نشـــریه بین المللـــی مهنـــدسی صنـــایع و مدیـــریت تولیـــد

شمـــاره ۳، جلـــد ۲۲، پاییـــــز ۱۳۹۰
صفحـــــه ۲۷۲-۲۶۴

http://IJIEPM.iust.ac.ir/

نشریه بین المللی مهندسی صنایع و مدیریت تولید
دانشگاه علم و صنعت ایران

ISSN: 2008-4870

حل مساله زمان‌بندی تک‌ماشینه با درنظر گرفتن جریمه‌های زودکرد و دیرکرد مرتبه دوم

کامران کیانفر و قاسم مصلحی*

کلمات کلیدی

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

چکیده:

امروزه مسائل زمان‌بندی در بسیاری از سیستم‌های تولیدی و خدماتی کاربرد وسیعی یافته‌اند. در این مقاله مساله‌ی زمان‌بندی n کار بر روی تک‌ماشین و با درنظر گرفتن فرض عدم بیکاری ماشین و همچنین مجاز نبودن انقطاع کارها و تساوی زمان‌های پردازش مورد بررسی قرار می‌گیرد. در بسیاری از مطالعات انجام شده تاکنون به دلیل وجود هزینه‌های ناشی از دیر تحویل دادن و یا هزینه‌های نگهداری، جریمه‌های زودکرد و دیرکرد کارها به‌صورت ترکیبات خطی در تابع هدف ظاهر می‌شود درصورتی‌که تابع هدف این مقاله کمینه کردن مجموع وزن‌دار جریمه‌های دیرکرد و زودکرد درجه دوم می‌باشد. در این پژوهش برای حل مساله‌ی فوق از روش شاخه و کران استفاده شده است و برای پیاده‌سازی الگوریتم شاخه وکران تعدادی اصل غلبه، حدود پایین و حدود بالا پیشنهاد شده است. در نهایت روش پیشنهادی با استفاده از تولید مسائل نمونه و پیاده‌سازی روش بر روی آنها مورد بررسی قرار گرفته و کارایی آن در حل مسائل به‌اثبات رسیده است.

۱. مقدمه

مسائل تک‌ماشینه از جمله مسائل متداول در حیطه مسائل زمان‌بندی به‌شمار رفته که در بسیاری از سیستم‌های تولیدی عملی کاربرد دارند. به‌علاوه در بسیاری از سیستم‌های تولیدی پیچیده‌تر نیز شرایط تولیدی معمولاً به یک ماشین که گلوگاه نامیده می‌شود وابسته است و ارائه یک برنامه زمان‌بندی خوب برای آن تک ماشین موجب بهبود کارایی کل سیستم و تولید بیشتر خواهد شد.

مدل‌های زمان‌بندی با درنظر گرفتن جریمه‌های زودکرد و دیرکرد به‌صورت همزمان را می‌توان با فلسفه تولید بهنگام (JIT) مقایسه نمود. تولید بهنگام بر این نکته تاکید می‌کند 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. مساله‌ی مورد بررسی در این تحقیق نیز بر مبانی فلسفه فوق طراحی شده است و هزینه‌های زودکرد و دیرکرد تولید را به‌صورت همزمان در مدل لحاظ می‌کند اما با این تفاوت که در اینجا به‌جای درنظر گرفتن جریمه‌های دیرکرد و زودکرد به‌صورت خطی که در ادبیات موضوع متداول است، این دو هزینه به‌شکل درجه دوم در مساله وارد شده‌اند. بنابراین محصولاتی که دیرکرد یا زودکرد داشته باشند با شدت بیشتری جریمه خواهند شد. از طرف دیگر، این نوع مدل‌سازی مساله تک‌ماشینه مانع از آن می‌شود تا تعداد کمی از کارها بخش بزرگی از جریمه را به خود اختصاص دهند و بنابراین موجب توزیع مناسب‌تر جریمه‌های مربوط به عدم تحویل به‌موقع بین تمامی کارها می‌شود. همچنین در بسیاری از سیستم‌های تولید عملی، ساختار هزینه به‌صورت غیرخطی تغییر می‌کند و بنابراین درنظر گرفتن هزینه‌ها به‌شکل غیرخطی در مساله، فرضی دور از واقعیت نیست.

