sina-pub.ir

به حداقل رساندن هزینه وزنی دیرکرد در منبع در برنامه ریزی پروژه

چکیده

در بسیاری از پروژه ها، منابع در طول چرخه عمر پروژه برای مدت زمان محدود در دسترس هستند و دسترس نگه داشتن آنها بعد از آن, منجر به برخی از هزینه های دیرکرد می شود. به حداقل رساندن هزینه دیرکرد منبع یک هدف مورد نظر در برنامه ریزی این پروژه ها است. در این راستا، ما یک مسئله برنامه ریزی پروژه را مطالعه می نماییم که مسئله برنامه ریزی پروژه با منبع محدود، حداقل کردن مجموع هزینه جریمه دیرکرد وزنی منبع (RCPSP-TWRTPC) نامیده می شود. در این مسئله، این پروژه منوط به منابع تجدید پذیر است، هر یک از منابع تجدید پذیر برای مدت زمان محدود در طول چرخه عمر پروژه در دسترس هستند و حفظ منابع برای هر دوره اضافی به برخی از هزینه های جریمه دیرکرد منجر می شود. ما یک الگوریتم شاخه و کران را برای حل دقیق این مسئله معرفی می کنیم و از چند محدوده، تفکر، و اصول تسلط در الگوریتم خود برای کوتاه شدن فرآیند شمارش استفاده می کنیم. ما به پارامترهای موثر بر درجه دشواری RCPSP-TWRTPC اشاره می کنیم، مجموعه های گسترده ای از موارد نمونه را برای این مسئله تولید می کنیم و تجزیه و تحلیل تجربی جامعی را با استفاده از الگوریتم های سفارشی و همچنین حل کننده CPLEX انجام می دهیم. ما رفتار این الگوریتم را با توجه به تغییرات در درجه سختی نمونه ها تجزیه و تحلیل می کنیم و عملکرد آن را برای موارد مختلف با حل کننده CPLEX مقایسه می کنیم. نتایج, کارآیی الگوریتم را نشان می دهند.

کلمات کلیدی: مولفه؛ برنامه ریزی پروژه؛ دیرکرد منبع؛ به حداقل رساندن هزینه. شاخه و مرز. درخت اولویت

1. مقدمه

مسائل برنامه ریزی پروژه محدود در منبع به طور گسترده ای در نوشته های آزاد مورد مطالعه قرار گرفته اند. معیارهای مختلف در این مسائل مورد مطالعه قرار گرفته اند. به حداقل رساندن طول پروژه, یکی از پرمطالعه ترین اهداف است که

sina-pub.ir

در مسئله برنامه ریزی پروژه محدود در منبع (RCPSP) هدف گذاری شده است. به حداقل رساندن دیرکرد منبع, هدف دیگری است که بسیار کمتر در نوشته ها مطالعه شده است. این هدف در مقاله حاضر مورد مطالعه قرار گرفته است.

به حداقل رساندن دیرکرد منبع, یک هدف فراگیر در موارد عملی برنامه ریزی پروژه است. در بسیاری از پروژه ها، منابع برای دوره محدود از چرخه عمر پروژه در دسترس هستند و در دسترس نگه داشتن آنها پس از دوره های مربوط به این موضوع به مقدار هزینه دیرکرد نیاز دارد. منابع تجدیدپذیر که از خارج از پروژه اجاره می شوند و یا آنهایی که در بین پروژه های مختلف مشترک هستند معمولا تحت این شرایط قرار می گیرند. همچنین این منابع می توانند زمان های آماده داشته باشند، به عنوان مثال، آنها ممکن است در آغاز پروژه آماده نباشند. به حداقل رساندن هزینه دیرکرد منبع, یک هدف مورد نظر در برنامه ریزی این پروژه ها است.

در این مقاله, ما یک گسترش از مسئله RCPSP را مطالعه می کنیم که مسئله برنامه ریزی پروژه محدود در منبع، حداقل کردن مجموع هزینه جریمه دیرکرد وزنی منبع (RCPSP-TWRTPC) نامیده می شود [1]. در این مسئله, این پروژه منوط به منابع تجدیدپذیر است. هر منابع تجدید پذیر برای مدت زمان محدود در طول چرخه عمر پروژه در دسترس است و حفظ منابع برای دوره های اضافی به مقداری هزینه جریمه دیرکرد منجر می شود. هدف, به حداقل رساندن هزینه کلی جریمه دیرکرد منبع است. ما الگوریتم شاخه و کران برای حل دقیق مسئله را معرفی می کنیم. ساختار انشعاب سازی در این الگوریتم, شبیه به روش درخت اولویت است که یک رویکرد اساسی برای حل مسئله RCPSP است [2]. ما از چند محدوده، تفکر، و اصول تسلط برای الگوریتم خود به منظور کوتاه شدن فرآیند شمارش استفاده می کنیم. عملکرد الگوریتم از نظر عددی مورد بررسی قرار گرفت. نتایج, کارآیی الگوریتم را نشان می دهد.

نوشته ها در مسئله RCPSP به دهه 1960 باز می گردد [3]. این مسئله در مقالاتی چون [4]، [5] و [6] مورد مطالعه قرار گرفته است. با توجه به این واقعیت که این مسئله سخت-NP است [7]، بسیاری از این تلاش ها به توسعه الگوریتم های نادرست و یا محدودکننده اختصاص داده شده است. بسیاری از مقالات بررسی, نوشته های این مسئله مانند [8]، [9] و [10] خلاصه نموده است. به حداقل رساندن هزینه دیرکرد منبع در چند مقاله مطالعه مورد قرار گرفته است. مرجع [1], RCPSP-TWRTPC را معرفی نمود و یک الگوریتم حد و انشعاب را برای این مسئله ارائه نموده است. البته این مسئله که آنها مطالعه نمودند, متفاوت از مسئله مورد مطالعه ما در اینجاست، زیرا در مسئله آنها,

sina-pub.ir

تنها منابع تجدید پذیر یگانی در نظر گرفته شده اند، به عنوان مثال، حداکثر در دسترس بودن منابع در هر دوره و همچنین نیاز به منابع حداکثر در هر یک از فعالیت ها, یک هستند. در [11], یک الگوریتم ترکیبی برای این مسئله ارائه شده است. در [12], یک فرم بسط یافته از RCPSP-TWRTPC معرفی شد و مورد مطالعه قرار گرفت. در این مسئله که RCPSP-TWRTPC (MRCPSP-TWRTPCP) چند حالته نامیده می شود، فعالیت های پروژه چند حالته هستند و علاوه بر منابع تجدید پذیر, آنها تحت منابع غیر قابل تجدید هستند. هدف از این مسئله, به حداقل رساندن هزینه کل منابع، از جمله هزینه وزنی دیرکرد منابع تجدید پذیر و هزینه استفاده از منابع غیر قابل تجدید به واسطه فعالیت های پروژه است. برای حل این مسئله، یک الگوریتم فراابتکاری بر اساس یک نسخه اصلاح شده از روش بهینه سازی ازدحام ذرات معرفی شد.

