط اصلاح انتساب حالت-قسمت اول خواهیم بود. از این رو، این دو روش یکی پس از دیگری استفاده می شوند تا زمانی که هیچ بهبودی را نتوان در برنامه به دست آورد.

۴.۸. اصلاح انتساب حالت-قسمت دوم

ما مجموعه ای از فعالیت های پروژه را با جانشینان در نظر می گیریم و این مجموعه را NSA می نامیم. فعالیت های ماقبل مستقیم فعالیت ساختگی n ، NSA را می سازد. برای هر $l \in NSA$ ، اگر حالت فعالیت l را تغییر دهیم و آن را برنامه ریزی مجدد نماییم، برنامه ریزی هیچ تغییر فعالیت دیگر و مقدار تغییر هزینه این پروژه $\Delta c = \Delta NRC_j + \sum_{k=1}^R p_k \Delta l_{k,r}$ است که در آن ΔNRC_j تغییر هزینه منبع تجدید ناپذیر از حالت برای فعالیت j و $\Delta l_{k,r}$ تغییر تأخیر منبع k ام است. از آنجا که فعالیت j هیچ جانشینی ندارد، $\Delta l_{k,r}$ را به راحتی می توان محاسبه نمود. اگر مقدار اثر تغییر بر هزینه کل منفی باشد، تغییر حالت برای فعالیت $j \in NSA$ قابل توجیه است. با توجه به نکات فوق، روش جستجوی موضعی زیر را به عنوان "انتساب حالت اصلاح قسمت دوم" تعریف می کنیم و از آن در الگوریتم خود استفاده می کنیم.

(۱) در نظر بگیرید که {فعالیت j ، ماقبل مستقیم فعالیت جعلی m است} $NSA = \{j \mid$

(۲) برای هر $j \in NSA$ ، برخی از حالات را برای فعالیت j در نظر بگیرید که در آن هزینه مورد نیاز منبع غیر قابل تجدید کمتر است و این صرفه جویی در هزینه بیش از افزایش احتمالی هزینه جریمه از منبع تجدید پذیر است. ΔC را برای هر یک از آنها محاسبه کنید. با توجه به محدودیت های منبع تجدید پذیر، اگر حداقل ΔC منفی باشد، حالت مربوط به آن با حالت حال حاضر فعالیت l جایگزین می شود.

۴.۹. به روز رسانی بهترین راه حل های موضعی و کلی

همانطور که قبلاً ذکر شد، الگوریتم PSO بهترین راه حل موضعی به دست آمده توسط هر راه حل بهترین کلی ذرات به دست آمده توسط کل ازدحام را را ذخیره می کند. بنابراین، پس از ارزیابی تمام ذرات ازدحام در تکرار، بهترین راه حل موضعی برای هر ذره تا تکرار فعلی با راه حل حال حاضر ذره مقایسه می شود. اگر راه حل حال حاضر دارای هزینه های کلی کمتر باشد، بهترین راه حل موضعی ذره توسط راه حل حال حاضر ذره جایگزین می شود. به طور مشابه، بهترین راه حل جهانی به روز سازی می شود.

۴.۱۰. به روز رسانی موقعیت و سرعت ذرات

برای به روز رسانی موقعیت ذرات در تولید راه حل های جدید برای تکرار بعدی، در مرحله اول، سرعت ذرات با استفاده از (۳،۴) به روز می شود. در این فرایند، هر عنصر بردار سرعت که بیش از Mi است، به Mi تغییر می یابد و هر عنصر از بردار سرعت که کمتر از $-Mi$ است به $-Mi$ تغییر می یابد. پس از آن، موقعیت های ذرات با استفاده از (۳،۳) به روز می شوند. مشابه با رویه به روز رسانی سرعت هر ذره، هر عنصر از بردار موقعیت جدید که بیش از Mi است و هر عنصر بردار موقعیت جدید که کمتر از -1 است به -1 تغییر می یابد. زمانی که عناصر بردار موقعیت، انتساب حالت از فعالیت ها را تعیین می کنند و باید عدد صحیح باشد، هر عنصر غیر عددی به نزدیک ترین عدد صحیح گرد می شود.