تاریخ وصول: ۹۰/۲/۱۲
تاریخ تصویب: ۹۰/۳/۲۵

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

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

فرض دیگری که در مساله‌ی مورد بررسی لحاظ شده است، فرض عدم بیکاری ماشین است. توجه به این فرض در بسیاری از سیستم‌های تولیدی ضروری است، مثلاً در زمانی که ظرفیت ماشین در مقایسه با میزان تقاضا محدود باشد، لازم است تا برای برآوردن سفارشات مشتری‌ها ماشین به‌صورت پیوسته کار کند. همچنین در زمانی‌که هزینه‌های توقف و راه‌اندازی مجدد ماشین‌آلات نسبت به جریمه‌های زودکرد و دیرکرد تولید بزرگتر باشد

از دیگر موارد به‌کار بردن فرض عدم بیکاری برای ماشین‌ها به‌شمار می‌رود. با توجه به فرضیات بیان شده می‌توان مساله را با نماد $1\left|p_j = p\right|\sum h_j E_j^2 + w_j T_j^2$ 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 جریمه‌های زودکرد و دیرکرد را به‌صورت غیرخطی و وزن‌دار درنظر می‌گیرد نیز مساله‌ای NP-Hard می‌باشد.

۲. مرور ادبیات موضوع

مساله مشابه با فرض عدم بیکاری ماشین و جریمه‌های زودکرد و دیرکرد وزن‌دار و خطی تاکنون توسط محققین مختلفی مطالعه شده و روش‌های دقیق و ابتکاری متنوعی نیز برای آن ارائه شده است. از میان روش‌های دقیق می‌توان به الگوریتم‌های 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 روش‌های ابتکاری مختلفی نیز تاکنون برای مساله مذکور ارائه شده که از جمله می‌توان قوانین توزیع و الگوریتم‌های جستجوی پرتویی[1] معرفی شده توسط او و مورتن[۵] و والنته و مارتز[۶-۷] را نام برد.

مساله تک‌ماشینه با جریمه‌های زودکرد و دیرکرد خطی و بدون وزن با فرض مجاز بودن بیکاری برای ماشین توسط کیم و یانو [۸] و ونتورا و رادهاکریشنان [۹] مطرح شده است که در تحقیق اول، یک الگوریتم شاخه و کران و در تحقیق دوم یک الگوریتم آزادسازی لاگرانژی برای مساله‌ی 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 می‌توان به تحقیق اسکلار [۱۱] اشاره کرد که در آن مقاله مساله‌ی تک‌ماشین با فرض مجاز بودن بیکاری ماشین و جریمه‌های زودکرد خطی و دیرکرد درجه دوم لحاظ شده است. وی یک فرایند جدول‌بندی زمانی[2] برای قرار دادن زمان‌های 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؛ درحالی‌که سو و چانگ [۱۳] و اسکالار [۱۴] همین مساله را در حالتی‌که بیکاری ماشین مجاز باشد درنظر گرفته و برای آن یک فرایند جدول‌بندی زمانی و چند روش ابتکاری دیگر پیشنهاد کردند. سن و سایرین [۱۵] برای مساله‌ی تک‌ماشینه با تابع هدف هزینه دیرکرد درجه دوم وزن‌دار در حالتی‌که 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 زمان‌بندی با جریمه‌های دیرکرد و زودکرد نیز می‌توان به تحقیقات بیکر و سودر [۱۶]، کانت و سریدهاران [۱۷] اشاره نمود.

در این تحقیق از میان روش‌های حل دقیق مورد استفاده در مسائل زمان‌بندی، روش شاخه و کران انتخاب شده است. در روش شاخه و کران سه جزء اصلی وجود دارد که می‌بایست برای پیاده‌سازی الگوریتم تعریف شده باشند؛ این سه جزء اصلی شامل حدود بالا، حدود پایین و اصول غلبه می‌باشند که در ادامه برای حالات مختلف مساله‌ی 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 امکان‌پذیر 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 درنظر گرفتن ترتیب آنها در توالی زودکرددار باشند. برای این حالت اگر F_{ij} و F_{ji} به‌ترتیب جریمه‌های زودکرد کارها قبل و بعد از جابجایی باشند، اصل غلبه مربوطه را می‌توان به‌صورت زیر اثبات نمود. در این رابطه t و p به‌ترتیب زمان شروع دو کار مجاور و زمان پردازش کارها می‌باشند.

