



**هشتمین کنفرانس بین‌المللی مهندسی  
صنایع**

دانشگاه صنعتی امیرکبیر، تهران

۱۳۹۰ و ۲۷ بهمن

**کمینه کردن مجموع وزن دار تعداد واحدهای کاری دارای دیرکرد در زمانبندی کارها بر روی پردازنده های موازی با  
تاخیر ارتباطی**

محمد رنجبر<sup>۱</sup>, مجید سالاری<sup>۲</sup>, فروغ عباسیان<sup>۳</sup>

<sup>۱</sup> استادیار گروه مهندسی صنایع، دانشکده مهندسی، دانشگاه فردوسی مشهد; m\_ranjbar@um.ac.ir

<sup>۲</sup> استادیار گروه مهندسی صنایع، دانشکده مهندسی، دانشگاه فردوسی مشهد; msalari@um.ac.ir

<sup>۳</sup> دانشجوی کارشناسی ارشد گروه مهندسی صنایع، دانشکده مهندسی، دانشگاه فردوسی مشهد; foroogh.abasian@stu-mail.um.ac.ir

**چکیده**

در این مقاله مسئله زمانبندی  $P_m|prec, c_{jk}|Y_W$  مورد بررسی قرار گرفته است که در آن تعدادی کار با روابط پیشنازی مشخص بر روی  $m$  پردازنده موازی مشابه پردازش می شوند. روابط پیشنازی مابین فعالیتها نشان دهنده تبادل اطلاعات است. تبادل اطلاعات بین دو کاری اتفاق می افتد که بطور مستقیم، وابسته بوده و این تبادل متناظر با یک زمان تاخیر ارتباطی است، اگر این دو کار بر روی یک پردازنده پردازش نشوند. در این مقاله مسئله ای که در آن زمان تاخیر ارتباطی مستقل از پردازنده ها بوده و فقط وابسته به کارها می باشد را معرفی می کنیم. همچنین، فرض می کنیم برای هر کار یک زمان تحويل از پیش تعیین شده داده شده است و هدف ما یافتن زمانبندی است که دارای کمترین مجموع وزن دار تعداد واحدهای کاری دارای دیرکرد باشد. ما این مسئله را بصورت یک مدل برنامه ریزی عدد صحیح خطی فرموله کرده، یک الگوریتم شاخه و کران برای آن توسعه داده و روش های ایجاد حدود بالا و پایین را برای مسئله معرفی می کنیم.

**کلمات کلیدی**

پردازنده های موازی، تاخیر ارتباطی، مجموع وزنی واحدهای کاری دارای دیرکرد، شاخه و کران

**Minimizing total weighted late work in scheduling of jobs on identical parallel processors with communication delay**

M. Ranjbar., M. Salari., F. Abasian.

**Abstract**

In this paper, the scheduling problem  $P_m|prec, c_{jk}|Y_W$  has been considered in which a number of jobs with given precedence relations is to be scheduled on a set of identical parallel processors. The precedence relations among jobs represent information exchange. The transfer of information occurs between two jobs that are directly dependent and this is along a constant communication delay time if they are not processed on the same machine. We introduce the problem in which communication delay time is supposed to be independent from processors on which they are being executed while it depends on two corresponding jobs. Also, we assume for each task a predetermined due date is given and we goal to find the schedule with minimum total weighted late works where late work indicate the number of unit works processed after due date. We formulate the problem as a linear integer programming model, develop a branch-and-bound algorithm and introduce strong upper and lower bounding procedures for the studied problem.

**Keywords**

Parallel processors, communication delay, total weighted late work, branch-and-bound

موعد تحويل و وزن آن داده شده است و همهی پارامترها قطعی است. همانطور که قبل ذکر شد، فرض بر این است که زمان های تأخیر ارتباطی بین دو کار وابسته، مستقل از پردازنده هاست و فقط بستگی به آن دو کار دارد. در جدول 1 پارامترهای مساله ذکر شده است.

