

حل مساله زمان بندی تک ماشینه با در نظر گرفتن جریمه های زودکرد و دیرکرد مرتبه دوم

کامران کیانفر و قاسم مصلحی*

کلمات کلیدی

زمان بندی عملیات،
مساله تک ماشین،
جریمه زودکرد و دیرکرد مرتبه دوم،
روش شاخه و کران

چکیده:

امروزه مسائل زمان بندی در بسیاری از سیستم های تولیدی و خدماتی کاربرد وسیعی یافته اند. در این مقاله مساله ی زمان بندی n کار بر روی تک ماشین و با در نظر گرفتن فرض عدم بیکاری ماشین و همچنین مجاز نبودن انقطاع کارها و تساوی زمان های پردازش مورد بررسی قرار می گیرد. در بسیاری از مطالعات انجام شده تاکنون به دلیل وجود هزینه های ناشی از دیر تحویل دادن و یا هزینه های نگهداری، جریمه های زودکرد و دیرکرد کارها به صورت ترکیبات خطی در تابع هدف ظاهر می شود در صورتی که تابع هدف این مقاله کمینه کردن مجموع وزن دار جریمه های دیرکرد و زودکرد درجه دوم می باشد. در این پژوهش برای حل مساله ی فوق از روش شاخه و کران استفاده شده است و برای پیاده سازی الگوریتم شاخه و کران تعدادی اصل غلبه، حدود پایین و حدود بالا پیشنهاد شده است. در نهایت روش پیشنهادی با استفاده از تولید مسائل نمونه و پیاده سازی روش بر روی آنها مورد بررسی قرار گرفته و کارایی آن در حل مسائل به اثبات رسیده است.

۱. مقدمه

مسائل تک ماشینه از جمله مسائل متداول در حیطه مسائل زمان بندی به شمار رفته که در بسیاری از سیستم های تولیدی عملی کاربرد دارند. به علاوه در بسیاری از سیستم های تولیدی پیچیده تر نیز شرایط تولیدی معمولاً به یک ماشین که گلوگاه نامیده می شود وابسته است و ارائه یک برنامه زمان بندی خوب برای آن تک ماشین موجب بهبود کارایی کل سیستم و تولید بیشتر خواهد شد.

مدل های زمان بندی با در نظر گرفتن جریمه های زودکرد و دیرکرد به صورت همزمان را می توان با فلسفه تولید بهنگام (JIT) مقایسه

نمود. تولید بهنگام بر این نکته تاکید می کند که محصولات دقیقاً در زمانی که به آنها نیاز داریم تولید شوند و هرگونه زودکرد یا دیرکردی در تولید باعث ایجاد هزینه در سیستم تولید بهنگام خواهد شد. مساله ی مورد بررسی در این تحقیق نیز بر مبنای فلسفه فوق طراحی شده است و هزینه های زودکرد و دیرکرد تولید را به صورت همزمان در مدل لحاظ می کند اما با این تفاوت که در اینجا به جای در نظر گرفتن جریمه های دیرکرد و زودکرد به صورت خطی که در ادبیات موضوع متداول است، این دو هزینه به شکل درجه دوم در مساله وارد شده اند. بنابراین محصولاتی که دیرکرد یا زودکرد داشته باشند با شدت بیشتری جریمه خواهند شد. از طرف دیگر، این نوع مدل سازی مساله تک ماشینه مانع از آن می شود تا تعداد کمی از کارها بخش بزرگی از جریمه را به خود اختصاص دهند و بنابراین موجب توزیع مناسب تر جریمه های مربوط به عدم تحویل به موقع بین تمامی کارها می شود. همچنین در بسیاری از سیستم های تولیدی عملی، ساختار هزینه به صورت غیرخطی تغییر می کند و بنابراین در نظر گرفتن هزینه ها به شکل غیرخطی در مساله، فرضی دور از واقعیت نیست.

تاریخ وصول: ۹۰/۲/۱۲

تاریخ تصویب: ۹۰/۳/۲۵

کامران کیانفر، دانشکده مهندسی صنایع و سیستم ها، دانشگاه صنعتی اصفهان، k.kianfar@in.iut.ac.ir

*نویسنده مسئول مقاله: دکتر قاسم مصلحی، دانشکده مهندسی صنایع و سیستم ها، دانشگاه صنعتی اصفهان، moslehi@cc.iut.ac.ir

فرض دیگری که در مساله ای مورد بررسی لحاظ شده است، فرض عدم بیکاری ماشین است. توجه به این فرض در بسیاری از سیستم های تولیدی ضروری است، مثلاً در زمانی که ظرفیت ماشین در مقایسه با میزان تقاضا محدود باشد، لازم است تا برای برآوردن سفارشات مشتری ها ماشین به صورت پیوسته کار کند. همچنین در زمانی که هزینه های توقف و راه اندازی مجدد ماشین آلات نسبت به جریمه های زودکرد و دیرکرد تولید بزرگتر باشد

از دیگر موارد به کار بردن فرض عدم بیکاری برای ماشین ها به شمار می رود. با توجه به فرضیات بیان شده می توان مساله را با نماد $1|p_j = p|\sum h_j E_j^2 + w_j T_j^2$ نمایش داد. از آنجا که در ادبیات موضوع مساله کمینه کردن جریمه زودکرد و دیرکرد خطی به تنهایی، به عنوان یک مساله NP-Hard شناخته شده است، بنابراین مساله حاضر که جریمه های زودکرد و دیرکرد را به صورت غیرخطی و وزن دار در نظر می گیرد نیز مساله ای NP-Hard می باشد.

۲. مرور ادبیات موضوع

مساله مشابه با فرض عدم بیکاری ماشین و جریمه های زودکرد و دیرکرد وزن دار و خطی تاکنون توسط محققین مختلفی مطالعه شده و روش های دقیق و ابتکاری متنوعی نیز برای آن ارائه شده است. از میان روش های دقیق می توان به الگوریتم های شاخه و کرانی که توسط عبدالرزاق و پاتس [۱]، لی [۲] و لیائو [۳] و والنته و آلوز [۴] توسعه داده شده اشاره کرد.