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

$$F_{ij}-F_{ji} = h_i(-3p^2+2d_ip-2tp)+h_j(3p^2-2d_jp+2tp)\le 0$$

$$\Rightarrow (3p^2+2tp)(h_j-h_i)\le 2p(h_jd_j-h_id_i)$$

$$\Rightarrow \frac{3}{2}p+t\le\frac{h_jd_j-h_id_i}{h_j-h_i}$$

---

[1] - *beam search*

[2] - *time-tabling*

جدول ۱. اصول غلبه مربوط به جفت کارهای مجاور در مساله‌ی $1 \mid p_j = p \mid h_jT_j^2 + w_jT_j^2$

وضعیت کارهای مجاور	اصل غلبه مربوطه
E_iE_j / E_jE_i	$\frac{3}{2}p + t \le \frac{h_jd_j - h_id_i}{h_j - h_i}$
T_iT_j / T_jT_i	$\frac{3}{2}p + t \le \frac{w_jd_j - w_id_i}{w_j - w_i}$
E_iT_j / T_jE_i	$\frac{3}{2}p + t \le \frac{w_jd_j - h_id_i}{w_j - h_i}$
T_iE_j / E_jT_i	$\frac{3}{2}p + t \le \frac{h_jd_j - w_id_i}{h_j - w_i}$
E_iE_j / E_jT_i	$\frac{3}{2}p - d_j + t \le \frac{h_i(d_i - t - p)^2 - w_i(d_i - t - 2p)^2}{2ph_j}$
T_iT_j / E_jT_i	$\frac{3}{2}p - d_i + t \le \frac{h_j(d_j - t - p)^2 - w_j(d_j - t - 2p)^2}{2pw_i}$
E_iT_j / E_jE_i	$\frac{3}{2}p - d_i + t \le \frac{h_j(d_j - t - p)^2 - w_j(d_j - t - 2p)^2}{2ph_i}$
E_iT_j / T_jT_i	$\frac{3}{2}p - d_j + t \le \frac{w_i(d_i - t - 2p)^2 - h_i(d_i - t - p)^2}{2pw_j}$
E_iT_j / E_jT_i	$h_i(d_i - t - p)^2 + w_j(d_j - t - 2p)^2 \le w_i(d_i - t - 2p)^2 + h_j(d_j - t - p)^2$

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 می‌دهند؛ بدین ترتیب که مثلاً E_iT_j / T_jT_i برای حالتی به کار می‌رود که قبل از جابجایی کار i زودکرددار و کار j دیرکرددار باشد ولی پس از تعویض جای کارها هر دو کار دیرکرددار شوند و یا نماد E_iT_j/T_jE_i یعنی کار i در هر دو حالت زودکرددار و کار j نیز در هر دو حالت دیرکرددار است.

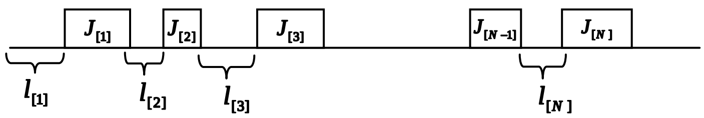
۴. حد پایین

برای محاسبه این حد پایین برای مساله‌ی $1 \mid p_j = p \mid h_jE_j^2 + w_jT_j^2$ ابتدا تمام کارها در موعد تحویلشان زمان‌بندی می‌شوند.

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 می‌شود تا جریمه ناشی از حذف این دو حالت غیر موجه (تداخل کارها و بیکاری ماشین) محاسبه شده و مجموع آن به عنوان حدپایینی برای مساله درنظر گرفته شود.

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

برای حذف فاصله‌های بیکاری مشین و محاسبه‌ی حدپایین هنگامی که هیچ کدام از کارها با هم تداخل نداشته باشند به صورت زیر عمل می‌شود. در این حالت اگر تعداد کارها با N و کار nام موجود در توالی با $J_{[n]}$ نمایش داده شود، آنگاه قبل از کار nام یک بازه‌ی بیکاری ماشین به طول $l_{[n]} \ge 0$ وجود خواهد داشت. (مطابق شکل ۱)



شکل ۱. وضعیت قرار گرفتن کارها و بازه‌های بیکاری

با فرض ثابت بودن ترتیب کارها در توالی فوق، جریمه ناشی از حذف بازه‌های بیکاری ماشین از رابطه‌ی زیر محاسبه خواهد شد.

