

ارائه یک مدل ریاضی و یک الگوریتم شاخه‌وکران برای مسأله زمان‌بندی تک‌ماشین با فرض زوال خطی و ورود غیرهم‌زمان کارها

عباسعلی جعفری‌ندوشن^{۱*}، محمدحسین دهقانی‌صدرآبادی^۲، علی بزرگی‌امیری^۳

۱. استادیار گروه مهندسی صنایع، دانشکده فنی و مهندسی، دانشگاه میبد، میبد، ایران

۲. دانشجوی دکتری، گروه مهندسی صنایع، دانشکده مهندسی صنایع، دانشگاه علم و صنعت ایران، تهران، ایران

۳. دانشیار گروه مهندسی صنایع، دانشکده مهندسی صنایع، پردیس دانشکده‌های فنی، دانشگاه تهران، تهران، ایران

خلاصه

در این مقاله مسأله زمان‌بندی تک‌ماشین با فعالیت‌های روبه زوال خطی و فرض ورود غیرهم‌زمان کارها مورد بررسی قرار گرفته شده است که هدف حداقل کردن تعداد کارهای دارای دیرکرد می‌باشد. با تکیه بر ادبیات موضوع ثابت می‌گردد که مسأله موردنظر یک مسأله NP -hard است. در ابتدا یک مدل ریاضی برای مسأله ارائه شده و جهت حل مسأله به صورت بهینه نیز یک الگوریتم شاخه‌وکران با در نظر گرفتن اصول غلبه و حدود پایین پیشنهاد گردیده است. به منظور بررسی عملکرد الگوریتم شاخه‌وکران پیشنهادی و همچنین تأثیر پارامترهای مرتبط روی این الگوریتم، نتایج محاسباتی در چهار مرحله ارائه شده است. بر اساس آزمون تحلیل واریانس مشخص گردید که کارایی الگوریتم شاخه‌وکران بالاست به طوری که قادر به حل اکثر مسائل با ابعاد ۳۰ فعالیت در مدت زمان قابل قبولی بوده و متوسط درصد کل گره‌های قطع شده در تمامی مسائل حداقل برابر با ۸۵٫۶۱ درصد می‌باشد. همچنین نشان داده شد که مسائل با لاندای بزرگ‌تر و نرخ زوال کوچک‌تر سخت هستند و متوسط زمان حل الگوریتم در آن‌ها بالا می‌باشد. از طرفی اگر موعد تحویل کارها بزرگ یا کوچک باشند نیز مسأله ساده بوده و زمان حل آن نسبت به مسائل با موعد تحویل متوسط کمتر است.

اطلاعات مقاله

تاریخچه مقاله:

دریافت ۱۴۰۰/۲/۱۷

پذیرش ۱۴۰۰/۹/۰۵

(مقاله پژوهشی)

کلمات کلیدی:

فعالیت‌های روبه زوال

زمان‌بندی

تعداد کارهای دارای دیرکرد

ورود غیرهم‌زمان

الگوریتم شاخه وکران

۱. مقدمه

به طور کلی هدف در مسائل زمان‌بندی بهره‌برداری کارا از منابع، پاسخ‌گویی سریع به تقاضا و انطباق دقیق زمان‌های تحویل با موعد تحویل تعیین شده می‌باشد [۱]. امروزه مسائل زمان‌بندی در بسیاری از سیستم‌های تولیدی و خدماتی کاربرد وسیعی یافته‌اند. از جمله نقاط ضعف در مسائل زمان‌بندی کلاسیک، ثابت بودن زمان پردازش فعالیت‌ها می‌باشد اگرچه این فرض در بعضی از حالت‌ها درست است ولی به طور کلی باعث عدم تطابق بین مدل‌های موجود با شرایط واقعی و عملیاتی می‌گردد [۲]. در بسیاری از موارد عواملی مانند تأثیرات فراموشی، خستگی، یادگیری و زوال از جمله موضوعاتی هستند که در

شرایط واقعی وجود دارند و باعث تغییر در زمان پردازش فعالیت‌ها و یا حتی زمان آماده‌سازی می‌گردند [۳]. میزان و نحوه تأثیر این عوامل روی زمان پردازش متفاوت است به طوری که برخی از آن‌ها باعث افزایش و برخی دیگر باعث کاهش زمان می‌گردد. اثر یادگیری در سرعت اپراتورها و کاهش زمان تولید مشهود است به طوری که با گذر زمان و یادگیری اپراتورها، سرعت آن‌ها افزایش یافته که این کاهش زمان پردازش فعالیت‌ها را به همراه خواهد داشت [۴]. از طرف دیگر، عواملی مانند خستگی، فراموشی و زوال منجر به افزایش زمان می‌گردد. خستگی باعث می‌شود که فرد توانایی لازم جهت انجام فعالیت را نداشته باشد که این می‌تواند روی عملکرد فرد و در نتیجه زمان انجام کار

* نویسنده مسئول: عباسعلی جعفری‌ندوشن

تلفن: ۰۳۵-۲۲۲۲۴۱۶؛ پست الکترونیکی: a.jafari@meybod.ac.ir

بخش ۴ روش حل مسأله شامل یک رویکرد شاخه و کران به همراه اصول غلبه و حدود پایین ارائه می‌گردد. نتایج محاسباتی و ارزیابی عملکرد الگوریتم پیشنهادی در بخش ۴ مورد بحث قرار خواهد گرفت. بخش ۵ نیز به جمع بندی، نتیجه گیری و پیشنهادات آتی اختصاص داده شده است.

۲. پیشینه پژوهش

چنگ و همکاران [۱۱] مسأله زمان بندی تک ماشین با هدف حداقل کردن زمان تکمیل کل کارها را بررسی کردند که زمان پردازش هر کار تابعی از زمان شروع آن می‌باشد. آن‌ها درجه پیچیدگی مسأله را بررسی و یک الگوریتم برنامه ریزی پویا برای حل آن ارائه نمودند. براون و یچپالی [۱۲] در تحقیق خود ثابت کردند که جواب بهینه در مسأله $1 | P_i = a_i + b_i S_i | C_{max}$ براساس توالی ایجاد شده با نرخ غیرکاهشی a_i/b_i به دست می‌آید. لی و همکاران [۱۳] مسأله زمان بندی فعالیت‌های روبه زوال خطی با فرض ورود غیرهم زمان کارها در محیط تک ماشین با هدف حداقل نمودن دامنه عملیات را بررسی نمودند $(1 | r_i, P_i = a_i + b_i S_i | C_{max})$. آن‌ها در این تحقیق، یک رویکرد شاخه و کران ارائه کردند که مسائل با ابعاد ۲۸ فعالیت را در مدت زمان کوتاهی حل می‌نماید. آن‌ها همچنین در این تحقیق دو الگوریتم ابتکاری برای مسأله ارائه و نشان دادند که خطای بهترین جواب الگوریتم ابتکاری بسیار کم می‌باشد. جعفری و مصلحی [۱۴] مسأله $1 | P_i = a_i + b_i S_i | \sum U_i$ را مورد بررسی قرار دادند که در آن نرخ زوال برای تمام کارها یکسان فرض شده است. آن‌ها ثابت کردند که این مسأله یک مسأله $NP-hard$ می‌باشد بنابراین جهت حل آن به صورت بهینه یک رویکرد شاخه و کران ارائه نمودند که مسائل با ابعاد ۲۸ فعالیت را در زمان مناسبی حل می‌نماید. سو و لین [۱۵] مسأله $1 | P_i = a_i + b_i S_i | L_{max}$ را مطالعه کرده و برای حل آن یک الگوریتم شاخه و کران ارائه نمودند که مسائل با ابعاد ۱۰۰ فعالیت را به طور متوسط در مدت زمان ۱ دقیقه حل می‌کند. چنگ و همکاران [۱۶] در تحقیق خود زوال را علاوه بر زمان پردازش کارها برای زمان آماده سازی نیز در نظر گرفته‌اند. آن‌ها تابع زوال را به صورت خطی ساده در نظر گرفته و مسأله $1 | P_i = b_i S_i, ST_i = b'_i S_i | T_{max}$ را بررسی و به منظور حل بهینه آن یک الگوریتم شاخه و کران ارائه کردند به طوری که مسائل با ابعاد ۱۰۰۰ کار را در زمان مناسبی حل می‌نماید. لی و همکاران [۱۷] برای حل مسأله $1 | P_i = b_i S_i, ST_i = b'_i S_i | \sum U_i$ یک رویکرد شاخه و کران ارائه نمودند که توانایی حل مسائل با ابعاد ۱۰۰۰ کار را دارد. همچنین لی و لو [۱۸] در تحقیقی مسأله $1 | P_i = b_i S_i, ST_i = b'_i S_i | \sum w_i U_i$ را مطالعه کرده و جهت حل آن به صورت بهینه یک الگوریتم شاخه و کران ارائه نمودند که مسائل با ابعاد ۱۵۰۰ کار را در زمان مناسبی حل می‌نماید. وو و همکاران [۱۹] مسأله زمان بندی فعالیت‌های روبه زوال خطی با ورود غیرهم زمان کارها و حداقل کردن مجموع دیرکرد فعالیت‌ها $(1 | r_i, P_i = a_i + b_i S_i | \sum T_i)$ را مورد بررسی قرار دادند. آن‌ها برای

تأثیرگذار باشد. فراموشی نیز باعث می‌شود که فرد موارد مرتبط با فعالیت خود را به یاد نیاورد که این نیز می‌تواند موجب افزایش در زمان انجام فعالیت گردد [۵]. یکی از حوزه‌هایی که اخیراً توسط محققین مورد توجه قرار گرفته شده است مسأله زمان بندی فعالیت‌های روبه زوال می‌باشد. فعالیت روبه زوال فعالیت است که زمان پردازش آن ثابت نبوده و به کارهای زمان بندی شده قبلی بستگی دارد. به عبارت دیگر یک فعالیت که دیرتر پردازش می‌شود نسبت به فعالیت که زودتر مورد پردازش قرار می‌گیرد نیاز به زمان بیشتری جهت پردازش دارد [۶]. کاربرد چنین مسائلی در حوزه‌هایی مانند فرآیند نورد در صنایع فولاد، فرآیند تراش کاری، عملکرد روبه‌های درمانی، مهار کردن آتش سوزی، تعمیر ماشین‌آلات و وسایل نقلیه مشاهده می‌گردد. به عنوان مثال در صنایع فولاد هرشمش که زودتر نورد شود تبادل گرمای کمتری با محیط داشته و بنابراین مدت زمان نورد آن کمتر طول می‌کشد. همچنین ابزارآلات تراش کاری با انجام فرآیند تراش کاری فرسوده می‌شوند. به دلیل فرسودگی این ابزارآلات، زمان مورد نیاز برای تراش کاری قطعات بعدی براساس زمان پردازش قطعات تراش کاری شده افزایش می‌یابد [۷]. اگر درمان بیماری با تأخیر صورت گیرد آنگاه دوره درمان بیماری افزایش می‌یابد. همچنین اگر مهار آتش به موقع صورت نگیرد وسعت آتش بیشتر شده و مدت زمان فرآیند خاموش کردن آن افزایش می‌یابد. هرچه قطعه معیوب ماشین‌آلات یا وسایل نقلیه دیرتر تعمیر شود، بیشتر روبه زوال رفته و مدت زمان تعمیر آن افزایش می‌یابد. این مثال نشان می‌دهد که فعالیت‌های روبه زوال در زمینه برنامه ریزی و زمان بندی نت نیز مورد استفاده قرار می‌گیرد [۸].

در مسائل زمان بندی فعالیت‌های روبه زوال زمان واقعی پردازش کارها و یا زمان آماده سازی براساس یک تابع زوال محاسبه می‌گردد که این توابع به سه دسته اصلی خطی، خطی تکه‌ای و نمایی تقسیم می‌شوند. در تابع خطی زمان پردازش واقعی کار i (J_i) براساس رابطه خطی $P_i = a_i + b_i S_i$ به زمان شروع آن کار بستگی دارد که در آن a_i, b_i, S_i و P_i به ترتیب مدت زمان پردازش پایه J_i ، نرخ زوال J_i ، زمان شروع واقعی J_i و زمان پردازش واقعی J_i می‌باشد [۹]. از آنجاکه محققین در تحقیقات خود بیشتر تابع زوال خطی را بررسی نموده‌اند لذا در این تحقیق نیز تابع زوال به صورت خطی فرض می‌گردد. با توجه به فرضیاتی که برای دو متغیر a_i و b_i می‌توان در نظر گرفت تابع خطی می‌تواند به سه صورت $P_i = a_i + b_i S_i$ ، $P_i = a_i + b_i S_i$ و $P_i = b_i S_i$ فرض شود که در ادامه تحقیقات صورت گرفته در این زمینه ارائه می‌گردد. به منظور بیان ساده تر مسائل زمان بندی، از نمادگذاری گراهام [۱۰] به صورت $\gamma | \beta | \alpha$ استفاده شده که در آن α نشان دهنده محیط ماشین، β نشان دهنده ویژگی‌های مسأله و γ نشان دهنده تابع هدف مسأله می‌باشد.

در ادامه در بخش ۲ تحقیقات مرتبط بررسی و پیشینه پژوهش ارائه می‌گردد. ساختار مسأله، مفاهیم اصلی مورد استفاده در مقاله به همراه مدل ریاضی مسأله و پیچیدگی آن در بخش ۳ تشریح می‌گردد. در

لی و لو [۲۷] مسأله زمان‌بندی در محیط تک‌ماشین با فرض زوال خطی، پردازش دسته‌ای و رد کردن برخی سفارشات را بررسی نمودند. هدف در این تحقیق حداقل کردن دامنه عملیات و مجموع زمان تکمیل کارها می‌باشد. آن‌ها پیچیدگی مسأله را بررسی و ثابت کردند که این مسأله NP -complete می‌باشد که برای حل آن یک رویکرد برنامه‌ریزی پویا ارائه کردند. مور و موشیو [۲۸] مسأله زوال خطی را در محیط ماشین‌های موازی با هدف حداقل کردن زمان تکمیل کل بررسی نمودند. آن‌ها نشان دادند که مسأله NP -hard می‌باشد و برای حل آن یک الگوریتم ابتکاری با کیفیت جواب مناسب ارائه نمودند. بای و همکاران [۲۹] برای حل مسأله زمان‌بندی با فرض اثرات یادگیری و ورود غیرهم‌زمان کارها در محیط جریان کارگاهی باهدف حداقل کردن ماکزیمم مغایرت، از یک رویکرد شاخه‌وکران و الگوریتم فراابتکاری زنبور عسل استفاده نمودند.