همچنین روش های ابتکاری مختلفی نیز تاکنون برای مساله مذکور ارائه شده که از جمله می توان قوانین توزیع و الگوریتم های جستجوی پرتویی^۱ معرفی شده توسط او و مورتن [۵] و والنته و مارتز [۶-۷] را نام برد.

مساله تک ماشینه با جریمه های زودکرد و دیرکرد خطی و بدون وزن با فرض مجاز بودن بیکاری برای ماشین توسط کیم و یانو [۸] و ونتورا و رادهاکریشن [۹] مطرح شده است که در تحقیق اول، یک الگوریتم شاخه و کران و در تحقیق دوم یک الگوریتم آزادسازی لاگرانژی برای مساله ای مذکور پیشنهاد شده است. همچنین مساله فوق در حالت وجود موعدهای تحویل یکسان برای تمام کارها توسط ساندراهاهان و احمد [۱۰] تحلیل شده است.

مسائلی که توابع هدف غیرخطی دارند نیز در مطالعات گوناگون بررسی شده است که از جمله می توان به تحقیق اسکالر [۱۱] اشاره کرد که در آن مقاله مساله ای تک ماشینه با فرض مجاز بودن بیکاری ماشین و جریمه های زودکرد خطی و دیرکرد درجه دوم

لحاظ شده است. وی یک فرایند جدول بندی زمانی^۲ برای قرار دادن زمان های بیکاری در میان توالی را به همراه چند روش شاخه و کران و ابتکاری پیشنهاد کرد. گوپتا و سن [۱۲] برای حالت جریمه زودکرد درجه دوم و عدم بیکاری ماشین، یک الگوریتم شاخه و کران و یک روش ابتکاری را توسعه دادند؛ درحالی که سو و چانگ [۱۳] و اسکالر [۱۴] همین مساله را در حالتی که بیکاری ماشین مجاز باشد در نظر گرفته و برای آن یک فرایند جدول بندی زمانی و چند روش ابتکاری دیگر پیشنهاد کردند. سن و سایرین [۱۵] برای مساله ای تک ماشینه با تابع هدف هزینه دیرکرد درجه دوم وزن دار در حالتی که بیکاری ماشین تنها قبل از شروع پردازش اولین کار مجاز باشد، یک الگوریتم شاخه و کران معرفی کردند. از میان مقالات مروری موجود در زمینه مسائل زمان بندی با جریمه های دیرکرد و زودکرد نیز می توان به تحقیقات بیکر و سودر [۱۶]، کانت و سریده هاران [۱۷] اشاره نمود.

در این تحقیق از میان روش های حل دقیق مورد استفاده در مسائل زمان بندی، روش شاخه و کران انتخاب شده است. در روش شاخه و کران سه جزء اصلی وجود دارد که می بایست برای پیاده سازی الگوریتم تعریف شده باشند؛ این سه جزء اصلی شامل حدود بالا، حدود پایین و اصول غلبه می باشند که در ادامه برای حالات مختلف مساله ای مورد بحث به تفصیل معرفی خواهند شد.

۳. اصول غلبه

برای قرار گرفتن دو کار همجوار در یک توالی با توجه به زودکرددار یا دیرکرددار بودن دو کار قبل و بعد از جابجایی، ۱۶ حالت مختلف امکان پذیر است که از میان این حالات ۹ حالت موجه و ۷ حالت دیگر غیرموجه خواهد بود. یکی از حالات موجه، زمانی است که دو کار i و j بدون در نظر گرفتن ترتیب آنها در توالی زودکرددار باشند. برای این حالت اگر F_{ji} و F_{ij} به ترتیب جریمه های زودکرد کارها قبل و بعد از جابجایی باشند، اصل غلبه مربوطه را می توان به صورت زیر اثبات نمود. در این رابطه p و t به ترتیب زمان شروع دو کار مجاور و زمان پردازش کارها می باشند.

$$\left. \begin{aligned} F_{ij} &= h_i(d_i - t - p)^2 + h_j(d_j - t - 2p)^2 \\ F_{ji} &= h_i(d_i - t - 2p)^2 + h_j(d_j - t - p)^2 \end{aligned} \right\} \Rightarrow$$

$$F_{ij} - F_{ji} = h_i(-3p^2 + 2d_i p - 2tp) + h_j(3p^2 - 2d_j p + 2tp) \leq 0$$

$$\Rightarrow (3p^2 + 2tp)(h_j - h_i) \leq 2p(h_j d_j - h_i d_i)$$

$$\Rightarrow \frac{3}{2}p + t \leq \frac{h_j d_j - h_i d_i}{h_j - h_i}$$

^۲ - time-tabling

^۱ - beam search

$T_i T_i$ برای حالتی به کار می رود که قبل از جابجایی کار i زودکردار و کار j دیرکردار باشد ولی پس از تعویض جای کارها هر دو کار دیرکردار شوند و یا نماد $E_i T_j / T_j E_i$ یعنی کار i در هر دو حالت زودکردار و کار j نیز در هر دو حالت دیرکردار است.

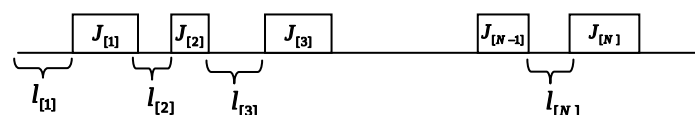
۴. حد پایین

برای محاسبه این حد پایین برای مساله ی $1 | p_j = p | h_j E_j^2 + w_j T_j^2$ ابتدا تمام کارها در موعد تحویلشان زمان بندی می شوند.

در این حالت ممکن است تعدادی از کارها با هم تداخل داشته باشند و یا اینکه بین کارها فاصله ایجاد شود؛ حالت اول موجب غیر موجه شدن توالی کارها شده و حالت دوم نیز با فرض عدم بیکاری ماشین در تناقض است. در اینجا سعی می شود تا جریمه ناشی از حذف این دو حالت غیر موجه (تداخل کارها و بیکاری ماشین) محاسبه شده و مجموع آن به عنوان حد پایینی برای مساله در نظر گرفته شود.