$$TP = h_{J_{[1]}}\left(l_{[1]}\right)^2 + h_{J_{[2]}}\left(l_{[1]} + l_{[2]}\right)^2 + h_{J_{[3]}}\left(l_{[1]} + l_{[2]} + l_{[3]}\right)^2 + \ldots + h_{J_{[N]}}\left(l_{[1]} + l_{[2]} + \ldots + l_{[N]}\right)^2 = \sum_{n=1}^{N}\left[h_{J_{[n]}}\left(\sum_{i=1}^{n} l_{[i]}\right)^2\right] \quad (۱)$$

از آنجایی که ترتیب بهینه کارها برای حداقل شدن مجموع جریمه‌های زودکرد از قبل مشخص نیست، بنابراین باید ترتیبی از کارها انتخاب شود که حد پایین بودن جریمه نهایی را تضمین کند. رابطه‌ی فوق را می‌توان در فضای برداری به‌صورت زیر نیز نوشت.

$$TP = \left[h_{J_{[1]}}, h_{J_{[2]}}, \ldots, h_{J_{[N]}} \right]. \left[\left(l_{[1]}\right)^2, \left(l_{[1]}+l_{[2]}\right)^2, \ldots, \left(l_{[1]}+l_{[2]}+\ldots+l_{[N]}\right)^2 \right]^T \qquad (۲)$$

با توجه به این نکته که مقادیر درایه‌های بردار دوم افزایشی است، بنابراین برای حداقل شدن TP می‌بایست درایه‌های بردار اول به‌صورت غیر صعودی چیده شده و سپس جریمه زودکرد محاسبه گردد. گام‌های الگوریتم مربوط به این حد پایین به‌صورت زیر است. گام‌های اول تا پنجم مربوط به محاسبه جریمه ناشی از حذف تداخل کارها و گام‌های ششم تا نهم مربوط به محاسبه جریمه ناشی از حذف بیکاری ماشین می‌باشد.

گام۱) تمام کارها را در موعد تحویل خود زمان‌بندی کنید. (در این حالت جریمه زودکرد و دیرکرد کارها صفر است)

گام۲) دو کار اولی از توالی که با هم تداخل دارند را یافته و آنها را به‌ترتیب کارهای i و j بنامید.

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

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

اگر $Z^{(2)} > 1+\frac{2p}{f}$ بود $F_2 = Z_4$ و در غیر این‌صورت 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 $IP_{i,j} = \min\left\{F_1, F_2\right\}$ که نشان‌دهندی 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 جریمه‌های محاسبه شده در گام‌های فوق را TP_1بنامید.

گام ۶) کارهایی که در گام‌های قبلی برای آنها جریمه تداخل محاسبه شده است را با توجه به حالتی‌که کمترین جریمه را داشته‌اند زمان‌بندی کرده و سایر کارها را در موعدهای تحویلشان قرار دهید.

گام ۷) کارها را دسته‌بندی کنید به‌نحوی که کارهای متداخل و بهم چسبیده در یک دسته قرار گیرند. کارهای موجود در دسته‌ی Kام را به‌عنوان یک کار جدید k' درنظر گرفته و مشخصات کار جدید را از روابط زیر محاسبه نمایید.

$$S_{k'} = \min_{j\in K}\left\{S_j\right\} \qquad F_{k'} = \max_{j\in K}\left\{F_j\right\}$$
$$h_{k'} = \min_{j\in K}\left\{h_j\right\} \qquad w_{k'} = \min_{j\in K}\left\{w_j\right\}$$

گام ۸) فرض کنید تعداد کارهای جدید M باشد (تعداد کارهای جدید با تعداد بازه‌های بیکاری ماشین برابر است) و طول بازه‌ی بیکاری mام را برابر با $l_{[m]}$ درنظر بگیرید. جریمه زودکرد کارهای جدید $(h_{k'})$را به ترتیب غیرصعودی مرتب کنید به‌نحوی که $h_{[1]} \ge h_{[2]} \ge \ldots \ge h_{[M-1]} \ge h_{[M]}$ 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 مساله‌ی $1|p_j = p|h_j E_j^2 + w_j T_j^2$ به‌صورت $LB = TP_1 + TP_2$ محاسبه می‌شود.