در جدول (۱) خلاصه وضعیت مهم‌ترین تحقیقات انجام شده در زمینه زمان‌بندی فعالیت‌های روبه زوال خطی ارائه شده است و تحقیق حاضر نیز با این مقالات مقایسه گردیده است. با بررسی مقالات موجود در ادبیات موضوع می‌توان دریافت که در سال‌های اخیر زمان‌بندی فعالیت‌های روبه زوال یک موضوع جذاب و قابل توجه برای محققان بوده است. براساس جدول (۱) و با تکیه بر ادبیات موضوع، می‌توان بیان کرد که در اکثر تحقیقات صورت گرفته در زمینه فعالیت‌های روبه زوال فرض ورود غیرهم‌زمان کارها که یک فرض واقعی در مسائل تولید می‌باشد نادیده گرفته شده است. از طرفی ورود غیرهم‌زمان کارها باهدف حداقل کردن تعداد کارهای دارای دیرکرد در حوزه فعالیت‌های روبه زوال موضوعی است که تاکنون تحقیقی در مورد آن یافت نشده است. بنابراین در این مقاله مسأله زمان‌بندی فعالیت‌های روبه زوال خطی در محیط تک‌ماشین با هدف حداقل کردن تعداد کارهای دارای دیرکرد با فرض ورود غیرهم‌زمان کارها مورد بررسی قرار گرفته شده است.

حل مسأله یک رویکرد شاخه‌وکران ارائه نمودند که قادر به حل مسائل با ابعاد ۲۰ فعالیت در زمان مناسبی می‌باشد. لی و ژائو [۲۰] مسأله زمان‌بندی فعالیت‌های روبه زوال خطی روی تک‌ماشین با فرض محدودیت دسترسی به ماشین و ورود غیرهم‌زمان کارها را بررسی نمودند که هدف حداقل کردن دامنه عملیات است. آن‌ها نشان دادند که مسأله دارای پیچیدگی بالایی می‌باشد.

جعفری و همکاران [۲۱] مسأله $F2 | P_i = a_i + b_i S_i | C_{max}$ را در تحقیق خود مورد بررسی قرار داده و برای حل آن یک الگوریتم شاخه‌وکران ارائه نمودند که قادر به حل مسائل با ابعاد ۲۶ فعالیت می‌باشد. همچنین آن‌ها جهت محاسبه جواب نزدیک به بهینه چند الگوریتم ابتکاری پیشنهاد کردند. لی و همکاران [۲۲] یک الگوریتم شاخه‌وکران و چندین الگوریتم ابتکاری با خطای حداکثر ۴٪ را جهت حل مسأله $F2 | P_i = a_i + b S_i | C_{max}$ ارائه نمودند. الگوریتم شاخه‌وکران پیشنهادی آن‌ها مسائل با ابعاد ۲۸ فعالیت را در زمان مناسبی حل می‌نماید. وانگ و وانگ [۲۳] جهت تعیین جواب نزدیک به بهینه برای مسأله $F3 | P_i = a_i + b S_i | C_{max}$ دو الگوریتم ابتکاری و برای حل مسأله به صورت بهینه یک رویکرد شاخه‌وکران شامل چندین اصول غلبه و حدود پایین ارائه نمودند که قادر به حل مسائل با ابعاد ۲۸ فعالیت به صورت بهینه می‌باشد. جعفری و همکاران [۲۴] با یک مثال نقض نشان دادند که این اصول غلبه که در الگوریتم شاخه‌وکران توسط وانگ و وانگ [۲۳] ارائه شده صحیح نمی‌باشد. وانگ و همکاران [۲۵] مسأله $F3 | P_i = b_i S_i | C_{max}$ را بررسی و جهت حل آن یک الگوریتم ابتکاری و یک الگوریتم شاخه‌وکران شامل چندین اصول غلبه، حدود پایین و حدود بالا ارائه نمودند که مسائل با ابعاد ۱۴ فعالیت را در زمان مناسب حل می‌نماید. جعفری و همکاران [۲۶] در تحقیق خود نشان دادند که اصول غلبه ارائه شده در الگوریتم شاخه‌وکران توسط وانگ و همکاران [۲۵] صحیح نمی‌باشد.

جدول (۱): مقایسه منابع مهم و مرتبط در ادبیات موضوع با تحقیق حاضر

ابعاد مسائل حل شده	نوع محیط		ورود غیرهم‌زمان کارها	تابع زوال		نوع مسأله	تابع هدف	نویسنده (گان)
	تک‌ماشین	جریان کارگاهی		خطی ساده	خطی با نرخ زوال یکسان			
قاعده بهینه	✓			✓		$1 P_i = a_i + b_i S_i C_{max}$		براون و بیچالی [۱۲]
۲۸ فعالیت		✓	✓		✓	$1 r_i, P_i = a_i + b S_i C_{max}$	دامنه عملیات	لی و همکاران [۱۳]
۲۸ فعالیت		✓			✓	$1 P_i = a_i + b S_i \sum U_i$	تعداد کارهای دارای دیرکرد	جعفری و مصلحی [۱۴]
۱۰۰ فعالیت				✓		$1 P_i = a_i + b_i S_i L_{max}$	حداکثر مغایرت	سو و لین [۱۵]
۲۰ فعالیت		✓	✓		✓	$1 r_i, P_i = a_i + b S_i \sum T_i$	دیرکرد کل	وو و همکاران [۱۹]
۲۶ فعالیت	✓			✓		$F2 P_i = a_i + b_i S_i C_{max}$	دامنه عملیات	جعفری و همکاران [۲۱]

۲۸ فعالیت	✓		✓	$F2 P_i = a_i + b_i S_i C_{max}$	دامنه عملیات	لی و همکاران [۲۲]
۲۸ فعالیت	✓		✓	$F3 P_i = a_i + b_i S_i C_{max}$	دامنه عملیات	وانگ و وانگ [۲۳]
۱۴ فعالیت	✓		✓	$F3 P_i = b_i S_i C_{max}$	دامنه عملیات	وانگ و همکاران [۲۵]
۱۰۰۰ فعالیت		✓	✓	$1 P_i = b_i S_i, ST_i = b'_i S_i \sum U_i$	تعداد کارهای دارای دیرکرد	لی و همکاران [۱۷]
۱۵۰۰ فعالیت		✓	✓	$1 P_i = b_i S_i, ST_i = b'_i S_i \sum w_i U_i$	تعداد کارهای دارای دیرکرد وزنی	لی و لو [۱۸]
۱۰۰۰ فعالیت		✓	✓	$1 P_i = b_i S_i, ST_i = b'_i S_i T_{max}$	حداکثر دیرکرد	چنگ و همکاران [۱۶]
۳۰ فعالیت	✓	✓	✓	$1 r_i, P_i = a_i + b_i S_i \sum U_i$	تعداد کارهای دارای دیرکرد	تحقیق حاضر

۳. تشریح ساختار مسأله

در این قسمت به ترتیب پارامترها و متغیرهای مورد استفاده در مقاله معرفی، مسأله موردنظر و فرضیات آن تعریف و پیچیدگی مسأله بررسی می‌شود.

۳-۱. پارامترها، نمادها و متغیرهای مسأله

پارامترها، نمادها و متغیرهای تصمیم مورد استفاده در مقاله به شرح زیر می‌باشد:

n : تعداد کل کارهایی که باید زمان بندی گردد.

J_i : کار نام $(i=1,2,\dots,n)$

a_i : زمان پردازش پایه J_i

b_i : نرخ زوال J_i

r_i : زمان ورود J_i

d_i : موعد تحویل J_i

نمادها:

δ : توالی جزئی شامل کارهای زمان بندی شده

δ' : مجموعه کارهای زمان بندی نشده و مکمل δ

δ_0 : یک توالی دلخواه از مجموعه δ'

δ_i : توالی جزئی ایجاد شده ناشی از زمان بندی J_i بعد از توالی جزئی δ

δ_{Early} : مجموعه کارهای دارای زودکرد

δ_{Tardy} : مجموعه کارهای دارای دیرکرد

متغیرهای تصمیم:

T : زمان تکمیل آخرین کار مجموعه δ_{Early}

k_i : زمان شروع J_i بلافاصله بعد از توالی δ_{Early} که یک متغیر اعشاری بوده و برابر با ماکزیمم بین زمان ورود آن کار (r_i) و زمان تکمیل آخرین کار زمان بندی شده می‌باشد. زمان شروع اولین کار زمان بندی شده صفر در نظر گرفته می‌شود.

P_i : زمان پردازش واقعی J_i که یک متغیر اعشاری بوده و در این مقاله براساس تابع زوال خطی $P_i = a_i + b_i S_i$ محاسبه می‌گردد.

C_i : زمان تکمیل J_i که یک متغیر اعشاری بوده و برابر با مجموع زمان شروع و زمان پردازش واقعی آن کار می‌باشد.
 $C_i(\delta_i)$: زمان تکمیل J_i در توالی جزئی δ_i
 $C_{[k]}$: زمان تکمیل کار زمان بندی شده در موقعیت k که یک متغیر اعشاری می‌باشد.

$X_{i[k]}$: متغیر تصمیم باینری که برابر با ۱ است اگر J_i در موقعیت k زمان بندی گردد. در غیر این صورت برابر با ۰ است.

U_i : متغیر تصمیم باینری که وضعیت تحویل دهی به موقع J_i را بیان می‌کند. اگر J_i به موقع تحویل گردد ($d_i \geq C_i$) یعنی J_i دارای زودکرد می‌باشد و مقدار U_i برابر با صفر می‌گردد. در غیر این صورت ($d_i < C_i$) مقدار U_i برابر با یک بوده و J_i دارای دیرکرد می‌باشد.

$$N_T = \sum_{i=1}^n U_i$$

تعداد کارهای دارای دیرکرد

$$N_T(\delta) = \sum_{i=1}^n U_i(\delta)$$

تعداد کارهای دارای دیرکرد در مجموعه δ

C_{max} : دامنه عملیات که برابر با زمان تکمیل n امین کار می‌باشد.

۳-۲. تعریف مسأله

در این پژوهش مسأله زمان بندی فعالیت‌های روبه زوال خطی با ورود غیرهم‌زمان کارها در محیط تک‌ماشین بررسی می‌گردد. زمان پردازش واقعی کارها براساس یک تابع زوال خطی با نرخ غیریکسان برای کارها در نظر گرفته می‌شود. باتوجه به این که ورود کارها به صورت غیرهم‌زمان به مسائل واقعی نزدیک‌تر است در این تحقیق مسأله با فرض ورود غیرهم‌زمان کارها بررسی می‌گردد. لازم است n کار روی یک ماشین در موقعیت‌های ۱ تا n و باتوجه به فرضیات زیر زمان بندی گردد به طوری که تعداد کارهای دارای دیرکرد حداقل شود.

- زمان پردازش واقعی هر کار براساس تابع زوال خطی $P_i = a_i + b_i S_i$ محاسبه می‌گردد.
- نرخ زوال برای تمام کارها غیریکسان و بزرگ‌تر از صفر می‌باشد.
- ورود کارها به صورت غیرهم‌زمان می‌باشد یعنی هر کار زمان ورود مخصوص به خود دارد که لزوماً با زمان ورود کارهای دیگر یکسان

نیست.

- انقطاع و شکست جایز نمی‌باشد.
- ماشین در طول افق برنامه‌ریزی همیشه در دسترس می‌باشد و توقف ندارد.
- ماشین در هر لحظه می‌تواند فقط یک کار را پردازش کند.
- باتوجه به ورود غیرهم‌زمان کارها، بیکاری عمدی مجاز فرض می‌شود.

این مسأله به صورت $1 | r_i, P_i = a_i + b_i S_i | \sum U_i$ نمایش داده می‌شود که هدف عبارت است از حداقل کردن تعداد کارهای دارای دیرکرد به طوری که کلیه کارها با در نظر گرفتن شرایط و فرضیات فوق روی یک ماشین زمان بندی گردد.

۳-۳. مدل ریاضی مسأله

در این بخش براساس مفروضات، پارامترها و متغیرهای تعریف شده یک مدل ریاضی به شرح زیر برای مسأله ارائه می‌گردد.

$$\text{Min } Z = \sum_{i=1}^n U_i \quad (1)$$

$$\sum_{k=1}^n X_{i[k]} = 1 \quad \forall i \quad (2)$$

$$\sum_{i=1}^n X_{i[k]} = 1 \quad \forall i \quad (3)$$

$$P_i = a_i + b_i S_i \quad \forall i \quad (4)$$

$$S_i \geq C_{[k-1]} X_{i[k]} \quad \forall i, k \quad (5)$$

$$S_i \geq r_i \quad \forall i \quad (6)$$

$$C_i = S_i + P_i \quad \forall i \quad (7)$$

$$C_{[k]} = C_i X_{i[k]} \quad \forall i, k \quad (8)$$

$$C_{[k]} - d_i X_{i[k]} \leq M U_i \quad \forall i, k \quad (9)$$

$$C_{[0]} = 0 \quad (10)$$

$$X_{i[k]}, U_i \in \{0, 1\} \quad \forall i, k \quad (11)$$

$$P_i, S_i, C_i, C_{[k]} \geq 0 \quad \forall i, k \quad (12)$$

تابع هدف مسأله در رابطه (۱) شامل حداقل کردن تعداد کارهای دارای دیرکرد است که براساس رابطه (۹) و بمقایسه بین موعد تحویل و زمان تکمیل کار زمان بندی شده در موقعیت k ام به دست می‌آید. محدودیت (۲) نشان می‌دهد که هر کار دقیقاً یک مرتبه در توالی مورد نظر زمان بندی می‌شود. همچنین رابطه (۳) نیز تضمین می‌کند که در هر موقعیت k دقیقاً یک کار زمان بندی می‌شود. زمان پردازش واقعی هر کار براساس رابطه (۴) که به صورت یک تابع زوال خطی می‌باشد محاسبه می‌گردد. زمان شروع هر کار جهت زمان بندی در موقعیت k به دو عامل زمان تکمیل کار زمان بندی شده در موقعیت

$k-1$ و زمان ورود آن کار بستگی دارد که به ترتیب در روابط (۵) و (۶) لحاظ می‌گردد. زمان تکمیل کار زمان بندی شده در موقعیت k نیز برابر با مجموع زمان پردازش و زمان شروع آن کار در آن موقعیت می‌باشد که روابط (۷) و (۸) آن را تضمین می‌نماید. برای اجرای مدل لازم است مقدار زمان تکمیل موقعیت 0 نیز مشخص گردد که براساس محدودیت (۱۰) برابر با 0 فرض می‌شود. محدودیت‌های (۱۱) و (۱۲) شرایط متغیرهای مسأله را تعیین می‌نماید.