از میان بینهایت حالتی که برای از بین بردن تداخل میان کارهای i و j ممکن است وجود داشته باشد، می توان اثبات کرد که ۶ حالت خاص نسبت به سایر حالات جریمه کمتری دارند که حداقل جریمه ناشی از این حالات در گام های ۲ و ۳ الگوریتم محاسبه شده است.

برای حذف فاصله های بیکاری ماشین و محاسبه ی حد پایین هنگامی که هیچ کدام از کارها با هم تداخل نداشته باشند به صورت زیر عمل می شود. در این حالت اگر تعداد کارها با N کار n ام موجود در توالی با $J_{[n]}$ نمایش داده شود، آنگاه قبل از کار n ام یک بازه ی بیکاری ماشین به طول $l_{[n]} \geq 0$ وجود خواهد داشت. (مطابق شکل ۱)



شکل ۱. وضعیت قرار گرفتن کارها و بازه های بیکاری

از آنجایی که ترتیب بهینه کارها برای حداقل شدن مجموع جریمه های زودکرد از قبل مشخص نیست، بنابراین باید ترتیبی از کارها انتخاب شود که حد پایین بودن جریمه نهایی را تضمین کند. رابطه ی فوق را می توان در فضای برداری به صورت زیر نیز نوشت.

جدول ۱. اصول غلبه مربوط به جفت کارهای مجاور در

$$\text{مساله ی } 1 | p_j = p | h_j T_j^2 + w_j T_j^2$$

وضعیت کارهای مجاور	اصل غلبه مربوطه
$E_i E_j / E_j E_i$	$\frac{3}{2}p + t \leq \frac{h_j d_j - h_i d_i}{h_j - h_i}$
$T_i T_j / T_j T_i$	$\frac{3}{2}p + t \leq \frac{w_j d_j - w_i d_i}{w_j - w_i}$
$E_i T_j / T_j E_i$	$\frac{3}{2}p + t \leq \frac{w_j d_j - h_i d_i}{w_j - h_i}$
$T_i E_j / E_j T_i$	$\frac{3}{2}p + t \leq \frac{h_j d_j - w_i d_i}{h_j - w_i}$
$E_i E_j / E_j T_i$	$\frac{3}{2}p - d_j + t \leq \frac{h_i (d_i - t - p)^2 - w_i (d_i - t - 2p)^2}{2ph_j}$
$T_i T_j / E_j T_i$	$\frac{3}{2}p - d_i + t \leq \frac{h_j (d_j - t - p)^2 - w_j (d_j - t - 2p)^2}{2pw_i}$
$E_i T_j / E_j E_i$	$\frac{3}{2}p - d_i + t \leq \frac{h_j (d_j - t - p)^2 - w_j (d_j - t - 2p)^2}{2ph_i}$
$E_i T_j / T_j T_i$	$\frac{3}{2}p - d_j + t \leq \frac{w_i (d_i - t - 2p)^2 - h_i (d_i - t - p)^2}{2pw_j}$
$E_i T_j / E_j T_i$	$h_i (d_i - t - p)^2 + w_j (d_j - t - 2p)^2 \leq w_i (d_i - t - 2p)^2 + h_j (d_j - t - p)^2$

سایر حالات موجه به همراه اصول غلبه مربوطه در جدول (۱) خلاصه شده است.

نمادهای موجود در ستون اول این جدول وضعیت دو کار مجاور قبل و بعد از جابجا را نشان می دهند؛ بدین ترتیب که مثلاً $E_i T_j /$

با فرض ثابت بودن ترتیب کارها در توالی فوق، جریمه ناشی از حذف بازه های بیکاری ماشین از رابطه ی زیر محاسبه خواهد شد.

$$TP = h_{J_{[1]}} (l_{[1]})^2 + h_{J_{[2]}} (l_{[1]} + l_{[2]})^2 + h_{J_{[3]}} (l_{[1]} + l_{[2]} + l_{[3]})^2 + \dots + h_{J_{[N]}} (l_{[1]} + l_{[2]} + \dots + l_{[N]})^2 = \sum_{n=1}^N \left[h_{J_{[n]}} \left(\sum_{i=1}^n l_{[i]} \right)^2 \right] \quad (1)$$

گام ۸) فرض کنید تعداد کارهای جدید M باشد (تعداد کارهای جدید با تعداد بازه های بیکاری ماشین برابر است) و طول بازه ی بیکاری m را برابر با $l_{[m]}$ در نظر بگیرید. جریمه زودکرد کارهای جدید (h_k) را به ترتیب غیر صعودی مرتب کنید به نحوی که $h_{[1]} \geq h_{[2]} \geq \dots \geq h_{[M-1]} \geq h_{[M]}$ باشد. آنگاه خواهیم داشت:

$$TP_2 = \sum_{m=1}^M \left[h_{[m]} \left(\sum_{m=1}^M l_{[m]} \right)^2 \right]$$

گام ۹) حد پایین مساله ی $1|p_j = p|h_j E_j^2 + w_j T_j^2$ به صورت $LB = TP_1 + TP_2$ محاسبه می شود.

۵. الگوریتم ابتکاری اول (HA1)

در این روش ابتکاری از یک الگوریتم حریصانه برای یافتن جواب های مناسب و قابل قبول برای مساله ی مورد بررسی استفاده می شود. قبل از بیان گام های روش لازم است تا به این نکته اشاره شود که اگرچه این الگوریتم برای مساله ی $1|p_j = p|h_j E_j^2 + w_j T_j^2$ یافتن توالی های ابتکاری برای مساله ی $1|h_j E_j^2 + w_j T_j^2$ نیز می توان استفاده کرد.

گام ۱) تمام کارها را به ترتیب موعدهای تحویلشان شماره گذاری کنید به نحوی که

$$d_1 \leq d_2 \leq \dots \leq d_{N-1} \leq d_N$$

گام ۲) اولین کار در ترتیب فوق را در ابتدای توالی قرار داده و جریمه زودکرد یا دیرکرد آنرا محاسبه نمایید. $i = 2$

گام ۳) کار i ام را انتخاب کرده و آنرا در i مکان امکان پذیر در توالی قرار دهید. مجموع جریمه های زودکرد و دیرکرد کارها را در هر مرتبه محاسبه کرده و کار i را در مکانی که کوچکترین جریمه را ایجاد می کند زمان بندی نمایید.

