

کمینه‌کردن تعداد کارهای تأخیردار در زمان‌بندی جریان کارگاهی با کارهای رو به زوال و ورودی‌های غیر همزمان

محمدباقر فخرزاد* (دانشیار)

محمدسلیمان ندوشن (دانشجوی کارشناسی ارشد)

گروه مهندسی صنایع، دانشگاه یزد

مهندسی صنایع و مدیریت شریف، تابستان ۱۳۹۶ (دوره ۱، شماره ۱/۱، ص. ۷۵-۸۵)

در این مقاله، زمان‌بندی کارهای رو به زوال با تابع زوال خطی روی جریان کارگاهی دو ماشین با هدف کمینه‌کردن تعداد کارهای تأخیردار و با فرض ورود غیر همزمان کارها بررسی شده است. برای به دست آوردن جوابی نزدیک به بهینه در زمانی کوتاه، یک روش ابتکاری و برای حل دقیق آن یک الگوریتم شاخه و کران ارائه شده است. نتایج نشان می‌دهد الگوریتم شاخه و کران مسائل را تا ۲۴ کار در رده مسائل بزرگ و ۲۲ کار در رده مسائل کوچک، در زمان منطقی حل می‌کند. همچنین درصد بالایی از گره‌ها در روش شاخه و کران توسط اصول غلبه و حدود پایین قطع می‌شود که نشان‌دهنده کارایی الگوریتم شاخه و کران است. متوسط نسبت جواب بهینه به جواب الگوریتم ابتکاری حداکثر برابر ۱/۱۵ است که این عدد در مقایسه با سایر تحقیقات مربوط به تعداد کارهای تأخیردار عدد بسیار خوبی است.

واژگان کلیدی: فعالیت‌های رو به زوال، جریان کارگاهی، تعداد کارهای تأخیردار، شاخه و کران، ورود غیر همزمان.

mfakhrzad@yazd.ac.ir
mohamadsoleimian88@gmail.com

۱. مقدمه

ماشین، β نشان‌دهنده ویژگی‌های مسئله و γ نشان‌دهنده تابع هدف مسئله است.

لی و همکاران^[۱] مسئله $|P_i = a_i + bt, r_i|C_{\max}$ را مورد بررسی قرار دادند. آنها در این بررسی یک الگوریتم شاخه و کران به همراه چندین اصل غلبه و دو حد پایین ارائه کرده‌اند به طوری که مسائل با تعداد 2^0 فعالیت را در زمان مناسبی حل می‌کنند.

لی و همکاران^[۵] در بررسی دیگری مسئله $F2|P_i = a_i + bt|C_{\max}$ را مطالعه کرده‌اند. آنها در این بررسی یک الگوریتم شاخه و کران را با در نظر گرفتن اصول غلبه و دو حد پایین ارائه، و مسائل شامل ۲۴ فعالیت را در زمان مناسبی حل کرده‌اند. آنها همچنین چند الگوریتم ابتکاری مبتنی بر قاعده‌ی کوچک‌ترین زمان پردازش^۲ ارائه و نشان داده‌اند که خطای این الگوریتم کم، و حداکثر برابر با ۱۶٪ است. همچنین نشان داده شده که توالی موجود طبق قاعده‌ی SPT براساس a_i ، بهترین جواب در بین الگوریتم‌های ابتکاری است.

وانگ و همکاران^[۶] در بررسی خود عنوان کرده‌اند که مسئله‌ی زمان‌بندی جریان کارگاهی برای کمینه‌کردن زمان تکمیل کل در حالت پایه یک مسئله‌ی NP-hard است. بنابراین این مسئله با در نظر گرفتن تابع رو به زوال خطی نیز NP-hard است. آنها در تحقیق خود مسئله $F2|P_i = b_i t|C_i$ را

در مسائل پایه‌ی زمان‌بندی فرض می‌شود که زمان پردازش فعالیت‌ها مقدار ثابت و از پیش تعیین شده‌ی است. این فرض با توجه به این که ماشین یا ابزارآلات در طول زمان مستهلک می‌شود و کارایی آن پایین می‌آید، در بیشتر موارد غیر واقعی به نظر می‌رسد. به علاوه در برخی از صنایع مانند صنعت فولاد منتظر ماندن یک کار برای پردازش، زمان پردازش را افزایش می‌دهد. زمان پردازش در این حالت‌ها براساس یک تابع غیرکاهشی یا غیرافزایشی در نظر گرفته می‌شود. در ادبیات زمان‌بندی، این پدیده را «زوال»^۱ می‌نامند.

نمونه‌هایی از کاربرد این مسئله در کارخانه‌های فولاد و نساجی مشاهده می‌شود. قطعات حاصل از فرایند تراشکاری نیز نمونه‌ی عملی دیگری در این زمینه به شمار می‌آید. ابزارآلات تراشکاری با انجام فرایند تراشکاری فرسوده می‌شوند، و به همین دلیل زمان مورد نیاز برای تراشکاری قطعات بعدی براساس زمان پردازش قطعات تراشکاری شده افزایش می‌یابد.^[۱] بیشتر تحقیقات انجام شده در مورد کارهای رو به زوال روی تک‌ماشین انجام شده است. از نظر تابع هدف نیز بیشتر تحقیقات روی کمینه‌کردن بیشترین زمان تکمیل متمرکز شده‌اند.^[۲]

در ادامه برای سهولت در نشان دادن مسئله از نمادگذاری گراهام^[۳] استفاده می‌شود. این نمادگذاری به صورت $\alpha|\beta|\gamma$ است که در آن α نشان‌دهنده محیط

* نویسنده مسئول

تاریخ دریافت: ۱۳۹۳/۱۰/۰۷، اصلاحیه ۱۳۹۴/۰۱/۱۵، پذیرش ۱۳۹۴/۰۶/۲۸.

در دو بخش مورد بررسی قرار داده اند. در بخش اول مسئله را در چهار حالت خاص بررسی و ترتیبی بهینه برای آن ارائه کرده اند. در بخش دوم برای مسئله در حالت کلی یک الگوریتم شاخه و کران با در نظر گرفتن چند ویژگی غلبه و ۲ حد پایین ارائه و مسائل با ۱۴ فعالیت را حل کرده اند. لی و همکاران^[۷] در مطالعه خود مسئله $PfM|P_i = a_i + b_i t|\sum C_i$ را بررسی کرده اند. آن‌ها در این بررسی یک الگوریتم شاخه و کران با در نظر گرفتن اصول غلبه و حد پایین ارائه و مسائل با ابعاد ۱۸ فعالیت و ۲ ماشین را در زمان قابل قبولی به صورت بهینه حل کرده اند. همچنین چند الگوریتم ابتکاری ارائه و نشان داده اند که همگی کارا هستند. لی و همکاران^[۸] مسئله کمیته کردن دیرکرد در محیط جریان کارگاهی و با کارهای رو به زوال را بررسی کردند و یک روش فراابتکاری مبتنی بر الگوریتم ژنتیک و یک روش شاخه و کران برای حل پیشنهاد کردند، که روش شاخه و کران پیشنهادی مسائلی با ۲۰ کار را به صورت بهینه در زمانی مناسب حل می کند.

یکی از معیارهای مهم که در مسائل زمان بندی در نظر گرفته می شود کمیته کردن تعداد کارهای تأخیردار است که متناظر با هزینه کمیته در محیط های تولیدی است. جعفری و مصلحی^[۹] ضمن بررسی مسئله $\sum U_i$ نشان $|p_i = a_i + bS|$ نشان داده اند که مسئله از نوع NP-hard است. آن‌ها در این بررسی یک الگوریتم شاخه و کران که قادر است مسائلی با ۲۸ کار را در زمانی کوتاه حل کند و یک الگوریتم ابتکاری برای مسائل بزرگ تر ارائه کرده اند؛ سپس با حل ۱۸۰۰ مسئله نمونه نشان داده اند که الگوریتم ابتکاری از کارایی بالایی برخوردار است و می تواند جواب هایی نزدیک به بهینه تولید کند.

حکیمیان و همکاران^[۱۰] مسئله $F2|r_i|\sum U_i$ را مورد بررسی قرار داده و نشان داده اند که این مسئله NP-hard است. سپس برای حل آن سه الگوریتم ابتکاری و یک الگوریتم شاخه و کران به همراه چندین اصل غلبه و سه حد پایین ارائه کرده و مسائل شامل ۲۸ فعالیت را در زمان مناسبی حل کرده اند. همانطور که مشاهده شد، مسئله زمان بندی کارهای رو به زوال روی جریان کارگاهی دو ماشین با هدف کمیته کردن تعداد کارهای تأخیردار و با ورود غیر همزمان کارها تاکنون مورد بررسی قرار نگرفته است. بنابراین در این مقاله این مسئله مورد بررسی قرار می گیرد.

در ادامه در بخش دوم مسئله مورد نظر معرفی، و سپس رویکرد حل مسئله بیان می شود. سپس نتایج محاسباتی ارائه می شود، و در بخش چهارم نتیجه گیری و پیشنهاد برای پژوهش های آینده ارائه خواهد شد.

یک توالی نزدیک به بهینه در زمانی کوتاه ارائه می شود. سپس الگوریتم شاخه و کران ارائه و لم های توسعه داده شده برای بالا بردن کارایی این الگوریتم بیان و اثبات می شود.

۳. پارامترها و متغیرهای مسئله

A : زمان تکمیل توالی جزئی σ روی ماشین ۱؛

a_m : زمان پردازش پایه یی کار i روی ماشین m ؛

B : زمان تکمیل توالی جزئی σ روی ماشین ۲؛

b : نرخ زوال؛

C_{im} : زمان تکمیل کار i روی ماشین m ؛

d_i : موعد تحویل کار i ؛

j : کار i ؛

M : یک عدد بزرگ؛

n : تعداد کارهایی که باید زمان بندی شود؛

r_i : زمان ورود کار i ؛

P_{im} : زمان پردازش واقعی کار i روی ماشین m که از مجموع زمان پایه و مقدار زوال به دست می آید؛

S_{im} : زمان شروع فعالیت i روی ماشین m ؛

U_i : اگر $d_i \geq C_{i2}$ معادل صفر و در غیر این صورت برابر با ۱ است؛

X_{ij} : اگر کار i در نوبت j از توالی انجام شود ۱ و در غیر این صورت صفر است؛

σ : یک توالی جزئی زمان بندی شده شامل کارهای ۱ تا i ؛

σ' : مجموعه ی کارهای زمان بندی نشده (مکمل مجموعه ی σ)؛

π : یک توالی شامل تمام کارها؛

T_{ij} : مقدار دیرکرد کار در نوبت j ؛

i : اندیس کار ($i = 1, 2, 3, \dots, n$)؛

m : اندیس ماشین ($m = 1, 2$)؛

j : اندیس نوبت توالی ($j = 1, 2, 3, \dots, n$)؛

۴. مدل برنامه ریزی خطی

مدل برنامه ریزی خطی عدد صحیح مختلط مسئله به صورت زیر است. در ادامه پس از ارائه ی مدل طراحی شده تابع هدف و محدودیت های مدل نیز تشریح می شوند.

$$\text{Minimize } N_T = \sum_{j=1}^n U_{[j]} \quad (1)$$

s.t :

$$\sum_{j=1}^n x_{i[j]} = 1 \quad i = 1, 2, \dots, n \quad (2)$$

$$\sum_{i=1}^n x_{i[j]} = 1 \quad j = 1, 2, \dots, n \quad (3)$$

$$C_{[j]1} \geq \sum_{i=1}^n r_i x_{i[j]} + \sum_{i=1}^n P_{i1} x_{i[j]} \quad j = 1, 2, \dots, n \quad (4)$$

