



# حل مساله زمانبندی تک ماشینه با درنظر گرفتن جریمه‌های زودکرد و دیرکرد مرتبه دوم

\* کامران کیانفر و قاسم مصلحی

## چکیده:

امروزه مسائل زمانبندی در بسیاری از سیستم‌های تولیدی و خدماتی کاربرد وسیعی یافته‌اند. در این مقاله مساله‌ی زمانبندی  $n$  کار بر روی تک ماشین 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 تحويل دادن و یا هزینه‌های نگهداری، جریمه‌های زودکرد و دیرکرد کارها به صورت ترکیبات خطی در تابع هدف ظاهر می‌شود در صورتی که تابع هدف این مقاله کمینه کردن مجموع وزن دار 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 و کران 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.

## کلمات کلیدی

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

## ۱. مقدمه

مسائل تک ماشینه از جمله مسائل متداول در حیطه مسائل زمانبندی بهشمار رفته که در بسیاری از سیستم‌های تولیدی عملی کاربرد دارند. به علاوه 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 به صورت همزمان را می‌توان با فلسفه تولید بهنگام (JIT) مقایسه

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 moslehi@cc.iut.ac.ir

لحاظ شده است. وی یک فرایند جدول‌بندی زمانی<sup>۳</sup> 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؛ درحالی که سو و چانگ [۱۳] و اسکالار [۱۴] همین مساله را در حالتی که 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 باتابع 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 معروفی 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 می‌باشد 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 هم‌جاور در یک توالی با توجه به زودکردار یا دیرکردار بودن دو کار قبل و بعد از جایجایی، 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  $\alpha$  و  $\beta$  بدون درنظر گرفتن ترتیب آنها در توالی زودکردار باشند. برای این حالت اگر  $F_{ij} = F_{ji}$  به ترتیب جریمه‌های زودکرد کارها قبل و بعد از جایجایی باشند، اصل غلبه مربوطه را می‌توان به صورت زیر اثبات نمود. در این رابطه  $t$  و  $p$  به ترتیب زمان شروع دو کار هم‌جاور و زمان پردازش کارها می‌باشند.

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

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

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

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

<sup>۲</sup> - time-tabling

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 به صورت پیوسته کار کند. همچنین در زمانی که هزینه‌های توقف و راهاندازی مجدد ماشین آلات نسبت به جریمه‌های زودکرد و دیرکرد تولید بزرگ‌تر باشد

از دیگر موارد به کار بردن فرض عدم بیکاری برای ماشین‌ها به شمار می‌رود. با توجه به فرضیات بیان شده می‌توان مساله را با نماد  $\sum h_j E_j^2 + w T_j = p | \sum h_j E_j^2 + w T_j = p$  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 به صورت غیرخطی و وزن دار درنظر می‌گیرد نیز مساله‌ای NP-Hard می‌باشد.

### ۲. مرور ادبیات موضوع

مساله مشابه با فرض عدم بیکاری ماشین و جریمه‌های زودکرد و دیرکرد وزن دار و خطی تاکنون توسط محققین مختلفی مطالعه شده و روش‌های دقیق و ابتکاری متعددی نیز برای آن ارائه شده است. از میان روش‌های دقیق می‌توان به الگوریتم‌های شاخه و کرانی که توسط عبدالرازاق و پاتنس [۱]، لی [۲] و لیانو [۳] و والته 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<sup>۱</sup> معرفی شده توسط او و مورتن [۵] و والته 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 راداکریشنان [۹] مطرح شده است که در تحقیق اول، یک الگوریتم شاخه و کران و در تحقیق دوم یک الگوریتم آزادسازی لاگرانژی برای مساله مذکور پیشنهاد شده است. همچنین مساله فوق در حالت وجود موعدهای تحويل یکسان برای تمام کارها توسط ساندراهاهن و احمد [۱۰] تحلیل شده است.