گام ۴) اگر $i = N$ بود توالی نهایی به دست آمده است (پایان الگوریتم) و در غیر این صورت قرار دهید $i = i + 1$ و به گام سوم برگردید.

۶. الگوریتم ابتکاری دوم (HA2)

قبل از اینکه به تعریف گام های الگوریتم پرداخته شود لم های به کار رفته در این روش ابتکاری تشریح می شوند.

لم ۱: در مساله $1|h_j E_j^2 + w_j T_j^2$ اگر تعدادی کار دلخواه بهم چسبیده باشند و لحظه ی شروع اولین کار t باشد، در صورتی که مجموعه ی کارهای زودکردار در این حالت را با $G_E(t)$ و

$$TP = [h_{j_{[1]}}, h_{j_{[2]}}, \dots, h_{j_{[N]}}]$$

$$\left[(l_{[1]})^2, (l_{[1]} + l_{[2]})^2, \dots, (l_{[1]} + l_{[2]} + \dots + l_{[N]})^2 \right]^T \quad (2)$$

با توجه به این نکته که مقادیر درایه های بردار دوم افزایشی است، بنابراین برای حداقل شدن TP می بایست درایه های بردار اول به صورت غیر صعودی چیده شده و سپس جریمه زودکرد محاسبه گردد. گام های الگوریتم مربوط به این حد پایین به صورت زیر است. گام های اول تا پنجم مربوط به محاسبه جریمه ناشی از حذف تداخل کارها و گام های ششم تا نهم مربوط به محاسبه جریمه ناشی از حذف بیکاری ماشین می باشد.

گام ۱) تمام کارها را در موعد تحویل خود زمان بندی کنید. (در این حالت جریمه زودکرد و دیرکرد کارها صفر است)

گام ۲) دو کار اولی از توالی که با هم تداخل دارند را یافته و آنها را به ترتیب کارهای i و j بنامید.

گام ۳) قرار دهید $Z^{(1)} = h_j / w_j$ و $Z^{(2)} = h_i / w_i$

اگر $Z^{(1)} > 1$ بود $F_1 = Z_1$ و در غیر این صورت $F_1 = Z_2$ است.

اگر $Z^{(2)} > 1 + \frac{2p}{f}$ بود $F_2 = Z_4$ و در غیر این صورت اگر $Z^{(2)} > 1$ بود $F_2 = Z_5$ و در غیر این صورت اگر $Z^{(2)} > \left(1 + \frac{2p}{f}\right)^{-1}$ بود $F_2 = Z_6$ و در غیر این صورت $F_2 = Z_3$ است.

گام ۴) قرار دهید $IP_{i,j} = \min\{F_1, F_2\}$ که نشان دهنده حداقل جریمه ناشی از حذف تداخل میان دو کار i و j است.

گام ۵) کارهای i و j را از توالی حذف کرده و به گام دوم بازگردید. این کار را تا جایی ادامه دهید که هیچ دو کار متداخلی در توالی باقی نماند. مجموع جریمه های محاسبه شده در گام های فوق را TP_1 بنامید.

گام ۶) کارهایی که در گام های قبلی برای آنها جریمه تداخل محاسبه شده است را با توجه به حالتی که کمترین جریمه را داشته اند زمان بندی کرده و سایر کارها را در موعدهای تحویلشان قرار دهید.

گام ۷) کارها را دسته بندی کنید به نحوی که کارهای متداخل و بهم چسبیده در یک دسته قرار گیرند. کارهای موجود در دسته K را به عنوان یک کار جدید k' در نظر گرفته و مشخصات کار جدید را از روابط زیر محاسبه نمایید.

$$\begin{aligned} S_{k'} &= \min_{j \in K} \{S_j\} & F_{k'} &= \max_{j \in K} \{F_j\} \\ h_{k'} &= \min_{j \in K} \{h_j\} & w_{k'} &= \min_{j \in K} \{w_j\} \end{aligned}$$

این نقطه نزولی است و بنابراین می توان نتیجه گرفت که تابع مذکور محدب بوده و دارای نقطه ی مینیمم برابر با $t = t^*$ می باشد.

نتیجه ۲: نقاط شکست جریمه برای مساله ی

$$1|h_j E_j^2 + w_j T_j^2$$

در لم ۲ مشاهده شد که برای مساله ی $1|h_j E_j^2 + w_j T_j^2$ اگر تعدادی کار دلخواه بهم چسبیده (مثلاً مجموعه کارهای G) به سمت چپ و راست توالی جابجا شوند، نمودار هزینه زودکرد و دیرکرد محدب است. همچنین اثبات شد که نمودار مذکور پیوسته و قطعه به قطعه درجه دو است و نقاط شکست آن در نقاطی قرار دارد که زمان تکمیل یکی از کارهای مجموعه G با موعد تحویل یکی از کارهای این مجموعه بر هم منطبق شوند. یعنی اگر در مجموعه G به اندازه ی m عدد کار وجود داشته باشد، حداکثر تعداد نقاط شکست برای تابع جریمه برابر با m^2 خواهد بود.

ابتدا به این نکته اشاره می شود که اگرچه این الگوریتم برای مساله ی $1|h_j E_j^2 + w_j T_j^2$ ارائه شده است ولی از آن برای یافتن توالی های ابتکاری برای مساله ی $1|h_j E_j^2 + w_j T_j^2$ نیز می توان استفاده کرد. این الگوریتم ابتکاری در دو مرحله اجرا می شود که در مرحله اول (گام های ۱ تا ۷) سعی شده تا با کمترین میزان جریمه تداخل میان کارها حذف شده و در مرحله دوم (گام های ۸ تا ۱۳) بازه های بیکاری ماشین برداشته شود تا توالی حاصل هم امکان پذیر بوده و هم اینکه با فرض عدم بیکاری ماشین در تناقض نباشد.