۵. الگوریتم ابتکاری اول (HA1)

در این روش ابتکاری از یک الگوریتم حریصانه برای یافتن جواب‌های مناسب و قابل قبول برای مساله‌ی مورد بررسی استفاده می‌شود. قبل از بیان گام‌های روش لازم است تا به این نکته اشاره شود که اگرچه این الگوریتم برای مساله‌ی $1|p_j = p|h_j E_j^2 + w_j T_j^2$ ارائه شده است ولی از آن برای یافتن توالی‌های ابتکاری برای مساله‌ی $1|h_j E_j^2 + w_j T_j^2$ نیز می‌توان استفاده کرد.

گام ۱) تمام کارها را به‌ترتیب موعدهای تحویلشان شماره‌گذاری کنید به‌نحوی که

$$d_1 \le d_2 \le \ldots \le d_{N-1} \le d_N$$

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 iام را انتخاب کرده و آنرا در i مکان امکان‌پذیر در توالی قرار دهید. مجموع جریمه‌های زودکرد و دیرکرد کارها را در هر مرتبه محاسبه کرده و کار i را در مکانی که کوچکترین جریمه را ایجاد می‌کند زمان‌بندی نمایید.

گام ۴) اگر $i = N$ بود توالی نهایی به‌دست 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 گام‌های الگوریتم پرداخته شود لم‌های به‌کار رفته در این روش ابتکاری تشریح می‌شوند.

لم۱ : در مساله $1||h_j E_j^2 + w_j T_j^2$ اگر تعدادی کار دلخواه بهم چسبیده باشند و لحظه‌ی شروع اولین کار t باشد، درصورتی‌که مجموعه‌ی کارهای زودکرددار در این حالت را با $G_E(t)$ و

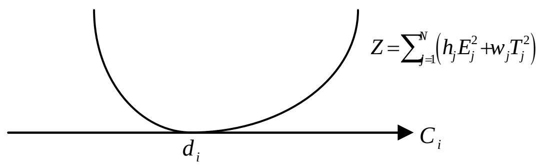
مجموعه‌ی کارهای دیرکرددار را با $G_T(t)$ نمایش دهیم، آنگاه تابع $F(t)=\sum_{i\in G_E(t)}\left(h_iE_i(t)\right)-\sum_{j\in G_T(t)}\left(w_jT_j(t)\right)$ تابعی پیوسته و نزولی بوده و به‌ازای یک مقدار حقیقی $t=t^*$ رابطه‌ی $F(t^*)=0$ برقرار خواهد بود.

برای اثبات این لم می‌توان نشان داد که با کم شدن t به سمت منفی بینهایت، تعداد کارهای زودکرددار افزایش می‌یابد و درنهایت رابطه‌ی $F(t)$ مقداری مثبت می‌گیرد و برعکس با زیاد شدن t به‌سمت مثبت بینهایت، تعداد کارهای دیرکرددار افزایش یافته و رابطه‌ی $F(t)$ منفی خواهد شد. اکنون با توجه به پیوسته بودن $F(t)$ می‌توان نتیجه گرفت که

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

لم ۲: برای مساله‌ی $1|h_jE_j^2+w_jT_j^2$ اگر تعدادی کار را به دلخواه کنار هم چیده شده باشند و بعضی زودکرددار و بعضی دیرکرددار 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 می‌توان نشان داد که نمودار هزینه‌ی کل برای هر ترتیب دلخواه از کارهای چیده شده کنار هم نموداری پیوسته است. اگر مجموع جریمه زودکرد و دیرکرد کارها به‌صورت $Z=\sum_{j=1}^{N}\left(h_jE_j^2+w_jT_j^2\right)$ نشان داده شود، می‌توان مقدار Z را برابر با مجموع جریمه‌های تک‌تک کارها درنظر گرفت. مشخص است که نمودار جریمه زودکرد و دیرکرد یک کار نسبت به زمان تکمیل آن کار، یک نمودار محدب و پیوسته است (مطابق شکل۲) 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 زمان‌های تکمیل