مسئلی که توابع هدف غیرخطی دارند نیز در مطالعات گوناگون بررسی شده است که از جمله می‌توان به تحقیق اسکالار [۱۱] اشاره کرد که در آن مقاله مساله‌ی تک‌ماشین با فرض مجاز بودن بیکاری ماشین و جریمه‌های زودکرد خطی و دیرکرد درجه دوم

<sup>۱</sup> - beam search

برای حالتی به کار می‌رود 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 مساله‌ی  $1/p_j = p / h_j E_j^2 + w_j T_j^2$  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 تنافض 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  $6$  حالت خاص نسبت به سایر حالات جریمه کمتری دارند که حداقل جریمه ناشی از این حالات در گام‌های  $2$  و  $3$  الگوریتم محاسبه شده است.

برای حذف فاصله‌های بیکاری مشین و محاسبه 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  $n$  موجود در توالی با  $J_{[n]}$  نمایش داده شود، آنگاه قبل از کار  $n$  یک بازه‌ی بیکاری ماشین به طول  $0 \geq l_{[n]}$  وجود خواهد داشت. (مطابق شکل  $(1)$ )

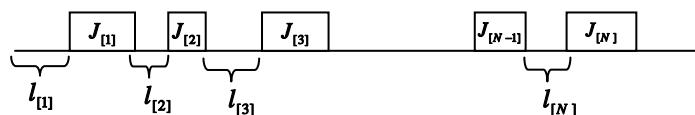
جدول ۱. اصول غلبه مربوط به جفت کارهای مجاور در

$$\text{مساله‌ی } 1/p_j = p / h_j T_j^2 + w_j T_j^2$$

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

سایر حالات موجه به همراه اصول غلبه مربوطه در جدول  $(1)$  خلاصه شده است.

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



شکل ۱. وضعیت قرار گرفتن کارها و بازه‌های بیکاری

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 به صورت زیر نیز نوشت.

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

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

**گام ۸)** فرض کنید تعداد کارهای جدید  $M$  باشد (تعداد کارهای جدید با تعداد بازه‌های بیکاری ماشین برابر است) و طول بازه‌ی بیکاری  $m$  را برابر با  $I_{[m]}$  درنظر بگیرید. جریمه زودکرد کارهای جدید ( $h_k$ ) را به ترتیب غیرصعودی مرتب کنید بهنحوی که  $h_{[1]} \geq h_{[2]} \geq \dots \geq h_{[M-1]} \geq h_{[M]}$  باشد. آنگاه خواهیم داشت:

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

**گام ۹)** حد پایین مساله‌ی  $p_j = p | h_j E_j^2 + w_j T_j^2 |$  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 قبل 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  $p_j = p | h_j E_j^2 + w_j T_j^2 |$  ارائه شده است ولی از آن برای یافتن توالی‌های ابتکاری برای مساله‌ی  $Z = \min \{h_j E_j^2 + w_j T_j^2\}$  نیز می‌توان استفاده کرد.

**گام ۱)** تمام کارها را به ترتیب موعدهای تحویلشان شماره‌گذاری کنید بهنحوی 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  $i$  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 به دست آمده است (پایان الگوریتم) و در غیر این صورت قرار دهید  $i+1 = i$  و به گام سوم برگردید.

### ۶. الگوریتم ابتکاری دوم (HA2)

قبل از اینکه به تعریف گام‌های الگوریتم پرداخته شود لم‌های به کار رفته در این روش ابتکاری تشریح می‌شوند.

**لم ۱:** در مساله  $Z = \min \{h_j E_j^2 + w_j T_j^2\}$  اگر تعدادی کار دلخواه بهم چسبیده باشند و لحظه‌ی شروع اولین کار  $t$  باشد، درصورتی که مجموعه کارهای زودکردار در این حالت را با  $(t, G_E(t))$  و

$$TP = \left[ h_{J_{[1]}}, h_{J_{[2]}}, \dots, h_{J_{[N]}} \right]. \quad (2)$$

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

با توجه به این نکته که مقادیر درایه‌های بردار دوم افزایشی است، بنابراین برای حداقل شدن  $TP$  می‌بایست درایه‌های بردار اول به صورت غیرصعودی چیده شده و سپس جریمه زودکرد محاسبه گردد. گام‌های الگوریتم مربوط به این حد پایین به صورت 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.