جدول 1. پارامترها

تعاریف	پارامترها
مجموعه کارها	$J = \{J_1, J_2, \dots, J_n\}$
مجموعه پردازنده ها	$P = \{P_1, P_2, \dots, P_m\}$
مدت زمان پردازش کار $j$	$pt_j$
موعد تحويل کار $j$	$d_j$
وزن کار $j$	$w_j$
مجموعه روابط پیشنایازی شروع به پایان بین کارها	$A$
زمان های تأخیر ارتباطی بین دو کار $(j, k) \in A$	$\Delta_{jk}$

هدف ما یافتن زمانبندی با حداقل مجموع وزنی  $LW$  است. بدون از دست دادن هیچ کلیتی، فرض  $d_n \leq d_1 \leq d_2 \leq \dots \leq d_j$  را در نظر می‌گیریم. برای مدل‌سازی مساله، نیازمند تعریف متغیرهایی هستیم که در جدول 2 آمده‌اند. مقدار  $LW_j$  برای کار  $J_j$  به صورت  $LW_j = \min\{T_j, pt_j\}$  تعریف می‌شود. در این معادله  $T_j = \max\{c_j - d_j, 0\}$  است و  $c_j$  نمایانگر زمان تکمیل کار  $J_j$  است.

جدول 2. متغیرها

تعاریف	متغیرها
میزان $LW$ برای کار $j, j = 1, \dots, n$	$LW_j$
میزان تأخیر کار $j$	$T_j$
متغیر صفوی یک که یک نمایانگر اتمام کار $J_j$ روی پردازنده $i$ در زمان $t$ است، و صفر در غیر اینصورت	$X_{jit}$
متغیر صفوی یک که یک نمایانگر پردازش کار $j$ و $k$ روی دو پردازنده متفاوت و صفر در غیر اینصورت	$Y_{jk}$
متغیر صفوی یک که یک نمایانگر پردازش کار $J_j$ روی ماشین $i$ و صفر در غیر اینصورت	$V_{ji}$

با استفاده از متغیرهای تصمیم  $X_{jit}$  و  $Y_{jk}$ ، مسأله بصورت ذیل مدل می‌شود:

## 1- مقدمه

در دو دهه اخیر، پردازشگرهای موازی سبب بهبود عملکرد سیستم های محاسباتی مانند پردازش اطلاعات، پایگاه های داده، و پردازش تصویر شده‌اند. در همهی این سیستم ها، مجموعه‌ای از اطلاعات (کارها)، باید روی  $m$  پردازنده موازی مشابه پردازش شوند. همچنین روابط پیشنایازی بین کارها با گراف غیرمدور،  $G(N, A)$  نمایش داده می‌شود. در این گراف  $N$  نمایانگر مجموعه کارهاست و  $A$  نمایانگر مجموعه کمانها و بیانگر نحوه اجرای کارها (از لحاظ روابط تقدم و تأخیر) می‌باشد. به عنوان مثال،  $(i, j) \in A$  به این معنی است که کار  $J_i$  برای شروع پردازش خود نیازمند خروجی کار  $J_j$  است. مزیت اصلی این سیستم ها، تقسیم کارها بین پردازنده‌ها برای بدست آوردن حداکثر کارایی می‌باشد. الگوریتم‌های ابتکاری متعددی برای حل این مسأله ارائه شده است [1-3] که اغلب آنها در [4] مرور گردیده‌اند. در اغلب این مقالات، هدف کمینه سازی طول زمانبندی است. همچنین انواع الگوریتم‌های ابتکاری و پیچیدگی محاسباتی این مسئله در مقالات متعددی مورد بررسی قرار گرفته‌اند [5].