۲. معرفی مسئله

در این قسمت مسئله زمان بندی کارهای رو به زوال با تابع زوال خطی روی جریان کارگاهی دو ماشین با هدف کمیته کردن تعداد کارهای تأخیردار و با فرض ورود غیر همزمان کارها بررسی می شود. در این مسئله نرخ زوال برای تمام کارها یکسان، و کارها مستقل از یکدیگر فرض می شوند. همچنین ماشین ها در تمام طول دوره برنامه ریزی در دسترس فرض می شود. این مسئله به صورت متغیرهای مسئله معرفی می شود. سپس مدل برنامه ریزی خطی عدد صحیح مسئله معرفی می شود. در ادامه پیچیدگی مسئله مورد بررسی قرار گرفته و ثابت می شود مسئله از نوع NP-hard است. در قسمت بعد روش ابتکاری برای دستیابی به

چنگ و دینگ^[۱۲] نشان داده‌اند در صورتی که سایر فرضیات ثابت باشد مسائل $C_{[j]1} \geq C_{[j-1]1} + \sum_{i=1}^n P_{i1} x_{i[j]}$ و $\forall |r_i| C_{max}$ و $\forall |L_{max}$ زوال خطی و با نرخ زوال یکسان برای همه کارها به هر دو مسئله اضافه شود مسائل $\forall |P_i = a_i + bS, r_i| C_{max}$ و $\forall |P_i = a_i + bS| L_{max}$ خواهند بود. ثابت شده است که مسئله $\forall |P_i = a_i + bS, r_i| C_{max}$ دارای پیچیدگی NP-hard است. بنابراین نتیجه می‌شود که مسئله $\forall |P_i = a_i + bS| L_{max}$ نیز NP-hard است.^[۱۳] حال با توجه به این که مسئله $\forall |P_i = a_i + bS| L_{max}$ قابل کاهش به مسئله $\forall |P_i = a_i + bS| \sum U_i$ و این مسئله نیز قابل کاهش به $Pf2 |P_i = a_i + bS, r_i| \sum_{i=1}^n U_i$ است نتیجه می‌شود که این مسئله نیز NP-hard است.

با توجه به این اثبات، منطقی است که برای حل بهینه‌ی این مسئله از روش‌های عمومی حل مسائل زمان‌بندی مانند شاخه و کران و برای به دست آوردن جوابی نزدیک به بهینه برای آن از روش‌های ابتکاری استفاده شود.

۶. حل مسئله

۶.۱. الگوریتم ابتکاری

الگوریتم مور-هاجسون برای مسئله $\forall | \sum U_i$ منجر به جواب بهینه می‌شود.^[۱۳] لذا با الهام از این الگوریتم، برای مسئله $Pf2 |P_i = a_i + bS, r_i| \sum_{i=1}^n U_i$ الگوریتمی ابتکاری به نام H۱ در چهار گام ارائه شده است:

۱. ابتدا کارها به صورت زودترین موعد تحویل^۳ مرتب شوند.
۲. سپس کارها براساس ترتیب زودترین موعد تحویل و با توجه به زمان در دسترس هر کار (r_i)، به ماشین اول و دوم تخصیص یابد و زمان تکمیل هر کار محاسبه شود.
۳. اگر اولین کار تأخیردار در نوبت z از توالی باشد، از ابتدای ترتیب تا نوبت z هر دفعه یکی از کارها حذف شده و میزان صرفه‌جویی در زمان ختم کار در نوبت z به دست آورده شود. کاری که حذف آن منجر به بیشترین کاهش در زمان ختم کار در نوبت z ، روی ماشین دوم می‌شود در آخر توالی قرار گرفته و این کار تأخیردار است. هر موقع کاری به آخر توالی برده شود از تعداد کل کارها یکی کم می‌شود (یعنی اولین کار حذف شده در آخر توالی قرار می‌گیرد و دومین کار حذف شده در مرحله‌ی بعدی در نوبت $z-1$ ، و به همین ترتیب محل کارهای تأخیردار بعدی مشخص می‌شود).

۴. برای بقیه‌ی کارها مجدداً گام‌های ۲ و ۳، مادامی که کار تأخیردار یافت شود اجرا می‌شود.

نتایج حل نشان می‌دهد که این الگوریتم از کارایی خوبی برخوردار است. سه الگوریتم ابتکاری دیگر به نام H۲، H۳ و H۴ نیز مورد بررسی قرار گرفته است. در H۲ به جای کار حذف شده در گام سوم الگوریتم H۱، کاری که بیشترین زمان پردازش روی ماشین اول و دوم داشته باشد حذف می‌شود. به عبارت دیگر، کار با بیشترین $P_{i1} + P_{i2}$ حذف می‌شود. در الگوریتم‌های H۳ و H۴ به ترتیب یک گام بین گام‌های ۲ و ۳ الگوریتم‌های H۱ و H۲ اضافه می‌شود تا بیکاری‌های غیرعمد ناشی از ورود غیرهمزمان کارها کاهش یابد. این گام به صورت زیر اجرا می‌شود:

گام ۳ الگوریتم‌های H۳ و H۴: برای هر نوبت z اگر $r_{i[z]} > C_{i[z-1]}$ باشد، از بین کارهای قرار گرفته در نوبت z و نوبت‌های بعد از آن، کاری که کوچک‌ترین r_i

$$C_{[j]1} \geq C_{[j-1]1} + \sum_{i=1}^n P_{i1} x_{i[j]} \quad j = 1, 2, \dots, n \quad (5)$$

$$P_{i1} = a_{i1} + bS_{i1} \quad i = 1, 2, \dots, n \quad (6)$$

$$S_{i1} \geq \sum_{j=1}^n C_{[j-1]1} x_{i[j]} \quad i = 1, 2, \dots, n \quad (7)$$

$$S_{i1} \geq r_i \quad i = 1, 2, \dots, n \quad (8)$$

$$C_{[j]2} \geq C_{[j]1} + \sum_{i=1}^n P_{i2} x_{i[j]} \quad j = 1, 2, \dots, n \quad (9)$$

$$C_{[j]2} \geq C_{[j-1]2} + \sum_{i=1}^n P_{i2} x_{i[j]} \quad j = 1, 2, \dots, n \quad (10)$$

$$P_{i2} = a_{i2} + bS_{i2} \quad i = 1, 2, \dots, n \quad (11)$$

$$S_{i2} \geq \sum_{j=1}^n C_{[j-1]2} x_{i[j]} \quad i = 1, 2, \dots, n \quad (12)$$

$$S_{i2} \geq \sum_{j=1}^n C_{[j]1} x_{i[j]} \quad i = 1, 2, \dots, n \quad (13)$$

$$T_{[j]} \geq C_{[j]2} - \sum_{i=1}^n d_i x_{i[j]} \quad i = 1, 2, \dots, n \quad (14)$$

$$T_{[j]} \geq 0 \quad j = 1, 2, \dots, n \quad (15)$$

$$T_{[j]} \leq M \cdot U_{[j]} \quad j = 1, 2, \dots, n \quad (16)$$

$$x_{i[j]} \in \{0, 1\} \quad i, j = 1, 2, \dots, n \quad (17)$$

$$U_{[j]} \in \{0, 1\} \quad j = 1, 2, \dots, n \quad (18)$$

رابطه‌ی ۱ تابع هدف مسئله را نشان می‌دهد. رابطه‌ی ۲ تضمین می‌کند که هر یک از کارها حتماً یک مرتبه در یکی از n نوبت از توالی قرارگیرد. رابطه‌ی ۳ نشان می‌دهد که هر نوبت مانند z حتماً باید شامل یک کار باشد. محدودیت‌های ۴ و ۵ زمان تکمیل هر یک از کارها روی ماشین ۱ را تعیین می‌کند. رابطه‌ی ۶ نشان می‌دهد که زمان پردازش واقعی کارها روی ماشین ۱ چگونه به دست می‌آید. محدودیت‌های ۷ و ۸ زمان شروع هر یک از کارها روی ماشین ۱ را تعیین می‌کند. محدودیت‌های ۹ تا ۱۳ عملکردی مشابه محدودیت‌های ۴ تا ۸ دارند و به ترتیب زمان تکمیل، زمان پردازش واقعی و زمان شروع کارها روی ماشین ۲ را تعیین می‌کنند. از محدودیت‌های ۱۴ تا ۱۶ برای تعیین برخورداری از دیرکرد کارها استفاده شده است.

۵. بررسی پیچیدگی مسئله

زمان حل مسئله‌ی زمان‌بندی رابطه‌ی مستقیمی با درجه‌ی سختی محاسبات آن دارد. سختی محاسبات یک مسئله براساس نظریه‌ی پیچیدگی مشخص می‌شود. این نظریه مشخص می‌کند مسائل از نظر محاسباتی در کدام یک از دو گروه ساده و سخت قرار گرفته، و آیا یک مسئله در زمان چندجمله‌یی قابل حل خواهد بود یا خیر.^[۱۱] با توجه به این که در ادبیات موضوع تاکنون مطالعه‌ی روی مسئله‌ی $Pf2 |P_i = a_i + bS, r_i| \sum_{i=1}^n U_i$ مشاهده نشده است، لازم است ابتدا پیچیدگی آن بررسی شود. بروکر^[۱۱] نشان داده است هر مسئله با تابع هدف L_{max} در صورتی که سایر فرضیات ثابت باشد قابل کاهش به مسئله‌ی مشابهی با تابع هدف $\sum U_i$ است.

را داشته باشد (مثلاً کار نوبت k) را به نوبت z منتقل کرده و کار قبلی در نوبت z را به $z + 1$ منتقل کنید. سپس کل کارهای نوبت z تا $k - 1$ یک نوبت به سمت انتهای توالی انتقال می‌یابد.

۲.۶. الگوریتم شاخه و کران

برای به دست آوردن جواب بهینه‌ی مسئله، یک الگوریتم شاخه و کران که از استراتژی پرسو^۴ بهره می‌گیرد استفاده شده است. برای عملکرد بهتر این الگوریتم از بهترین جواب به دست آمده توسط الگوریتم‌های ابتکاری به عنوان حد بالا استفاده می‌شود. همچنین تعدادی حد پایین و اصول غلبه توسعه داده شده است.

برای بیان اصول غلبه و حدود پایین فرض می‌شود که توالی جزئی σ با زمان تکمیل A روی ماشین ۱ و B روی ماشین ۲ در دست است. اگر کار i بعد از σ قرار گیرد با σi قرار گیرد با $z i$ قرار گیرد با $z i$ نشان داده می‌شود. توالی $\sigma z i$ از جابه‌جایی جفت مجاور i و z در توالی $z i$ به دست می‌آید.

۱.۲.۶ اصول غلبه

استفاده از اصول غلبه در حل مسائل زمان‌بندی باعث می‌شود که زیرمجموعه‌ی از مجموعه‌ی جواب که جواب بهینه در آن قرار دارد مشخص شود، و با کوچک‌تر شدن فضای جست‌وجو و در نتیجه افزایش سرعت حل مسئله، توالی بهینه در مدت زمان کوتاه‌تری به دست آید. به منظور کاهش زمان حل در الگوریتم شاخه و کران ارائه شده، تعدادی لم بیان و اثبات شده است.

لم ۱. در مسئله $P_i = a_i + bS, r_i | \sum_{i=1}^n U_i$ اگر $PF2$ اگر رابطه‌ی $\max\{C_{max}\} \geq d_i$ در مورد کار i برقرار باشد آنگاه می‌توان در توالی بهینه این کار را بدون دیرکرد در انتهای توالی قرار داد.