بقیه این مقاله به شرح زیر است. در بخش بعدی، مسئله RCPSP-TWRTPC و در بخش سوم، الگوریتم شاخه و کران برای حل این مسئله به طور مفصل شرح داده شده است. بخش چهارم به تجزیه و تحلیل تجربی اختصاص داده شده است و در نهایت بخش V شامل چند نکته پایانی می شود.

2. توصیف مسئله

یک پروژه با n فعالیت غیر ساختگی در نظر گرفته می شود. روابط تقدم پایان-شروع بین فعالیت هایی که با استفاده از یک شبکه فعالیت-روی-گره (AON) کمتر- حلقه شبکه با گره های ساختگی 1 و n + 1 به ترتیب به عنوان گره های اولیه و ترمینال نشان داده می شوند وجود دارد. در نظر بگیرید که $K = \{1,\ldots, NR\}$, مجموعه منابع تجدید پذیر باشد. هر فعالیت j $(j=1,\ldots,n+1)$ دارای مدت زمان ثابت شده dj است و به rjk واحد از منابع تجدید پذیر k $(k \in K)$ برای هر واحد از زمان در مدت زمان آن نیاز دارد. علاوه بر این، فعالیت j دارای مجموعه از فعالیت های پیشینیان، PJ است. هر k منبع تجدید پذیر دارای در دسترس بودن ثابت RK برای هر دوره در مدت زمان پروژه، RTK زمان آماده، DTK مهلت، و هزینه ی جریمه ی دیرکرد واحد PCK است, به عنوان مثال، منبع قبل از دوره RTK در دسترس نیست، و پس از دوره DTK برای هر دوره دیرکرد منبع، هزینه مجازات PCK لحاظ می شود. هیچ پیشدستی در طول اجرای فعالیت ها مجاز نیست، همه فعالیت ها دارای حالت تک هستند، و آنها در ابتدای افق پروژه آماده

sina-pub.ir

هستند. تمام پارامترها, قطعی و انتگرال هستند. مسئله, پیدا کردن زمان شروع هر J فعالیت، SJ، $(j=1,\ldots,n+1)$. است، به طوری که تمام محدودیت های مسئله برآورده شوند و کل هزینه دیرکرد منبع تجدید پذیر به حداقل می رسد.

فرض کنید که اولین زمان شروع هر فعالیت J ، ESTJ، $(j=1,\ldots,n+1),$ ، با پاس رو به جلو تعیین می شود. به منظور دریافت EST مناسب تر برای هر فعالیت، ما به طور حداقل EST فعالیت ها را با استفاده از محدودیت های منابع تجدید پذیر محدود می کنیم. بر این اساس، اولین زمان شروع هر فعالیت نمی تواند زودتر از اولین زمانی باشد که که تمام منابع تجدید پذیر مورد نیاز توسط فعالیت آماده هستند. همچنین از آخرین زمان شروع هر فعالیت j ، LSTj، $(j=1,\ldots,n+1)$. را با استفاده از پاس رو به عقب تعیین می کنیم. به منظور انجام این کار، ما یک T حد بالا را بر روی طول این پروژه در هر راه حل بهینه برای مسئله مشخص می کنیم و آخرین زمان شروع فعالیت های ساختگی گذشته را برابر با این حد بالا قرار می دهیم. ما T را با جمع کردن مدت زمان کل فعالیت ها و اضافه کردن شدن آخرین زمان آماده منابع تجدید پذیر به این جمع تعیین می کنیم. می توان به راحتی نشان داد که T, یک کران بالا روی طول حداقل یک راه حل بهینه برای این مسئله است.

ما قبلاً متغیرهای تصمیم گیری زیر را تعریف می کنیم:

$$x_{jt} = \begin{cases} 1 & \text{if activity j starts in period t} \\ 0 & \text{otherwise} \end{cases}$$

$$j=1,\ldots,n+1, \quad t=EST_j,\ldots,LST_j.$$

$$y_{kt} = \begin{cases} 1 & \text{if renewable resource k is used in period t} \\ 0 & \text{otherwise} \end{cases}$$

$$k \in K, \quad t = 0,\ldots,T$$

LK: دیرکرد k منبع تجدیدپذیر

sina-pub.ir

$$S_j = \sum_{t=EST_j}^{LST_j} t.x_{jt}$$

سپس ما را خواهیم داشت و مدل ریاضی این مسئله به صورت زیر است

$$\text{Min} \quad \sum_{k=1}^{NR} pc_k.l_k \tag{1}$$

$$\sum_{t=EST_j}^{LST_j} x_{jt} = 1, \qquad j = 1,\ldots,n+1 \tag{2}$$

$$\sum_{t=EST_i}^{LST_i} (t + d_i).x_{it} \le \sum_{t=EST_j}^{LST_j} t.x_{jt}, \tag{3}$$
$$j = 1,\ldots,n+1, \quad i \in P_j$$

$$\sum_{j=1}^{n} \sum_{\tau=\max(t-d_j+1,EST_j)}^{\min(t,LST_j)} r_{jk} x_{j\tau} \le R_k.y_{kt}, \tag{4}$$
$$k \in K, \quad t = 0,\ldots,LST_n$$

$$\sum_{k=1}^{NR} \sum_{t=0}^{rt_k-1} y_{kt} = 0 \tag{5}$$

$$t.y_{kt} - dt_k \le l_k, \tag{6}$$
$$k \in K, \quad t = dt_k,\ldots,LST_n$$

$$x_{jt} \in \{0,1\}, \tag{7}$$
$$j = 1,\ldots,n+1, \quad t = EST_j,\ldots,LST_j$$

$$y_{kt} \in \{0,1\}, \tag{8}$$
$$k \in K, \quad t = 0,\ldots,LST_{n+1}$$

$$l_k \ge 0, \qquad k \in K. \tag{9}$$

در مدل فوق، هدف (1)، به حداقل رساندن هزینه برنامه ریزی پروژه است. محدودیت ها (2) تضمین می کنند که هر فعالیت j تنها می تواند دارای زمان شروع تنها از دوره [LSTj ،ESTJ] است. محدودیت های (3)، روابط تقدم بین هر جفت از فعالیت های (I، J) را در نظر می گیرند که در آن i، یک ماقبل از j است. محدودیت های (4)، محدودیت استفاده از منابع تجدید پذیر را با توجه به اینکه محدودیت های (5) منابع تجدید پذیر ممکن است پس از زمان آماده قابل استفاده باشند در نظر می گیرد، محدودیت های (6)، مقدار دیرکرد هر منبع تجدید پذیر را تعیین می کند و در نهایت، محدودیت های (7)، (8) و (9) نشان دهنده دامنه متغیرها است.