هدف از زمانبندی در این مقاله کمینه کردن مجموع وزنی تعداد واحدهای کاری دارای دیکرد<sup>1</sup> (TWLW) است. این هدف در سیاری از سیستم هایی که به زمان و موعد تحويل کارها حساس هستند، موجود است. بطور مثال در سیستم های کنترلی [6]، که منشأ پیدایش تابع  $LW^2$  نیز بوده‌اند، دقت و کیفیت فرآیند کنترلی بستگی به مقدار اطلاعاتی دارد که بعنوان ورودی قبل از تصمیم‌گیری فراهم می‌شود. به گونه‌ای همه اطلاعاتی که توسط هر کار بعد از موعد تحويل می‌رسند، بی‌استفاده هستند، چرا که تصمیم‌گیری بدون آن اطلاعات گرفته شده‌است. واضح است که مقدار اطلاعات جمع آوری شده قبل از تصمیم‌گیری (تعداد واحدهای کارهای انجام شده قبل از موعد تحويل)، دقت و کیفیت سیستم کنترلی را تحت تأثیر قرار می‌دهد. اخیراً مرور ادبیات جامعی در مورد مسائل مطرح شده با تابع هدف  $LW$ ، چاپ شده است [7].

در این مقاله مسئله زمانبندی بر روی پردازشگرهای موازی با در نظر گرفتن تبادل اطلاعات و با تابع هدف  $TWLW$  مطرح می‌شود. نوآوری این مقاله در دو زمینه است: (الف) در نظر گرفتن تابع هدف  $TWLW$  برای مسأله زمانبندی کارها با روابط پیشنایازی مشخص بر روی پردازشگرهای موازی، (ب) ارائه یک روش حل دقیق مبتنی بر الگوریتم شاخه و کران برای مسأله.

## 2- مدل‌سازی و فرمول‌بندی مسأله

آنچه در این مقاله مورد بررسی قرار می‌گیرد، زمانبندی مجموعه‌ای از کارها،  $J$ ، بر روی مجموعه‌ای از پردازنده های موازی  $P$  و با در نظر گرفتن روابط پیشنایازی  $A$  بین کارهاست. برای هر کار، زمان پردازش،



## هشتمین کنفرانس بین‌المللی مهندسی صنایع

دانشگاه صنعتی امیرکبیر، تهران

1390 و 27 بهمن

محدودیت های (3) و (4) مدلسازی شده است. همچنین این محدودیت (3) است که مدل ارائه شده را از حالت خطی به غیر خطی تبدیل کرده است چراکه در آن متغیر  $T_j$  و  $Z_{1j}$  در هم ضرب شده اند و بنابراین قابلیت حل این مدل بصورت دقیق توسط نرم افزارهای حل مدل خطی وجود ندارد. در ادامه بررسی محدودیتها به محدودیت (5) میرسیم، از آنجا که در یک لحظه زمانی بر روی یک پردازنده حداکثر یک کار قابل پردازش می باشد، محدودیت (5) در مدل لحاظ شده است. همچنین بدیهی است که در طول پردازش یک کار بر روی یک پردازنده کار دیگری نمی تواند بر روی آن پردازنده پردازش شود که این شرط نیز در محدودیت (6) لحاظ شده است. در محدودیت (7) تأخیرهای ارتباطی که به منظور تبادل اطلاعات بین دو کار وابسته به هم که بر روی پردازنده های متفاوت ایجاد می شود، در نظر گرفته می شود. محدودیتهای شماره (8)، (9) و (10) ارتباط بین متغیرهای  $X_{jkt}$  و  $V_{ji}$  را نشان می دهند. پنجم محدودیت آخر نیز نمایانگر مقادیر ممکن برای متغیرهای تصمیم مدل است.

### 3- شاخه و کران

از روش های دقیق در حل مسائل زمانبندی، روش شمارشی شاخه و کران می باشد. در این روش تمام جواب های ممکن بوسیله یک درخت، موسوم به درخت جستجو، بررسی می شود. اما به دلیل پیچیدگی مسئله و گستردگی فضای جواب، بررسی تمام جوابها سیار زمان بر می باشد. هر درخت جستجو از گره ریشه شروع می شود، این گره نمایانگر کل فضای جواب است. در مساله مورد بحث، درخت جستجو شامل  $n$  سطح است، بطوريکه در هر سطح یک کار بر روی یک پردازنده زمانبندی می شود. بنابراین هر گره در درخت جستجو منطبق بر یک زمانبندی جزئی است.