۵. تجزیه و تحلیل تجربی

در این بخش، ما تجزیه و تحلیل تجربی این الگوریتم را ارائه می دهیم. همه برنامه ها کدگذاری می شوند و بر روی پلت فرم # .NET 2008 C بر روی یک کامپیوتر با پردازنده Core 2 Duo CPU 2.53GHz و حافظه ۳GB اجرا می شوند.

۵.۱. مسائل نمونه

ما از کتابخانه نمونه مسائل PSPLIB (27) استفاده می کنیم و سه مجموعه از مسائل برنامه ریزی پروژه چند حالتی، J10, J16, J20 را به عنوان مسائل با اندازه کوچک و J30 به عنوان مسئله با اندازه متوسط انتخاب می کنیم. علاوه بر این، دو مجموعه از مسائل اندازه بزرگ، J90 و J60 با همان پارامترهای J30 تولید می شوند، اما با فعالیت های 60 و 90. همچنین، به منظور مشاهده اثر منابع در این مسئله، دو مجموعه فوق العاده از مسائل برنامه ریزی پروژه توسط Kolisch و همکاران [28] انجام می شوند که ما J30 R4 n4 و R4 j60 N4 را می نامیم. تمام پارامترها در این مجموعه ها، شبیه به پارامترها در مجموعه های J30 و J60 هستند، اما به جای داشتن 2 منبع تجدید پذیر و تجدید ناپذیر، 4 منبع از هر نوع وجود دارد.

توزیع یکنواخت گسسته در نوشته های مرتبط برنامه ریزی پروژه استفاده شده است؛ برای مثال، Ranjbar و همکاران (2) و Khalilzadeh و همکاران (29) توزیع یکنواخت گسسته برای تواریخ آماده و تاریخ مقرر منبع را مورد استفاده قرار دادند. از این رو، در این مقاله ما از توزیع یکنواخت گسسته برای انتخاب پارامترها استفاده کرده ایم. واحد هزینه منبع غیر قابل تجدید به طور تصادفی از توزیع یکنواخت گسسته (2,6) انتخاب شدند، هزینه واحد جریمه تأخیر از منبع تجدید پذیر به طور تصادفی از توزیع یکنواخت گسسته (10,30) انتخاب شدند. زمان آماده منبع تجدید پذیر به طور تصادفی از توزیع یکنواخت گسسته (0,15) تولید شد، و در نهایت، تواریخ مقرر منبع تجدید پذیر به طور تصادفی از توزیع یکنواخت گسسته (5,15) به علاوه مقدار زمان آماده خود برداشته شد.

5.2. تنظیم پارامترها

پنج پارامتر، نرخ تأخیر، (q_0) ، تعداد ذرات در ازدحام P، و همچنین، C1، C2، C3 و در الگوریتم PSO ما وجود دارند. برای تنظیم این پارامترها، ما از طراحی 3 نقطه فاکتوریل استفاده می کنیم همانطور که در جدول 1 نشان داده شده است. همانند بخش 3، پارامترهای C1، C2، C3 باید در محدوده های معین توسط (3,6) و (3,7)، و یا (3,6) و (3,8) انتخاب شوند. برای هر این دو رشته، سه نقطه انتخاب شده اند.

مجموعه ای از ۱۰۰ نمونه آزمون به صورت تصادفی از مجموعه J16 انتخاب شد و با محدودیت زمانی CPU 150 میلی ثانیه حل شد. پس از آن، حدود ۴ تا ۳۰ تکرار، بسته به تعداد ذرات در ازدحام اجرا شدند.

جدول ۱: تنظیمات پارامتر

پارامتر	سطوح	
Parameter	Levels	
q_0	0.2-0.4-0.6	
P	10-30-60	
c_1	0.2-0.45-0.7	
	Satisfying relations (3.6) and (3.7)	Satisfying relations (3.6) and (3.8)
c_2	0.5-1-1.5	2.5-3-3.5
c_3	0.5-1-1.5	2.5-3-3.5

جدول ۲: ارزیابی اعتبار الگوریتم.