می توان نشان داد که مسئله RCPSP-TWRTPC یک فرمت از RCPSP است و در نتیجه، NP-سخت است.

sina-pub.ir

3. الگوریتم شاخه و کران برای RCPSP-TWRTPC

در این بخش ما الگوریتم شاخه و کران را برای حل RCPSP-TWRTPC توصیف می کنیم. این روش مبتنی بر برنامه ریزی های جزئی است که در آن تنها بخش هایی از فعالیت های پروژه برنامه ریزی می شوند. در هر گره از درخت شاخه و کران، یک فعالیت انتخاب می شود و برنامه ریزی می شود تا زمانی که تمام فعالیت های برنامه ریزی شوند. محدوده های مختلف، تفکر، و اصول تسلط در الگوریتم برای کوتاه کردن پروسه شمارش مورد استفاده قرار می گیرند. شکل 1 طرح اصلی این الگوریتم را نشان می دهد. ما هر بخش از الگوریتم را با جزئیات آن در زیر توصیف می کنیم.

1. انجام پیش پردازش و توقف, اگر نمونه غیرعملی باشد
2. تعیین جفت های منفصل از فعالیت ها
3. تعیین حد بالای اولیه برای مقدار تابع هدف بهینه
4. تولید گره اولیه و انتخاب آن برای انشعاب دادن
5. انشعاب گره انتخاب شده
6. بررسی هر گره جدید با توجه به قوانین تسلط و درک گره کنترل شده در صورت لزوم
7. بستن هر گره جدید شامل راه حل عملی و به روز رسانی کران فعلی
8. در صورتی که هنوز حداقل یک گره باز وجود دارد:

انتخاب یک گره جدید برای انشعاب با توجه به روش انتخاب

ب. انجام کنترل های تفکر در گره انتخاب شده و در ادامه از مرحله 5 اگر گره مورد اندازه نگرفته شده باشد. در غیر این صورت ادامه از مرحله 8

در غیراینصورت:

گزارش بهترین راه حل عملی به دست آمده و توقف کار

شکل 1. طرح عمومی الگوریتم شاخه و کران برای حل RCPSP-TWRTPC

A. پیش پردازش

مرجع سینا sina-pub.ir

sina-pub.ir

تنها یک مورد وجود دارد که در آن یک مثال معین از RCPSP-TWRTPC هیچ راه حل عملی ندارد. در این مورد، کمبود برای حداقل یکی از منابع تجدید پذیر وجود دارد، به عنوان مثال، یک فعالیت مانند j و یک منبع تجدید پذیر مانند k در مسئله که در آن $r_{jk}>R_{k}$ است وجود دارد. در این وضعیت, این فعالیت قابل اجرا نیست و هیچ راه حل عملی برای مسئله وجود ندارد. الگوریتم ما, امکان سنجی مثال در مرحله اول را چک می کند.

B. جفت های منفصل از فعالیت ها

دو فعالیت در گسست (انفصال) هستند, اگر منابع موجود در مسئله به اندازه کافی برای هر دو آنها قابل اجرا به صورت همزمان نباشند، بنابراین یکی از این فعالیت های منفصل باید قبل از شروع دیگری به پایان برسد. برای یک جفت از فعالیت های منفصل، آزمون زیر را می توان با برخی از روابط تقدم احتمالی بین آنها معرفی نمود:

آزمون ثبات منفصل بر اساس بازه [13]: برای دو فعالیت منفصل i و j، اگر $LST_i-EST_j<d_j$ ، i باید نسبت به j تقدم داشته باشد.

ما فعالیت های منفصل را در ابتدای الگوریتم خود برای انجام این آزمایش در طول چک کردن تفکر در هر گره از درخت شاخه بندی تعیین می کنیم. اگر برخی از رابطه های اولویت قبلاً بین یک جفت از فعالیت ها وجود داشته باشند، این آزمون هیچ اطلاعات جدیدی را برای روش حل به ارمغان نمی آورد. بنابراین ما فقط جفت های منفصل را بدون هر گونه روابط تقدم موجود بین فعالیت های مرتبط تعیین می کنیم.

C. مشخصات کران بالای اولیه

به منظور تعیین کران بالای اولیه روی مقدار تابع هدف بهینه، یک راه حل عملی را برای این مسئله با استفاده از روش اکتشافی زیر تولید می کنیم و از مقدار تابع هدف مرتبط آن به عنوان کران بالای اولیه استفاده می کنیم. در این روش ابتکاری، ابتدا، یک لیست فعالیت (AL) تولید می شود که اولویت فعالیت ها را در تولید برنامه ریزی مرتبط نشان می دهد، یعنی عنصر iام در لیست دارای اولویت i در فرآیند تولید برنامه ریزی است. به منظور تولید یک برنامه اولویت-امکان، ما از اولویت لیست فعالیت های عملی که در آن هر یک از فعالیت دارای اولویت پایین تر از پیشینیان خود را

sina-pub.ir

تولید کند. به منظور تولید این لیست، ما از مقدار مربوط به هر فعالیت استفاده می کنیم که ما آن را حداکثر هزینه دیرکرد منبع برای دوره واحد اضافی (MTC) می نامیم. MTC برای هر فعالیت, مجموع هزینه دیرکرد واحد برای آن دسته از منابع تجدید پذیر است که فعالیت نیاز دارد. به منظور تولید AL، فعالیت ها به صورت غیر افزایشی از MTC خود با توجه به روابط اولویت مرتب می شوند.

در گام دوم, پس از تولید AL، فعالیت ها با استفاده از طرح تولید برنامه ریزی سریال (SSGS) برنامه ریزی می شوند. برنامه تولید شده بدین طریق, یک راه حل عملی و معمولاً خوب برای این مسئله است.

D. تولید گره اولیه

درخت شاخه بندی الگوریتم با گره اولیه آغاز می شود. برنامه ریزی مرتبط با آن در این گره شامل اولین فعالیت ساختگی می شود که در دوره اول برنامه ریزی می شود. این گره تولید می شود و سپس با توجه به ساختار خاص الگوریتم شاخه و کران به عنوان تنها گره دردسترس منشعب می شود.

E. انشعاب

در الگوریتم ما، انشعاب سازی شبیه به الگوریتم درخت اولویت برای مسئله RCPSP انجام می شود. با توجه به هر g گره انتخاب شده برای انشعاب زنی، مجموعه ای از فعالیت های واجد شرایط (EJg) برای این گره تعیین می شود که شامل تمام فعالیت های برنامه ریزی نشده می شود که پیشینیان آنها قبلاً برنامه ریزی شده اند. برای هر عضو از این مجموعه مانند Jg، یک گره جدید تولید می شود و Jg در اولین زمان ممکن با توجه به محدودیت های اولویت و منبع برنامه ریزی می شود. این را می توان نشان داد که حداقل یک برنامه ریزی بهینه در میان برنامه های تولید شده در این راه به دست می آید.

F. اصول تسلط

sina-pub.ir

چند اصول تسلط برای الگوریتم درخت اولویت برای حل RCPSP معرفی می شوند. برخی از آنها در [14] بررسی شده است. با توجه به این قوانین تسلط، برخی از گره ها در درخت شاخه بندی را می توان قبل از انشعاب آنها برای کوتاه شدن فرآیند شمارش پی ریزی می شوند زیرا گره ها با همان کیفیت یا بهتر در درخت انشعاب وجود دارند.