ممکن است هر الگوریتم شاخه و کران شامل دو طرح کلیست: طرح شاخه زدن و طرح کران گذاری. در طرح شاخه زدن، درخت جستجو زمانبندی های موجه را تولید می کند و از سوی دیگر طرح کران گذاری سعی در مسدود کردن آن گره ها دارد. لازم به ذکر است که در کلیه محدودیتهای مدل، مقدار  $T$  یک حد بالا برای اندیس گسسته شمارنده زمان است که به راحتی قابل محاسبه است. مثلاً فرض کنید تمامی کارها را بر روی یک پردازنشگر و به صورت پشت سر هم انجام دهیم. در این

$$\text{Min} \sum_{j=1}^n w_j LW_j \quad (1)$$

Subject to:

$$T_j \geq t X_{jkt} - d_j; \forall j \in J, \forall i \in P, \forall t = 1, \dots, T \quad (2)$$

$$LW_j = T_j Z_{1j} + pt_j Z_{2j}; \forall j \in J \quad (3)$$

$$Z_{1j} + Z_{2j} = 1; \forall j \in J \quad (4)$$

$$\sum_{i=1}^m \sum_{t=1}^T X_{jkt} = 1; \forall j \in J \quad (5)$$

$$\sum_{j=1}^n X_{jkt} + \sum_{k=1}^{T-t} \sum_{j|pt_j > k} X_{ji(t+k)} \leq 1; \forall i \in P, \forall t = 1, \dots, T \quad (6)$$

$$\sum_{i=1}^m \sum_{t=1}^T tX_{jkt} + \Delta_{jk} Y_{jk} \leq \sum_{i=1}^m \sum_{t=1}^T tX_{kit} - pt_k; \forall (j, k) \in A \quad (7)$$

$$\sum_{t=1}^T X_{jkt} \leq V_{ji}; \forall j \in J, \forall i \in P \quad (8)$$

$$Y_{jk} \geq V_{ji} + V_{ki} - 1; \forall j, k \in J, \forall i \in P \quad (9)$$

$$Y_{jk} \geq V_{ji} + V_{ki}; \forall (j, k) \in A \quad (10)$$

$$X_{jkt} \in \{0, 1\}; \forall j \in J, \forall i \in P, t = 1, \dots, T \quad (11)$$

$$Y_{jk} \in \{0, 1\}; \forall (j, k) \in A \quad (12)$$

$$Z_{1j} \text{ و } Z_{2j} \in \{0, 1\}; \forall j \in J \quad (13)$$

$$T_j \text{ و } LW_j \in \mathbb{Z}^+; \forall j \in J \quad (14)$$

$$V_{ji} \in \{0, 1\}; \forall j \in J, \forall i \in P \quad (15)$$

تابع هدف (1)، نمایانگر حداقل مجموع وزنی واحدهای کاری دارای دیرکرد است. محدودیت (2) نمایانگر میزان تأخیر گار  $J$  است که در آن  $tX_{jkt}$  نشاندهنده زمان تکمیل آن کار است. رابطه  $LW_j = \min\{T_j, pt_j\}$  با استفاده از متغیرهای صفر و یک  $Z_{1j}$  و  $Z_{2j}$

این اصل غلبه برگرفته شده از اصل غلبه موجود در [8] است.

برای هر زمانبندی جزئی  $S$  که در آن برای دو کار مستقل  $i$  و  $j$  که بر روی یک ماشین،  $k$ ، پردازش می‌شوند،  $C_j \leq C_i$  باشد. اگر  $j \leq pt_i$  و  $|B_{ij}(S)| + LW_j(S') + LV$  برقار باشد آنگاه زمانبندی جزئی  $S'$  که دقیقاً شبیه زمانبندی  $S$  است و فقط در آن جای  $i$  و  $j$  جابجا شده است، بر زمانبندی  $S$  ترجیح داده می‌شود. در عبارت بالا  $B_{ij}(S)$  کارهای قرار گرفته بین کار  $i$  و زبرروی ماشین  $k$  است. لازم بذکر است این رابطه زمانی برقرار است که در  $(S) B_{ij}(S)$  کاری که پسندیاز کار  $i$  باشد وجود نداشته باشد.