مقدار $\max\{C_{max}\}$ از توالی به دست می‌آید که در آن کارها روی ماشین اول به ترتیب غیر افزایشی a_{i1} ها و روی ماشین دوم به ترتیب غیر کاهششی a_{i2} ها مرتب شده باشند و زمان شروع پردازش اولین کار روی ماشین اول برابر $\max r_i$ باشد. گام‌های لازم برای محاسبه‌ی $\max\{C_{max}\}$ به ترتیب عبارت است از:

۱. زمان‌های پردازش کارها روی ماشین ۱ (a_{i1}) را به ترتیب غیر افزایشی مرتب کنید.
۲. زمان‌های پردازش کارها روی ماشین ۲ (a_{i2}) را به ترتیب غیر کاهششی مرتب کنید.

۳. زمان‌های در دسترس کارها (r_i) را به ترتیب غیر کاهششی مرتب کنید.

۴. از مجموعه‌های مرتب شده‌ی فوق اولین a_{i1} و a_{i2} را به نوبت اول در توالی اختصاص دهید و زمان شروع پردازش در این نوبت را برابر $\max r_i$ قرار دهید. به ترتیب بقیه‌ی a_{i1} ها و a_{i2} ها را از مجموعه‌های مرتب شده‌ی فوق انتخاب کنید و به نوبت‌های بعدی اختصاص دهید.

۵. زمان تکمیل آخرین نوبت را محاسبه کنید. این زمان برابر $\max\{C_{max}\}$ است.

اثبات: با شروع پردازش اولین کار روی ماشین اول در زمان $\max r_i$ بیشترین بیکاری عمده‌ی روی ماشین ۱ خواهیم داشت. از طرفی وقتی a_{i1} ها را به ترتیب غیر افزایشی و a_{i2} ها را به ترتیب غیر کاهششی به نوبت‌های توالی تخصیص دهیم باعث می‌شود روی ماشین ۲ نیز بیشترین بیکاری را داشته باشیم. همچنین ترتیب غیر افزایشی a_{i1} ها باعث می‌شود زمان تکمیل کارها روی ماشین اول بیشترین مقدار ممکن را داشته باشد؛ در نتیجه بیشترین مقدار زوال برای انجام کارها روی ماشین ۱ ایجاد می‌شود. بنابراین بیشترین مقدار C_{max} در این حالت به دست می‌آید. حال

اگر شرط $d_i \geq \max\{C_{max}\}$ در مورد کار i برقرار باشد، این کار در هر نوبت از توالی که قرار گیرد تأخیردار نخواهد شد. بنابراین اگر در انتهای توالی قرار گیرد تعداد کارهای تأخیردار را افزایش نخواهد داد. □

لم ۲. در مسئله $P_i = a_i + bS, r_i | \sum_{i=1}^n U_i$ اگر $PF2$ اگر $r_i = \min_{j \in \sigma'} \{r_j\}$ ، $a_{i1} = \min_{j \in \sigma'} \{a_{j1}\}$ ، $a_{i2} = \min_{j \in \sigma'} \{a_{j2}\}$ و $d_i = \min_{j \in \sigma'} \{d_j\}$ باشد، آنگاه چنانچه

کار i در توالی σi تأخیردار نباشد، در توالی بهینه در زمان A زمان‌بندی می‌شود. اثبات: از آنجا که کار i کم‌ترین زمان پردازش پایه را روی ماشین ۱ و ۲ دارد بنابراین کم‌ترین زمان تکمیل را خواهد داشت و به تبع آن، زمان شروع و پردازش کارهای بعدی نیز کم‌تر می‌شود. از طرفی به دلیل داشتن کم‌ترین موعد تحویل و تأخیردار نشدن آن می‌توان مطمئن شد که زمان‌بندی کار i در زمان A تعداد کارهای تأخیردار را افزایش نمی‌دهد. □

نتیجه ۱. با اجرای این لم در ابتدای شروع الگوریتم فضای جواب یک واحد کاهش می‌یابد. ممکن است بتوان چندین بار لم فوق را به کار برد. این امر باعث کاهش چندین واحدی فضای جواب می‌شود. علاوه بر آن در فرایند جست‌جوی عمقی در الگوریتم شاخه و کران، اجرای این لم باعث می‌شود بهترین کار از مجموعه‌ی کارهای زمان‌بندی نشده (مجموعه σ') انتخاب شود و بنابراین نیازی به بررسی شاخه‌های دیگر نیست.

لم ۳. اگر در گره L کار i از مجموعه‌ی کارهای زمان‌بندی نشده برای چیدن در انتهای توالی σ انتخاب شود و کار i در این نوبت تأخیردار شود ($d_i < C_{i2}(\sigma i)$)، جواب بهینه‌ی وجود دارد که در آن کار i در آخر توالی قرار می‌گیرد.

اثبات: اگر زمان‌بندی i در زمان A موجب تأخیردار شدن آن شود در هر نوبتی بعد از آن هم تأخیردار خواهد بود. از طرفی واضح است که با انتقال کار تأخیردار به آخر توالی، زمان شروع بقیه کارها افزایش نمی‌یابد. بنابراین با انتقال این کار به آخر توالی تعداد کارهای تأخیردار افزایش نمی‌یابد. □

نتیجه ۲. در مسئله $P_i = a_i + bS, r_i | \sum_{i=1}^n U_i$ اگر رابطه‌ی $d_i < a_{i1} + a_{i2} + r_i$ برقرار باشد، این کار در هر نوبتی قرار گیرد حتماً تأخیردار است. بنابراین یک توالی بهینه وجود دارد که در آن کار i در مجموعه‌ی کارهای تأخیردار است.

لم ۴. اگر دو کار i و z در توالی $\sigma z i$ دیرکرد نداشته باشند و روابط ۱۹ و ۲۰ برقرار باشد، آنگاه جواب بهینه‌ی وجود دارد که در آن کار z بعد از کار i پردازش می‌شود.

$$C_1(\sigma i j) \leq C_1(\sigma j i) \quad (19)$$

$$C_2(\sigma i j) \leq C_2(\sigma j i) \quad (20)$$

اثبات: با توجه به این که دو کار i و z در توالی $\sigma z i$ دیرکرد ندارند رابطه‌ی ۲۱ برقرار است:

$$N_T(\sigma i j) = N_T(\sigma) \quad (21)$$

ولی در توالی جزئی $\sigma z i$ با توجه به تغییر زمان تکمیل کارها رابطه‌ی ۲۲ برقرار می‌شود:

$$N_T(\sigma) \leq N_T(\sigma i j) \leq N_T(\sigma) + 2 \quad (22)$$

با مقایسه‌ی روابط ۲۱ و ۲۲، و نیز با توجه به این که زمان تکمیل توالی $\sigma z i$ روی هر دو ماشین بیشتر از توالی $\sigma z i$ نیست، نتیجه می‌شود که توالی $\sigma i j$ بر توالی $\sigma z i$ غالب است. □

کامی مثل i زمان بندی شده باشد، برابر است با $\max\{C_{j1}, C_{i2}\} + a_{j2} + bS_{j2}$ که bS_{j2} برابر مقدار زوال واقعی مربوط به زمان شروع کار j روی ماشین ۲ است. در صورتی که به جای a_{j2} مقدار $\min_{i \in \sigma'}(a_{i2})$ و به جای مقدار زوال واقعی، مقدار زوال مربوط به زمان $C_{\sigma 2}$ (زمان تکمیل آخرین کار توالی جزئی σ روی ماشین ۲) یعنی مقدار $C_{\sigma 2}b$ را قرار دهیم، یک حد پایین برای زمان تکمیل کار j در نظر گرفته شده است.

از طرفی زمان شروع کار j روی ماشین ۲ برابر با $\max\{C_{j1}, C_{j2}\}$ است. حال اگر فرض کنیم ماشین ۲ به تعداد زیاد موجود است به گونه‌ای که هیچ‌یک از کارها برای شروع پردازش روی ماشین ۲ نباید در صف بماند و به محض تکمیل پردازش هر کار روی ماشین ۱، پردازش آن روی ماشین ۲ شروع می‌شود، در این صورت زمان شروع پردازش کار j روی ماشین ۲ برابر C_{j1} می‌شود که $C_{j1} \leq \max\{C_{j1}, C_{i2}\}$ به این ترتیب یک حد پایین برای شروع پردازش کار j روی ماشین ۲ در نظر گرفته شده است.

با فرضیات فوق، پس از تکمیل پردازش هر کار j از مجموعه σ' روی ماشین ۱ به اندازه $\min_{i \in \sigma'}(a_{i2}) + C_{\sigma 2}b$ طول می‌کشد تا پردازش آن کار روی ماشین ۲ هم تمام شود، یعنی $C_{j2} = C_{j1} + \min_{i \in \sigma'}(a_{i2}) + C_{\sigma 2}b$. برای این که کار j تأخیردار نشود باید $C_{j2} \leq d_j$ باشد، یعنی $C_{j1} + \min_{i \in \sigma'}(a_{i2}) + C_{\sigma 2}b \leq d_j$ ، به عبارت دیگر باید رابطه $C_{j1} \leq d_j - (\min_{i \in \sigma'}(a_{i2}) + C_{\sigma 2}b)$ برقرار باشد. بنابراین نتیجه می‌شود که با در نظر گرفتن فرضیات فوق و کم کردن مقدار $\min_{i \in \sigma'}(a_{i2}) + C_{\sigma 2}b$ از موعد تحویل تمام کارهای مجموعه σ' می‌توان مسئله را به یک مسئله تک ماشین که فقط ماشین شماره ۱ در آن وجود دارد تقلیل داد و به جای موعد تحویل هر یک از کارها مقدار d'_j را قرار داد که $d'_j = d_j - (\min_{i \in \sigma'}(a_{i2}) + C_{\sigma 2}b)$. حال فقط با در نظر گرفتن موعد تحویل‌های جدید (d'_j) می‌توان لم ۸ را برای به دست آوردن یک حد پایین مسئله به کار گرفت.