۳-۴. پیچیدگی مسأله

با بررسی ادبیات موضوع و جستجو در زمینه فعالیت‌های روبه زوال، تاکنون مطالعه‌ای روی مسأله $1 | r_i, P_i = a_i + b_i S_i | \sum U_i$ مشاهده نشده است بنابراین لازم است در ابتدا درجه سختی و پیچیدگی مسأله مشخص گردد. همان‌گونه که از جدول (۱) مشاهده می‌شود فرض زوال خطی ساده و زوال خطی می‌تواند روی پیچیدگی مسائل تأثیرگذار باشد. مقایسه ابعاد مسائل حل شده در تحقیقات مرتبط نشان می‌دهد که فرض زوال خطی باعث افزایش پیچیدگی مسأله می‌شود به گونه‌ای که جعفری و مصلحی [۱۴] با حل مسأله زوال خطی در تحقیق خود موفق به حل مسائل با ابعاد ۲۸ فعالیت به صورت بهینه شده‌اند در صورتی که لی و همکاران [۱۷] این مسأله را با فرض زوال خطی ساده بررسی نمودند و مسائل با ابعاد ۱۰۰۰ فعالیت را به صورت بهینه حل کرده‌اند. این روند در تحقیقات لی و لو [۱۸] و چنگ و همکاران [۱۶] نیز مشاهده می‌شود. همچنین براساس جدول (۱) و بمقایسه تحقیقات نزدیک به هم می‌توان نتیجه گرفت که اضافه شدن فرض ورود غیرهم‌زمان کارها نیز باعث افزایش درجه پیچیدگی مسائل می‌گردد. براون و یچیالی [۱۲] مسأله زوال خطی را با یک قاعده ساده تقدیمی به صورت بهینه حل نموده است ولی لی و همکاران [۱۳] همین مسأله را با اضافه شدن فرض ورود غیرهم‌زمان کارها بررسی کردند که موفق به حل بهینه مسائل با ابعاد ۲۸ فعالیت شدند. بنابراین براساس ابعاد مسائل حل شده در ادبیات تحقیق می‌توان به این مسأله اشاره نمود که اضافه شدن فرض زوال و ورود غیرهم‌زمان کارها باعث افزایش پیچیدگی مسأله می‌گردد. لی و همکاران [۸] نشان دادند که مسأله $1 | r_i, P_i = a_i + b S_i | C_{max}$ از آنجاکه با ثابت بودن تمام فرض‌های یک مسأله، مسأله باهدف C_{max} قابل کاهش به مسأله باهدف $\sum U_i$ می‌باشد [۳۰] بنابراین مسأله $1 | r_i, P_i = a_i + b S_i | C_{max}$ قابل کاهش به مسأله $1 | r_i, P_i = a_i + b S_i | \sum U_i$ است. باتوجه به $NP-hard$ بودن مسأله $1 | r_i, P_i = a_i + b S_i | C_{max}$ می‌توان نتیجه گرفت که مسأله $1 | r_i, P_i = a_i + b_i S_i | \sum U_i$ نیز یک مسأله $NP-hard$ می‌باشد و لازم است جهت یافتن جواب بهینه از الگوریتم‌های عمومی مانند شاخه‌وکران استفاده نمود.

۴. روش حل مسأله

در این تحقیق به منظور دستیابی به جواب بهینه برای مسأله $1 | r_i, P_i = a_i + b_i S_i | \sum U_i$ یک رویکرد شاخه‌وکران با در نظر

گرفتن چندین اصول غلبه، حدود پایین و بالا ارائه می‌گردد که هر یک از آن‌ها در ادامه تشریح می‌شود.

۴-۱. الگوریتم شاخه و کران

در الگوریتم شاخه و کران مورد نظر یک شاخه انتخاب شده و به طور سیستماتیک با استراتژی جستجوی عمقی و رویکرد روبه جلو، درخت جستجو ایجاد می‌گردد به طوری که با شروع از موقعیت اول، کارها در موقعیت‌های بعدی زمان بندی می‌گردد. هر توالی جزئی δ مانند شکل (۱) شامل دو مجموعه δ_{Early} و δ_{Tardy} است که به ترتیب در ابتدا و انتهای یک توالی کامل قرار می‌گیرند به طوری که مجموعه δ_{Tardy} به صورت دلخواه زمان بندی می‌گردد. این موضوع باعث می‌شود در درخت جستجو، تنها توالی‌هایی که در آن کارهای مجاور دارای زودکرد می‌شوند بررسی گردد که کاهش چشم‌گیر تعداد گره‌های بررسی شده و افزایش کارایی الگوریتم را به همراه خواهد داشت.

گرفتن چندین اصول غلبه، حدود پایین و بالا ارائه می‌گردد که هر یک از آن‌ها در ادامه تشریح می‌شود.

۴-۱. الگوریتم شاخه و کران

در الگوریتم شاخه و کران مورد نظر یک شاخه انتخاب شده و به طور سیستماتیک با استراتژی جستجوی عمقی و رویکرد روبه جلو، درخت جستجو ایجاد می‌گردد به طوری که با شروع از موقعیت اول، کارها در موقعیت‌های بعدی زمان بندی می‌گردد. هر توالی جزئی δ مانند شکل (۱) شامل دو مجموعه δ_{Early} و δ_{Tardy} است که به ترتیب در ابتدا و انتهای یک توالی کامل قرار می‌گیرند به طوری که مجموعه δ_{Tardy} به صورت دلخواه زمان بندی می‌گردد. این موضوع باعث می‌شود در درخت جستجو، تنها توالی‌هایی که در آن کارهای مجاور دارای زودکرد می‌شوند بررسی گردد که کاهش چشم‌گیر تعداد گره‌های بررسی شده و افزایش کارایی الگوریتم را به همراه خواهد داشت.

δ_{Early}	δ'	δ_{Tardy}
------------------	-----------	------------------

شکل (۱): توالی جزئی δ

باتوجه به نوع تابع هدف و نحوه اجرای الگوریتم شاخه و کران، در هر مرحله که یک کار از مجموعه δ' جهت زمان بندی انتخاب می‌شود در ابتدا نکته ۱ که در ادامه شرح داده می‌شود بررسی می‌گردد. اگر نکته ۱ برقرار باشد کار دارای دیرکرد می‌باشد و این کار به مجموعه δ_{Tardy} اضافه گردیده و مقدار تابع هدف یک واحد افزایش می‌یابد در غیر این صورت اضافه شدن آن به انتهای مجموعه δ_{Early} ، توسط اصول غلبه و مقایسه حدود پایین با حدود بالا بررسی می‌شود. در صورتی که اصول غلبه و حدود پایین نتواند گره مربوطه را قطع کند آن کار به انتهای مجموعه δ_{Early} اضافه خواهد شد. سپس کار دیگری از مجموعه δ' انتخاب و مجدداً بررسی نکته ۱ و مراحل بعد از آن انجام می‌شود. اگر گره مربوطه قطع شود و یا درخت جستجو به گره آخر برسد شاخه دیگری انتخاب و رویه شرح داده شده تکرار می‌شود تا جایی که تمام شاخه‌ها بررسی گردند. اگر تعداد کارهای دیرکردار در آخرین توالی کامل، کاهش یابد این جواب جایگزین حد بالای مسأله خواهد شد. در ادامه حد بالا، اصول غلبه و حدود پایین الگوریتم تشریح می‌گردد.

$$S_i(\delta_1) = S_i = \text{Max}\{T, r_i\} = T \quad (۱۳)$$

همچنین باتوجه به تعریف مدت زمان پردازش واقعی J_i و رابطه (۱۳)، زمان تکمیل J_i در توالی δ_1 براساس رابطه (۱۴) تعریف می‌گردد.

$$C_i(\delta_1) = P_i + S_i = (a_i + b_i S_i) + S_i = a_i + (1 + b_i) S_i = a_i + (1 + b_i) T \quad (۱۴)$$

با تکیه بر رابطه $a_i + (1 + b_i) T > r_j$ در عنوان لم و همچنین رابطه (۱۴)، زمان شروع و تکمیل J_j در توالی δ_1 به ترتیب براساس روابط (۱۵) و (۱۶) محاسبه می‌شود.

$$S_j(\delta_1) = S_j = \text{Max}\{C_i(\delta_1), r_j\} = \text{Max}\{a_i + (1 + b_i) T, r_j\} = a_i + (1 + b_i) T \quad (۱۵)$$

$$C_j(\delta_1) = P_j + S_j = (a_j + b_j S_j) + S_j = a_j + (1 + b_j) S_j = a_j + (1 + b_j)(a_i + (1 + b_i) T) \quad (۱۶)$$

با توجیهی مشابه توجیه فوق و در نظر گرفتن رابطه $T < r_j$ در فرض لم، زمان شروع و تکمیل J_j در توالی δ_2 براساس روابط (۱۷) و (۱۸) محاسبه خواهد شد.

$$S_j(\delta_2) = S_j = \text{Max}\{T, r_j\} = r_j \quad (۱۷)$$

$$C_j(\delta_2) = P_j + S_j = (a_j + b_j S_j) + S_j = a_j + (1 + b_j) S_j = a_j + (1 + b_j) r_j \quad (۱۸)$$

به همین ترتیب با تکیه بر فرض $r_i < T < r_j$ موجود در صورت لم و رابطه (۱۸)، زمان شروع و تکمیل J_i در توالی δ_2 به ترتیب براساس روابط (۱۹) و (۲۰) در نظر گرفته می‌شود.

$$S_i(\delta_2) = S_i = \text{Max}\{C_j(\delta_2), r_i\} = \text{Max}\{a_j + (1 + b_j) r_j, r_i\} = a_j + (1 + b_j) r_j \quad (۱۹)$$

$$C_i(\delta_2) = P_i + S_i = (a_i + b_i S_i) + S_i = a_i + (1 + b_i) S_i = a_i + (1 + b_i)(a_j + (1 + b_j) r_j) \quad (۲۰)$$

باتوجه به رابطه $T < r_j$ در فرضیات لم رابطه (۲۱) برقرار می‌باشد.

باتوجه به نوع تابع هدف و نحوه اجرای الگوریتم شاخه و کران، در هر مرحله که یک کار از مجموعه δ' جهت زمان بندی انتخاب می‌شود در ابتدا نکته ۱ که در ادامه شرح داده می‌شود بررسی می‌گردد. اگر نکته ۱ برقرار باشد کار دارای دیرکرد می‌باشد و این کار به مجموعه δ_{Tardy} اضافه گردیده و مقدار تابع هدف یک واحد افزایش می‌یابد در غیر این صورت اضافه شدن آن به انتهای مجموعه δ_{Early} ، توسط اصول غلبه و مقایسه حدود پایین با حدود بالا بررسی می‌شود. در صورتی که اصول غلبه و حدود پایین نتواند گره مربوطه را قطع کند آن کار به انتهای مجموعه δ_{Early} اضافه خواهد شد. سپس کار دیگری از مجموعه δ' انتخاب و مجدداً بررسی نکته ۱ و مراحل بعد از آن انجام می‌شود. اگر گره مربوطه قطع شود و یا درخت جستجو به گره آخر برسد شاخه دیگری انتخاب و رویه شرح داده شده تکرار می‌شود تا جایی که تمام شاخه‌ها بررسی گردند. اگر تعداد کارهای دیرکردار در آخرین توالی کامل، کاهش یابد این جواب جایگزین حد بالای مسأله خواهد شد. در ادامه حد بالا، اصول غلبه و حدود پایین الگوریتم تشریح می‌گردد.

حد بالا

بهترین جواب از زمان بندی صورت گرفته براساس کمترین موعد تحویل، کمترین زمان پردازش پایه، کمترین نرخ زوال و کمترین نرخ a_i/b_i به عنوان حد بالای مسأله در نظر گرفته می‌شود.

اصول غلبه

جهت افزایش کارایی الگوریتم شاخه و کران و حذف گره‌ها و شاخه‌هایی که به جواب بهینه نمی‌رسد لازم است یک سری اصول غلبه برای مسأله تعریف گردد. بدین منظور فرض می‌شود که توالی δ_1 با دو کار مجاور J_i و J_j موجود است به طوری که J_i بلافاصله قبل از J_j زمان بندی می‌گردد. با یک تعویض جفتی بین J_i و J_j در توالی δ_1 توالی جدید δ_2 ایجاد می‌شود. بنابراین دو توالی کامل $(\delta_{Early}, J_i, J_j, \delta_o, \delta_{Tardy})$ و $(\delta_{Early}, J_j, J_i, \delta_o, \delta_{Tardy})$ را به همراه خواهد داشت.

همچنین زمان تکمیل J_i بلافاصله بعد از δ_{Early} متناسب با رابطه زیر می‌باشد.

$$C_i = a_i + (1 + b_i)S_i = a_i + (1 + b_i)Max\{T, r_i\} \quad (28)$$

باتوجه به رابطه (۲۸)، رابطه $d_i < a_i + (1 + b_i)Max\{T, r_i\}$ در نکته ۱ بیان‌کننده وضعیت دیرکرد برای J_i بلافاصله بعد از δ_{Early} می‌باشد. این بدان معناست که زمان‌بندی J_i در موقعیت‌های بعد از مجموعه δ_{Early} و همچنین مجموعه δ_{Tardy} باعث می‌شود که آن کار دارای دیرکرد گردد. بنابراین جهت دستیابی به جواب بهتر باید این کار به مجموعه δ_{Tardy} اضافه گردد. این جابه‌جایی نه تنها باعث افزایش زمان شروع کارهای بعدی نمی‌گردد بلکه کاهش زمان شروع برای موقعیت‌های بعدی و به تبع آن تابع هدف بهتر را به همراه خواهد داشت. بدیهی است این زمان‌بندی تاجایی که مجموعه δ_{Early} تغییر نکند برقرار خواهد بود.

حدود پایین

در این بخش باتوجه به اهمیت و تأثیر حد پایین روی الگوریتم شاخه‌وکران، سه حد پایین برای مسأله ارائه شده است. جهت محاسبه مقدار حد پایین که به صورت LB نمایش داده می‌شود می‌توان از قضایای زیر استفاده نمود.

یکی از ایده‌های موردنظر جهت محاسبه حد پایین، در نظر گرفتن مقدار زوال برای هر J_i از مجموعه δ' در زودترین زمان شروع ممکن یعنی $Max\{T, r_i\}$ می‌باشد. اگر زمان پردازش کارهای موجود در مجموعه δ' با فرض شروع آن کار در زمان $Max\{T, r_i\}$ محاسبه و باتوجه به الگوریتم مور و هاجسون زمان‌بندی گردد قضیه ۱ را می‌توان به صورت زیر بیان نمود.