دو تا از قوانین تسلط الگوریتم درخت اولویت برای RCPSP در الگوریتم ما در اینجا قابل اجرا هستند. تجزیه و تحلیل تجربی نشان می دهد که این قوانین در بهره وری الگوریتم بسیار موثر هستند. شرح این قوانین به شرح زیر است.

با توجه به هر برنامه زمانبندی جزئی در هر گره، ما می توانیم یک لیست فعالیت را مشخص می کنیم که شامل لیستی از فعالیت ها در مرتبه زمان شروع آنها می شود. هنگامی که برخی از فعالیت ها دارای زمان شروع یکسان باشند، مرتبط نمودن یک لیست فعالیت به یک برنامه جزئی معین ممکن است. در این مورد، یک گره با هر یک از این لیست فعالیت در درخت انشعاب می تواند مرتبط شود. بنابراین اگر ما دو گره با لیست های فعالیت مختلف داشته باشیم که برنامه ریزی های جزئی مرتبط آنها یکسان باشند، می توانیم یکی از آنها درک کنیم، چرا که آنها به راه حل های مشابه مرتبط می شوند. با توجه به این، اگر در برخی از گره ها، برای یک فعالیت **J** انتخاب شده و یک فعالیت برنامه ریزی شده **i**, داشته باشیم: $S_j<S_i$ ، گره تولید شده را درک نموده ایم، به دلیل اینکه شامل برنامه ریزی جزئی می شود, زیرا گره دیگر که لیست فعالیت آن یکسان است، اما مرتبه های فعالیت های **i** و **j** در لیست معکوس می باشند. مورد دیگر زمانی است که فعالیت **j** در گره با همان زمان شروع به عنوان فعالیت **i** قبلاً برنامه ریزی شده زمانبندی می شود. در این مورد, این گره شامل همان برنامه ریزی جزئی با یک دیگر می شود که در آن مرتبه **i** و **j** جایگزین می شوند. بنابراین می توانیم یکی از این گره ها را درک کنیم و به عنوان یک قاعده، در چنین مواردی ما گره را جدید درک نماییم اگر $j<i$

G. بستن گره های عملی و به روز رسانی کران بالا

sina-pub.ir

sina-pub.ir

پس از هر فرآیند انشعاب، هر گره به تازگی تولید شده مثل g که برنامه ریزی مرتبط شامل فعالیت های پروژه کلی می شود. پس از آن مقدار تابع هدف مرتبط برنامه ریزی، Fg، با کران جاری بالا و در صورتی کران بالا به صورت Fg به روزرسانی می شود که بیش از Fg باشد.

H. انتخاب گره برای انشعاب

ما از روش جستجوی-عمق-اول برای کم نگهداشتن الزامات حافظه الگوریتم استفاده می کنیم. با توجه به این روش، یک گره باز از سطح بالاتری از درخت انشعاب انتخاب می شود. اگر تساوی اتفاق افتد، سه قانون در مرتبه مورد نظر استفاده می شوند تا زمانی که یک گره منفرد انتخاب شود، از جمله انتخاب یک گره با فعالیت های برنامه ریزی شده، یک گره با حداقل کران پایین مرتبط، و گره با حداقل تعداد شاخص.

1. بررسی درک

بررسی درک در هر گره باز g انتخاب شده برای انشعاب اجرا می شود. اثربخشی این آزمون به ارزش فعلی کران بالا بستگی دارد، یعنی، مقادیر حد بالایی باریک تر به اثربخشی بیشتر در آزمون می شود. از آنجا که حد بالا ممکن است در طول الگوریتم بهبود یابد، ما چک کردن درک در هر گره درک نشده را به اندازه ممکن در تاخیر قرار می دهیم. بنابراین این بررسی زمانی انجام می شود که گره برای انشعاب انتخاب می شود.

شکل 2 مراحل اصلی این فرآیند را نشان می دهد. این فرایند با جزئیات در زیر توصیف شده است.

1. تعیین EST فعالیت های برنامه ریزی نشده
2. تعیین کران مرتبط پایین تر از گره
3. درک گره در صورتی که گره پایین تر مرتبط کمتر از حد بالاتر کنونی باشد و رفتن به مرحله 8.
4. تعیین LST فعالیت ها
5. درک گره در صورتی که EST یک فعالیت برنامه ریزی نشده بیشتر از LST آن باشد و رفتن به مرحله 8
6. آزمون سازگاری غیرمنفصل مبتنی بر بازه را انجام دهید و روابط جدید را تاثیر دهید.

مرجع سیبا sina-pub.ir

sina-pub.ir

7. ادامه از مرحله 5 اگر گره درک نشود و EST از هر گونه فعالیت برنامه ریزی نشده اصلاح شده باشد

8. پایان دادن به روند درک

شکل 2. شبه کد کنترل درک هر گره انتخاب شده برای انشعاب

1) تعیین EST فعالیت های برنامه ریزی نشده

EST هر فعالیت برنامه ریزی نشده بر اساس روابط اولویت با استفاده از پاس رو به جلو تعیین می شود. با این حال، فرآیند پاس رو به جلو در اینجا با ساخت سه تغییر اصلاح می شود. در مرحله اول، EST هر فعالیت برنامه ریزی شده برابر با زمان شروع آن تثبیت می شود، در مرحله دوم، یک مینیمم روی EST هر فعالیت برنامه ریزی نشده برابر با اولین زمانی که تمام منابع تجدید پذیر مورد نیاز توسط فعالیت آماده هستند, تعیین می شود و سوماً یک مینیمم روی زمان شروع هر فعالیت برنامه ریزی نشده با استفاده از دو قانون تسلط که در بخش 3 شرح داده شد تعیین می شود.

2) تعیین کران پایین مرتبط برای گره

کران پایین در مقادیر تابع هدف از تمام گره هایی که ممکن است از طریق انشعاب گره g انتخاب شده تولید شوند, تعیین می شود. به منظور انجام این کار، زمان شروع هر فعالیت برنامه ریزی نشده برابر با EST آن فرض می شود، پس از آن کران پایین برابر باه مقدار تابع هدف برنامه ریزی حاصل مشخص می شود.

3) تعیین LST فعالیت ها

LST هر فعالیت برنامه ریزی شده, برابر با زمان شروع آن در نظر گرفته می شود و یک کران بالا روی زمان شروع هر فعالیت برنامه ریزی نشده LSTi ،i ، با استفاده از کران بالایی جاری تعیین می شود. به منظور انجام این کار، فرض می شود که برنامه جزئی مربوط با گره g با برنامه ریزی فعالیت i در یک دوره si کامل می شود و هر فعالیت دیگر برنامه ریزی شده در اولین دوره عملی با توجه به روابط تقدم به پایان می رسد. سپس LSTi به عنوان آخرین مقدار ممکن برای برنامه زمانبندی تکمیل شده مربوط به مقدار تابع هدف از برنامه به پایان کمتر از حد بالای فعلی باقی می ماند مشخص شده است.