لم ۸. اگر $NT(\sigma'_{moore})$ براساس الگوریتم مور - هاجسون با توجه به زمان پردازش کارهای مجموعه σ' روی ماشین ۱ و با در نظر گرفتن مقدار زوال در زمان $A = C_{\sigma 1}$ برای همه کارهای مجموعه σ' و نیز با در نظر گرفتن مقادیر d'_j به عنوان موعد تحویل کارهای مجموعه σ' که $d'_j = d_j - (\min_{i \in \sigma'}(a_{i2}) + C_{\sigma 2}b)$ به دست آید، آنگاه یک حد پایین برای مسئله $PF2 | P_i = a_i + bS, r_i | \sum_{i=1}^n U_i$ به صورت رابطه ۳۰ به دست می‌آید:

$$LB_2 = NT(\sigma) + NT(\sigma'_{moore}) \quad (30)$$

اثبات: بدیهی است اگر به کارهای مجموعه σ' مقدار زوالی برابر با حالتی که پردازش کار در زمان A شروع شود نسبت دهیم، این مقدار از زوال واقعی آن کم‌تر خواهد بود. همچنین با توجه به این که الگوریتم مور - هاجسون تعداد کارهای تأخیردار را در مسئله پایه (تک ماشین بدون زوال) کمیته می‌کند. بنابراین با چشم‌پوشی از مقدار زوال واقعی کارها و در نظر گرفتن مقدار زوال در زمان A برای همه کارهای مجموعه σ' ، و نیز با توجه به این که از زمان عملیات کارها روی ماشین ۲ صرف نظر شده است، مطابق لم ۷ در این حالت می‌توان مقدار موعد تحویل هر یک از کارها را برابر با $d'_j = d_j - (\min_{i \in \sigma'}(a_{i2}) + C_{\sigma 2}b)$ در نظر گرفت و مطمئن شد که تعداد کارهای تأخیردار مجموعه σ' کم‌تر از $NT(\sigma'_{moore})$ نخواهد بود. □

لم ۹. با استدلالی مشابه لم ۸ و با در نظر گرفتن موعد تحویل واقعی کارها می‌توان ثابت کرد که اگر به کارهای مجموعه σ' مقدار زوال مربوط به زمان T_2 را نسبت

لم ۵. برای دو کار i و j که از بین کارهای مجموعه σ' کوچک‌ترین زمان‌های در دسترس را دارند، اگر شروط ۲۳ و ۲۴ برقرار باشد حتماً یک جواب بهینه وجود دارد که کار i قبل از کار j پردازش می‌شود.

$$r_i \geq 0, \quad r_j \geq \max(r_i + P_{i1}, A + P_{i1}) \quad (23)$$

$$\max(r_i, A) + P_{i1} + P_{i2} \leq r_j + P_{j1} \quad (24)$$

اثبات: هنگامی که کارهای مجموعه σ چیده شده باشد و دو کار دارای کوچک‌ترین زمان ورود بوده و روابط ۲۳ و ۲۴ را نیز برآورده کنند مسلماً کار i باید قبل از کار j انجام شود، چون انجام یا عدم انجام کار i در زمان شروع کار j تأثیری ندارد. بنابراین برای کاهش تعداد کارهای تأخیردار، کار i باید حتماً قبل از کار j انجام شود. زمان‌های تکمیل دو توالی جزئی ij و ji بعد از σ مطابق روابط ۲۵ و ۲۶ محاسبه می‌شود:

$$C_2(\sigma ij) = r_j + P_{j1} + P_{j2} \quad (25)$$

$$C_2(\sigma ji) = r_j + P_{j1} + \max(P_{i1}, P_{j2}) + P_{i2} \quad (26)$$

به سادگی دیده می‌شود که $C_2(\sigma ji) \geq C_2(\sigma ij)$ ، بنابراین کار i باید قبل از کار j انجام شود. □

۲.۲.۶. حد پایین

در هر گره، حد پایین تابع هدف از جمع تابع هدف توالی جزئی σ و تخمینی از مقدار تابع هدف برای مجموعه کارهای زمان‌بندی نشده σ' به دست می‌آید. برای محاسبه حد پایین براساس لم‌های زیر عمل می‌شود.

لم ۶. در مسئله $PF2 | P_i = a_i + bS, r_i | \sum_{i=1}^n U_i$ برای هر گره در روش شاخه و کران که متناظر با یک توالی جزئی σ است، یک حد پایین به صورت زیر به دست می‌آید:

$$LB_1 = NT(\sigma) + NT(\sigma') \quad (27)$$

در این رابطه مقدار $NT(\sigma)$ برابر تعداد کارهای تأخیردار در توالی جزئی چیده شده است. همچنین اگر تمام کارهای مجموعه σ' به صورت جداگانه در زمان A ، که در آن $A = C_1(\sigma)$ ، روی ماشین ۱ زمان‌بندی شوند $NT(\sigma')$ تعداد کارهای تأخیردار مجموعه σ' را در این حالت نشان می‌دهد و از رابطه ۲۸ به دست می‌آید.

$$NT(\sigma') = \sum_{i \in \sigma'} X(\sigma i) \quad (28)$$

$$X(\sigma i) = \begin{cases} 1 & \text{if } C_{i2}(\sigma i) > d_i \\ 0 & \text{if } C_{i2}(\sigma i) \leq d_i \end{cases} \quad (29)$$

اثبات: واضح است که توالی کارهای مجموعه σ' به هر ترتیبی که باشد تأثیری بر مقدار تابع هدف توالی جزئی σ ندارد. همچنین اگر زمان‌بندی هر یک از کارهای مجموعه σ' در زمان A روی ماشین ۱، موجب دیرکرد آن کار شود، در هر زمانی بعد از A نیز تأخیردار می‌شود. بنابراین تعداد کارهای تأخیردار مجموعه σ' هیچ‌گاه از مقدار $NT(\sigma')$ کم‌تر نخواهد بود. لذا مقدار تابع هدف هر توالی کامل که براساس توالی σ به دست آید از مقدار LB_1 کم‌تر نخواهد بود. □

لم ۷. زمان تکمیل هر کار j از مجموعه σ' روی ماشین ۲، در صورتی که قبل از آن

دهیم و آن‌ها را براساس الگوریتم مور- هاجسون و با شروع از زمان T_2 روی ماشین ۲ زمان بندی کنیم، مطمئن خواهیم بود که تعداد کارهای تأخیردار مجموعه‌ی کم‌تر از $N_T(\sigma'_{more\epsilon\tau})$ نخواهد بود. بنابراین:

$$LB_{\tau} = N_T(\sigma) + N_T(\sigma'_{more\epsilon\tau}) \quad (31)$$

$$T_{\tau} = \max\{\min_{i \in \sigma'}[\max(A_i, r_i) + P_i], C_{\sigma\tau}\} \quad (32)$$

لم ۱۰. براون و یاجیلی^[۱۴] ثابت کردند که در مسئله‌ی C_{max} $|P_i = a_i + bS|$ جواب بهینه زمانی به دست می‌آید که کارها به ترتیب غیر نزولی a_i زمان بندی شوند. بنابراین اگر این قاعده برای هر نوبت بعد از توالی جزئی σ به کار گرفته شود هم کم‌ترین زمان تکمیل برای آن نوبت به دست می‌آید و هم کم‌ترین زوال برای نوبت بعدی ایجاد می‌شود. بدین منظور می‌توان این قاعده را به صورت زیر برای جریان کارگاهی با دو ماشین توسعه داد و از آن برای محاسبه یک حد پایین برای مسئله استفاده کرد. بنابراین یک حد پایین برای مسئله $PFT_2 | P_i = a_i + bS, r_i | \sum_{i=1}^n U_i$ با انجام رویه زیر به دست می‌آید:

برای کارهای مجموعه‌ی σ' زمان پایه به منظور پردازش کارها روی ماشین ۱ (a_{i1}) را به ترتیب غیر نزولی مرتب کنید. همین کار را برای a_{i2} ها و برای r_i ها نیز انجام دهید. کوچک‌ترین a_{i2}, a_{i1} و r_i را به نوبت M که اولین نوبت بعد از توالی σ است نسبت دهید (زمان شروع عملیات نوبت M روی ماشین ۱ برابر با $T_1 = \max\{\min_{i \in \sigma'}(r_i), C_{\sigma 1}\}$ است). زمان تکمیل نوبت M روی ماشین ۲ را به دست آورید. سپس کوچک‌ترین موعد تحویل را از مجموعه‌ی $D_{\sigma'}$ انتخاب، و با زمان تکمیل به دست آمده مقایسه کنید. اگر موعد تحویل انتخاب شده باعث تأخیردار شدن نوبت M شد یک واحد به تعداد کارهای تأخیردار اضافه شده، آن موعد تحویل از مجموعه‌ی $D_{\sigma'}$ حذف می‌شود و کوچک‌ترین موعد تحویل بعدی با زمان تکمیل نوبت M مقایسه می‌شود. این کار ادامه می‌یابد تا زمانی که موعد تحویل انتخاب شده موجب تأخیردار شدن نوبت M نشود. در این حالت موعد تحویل مربوطه از مجموعه‌ی $D_{\sigma'}$ حذف می‌شود و زمان تکمیل نوبت بعدی محاسبه شده و با کوچک‌ترین موعد تحویل بعدی مقایسه می‌شود.

۳.۶ الگوریتم LB_{τ}

گام ۰. مجموعه‌های $\sigma' := \{j_1, j_2, \dots, j_i\}$ و $\sigma'_{[i]} := \{a'_{j_1}, a'_{j_2}, \dots, a'_{j_i}\}$ را به گونه‌ی $a'_{j_1} \leq a'_{j_2} \leq \dots \leq a'_{j_i}$ و $a'_{j_1} \leq a'_{j_2} \leq \dots \leq a'_{j_i}$ مرتب کنید. به همین ترتیب مجموعه‌های $A_{\sigma'1}, R_{\sigma'1}$ و $D_{\sigma'1}$ را تشکیل دهید، طوری که در آن‌ها زمان‌های پردازش روی ماشین دوم، زمان‌های در دسترس کارها و موعد تحویل کارهای مجموعه‌ی σ' به ترتیب غیر نزولی مرتب شده باشند. همچنین $k := 1$ و $NT(\sigma') := 0$ قرار دهید.

گام ۱. کوچک‌ترین a'_{i1}, a'_{i2} و r'_i را انتخاب کنید. نوبت M را در نظر گرفته و قرار دهید $a_{M1} := a'_{i1}, a_{M2} := a'_{i2}, r_M := r'_i$ و در زمان SM_1 روی ماشین ۱ و SM_2 روی ماشین ۲ براساس روابط $SM_1 := \max\{r_M, C_{\sigma 1}\}$ و $SM_2 := \max\{C_{M1}, C_{\sigma 2}\}$ زمان بندی کنید. زمان تکمیل نوبت M روی ماشین ۲ را حساب کرده و $A_{\sigma'1} := A_{\sigma'1} - a'_{i1}, A_{\sigma'2} := A_{\sigma'2} - a'_{i2}$ و $R_{\sigma'} := R_{\sigma'} - r'_i$ قرار دهید.

گام ۲. اگر $C_{M2} \leq d_{[k]}$ آنگاه قرار دهید $k := k + 1$ و به گام ۳ بروید. در غیر

این صورت $NT(\sigma') := NT(\sigma') + 1$ و $k := k + 1$ و این گام را تکرار کنید.

گام ۳. اگر $k \leq i$ ، آنگاه قرار دهید $C_{M1} := C_{M1} + 1$ و $C_{\sigma 2} := C_{\sigma 2} + 1$ و به گام ۱ بروید. در غیر این صورت $NT(\sigma')$ به دست آمده است.

پس از محاسبه‌ی $NT(\sigma')$ ، مقدار LB_{τ} از رابطه‌ی ۳۳ به دست می‌آید:

$$LB_{\tau} = N_T(\sigma) + N_T(\sigma') \quad (33)$$

اثبات: چنان که پیش‌تر بیان شد مبنای محاسبه‌ی $NT(\sigma')$ در الگوریتم LB_{τ} براساس ایجاد کوچک‌ترین زمان تکمیل برای هر نوبت و مقایسه‌ی آن با بهترین موعد تحویل موجود است. با توجه به این که در مسئله $|P_i = a_i + bS|$ توالی به دست آمده از ترتیب غیر نزولی a_i ها توالی بهینه است، اگر این قاعده به نحوی که قبل از الگوریتم LB_{τ} توضیح داده شد برای هر نوبت بعد از توالی جزئی σ به کار گرفته شود کم‌ترین زمان تکمیل برای آن نوبت به دست می‌آید. حال اگر کوچک‌ترین موعد تحویل موجود را با زمان تکمیل نوبت M مقایسه کنیم و این نوبت تأخیردار شود، چون نوبت M کوچک‌ترین زمان تکمیل ممکن را دارد، می‌توان نتیجه گرفت که این موعد تحویل موجب تأخیردار شدن نوبت‌های بعدی نیز می‌شود. بنابراین تعداد کارهای تأخیردار برابر با تعداد موعد تحویل‌هایی است که موجب تأخیردار شدن نوبت M می‌شوند. چون نوبت M کم‌ترین زمان تکمیل ممکن را دارد مطمئن خواهیم بود که تعداد کارهای تأخیردار مجموعه‌ی σ' کم‌تر از $NT(\sigma')$ نخواهد بود. از طرفی بدیهی است که تعداد کارهای تأخیردار مجموعه‌ی σ' تأثیری بر تعداد کارهای تأخیردار توالی جزئی σ نخواهد داشت. بنابراین مقدار تابع هدف هر توالی کامل از مقدار LB_{τ} کم‌تر نخواهد بود. □

پس از محاسبه‌ی مقادیر LB_1, LB_2, LB_3, LB_4 ، حد پایین هر گره در روش شاخه و کران از رابطه‌ی ۳۴ به دست می‌آید:

$$LB^* = \max\{LB_1, LB_2, LB_3, LB_4\} \quad (34)$$

۱.۳.۶. حد بالا

در روش شاخه و کران ارائه شده ابتدا الگوریتم‌های ابتکاری H_1 تا H_4 اجرا می‌شود و کم‌ترین جواب حاصل از آن‌ها به عنوان حد بالای الگوریتم شاخه و کران در نظر گرفته می‌شود.

$$UB = \min\{NT_{H1}, NT_{H2}, NT_{H3}, NT_{H4}, NT_{BS}\} \quad (35)$$

NT_{BS} برابر است با بهترین جوابی که تاکنون از یک توالی کامل به دست آمده در روش شاخه و کران پس‌رو به دست آمده است.