مجموعه مسائل	زمان متوسط CPU (چند میلی ثانیه)	متوسط d	انحراف معیار
Problems set	Average CPU time (millisecond)	Average d	Standard deviation of d
J10 (536 problems)	278	1.19	2.18
J16 (550 problems)	446	3.64	4.75
J20 (554 problems)	543	4.70	5.65
J30 (640 problems)	825	7.40	8.13
J60 (640 problems)	1818	10.56	11.06
J90 (640 problems)	2916	11.76	12.21
J30-r4-n4 (640 problems)	1509	7.78	8.73
J60-r4-n4 (640 problems)	3302	11.69	12.56

هر نمونه آزمون برای همه $3^5 \times 2 = 486$ جایگشت از مقادیر پارامتر، در مجموع ۴۸۶۰۰ مسائل اجرا شدند.

مقادیر تنظیم پارامترها $q_0 = 0.4, P = 60, c_1 = 0.2, c_2 = 0.5, c_3 = 1$ هستند.

۵.۳. اعتبار الگوریتم

همانطور که پژوهش جدید ما است، ما نمی توانیم هر گونه مسئله حل شده در نوشته ها پیدا کنیم. بنابراین ما به عنوان مثال برخی از مسائل ها را توسعه می دهیم که مقادیر تابع هدف بهینه در دست هستند. سپس ما این موارد را با الگوریتم خود حل می کنیم و نتایج را آزمایش می کنیم. به منظور تولید این موارد، ما از مسائل نمونه معرفی شده در بخش ۵،۱ استفاده نمودیم؛ با این حال، ما تواریخ وظیفه مناع تجدید پذیر را به شرح زیر تغییر می دهیم. ما یک برنامه تصادفی عملی برای هر نمونه را پس از تعیین حداقل حالت هزینه تجدید ناپذیر برای هر فعالیت پروژه اصلاح می کنیم. در این برنامه، ما زمان انتشار هر منبع تجدید پذیر را تعیین نمودیم. پس از آن، ما تاریخ مقرر برای هر منبع تجدید پذیر را برابر با زمان انتشار آن تنظیم می کنیم. از این رو، این برنامه دارای هزینه جریمه دیرکرد صفر است و مقدار تابع هدف آن با هزینه منبع غیر قابل تجدید برابر است. زمانی که ما حداقل حالات هزینه تجدید ناپذیر را به فعالیت ها اختصاص می دهیم، این برنامه بهینه است و مقدار بهینه تابع هدف آن در دسترس است.

همه مسائل نمونه با روش فوق اصلاح و با الگوریتم PSO شدند. معیار خاتمه، تولید ۶۰۰ برنامه بود. به منظور ارزیابی اعتبار الگوریتم، d ، انحراف درصد مقدار تابع هدف از مقدار بهینه، برای هر مسئله آزمون حل شده محاسبه می شود که در آن $d = 100 \times (Z_p - Z_{opt}) / Z_{opt}$ ، مقدار تابع هدف بهترین راه حل به دست آمده توسط الگوریتم PSO و Z_{opt} ، مقدار بهینه تابع هدف نمونه است. جدول شماره ۲، میانگین و انحراف معیار d برای هر مجموعه مسئله را نشان می دهد. مقادیر کم میانگین و انحراف معیار d ، عملکرد خوب الگوریتم با گذشت زمان CPU کوچک را نشان می دهد، اگر چه ما هیچ استاندارد برای مقایسه نداریم.

جدول ۳: کنترل پایداری الگوریتم

Problem set	120 schedules		900 schedules	
	Average CPU time (millisecond)	Average percent deviation	Average CPU time (millisecond)	Average percent deviation
j10	69	6.18	352	5.17
j16	112	4.74	541	5.01
j20	135	4.14	648	4.14
J30	216	4.17	1039	3.65
J60	471	3.44	2220	3.35
J90	776	2.13	3652	2.08
J30-r4-n4	410	4.13	1893	3.29
J60-r4-n4	880	2.77	4085	2.76

۵.۴. پایداری الگوریتم

به منظور بررسی پایداری الگوریتم PSO، ما از این الگوریتم برای چندین نمونه استفاده کرده ایم. هر نمونه چندین بار حل شده است، و d انحراف درصد از مقدار تابع هدف برای هر نمونه محاسبه شده است.

از هر مجموعه ای از مسائل، ۱۵ نمونه به صورت تصادفی انتخاب شده استفاده می شود و هر نمونه تا ۳۰ برابر با استفاده از الگوریتم PSO حل می شود و هر بار تحت دو معیار، نسل برنامه های ۱۲۰ و ۹۰۰ انجام می شود.