قضیه ۱. مقدار حد پایین LB_1 برای مسأله موردنظر براساس رابطه زیر محاسبه می‌گردد.

$$LB_1 = N_T(\delta) + N_T(\delta') \quad (29)$$

که مقدار $N_T(\delta')$ از توالی حاصل از زمان‌بندی کارهای مجموعه δ' براساس الگوریتم مور و هاجسون و با فرض مقدار زوال J_i در زمان $Max\{T, r_i\}$ به دست می‌آید.

اثبات: رابطه (۲۹) از دو جمله با مقادیر واقعی و تخمینی تشکیل شده است. جمله $N_T(\delta)$ دارای یک مقدار واقعی و بیان‌کننده تعداد کارهای دارای دیرکرد در مجموعه δ می‌باشد. مقدار $N_T(\delta')$ یک مقدار تخمینی از تعداد کارهای دارای دیرکرد در مجموعه δ' می‌باشد. بدیهی است که هر توالی دلخواه از کارهای مجموعه δ' ، تأثیری بر مقدار $N_T(\delta)$ نخواهد داشت. بنابراین اگر کوچک‌تر بودن مقدار تخمینی $N_T(\delta')$ از مقدار واقعی تعداد کارهای دارای دیرکرد در مجموعه δ' ثابت گردد می‌توان نشان داد که رابطه (۲۹) یک حد پایین برای مسأله می‌باشد.

الگوریتم مور و هاجسون تعداد کارهای دارای دیرکرد در مسأله تک‌ماشین با فرضیات پایه را حداقل می‌کند [۳۱]. محاسبه مدت زمان پردازش هر J_i از مجموعه δ' در زمان $Max\{T, r_i\}$ علاوه بر ثابت ماندن زمان پردازش‌ها، کوچک‌تر بودن این زمان‌ها از زمان‌های واقعی را نیز

$$T(1 + b_i + b_j + b_i b_j) < r_j(1 + b_i + b_j + b_i b_j) \quad (21)$$

همچنین با تکیه بر رابطه $a_i b_j \leq a_j b_i$ رابطه زیر برقرار خواهد بود.

$$T(1 + b_i + b_j + b_i b_j) + a_i b_j < \quad (22)$$

$$r_j(1 + b_i + b_j + b_i b_j) + a_j b_i$$

وجود رابطه (۲۲) و در نظر گرفتن روابط (۱۶) و (۲۰) باعث برقراری رابطه زیر می‌گردد.

$$C_j(\delta_1) \leq C_i(\delta_2) \quad (23)$$

باتوجه به فرض زودکردار شدن هر دو کار J_i و J_j در توالی δ_1 رابطه زیر برقرار است.

$$N_T(\delta_1) = N_T(\delta_0) + N_T(\delta_{Tardy}) \quad (24)$$

از طرفی ممکن است J_i و J_j در توالی δ_2 دیرکردار گردد بنابراین رابطه زیر برای این توالی برقرار می‌گردد.

$$N_T(\delta_0) + N_T(\delta_{Tardy}) \leq N_T(\delta_2) \leq N_T(\delta_0) + N_T(\delta_{Tardy}) + 2 \quad (25)$$

بامقایسه روابط (۲۴) و (۲۵) رابطه زیر برقرار خواهد شد.

$$N_T(\delta_1) \leq N_T(\delta_2) \quad (26)$$

باتوجه به برقراری دو رابطه (۲۳) و (۲۶) یعنی $C_j(\delta_1) \leq C_i(\delta_2)$ و $N_T(\delta_1) \leq N_T(\delta_2)$ به طور هم‌زمان، ثابت می‌گردد که توالی δ_1 بر δ_2 غالب می‌باشد.

لم ۲. اگر هر دو کار J_i و J_j در توالی δ_1 دارای زودکرد باشند و روابط $a_i b_j \leq a_j b_i$ و $T > Max\{r_j, r_i\}$ برقرار باشد آنگاه در توالی بهینه J_i قبل از J_j زمان‌بندی می‌شود.

لم ۳. اگر هر دو کار J_i و J_j در توالی δ_1 دارای زودکرد باشند و روابط $r_j < T < r_i$ ، $a_j b_i - a_i b_j + (1 + b_i)(1 + b_j)(T - r_i) > 0$ و $a_j + (1 + b_j)T > r_i$ برقرار باشد آنگاه در توالی بهینه J_i قبل از J_j زمان‌بندی می‌شود.

لم ۴. اگر هر دو کار J_i و J_j در توالی δ_1 دارای زودکرد باشند و روابط $r_i < T$ و $a_i + (1 + b_i)T < r_j$ برقرار باشد آنگاه در توالی بهینه J_i قبل از J_j زمان‌بندی می‌شود.

لم ۵. اگر هر دو کار J_i و J_j در توالی δ_1 دارای زودکرد باشند و روابط $T < r_i$ و $a_i + (1 + b_i)r_i < r_j$ برقرار باشد آنگاه در توالی بهینه J_i قبل از J_j زمان‌بندی می‌شود.

همان‌طور که قبلاً اشاره گردید در هر گره از درخت جستجو اگر کار انتخاب شده از مجموعه δ' جهت زمان‌بندی دارای دیرکرد شد براساس نکته ۱ زمان‌بندی می‌شود.

نکته ۱. برای هر J_i دلخواه از مجموعه δ' جهت زمان‌بندی در انتهای مجموعه δ_{Early} ، اگر رابطه $d_i < a_i + (1 + b_i)Max\{T, r_i\}$ برقرار باشد J_i در مجموعه δ_{Tardy} قرار گرفته و یک واحد به مقدار تابع هدف اضافه می‌گردد مادامی که δ_{Early} تغییر نکند.

اثبات: براساس تعریف متغیرهای تصمیم، زمان شروع J_i براساس رابطه زیر محاسبه می‌گردد.

$$S_i = Max\{T, r_i\} \quad (27)$$

قدم ۳. M تا از کوچکترین نرخ زوالها را انتخاب و به صورت نزولی به موقعیت‌های ۱ تا M تخصیص دهید. با در نظر گرفتن C به عنوان زمان شروع موقعیت اول، موقعیت‌ها را محاسبه کنید.

قدم ۴. اگر مجموعه موعدها تحویل‌ها تهی است $N_T(\delta')$ به عنوان خروجی الگوریتم است و توقف. در غیر این صورت کمترین موعدها تحویل را از این مجموعه انتخاب و آن را از مجموعه حذف کنید. موعدها تحویل را با زمان تکمیل موقعیت M مقایسه کنید. اگر موقعیت M دارای دیرکرد شد قرار دهید $N_T(\delta') = N_T(\delta') + 1$ و مجدداً این قدم را تکرار کنید. در غیر این صورت قرار دهید $M = M + 1$ و به قدم ۲ بروید. با این توضیحات می‌توان قضیه ۲ را به عنوان حد پایین مسأله معرفی نمود.

قضیه ۲. مقدار حد پایین LB_2 برای مسأله مورد نظر براساس رابطه زیر می‌باشد که مقدار $N_T(\delta')$ از اجرای الگوریتم تخمین $N_T(\delta')$ محاسبه می‌شود.

$$LB_2 = N_T(\delta) + N_T(\delta') \quad (30)$$

اثبات: همان‌طور که قبلاً اشاره شد $N_T(\delta')$ تخمینی از تعداد کارهای دارای دیرکرد در مجموعه δ' می‌باشد که با اجرای الگوریتم تخمین $N_T(\delta')$ و براساس تعیین حداقل زمان تکمیل برای هر موقعیت و مقایسه آن با کوچکترین موعدها تحویل به دست می‌آید. با توجه به قاعده بهینه a_i/b_i در مسأله $1 | P_i = a_i + b_i S_i | C_{max}$ ، اگر این قاعده برای هر موقعیت M بعد از توالی δ_{Early} نیز رعایت گردد حداقل زمان تکمیل برای آن موقعیت محاسبه می‌شود که روند این محاسبه قبل از الگوریتم تخمین $N_T(\delta')$ تشریح گردید. پس از محاسبه کمترین زمان تکمیل برای موقعیت M ، حداقل موعدها تحویل موجود با آن مقایسه می‌شود. اگر این موقعیت دارای دیرکرد شد از آنجا که کمترین زمان تکمیل را داشته است موقعیت‌های بعد از M نیز توسط این موعدها تحویل قطعاً دارای دیرکرد خواهند شد. این بدین معناست که تعداد کارهای دارای دیرکرد حداقل برابر با تعداد موعدها تحویل‌هایی است که توسط آن‌ها موقعیت M دارای دیرکرد شده است. بنابراین تعداد کارهای دارای دیرکرد واقعی در مجموعه δ' از مقدار تخمینی $N_T(\delta')$ کمتر نخواهد بود. بدیهی است که هر توالی دلخواه از کارهای مجموعه δ' ، تأثیری روی مقدار $N_T(\delta)$ ندارد. با توجه به این توضیحات ثابت می‌شود که تابع هدف هر توالی کامل از مقدار LB_2 کمتر نیست و LB_2 یک حد پایین برای مسأله می‌باشد.

امکان ارائه یک حد پایین دیگر برای مسأله وجود دارد. اگر هر J_i در مجموعه δ' بعد از توالی جزئی δ_{Early} و در زمان $Max\{T, r_i\}$ زمان بندی گردد می‌توان تعداد کارهای دارای دیرکرد را در مجموعه δ' تخمین زد که در قالب قضیه ۳ ارائه می‌گردد.

قضیه ۳. مقدار حد پایین LB_3 برای مسأله مورد نظر براساس رابطه زیر محاسبه می‌گردد.

$$LB_3 = N_T(\delta) + N_T(\delta') \quad (31)$$

که مقدار $N_T(\delta')$ معادل با تعداد کارهای دارای دیرکرد مجموعه δ' با زمان بندی هر J_i در زمان $Max\{T, r_i\}$ می‌باشد.

به همراه خواهد داشت. بنابراین با رهسازی زوال واقعی، قسمت دوم مسأله اصلی به یک زیرمسأله تک ماشین پایه با زمان‌های پردازش کوچکتر تبدیل می‌شود. با اجرای الگوریتم مور و هاجسون مقدار بهینه تعداد کارهای دارای دیرکرد برای این زیرمسأله با مقدار $N_T(\delta')$ محاسبه می‌گردد. با توجه به بهینه بودن $N_T(\delta')$ برای این زیرمسأله و همچنین کوچکتر بودن زمان پردازش در آن، می‌توان نتیجه گرفت که این مقدار بهینه از تعداد کارهای دارای دیرکرد واقعی در قسمت دوم مسأله اصلی کمتر خواهد بود. این بدان معناست که مقدار LB_1 کوچکتر از مقدار تابع هدف برای یک توالی کامل در مسأله اصلی می‌باشد بنابراین ثابت می‌گردد که رابطه (۲۹) یک حد پایین برای مسأله اصلی است.

اگر بتوان کمترین زمان تکمیل را برای هر موقعیت بعد از توالی δ_{Early} محاسبه نمود، با مقایسه آن با کمترین موعدها تحویل موجود، امکان ارائه یک حد پایین دیگر برای مسأله وجود دارد. همان‌طور که قبلاً اشاره شد براون و یچیالی [۱۲] در تحقیق خود نشان دادند که مسأله $1 | P_i = a_i + b_i S_i | C_{max}$ با توالی حاصل از نرخ غیرکاهشی a_i/b_i بهینه می‌شود. اگر این قاعده برای هر موقعیت M بعد از توالی δ_{Early} در مسأله $1 | r_i, P_i = a_i + b_i S_i | \sum U_i$ نیز رعایت گردد حداقل زمان تکمیل برای آن موقعیت به دست می‌آید. بدین منظور در ابتدا کمترین مقدار a_i باقی مانده در مجموعه δ' انتخاب و به موقعیت M تخصیص می‌یابد. مقدار a_i برای موقعیت M ، بزرگتر مساوی مقدار a_i برای موقعیت‌های قبل از آن می‌باشد و برای ایجاد نرخ غیرکاهشی a_i/b_i لازم است کمترین نرخ زوال به موقعیت M تخصیص یابد. بنابراین M تا از کوچکترین نرخ زوال‌های مجموعه δ' انتخاب و به صورت نزولی به موقعیت‌های ۱ تا M تخصیص می‌یابد. با در نظر گرفتن $Max\{T, Min\{r_i\}\}$ به عنوان زمان شروع موقعیت اول و محاسبه زمان تکمیل موقعیت‌های ۱ تا $M-1$ ، زمان تکمیل موقعیت M محاسبه می‌گردد. جهت تخمین تعداد کارهای دیرکردار لازم است پس از محاسبه زمان تکمیل موقعیت M ، یک موعدها تحویل مناسب با آن مقایسه گردد. کمترین موعدها تحویل موجود انتخاب و با زمان تکمیل موقعیت M مقایسه می‌گردد. اگر این موقعیت توسط این موعدها تحویل دارای دیرکرد شد یک واحد به تعداد کارهای دارای دیرکرد اضافه می‌شود، موعدها تحویل از مجموعه حذف و موعدها تحویل بعدی انتخاب و مجدداً با زمان تکمیل موقعیت M مقایسه می‌گردد. این فرآیند انتخاب و مقایسه تازمانی ادامه می‌یابد که موقعیت مورد نظر توسط موعدها تحویل انتخاب شده دارای دیرکرد نشود. در این حالت موعدها تحویل از مجموعه حذف و موقعیت بعدی انتخاب و جهت محاسبه زمان تکمیل، روند فوق مجدداً تکرار می‌گردد. این روند به صورت خلاصه در الگوریتم تخمین $N_T(\delta')$ ارائه شده است.

الگوریتم تخمین $N_T(\delta')$

قدم ۱. قرار دهید $C = Max\{T, Min\{r_i\}\}$ ، $N_T(\delta') = 0$ و $M=1$

قدم ۲. کمترین مقدار a_i را از مجموعه زمان پردازش‌های پایه انتخاب و به موقعیت M تخصیص دهید. a_i را از این مجموعه حذف کنید.

آن به تفکیک ۱۲ دسته در جدول (۲) ارائه شده است.