گام ۱) تمام کارها را در مواعدهای تحویلشان قرار دهید.

گام ۲) اولین کار تداخلی (کاری با کوچکترین موعد تحویل که حداقل با یکی از کارهای دیگر تداخل داشته باشد) را کار i بنامید. کاری که با کار i تداخل داشته و کوچکترین موعد تحویل را داشته باشد را کار j بنامید.

گام ۳) فرض کنید کار i به گروهی از کارها (مثلاً G متعلق باشد، در غیر این صورت کار i را به عنوان گروه جدید G در نظر بگیرید. فرض کنید تعداد کارهای موجود در گروه G برابر با nG باشد. قرار دهید $m = nG + 1$

گام ۴) کار j را در m امین مکان بین کارهای مجموعه G قرار داده و کارهای قبلی موجود در مکان های m تا $nG + 1$ از مجموعه G را به اندازه ی p_j (زمان پردازش کار j) به سمت راست جابجا کنید تا داخلی میان کارهای مجموعه G نباشد. جریمه زودکرد و دیرکرد کارهای مجموعه G را محاسبه کرده و با Z_G نشان دهید.

گام ۵) مجموعه کارهای G را تا رسیدن به یکی از نقاط شکست

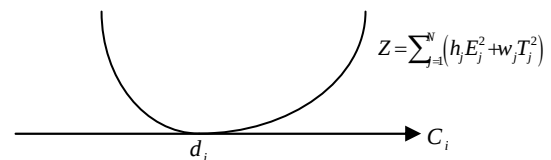
مجموعه ی کارهای دیرکردار را با $G_T(t)$ نمایش دهیم، آنگاه تابع $F(t) = \sum_{i \in G_E(t)} (h_i E_i(t)) - \sum_{j \in G_T(t)} (w_j T_j(t))$ تابعی پیوسته و نزولی بوده و به ازای یک مقدار حقیقی $t = t^*$ رابطه ی $F(t^*) = 0$ برقرار خواهد بود.

برای اثبات این لم می توان نشان داد که با کم شدن t به سمت منفی بینهایت، تعداد کارهای زودکردار افزایش می یابد و در نهایت رابطه ی $F(t)$ مقداری مثبت می گیرد و برعکس با زیاد شدن t به سمت مثبت بینهایت، تعداد کارهای دیرکردار افزایش یافته و رابطه ی $F(t)$ منفی خواهد شد. اکنون با توجه به پیوسته بودن $F(t)$ می توان نتیجه گرفت که

$$\exists t^* \in t \mid F(t^*) = 0$$

لم ۲: برای مساله ی $1|h_j E_j^2 + w_j T_j^2$ اگر تعدادی کار را به دلخواه کنار هم چیده شده باشند و بعضی زودکردار و بعضی دیرکردار باشند، آنگاه اگر مجموعه کارها را بدون جابجا کردن ترتیب کارها به سمت چپ یا راست منتقل کنیم، نمودار هزینه دیرکرد و زودکرد محدب خواهد بود.

برای اثبات این لم می توان نشان داد که نمودار هزینه ی کل برای هر ترتیب دلخواه از کارهای چیده شده کنار هم نموداری پیوسته است. اگر مجموع جریمه زودکرد و دیرکرد کارها به صورت $Z = \sum_{j=1}^N (h_j E_j^2 + w_j T_j^2)$ نشان داده شود، می توان مقدار Z را برابر با مجموع جریمه های تک تک کارها در نظر گرفت. مشخص است که نمودار جریمه زودکرد و دیرکرد یک کار نسبت به زمان تکمیل آن کار، یک نمودار محدب و پیوسته است (مطابق شکل ۲) و چون Z برابر با مجموع چند تابع پیوسته تعریف شده است، خود نیز تابعی پیوسته خواهد بود.



شکل ۲. نمودار جریمه زودکرد و دیرکرد درجه دوم کارها براساس زمان های تکمیل

در مساله ی $1|h_j E_j^2 + w_j T_j^2$ حالتی را در نظر بگیرید که تعدادی از کارها بهم چسبیده باشند و زمان شروع اولین کار t باشد. اگر مجموعه ی کارهای زودکردار در این شرایط با $G_E(t)$ و مجموعه کارهای دیرکردار با $G_T(t)$ نمایش داده شوند، در لم ۱ نشان داده شد که

$$F(t) = \sum_{i \in G_E(t)} (h_i E_i(t)) - \sum_{j \in G_T(t)} (w_j T_j(t))$$

تابعی پیوسته و نزولی است و همچنین به ازای مقداری مانند $t = t^*$ عبارت $F(t^*) = 0$ خواهد بود. همچنین اثبات می شود که تابع $Z(t)$ در سمت راست نقطه ی $t = t^*$ صعودی و در سمت چپ

استفاده شود. نتایج محاسباتی نشان می دهد که این الگوریتم قادر است تا در مدت زمان بسیار اندک تا حدود زیادی جریمه زودکرد و دیرکرد توالی اولیه ورودی را کاهش دهد. از این الگوریتم برای بهبود توالی های اولیه مربوط به مساله $1|h_j E_j^2 + w_j T_j^2$ نیز می توان استفاده کرد. در الگوریتم شاخه و کران پیشنهادی در صورتی که روش بهبوددهنده بر روی الگوریتم ابتکاری اول (HA1) پیاده شود به این مجموعه IHA^1 و در صورتی که بر روی الگوریتم HA2 پیاده سازی شود به آن $IHA2$ اطلاق می شود. گام های این الگوریتم ابتکاری به صورت زیر است.

گام ۱) از انتهای توالی اولیه شروع کرده و به اولین کار دیرکرداری که رسیدید آنرا کار i بنامید.

گام ۲) اصول غلبه مربوط به جفت کارهای مجاور را برای کار i و کار ماقبل آن تست کرده و در صورت لزوم جای دو کار را تعویض نمایید. در صورتی که جابجایی انجام گیرد مجدداً به گام ۲ برگشته و در غیر این صورت به گام ۳ بروید.