**گام ۱)** تمام کارها را در موعد تحویل خود زمان‌بندی 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  $Z^{(1)} = \frac{h_i}{w_i}$

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

اگر  $Z^{(2)} > 1 + \frac{2p}{f}$  بود  $Z_2 = Z_4$  و در غیر این صورت اگر  $Z^{(2)} > 1$  بود  $Z_2 = Z_5$  و در غیر این صورت اگر  $Z^{(2)} > \left(1 + \frac{2p}{f}\right)^{-1}$  بود  $Z_2 = Z_3$  است.

**گام ۴)** قرار دهید  $\{F_i, F_2\} = \min \{F_i, F_2\}$  که نشان‌دهنده 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 متداخلي در توالی باقی نماند. مجموع جریمه‌های محاسبه شده در گام‌های فوق را  $TP_1$  بنامید.

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

**گام ۷)** کارها را دسته‌بندی کنید بهنحوی که کارهای متداخلي و بهم چسبیده در یک دسته قرار گیرند. کارهای موجود در دسته‌ی  $K$  را به عنوان یک کار جدید  $k'$  درنظر گرفته و مشخصات کار جدید را از روابط زیر محاسبه نمایید.

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

$$h_{k'} = \min_{j \in K} \{h_j\} \quad w_{k'} = \min_{j \in K} \{w_j\}$$

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

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

در لم ۲ مشاهده شد که برای مساله‌ی  $1|h_j E_j^2 + w_j T_j^2|$  اگر تعدادی کار دلخواه بهم چسبیده (مثالاً مجموعه کارهای  $G$ ) به سمت چپ و راست توالی جابجا شوند، نمودار هزینه زودگرد و دیرگرد 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 موعود تحویل یکی از کارهای این مجموعه بر هم منطبق شوند. یعنی اگر در مجموعه  $G$  به اندازه‌ی  $m$  عدد کار وجود داشته باشد، حداقل تعداد نقاط شکست برای تابع جریمه برابر با  $m^2$  خواهد بود.

ابتدا به این نکته اشاره می‌شود که اگرچه این الگوریتم برای مساله‌ی  $1|h_j E_j^2 + w_j T_j^2|$  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. 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 تا ۱۳) بازه‌های بیکاری ماشین برداشته شود تا توالی حاصل هم امکان‌پذیر بوده و هم اینکه با فرض عدم بیکاری ماشین در تنافض نباشد.

**گام ۱)** تمام کارها را در موعدهای تحویلشان قرار دهید.

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

**گام ۳)** فرض کنید کار  $i$  به گروهی از کارها (مثالاً  $G$  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  $1||h_j E_j^2 + w_j T_j^2|$  حالتی را درنظر بگیرید که تعدادی از کارها بهم چسبیده باشند و زمان شروع اولین کار  $t$  باشد. اگر مجموعه‌ی کارهای زودگرد دار در این شرایط با  $G_E(t)$  و مجموعه کارهای دیرگرد دار با  $G_T(t)$  نمایش داده شوند، در لم ۱ نشان داده شد که

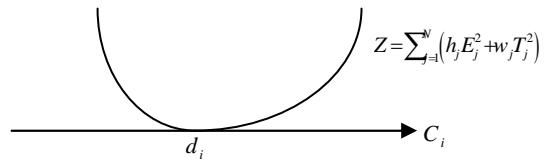
مجموعه‌ی کارهای دیرگرد دار را با  $(t) G_T$  نمایش دهیم، آنگاه  $F(t) = \sum_{i \in G_E(t)} (h_i E_i(t)) - \sum_{j \in G_T(t)} (w_j T_j(t))$  تابعی پیوسته و نزولی بوده و به ازای یک مقدار حقیقی  $t = t^*$  رابطه‌ی  $F(t^*) = 0$  برقرار خواهد بود.