در جدول (۲) تعداد نمونه بهینه و متوسط زمان حل در مدل ریاضی (A1)، الگوریتم شاخه‌وکران (A2) و الگوریتم شاخه‌وکران با در نظر گرفتن اصول غلبه و حدود پایین پیشنهادی (A3) گزارش شده است. همچنین متوسط تعداد گره‌های بررسی شده در دو حالت A2 و A3 و متوسط درصد کل گره‌های قطع شده در A3 نیز در جدول (۲) خلاصه شده است. از آنجا که متوسط درصد کل گره‌های قطع شده در حالت A2 تقریباً برابر با صفر می‌باشد از ارائه آن در جدول خودداری شده است. ستون تعداد نمونه بهینه نشان می‌دهد که مدل ریاضی تنها قادر به حل تمام مسائل با ابعاد ۵ فعالیت و برخی مسائل با ابعاد ۱۰ فعالیت می‌باشد و هیچ‌یک از مسائل با ابعاد ۱۵ فعالیت را حل نمی‌نماید. برای حالت A2 که مربوط به الگوریتم شاخه‌وکران بدون در نظر گرفتن اصول غلبه و حدود پایین است نیز علاوه بر مسائل با ابعاد ۵ فعالیت، برخی از مسائل با ابعاد ۱۰ فعالیت و اندکی از مسائل با ابعاد ۱۵ فعالیت حل گردیده ولی هیچ‌کدام از مسائل با ابعاد ۲۰ فعالیت حل نشده است. این در حالی است که تعداد نمونه بهینه در حالت A3 نشان می‌دهد که الگوریتم شاخه‌وکران پیشنهادی اکثر مسائل با ابعاد ۳۰ فعالیت را حل می‌نماید. همچنین مقایسه ستون مربوط به متوسط زمان حل نیز نشان می‌دهد که حالت A1 و A2 برای حل مسائل به صورت بهینه نیاز به زمان بسیار بالا و قابل توجهی دارند در صورتی که متوسط زمان حل در حالت A3 بسیار پایین است. بنابراین با مقایسه تعداد نمونه بهینه و متوسط زمان حل در حالت‌های A1 و A3 می‌توان به درجه پیچیدگی مسئله پی برد. نتایج در جدول (۲) نشان می‌دهد که متوسط تعداد گره‌های بررسی شده در حالت A2 در مقایسه با حالت A3 به شدت بیشتر است. همچنین متوسط درصد کل گره‌های قطع شده در A2 تقریباً صفر می‌باشد در صورتی که این شاخص برای حالت A3 حداقل برابر با ۸۵ درصد است. با مقایسه تعداد نمونه بهینه، متوسط زمان حل، متوسط تعداد گره‌های بررسی شده و متوسط درصد کل گره‌های قطع شده در دو حالت A2 و A3 می‌توان کارایی الگوریتم شاخه‌وکران پیشنهادی و اثربخشی اصول غلبه و حدود پایین که برای آن در نظر گرفته شده را نتیجه گرفت.

نتایج نشان می‌دهد که الگوریتم شاخه‌وکران با در نظر گرفتن اصول غلبه و حدود پایین قادر به حل بهینه مسائل با ابعاد ۳۰ فعالیت در تمام دسته‌ها به جز در دسته‌های ۵، ۶ و ۹ در زمان مناسب می‌باشد که مقایسه آن با حالت A1 و A2 نشان‌دهنده کارایی و اثربخشی این الگوریتم می‌باشد. با تکیه بر ادبیات موضوع و جدول (۱) می‌توان ادعا کرد که الگوریتم پیشنهادی در مقایسه با پژوهش‌های مرتبط کیفیت بالایی دارد. همان‌طور که قبلاً اشاره شد جعفری و مصلحی [۱۴] برای مسئله $\sum U_i | P_i = a_i + bS_i | 1$ توانستند مسائل با ابعاد ۲۸ فعالیت را به صورت بهینه حل نمایند. همچنین لی و همکاران [۱۳] نیز موفق به حل بهینه مسائل با ابعاد ۲۸ فعالیت برای مسئله $\sum C_{max} | bS_i | 1$ شدند. از آنجا که مسئله بررسی شده در این پژوهش دارای پیچیدگی بیشتری نسبت به این دو تحقیق می‌باشد بنابراین می‌توان

اثبات: با فرض زمان‌بندی کارها در زمان $Max\{T, r_i\}$ هر J_i از مجموعه δ' در زودترین زمان ممکن شروع می‌شود و این کمترین زمان تکمیل را برای آن کار به همراه خواهد داشت. اگر J_i با این فرض دارای دیرکرد شود به دلیل افزایش زمان شروع در موقعیت‌های بعدی و حتی در توالی واقعی نیز دارای دیرکرد خواهد بود. اگر J_i دارای زودکرد گردد ممکن است در موقعیت‌های بعدی و در توالی واقعی دارای دیرکرد می‌توان نتیجه گرفت که تعداد کارهای دارای دیرکرد واقعی در مجموعه δ' از مقدار تخمینی $N_T(\delta')$ کمتر نخواهد بود. از آنجا که زمان‌بندی دلخواه کارهای مجموعه δ' تأثیری روی مقدار $N_T(\delta)$ ندارد می‌توان ادعا کرد که مقدار LB_3 کوچک‌تر از مقدار تابع هدف برای یک توالی کامل در مسئله می‌باشد و این بیان‌کننده حد پایین بودن LB_3 برای مسئله می‌باشد.

باتوجه به تعریف سه حد پایین برای مسئله، حداکثر مقدار آن به عنوان حد پایین هر گره انتخاب می‌شود که براساس رابطه (۳۲) محاسبه می‌شود.

$$LB = Max\{LB_1, LB_2, LB_3\} \quad (32)$$

۵. نتایج محاسباتی

جهت ارزیابی مدل ریاضی و الگوریتم شاخه‌وکران پیشنهادی مجموعه‌ای از مسائل تصادفی با تکیه بر ادبیات موضوع تولید و حل شده‌اند. مدل مسئله در نرم‌افزار GAMS و الگوریتم شاخه‌وکران در محیط برنامه‌نویسی ++C کدنویسی شده و مسائل بر روی کامپیوتر ۵ هسته‌ای با سرعت ۳/۴ گیگاهرتز و رم ۸ گیگابایت حل شده‌اند. در الگوریتم شاخه‌وکران محدودیت زمانی ۴۰۰۰ ثانیه برای حل هر مسئله در نظر گرفته شده به طوری که اگر مسئله‌ای تا این زمان به جواب بهینه نرسد رویه حل برای آن مسئله متوقف شده و به عنوان مسئله حل نشده در نتایج ارائه می‌گردد. نتایج محاسباتی در این تحقیق شامل چهار مرحله می‌باشد که در تمام آن‌ها زمان پردازش پایه هر کار از توزیع یکنواخت گسسته در بازه (۰، ۱۰۰) و زمان ورود کارها نیز از توزیع یکنواخت پیوسته در بازه (0, 50.5nλ) تولید می‌گردد که n تعداد کارها و λ متغیر کنترل می‌باشد [۱۳].

در مرحله اول نتایج محاسباتی عملکرد مدل ریاضی، الگوریتم شاخه‌وکران و الگوریتم شاخه‌وکران با در نظر گرفتن اصول غلبه و حدود پایین پیشنهادی بررسی می‌شود که برای سهولت در ارائه نتایج به ترتیب با عنوان A1، A2 و A3 نام‌گذاری می‌شود. تعداد کارها برابر با ۵، ۱۰، ۱۵، ۲۰، ۲۵ و ۳۰ فرض شده، نرخ زوال براساس توزیع یکنواخت پیوسته در بازه (۰، ۱) تولید می‌شود و برای متغیر کنترل λ نیز مقادیر ۰/۲، ۱ و ۳ در نظر گرفته شده است. موعد تحویل نیز برای هر کار به صورت تصادفی از توزیع یکنواخت پیوسته در چهار بازه $[0, .5C_{max}]$ ، $[.5C_{max}, C_{max}]$ و $(0, C_{max})$ و $(0, 1.5C_{max})$ که به ترتیب $d1$ تا $d4$ نامیده می‌شوند تولید می‌گردد و در آن C_{max} مقدار دامنه عملیات توالی ایجاد شده براساس نسبت a_i/b_i می‌باشد [۳۲]. برای هر حالت ۵۰ مسئله به صورت تصادفی تولید و حل می‌گردد که نتایج

و در مراحل بعدی نتایج تشریح می‌گردد. ستون متوسط درصد کل گره‌های قطع شده بیان‌کننده نسبت تعداد گره‌های قطع شده به کل گره‌های بررسی شده می‌باشد و این مقدار حداقل برابر با ۸۵/۶۱ درصد است که کارایی بالای الگوریتم پیشنهادی را نشان می‌دهد. البته این شاخص در دسته‌های مختلف متفاوت است و به مقادیر λ نرخ زوال و موعد تحویل وابسته می‌باشد که در ادامه تأثیر این متغیرها روی الگوریتم پیشنهادی تحقیق می‌گردد.

نتیجه گرفت که الگوریتم پیشنهادی و ابعاد مسائل حل شده شرایط مناسبی دارند. در دسته‌های ۵، ۶ و ۹ که الگوریتم موفق به حل تمام مسائل با ابعاد ۳۰ فعالیت نشده است یا موعد تحویل از بازه $[.5C_{max}, C_{max}]$ تولید گردیده و یا مقدار λ برابر با ۳ فرض شده است. همچنین با افزایش مقدار m تعداد گره‌ها و زمان حل مسائل افزایش می‌یابد به طوری که بدترین حالت در دسته ۶ اتفاق افتاده است که به طور هم‌زمان دارای موعد تحویل $[.5C_{max}, C_{max}]$ و مقدار λ برابر با ۳ است و جزء سخت‌ترین دسته مسائل می‌باشد که دلایل آن در ادامه

جدول (۲): عملکرد الگوریتم شاخه و کران در دسته‌های مختلف

متوسط درصد کل گره‌های قطع شده در	متوسط تعداد گره‌های بررسی شده در		متوسط زمان حل در			تعداد نمونه بهینه در			بازه موعد تحویل	λ	n	دسته
	A3	A2	A3	A2	A1	A3	A2	A1				
	A3	A3	A2	A1	A3	A2	A1					
۹۴/۵۰	۸/۳۰E+۰۰	۱/۰۸E+۰۲	۰/۰۰	۶/۵۱	۱۰/۰۴	۵۰	۵۰	۵۰	d1	۰/۲	۵	۱
۹۳/۸۸	۶/۹۲E+۰۲	۳/۲۷E+۰۶	۰/۰۱	۴۰۹/۶۱	۷۵۹/۲۰	۵۰	۴۸	۴۴			۱۰	
۹۲/۱۰	۴/۵۷E+۰۶	۱/۱۸E+۱۲	۰/۲۳	۳۲۲۷/۰۳	---	۵۰	۸	۰			۱۵	
۹۴/۵۲	۹/۱۱E+۱۰	---	۱۴/۷۵	---	---	۵۰	۰	۰			۲۰	
۹۲/۶۶	۵/۸۵E+۱۴	---	۵۱/۸۳	---	---	۵۰	۰	۰			۲۵	
۹۵/۰۳	۱/۲۰E+۱۶	---	۴۲۰/۶۷	---	---	۵۰	۰	۰	۳۰			
۹۰/۶۷	۱/۲۲E+۰۱	۱/۰۹E+۰۲	۰/۰۰	۱۰/۳۵	۱۲/۴۶	۵۰	۵۰	۵۰	d1	۱	۵	۲
۹۲/۱۴	۹/۷۶E+۰۲	۳/۳۰E+۰۶	۰/۰۳	۸۰۷/۰۰	۱۰۰۱/۴۷	۵۰	۴۳	۴۰			۱۰	
۹۰/۲۹	۸/۴۶E+۰۶	۱/۱۹E+۱۲	۲/۴۶	۳۵۱۹/۲۷	---	۵۰	۶	۰			۱۵	
۹۱/۶۵	۷/۵۶E+۱۱	---	۲۱/۰۴	---	---	۵۰	۰	۰			۲۰	
۹۲/۰۴	۸/۷۴E+۱۵	---	۹۷/۲۶	---	---	۵۰	۰	۰			۲۵	
۹۰/۳۲	۶/۵۲E+۱۷	---	۸۰۴/۸۱	---	---	۵۰	۰	۰	۳۰			
۹۰/۸۸	۲/۴۵E+۰۱	۱/۱۲E+۰۲	۰/۰۲	۱۲/۹۱	۱۱/۲۳	۵۰	۵۰	۵۰	d1	۳	۵	۳
۹۲/۵۱	۲/۳۴E+۰۳	۳/۳۷E+۰۶	۲/۰۹	۹۵۱/۶۶	۱۴۲۰/۵۱	۵۰	۴۰	۳۸			۱۰	
۹۱/۶۰	۴/۳۱E+۰۷	۱/۲۲E+۱۲	۸/۶۷	۳۴۲۷/۴۵	---	۵۰	۶	۰			۱۵	
۹۰/۷۳	۳/۳۳E+۱۲	---	۵۹/۳۱	---	---	۵۰	۰	۰			۲۰	
۸۹/۰۳	۵/۰۹E+۱۶	---	۳۷۱/۰۶	---	---	۵۰	۰	۰			۲۵	
۹۰/۲۴	۳/۲۷E+۱۸	---	۱۲۴۰/۱۲	---	---	۵۰	۰	۰	۳۰			
۹۳/۰۰	۳/۸۲E+۰۱	۱/۱۴E+۰۲	۰/۱۵	۱۳/۶۰	۱۴/۰۳	۵۰	۵۰	۵۰	d2	۰/۲	۵	۴
۹۰/۵۸	۶/۳۸E+۰۳	۳/۴۵E+۰۶	۲/۶۴	۱۰۹۴/۸۱	۱۳۳۸/۱۰	۵۰	۴۱	۴۰			۱۰	
۹۱/۱۳	۸/۹۴E+۰۷	۱/۲۴E+۱۲	۷/۲۱	۳۵۰۷/۲۵	---	۵۰	۷	۰			۱۵	
۸۹/۲۶	۹/۶۵E+۱۲	---	۶۹/۴۴	---	---	۵۰	۰	۰			۲۰	
۸۹/۰۴	۸/۷۳E+۱۷	---	۷۸۱/۲۷	---	---	۵۰	۰	۰			۲۵	
۹۱/۶۳	۹/۸۵E+۱۹	---	۱۲۳۶/۰۱	---	---	۵۰	۰	۰	۳۰			
۹۱/۱۲	۵/۱۴E+۰۱	۱/۱۶E+۰۲	۱/۰۵	۲۱/۵۰	۱۹/۴۴	۵۰	۵۰	۵۰	d2	۱	۵	۵
۸۹/۶۳	۸/۵۴E+۰۳	۳/۵۲E+۰۶	۳/۱۲	۱۴۰۰/۰۳	۱۶۷۲/۶۷	۵۰	۳۹	۳۵			۱۰	
۹۰/۳۱	۵/۴۳E+۰۸	۱/۲۷E+۱۲	۱۰/۶۲	۳۴۱۷/۴۹	---	۵۰	۴	۰			۱۵	
۹۰/۱۴	۸/۲۴E+۱۳	---	۱۰۹/۸۷	---	---	۵۰	۰	۰			۲۰	
۹۱/۸۸	۹/۱۴E+۱۸	---	۱۰۰۲/۲۵	---	---	۵۰	۰	۰			۲۵	
۸۹/۷۲	۷/۱۵E+۲۱	---	۱۷۸۴/۱۰	---	---	۴۷	۰	۰	۳۰			