مرجع سینا sina-pub.ir

sina-pub.ir

بر اساس این تعریف برای LST از یک فعالیت برنامه ریزی نشده، ما سه مرحله اصلی زیر را برای تعیین آخرین زمان شروع هر فعالیت برنامه ریزی نشده انجام می دهیم. در این مراحل، توجه داشته باشید که LST یک فعالیت نمی تواند بیش از LST خود در گره والدین خود باشد. همچنین ما برخی از مجموعه های فعالیت ها را استفاده می کنیم که ما به صورت زیر آنها را تعریف می کنیم. برای هر k منبع تجدید پذیر در این مسئله، مجموعه فعالیت های پروژه نیازمند آخرین K منبع تجدید پذیر (SLAk) را به عنوان مجموعه ای از فعالیت ها تعریف می کنیم که به K منبع برای اجرا نیاز دارد، اما هیچ یک از جانشینان آنها به این منبع نیاز ندارند. بر اساس این تعریف، آخرین دوره که k منبع که در طول چرخه عمر پروژه مورد استفاده قرار می گیرد برابر با آخرین دوره اجرای فعالیت های SLAk است.

مرحله 1- برای هر فعالیت برنامه ریزی نشده i و هر k منبع تجدید پذیر، ما آخرین دوره را که فعالیت i را می توان در برنامه جزئی فعلی برنامه ریزی نمود تعیین می کنیم, به طوری که دیرکرد کنونی k منبع تغییر نکند. ما این دوره را RLSTik می نامیم و آن را برای هر k منبع و هر فعالیت برنامه ریزی نشده i در مرتبه کاهشی اعداد فعالیت ها به شرح زیر تعیین می کنیم:

• اگر i∈SLAk، RLSTik برابر با آخرین دوره ای می شود که K منبع در برنامه زمانبیند جزئی فعلی منهای گره (DI + 1) مورد نیاز است.

• در صورتی که فعالیت i و هیچ یک از جانشینان آن متعلق به SLAk، RLSTik ممکن است به بزرگی که ممکن است. در این مورد ما مجموعه RLSTik به یک حد بالا برای طول پروژه را از راه حل بهینه برابر است.

• اگر به فعالیت من SLAk تعلق ندارد، اما حداقل یکی از جانشینان خود را می کند، پس از آن RLSTik مصمم است با استفاده از عقب تصویب در RLSTjk از همه فعالیت های دیگر J< من است.

گام 2- در این مرحله، ابتدا LST از هر فعالیت برنامه ریزی نشده به EST آن برابر تنظیم شده است. سپس ما آن LST یک واحد تکراری را افزایش دهد و در هر تکرار، ما منظور افزایش مرتبط در هزینه دیرکرد از منابع است. برای هر مقدار از LST برای هر فعالیت برنامه ریزی نشده من، افزایش دیرکرد یک K منبع مساوی حداکثر صفر و LSTi-RLSTik.

sina-pub.ir

بنابراین LST می تواند به عنوان بسیار افزایش یافته است که در کل افزایش در هزینه دیرکرد به علاوه قبلاً پایین تر از گره کمتر از جریان کران از مسئله باقی مانده است.

مرحله 3- در این مرحله امکان سنجی شدن زمان شروع فعالیت هایی که در مرحله قبل مشخص شد بررسی می شود بر اساس روابط مقدم است. پاس رو به عقب است انجام و در صورت لزوم، شروع می شدن کاهش بار و اصلاح شده است.

4) انجام گسسته آزمون ثبات فاصله بر اساس

پس از تعیین EST و LST از فعالیت در گره، ما انجام بر اساس آزمون ثبات منفصل فاصله برای هر جفت از فعالیت منفصل. توجه داشته باشید که روابط ممکن است با این آزمون در هر گره گرم معرفی تنها به گرم گره ها و شاخه های آن بیشتر خاص می باشد.

پس از در گرم گره انجام آزمون، اگر برخی از روابط جدید بین فعالیت های معرفی، ما آنها را با به روز رسانی EST و LST از فعالیت های موثر ساخته شده است. بیایید فرض کنیم که یک رابطه جدید را می سازد یک فعالیت من پیشینیان فوری J فعالیت در یک گرم گره. این رابطه ی جدید در راه زیر موثر است:

- اگر فعالیت J یک فعالیت برنامه ریزی نشده در گره فعالیت من برنامه ریزی شده است یا نه، یا:
  - اگر از Esti + دی> ESTJ، ESTJ به عنوان Esti های + دی و در صورت لزوم به روز شده، EST هر جانشین فعالیت J به روز شده است با استفاده از مستقیم منتقل می کند.
  - اگر EST فعالیت افزایش در اینجا، مرتبط پایین تر از گره باید خیلی به روز و اگر آن را بیش از حد بالا قبلاً از مسئله می شود، گره به fathomed شود.
- اگر J فعالیت و نه فعالیت من برنامه ریزی شده است:
  - اگر LSTi + دی> LSTj، LSTi به عنوان LSTj-DI و در صورت لزوم به روز شده، LST از هر پیشینیان از فعالیت من به روز شده است با استفاده از پاس رو به عقب. در این مورد، اگر LST از فعالیت i یا یکی از پیشینیان خود می رسد کمتر از EST از فعالیت های مربوط به، گره fathomed است.

sina-pub.ir

4. تجزیه و تحلیل تجربی

در این بخش ما نتایج حاصل از یک آزمایش جامع محاسباتی که ما با توجه به الگوریتم ارائه شده در این مقاله برای RCPSP-TWRTPC انجام شود. الگوریتم کدگذاری شد و اجرا بر روی 2 .NET # 010،C پلت فرم. علاوه بر این، نسخه حل کننده CPLEX 12.4 در تجزیه و تحلیل استفاده شد. آزمون های تجربی بر روی یک کامپیوتر با پردازنده Core 2 Duo 2.53 گیگاهرتز CPU و RAM 3 GBS انجام شد.

در بخش های زیر، ما ابتدا از مسائل نمونه که ما استفاده شرح داده و سپس ما قبلاً و نتایج طبقه بندی شده در چهار بخش است. ما با استفاده از الگوریتم های سفارشی و همچنین حل کننده CPLEX برای حل هر مسئله نمونه. یک محدودیت زمانی از 10 ثانیه در زمان اجرا برای حل هر نمونه توسط الگوریتم و حل کننده CPLEX تحمیل شد، به طوری که می تواند آزمایش در مدت زمان معقول کامل است. به منظور بررسی اثر الگوریتم برای حل RCPSP-TWRTPC، ما دو متریک استفاده می شود. اولین متریک تعداد نمونه حل در عرض 10 ثانیه است و دومین متریک زمان اجرای واقعی برای حل هر نمونه که در عرض 10 ثانیه حل شده است.