۲.۳.۶. ترتیب اجرای لم‌ها و حدود پایین در الگوریتم شاخه و کران

عملکرد هر یک از لم‌ها، نتایج و حدود پایین معرفی شده با توجه به زمان استفاده از آن‌ها در روش شاخه و کران را می‌توان تغییر داد.^[۹] به عنوان مثال بعضی از نتایج لم‌ها ممکن است در ابتدای الگوریتم شاخه و کران عملکرد بهتری داشته باشند. بنابراین با توجه به نحوه عملکرد هر یک از اصول غلبه، نتایج و حدود پایین، ترتیب اجرای آن‌ها براساس جدول ۱ در نظر گرفته شده است.

- نتیجه‌ی ۲، لم ۲ و لم ۳ این توانایی را دارند که به سرعت یک بعد یا بیشتر، فضای جواب را کاهش دهند.
- لم ۱ یک بعد یا بیشتر، فضای جواب را کاهش می‌دهد و اجرای آن پس از لم ۳ مناسب‌تر است.

جدول ۱. ترتیب اجرای لم‌ها، نتایج و حدود پایین در الگوریتم شاخه و کران.

ترتیب اجرا	لم، نتیجه، حد پایین	ویژگی لم، نتیجه، حد پایین	پارامتر تأثیرگذار	دسته‌ی بالا	دسته‌ی پایین
۱	نتیجه ۲	در صورت اجرا یک بعد از فضای جواب کم می‌شود.		کارها در یک بازه زمانی	کارها در یک بازه زمانی کوچک
۲	لم ۲	در صورت اجرا یک بعد از فضای جواب کم می‌شود.	r_i	بزرگ در دسترس قرار می‌گیرند	و نزدیک به زمان شروع برنامه‌ریزی شده در دسترس قرار می‌گیرند
۳	لم ۳	اضافه کردن کار i به مجموعه کارهای تأخیردار و انتقال آن به انتهای توالی در صورتی که زمان‌بندی کار i بلافاصله پس از توالی σ باعث تأخیردار شدن آن شود.	d_i	بازه زمان‌های تحویل باز	بازه زمان‌های تحویل فشرده
۴	لم ۱	انتقال کار i بدون دیرکرد به انتهای توالی.		زمان‌های تحویل متفاوت	زمان‌های تحویل مشابه
۵	لم ۶	اجرای حد پایین ۱ که زمان‌بندی کلیه کارهای مجموعه‌ی σ' و شمارش تعداد کارهای تأخیردار را شامل می‌شود.		تعداد کارهای تأخیردار بیشتر	تعداد کارهای تأخیردار کم‌تر
۶	لم ۸	اجرای حد پایین ۲ که زمان‌بندی تمامی کارهای مجموعه‌ی σ' براساس الگوریتم مور-هاجسون روی ماشین ۱ و با فرض حداقل زوال برای تمام کارها را شامل می‌شود.			
۷	لم ۹	اجرای حد پایین ۳ که زمان‌بندی کلیه کارهای مجموعه‌ی σ' براساس الگوریتم مور-هاجسون روی ماشین ۲ و با فرض حداقل زوال برای تمام کارها را شامل می‌شود.			
۸	لم ۱۰	اجرای حد پایین ۴.			
۹	لم ۵	بررسی دو کار مجاور i و j .			
۱۰	لم ۴	بررسی دو کار مجاور i و j .			

• پس از آن لم‌های مربوط به حد پایین و سپس به ترتیب لم ۵ و لم ۴ که در حذف گره‌های میانی توانایی خوبی دارند، اجرا می‌شود.

۷. نتایج محاسباتی

الگوریتم شاخه و کران و روش ابتکاری با زبان برنامه نویسی ++C کد نویسی شده و روی رایانه‌یی با مشخصات Intel Core i3، سرعت ۲٫۱ گیگاهرتز و حافظه ۲ گیگابایت اجرا شده‌اند. برای بررسی کارایی روش ارائه شده مسائلی به صورت تصادفی و با استفاده از روش اشاره شده [۱۵] تولید شده‌اند. زمان‌های پردازش پایه از توزیع یکنواخت گسسته در بازه $[1, 10]$ تولید شده‌اند. مقادیر موعده تحویل از توزیع یکنواخت گسسته در بازه $(P\alpha, P\beta)$ به دست می‌آید که مقدار P از رابطه‌ی ۳۶ به دست می‌آید.

$$P = \frac{1}{n} * [\sum_{i=1}^n \sum_{j=1}^2 a_{ij} + (n-1) \max\{\sum_{i=1}^n a_{i1}, \sum_{i=1}^n a_{i2}\}] \quad (36)$$

مقادیر α و β که به ترتیب حد پایین و حد بالای موعدهای تحویل را نشان می‌دهند از دو مجموعه‌ی $\alpha \in \{0.4, 0.6\}$ و $\beta \in \{0.8, 1.2\}$ انتخاب می‌شود. زمان‌های در دسترس بودن یعنی r_i از توزیع یکنواخت گسسته $(P\delta, P\varepsilon)$ تولید شده که برای δ مقدار صفر و برای ε مقادیر $\{0.5, 0.25\}$ در نظر گرفته شده‌اند. برای داده‌ها دو دسته‌ی پایین ۵ و بالا ۶ در نظر گرفته شده است. پارامترهای دسته‌ی

جدول ۲. تفاوت‌های بین دو دسته مسئله‌ی تولید شده.

پارامتر تأثیرگذار	دسته‌ی بالا	دسته‌ی پایین
	کارها در یک بازه زمانی	کارها در یک بازه زمانی کوچک
r_i	بزرگ در دسترس قرار می‌گیرند	و نزدیک به زمان شروع برنامه‌ریزی شده در دسترس قرار می‌گیرند
d_i	بازه زمان‌های تحویل باز	بازه زمان‌های تحویل فشرده
d_i	زمان‌های تحویل متفاوت	زمان‌های تحویل مشابه
تعداد کارهای تأخیردار	تعداد کارهای تأخیردار بیشتر	تعداد کارهای تأخیردار کم‌تر

پایین به صورت $\{\alpha = 0.6, \beta = 0.8, \varepsilon = 0.5\}$ و دسته‌ی بالا به صورت $\{\alpha = 0.4, \beta = 1.2, \varepsilon = 0.25\}$ تولید شده‌اند. [۱۵] مسائل دسته‌ی بالا دارای شرایط بسیار پویاتری نسبت به مسائل دسته‌ی پایین هستند. به عبارت دیگر مسائل دسته‌ی پایین مسائل سخت‌تری نسبت به مسائل دسته‌ی بالا هستند. جدول ۲ تفاوت‌های بین دو دسته مسئله را نشان می‌دهد.

برای نرخ زوال مقادیر $0.5, 0.1$ و 0.25 در نظر گرفته شده است. بنابراین با توجه به مقادیر زوال، برای هر یک از دسته‌های پایین و بالا سه گروه داده تولید شده است. مقدار نرخ زوال برای گروه Low ۱ و High ۱ برابر 0.5 ، برای Low ۲ و High ۲ برابر 0.1 و برای Low ۳ و High ۳ برابر 0.25 است.

برای بررسی کارایی روش‌های ارائه شده مسائلی با تعداد کار ۸، ۱۲، ۱۶، ۲۰، ۲۴، ۲۶ و ۲۸ و از هر مسئله ۵ نمونه تولید شده است. پارامترهای مورد نیاز برای تولید هر مسئله با استفاده از تکنیک تولید اعداد تصادفی و از بین مقادیر و بازه‌هایی که در پاراگراف قبل توضیح داده شد تولید شده‌اند. بنابراین در مجموع ۲۴۰ مسئله‌ی نمونه تولید و حل شده است. برای هر مسئله محدودیت زمانی برابر 3600 ثانیه در نظر گرفته شده است. اگر در این مدت زمان جواب بهینه توسط روش شاخه و کران به دست نیاید روند اجرای الگوریتم شاخه و کران متوقف می‌شود. نتایج حاصل از حل هر یک از گروه‌ها در جداول ۳ و ۴ نشان داده شده است.

چنان‌که مشاهده می‌شود، الگوریتم شاخه و کران پیشنهادی قادر است مسائلی را تا اندازه ۲۴ کار در دسته‌ی بالا و ۲۲ کار در دسته‌ی پایین به طور کامل و در زمان منطقی حل کند. در بعضی گروه‌ها نظیر High ۲، High ۳ و Low ۳ مسائل با ابعاد بزرگ‌تر نیز حل شده است. با توجه به این که مسئله‌ی کمیته کردن تعداد کارهای تأخیردار با فرض زوال روی جریان کارگاهی دو ماشین تاکنون با هیچ فرضی با روش‌های بهینه حل نشده، برای مقایسه می‌توان به مطالعات مشابه انجام شده اشاره کرد. چنان‌که در جدول ۵ نشان داده شده با استفاده از الگوریتم‌های پیشنهادی، در سایر مطالعات مسائلی با ابعاد ۱۴، ۱۸، ۲۰ و ۲۸ فعالیت حل شده است. با توجه به این که همه‌ی آن‌ها مسائلی با فرض‌های ساده‌تر را حل کرده‌اند بنابراین می‌توان بیان کرد که الگوریتم شاخه و کران ارائه شده از توانایی خوبی در حل بهینه‌ی مسائل برخوردار است.

متوسط زمان حل الگوریتم شاخه و کران نیز حداکثر برابر 1620 ثانیه است که در مقایسه با سایر تحقیقات الگوریتم در زمان خیلی خوبی مسائل را حل می‌کند. همچنین متوسط درصد گره‌های قطع شده توسط مجموعه‌ی لم‌ها و حدود پایین در همه مسائل درصد بالایی است و کم‌ترین مقدار آن برابر 81.31 درصد است که عملکرد خوب لم‌ها و حدود پایین را در مقایسه با سایر تحقیقات [۹] نشان می‌دهد.

جدول ۳. نتایج حل برای الگوریتم شاخه و کران و ابتکاری برای گروه High ۱.