ادامه جدول (۲): عملکرد الگوریتم شاخه‌وکران در دسته‌های مختلف

دسته	n	λ	بازه موعده تحویل	تعداد نمونه بهینه در			متوسط زمان حل در			متوسط تعداد گره بررسی شده در		متوسط درصد کل گره‌های قطع شده در A3
				A3	A2	A1	A3	A2	A1	A3	A2	
				۵	۵۰	۵۰	۵۰	۳/۳۷	۱۶/۸۰	۱۸/۵۴	۷/۵۳E+۰۱	
۱۰	۵۰	۳۲	۳۸	۷/۶۱	۱۶۴۰/۲	۱۹۰۴/۱	۹/۰۱E+۰۴	۳/۵۹E+۰۶	۸۷/۱۴			
۱۵	۵۰	۴	۰	۲۹/۵۰	۳۸۱۰/۲	---	۴/۰۷E+۰۹	۱/۲۹E+۱۲	۸۶/۳۷			
۲۰	۵۰	۰	۰	۶۰۲/۰۴	---	---	۶/۹۶E+۱۴	---	۸۹/۲۸			
۲۵	۵۰	۰	۰	۱۶۹۹/۷	---	---	۴/۰۲E+۲۰	---	۹۰/۷۵			
۳۰	۴۵	۰	۰	۳۱۳۴/۹	---	---	۵/۶۴E+۲۳	---	۸۷/۲۰			
۵	۵۰	۵۰	۵۰	۰/۰۹	۱۴/۶۰	۱۳/۲۰	۱/۸۷E+۰۱	۱/۰۹E+۰۲	۹۲/۰۵			
۱۰	۵۰	۳۵	۳۸	۱/۲۴	۱۵۸۳/۱	۱۸۱۲/۹	۶/۰۳E+۰۳	۳/۳۴E+۰۶	۹۰/۳۳			
۱۵	۵۰	۵	۰	۴/۸۶	۳۴۷۱/۰	---	۵/۲۱E+۰۷	۱/۲۰E+۱۲	۹۱/۵۵			
۲۰	۵۰	۰	۰	۳۵/۷۷	---	---	۶/۷۱E+۱۱	---	۹۲/۴۳			
۲۵	۵۰	۰	۰	۴۲۵/۹۵	---	---	۹/۳۱E+۱۵	---	۹۲/۶۵			
۳۰	۵۰	۰	۰	۸۷۸/۵۵	---	---	۴/۵۱E+۱۸	---	۹۰/۱۴			
۵	۵۰	۵۰	۵۰	۰/۲۳	۱۳/۷۰	۱۴/۰	۲/۳۷E+۰۱	۱/۱۲E+۰۲	۹۰/۱۸			
۱۰	۵۰	۴۲	۳۷	۲/۴۰	۱۸۴۳/۶	۱۷۰۹/۲	۷/۲۴E+۰۳	۳/۳۷E+۰۶	۹۱/۲۳			
۱۵	۵۰	۲	۰	۷/۳۱	۳۹۱۲/۹	---	۲/۹۱E+۰۸	۱/۲۲E+۱۲	۸۹/۸۴			
۲۰	۵۰	۰	۰	۶۴/۲۲	---	---	۷/۳۷E+۱۲	---	۹۱/۷۷			
۲۵	۵۰	۰	۰	۷۶۲/۳۴	---	---	۳/۲۷E+۱۶	---	۹۲/۱۸			
۳۰	۵۰	۰	۰	۱۲۳۶/۷	---	---	۷/۲۳E+۲۰	---	۹۰/۲۰			
۵	۵۰	۵۰	۵۰	۰/۹۵	۱۶/۰۰	۱۶/۳	۳/۷۵E+۰۱	۱/۱۴E+۰۲	۹۲/۳۳			
۱۰	۵۰	۴۰	۴۰	۳/۱۷	۱۷۸۲/۹	۱۸۲۰/۴	۲/۳۳E+۰۴	۳/۴۵E+۰۶	۹۱/۱۷			
۱۵	۵۰	۵	۰	۱۴/۸۸	۳۶۱۴/۶	---	۶/۸۲E+۰۸	۱/۲۴E+۱۲	۸۹/۷۸			
۲۰	۵۰	۰	۰	۱۷۳/۱۱	---	---	۳/۰۱E+۱۳	---	۹۰/۶۴			
۲۵	۵۰	۰	۰	۱۰۴۹/۶	---	---	۸/۰۳E+۱۷	---	۸۹/۲۲			
۳۰	۴۸	۰	۰	۱۸۴۱/۹	---	---	۳/۵۶E+۲۱	---	۹۰/۰۹			
۵	۵۰	۵۰	۵۰	۰/۰۰	۱۰/۷۰	۱۱/۴۰	۲/۱۲E+۰۰	۹/۶۰E+۰۲	۹۶/۵۰			
۱۰	۵۰	۴۰	۴۱	۰/۰۰	۱۲۳۶/۲	۱۶۴۲/۷	۲/۱۴E+۰۲	۲/۹۰E+۰۶	۹۷/۱۵			
۱۵	۵۰	۱۸	۰	۰/۰۱	۳۱۶۲/۰	---	۱/۶۳E+۰۳	۱/۰۱E+۱۲	۹۵/۲۹			
۲۰	۵۰	۰	۰	۳/۹۱	---	---	۵/۱۵E+۰۵	---	۹۵/۸۸			
۲۵	۵۰	۰	۰	۱۵/۸۴	---	---	۹/۸۱E+۰۸	---	۹۹/۰۶			
۳۰	۵۰	۰	۰	۲۰۳/۲۵	---	---	۱/۹۸E+۱۰	---	۹۷/۴۰			
۵	۵۰	۵۰	۵۰	۰/۰۰	۱۲/۶۳	۱۰/۲۰	۶/۷۲E+۰۰	۱/۰۱E+۰۲	۹۷/۶۷			
۱۰	۵۰	۴۲	۳۸	۰/۰۲	۱۳۵۵/۳	۱۵۹۰/۱	۳/۰۱E+۰۲	۳/۰۸E+۰۶	۹۴/۲۱			
۱۵	۵۰	۱۶	۰	۱/۱۲	۳۳۸۵/۲	---	۸/۱۳E+۰۳	۱/۱۱E+۱۲	۹۵/۵۴			
۲۰	۵۰	۰	۰	۷/۸۸	---	---	۳/۶۵E+۰۶	---	۹۲/۳۶			
۲۵	۵۰	۰	۰	۴۰/۲۹	---	---	۱/۹۲E+۰۹	---	۹۰/۳۷			
۳۰	۵۰	۰	۰	۵۰۰/۴	---	---	۲/۴۰E+۱۲	---	۹۱/۰۴			
۵	۵۰	۵۰	۵۰	۰/۰۰	۱۳/۴۵	۱۱/۶۱	۱/۰۵E+۰۱	۱/۰۶E+۰۲	۹۶/۱۹			
۱۰	۵۰	۴۳	۴۰	۰/۷۳	۱۴۳۰/۹	۱۶۲۷/۵	۹/۰۳E+۰۳	۳/۱۹E+۰۶	۹۲/۷۰			
۱۵	۵۰	۱۲	۰	۴/۳۰	۳۴۲۷/۷	---	۳/۱۳E+۰۵	۱/۱۵E+۱۲	۹۰/۰۸			
۲۰	۵۰	۰	۰	۲۵/۰۹	---	---	۶/۰۴E+۰۸	---	۹۱/۸۹			
۲۵	۵۰	۰	۰	۱۳۴/۵۷	---	---	۶/۰۵E+۱۰	---	۸۹/۵۱			
۳۰	۵۰	۰	۰	۹۶۸/۷۲	---	---	۴/۶۹E+۱۴	---	۱۲/۶۲			

آزمون در جدول (۳) خلاصه گردیده است. باتوجه به مقادیر Sig که برابر با $0/000$ می باشد می توان بیان کرد که حالت های $A1$ ، $A2$ و $A3$ تأثیر بسیار معناداری روی تعداد نمونه بهینه و متوسط زمان حل مسأله دارند به طوری که تعداد نمونه بهینه در حالت $A3$ نسبت به $A2$ و $A2$ نسبت به $A1$ بیشتر می باشد. همچنین متوسط زمان حل نیز در حالت $A3$ نسبت به دو حالت دیگر بسیار کمتر می باشد. این تحلیل آماری بیان کننده کارایی بالای الگوریتم پیشنهادی در این تحقیق می باشد به گونه ای که نتایج آن در مقایسه با دو حالت دیگر دارای اختلاف معناداری می باشد.

همان طور که اشاره گردید نتایج محاسباتی، کارایی بالای الگوریتم شاخه و کران با در نظر گرفتن اصول غلبه و حدود پایین پیشنهادی را نشان می دهد. به منظور بررسی این موضوع که آیا الگوریتم پیشنهادی دارای تأثیر معنادار می باشد یا خیر لازم است که تحلیل آماری دقیقی در این زمینه صورت گیرد. باتوجه به شرایط مسأله از آزمون تحلیل واریانس با تکیه بر دو شاخص تعداد نمونه بهینه و متوسط زمان حل در سه حالت $A1$ ، $A2$ و $A3$ بهره گرفته شده است. به عبارت دیگر تأثیر سه حالت $A1$ ، $A2$ و $A3$ روی تعداد نمونه بهینه و متوسط زمان حل بر اساس آزمون تحلیل واریانس بررسی شده است که حل نتایج این

جدول (۳): بررسی تأثیر حالت های $A1$ ، $A2$ و $A3$ روی تعداد نمونه بهینه و متوسط زمان حل مسأله

مقدار Sig	آماره F	میانگین مربع	درجه آزادی	مجموع مربعات		
0/000	11/87	25/16	2	50/32	بین گروهها	تأثیر سه حالت $A1$
		2/12	69	146/59	داخل گروهها	$A2$ و $A3$ روی تعداد
			71	196/91	کل	نمونه بهینه
0/000	15/28	28/87	2	57/73	بین گروهها	تأثیر سه حالت $A1$
		1/89	69	130/18	داخل گروهها	$A2$ و $A3$ روی متوسط
			71	187/91	کل	زمان حل نمونه بهینه

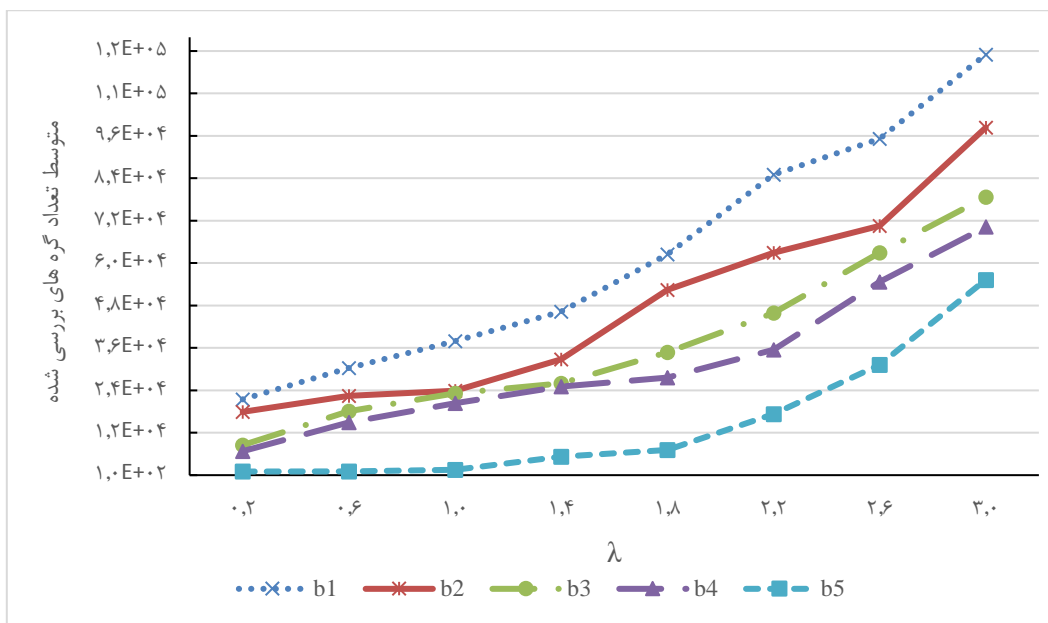
متوسط تعداد گره های بررسی شده و زمان حل را به همراه خواهد داشت.

تأثیر نرخ زوال روی عملکرد الگوریتم شاخه و کران در مرحله سوم از نتایج محاسباتی مورد بحث قرار می گیرد. نرخ زوال کارها از مقدار 0 تا 1 در قالب 10 بازه متفاوت $[0, 0/1]$ ، $(0/1, 0/2]$ و ... به طول 0/1 تولید می شود که به صورت $b1$ تا $b10$ نام گذاری می گردد. تعداد کارها و موعد تحویل آن ها مانند مرحله دوم می باشد و برای متغیر λ نیز برای هر حالت از نرخ زوال، مقادیر $0/2$ ، 1 و 3 فرض می شود. در این مرحله نیز برای هر حالت 1000 مسأله به صورت تصادفی تولید و حل شده است که متوسط زمان حل و تعداد گره های بررسی شده در الگوریتم به ترتیب در شکل (4) و (5) ارائه شده است.