در مساله‌ی $1||h_jE_j^2+w_jT_j^2$ حالتی را درنظر بگیرید که تعدادی از کارها بهم چسبیده باشند و زمان شروع اولین کار t باشد. اگر مجموعه‌ی کارهای زودکرددار در این شرایط با $G_E(t)$ و مجموعه کارهای دیرکرددار با $G_T(t)$ نمایش داده شوند، در لم۱ نشان داده شد که تابعی $F(t)=\sum_{i\in G_E(t)}\left(h_iE_i(t)\right)-\sum_{j\in G_T(t)}\left(w_jT_j(t)\right)$ پیوسته و نزولی است و همچنین به‌ازای مقداری مانند $t=t^*$ عبارت $F\left(t^*\right)=0$ خواهد بود. همچنین اثبات می‌شود که تابع $Z(t)$ در سمت راست نقطه‌ی $t=t^*$ صعودی و در سمت چپ این نقطه نزولی است و بنابراین می‌توان نتیجه گرفت که تابع مذکور محدب بوده و دارای نقطه‌ی مینیمم برابر با $t=t^*$ می‌باشد.

نتیجه لم ۲: نقاط شکست جریمه برای مساله‌ی $1|h_jE_j^2+w_jT_j^2$

در لم۲ مشاهده شد که برای مساله‌ی $1|h_jE_j^2+w_jT_j^2$ 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 اندازه‌ی m عدد کار وجود داشته باشد، حداکثر تعداد نقاط شکست برای تابع جریمه برابر با m^2 خواهد بود.

ابتدا به این نکته اشاره می‌شود که اگرچه این الگوریتم برای مساله‌ی $1|p_j=p|h_jE_j^2+w_jT_j^2$ ارائه شده است ولی از آن برای یافتن توالی‌های ابتکاری برای مساله‌ی $1|h_jE_j^2+w_jT_j^2$ نیز می‌توان استفاده کرد. این الگوریتم ابتکاری در دو مرحله اجرا می‌شود که در مرحله اول (گام‌های ۱ تا ۷) سعی شده تا با کمترین میزان جریمه تداخل میان کارها حذف شده و در مرحله دوم (گام‌های ۸ تا۱۳) بازه‌های بیکاری ماشین برداشته شود تا توالی حاصل هم امکان‌پذیر بوده و هم اینکه با فرض عدم بیکاری ماشین در تناقض نباشد.

گام ۱) تمام کارها را در موعدهای 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 درنظر بگیرید. فرض کنید تعداد کارهای موجود در گروه G برابر با nG باشد. قرار دهید $m=nG+1$

گام ۴) کار j را در mامین مکان بین کارهای مجموعه G قرار داده و کارهای قبلی موجود در مکان‌های m تا $nG+1$ از مجموعه G را به‌اندازه‌ی 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

جریمه (طبق لم۲) 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 نقطه‌ی 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 $m \ge 1$ باشد قرار دهید $m = m - 1$ و به گام ۴ برگردید وگرنه به گام ۷ بروید.

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

گام۸) کارها را با همان ترتیب حاصل از مرحله اول درنظر گرفته و آنها را به‌نحوی دسته‌بدی کنید که کارهای چسبیده بهم در یک دسته قرار گیرند. دسته‌ی kام از کارها را به‌عنوان یک کار جدید k' تعریف کرده و مشخصات آنرا از روابط زیر محاسبه نمایید.

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

$$h_{k'} = \underset{j \in K}{ave} \left\{ h_j \right\} \qquad w_{k'} = \underset{j \in K}{ave} \left\{ w_j \right\}$$

گام۹) اولین کار موجود در توالی را در زمان صفر (لحظه‌ی شروع) زمان‌بندی کنید.

گام۱۰) اگر $j \le 1$ بود به گام ۱۳ و در غیر این صورت به گام ۱۱ بروید.

گام۱۱) کار موجود در مکان jام توالی $\left(J_{[j]} \right)$ و کار موجود در مکان $j-1$ام توالی $\left(J_{[j-1]} \right)$ را انتخاب کنید. اگر $L_{j-1} > p_{J_{[j]}}$ و نیز $\frac{h_{J_{[j]}} p_{J_{[j-1]}}}{h_{J_{[j-1]}} p_{J_{[j]}}} \le \frac{2L_{j-1} - p_{J_{[j]}}}{2L_j + p_{J_{[j-1]}}}$ بود و با اینکه اگر $L_{j-1} \le p_{J_{[j]}}$و $\frac{h_{J_{[j]}} p_{J_{[j-1]}}}{h_{J_{[j-1]}} p_{J_{[j]}}} \le \frac{\left(L_{j-1} \right)^2}{2L_j + p_{J_{[j-1]}}}$ برقرار بود جای دو کار را در توالی باهم تعویض کنید.