مسائل A. نمونه

به عنوان RCPSP-TWRTPC یک فرمت از RCPSP، تمام پارامترهای که درجه سختی یک نمونه از RCPSP تحت تاثیر قرار است (به عنوان مثال، نیاز محاسباتی هر الگوریتم برای حل مثال) به احتمال زیاد بر درجه مربوطه از مسئله از یک نمونه از RCPSP-TWRTPC، TOO. علاوه بر این، پارامترهای اضافی با توجه به منابع تجدید پذیر ممکن است درجه سختی از نمونه RCPSP-TWRTPC تاثیر می گذارد. ما دو پارامتر در این زمینه است که ما در زیر توصیف مشخص شده است. اثرات واقعی از این پارامترها در درجه سختی از موارد می تجربی در نتایج محاسباتی در بخش های بعدی مشاهده شده است.

پارامتر اول است با توجه به چگونه همانند پارامترهای منابع مختلف تجدید پذیر، به طور خاص، کمتر مختلف آماده بار، ضرب العجل ها، و هزینه های واحد پنالتی از منابع مختلف تجدید پذیر هستند، بیشتر از درجه سختی از نمونه مرتبط است. تقریبا استنتاج، این است که عمدتا به دلیل که تفاوت بین مقدار تابع هدف از راه حل های مختلف کمتر قابل توجه برای منابع بیشتر شبیه هستند، پس از آن سخت تر خواهد شد به تشخیص بهینگی گزینه های مختلف در

sina-pub.ir

مراحل میانی از فرایند حل مسئله . استدلال مشابه قابل توجهی برای پارامتر دیگر که توجه به ارزش زودرسی مهلت منابع است. بر اساس پارامتر، زودتر از مهلت منابع، محاسبات بیشتری مورد نیاز است برای حل موارد.

بر اساس پارامترهای فوق موثر بر درجه سختی از موارد، ما چهار گروه از نمونه ساخته شده است. به منظور آن، ما تولید مجموعه ای از بیست و موارد به عنوان مجموعه ای پایه و در هر گروه شامل آن است. علاوه بر این، در هر گروه ما شامل چندین مجموعه دیگر که موارد همان مجموعه ای از پایگاه بود، اما در فقط یک پارامتر موثر بر تفاوت درجه سختی. این موارد دیگر در بخش های زیر می باشد.

به منظور داشتن یک طراحی کامل فاکتوریل از پارامترهای که به RCPSP مربوط، ما مسائل نمونه از پایه مجموعه ای از کتابخانه برنامه ریزی پروژه شناخته شده (PSPLIB) انتخاب کردم، اما به جای استفاده از موارد RCPSP را از این کتابخانه، ما با استفاده از چند حالت RCPSP (MRCPSP) موارد، چرا که آنها به طور کلی به تعداد کمتر از فعالیت ها و منابع تجدیدپذیر قرار گرفتند. ما به طور تصادفی بیست موارد از j30 مجموعه ای از موارد MRCPSP از کتابخانه انتخاب شده و هر نمونه به یک نمونه از RCPSP-TWRTPC با استفاده از روش زیر تبدیل:

• در موارد j30 مجموعه، هر یک از فعالیت غیر ساختگی دارای سه حالت اعدام. برای تبدیل هر نمونه انتخاب شده، یک حالت به طور تصادفی برای هر فعالیت، انتخاب شد. مدت زمان و نیاز از منابع تجدید پذیر از فعالیت خود را به مقدار مربوط به حالت انتخاب شده برابر تنظیم شد.

• ما به طور تصادفی تولید زمان آماده از منابع تجدید پذیر با استفاده از توزیع نرمال. ما در میانگین و انحراف استاندارد توزیع را به یک سوم و یک دوازدهم از طول مسیر بحرانی پروژه، ترتیب برابر تنظیم شده است.

• ما به طور تصادفی تولید مهلت مقرر از منابع تجدید پذیر استفاده از نرمال. ما در میانگین و انحراف استاندارد توزیع به سه چهارم و یک چهارم طول مسیر بحرانی پروژه، ترتیب برابر تنظیم شده است.

• ما به طور تصادفی تولید هزینه واحد پنالتی از منابع تجدید پذیر با استفاده از توزیع نرمال. ما مجموعه ای هر دو از میانگین و واریانس توزیع به سه برابر است.

B. گروه 1 - نمونه با تعداد مختلف از فعالیت

sina-pub.ir

ما شامل موارد با تعداد مختلف از فعالیت های در گروه اول. ما تولید شده و شامل سه مجموعه در همان راه از نسل مجموعه ای از پایگاه، اما به جای استفاده از مجموعه ای j30 از موارد MRCPSP از PSPLIB، ما موارد را با همان پارامترها اما با 10، 20، و 60 فعالیت استفاده می شود. مربوط به موارد با 10 و 20 فعالیت، ما J10 و مجموعه J20 از موارد MRCPSP از PSPLIB استفاده می شود. از آنجا که هیچ عنوان مثال نمونه MRCPSP با بیش از 30 فعالیت در PSPLIB در دسترس است، ما تا به حال برای تولید موارد MRCPSP مرتبط با 60 فعالیت های. ما با استفاده از نرم افزار ژنراتور پروژه Progen به تولید این موارد.

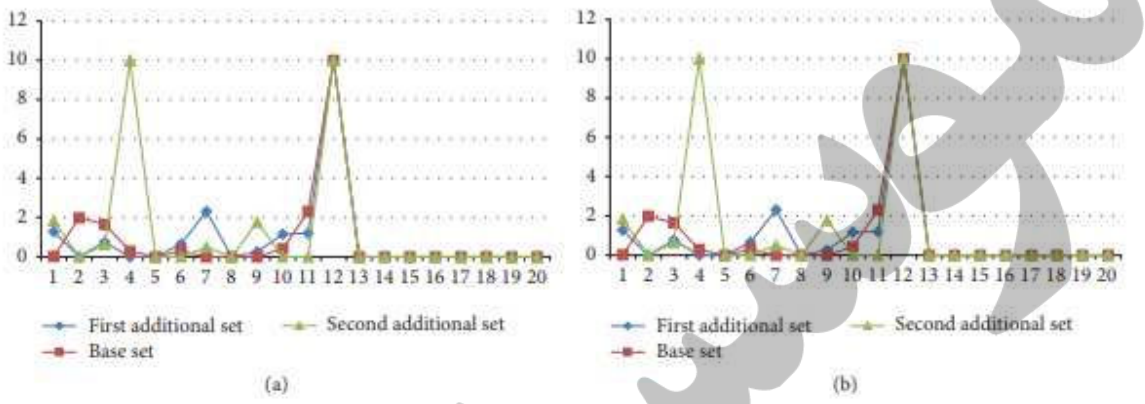
جدول 1 تعداد نمونه به بهینگی در عرض 10 ثانیه از هر مجموعه ای در گروه های من حل را نشان می دهد، و به شکل 3 نشان می دهد زمان حل هر نمونه توسط هر یک از حل کننده. در این شکل، زمان حل از موارد حل نشده در محدودیت زمانی نشان داده شده است 10 ثانیه. همانطور که تعدادی از فعالیت های در موارد را افزایش می دهد، با توجه به جدول، ما می توانیم مشاهده کنیم که تعدادی از نمونه های حل در عرض 10 ثانیه کاهش می یابد و همچنین، با توجه به شکل، زمان حل از موارد به طور کلی افزایش می دهد. این روند حاکی از یک درجه بالاتر از مسئله برای حل مسئله به عنوان تعدادی از فعالیت های افزایش می دهد، که البته سازگار با انتظار است. مقایسه نتایج حاصل از الگوریتم شاخه و کران با نتایج حاصل از حل کننده CPLEX در جدول نشان می دهد که آنها انجام تقریبا همین کار را برای نمونه با 10 و 20 فعالیت، اما الگوریتم شاخه و کران کار می کند قابل ملاحظه ای بهتر از حل کننده CPLEX برای حل موارد با تعداد بیشتری از فعالیت های. همچنین با توجه به شکل، ما می توانید ببینید که الگوریتم شاخه و کران انجام بسیار سریعتر از حل کننده CPLEX برای اکثر موارد.