کران بالا

برای محاسبه کران بالا دو اصل اساسی در نظر گرفته شد یکی یافتن لیست اولویتی از پردازش کارها و دیگری تخصیص کارها به ترتیب لیست اولویت برروی پردازندهای که در زودترین زمان ممکن پردازش را انجام دهد. در این مقاله برای محاسبه کران بالا اولویت پردازش با کاری بود که در گراف غیر مدور سطح بالاتری داشته باشد و برای کارهای در یک سطح اولویت با کار با زمان پردازش بزرگتر بوده است. این کران بالا به سرعت و به سادگی قابل محاسبه است و به محض یافتن جوابی با کیفیت بالاتر این مقدار بهبود داده می‌شود.

کران پایین

در زمانبندی جزئی  $S(N_l^k)$  کارها به دو دستهی کارهای زمانبندی شده،  $SA(N_l^k)$  و کارهای باقیمانده،  $RA(N_l^k)$  تقسیم می‌شوند. کران پایین در این گره بصورت زیر محاسبه می‌شود:

$$LB(N_l^k) = \sum_{J_j \in SA} \dots + \sum_{J_j \in RA}$$

همانطور که از رابطه‌ی بالا پیداست، برای تعیین کران پایین نیاز به زودترین زمان اتمام برای کار  $j$  ( $ef_j$ ) داریم، اما از آنجاییکه در این مسئله تخصیص پردازنده مناسب به هر کار نیز مد نظر است، این زودترین زمان اتمام هر کار را برای هر ماشین جداگانه محاسبه می‌کنیم. بدین ترتیب  $ef_j = \min_{i \in EP} \{ef_{ij}\}$

$$ef_{ij} = \max \left\{ \max_{\substack{i' \in EP \& i' \neq i \\ k' \in Pred(j) \& k' \neq K}} (ef_{i'k'} + \Delta_{k'j} + pt_k), ef_{ik} + pt_j \right\}$$

که در آن  $ef_{ij}$  به معنای زودترین زمان اتمام کار  $j$  روی پردازنده  $i$  است. کار  $k$  عضو مجموعه پیشنهایی کار  $j$  است.

#### 4- نتایج محاسباتی

حالت، طول زمانبندی حاصل می‌تواند به عنوان مقدار  $T$  در نظر گرفته شود.

#### 3-1- طرح شاخه‌زدن

گرهی  $N_l^k$  را امین گرهی موجود در سطح  $l$  از درخت جستجو در نظر بگیرید. این گره منطبق بر زمانبندی جزئی  $S(N_l^k)$  است. همچنین مجموعه کارهای مجاز، مجموعه‌ای شامل همهی کارهایی که پیشنهای آنها در  $S$  زمانبندی شده‌اند، با  $E(N_l^k)$  نمایش داده می‌شود. هر فرزند گره  $N_l^k$  با تخصیص یک کار از مجموعه  $E(N_l^k)$  به یک پردازنده ایجاد می‌شود. به عبارت دیگر از هر گره  $E(N_l^k)$   $m$  تعداد گره فرزند  $|E(N_l^k)| \times m$  وجود دارد. در بدترین حالت که هیچ گونه روابط پیشنهایی بین کارها ایجاد می‌شود. در بدترین حالت که در آن  $n$  گره  $N_l^k$  از گره  $A = \emptyset$  است،  $l = 1, \dots, n$ . که در آن  $l$  تعداد  $\times m^{l+1} \times (n-l) \dots n$  گره فرزند در سطح  $l+1$  ایجاد می‌شود. در این حالت حداقل  $n! m^n$  زمانبندی ایجاد می‌شود و بنابراین پیچیدگی محاسباتی از درجه  $O(n! m^n)$  خواهد بود. این پیچیدگی محاسباتی نشان می‌دهد که الگوریتم شاخه و کران برای مقادیر بزرگ  $m$  و  $n$  قبل اجرا نیست مگر اینکه طرح کران گذاری کارایی برای مسدود کردن گره‌های زائد وجود داشته باشد.

#### 3-2- طرح کران گذاری

طرح کران گذاری شامل سه اصل غلبه است. اصل غلبه اول از ایجاد جواب‌های تکراری جلوگیری می‌کند و اصل‌های غلبه 2 و 3 از بوجود آمدن جواب‌های بی کیفیت جلوگیری می‌کند.