گام۱۲) قرار دهید $j = j - 1$ و مجدداً به گام ۱۱ برگردید.

گام۱۳) اگر i برابر با تعداد کارها بود، آنگاه کارها را براساس ترتیب به‌دست آمده کنار هم بچینید (پایان الگوریتم) و در غیر این صورت ابتدا قرار دهید $i = i + 1$ و سپس $j = i$ را درنظر 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 استفاده شود. نتایج محاسباتی نشان می‌دهد 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 توالی‌های اولیه مربوط به مساله‌ی $1|h_j E_j^2 + w_j T_j^2$ نیز می‌توان استفاده کرد. در الگوریتم شاخه و کران پیشنهادی درصورتی‌که روش بهبوددهنده برروی الگوریتم ابتکاری اول (HA1) پیاده شود به این مجموعه IHA[1] و درصورتی که برروی الگوریتم HA2 پیاده‌سازی شود به ان IHA2 اطلاق می‌شود. گام‌های این الگوریتم ابتکاری به‌صورت 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) نزدیک‌ترین کار دیرکردار قبل از کار i را کار j بنامید. مجموعه کارهای iو j و کارهای بین آنها را مجموعه G بنامید. در صورتی‌که 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) نزدیک‌ترین کار زودکردار قبل از مجموعه G را کار kبنامید. جای کار k را با تک‌تک کارهای دیرکردار مجموعه G تعویض کرده و حالتی‌که بیشترین کاهش جریمه را ایجاد می‌کند را انتخاب نمایید.

گام ۵) درصورتی 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) نزدیک‌ترین کار زودکردار بعد از کار i را کار j بنامید. مجموعه کارهای iو j و کارهای بین آنها را مجموعه G بنامید. درصورتی 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) نزدیک‌ترین کار دیرکردار بعد از مجموعه G را

[1] - *Improved Heuristic Algorithm 1*

کار k بنامید. جای کار k را با تک‌تک کارهای زودکرددار مجموعه G تعویض کرده و حالتی‌که بیشترین کاهش جریمه را ایجاد می‌کند را انتخاب نمایید.

گام ۱۰) درصورتی که بعد از کار i کار دیرکرددار 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 به‌منظور سنجش میزان کارایی الگوریتم‌ها و روش‌های ارائه شده ابتدا تعدادی مساله‌ی نمونه به‌صورت تصادفی تولید شده و سپس روش‌های پیشنهادی برروی آنها پیاده‌سازی و اجرا می‌شوند.

برای تولید مسائل تصادفی می‌بایست مقادیر چهار پارامتر جریمه زودکرد، جریمه دیرکرد، زمان‌های پردازش و موعدهای تحویل تعیین شود. در مسائل تولید شده جریمه‌های زودکرد، دیرکرد و زمان‌های پردازش یک مرتبه از توزیع یکنواخت گسسته در بازه‌ی [45,55] و بار دیگر از بازه‌ی [1,100] تولید شده‌اند[۱۸]. دسته اول میزان پراکندگی کم و دسته دوم پراکندگی زیاد را برای جریمه‌ها و زمان‌های پردازش نشان می‌دهد و با توجه به آزمایش‌های انجام گرفته برروی مسائل نمونه مشخص شده است که مسائل با پراکندگی کمتر، ساده‌تر بوده و در زمان کمتری نسبت به مسائل با پراکندگی بالا حل می‌شوند. برای تولید موعدهای تحویل کارها از دو پارامتر عامل دیرکرد (τ) و عامل دامنه موعد تحویل (R) استفاده می‌شود. اگر متوسط موعد تحویل کارها را با $\bar{d}$ و متوسط زمان‌های پردازش را با $\bar{P}$ نمایش دهیم، موعدهای تحویل از روابط $\bar{d}=(1-\tau)n\bar{P}$ و $d_i \in U\left[\bar{d}(۱-\frac{R}{۲}),\bar{d}(۱+\frac{R}{۲})\right]$ به‌دست خواهد آمد.