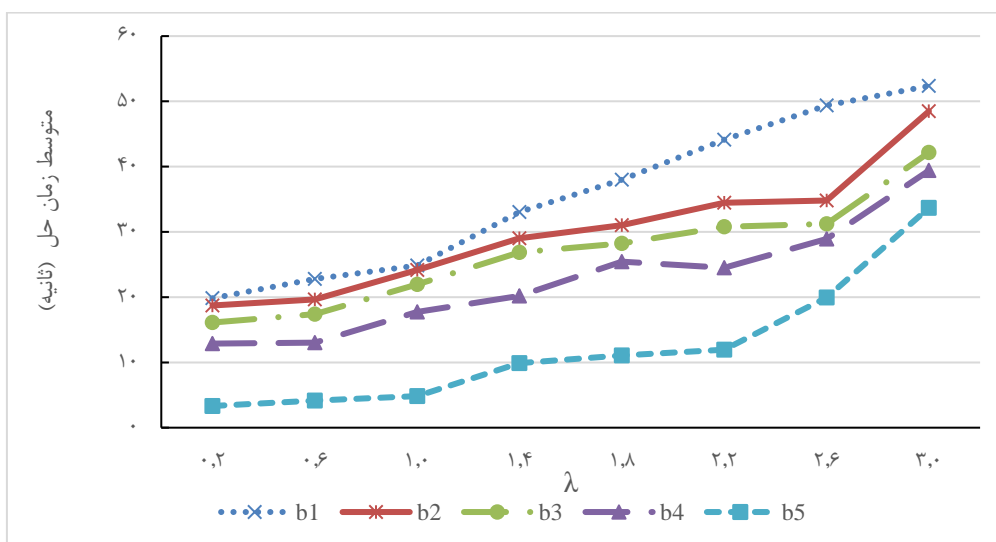
نتایج نشان می دهد که با افزایش مقدار نرخ زوال متوسط زمان حل و تعداد گره های بررسی شده کاهش می یابد که میزان کاهش آن برای λ بزرگ تر کمتر است. در مرحله دوم نشان داده شد که متوسط زمان حل و تعداد گره های بررسی شده برای مقادیر λ بزرگ تر بیشتر است و دلیل آن تشریح گردید که در این مرحله و بر اساس شکل (4) و (5) نیز این ادعا تایید می شود. دلیل کاهشی بودن دو پارامتر مورد نظر با افزایش نرخ زوال نیز به افزایش زمان تکمیل کارها و در نتیجه افزایش تعداد دفعات استفاده از اصول غلبه و حدود پایین در الگوریتم مربوط می شود. هرچه نرخ زوال افزایش یابد زمان تکمیل کارها نیز افزایش می یابد که این باعث افزایش احتمال قطع شدن گره ها می گردد. بنابراین تعداد گره های قطع شده افزایش و تعداد گره های بررسی شده کاهش می یابد و نتیجه آن کاهش زمان حل الگوریتم می باشد که در مرحله دوم نیز چنین روندی مشاهده گردید که مؤید این ادعا است.

در مرحله دوم نتایج محاسباتی، تأثیر متغیر کنترل λ روی عملکرد الگوریتم شاخه و کران بررسی می گردد که تعداد کارها برابر با مقدار ثابت 15 فرض شده و موعد تحویل آن ها نیز از بازه $[5C_{max}, C_{max}]$ تولید می شود. متغیر λ از 0/2 تا 3 مقدار دهی می شود و در هر مرحله با افزایش 0/4 مقدار بعدی ایجاد می گردد به طوری که 8 حالت برای آن در نظر گرفته می شود. نرخ زوال کارها نیز به ازای هر مقدار λ از 5 بازه $[0, 0/2]$ ، $(0/2, 0/4]$ ، $(0/4, 0/6]$ ، $(0/6, 0/8]$ ، $(0/8, 1]$ و $[1, 0/8]$ تولید می شود. در این مرحله برای هر حالت 1000 مسأله به صورت تصادفی تولید و حل گردیده است که نتایج آن بر اساس متوسط زمان حل و تعداد گره بررسی شده در الگوریتم شاخه و کران به ترتیب در شکل (2) و (3) نشان داده شده است.

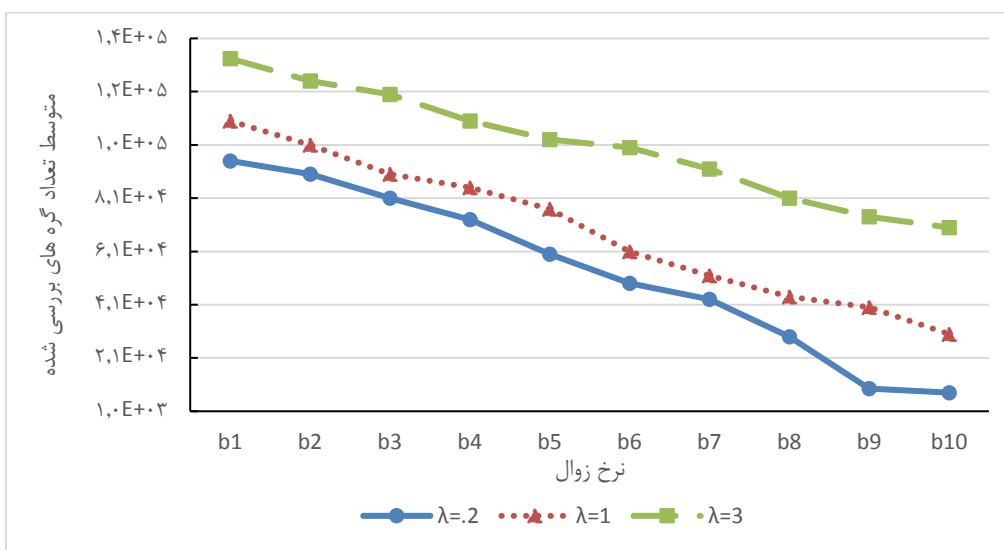
همان طور که از شکل (2) مشاهده می شود متوسط زمان حل الگوریتم با افزایش مقدار λ افزایش می یابد اگرچه این افزایش برای نرخ زوال های بزرگ تر کمتر می باشد ولی روند در تمام حالت افزایشی دارد. باتوجه به ثابت بودن تعداد کارها دلیل اصلی این روند به بازه تولید مقادیر زمان ورود کارها مربوط می شود. همان طور که قبلاً اشاره شد زمان ورود کارها به صورت تصادفی از بازه $[0, 50.5n\lambda]$ تولید می شود. با افزایش مقدار λ زمان ورود کارها از بازه بزرگ تری تولید می شود که این موضوع باعث کاهش تعداد دفعات استفاده از اصول غلبه و حدود پایین در الگوریتم می شود. این منجر به افزایش متوسط تعداد گره های بررسی شده و کاهش متوسط درصد کل گره های قطع شده می گردد که نتیجه آن افزایش متوسط زمان حل خواهد شد. روند موجود در شکل (3) که مشابه روند شکل (2) می باشد این ادعا را ثابت می کند که با افزایش مقدار λ مسأله سخت تر شده به طوری که افزایش



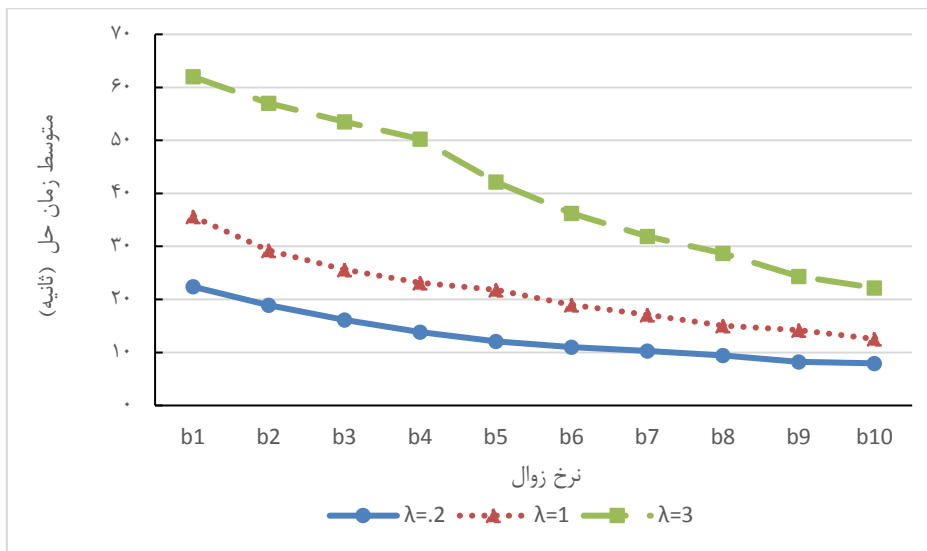
شکل (۲): متوسط تعداد گره‌های بررسی شده در الگوریتم شاخه‌وکران باتوجه به مقدار λ



شکل (۳): متوسط زمان حل الگوریتم شاخه‌وکران باتوجه به مقدار λ



شکل (۴): متوسط تعداد گره‌های بررسی شده در الگوریتم شاخه‌وکران باتوجه مقادیر نرخ زوال

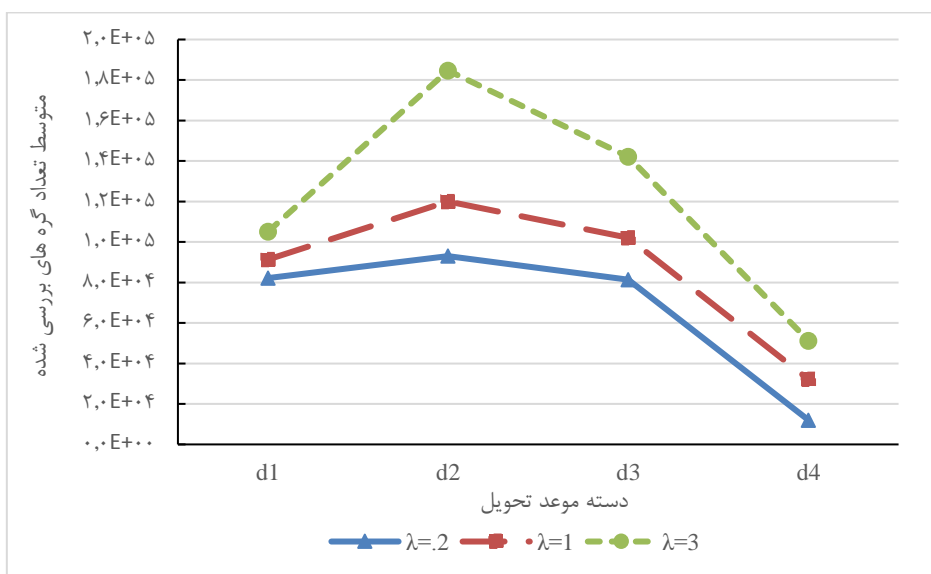


شکل (۵): متوسط زمان حل الگوریتم شاخه و کران با توجه به مقادیر نرخ زوال

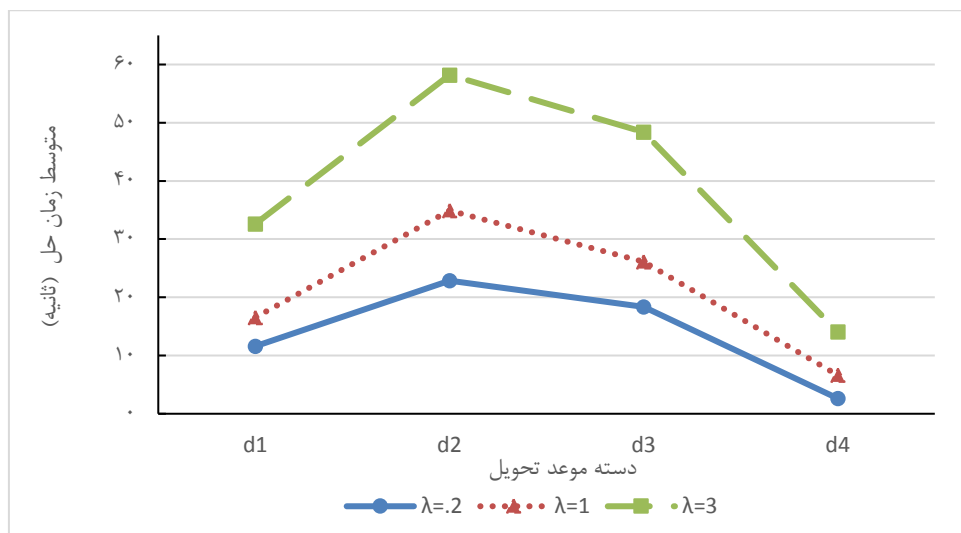
سایر مسائل کمتر بوده و به عبارتی این دسته مسائل به ترتیب جزء مسائل نسبتاً ساده و ساده می‌باشند. وقتی موعد تحویل کارها مقادیر بزرگ یا کوچک هستند مقدار تابع هدف به ترتیب بسیار کوچک یا بسیار بزرگ خواهد بود که این باعث افزایش تعداد دفعات استفاده از اصول غلبه و حدود پایین در الگوریتم و همچنین افزایش تعداد گره‌های قطع شده می‌شود. این موضوع در مسائل مربوط به گروه d1 و d4 اتفاق می‌افتد و با افزایش تعداد گره‌های قطع شده کاهش چشم‌گیر تعداد گره‌های بررسی شده و زمان حل در مسائل را به همراه دارد. از طرفی دیگر مسائلی که موعد تحویل کارها دارای مقادیر متوسط است سخت‌تر می‌باشد به طوری که متوسط زمان حل و تعداد گره‌های بررسی شده در شکل (۶) و (۷) نشان می‌دهد که سخت‌ترین مسائل مربوط به دسته با موعد تحویل d2 و d3 می‌باشد که مؤید نتیجه فوق است.

بررسی تأثیر موعد تحویل روی عملکرد الگوریتم شاخه و کران در مرحله چهارم انجام می‌شود. نرخ زوال کارها از بازه $[0.2, 0.3]$ تولید و برای متغیر λ مقادیر ۱، ۰.۲ و ۳ و تعداد کارها برابر با ۱۵ فرض می‌شود. موعد تحویل کارها نیز مشابه مرحله اول از چهار بازه $(0, 0.5C_{max}]$ ، $(0.5C_{max}, C_{max}]$ ، $(0, C_{max})$ و $(0, 1.5C_{max})$ که به صورت d1 تا d4 کدگذاری می‌شود انتخاب می‌گردد. مشابه دو آزمایش قبلی برای هر حالت ۱۰۰۰ مسأله به صورت تصادفی تولید و حل شده است که نتایج آن در شکل (۶) و (۷) نشان داده شده است.

در هر دو مرحله قبل بیان گردید که مسائل مربوط به دسته‌های با λ بزرگ‌تر دارای متوسط زمان حل و تعداد گره‌های بررسی شده بیشتری است که در این مرحله نیز این موضوع تایید شده است. همان‌طور که در شکل (۶) و (۷) مشاهده می‌شود متوسط تعداد گره‌های بررسی شده و زمان حل برای مسائل با موعد تحویل d1 و d4 نسبت به



شکل (۶): متوسط تعداد گره‌های بررسی شده در الگوریتم شاخه و کران با توجه به بازه موعد تحویل



شکل (۷): متوسط زمان حل الگوریتم شاخه‌وکران با توجه به بازه موعد تحویل

۶. نتیجه‌گیری

در این مقاله یک مسأله زمان‌بندی فعالیت‌های روبه زوال مورد بررسی قرار گرفته شد که در آن زمان پردازش واقعی فعالیت‌ها براساس یک تابع خطی محاسبه می‌شود. نرخ زوال کارها و زمان ورود آن‌ها غیریکسان فرض شده و هدف حداقل کردن تعداد کارهای دارای دیرکرد می‌باشد. نشان داده شد که درجه پیچیدگی مسأله $NP-hard$ است بنابراین یک مدل ریاضی برای مسأله ارائه و جهت حل مسأله به صورت بهینه یک رویکرد شاخه‌وکران پیشنهاد گردید. به منظور ارزیابی الگوریتم پیشنهاد شده نتایج محاسباتی شامل چهار مرحله طراحی و اجرا شده است. نتایج نشان داد که الگوریتم پیشنهادی قادر به حل مسائل با ابعاد ۳۰ فعالیت در زمان مناسب برای اکثر دسته مسائل مختلف می‌باشد که در مقایسه با تحقیقات مشابه کارایی بالاتری دارد. درصد کل گره‌های قطع شده توسط الگوریتم پیشنهادی حداقل برابر با ۸۵٫۶۱ درصد می‌باشد که نشان‌دهنده کارایی بالای آن است. تحلیل حساسیت روی مقادیر λ ، نرخ زوال و موعد تحویل نشان داد که با افزایش مقدار λ و کاهش نرخ زوال مسائل سخت‌تر می‌شود. همچنین مسائلی که موعد تحویل کارها از بازه $[5C_{max}, C_{max}]$ تولید می‌شود نسبت به دیگر مسائل سخت‌تر بوده و متوسط زمان حل آن‌ها بیشتر است. استفاده از الگوریتم‌های ابتکاری یا فراابتکاری و مقایسه آن با نتایج این تحقیق از جمله پیشنهاداتی است که می‌تواند در آینده مورد بررسی قرار گیرد. همچنین این مسأله می‌تواند با اضافه شدن فرض محدودیت دسترسی به ماشین نیز در نظر گرفته شود. بررسی مسأله در محیط‌های چندماشین نیز می‌تواند در آینده توسط محققین انجام گیرد.