گروه داده	تعداد کار	تعداد نمونه‌های حل نشده توسط B&B	متوسط زمان حل B&B (ثانیه)	متوسط درصد گروه‌های قطع شده توسط										متوسط زمان حل B&B (ثانیه)	تعداد جواب بهینه به ابتکاری	نسبت تعداد جواب متوسط درصد
				نتیجه ۲	لم ۲	لم ۳	لم ۴	حد پایین ۱	حد پایین ۲	حد پایین ۳	حد پایین ۴	لم ۵	لم ۴			
High ۱	۸	۰	۰٫۱۹	۰٫۸۱	۰٫۸۵	۰٫۵۰	۴٫۸۸	۸٫۶۷	۳٫۸۲	۶٫۳۳	۶۵٫۸۲	۶٫۱۶	۲٫۱۵	۹۳٫۷۲	۱٫۱۱	۱٫۲۶
	۱۲	۰	۰٫۸۲	۱٫۶۰	۳٫۱۷	۰٫۲۲	۱۰٫۸۳	۱۲٫۳۰	۱٫۵۹	۱٫۵۹	۵۵٫۹۲	۵٫۰۶	۹٫۳۲	۹۲٫۶۳	۱٫۱۳	۱٫۲۹
	۱۶	۰	۳٫۱۴۸	۲٫۳۲	۴٫۴۹	۱٫۸۳	۴٫۱۵	۶٫۹۵	۹٫۰۰	۱۶٫۲۶	۵۳٫۶۶	۰٫۴۷	۰٫۸۷	۹۲٫۳۷	۱٫۱۴	۱٫۲۶
	۲۰	۰	۴۲٫۵۰۶	۳٫۳۳	۸٫۲۰	۲٫۲۰	۵٫۵۴	۹٫۳۵	۷٫۴۴	۱۰٫۱۶	۴۰٫۶۱	۸٫۵۱	۴٫۶۶	۹۶٫۴۱	۱٫۰۶	۱٫۲۹
	۲۲	۰	۲۵۲٫۸۱۳	۱٫۷۳	۱۱٫۳۷	۱٫۳۴	۱۳٫۱۲	۴٫۶۰	۴٫۵۱	۵۹٫۰۸	۱٫۴۹	۲٫۷۶	۹۴٫۰۴	۹۴٫۰۴	۱٫۱۱	۱٫۲۲
	۲۴	۰	۶۰۶٫۴۶۵	۴٫۳۸	۷٫۸۳	۶٫۱۱	۱۳٫۱۵	۴٫۴۶	۶٫۶۱	۳۵٫۸۷	۷٫۵۶	۱۴٫۰۴	۹۴٫۳۴	۹۴٫۳۴	۱٫۱۴	۱٫۳۶
۲۶	۴	۱۱۹۶٫۱۹۲	۴٫۵۹	۸٫۱۶	۱٫۲۵	۷٫۰۸	۱۶٫۳۰	۶٫۷۴	۳۶٫۱۹	۶٫۸۹	۱۲٫۸۰	۹۴٫۷۱	۹۴٫۷۱	۱٫۱۰	۱٫۱۸	
High ۲	۸	۰	۰	۱٫۰۳	۲٫۱۴	۱٫۳۲	۴٫۵۶	۶٫۳۷	۸٫۵۹	۷٫۷۰	۶۸٫۱۸	۰٫۰۴	۰٫۰۷	۹۱٫۷۷	۱٫۱۲	۱٫۲۹
	۱۲	۰	۰٫۳۳	۰٫۱۵	۲٫۸۶	۲٫۵۷	۲٫۳۸	۹٫۳۵	۱۲٫۳۹	۴٫۸۹	۶۱٫۱۸	۱٫۴۸	۲٫۷۵	۹۲٫۹۴	۱٫۰۴	۱٫۲۱
	۱۶	۰	۱٫۷۰۲	۰٫۸۰	۳٫۷۶	۵٫۸۰	۳٫۳۱	۳٫۳۱	۲۰٫۱۸	۱۳٫۴۸	۴۸٫۹۹	۱٫۲۹	۲٫۳۹	۹۳٫۶۸	۱٫۰۸	۱٫۱۴
	۲۰	۰	۲۶٫۰۳۱	۰٫۱۵	۵٫۰۷	۴٫۹۹	۹٫۵۴	۷٫۲۴	۱۹٫۰۳	۴۱٫۵۰	۴٫۳۷	۸٫۱۱	۹۳٫۷۵	۹۳٫۷۵	۱٫۰۹	۱٫۱۷
	۲۲	۰	۱۶۸٫۳۰۵	۰٫۳۲	۶٫۷۸	۳٫۱۰	۶٫۸۴	۱۱٫۶۲	۱۵٫۲۹	۵٫۶۵	۴۹٫۷۸	۰٫۲۲	۰٫۴۱	۹۴٫۶۷	۱٫۱۰	۱٫۱۶
	۲۴	۰	۳۰۶٫۸۸۱	۰٫۱۰	۵٫۴۱	۵٫۵۰	۱۰٫۰۸	۶٫۲۵	۱۰٫۸۱	۳۷٫۹۱	۸٫۳۸	۱۵٫۵۶	۹۵٫۰۴	۹۵٫۰۴	۱٫۱۱	۱٫۲۶
۲۶	۱	۷۳۲٫۴۸۸	۰٫۵۷	۲۲٫۱۷	۵٫۰۱	۱۶٫۷۱	۱۰٫۱۹	۷٫۰۸	۲۲٫۹۳	۵٫۳۷	۹٫۹۷	۹۵٫۲۶	۹۵٫۲۶	۱٫۱۰	۱٫۳۳	
۲۸	۷	۹۳۱٫۴۳۵	۲٫۰۵	۷٫۰۲	۴٫۸۵	۱۲٫۵۰	۶٫۱۶	۱۸٫۷۱	۳۹٫۰۹	۳٫۳۷	۶٫۲۶	۹۵٫۱۷	۹۵٫۱۷	۱٫۰۹	۱٫۰۹	
High ۳	۸	۰	۰	۳٫۴۰	۲٫۲۵	۰٫۰۰	۶٫۴۹	۱۲٫۱۲	۱٫۱۱	۶۹٫۸۹	۱٫۶۶	۳٫۰۸	۹۳٫۲۵	۱٫۱۰	۱٫۲۶	
	۱۲	۰	۰٫۲۲	۵٫۳۹	۴٫۵۶	۰٫۰۰	۵٫۳۴	۶٫۵۳	۲٫۶۳	۶۵٫۲۹	۳٫۵۹	۶٫۶۷	۹۸٫۰۹	۱٫۰۳	۱٫۲۸	
	۱۶	۰	۱٫۰۵۷	۴٫۵۵	۵٫۵۷	۱٫۰۷	۱۰٫۷۶	۴٫۸۰	۳٫۵۰	۵۸٫۰۹	۱٫۴۰	۳٫۳۲	۹۵٫۳۶	۱٫۱۱	۱٫۲۵	
	۲۰	۰	۷٫۴۹۹	۶٫۲۰	۵٫۷۹	۴٫۳۱	۷٫۰۱	۳٫۶۵	۲٫۶۴	۶۷٫۲۲	۱٫۱۱	۲٫۰۶	۹۵٫۲۵	۱٫۰۸	۱٫۱۵	
	۲۲	۰	۷۰٫۴۱۸	۳٫۶۷	۵٫۶۴	۶٫۰۷	۹٫۴۷	۴٫۳۳	۱٫۸۵	۴۷٫۳۹	۷٫۵۵	۱۴٫۰۳	۹۶٫۷۰	۱٫۰۶	۱٫۲۲	
	۲۴	۰	۳۷۲٫۴۹۰	۰٫۰۰	۶٫۰۲	۳٫۵۵	۶٫۶۴	۰٫۲۹	۰٫۴۷	۷۶٫۳۱	۲٫۳۵	۴٫۳۶	۹۰٫۱۲	۱٫۰۸	۱٫۱۸	
۲۶	۰	۵۱۰٫۴۰۹	۰٫۳۷	۱۰٫۷۶	۱٫۴۲	۸٫۷۳	۱٫۹۸	۲٫۲۰	۵۰٫۷۵	۵٫۷۸	۱۱٫۰۳	۹۶٫۶۶	۹۶٫۶۶	۱٫۰۵	۱٫۱۵	
۲۸	۲	۴۸۷٫۴۱۱	۳٫۳۴	۱۰٫۲۸	۳٫۳۱	۶٫۲۸	۹٫۸۰	۴٫۰۷	۵۵٫۳۵	۴٫۱۵	۳٫۴۱	۹۵٫۶۶	۹۵٫۶۶	۱٫۰۹	۱٫۳۲	

نتایج حل نشان می‌دهد که عملکرد الگوریتم شاخه و کران برای گروه‌های مختلف داده‌ها متفاوت است. در گروه‌های Low ۱، Low ۲ و Low ۳ با توجه به این که بازه موعدهای تحویل (d_i) و زمان‌های در دسترس (r_i) بازه کوچک‌تری است مسائل سخت‌ترند. تعداد مسائل حل نشده در محدوده‌ی زمانی ۳۶۰۰ ثانیه با ابعاد ۲۶ کار در این گروه‌ها به ترتیب برابر ۱۵، ۱۳ و ۶ است در حالی که در گروه‌های High ۱، High ۲ و High ۳ به ترتیب ۴، صفر و صفر است که نشان‌گر سخت‌تر بودن مسائل گروه‌های پایین است. همچنین با مقایسه متوسط زمان حل مسائلی که به جواب بهینه رسیده‌اند مشخص می‌شود که عموماً زمان حل مسائل گروه‌های Low بیشتر از مسائل گروه‌های High است. زمان حل مسائل گروه‌های مختلف داده، و همچنین تعداد مسائل حل نشده هر گروه، به ترتیب در شکل ۱ و ۲ با هم مقایسه شده است.

همچنین می‌توان از نتایج حل به این نکته رسید که الگوریتم برای نرخ زوال بزرگ‌تر عملکرد بهتری دارد به گونه‌ی که برای گروه‌های High ۳ و Low ۳ که نرخ زوال برابر ۰٫۲۵ است. تعداد نمونه‌هایی که در محدوده زمانی در نظر گرفته شده به

جواب بهینه نرسیده‌اند برای مسائل با ابعاد ۲۶ کار کم‌تر است از گروه‌های High ۱ و Low ۱ که نرخ زوال آن‌ها برابر ۰٫۰۵ است. همچنین مقادیر متوسط زمان حل برای مسائلی که به جواب بهینه رسیده‌اند در گروه‌های High ۳ و Low ۳ عموماً کم‌تر از گروه‌های High ۱ و Low ۱ است. بولفین و حلاه^[۱۶] برای بررسی عملکرد الگوریتم ابتکاری از شاخص Z^{opt}/Z^{heu} استفاده کرده‌اند که در آن Z^{opt} و Z^{heu} به ترتیب جواب بهینه حاصل از الگوریتم شاخه و کران و بهترین جواب به دست آمده از الگوریتم‌های ابتکاری با تابع هدف $\sum(1 - U_i)$ است. مقدار متوسط این شاخص برای هر یک از گروه داده‌ها در جداول ۳ و ۴ نشان داده شده است. شاخص Z^{opt}/Z^{heu} همیشه بزرگ‌تر یا مساوی ۱ است و هر چه به عدد ۱ نزدیک‌تر باشد کارایی الگوریتم بالاتر است. مقدار متوسط شاخص Z^{opt}/Z^{heu} حداکثر برابر ۱٫۱۵ است. در سایر تحقیقات مربوط به تعداد کارهای تأخیردار متوسط این شاخص معمولاً مقداری نزدیک به همین عدد یا قدری بزرگ‌تر دارد که تا ۱٫۴۳ نیز می‌رسد.^[۱۷] در جدول ۶ مقدار این شاخص در تحقیقات مختلف با هم مقایسه شده است که با توجه به این که مسائلی با فرض‌های ساده‌تر را حل کرده‌اند می‌توان

جدول ۴. نتایج حل برای الگوریتم شاخه و کران و ابتکاری برای گروه Low۱.