از آنجایی‌که مشاهدات اولیه نشان می‌دهد تغییر پارامترهای τ و R تاثیر چندانی در میزان دشواری مسائل نمونه و زمان‌های اجرای الگوریتم شاخه و کران ندارد، برای این دو پارامتر دسته‌بندی انجام نشده و مقادیر آنها به‌صورت $\tau=۰/۴$ و $\mathrm{R}=۰/۸$ انتخاب می‌شود. تعداد مسائل نمونه ایجاد شده در هر دسته ۱۰ عدد می‌باشد.

روش شاخه و کران مبتنی بر جستجوی عمقی با استفاده از حدود بالا و پایین و نیز اصول غلبه مطرح شده در بخش‌های قبلی در محیط نرم‌افزار Visual C++ برنامه‌نویسی و پیاده‌سازی شده است. برای حل هر مساله در روش شاخه و کران محدودیت زمانی ۳۶۰۰ ثانیه درنظر گرفته شده است و اگر مساله‌ای تا این زمان با روش شاخه و کران به جواب نرسد، رویه حل شاخه و کران برای آن مساله متوقف می‌شود. نتایج حل دسته‌های گوناگون در جدول ۲ نمایش داده شده است.

جدول ۲. نتایج حل مسائل در دسته‌های مختلف

		روش ابتکاری IHA1		روش ابتکاری IHA2		روش شاخه و کران				
پراکندگی	تعداد کار	تعداد نمونه‌های بهینه	درصد اختلاف با جواب بهینه	تعداد نمونه‌های بهینه	درصد اختلاف با جواب بهینه	تعداد نمونه‌های بهینه	میانگین زمان حل (ثانیه)	میانگین درصد گره‌های قطع شده	میانگین درصد گره‌های قطع شده به دلیل حد پایین	میانگین درصد گره‌های قطع شده به دلیل اصول غلبه
کم	۱۰	۹	۰/۲۸	۹	۰/۲۷	۱۰	٦/۱۸	۹٤/۱۱	٦۲/۱۹	۳۷/۸۱
	۱۵	۸	۱/۱۳	۹	۰/۳۹	۱۰	۳٦٦/۱۱	۹۱/۲۸	۷٤/۷۵	۲۵/۲۵
	۲۰	٦	۲/۵۷	۸	۰/۹۵	۱۰	٦۳۱/۰۵	۹۵/۸۱	٦۷/۳۷	۳۲/٦۳
	۲۵	٦	۲/۷۳	۷	۱/۸۲	۱۰	۱۱۰۵/٤٤	۸۹/۳٦	۵۸/۹۳	٤۱/۰٦
	۳۰	۵	۳/۸٦	٦	۲/۸۸	۱۰	۱۹۳٤/۸۲	۹۳/۳٦	٦٦/۰۸	۳۳/۹۱
	۳۵	٤	۵/۰٦	٦	۳/٤۹	۱۰	۲۸۱٦/۳۷	۹۷/۰٤	۷۵/۸۲	۲٤/۱۷
زیاد	۱۰	۸	۱/٤۲	۹	۰/٤٦	۱۰	۸/٤۷	۸۹/۷۷	٦۱/۲۹	۳۸/۷۲
	۱۵	۵	۳/٦۱	۷	۱/۱۳	۱۰	٤۲۷/۳۷	۹۱/٦۱	۵۸/۱۱	٤۱/۹۰
	۲۰	٤	٤/۷۱	۵	۳/۹۲	۱۰	۹۱٦/۸۸	۸۵/۱۳	٦۷/۲٤	۳۲/۷٦
	۲۵	۲	۷/٤۸	۳	٦/۰۷	۱۰	۱۸٦۰/۳۹	۸۹/۷٦	٦۱/۸۸	۳۸/۱۳
	۳۰	۱	۱۱/۹۲	۲	۹/۳٦	۹	۲٤۳۵/۱۷	۸٦/۷۲	۷٤/۹۰	۲۵/۰۹
	۳۵	۱	۱۲/۲٤	۲	۱۰/۹۲	۵	۳٤۳۱/۷۲	۸۱/۹۲	۷۹/۳۹	۲۰/٦۱

در جدول ۲، ستون‌های «تعداد نمونه بهینه» نشان‌دهنده‌ی تعداد مسائلی از ۱۰ مساله می‌باشد که هرکدام از روش‌های ابتکاری و شاخه و کران به‌جواب 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 تقریبا 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 کاراتر 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 پشنهادی نشان می‌دهند. همچنین روش شاخه و کران نیز در ادامه برروی 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*

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