تعریف: پروفایل پردازنده  $i$ , در گره  $N_l^k$ , به صورت  $MP_{N_l^k}^i = 1, \dots, m$  در گره  $i$ , در سطح  $l$  درخت جستجو قابل زمانبندی نمایش داده می‌شود و مطابق زیر تعریف می‌شود:

اگر در بازی زمانی  $[t-1, t]$  هیچ کاری روی ماشین اپردازش نشود؛ در غیراینصورت  $MP_{N_l^k}^i = \begin{cases} 1 & \text{اصل غلبه 1:} \\ 0 & \end{cases}$

فرض کنید  $r_j$  در سطح  $l+1$  درخت جستجو قابل زمانبندی باشد، بطوریکه زمانبندی این گره روی پردازنده‌های مختلف توسط گره‌های خروجی از  $N_l^k$  مشخص باشد. اگر زودترین زمان شروع کار  $j$  روی یک مجموعه  $\bar{A}$  از پردازنده‌ها، زمان مشابه و ثابت  $t^*$  باشد و  $t = t^*, t^* + \dots, T$  برای همهی پردازنده‌های  $i \in \bar{A}$  و همهی زمان‌های  $t = t^*, t^* + 1, \dots, T$  یکسان باشد، زمانبندی  $r_j$  را فقط روی پردازنده با کمترین شماره در مجموعه  $\bar{A}$  قرار در نظر بگیرید.

از آنجاییکه بررسی پروفایل پردازنده بر روی هر گره کار بسیار زمانبندی است، ما این قاعده را فقط برای حالتی که  $MP_{N_l^k}^i(t) = 0$  و برای  $t = t^*, t^* + 1, \dots, T$  استفاده می‌کنیم.

اصل غلبه 2:



## هشتمین کنفرانس بین‌المللی مهندسی صنایع

دانشگاه صنعتی امیرکبیر، تهران

1390 و 27 بهمن

	0.5	38.2	0.0116	94.764
	0.75	55.4	0.0308	494.62
25	0.25	32	0.001	404.462
	0.5	37.8	0.0522	2598.03
30	0.75	63.2	0.0118	2001.87
	0.25	45.2	0.0088	1057.5
	0.5	44.6	0.0952	3097.25
	0.75	68.2	0.0368	3600

در جدول بالا "جب" نمایانگر متوسط جواب بهینه حاصل شده از نمونه تصادفی موجود در هر دسته است. همچنین "ش.ک" و "Cplex" بترتیب بیانگر متوسط زمان CPU برای رسیدن به جواب بهینه توسط الگوریتم شاخه و کران و Cplex است. برای اجرای Cplex محدودیت زمانی 3600 ثانیه‌ای گذاشته شده است. Cplex در چهار مورد جواب غیر موجه ایجاد کرده و در پنج مورد نیز در زمان مقرر جواب بهینه را نیافتد است.

### 5- مراجع

- T. Thanalapati, S. Dandamudi, "An efficient adaptive scheduling scheme for distributed memory multicomputers," IEEE Transactions on Parallel and Distributed Systems, vol. 12(7), pp.758–68, 2001. [1]
- N. Nissanke, A. Leulsegeld, S. Chillara, "Probabilistic performance analysis in multiprocessor scheduling," Journal of Computing and Control Engineering, vol.13(4), pp.171–9, 2002. [2]
- J. Corbalan, X. Martorell X, J. Labarta, "Performance -driven processor allocation," IEEE Transactions on Parallel and Distributed Systems, vol. 16(7), pp.599–611, 2005. [3]
- R. Hwang, M. Gen, H. Katayama, "A comparison of multiprocessor task scheduling algorithms with communication costs," Computers & Operations Research, vol. 35, pp. 976-993, 2008. [4]
- R.Giroudeau, J.C. Konig, F.K. Moulai, J. Palaysi, "Complexity and approximation for precedence constrained scheduling problems with large communication delays," Theoretical Computer Science, vol. 401, pp. 107-119, 2008. [5]
- J. Blazewicz, "Scheduling preemptible tasks on parallel processors with information loss," Technique et Science Informatiques, vol. 3(6), pp.415–20, 1984. [6]