# activities in instance set	# instances solved by the CPLEX solver	# instances solved by the branch and bound algorithm
10	20	20
20	19	20
30	4	19
60	0	19

جدول 1. جدول 1 تعدادی از نمونه های هر یک از مجموعه ای از گروه من از 20 در 10 ثانیه توسط شعبه و الگوریتم محدود و حل کننده CPLEX به بهینگی حل

sina-pub.ir

sina-pub.ir



شکل 3. زمان حل (ثانیه) از موارد از گروه های (الف) CPLEX حل کننده، (ب) الگوریتم شاخه و کران

C. گروه 2 - نمونه با تعداد مختلف از منابع تجدید پذیر

ما شامل موارد با تعداد مختلف از منابع تجدید پذیر در گروه دوم. ما تولید شده و شامل دو مجموعه از مجموعه پایه در این گروه. برای نمونه از مجموعه اول، یکی از منابع تجدید پذیر در نظر گرفته شد. ما در موارد از این مجموعه در همان راه به عنوان مجموعه ای از پایگاه تولید، اما در هر مورد، ما به طور تصادفی حذف یکی از دو محدودیت از منابع تجدید پذیر. در موارد از ثانیه تنظیم دیگر از مجموعه ای از پایگاه شامل، چهار منابع تجدیدپذیر در نظر گرفته شد. باز هم ما به مواردی از این مجموعه در همان راه به عنوان مجموعه ای از پایگاه تولید، با این حال ما تعداد از محدودیت منابع انرژی تجدیدپذیر افزایش یافته است. به منظور انجام این کار، در طول تحول هر نمونه انتخاب MRCPSP به یک نمونه RCPSP-TWRTPC از این مجموعه با توجه به روند که در بخش IV.A توصیف شده است، به جای استفاده از داده های مربوط به نیاز به منابع از فعالیت تحت انتخاب حالت، ما اطلاعات مربوط به حالت انتخاب نشده استفاده می شود. اطلاعات مربوط به هر یک از حالت انتخاب نشده به عنوان منابع مورد نیاز از فعالیت برای دو نفر از چهار منابع استفاده شد. همچنین، در کل قابل از منابع سوم و چهارم به ترتیب به کل قابل از منابع اول و دوم برابر، تعیین شد.

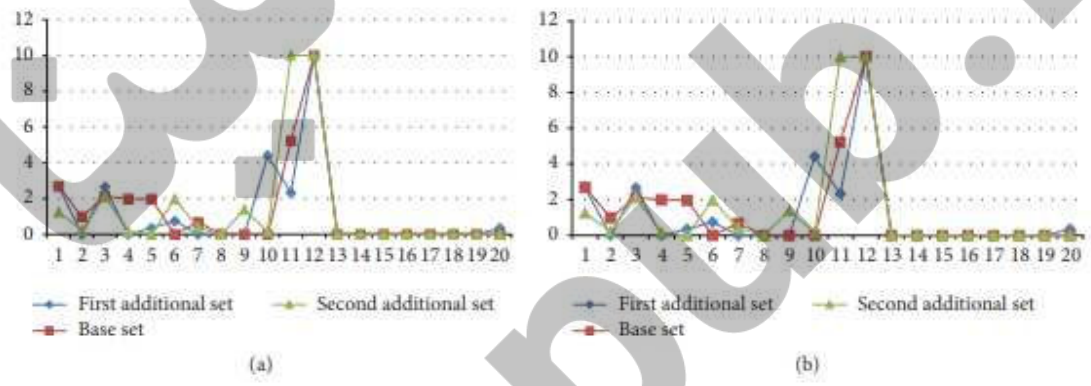
جدول 2 تعداد نمونه هر یک از مجموعه ای از گروه دوم، با شاخه و مرز و مرز و الگوریتم حل کننده CPLEX به بهینگی حل را نشان می دهد، و شکل 4 نشان می دهد زمان حل هر نمونه توسط هر یک از حل کننده. در این شکل، زمان حل از موارد حل نشده در محدودیت زمانی شده است نشان داده شده 10 ثانیه. همانطور که تعدادی از منابع تجدید پذیر در موارد را افزایش می دهد، ما می توانیم مشاهده کنیم که با توجه به جدول، تعدادی از مسائل توسط هر

sina-pub.ir

دو حل کننده به بهینگی حل کاهش می یابد و مطابق شکل، زمان حل هر نمونه به طور کلی افزایش می دهد. این روند به وضوح تاثیر مستقیم تعداد منابع تجدید پذیر در نیازهای محاسباتی برای حل مسئله را نشان دهد؟ بر اساس نتایج حاصل از جدول، الگوریتم شاخه و کران کار می کند قابل ملاحظه ای بهتر از حل کننده CPLEX برای حل موارد از کل مجموعه و افزایش اختلاف به عنوان تعدادی از منابع تجدید پذیر را افزایش می دهد. همچنین، بر اساس شکل، ما می توانید ببینید که الگوریتم شاخه و کران کار می کند بسیار سریع تر از حل کننده CPLEX برای حل اکثر موارد.