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

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

**لم ۲:** برای مساله‌ی  $1|h_j E_j^2 + w_j T_j^2|$  اگر تعدادی کار را به دلخواه کنار هم چیده شده باشند و بعضی زودگرد دار و بعضی دیرگرد دار باشند، آنگاه اگر مجموعه کارها را بدون جابجا کردن ترتیب کارها به سمت چپ یا راست منتقل کیم، نمودار هزینه دیرگرد و زودگرد 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 زودگرد و دیرگرد کارها به صورت  $Z = \sum_{j=1}^N (h_j E_j^2 + w_j T_j^2)$  نشان داده شود، می‌توان مقدار  $Z$  را برابر با مجموع جریمه‌های تک‌تک کارها درنظر گرفت. مشخص است که نمودار جریمه زودگرد و دیرگرد 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 تعريف شده است، خود نیز تابعی پیوسته خواهد بود.



شکل ۲. نمودار جریمه زودگرد و دیرگرد درجه دوم کارها براساس زمان‌های تکمیل

در مساله‌ی  $1||h_j E_j^2 + w_j T_j^2|$  حالتی را درنظر بگیرید که تعدادی از کارها بهم چسبیده باشند و زمان شروع اولین کار  $t$  باشد. اگر مجموعه‌ی کارهای زودگرد دار در این شرایط با  $G_E(t)$  و مجموعه کارهای دیرگرد دار 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 می‌شود که تابع  $Z(t)$  در سمت راست نقطه‌ی  $t = t^*$  صعودی و در سمت چپ

مجموعه‌ی کارهای  $G$  را تا رسیدن به یکی از نقاط شکست  $Z_G$  نشان دهد. چهارمین مکان بین کارهای مجموعه  $G$  قرار داده و کارهای قبلی موجود در مکان‌های  $m$  تا  $nG+1$  از  $mG$  را بهاندازه‌ی  $p_j$  (زمان برداش کار  $j$ ) به سمت راست جابجا کنید تا داخلی میان کارهای مجموعه  $G$  نباشد. جریمه زودگرد و دیرگرد 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

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 مساله<sup>۱</sup>  $T_j^2 + w_j E_j^2$  با  $1 \leq m \leq n$  می‌توان استفاده کرد. در الگوریتم شاخه و کران پیشنهادی در صورتی که روش بهبوددهنده برروی الگوریتم ابتکاری اول (HA1) پیاده شود به این مجموعه IHA<sup>۱</sup> و در صورتی که برروی الگوریتم HA2 پیاده‌سازی شود به ان IHA2 اطلاق می‌شود. گام‌های این الگوریتم ابتکاری به صورت زیر است.

**گام ۱)** از انتهای توالی اولیه شروع کرده و به اولین کار دیرکردداری 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 دیرکرددار 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)** نزدیکترین کار زودکرددار قبل از مجموعه  $G$  را کار  $k$  بنامید. جای کار  $k$  را با تک‌تک کارهای دیرکرددار مجموعه  $G$  تعویض کرده و حالتی که بیشترین کاهش جریمه را ایجاد می‌کند را انتخاب نمایید.

**گام ۵)** در صورتی که قبل از کار  $i$  کار دیرکرددار 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 زودکردداری 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 زودکرددار 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)** نزدیکترین کار دیرکرددار بعد از مجموعه  $G$  را

جریمه (طبق لم ۲) به سمت چپ جابجا کنید. اگر جریمه کل کم شد این کار را 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 ۱۰)** اگر  $m \geq 1$  باشد قرار دهید  $m = m - 1$  و به گام ۴ برگردید و گرنه به گام ۷ بروید.

**گام ۱۱)** فرض کنید  $Z_g^{m'} = \min_m \{Z_g^m\}$  باشد. کار  $j$  را در مکان  $m'$  ام بین کارهای مجموعه  $G$  زمان‌بندی کنید. کار  $j$  اکنون جزء مجموعه  $G$  است. در صورتی که هچ دو کاری تداخل نداشته باشند به مرحله دوم از الگوریتم (گام ۸) رفته و در غیر این صورت به گام ۲ برگردید.

**گام ۱۲)** کارها را با همان ترتیب حاصل از مرحله اول درنظر گرفته و آنها را به نحوی دسته‌بندی کنید که کارهای چسبیده بهم در یک دسته قرار گیرند. دسته‌ی  $k$  ام از کارها را به عنوان یک کار جدید  $k$  تعریف کرده و مشخصات آنرا از روابط زیر محاسبه نمایید.