گام ۳) نزدیک ترین کار دیرکردار قبل از کار i را کار j بنامید. مجموعه کارهای i و j و کارهای بین آنها را مجموعه G بنامید. در صورتی که مجموعه G به سمت چپ منتقل شود و جریمه کمتری ایجاد کند به گام ۴ رفته و در غیر این صورت به گام ۵ بروید.

گام ۴) نزدیک ترین کار زودکردار قبل از مجموعه G را کار k بنامید. جای کار k را با تک تک کارهای دیرکردار مجموعه G تعویض کرده و حالتی که بیشترین کاهش جریمه را ایجاد می کند را انتخاب نمایید.

گام ۵) در صورتی که قبل از کار i کار دیرکردار دیگری وجود داشته باشد، آنرا کار i بنامید و به گام ۲ برگردید و در غیر این صورت مرحله اول الگوریتم به اتمام رسیده است و به گام ۶ بروید. **گام ۶)** از ابتدای توالی اولیه شروع کرده و به اولین کار زودکرداری که رسیدید آنرا کار i بنامید.

گام ۷) اصول غلبه مربوط به جفت کارهای مجاور را برای کار i و کار بعد آن تست کرده و در صورت لزوم جای دو کار را تعویض نمایید. در صورتی که جابجایی انجام گیرد مجدداً به گام ۷ برگشته و در غیر این صورت به گام ۸ بروید.

گام ۸) نزدیک ترین کار زودکردار بعد از کار i را کار j بنامید. مجموعه کارهای i و j و کارهای بین آنها را مجموعه G بنامید. در صورتی که مجموعه G به سمت راست منتقل شود و جریمه کمتری ایجاد کند به گام ۹ رفته و در غیر این صورت به گام ۱۰ بروید.

گام ۹) نزدیک ترین کار دیرکردار بعد از مجموعه G را

جریمه (طبق لم ۲) به سمت چپ جابجا کنید. اگر جریمه کل کم شد این کار را تا رسید به حداقل جریمه برای نقاط شکست سمت چپ تکرار نمایید. در غیر این صورت مجموعه کارهای G را تا رسیدن به نقطه ی شکستی با کمترین جریمه به سمت راست جابجا کنید. حداقل جریمه کارهای مجموعه G را در این حالت با Z_G^m نشان دهید.

گام ۶) اگر $m \geq 1$ باشد قرار دهید $m = m - 1$ و به گام ۴ برگردید وگرنه به گام ۷ بروید.

گام ۷) فرض کنید $Z_g^{m'} = \min_m \{Z_g^m\}$ باشد. کار j را در مکان m' بین کارهای مجموعه G زمان بندی کنید. کار j اکنون جزء مجموعه G است. در صورتی که هیچ دو کاری تداخل نداشته باشند به مرحله دوم از الگوریتم (گام ۸) رفته و در غیر این صورت به گام ۲ برگردید.

گام ۸) کارها را با همان ترتیب حاصل از مرحله اول در نظر گرفته و آنها را به نحوی دسته بندی کنید که کارهای چسبیده بهم در یک دسته قرار گیرند. دسته ی k ام از کارها را به عنوان یک کار جدید k' تعریف کرده و مشخصات آنرا از روابط زیر محاسبه نمایید.

$$S_{k'} = \min_{j \in K} \{S_j\} \quad F_{k'} = \max_{j \in K} \{F_j\}$$

$$h_{k'} = \text{ave}_{j \in K} \{h_j\} \quad w_{k'} = \text{ave}_{j \in K} \{w_j\}$$

گام ۹) اولین کار موجود در توالی را در زمان صفر (لحظه ی شروع) زمان بندی کنید.

گام ۱۰) اگر $j \leq 1$ بود به گام ۱۳ و در غیر این صورت به گام ۱۱ بروید.

گام ۱۱) کار موجود در مکان j ام توالی $(J_{[j]})$ و کار موجود در مکان $j-1$ ام توالی $(J_{[j-1]})$ را انتخاب کنید. اگر $L_{j-1} > p_{J_{[j]}}$ و نیز $\frac{h_{J_{[j]}} p_{J_{[j-1]}}}{h_{J_{[j-1]}} p_{J_{[j]}}} \leq \frac{2L_{j-1} - p_{J_{[j]}}}{2L_j + p_{J_{[j-1]}}}$ بود و با اینکه اگر $L_{j-1} \leq p_{J_{[j]}}$ و $\frac{h_{J_{[j]}} p_{J_{[j-1]}}}{h_{J_{[j-1]}} p_{J_{[j]}}} \leq \frac{(L_{j-1})^2}{2L_j + p_{J_{[j-1]}}}$ برقرار بود جای دو کار را در توالی باهم تعویض کنید.

گام ۱۲) قرار دهید $j = j - 1$ و مجدداً به گام ۱۱ برگردید.

گام ۱۳) اگر i برابر با تعداد کارها بود، آنگاه کارها را براساس ترتیب به دست آمده کنار هم بچینید (پایان الگوریتم) و در غیر این صورت ابتدا قرار دهید $i = i + 1$ و سپس $j = i$ را در نظر بگیرید. به گام ۱۱ بروید.

۷. الگوریتم بهبوددهنده

در این الگوریتم سعی شده تا از اصول غلبه مربوط به جفت کارهای مجاور و همچنین لم ۱ برای بهبود کیفیت تابع هدف

¹ - Improved Heuristic Algorithm 1

نسبت به مسائل با پراکندگی بالا حل می شوند. برای تولید موعدهای تحویل کارها از دو پارامتر عامل دیرکرد (τ) و عامل دامنه موعد تحویل (R) استفاده می شود. اگر متوسط موعد تحویل کارها را با $\bar{d}$ و متوسط زمان های پردازش را با $\bar{P}$ نمایش دهیم، موعدهای تحویل از روابط $\bar{d} = (1-\tau)n\bar{P}$ و $d_i \in U\left[\bar{d}\left(1-\frac{R}{\tau}\right), \bar{d}\left(1+\frac{R}{\tau}\right)\right]$ به دست خواهد آمد.