گروه داده	تعداد کار	تعداد نمونه‌های حل نشده توسط B&B	متوسط زمان حل B&B (ثانیه)	متوسط درصد گره‌های قطع شده توسط										نسبت تعداد جواب بهینه به ابتکاری		
				نتیجه ۲	لم ۲	لم ۳	لم ۱	حد پایین ۱	حد پایین ۲	حد پایین ۳	حد پایین ۴	لم ۴	لم ۵			
Low۱	۸	۰	۰,۰۳۹	۴,۵۰	۰,۶۱	۲,۴۳	۰,۰۰	۴,۴۵	۷,۷۹	۱۱,۷۹	۵۳,۵۲	۶,۶۷	۸,۲۴	۹۲,۶۱	۱,۱۵	۱,۳۷
	۱۲	۰	۰,۵۹۱	۴,۱۸	۰,۸۲	۱۱,۴۰	۰,۰۰	۵,۸۳	۱۰,۷۷	۸,۱۴	۴۸,۱۱	۸,۹۶	۱,۷۹	۸۴,۳۶	۱,۰۹	۱,۳۰
	۱۶	۰	۷,۳۹۱	۷,۲۲	۰,۴۸	۱۰,۱۵	۴,۴۹	۶,۸۳	۷,۸۳	۶,۸۵	۴۸,۹۶	۱,۷۱	۶,۱۷	۸۹,۷۷	۱,۱۵	۱,۴۱
	۲۰	۰	۹۰,۰۹۱	۷,۳۶	۱,۳۰	۱۷,۰۸	۵,۶۶	۳,۲۲	۱۲,۷۴	۵۱,۱۰	۰,۵۴	۱,۰۰	۸۱,۳۱	۸۱,۳۱	۱,۱۳	۱,۳۰
	۲۲	۰	۸۸۰,۵۵۸	۹,۸۶	۷,۶۰	۱۸,۰۵	۰,۰۰	۶,۴۲	۱,۳۶	۴۴,۷۲	۸,۱۰	۳,۸۹	۳,۸۹	۸۸,۶۱	۱,۱۳	۱,۴۱
	۲۴	۰	۹۹۷,۶۴۴	۵,۷۵	۴,۵۲	۲۱,۱۳	۴,۰۶	۲,۲۹	۱,۰۹	۳۴,۸۶	۷,۰۷	۱۵,۱۳	۱۵,۱۳	۹۱,۵۲	۱,۱۴	۱,۵۱
	۲۶	۱۵	۱۶۲۰,۸۰۷	۰,۰۰	۱,۰۸	۴,۱۴	۴,۸۲	۴,۱۵	۱,۴۱	۱۹,۶۵	۴,۱۴	۷,۶۸	۷,۶۸	۹۰,۲۸	۱,۰۹	۱,۳۲
Low۲	۸	۰	۰,۰۱۶	۱,۱۷	۰,۰۰	۵,۹۸	۰,۰۰	۱,۱۰	۱۲,۴۳	۱۶,۷۷	۵۹,۳۱	۱,۱۴	۲,۱۱	۸۸,۳۹	۱,۱۵	۱,۲۷
	۱۲	۰	۰,۲۰۳	۲,۹۵	۲,۰۰	۱,۹۲	۰,۰۰	۵,۰۸	۸,۶۷	۱۱,۹۳	۵۷,۳۶	۲,۱۳	۷,۹۶	۹۰,۳۳	۱,۰۷	۱,۲۹
	۱۶	۰	۲,۷۳۹	۴,۱۶	۱,۶۶	۲۳,۰۰	۰,۰۰	۳,۱۳	۱۲,۹۳	۷,۲۰	۴۰,۱۷	۲,۷۱	۵,۰۴	۹۱,۱۹	۱,۱۴	۱,۵۰
	۲۰	۰	۱۷,۵۳۹	۴,۲۰	۲,۸۰	۱۸,۳۱	۰,۰۰	۶,۰۳	۱,۵۳	۱۹,۸۳	۳,۶۱	۳,۶۱	۶,۷۱	۸۲,۱۲	۱,۱۲	۱,۳۹
	۲۲	۰	۴۳۴,۹۴۷	۳,۴۴	۳,۹۶	۱۷,۳۱	۰,۰۰	۵,۰۱	۱,۷۹	۱۳,۶۲	۵۰,۹۱	۱,۳۹	۲,۵۷	۹۲,۶۶	۱,۱۲	۱,۳۵
	۲۴	۳	۱۰۹۸,۵۶۸	۲,۹۳	۵,۸۸	۱۹,۴۵	۰,۰۰	۰,۰۰	۰,۵۵	۲,۲۳	۵۵,۱۶	۴,۷۳	۹,۰۷	۸۱,۸۸	۱,۱۱	۱,۴۲
	۲۶	۱۳	۱۴۴۲,۸۹۷	۲,۴۸	۶,۵۶	۳,۱۲	۰,۰۰	۱۹,۱۲	۱۰,۲۵	۱۲,۵۰	۴۲,۸۷	۱,۰۹	۲,۰۲	۸۹,۳۸	۱,۱۴	۱,۱۹
Low۳	۸	۰	۰	۲,۴۶	۰,۵۶	۸,۱۵	۰,۰۰	۷,۴۳	۲,۱۴	۱۰,۳۱	۵۱,۲۹	۶,۱۸	۱۱,۴۸	۹۱,۸۴	۱,۱۳	۱,۳۶
	۱۲	۰	۰,۳۵۱	۳,۳۸	۰,۹۷	۸,۱۱	۴,۷۸	۴,۶۶	۰,۴۸	۷,۷۷	۶۸,۵۶	۰,۴۵	۰,۸۴	۹۰,۴۷	۱,۰۸	۱,۲۴
	۱۶	۰	۳,۵۴۲	۸,۰۴	۱,۱۰	۳,۳۱	۰,۰۰	۵,۴۳	۱,۰۷	۶,۸۹	۵۷,۹۵	۶,۱۷	۱۰,۰۳	۹۲,۲۷	۱,۰۸	۱,۳۷
	۲۰	۰	۶۲,۶۰۴	۶,۴۶	۴,۱۳	۳,۶۹	۰,۰۰	۶,۰۱	۱,۳۳	۳,۳۱	۶۱,۵۰	۴,۷۵	۸,۸۳	۸۷,۶۴	۱,۱۲	۱,۳۶
	۲۲	۰	۲۶۳,۴۵۷	۷,۰۵	۱,۵۹	۷,۲۶	۰,۰۰	۵,۰۱	۱,۷۴	۱۰,۵۴	۵۸,۲۵	۲,۱۰	۶,۴۶	۹۳,۴۹	۱,۱۱	۱,۴۰
	۲۴	۰	۶۱۲,۳۵۸	۹,۴۸	۵,۴۹	۳,۷۷	۰,۰۰	۲,۰۲	۰,۹۳	۶,۶۲	۵۳,۵۵	۶,۳۵	۱۱,۷۹	۹۳,۶۹	۱,۱۰	۱,۳۸
	۲۶	۶	۸۷۰,۸۸۷	۶,۷۹	۴,۱۷	۵,۲۶	۵,۱۳	۵,۰۰	۱,۲۷	۳,۰۸	۵۰,۰۸	۷,۱۳	۱۲,۱۰	۸۷,۲۰	۱,۱۲	۱,۴۰

جدول ۵. مقایسه‌ی ابعاد مسائل حل شده به صورت بهینه توسط الگوریتم شاخه و کران در مقالات مختلف.

شماره مرجع	سال	پردازشگر	تابع هدف	محیط ماشینی	شکل تابع زوال	فرض ورود غیرهمزمان	ابعاد مسائل حل بهینه
[۷]	۲۰۱۱	Intel Core۲, ۲,۱۳ GHz	$\sum C_i$	PF۲	Simple linear	—	۱۸
[۹]	۲۰۱۲	Pentium۴, ۲,۴ GHz	$\sum U_i$	۱	Simple linear	—	۲۸
[۲]	۲۰۱۳	Intel Core۲, ۲,۱۳ GHz	C_{max}	۱	Simple linear	r_i	۲۰
[۸]	۲۰۱۴	Intel Core۲ Quad CPU, ۲,۶۶ GHz	$\sum T_i$	F۲	Simple linear	—	۲۰
مقاله حاضر	—	Intel Core i۳, ۲,۱ GHz	$\sum U_i$	PF۲	Simple linear	r_i	۲۴

جدول ۶. مقایسه‌ی شاخص نسبت تعداد جواب بهینه به ابتکاری در تحقیقات مختلف.

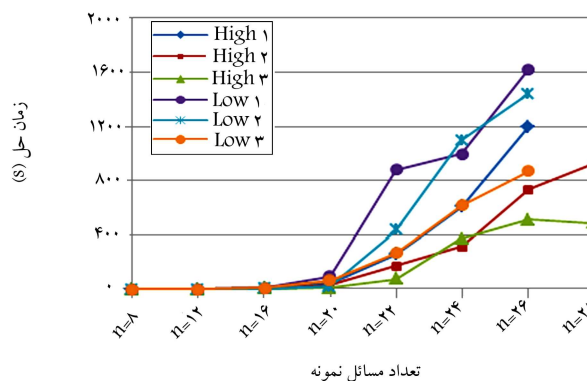
شماره مرجع	سال	پردازشگر	تابع هدف	محیط ماشینی	شکل تابع زوال	فرض ورود غیرهمزمان کارها	نسبت تعداد جواب بهینه به ابتکاری
[۵]	۲۰۰۸	Pentium۴, ۲,۴ GHz	C_{max}	F۲	Simple linear	—	۱,۲۰
[۷]	۲۰۱۱	Intel Core۲, ۲,۱۳ GHz	$\sum C_i$	PF۲	Simple linear	—	۱,۱۸
[۹]	۲۰۱۲	Pentium۴, ۲,۴ GHz	$\sum U_i$	۱	Simple linear	—	۱,۱۱
[۲]	۲۰۱۳	Intel Core۲, ۲,۱۳ GHz	C_{max}	۱	Simple linear	r_i	۱,۱۹
مقاله حاضر	—	Intel Core i۳, ۲,۱ GHz	$\sum U_i$	PF۲	Simple linear	r_i	۱,۱۵

جدول ۷. نتایج حل برای الگوریتم ابتکاری.