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

$$h_{k'} = ave_{j \in K} \{h_j\} \quad w_{k'} = ave_{j \in K} \{w_j\}$$

**گام ۱۳)** اولین کار موجود در توالی را در زمان صفر (لحظه‌ی شروع زمان‌بندی) کنید.

**گام ۱۴)** اگر  $1 \leq j \leq n$  بود به گام ۱۳ و در غیر این صورت به گام ۱۱ بروید.

**گام ۱۵)** کار موجود در مکان  $j$  ام توالی  $(J_{[j]})$  و کار موجود در مکان  $1-j$  ام توالی  $(J_{[j-1]})$  را انتخاب کنید. اگر  $p_{J_{[j]}} > p_{J_{[j-1]}}$  و  $L_{j-1} \leq p_{J_{[j]}}$  بود و با اینکه اگر  $\frac{h_{J_{[j]}} p_{J_{[j-1]}}}{h_{J_{[j-1]}} p_{J_{[j]}}} \leq \frac{2L_{j-1} - p_{J_{[j]}}}{2L_j + p_{J_{[j-1]}}}$  نیز  $L_{j-1} \leq p_{J_{[j-1]}}$  بود جای دو کار را در توالی  $\frac{h_{J_{[j]}} p_{J_{[j-1]}}}{h_{J_{[j-1]}} p_{J_{[j]}}} \leq \frac{(L_{j-1})^2}{2L_j + p_{J_{[j-1]}}}$  برقرار بود باهم تعویض کنید.

**گام ۱۶)** قرار دهید  $1-j = j$  و مجدداً به گام ۱۱ برگردید.

**گام ۱۷)** 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  $i+1 = i$  و سپس  $i = j$  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

<sup>۱</sup> - Improved Heuristic Algorithm I

نسبت به مسائل با پراکندگی بالا حل می‌شوند. برای تولید موعدهای تحويل کارها از دو پارامتر عامل دیرکرد ( $\tau$ ) و عامل دامنه موعد تحويل ( $R$ ) استفاده می‌شود. اگر متوسط موعد تحويل کارها را با  $\bar{d}$  و متوسط زمان‌های پردازش را با  $\bar{P}$  نمایش دهیم، موعدهای تحويل از روابط  $\bar{d} = (1 - \tau)R$  و  $\bar{P} = (1 - \tau)n\bar{R}$  باشند.

از آنجایی که مشاهدات اولیه نشان می‌دهد تغییر پارامترهای  $\tau$  و  $R$  تأثیر چندانی در میزان دشواری مسائل نمونه و زمان‌های اجرای الگوریتم شاخه و کران ندارد، برای این دو پارامتر دسته‌بندی انجام نشده و مقادیر آنها به صورت  $\tau = 0/4$  و  $R = 0/8$  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 روشن شاخه و کران محدودیت زمانی ۳۶۰۰ ثانیه درنظر گرفته شده است و اگر مسائلهای تا این زمان با روشن 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.

کار  $k$  بنامید. جای کار  $k$  را با تک‌تک کارهای زودکردادار مجموعه  $G$  تعویض کرده و حالتی که بیشترین کاهش جریمه را ایجاد می‌کند را انتخاب نمایید.

گام (۱۰) درصورتی که بعد از کار  $i$  کار دیرکردادار دیگری وجود داشته باشد، آنرا کار  $i$  بنامید و به گام ۷ برگردد و در غیر این صورت الگوریتم به پایان رسیده است.

## ۸. نتایج محاسباتی

در این بخش به منظور سنجش میزان کارایی الگوریتم‌ها و روش‌های ارائه شده ابتدا تعدادی مساله‌ی نمونه به صورت 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 تحويل تعیین شود. در مسائل تولید شده جریمه‌های زودکرد، دیرکرد و زمان‌های پردازش یک مرتبه از توزیع یکنواخت گستته در بازه‌ی [۴۵,۵۵] و بار دیگر از بازه‌ی [۱,۱۰۰] 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