از آنجایی که مشاهدات اولیه نشان می دهد تغییر پارامترهای τ و R تاثیر چندانی در میزان دشواری مسائل نمونه و زمان های اجرای الگوریتم شاخه و کران ندارد، برای این دو پارامتر دسته بندی انجام نشده و مقادیر آنها به صورت $\tau = 0/4$ و $R = 0/8$ انتخاب می شود. تعداد مسائل نمونه ایجاد شده در هر دسته ۱۰ عدد می باشد.

روش شاخه و کران مبتنی بر جستجوی عمقی با استفاده از حدود بالا و پایین و نیز اصول غلبه مطرح شده در بخش های قبلی در محیط نرم افزار Visual C++ برنامه نویسی و پیاده سازی شده است. برای حل هر مساله در روش شاخه و کران محدودیت زمانی ۳۶۰۰ ثانیه در نظر گرفته شده است و اگر مساله ای تا این زمان با روش شاخه و کران به جواب نرسد، رویه حل شاخه و کران برای آن مساله متوقف می شود. نتایج حل دسته های گوناگون در جدول ۲ نمایش داده شده است.

کار k بنامید. جای کار k را با تک تک کارهای زود کردار مجموعه G تعویض کرده و حالتی که بیشترین کاهش جریمه را ایجاد می کند را انتخاب نمایید.

گام ۱۰) در صورتی که بعد از کار i کار دیر کردار دیگری وجود داشته باشد، آنرا کار i بنامید و به گام ۷ برگردید و در غیر این صورت الگوریتم به پایان رسیده است.

۸. نتایج محاسباتی

در این بخش به منظور سنجش میزان کارایی الگوریتم ها و روش های ارائه شده ابتدا تعدادی مساله ی نمونه به صورت تصادفی تولید شده و سپس روش های پیشنهادی بر روی آنها پیاده سازی و اجرا می شوند.

برای تولید مسائل تصادفی می بایست مقادیر چهار پارامتر جریمه زود کرد، جریمه دیرکرد، زمان های پردازش و موعدهای تحویل تعیین شود. در مسائل تولید شده جریمه های زود کرد، دیرکرد و زمان های پردازش یک مرتبه از توزیع یکنواخت گسسته در بازه ی [45,55] و بار دیگر از بازه ی [1,100] تولید شده اند [۱۸]. دسته اول میزان پراکندگی کم و دسته دوم پراکندگی زیاد را برای جریمه ها و زمان های پردازش نشان می دهد و با توجه به آزمایش های انجام گرفته بر روی مسائل نمونه مشخص شده است که مسائل با پراکندگی کمتر، ساده تر بوده و در زمان کمتری

جدول ۲. نتایج حل مسائل در دسته های مختلف

روش ابتکاری IHA1		روش ابتکاری IHA2		روش شاخه و کران	
تعداد	درصد	تعداد	درصد	میانگین	میانگین درصد گره های
تعداد کار	نمونه های	تعداد کار	نمونه های	میانگین زمان	میانگین درصد گره های
پراکندگی	اختلاف با	پراکندگی	اختلاف با	حل (ثانیه)	قطع شده به دلیل
بهبینه	جواب بهینه	بهبینه	جواب بهینه	بهبینه	حد پایین اصول غلبه
۱۰	۰/۲۸	۹	۰/۲۷	۶/۱۸	۳۷/۸۱
۱۵	۱/۱۳	۹	۰/۳۹	۳۶/۱۱	۷۴/۷۵
۲۰	۲/۵۷	۸	۰/۹۵	۶۳۱/۰۵	۶۷/۳۷
۲۵	۲/۷۳	۷	۱/۸۲	۱۱۰۵/۴۴	۵۸/۹۳
۳۰	۳/۸۶	۶	۲/۸۸	۱۹۳۴/۸۲	۶۶/۰۸
۳۵	۵/۰۶	۶	۳/۴۹	۲۸۱۶/۳۷	۷۵/۸۲
۱۰	۱/۴۲	۹	۰/۴۶	۸/۴۷	۶۱/۲۹
۱۵	۳/۶۱	۷	۱/۱۳	۴۲۷/۳۷	۵۸/۱۱
۲۰	۴/۷۱	۵	۳/۹۲	۹۱۶/۸۸	۶۷/۲۴
۲۵	۷/۴۸	۳	۶/۰۷	۱۸۶۰/۳۹	۶۱/۸۸
۳۰	۱۱/۹۲	۲	۹/۳۶	۲۴۳۵/۱۷	۷۴/۹۰
۳۵	۱۲/۲۴	۲	۱۰/۹۲	۳۴۳۱/۷۲	۷۹/۳۹

می‌دهند. همچنین روش شاخه و کران نیز در ادامه بر روی مسائل نمونه پیاده‌سازی گردید که نتایج نشان می‌دهند این روش کارایی حل مسائل تا ابعاد حدود ۳۰ کار را دارد. به منظور انجام مطالعات آتی روی این موضوع می‌توان سایر فرضیات متداول موجود در مسائل زمان‌بندی مانند زمان‌های آماده‌سازی وابسته به توالی، مجاز بودن بیکاری ماشین و انقطاع کارها، زوال و یکسان نبودن زمان ورود کارها به سیستم اشاره نمود. همچنین تابع هدف جریمه‌های زودکرد و دیرکرد درجه دوم را می‌توان در سایر محیط‌های کارگاهی نیز مورد استفاده قرار داد.