برای بررسی ثبات الگوریتم، d' برای هر نمونه محاسبه می شود که در آن $d' = 100 \times Sd(Z_i) / E(Z_i)$ ، Z_i مقدار تابع هدف بهترین راه حل به دست آمده توسط حل مسئله برای اجرای i ام، $Sd(Z_i)$ انحراف استاندارد Z_i و $E(Z_i)$ میانگین Z_i است. متوسط d' برای مسائل هر مجموعه در جدول ۳ نشان داده شده است.

۵.۵. ارزیابی عملکرد روشهای بهبود

در این بخش، ما عملکرد روش های بهبود معرفی شده در بخش ۴.۶ و ۴.۸ را ارزیابی می کنیم. اولین روش، روش بهبود انتساب حالت- قسمت یک همراه با شیفت موضعی به سمت چپ است، و دیگری، روش بهبود انتساب حالت- بخش دوم است. به منظور ارزیابی هر روش بهبود، ما هر یک از الگوریتم های اصلی را برای رسیدن به دو الگوریتم ساده حذف می کنیم که هر کدام از آنها دارای دو روش بهبود هستند. ما از ۳۰ نمونه از هر مجموعه مسئله استفاده می کنیم و آنها را با استفاده از الگوریتم اصلی، همچنین، با دو الگوریتم ساده شده حل می نماییم. پس از آن،

مقایسه نتایج بدست آمده توسط الگوریتم ساده شده با نتایج حاصل از الگوریتم اصلی را انجام می دهیم. همه موارد توسط سه الگوریتم تحت سه معیار خاتمه مختلف، برای تولید ۱۲۰، ۶۰۰، و ۹۰۰ برنامه حل می شود. ما d'' را برای هر نمونه به منظور مقایسه نتایج محاسبه می کنیم که در آن $d'' = 100 \times (Z_s - Z_p) / Z_p$ ، Z_s راه حل نهایی مقدار تابع هدف PSO ساده شده است، Z_p ، راه حل نهایی مقدار تابع هدف به دست آمده توسط اصلی PSO، و d'' انحراف درصد Z_s از Z_p است.

انحراف میانگین و استاندارد d'' برای هر مجموعه محاسبه شده است. جدول ۴ اثر حذف بهبود انتساب حالت روش قسمت اول و شیفت موضعی چپ را نشان می دهد و ما می توانیم ببینیم که در تمام موارد، عملکرد الگوریتم به طور قابل ملاحظه ای رو به وخامت می گذارد و در اغلب موارد این زمان CPU میانگین، برحسب میلی ثانیه، بطور قابل توجهی افزایش می یابد اگر این جستجوی موضعی حذف شود.

جدول ۴: ارزیابی عملکرد بهبود، بخش اول

مجمعه مسائل		متوسط زمان سی پی یو در الگوریتم اصلی (میلی ثانیه)		۹۰۰ زمانبندی		متوسط زمان سی پی یو در الگوریتم ساده سازی شده (میلی ثانیه)		متوسط d		انحراف معیار d		متوسط زمان سی پی یو در الگوریتم اصلی (میلی ثانیه)		۹۰۰ زمانبندی		متوسط زمان سی پی یو در الگوریتم ساده سازی شده (میلی ثانیه)		متوسط d		انحراف معیار d		متوسط زمان سی پی یو در الگوریتم اصلی (میلی ثانیه)		۹۰۰ زمانبندی		متوسط زمان سی پی یو در الگوریتم ساده سازی شده (میلی ثانیه)		متوسط d		انحراف معیار d																																																																									
110	87	112	0.61	8.65	270	429	3.32	11.92	385	646	437	14.77	116	121	175	2.45	7.62	707	676	5.47	10.20	605	1032	2.04	8.66	120	157	220	1.30	7.41	515	842	0.38	5.85	725	1231	0.66	7.96	130	233	341	-1.52	8.06	787	1313	1.68	5.29	1120	1922	2.16	6.63	160	499	730	-0.54	3.19	1658	2858	-0.55	3.74	2419	4208	0.68	4.25	190	819	1167	0.06	3.55	2676	4575	-0.73	4.32	3960	6752	-0.93	2.84	130-4-44	437	618	-0.70	6.74	1423	2413	-0.04	3.61	2082	3611	0.39	4.27	160-4-44	887	1322	0.77	3.89	2950	5102	1.34	4.70	4294	7305	-0.62	3.45