الگوریتم شاخه و کران ارائه شده توسط زبان برنامه نویسی C کدنویسی شده است و توسط Visual C++ 6 اجرا شده است. نمونه ها بر روی کامپیوتر با مشخصات 1 GB RAM و Pentium IV 3 GHz که دارای ویندوز XP بود، اجرا شدند. 27 نمونه تصادفی با استفاده از نرم افزار RanGen [9] و در نظر گرفتن مقادیر مختلف برای تمامی فاکتورهای لازم برای طراحی، ایجاد شده است که عبارتند از: تعداد کارها (m = 6 و 4) و (n = 20 و 25)، تعداد پردازنده ها (OS = 0.25 و 0.5) و میزان ارتباط بین کارها (OS) که از تقسیم تعداد روابط پیشنهایزی بر حداکثر تعداد آنها حاصل می شود (OS = 0.75 و 0.5). هر مجموعه داده شامل 5 نمونه مساله تصادفی است و بنابراین 135 مسئله موردی مورد بررسی قرار می گیرد. وزن هر کار توسطتابع یکنواخت بین 1 تا 10 بطور تصادفی ایجاد می شود. موعد تحويل کارها مطابق با روش ذکر شده در [10] ایجاد شده است. در این روش، ابتدا بزرگترین موعد تحويل را معادل طول زمانبندی بر اساس روش مسیر بحرانی در نظر می گیریم و آن را به آخرین کار تخصیص می دهیم، بعد از آن بین 1 تا حداکثر موعد تحويل شروع به انتخاب اعداد تصادفی و تخصیص آن اعداد به کارهای باقیمانده می کنیم. در این روش اعداد تصادفی اتخاذ شده به ترتیب صعودی به کارها که به ترتیب صعودی شماره کارها مرتب شده اند تخصیص داده می شوند. به عبارت دیگر کوچکترین موعد تحويل به کار شماره 1 اختصاص می یابد و به همین ترتیب ادامه می یابد. همانطور که در قبل نیز اشاره شد مدل مسئله بصورت غیر خطی است و بنابراین صرفا جهت نمایش عملکرد الگوریتم ارائه شده با حذف محدودیتهای (2) تا (4) مسئله را به یک مسئله پردازندهای موازی با همه شرایط مذکور و با تابع هدف حداقل طول زمانبندی تبدیل کرده که جهت حل مدل ارائه شده، نرم افزار ILOG CPLEX 12.1 [11] مورد استفاده قرار گرفته است و نتایج حاصل از آن در جدول 3 آمده است. لازم به ذکر است که در حل تمامی نمونه ها توسط این نرم افزار، پارامترهای CPLEX پیشفرض مورد استفاده قرار گرفته اند.

جدول 3. نتایج محاسباتی با تابع هدف حداقل طول زمانبندی و m=4

n	OS	جب	ش.ک	Cplex
20	0.25	28.8	0.0048	70.46

- M. Sterna, "A survey of scheduling problems with late work [7]  
criteria," Omega, vol. 39, pp.120-129, 2011.
- SO. Shim, YD. Kim, "Scheduling on parallel identical [8]  
machines to minimize total tardiness," European Journal of  
Operational Research, vol. 177, pp. 135–146, 2007.
- E. Demeulemeester, M. Vanhoucke, W. Herroelen, "A [9]  
random generator for activity-on-the-node networks," Journal  
of Scheduling, vol.6, pp.13-34, 2003.
- M. Vanhoucke, E.L. Demeulemeester, W.S. Herroelen, "An [10]  
exact procedure for the resource-constrained weighted  
earliness-tardiness project scheduling problem," Annals of  
Operations Research, vol.102, pp.179–196, 2001.
- IBM ILOG, User's Guide CPLEX12.1, 2010, IBM ILOG. [11]

زیرنویس‌ها:

---

<sup>1</sup> Total Weighted Late Work

<sup>2</sup> Late Work

<sup>3</sup> Tardiness

<sup>4</sup> Order strength