مراجع

- [1] Abdul-Razaq, T., Potts, C.N., *Dynamic Programming State-Space Relaxation for Single Machine Scheduling*. Journal of the Operational Research Society 1988; 39:141–52.
- [2] Li, G., *Single Machine Earliness and Tardiness Scheduling*. European Journal of Operational Research 1997;96:546–58.
- [3] Liaw, C.F., *Abranch-and-Bound Algorithm for the Single Machine Earliness and Tardiness Scheduling Problem*. Computers and Operations Research 1999; 26: 679–93.
- [4] Valente, J.M.S., Alves, R.A.F.S., *Improved Lower Bounds for the Early/tardy Scheduling Problem with No Idle Time*. Journal of the Operational Research Society 2005; 56: 604–12.
- [5] Ow, P.S., Morton, T.E., *The Single Machine Early/tardy Problem*. Management Science 1989; 35: 177–91.
- [6] Valente, J.M.S., Alves, R.A.F.S., *Improved Heuristics for the Early/tardy Scheduling Problem with No Idle Time*. Computers and Operations Research 2005; 32: 557–69.
- [7] Valente, J.M.S., Alves, R.A.F.S., *Filtered and Recovering Beam Search Algorithms for the Early/Tardy Scheduling Problem with no Idle Time*. Computers and Industrial Engineering 2005;48: 363–75.
- [8] Kim, Y.D., Yano, C.A., *Minimizing Mean Tardiness and Earliness in Single-Machine Scheduling Problems with Unequal Due Dates*. Naval Research Logistics 1994;41:913–33.
- [9] Ventura, J.A., Radhakrishnan, S., *Single Machine Scheduling with Symmetric Earliness and Tardiness Penalties*. European Journal of Operational Research 2003;144:598–612.
- [10] Sundararaghavan, P.S., Ahmed, M.U., *Minimizing the Sum of Absolute Lateness in Single-Machine Scheduling*. Naval Research Logistics Quarterly 1984;31:325–33.
- [11] Schaller, J., *Single Machine Scheduling with Early*

در جدول ۲، ستون‌های «تعداد نمونه بهینه» نشان‌دهنده‌ی تعداد مسائلی از ۱۰ مساله می‌باشد که هرکدام از روش‌های ابتکاری و شاخه و کران به جواب بهینه رسیده‌اند. مشاهده می‌شود که روش شاخه و کران برای مسائل با پراکندگی کم (مسائل ساده‌تر) قادر به حل تمام مسائل نمونه تا ابعاد ۳۵ کار بوده است و برای مسائل با پراکندگی زیاد نیز تقریباً تمام مسائل تا ابعاد ۳۰ کار حل شده‌اند. با توجه ستون‌های «درصد اختلاف با جواب بهینه» برای روش‌های ابتکاری مشخص می‌شود که روش ابتکاری IHA2 اختلاف کمتری با جواب بهینه دارد و نسبت به روش ابتکاری IHA1 کارا تر است.

برای محاسبه‌ی «میانگین درصد گره‌های قطع شده» برای هر مساله، درصد تعداد گره‌های قطع شده نسبت به گره‌های طی شده در آن مساله محاسبه شده و سپس برای هر تعداد کار، میانگین این درصدها در مسائلی که روش شاخه و کران در آنها اجرا شده و به جواب بهینه رسیده است، محاسبه شده است. با بررسی مقادیر این ستون مشاهده می‌شود که تقریباً در تمامی مسائل با پراکندگی پایین، بیش از ۹۰ درصد گره‌ها و برای مسائل با پراکندگی بالا بیش از ۸۵ درصد گره‌ها قطع شده است که نشان‌دهنده‌ی بالا بودن کارایی اصول غلبه و حدپایین پیشنهادی می‌باشد. در ادامه جدول، دو ستون آورده شده است که در این ستون‌ها درصد تعداد گره‌های قطع شده به دلایل اصول غلبه و یا حد پایین نسبت به تعداد کل گره‌های قطع شده برای هر مساله محاسبه شده و سپس برای هر تعداد کار، میانگین این درصدها برای مسائلی که روش شاخه و کران برای آنها به جواب بهینه رسیده است، نمایش داده شده است. مقادیر این دو ستون نشان‌دهنده‌ی کارایی بیشتر حدپایین نسبت به اصول غلبه در قطع کردن شاخه‌های درخت می‌باشد.

۹. نتیجه‌گیری

در این مقاله مساله زمان‌بندی تک‌ماشینه با هدف کمینه کردن مجموع جریمه‌های زودکرد و دیرکرد درجه دوم مورد بررسی قرار گرفت. دو فرض دیگر درنظر گرفته شده در این مساله یکی مساوی بودن زمان‌های پردازش کارها و دیگری عدم بیکاری ماشین در طی دوره‌ی زمان‌بندی است. با توجه به پیچیدگی مساله‌ی مذکور که از نوع NP-hard است روش شاخه و کران که از جمله روش‌های متداول در تحلیل مسائل زمان‌بندی به‌شمار می‌رود برای حل آن به‌کار گرفته شده است.

در بخش‌های مختلف مقاله اصول غلبه و حدود بالا و پایین به‌کار رفته در روش شاخه و کران پیشنهادی تشریح شده است و در بخش نتایج محاسباتی نیز کارایی آنها توسط مسائل نمونه‌ای که به‌صورت تصادفی تولید شده است مورد ارزیابی قرار گرفت. این نتایج کارایی نسبتاً بالایی را برای الگوریتم‌های پیشنهادی نشان

- and Quadratic Tardy Penalties. Computers and Industrial Engineering 2004;46:511–32.
- [12] Gupta, S.K., Sen, T., *Minimizing a Quadratic Function of Job Lateness on a Single Machine*. Engineering Costs and Production Economics 1983;7: 187–94.
- [13] Su, L.H., Chang, P.C., *A Heuristic to Minimize a Quadratic Function of Job Lateness on a Single Machine*. International Journal of Production Economics 1998;55:169–75.
- [14] Schaller, J., *Minimizing the Sum of Squares Lateness on a Single Machine*. European Journal of Operational Research 2002;143:64–79.
- [15] Sen, T., Dileepan, P., Lind, M.R., *Minimizing a Weighted Quadratic Function of Job Lateness in the Single Machine System*. International Journal of Production Economics 1995;42:237–43.
- [16] Baker, K.R., Scudder, G.D., *Sequencing with Earliness and Tardiness Penalties: a Review*. Operations Research 1990;38:22–36.
- [17] Kanet, J.J., Sridharan, V., *Scheduling with Inserted Idle Time: Problem Taxonomy and Literature Review*. Operations Research 2000; 48: 99–110.
- [18] Valente, J.M.S., Alves, R.A.F.S., *Heuristics for the Single Machine Scheduling Problem with Quadratic Earliness and Tardiness Penalties*. Computers and Operations Research 2008;35: 3696–3713