# renewable resources in instance set	# instances solved by the CPLEX solver	# instances solved by the branch and bound algorithm
1	13	20
2	5	19
4	0	17

جدول 2. جدول 2 تعداد موارد هر مجموعه ای از گروه دوم از 20 در 10 ثانیه توسط شعبه و الگوریتم محدود و حل کننده CPLEX به بهینگی حل



شکل 4. زمان حل (ثانیه) از موارد از گروه های (الف) CPLEX حل کننده، (ب) الگوریتم شاخه و کران

D. گروه 3 – نمونه با قیاس های مختلف از منابع تجدید پذیر

ما شامل موارد با قیاس های مختلف از منابع تجدید پذیر در گروه سوم. به منظور تولید چنین مواردی مختلف، ما واریانس از سه تابع توزیع مورد استفاده در نسل از آماده هزینه مجازات هم، مهلت، و واحد از منابع تجدیدپذیر را تغییر داد. بیشتر که ارزش واریانس مربوط، منابع کمتر مشابه انتظار می رود که. ما تولید شده و شامل دو مجموعه در همان

sina-pub.ir

راه به عنوان پایه در مجموعه ای از گروه سوم، اما در اول و دوم مجموعه بیشتر، ما در ارزش پایه از سه واریانس که در پایه تعیین شده توسط 0.5 و 2 در نظر گرفته شد ضرب .

جدول 3 تعداد موارد هر مجموعه ای از گروه سوم، با شاخه و مرز و مرز و الگوریتم حل کننده CPLEX به بهینگی حل را نشان می دهد، و شکل 5 نشان می دهد که زمان حل هر نمونه توسط هر یک از حل کننده. در این شکل، زمان حل از موارد حل نشده در محدودیت زمانی نشان داده شده است 10 ثانیه. به عنوان نمونه با منابع کمتر مشابه هدف قرار، ما می توانیم مشاهده کنیم که بر اساس جدول، تعدادی از مسائل توسط CPLEX حل را افزایش می دهد به بهینگی حل و بر اساس شکل، زمان حل نمونه توسط هر دو حل کننده به طور کلی کاهش می یابد. این روند نشان می دهد تاثیر درجه قیاس از منابع تجدید پذیر در مورد نیاز محاسباتی برای حل مسئله. با این حال، جدول نشان می دهد که میزان تغییرات در درجه قیاس از منابع در موارد مختلف در نظر گرفته در اینجا هیچ تاثیری قابل توجه بر عملکرد الگوریتم شاخه و کران شده است.

با مقایسه نتایج حاصل از الگوریتم شاخه و کران با نتایج حاصل از حل کننده CPLEX در جدول 3 نشان میدهد که شاخه و مرز الگوریتم کار قابل ملاحظه ای بهتر از حل کننده CPLEX برای حل موارد از کل مجموعه و افزایش اختلاف به عنوان قیاس از تجدید پذیر منابع کاهش می یابد. همچنین، بر اساس شکل 5، ما می توانید ببینید که الگوریتم شاخه و کران کار می کند بسیار سریع تر از حل کننده CPLEX برای حل اکثر موارد.

E. گروه 4 - نمونه با ارزش های مختلف زودرسی از منابع تجدید پذیر مهلت

ما شامل موارد با ارزش های مختلف زودرسی مهلت منابع تجدید پذیر در گروه چهارم. به منظور تولید چنین مواردی مختلف، ما متوسط از توابع توزیع مهلت منابع تجدید پذیر را تغییر داد. ما تولید و شامل دو مجموعه در همان راه به عنوان پایه در مجموعه ای از گروه چهارم است، اما در اول و دوم مجموعه های اضافی، ما مجموعه ای از مقدار متوسط از منابع به نصف و یک زمان از پروژه طول مسیر بحرانی برابر ضرب العجل ها، به ترتیب.

جدول 4 تعداد موارد هر مجموعه ای از گروه چهارم توسط شعبه و الگوریتم محدود و حل کننده CPLEX به بهینگی حل را نشان می دهد، و شکل 6 نشان می دهد زمان حل هر نمونه های هر راهحل. در این شکل، زمان حل از موارد حل نشده در محدودیت زمانی نشان داده شده است 10 ثانیه. به عنوان نمونه با ارزش زودتر از مهلت منابع تجدید پذیر

sina-pub.ir

را هدف قرار، ما می توانیم مشاهده کنیم که بر اساس جدول، تعدادی از مسائل توسط CPLEX حل را افزایش می دهد به بهینگی حل و بر اساس شکل، زمان حل نمونه توسط هر دو حل کننده به طور کلی کاهش می یابد . این روند نشان می دهد تاثیر این پارامتر در مورد نیاز محاسباتی برای حل مسئله. با این حال، جدول نشان می دهد که میزان تغییرات در این پارامتر تاثیری قابل توجه بر عملکرد الگوریتم شاخه و کران شده است.

مقایسه نتایج حاصل از الگوریتم شاخه و کران با نتایج حاصل از حل کننده CPLEX در جدول 4 نشان می دهد که شاخه و مرز الگوریتم کار قابل ملاحظه ای بهتر از حل کننده CPLEX برای حل موارد از کل مجموعه و افزایش اختلاف به عنوان ارزش زودرسی از منابع تجدید پذیر مهلت کاهش می یابد. همچنین، بر اساس شکل 6، الگوریتم شاخه و کران کار می کند بسیار سریع تر از حل کننده CPLEX برای حل اکثر موارد.

جدول 4. جدول 4 تعدادی از نمونه های هر یک از مجموعه ای از گروه چهارم از 20 تا بهینگی در 10 ثانیه توسط شعبه و الگوریتم محدود و حل کننده CPLEX حل شده است.

شکل 6. زمان حل (ثانیه) از موارد از گروه چهارم توسط (الف) CPLEX حل کننده، (ب) الگوریتم شاخه و کران

5. نتیجه گیری ها

در این مقاله ما به معرفی و مورد مطالعه قرار مسئله RCPSP-TWRTPC. ما سفارشی شاخه اساسی و روش حد از RCPSP، الگوریتم درخت اولویت، برای حل مسئله. ما همچنین چند محدوده و قوانین تفکر در روش سفارشی معرفی، از جمله آنهایی استخراج شده از روش اصلی RCPSP و استفاده بدون هر گونه تغییر و یا بعد از سفارشی سازی، و آنهایی که به طور خاص برای RCPSP-TWRTPC طراحی شده است. ما یک آزمایش محاسباتی جامع با استفاده از الگوریتم های سفارشی انجام شده و نتایج گزارش شده است. ما همچنین از حل کننده CPLEX در تجزیه و تحلیل و

sina-pub.ir

مقایسه استفاده می شود. ما تولید شده و نمونه های مختلف با تعداد مختلف از فعالیت ها، تعداد از منابع تجدید پذیر، درجه قیاس از منابع تجدید پذیر، و ارزش زودرسی از مهلت منابع تجدید پذیر استفاده می شود. تجزیه و تحلیل نشان داد که این سه پارامتر را تحت تاثیر قرار نیاز محاسباتی برای حل مسئله و همچنین، عملکرد نسبی الگوریتم شاخه و کران و حل کننده CPLEX با توجه به هر یک از دیگر. الگوریتم شاخه و کران عملکرد خیلی بهتر از حل کننده CPLEX توسط حل نمونه های بیشتری از هر مجموعه در مدت زمان محدود و انجام بسیار سریعتر برای حل بسیاری از موارد، که نشان می دهد راندمان بالا در حل مسئله RCPSP-TWRTPC را نشان داد.

مرکز سینا sina-pub.ir