جدول ۲. نتایج حل مسائل در دسته‌های مختلف

روش شاخه و کران				IHA2				IHA1				روش ابتکاری			
میانگین میانگین درصد گوههای قطع شده	میانگین درصد قطع شده به دلیل حد پایین اصول غلبه	میانگین زمان حل (ثانیه)	تعداد نمونه‌های قطع شده	تعداد	درصد اختلاف با نمونه‌های بیهینه	درصد اختلاف با نمونه‌های بیهینه	درصد اختلاف با نمونه‌های بیهینه	تعداد	درصد اختلاف با نمونه‌های بیهینه	درصد اختلاف با نمونه‌های بیهینه	تعداد	پراکندگی	تعداد کار	زیاد	
۳۷/۸۱	۶۲/۱۹	۹۴/۱۱	۶/۱۸	۱۰	۰/۲۷	۹	۰/۲۸	۹	۰/۲۸	۹	۱۰				
۲۵/۲۰	۷۴/۷۵	۹۱/۲۸	۳۶۶/۱۱	۱۰	۰/۳۹	۹	۱/۱۳	۸	۱/۱۳	۸	۱۵				
۳۲/۶۳	۷۷/۳۷	۹۵/۸۱	۶۳۱/۰۵	۱۰	۰/۹۵	۸	۲/۵۷	۶	۲/۵۷	۶	۲۰				
۴۱/۰۶	۵۸/۹۳	۸۹/۳۶	۱۱۰/۴۴	۱۰	۱/۸۲	۷	۲/۷۳	۶	۲/۷۳	۶	۲۵				
۳۳/۹۱	۶۶/۰۸	۹۳/۳۶	۱۹۳۴/۸۲	۱۰	۲/۸۸	۶	۳/۸۶	۵	۳/۸۶	۵	۳۰				
۲۴/۱۷	۷۵/۸۲	۹۷/۰۴	۲۸۱۶/۳۷	۱۰	۳/۴۹	۶	۵/۰۶	۴	۵/۰۶	۴	۳۵				
۳۸/۷۲	۶۱/۲۹	۸۹/۷۷	۸/۴۷	۱۰	۰/۴۶	۹	۱/۴۲	۸	۱/۴۲	۸	۱۰				
۴۱/۹۰	۵۸/۱۱	۹۱/۶۱	۴۲۷/۳۷	۱۰	۱/۱۳	۷	۳/۶۱	۵	۳/۶۱	۵	۱۵				
۳۲/۷۶	۶۷/۲۴	۸۵/۱۳	۹۱۶/۸۸	۱۰	۳/۹۲	۵	۴/۷۱	۴	۴/۷۱	۴	۲۰				
۳۸/۱۳	۶۱/۸۸	۸۹/۷۶	۱۸۶۰/۳۹	۱۰	۶/۰۷	۳	۷/۴۸	۲	۷/۴۸	۲	۲۵				
۲۵/۰۹	۷۴/۹۰	۸۶/۷۲	۲۴۳۵/۱۷	۹	۹/۳۶	۲	۱۱/۹۲	۱	۱۱/۹۲	۱	۳۰				
۲۰/۶۱	۷۹/۳۹	۸۱/۹۲	۳۴۳۱/۷۲	۵	۱۰/۹۲	۲	۱۲/۲۴	۱	۱۲/۲۴	۱	۳۵				

می‌دهند. همچنین روش شاخه و کران نیز در ادامه بررسی مسائل نمونه پیاده‌سازی گردید که نتایج نشان می‌دهند این روش کارایی حل مسائل تا بعد 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 انقطع 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 هر کدام 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 بعد ۳۵ کار بوده است و برای مسائل با پراکندگی زیاد نیز تقریباً تمام مسائل تا بعد ۳۰ کار حل شده‌اند. با توجه ستون‌های «درصد اختلاف با جواب بهینه» برای IHA2 روش‌های ابتکاری مشخص می‌شود که روش ابتکاری IHA2 اختلاف کمتری با جواب بهینه دارد و نسبت به روش ابتکاری IHA1 کارتر 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 به کار گرفته شده است.

در بخش‌های مختلف مقاله اصول غلبه و حدود بالا و پایین به کار رفته در روش شاخه و کران پیشنهادی تشریح شده است و در بخش نتایج محاسباتی نیز کارایی آنها توسط مسائل نمونه‌ای که به صورت 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