جدول ۵ اثر حذف بهبود انتساب حالت روش بخش دوم را نشان می دهد. ما می توانیم ببینیم که در اکثر موارد، عملکرد الگوریتم به وخامت می گذارد و در تمام موارد، متوسط زمان CPU به طور قابل ملاحظه ای افزایش می یابد. در زیر، ما دلیلی برای افزایش متوسط زمان CPU را توضیح می دهیم.

۶. نتیجه گیری

در این مقاله، ما مسئله MRCPSP-TWRTPC را به عنوان یک مسئله برنامه ریزی هزینه پروژه به حداقل رساندن منبع گرا با معرفی توجه به هر دو هزینه های منبع تجدید پذیر و تجدید ناپذیر معرفی می کنیم. ما این مسئله را به عنوان مدل برنامه ریزی عدد صحیح مختلط فرموله و مدلسازی می کنیم و سختی-NP را مورد بحث قرار می دهیم. پس از آن، یک الگوریتم فوق ابتکاری برای حل مسئله برنامه ریزی پروژه را ارائه می دهیم. ما به طور خلاصه استفاده از الگوریتم PSO را برای حل مسائل بهینه سازی ترکیبی محدود نموده ایم. پس از آن، ما یک الگوریتم PSO اصلاح شده از جمله قوانین به روز رسانی اصلاح شده برای سرعت و موقعیت ذرات را استفاده می کنیم. به منظور تولید برنامه های امکان پذیر، ما از الگوریتم PSO برای انتساب حالت فعالیت استفاده می کنیم و یک تکنیک اکتشافی جدید برای اولویت بندی فعالیت در طرح برنامه ریزی موازی را توسعه می دهیم. دو روش اکتشافی بهبود، جستجوی موضعی تاخیر و شیفت سمت چپ موضعی همراه با دو روش حالت اصلاح انتساب، به منظور بهبود راه حل اجرا نمودیم. نتایج محاسباتی، پایداری الگوریتم مناسب را در حل نمونه های مختلف به طور خاص با تعداد تکرارهای بالا نشان می دهیم. همچنین، تحلیل اعتبار، انحرافات کوچک از راه حل های بهینه را برای نمونه های آزمون در زمان حل کردن معقول نشان می دهیم. نهایتاً، ما دو روش بهبود مورد استفاده را در الگوریتم خود برای نشان دادن عملکرد خوب آنها ارزیابی نمودیم.

References

- [1] E. Demeulemeester and W. S. Herroelen, *Project Scheduling, A Research Handbook*, Kluwer Academic, Dordrecht, The Netherlands, 2001.
- [2] M. Ranjbar, M. Khalilzadeh, F. Kianfar, and K. Etmnani, "An optimal procedure for minimizing total weighted resource tardiness penalty costs in the resource-constrained project scheduling problem," *Computers and Industrial Engineering*, vol. 62, no. 1, pp. 264–270, 2012.
- [3] R. Heilmann, "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, no. 2, pp. 348–365, 2003.
- [4] G. Zhu, J. F. Bard, and G. Yu, "A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem," *INFORMS Journal on Computing*, vol. 18, no. 3, pp. 377–390, 2006.
- [5] H. Zhang, C. M. Tam, and H. Li, "Multimode project scheduling based on particle swarm optimization," *Computer-Aided Civil and Infrastructure Engineering*, vol. 21, no. 2, pp. 93–103, 2006.
- [6] A. Lova, P. Tormos, and F. Barber, "Multi-mode resource constrained project scheduling: scheduling schemes, priority rules and mode selection rules," *Inteligencia Artificial*, vol. 10, no. 30, pp. 69–86, 2006.