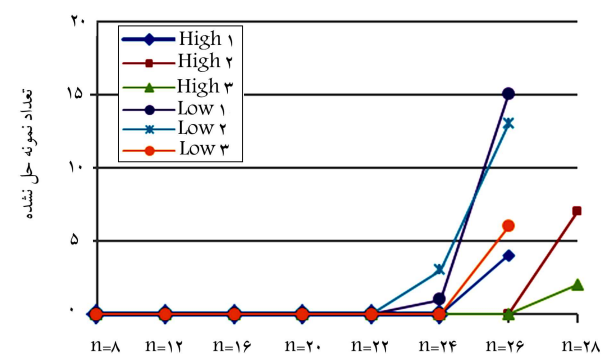
گروه داده	تعداد کار	زمان حل (ثانیه)		
		حداقل	متوسط	حداکثر
High ۱	۲۰۰	۰,۱۲۵	۰,۲۰۳	۰,۲۶۵
	۴۰۰	۱,۰۹۲	۱,۶۳۸	۲,۰۷۵
	۶۰۰	۳,۶۰۴	۴,۷۲۷	۵,۶۷۸
	۸۰۰	۸,۶۲۷	۱۱,۰۷۶	۱۳,۴۷۸
	۱۰۰۰	۱۷,۴۱۰	۲۱,۸۰۹	۲۵,۵۳۷
High ۲	۲۰۰	۰,۱۲۲	۰,۱۸۲	۰,۲۱۷
	۴۰۰	۱,۰۴۶	۱,۴۶۲	۲,۰۴۹
	۶۰۰	۳,۱۹۸	۴,۶۸۹	۵,۴۱۰
	۸۰۰	۹,۶۲۲	۱۲,۶۴۷	۱۴,۵۳۳
	۱۰۰۰	۱۶,۰۰۵	۲۵,۸۵۴	۳۴,۸۲۶
High ۳	۲۰۰	۰,۱۲۱	۰,۱۷۳	۰,۱۹۷
	۴۰۰	۰,۹۳۷	۱,۰۵۰	۱,۹۸۹
	۶۰۰	۴,۴۱۵	۵,۸۰۲	۶,۲۱۵
	۸۰۰	۷,۶۳۲	۸,۵۰۵	۱۰,۴۲۴
	۱۰۰۰	۱۶,۲۱۲	۱۸,۲۸۶	۲۲,۳۹۳
Low ۱	۲۰۰	۰,۲۸۶	۰,۳۲۵	۰,۳۹۰
	۴۰۰	۲,۱۳۲	۲,۴۰۵	۲,۶۷۸
	۶۰۰	۵,۲۸۰	۶,۲۶۹	۶,۳۱۵
	۸۰۰	۱۶,۸۶۱	۱۸,۷۹۸	۲۳,۲۴۶
	۱۰۰۰	۳۳,۲۱۵	۴۱,۷۰۴	۵۶,۴۹۸
Low ۲	۲۰۰	۰,۲۱۷	۰,۲۷۵	۰,۳۲۸
	۴۰۰	۱,۳۹۱	۱,۴۵۷	۱,۶۵۴
	۶۰۰	۶,۹۲۷	۶,۹۶۵	۷,۵۸۲
	۸۰۰	۱۰,۰۱۳	۱۰,۵۷۶	۱۹,۲۴۸
	۱۰۰۰	۲۵,۷۷۹	۲۷,۸۵۷	۳۳,۸۲۸
Low ۳	۲۰۰	۰,۱۸۴۴	۰,۲۴۴	۰,۳۰۶
	۴۰۰	۱,۴۶۶	۱,۸۸۶	۲,۲۳۴
	۶۰۰	۴,۹۴۶	۵,۸۳۶	۶,۵۹۷
	۸۰۰	۹,۵۳۵	۱۳,۶۵۹	۱۷,۶۰۴
	۱۰۰۰	۲۲,۹۷۲	۲۸,۷۳۳	۳۶,۶۸۸

۸. نتیجه‌گیری

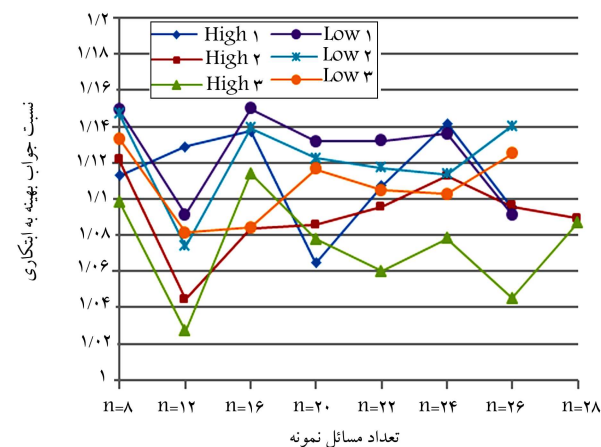
در این تحقیق مسئله‌ی زمان‌بندی کارهای رو به زوال با تابع زوال خطی روی جریان کارگاهی دو ماشین با هدف کمینه‌کردن تعداد کارهای تأخیردار و با فرض ورود غیر هم‌زمان کارها بررسی شد. با توجه به این که در ادبیات موضوع مطالعه‌ی پیرامون این مسئله مشاهده نشد، ابتدا پیچیدگی مسئله بررسی، و ثابت شد که مسئله از نوع NP-hard است. از این رو برای حل دقیق آن یک الگوریتم شاخه و کران، و برای به دست آوردن جوابی نزدیک به بهینه در زمانی کوتاه، یک روش ابتکاری شامل چهار الگوریتم ارائه شد. به منظور بررسی عملکرد راه حل‌های ارائه شده تعداد ۲۴۰۰ مسئله در ۴۸ سری و در قالب دو گروه اصلی بالا و پایین تولید و حل شد. نتایج



شکل ۱. متوسط زمان حل الگوریتم شاخه و کران.



شکل ۲. تعداد مسائل حل نشده در بازه زمانی ۳۶۰۰ ثانیه.



شکل ۳. نسبت جواب بهینه به ابتکاری برای سری‌های مختلف داده.

نتیجه گرفت که الگوریتم ابتکاری ارائه شده توانایی قابل قبولی در حل مسائل دارد. در شکل ۳ نسبت Z^{opt}/Z^{heu} برای سری‌های مختلف داده نشان داده شده است.

با توجه به کارایی بالای الگوریتم ابتکاری برای مسائل کوچک از آن برای حل مسائل بزرگ نیز استفاده شده است. و با الگوبررسی از مطالعات قبلی [۱۵]، ۱۵۰۰ مسئله با ابعاد ۲۰۰، ۴۰۰، ۶۰۰، ۸۰۰ و ۱۰۰۰ کار تولید و حل شده است (جدول ۷). متوسط زمان حل برای مسائل با ابعاد بزرگ حداکثر برابر ۴۱۷۰۴ ثانیه است که در مقایسه با داده‌های قبلی [۹]، الگوریتم ابتکاری در مسائل با ابعاد بزرگ نیز عملکرد مناسبی از خود نشان می‌دهد.

به عملکرد خوب الگوریتم ابتکاری در مسائل با ابعاد کوچک، تعداد ۱۵۰۰ مسئله با ابعاد بزرگ نیز تولید و حل و نتایج آن ارائه شده که نشان می‌دهد این الگوریتم برای مسائل بزرگ نیز عملکرد خیلی خوبی دارد.

مسائل زمان بندی فعالیت‌های روبه زوال یکی از زمینه‌های جذاب برای تحقیقات آینده است. حل مسئله‌ی حل شده در این مقاله روی جریان کارگاهی با ۳ یا تعداد بیشتری ماشین، یا حل آن روی ماشین‌های موازی و یا در محیط تولید کارگاهی، می‌تواند به عنوان زمینه‌ی برای تحقیقات آینده مطرح باشد. بررسی مسئله با فرض زوال خطی - تکه‌ی یا غیرخطی نیز می‌تواند یکی از ابعاد گسترش مسئله باشد. در نظر گرفتن فرض محدودیت دسترسی به ماشین از دیگر پیشنهاداتی است که می‌تواند جذاب باشد.

محاسباتی نشان داد که الگوریتم شاخه و کران قادر است تمام مسائل را تا اندازه ۲۴ کار در دسته‌ی بالا و ۲۲ کار در دسته‌ی پایین در زمان منطقی حل کند. در بعضی گروه‌ها نظیر High ۲، High ۳ و Low ۳ مسائل با ابعاد بزرگ‌تر نیز حل شده است. همچنین نتایج نشان داد که متوسط درصد گره‌های قطع شده توسط مجموعه‌ی لم‌ها و حدود پایین در همه‌ی مسائل درصد بالایی است و کم‌ترین مقدار آن برابر ۸۱/۳۱ درصد است که نشان می‌دهد الگوریتم شاخه و کران ارائه شده از کارایی بالایی برخوردار است. علاوه بر این عملکرد الگوریتم ابتکاری نیز مورد بررسی قرار گرفت و نتایج نشان داد که متوسط نسبت جواب بهینه به جواب الگوریتم ابتکاری با تابع هدف $\sum (1 - U_i)$ حداکثر برابر ۱/۱۵ است که این عدد در مقایسه با سایر تحقیقات مربوط به تعداد کارهای تأخیردار عدد بسیار خوبی است. همچنین با توجه

پانویس‌ها

1. deterioration
2. shortest processing time (SPT)
3. earliest due date (EDD)
4. back-tracking
5. low
6. high

منابع (References)

1. Wang, J.-B., Wang, L.-Y., Wang, D. and Wang, X.-Y. "Single-machine scheduling with a time-dependent deterioration", *The International Journal of Advanced Manufacturing Technology*, **43**(7-8), pp. 805-809 (2009).
2. Wang, J.-B., Ji, P., Cheng, T.C.E. and Wang D. "Minimizing makespan in a two-machine flow shop with effects of deterioration and learning", *Optimization Letters*, **6**(7), pp. 1393-1409 (2012).
3. Pinedo, M., *Scheduling: Theory, Algorithms, and Systems*, Prentice Hall (2002).
4. Lee, W.-C., Wu, C.-C. and Chung, Y.-H. "Scheduling deteriorating jobs on a single machine with release times", *Computers & Industrial Engineering*, **54**(3), pp. 441-452 (2013).
5. Lee, W.-C., Wu, C.-C., Wen, C.-C. Chung, Y.H. "A two-machine flowshop makespan scheduling problem with deteriorating jobs", *Computers & Industrial Engineering*, **54**(4), pp. 737-749 (2008).
6. Wang, J.-B., Ng, C.T.D., Cheng, T.C.E. and Liu, L.L "Minimizing total completion time in a two-machine flow shop with deteriorating jobs", *Applied Mathematics and Computation*, **180**(1), pp. 185-193 (2009).
7. Lee, W.-C., Wu, C.-C., Chung, Y.H. and Liu, H.C. "Minimizing the total completion time in permutation flow shop with machine-dependent job deterioration rates", *Computers & Operations Research*, **36**(6), pp. 2111-2121 (2011).
8. Lee, W.-C., Yeh, W.-C. and Chung, Y.-H. "Total tardiness minimization in permutation flowshop with deterioration consideration", *Applied Mathematical Modelling*, **38**(13), pp. 3081-3092 (2014).
9. Jafari, A. and Moslehi, G. "Scheduling linear deteriorating jobs to minimize the number of tardy jobs", *Journal of Global Optimization*, **54**(2), pp. 389-404 (2012).
10. Moslehi, G., Hakimian, A. and Abouei, A.M. "Minimizing the number of tardy jobs in a two-machine flowshop problem with non-simultaneous job entrance", *International Journal of Industrial Engineering*, **23**(4), pp. 389-400 (2013).
11. Brucker, P. and Brucker, P., *Scheduling Algorithms*, Springer, 371 p. (2007).
12. Cheng, T.E. and Ding, Q. "The complexity of scheduling starting time dependent tasks with release times", *Information Processing Letters*, **65**(2), pp. 75-79 (2005).
13. Moore, J.M. "An n job, one machine sequencing algorithm for minimizing the number of late jobs", *Management Science*, **15**(1), pp. 102-109 (1968).
14. Browne, S. and Yechiali, U. "Scheduling deteriorating jobs on a single processor", *Operations Research*, **38**(3), pp. 495-498 (1997).
15. Zhao, C. and Tang, H. "Two-machine flow shop scheduling with deteriorating jobs and chain precedence constraints", *International Journal of Production Economics*, **136**(1), pp. 131-136 (2012).
16. M'Hallah, R. and Bulfin, R. "Minimizing the weighted number of tardy jobs on parallel processors", *European Journal of Operational Research*, **160**(2), pp. 471-484 (2005).
17. Bulfin, R. and M'Hallah, R. "Minimizing the weighted number of tardy jobs on a two-machine flow shop", *Computers & Operations Research*, **30**(12), pp. 1887-1900 (2003).