- [7] A. Lova, P. Tormos, M. Cervantes, and F. Barber, "An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes," *International Journal of Production Economics*, vol. 117, no. 2, pp. 302–316, 2009.
- [8] B. Jarboui, N. Damak, P. Siarry, and A. Rebai, "A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems," *Applied Mathematics and Computation*, vol. 195, no. 1, pp. 299–308, 2008.
- [9] M. Ranjbar, B. De Reyck, and F. Kianfar, "A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling," *European Journal of Operational Research*, vol. 193, no. 1, pp. 35–48, 2009.
- [10] V. Van Peteghem and M. Vanhoucke, "A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 201, no. 2, pp. 409–418, 2010.
- [11] F. S. Kazemi and R. Tavakkoli-Moghaddam, "Solving a multi-objective multi-mode resource-constrained project scheduling problem with particle swarm optimization," *International Journal of Academic Research*, vol. 3, no. 1, pp. 103–110, 2011.
- [12] J. Błażewicz, J. K. Lenstra, and A. H. G. Rinnooy Kan, "Scheduling subject to resource constraints: classification and complexity," *Discrete Applied Mathematics*, vol. 5, no. 1, pp. 11–24, 1983.
- [13] P. Brucker, A. Drexel, R. Mohring, K. Neumann, and E. Pesch, "Resource-constrained project scheduling: notation, classification, models and methods," *European Journal of Operational Research*, vol. 113, pp. 3–41, 1999.
- [14] S. Kemmoé Tchomté and M. Gourgand, "Particle swarm optimization: a study of particle displacement for solving continuous and combinatorial optimization problems," *International Journal of Production Economics*, vol. 121, no. 1, pp. 57–67, 2009.
- [15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Piscataway, NJ, USA, December 1995.
- [16] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, Piscataway, NJ, USA, May 1998.
- [17] Y. Shi and R. Eberhart, "Parameter selection in particle swarm optimization," in *Annual Conference Evolutionary Programming*, San Diego, Calif, USA, 1998.
- [18] C. Y. Tsai and S. W. Yeh, "A multiple objective particle swarm optimization approach for inventory classification," *International Journal of Production Economics*, vol. 114, no. 2, pp. 656–666, 2008.
- [19] S. Liu, J. Tang, and J. Song, "Order-planning model and algorithm for manufacturing steel sheets," *International Journal of Production Economics*, vol. 100, no. 1, pp. 30–43, 2006.
- [20] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 945–1950, Piscataway, NJ, USA, 1999.
- [21] M. Clerc, "Discrete particle swarm optimization," in *New Optimization Techniques in Engineering*, G. C. Onwubolu and B. V. Babu, Eds., pp. 204–219, Springer, Berlin, Germany, 2004.
- [22] Kennedy and R. C. Eberhart, "Discrete binary version of the particle swarm algorithm," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, pp. 4104–4108, IEEE Computer Society, Washington, DC, USA, 1997.
- [23] F. Glover and M. Laguna, "Tabu search," in *Handbook of Combinatorial Optimization*, D.-Z. Du and P. M. Pardalos, Eds., vol. 3, pp. 621–757, Kluwer Academic, Boston, Mass, USA, 1998.
- [24] A. Sprecher, S. Hartmann, and A. Drexel, "An exact algorithm for project scheduling with multiple modes," *OR Spektrum*, vol. 19, no. 3, pp. 195–203, 1997.
- [25] S. Hartmann and D. Briskorn, "A survey of variants and extensions of the resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 207, no. 1, pp. 1–14, 2010.
- [26] R. M. Chen, C. L. Wu, C. M. Wang, and S. T. Lo, "Using novel particle swarm optimization scheme to solve resource-constrained scheduling problem in PSPLIB," *Expert Systems with Applications*, vol. 37, no. 3, pp. 1899–1910, 2010.
- [27] R. Kolisch and A. Sprecher, "PSPLIB—a project scheduling problem library," *European Journal of Operational Research*, vol. 96, no. 1, pp. 205–216, 1997.
- [28] R. Kolisch, A. Sprecher, and A. Drexel, "Characterization and generation of a general class of resource-constrained project scheduling problems," *Management Science*, vol. 41, pp. 693–1703, 1995.
- [29] M. Khalilzadeh, F. Kianfar, and M. Ranjbar, "A Scatter Search Algorithm for RCPSP with discounted weighted earliness-tardiness costs," *Life Science Journal*, vol. 8, no. 2, pp. 634–640, 2011.