مراجع

- انعطاف‌پذیر". نشریه پژوهش‌های مهندسی صنایع در سیستم‌های تولید، (۱): ۱۳-۲۴.
- [3] Delorme, M., Lori, M., Mendez, N.F.M. (2021). "Solution methods for scheduling problems with sequence-dependent deterioration and maintenance events". *European Journal of Operational Research*, 295(3): 823-837.
- [۴] بهنامیان، جواد، دیانت، فاطمه، (۱۳۹۵). "مقایسه سه روش فراابتکاری برای کمینه نمودن زمان چرخه در مسأله زمان‌بندی جریان کارگاهی مختلط دوره‌ای با در نظر گرفتن اثر یادگیری". نشریه پژوهش‌های مهندسی صنایع در سیستم‌های تولید، ۴(۸): ۱۱۷-۱۰۵.
- [5] Expósito-Izquierdo, C., Angel-Bello, F., Melián-Batista, B., Alvarez, A., Báez, S. (2019). "A metaheuristic algorithm and simulation to study the effect of learning or tiredness on sequence-dependent setup times in a parallel machine scheduling problem". *Expert Systems with Applications*, 117: 62-74.
- [6] Ouazene, Y., Yalaoui, F. (2018). "Identical parallel machine scheduling with time-dependent processing times". *Theoretical Computer Science*, 721: 70-77.
- [7] Cheng, T.C.E., Ding, Q., Lin, B.M.T. (2004). "A concise survey of scheduling with time-dependent processing times". *European Journal of Operational Research*, 152(1): 1-13.
- [8] Wang, J.B., Wang, J.J. (2015). "Single-machine scheduling problems with precedence constraints and simple linear deterioration". *Applied Mathematical Modelling*, 39(3): 1172-1182.
- [9] Wu, C.C., Lee, W.C. (2006). "Two-machine flowshop scheduling to minimize mean flow time under linear deterioration". *International Journal of Production Economics*, 103(2): 572-584.
- [10] Pinedo, M. (2002). *Scheduling: Theory, Algorithms, and Systems*. Prentice-Hall, New Jersey.
- [11] Cheng, T.C.E., Kravchenko, S.A., Lin, B.M.T. (2020). "Scheduling step-deteriorating jobs to minimize the total completion time". *Computers & Industrial Engineering*, 144: 106329.
- [12] Browne, S., Yechiali, U. (1990). "Scheduling deteriorating jobs on a single processor". *Computers & Operations Research*, 38(3): 495-499.
- [13] Lee, W.C., Wu, C.C., Chung, Y.H. (2008). "Scheduling deteriorating jobs on a single machine with release times". *Computers & Industrial Engineering*, 54(3): 441-452.

- [۱] جعفری ندوشن، عباسعلی. (۱۳۹۶). "زمان‌بندی فعالیت‌های روبه زوال خطی در محیط تک‌ماشین با فرض ورود غیرهم‌زمان کارها". چهاردهمین کنفرانس بین‌المللی مهندسی صنایع، دانشگاه علم و صنعت ایران.
- [۲] فخرزاد، محمدباقر، علی‌نژاد، اسماعیل. (۱۳۹۲). "برنامه‌ریزی و زمان‌بندی پیشرفته با در نظر گرفتن اثر یادگیری در سیستم‌های ساخت کارگاهی

- three machine flow shop with deteriorating jobs". *Computers & operations research*, 40(2): 547-557.
- [24] Jafari, A.A., Khademi-zare, H., Lotfi, M.M., Tavakkoli-Moghaddam, R. (2016). "A note on "minimizing makespan in three machine flowshop with deteriorating jobs"". *Computers & operations research*, 72: 93-96.
- [25] Wang, L., Sun, L.Y., Sun, L.H., Wang, J.B. (2010). "On three-machine flow shop scheduling with deteriorating jobs". *International Journal of Production Economics*, 125(1): 185-189.
- [26] Jafari, A.A., Khademi-zare, H., Lotfi, M.M., Tavakkoli-Moghaddam, R. (2017). "A note on "On three-machine flow shop scheduling with deteriorating jobs"". *International Journal of Production Economics*, 191: 250-252.
- [27] Li, D., Lu, X. (2020). "Parallel-batch scheduling with deterioration and rejection on a single machine". *Applied Mathematics-A Journal of Chinese Universities*, 35: 141-156.
- [28] Mor, B., Mosheiov, G. (2020). "Minimizing total load on parallel machines with linear deterioration". *Optimization Letters*, 14: 771-779.
- [29] Bai, D., Bai, X., Yang, J., Zhang, X., Ren, T., Xie, C., Liu, B. (2021). "Minimization of maximum lateness in a flowshop learning effect scheduling with release dates". *Computers & Industrial Engineering*, 158: 107309.
- [30] Brucker, P. (2006). *Scheduling algorithm*, 5th Edition, Springer, Berlin, Heidelberg, New York.
- [31] Chen, R., Yuan, J. (2020). "Single-machine scheduling of proportional-linearly deteriorating jobs with positional due indices". *Journal of Operation Research*, 18(2): 177-196.
- [32] Moslehi, G., Jafari, A. (2010). "Minimizing the number of tardy jobs under piecewise-linear deterioration". *Computers & Industrial Engineering*, 59(4): 573-584.
- [14] Jafari, A., Moslehi, G., (2011). "Scheduling linear deteriorating jobs to minimize the number of tardy jobs". *Journal of Global Optimization*, 54(2): 389-404.
- [15] Hsu, Y.S., Lin, B.M.T. (2003). "Minimization of maximum lateness under linear deterioration". *Omega*, 31(6): 459-469.
- [16] Cheng, T.C.E., Hsu, C.J., Huang, Y.C., Lee, W.C. (2011). "Single-machine scheduling with deteriorating jobs and setup times to minimize the maximum tardiness". *Computers & Operations Research*, 38(12): 1760-1765.
- [17] Lee, W.C., Lin, J.B., Shiau, Y.R. (2011). "Deteriorating job scheduling to minimize the number of late jobs with setup times". *Computers & Industrial Engineering*, 61(3): 782-787.
- [18] Lee, W.C., Lu, Z.S. (2012). "Group scheduling with deteriorating jobs to minimize the total weighted number of late jobs". *Applied Mathematics and Computation*, 218(17): 8750-8757.
- [19] Wu, C.C., Cheng, S.R., Wu, W.H., Yin, Y., Wu, W.H. (2013). "The single-machine total tardiness problem with unequal release times and a linear deterioration". *Applied Mathematics and Computation*, 219(20): 10401-10415.
- [20] Lee, W.X., Zhao, C.L. (2015). "Deteriorating jobs scheduling on a single machine with release dates, rejection and a fixed non-availability interval". *Journal of Applied Mathematics and Computing*, 48(1-2): 585-605
- [21] Jafari, A.A., Khademi-zare, H., Lotfi, M.M., Tavakkoli-Moghaddam, R. (2016). "Minimizing Makespan with Start Time-Dependent Jobs in a Two-Machine Flow Shop", *International Journal of Engineering, IJE TRANSACTIONS B: Applications*, 29(6): 778-787.
- [22] Lee, W.C., Wu, C.C., Wen, C.C., Chung, Y.H. (2008). "A two-machine flowshop makespan scheduling problem with deteriorating jobs". *Computers & Industrial Engineering*, 54(4): 737-749.
- [23] Wang, J.B., Wang, M.Z. (2013). "Minimizing makespan in

پیوست

اثبات لم ۲:

$$C_j(\delta_1) = P_j + S_j = (a_j + b_j S_j) + S_j$$

$$= a_j + (1 + b_j)S_j \quad (۴۹)$$

$$= a_j + (1 + b_j)(a_i + (1 + b_i)r_i)$$

$$S_j(\delta_2) = S_j = \text{Max}\{T, r_j\} = T \quad (۵۰)$$

$$C_j(\delta_2) = P_j + S_j = (a_j + b_j S_j) + S_j$$

$$= a_j + (1 + b_j)S_j = a_j + (1 + b_j)T \quad (۵۱)$$

$$S_i(\delta_2) = S_i = \text{Max}\{C_j(\delta_2), r_i\} = \text{Max}\{a_j + (1 + b_j)T, r_i\} = a_j + (1 + b_j)T \quad (۵۲)$$

$$C_i(\delta_2) = P_i + S_i = (a_i + b_i S_i) + S_i$$

$$= a_i + (1 + b_i)S_i =$$

$$a_i + (1 + b_i)(a_j + (1 + b_j)T) \quad (۵۳)$$

$$r_i(1 + b_i)(1 + b_j) + a_i b_j <$$

$$T(1 + b_i)(1 + b_j) + a_j b_i \quad (۵۴)$$

وجود رابطه (۵۴) و روابط (۴۹) و (۵۳) باعث برقراری رابطه زیر می‌گردد.

$$C_j(\delta_1) \leq C_i(\delta_2) \quad (۵۵)$$

باتوجه به فرض زودکردار شدن هر دو کار J_i و J_j در توالی δ_1 رابطه زیر برقرار است.

$$N_T(\delta_1) = N_T(\delta_0) + N_T(\delta_{Tardy}) \quad (۵۶)$$

ازطرفی ممکن است J_i و J_j در توالی δ_2 دیرکردار گردد بنابراین رابطه زیر برای این توالی برقرار می‌گردد.

$$N_T(\delta_0) + N_T(\delta_{Tardy}) \leq N_T(\delta_2)$$

$$\leq N_T(\delta_0) + N_T(\delta_{Tardy}) + 2 \quad (۵۷)$$

با مقایسه روابط (۵۶) و (۵۷) رابطه زیر برقرار خواهد شد.

$$N_T(\delta_1) \leq N_T(\delta_2) \quad (۵۸)$$

باتوجه به برقراری هم‌زمان دو رابطه $C_j(\delta_1) \leq C_i(\delta_2)$ و $N_T(\delta_1) \leq N_T(\delta_2)$ ، ثابت می‌گردد که توالی δ_1 بر δ_2 غالب می‌باشد.

اثبات لم ۴:

$$S_i(\delta_1) = S_i = \text{Max}\{T, r_i\} = T \quad (۵۹)$$

$$C_i(\delta_1) = P_i + S_i = (a_i + b_i S_i) + S_i = a_i + (1 + b_i)S_i = a_i + (1 + b_i)T \quad (۶۰)$$

$$S_j(\delta_1) = S_j = \text{Max}\{C_i(\delta_1), r_j\} = \text{Max}\{a_i + (1 + b_i)T, r_j\} = r_j \quad (۶۱)$$

$$C_j(\delta_1) = P_j + S_j = (a_j + b_j S_j) + S_j$$

$$= a_j + (1 + b_j)S_j = a_j + (1 + b_j)r_j \quad (۶۲)$$

$$S_j(\delta_2) = S_j = \text{Max}\{T, r_j\} = r_j \quad (۶۳)$$

$$C_j(\delta_2) = P_j + S_j = (a_j + b_j S_j) + S_j$$

$$= a_j + (1 + b_j)S_j = a_j + (1 + b_j)r_j \quad (۶۴)$$

$$S_i(\delta_1) = S_i = \text{Max}\{T, r_i\} = T \quad (۳۳)$$

$$C_i(\delta_1) = P_i + S_i = (a_i + b_i S_i) + S_i = a_i + (1 + b_i)S_i = a_i + (1 + b_i)T \quad (۳۴)$$

$$S_j(\delta_1) = S_j = \text{Max}\{C_i(\delta_1), r_j\} = \text{Max}\{a_i + (1 + b_i)T, r_j\} = a_i + (1 + b_i)T \quad (۳۵)$$

$$C_j(\delta_1) = P_j + S_j = (a_j + b_j S_j) + S_j$$

$$= a_j + (1 + b_j)S_j = a_j + (1 + b_j)(a_i + (1 + b_i)T) \quad (۳۶)$$

$$S_j(\delta_2) = S_j = \text{Max}\{T, r_j\} = T \quad (۳۷)$$

$$C_j(\delta_2) = P_j + S_j = (a_j + b_j S_j) + S_j$$

$$= a_j + (1 + b_j)S_j = a_j + (1 + b_j)T \quad (۳۸)$$

$$S_i(\delta_2) = S_i = \text{Max}\{C_j(\delta_2), r_i\} = \text{Max}\{a_j + (1 + b_j)T, r_i\} = a_j + (1 + b_j)T \quad (۳۹)$$

$$C_i(\delta_2) = P_i + S_i = (a_i + b_i S_i) + S_i$$

$$= a_i + (1 + b_i)S_i = a_i + (1 + b_i)(a_j + (1 + b_j)T) \quad (۴۰)$$

$$T(1 + b_i + b_j + b_i b_j) + a_i b_j <$$

$$T(1 + b_i + b_j + b_i b_j) + a_j b_i \quad (۴۱)$$

وجود رابطه (۴۱) و درنظر گرفتن روابط (۳۶) و (۴۰) باعث برقراری رابطه زیر می‌گردد.

$$C_j(\delta_1) \leq C_i(\delta_2) \quad (۴۲)$$

باتوجه به فرض زودکردار شدن هر دو کار J_i و J_j در توالی δ_1 رابطه زیر برقرار است.

$$N_T(\delta_1) = N_T(\delta_0) + N_T(\delta_{Tardy}) \quad (۴۳)$$

ازطرفی ممکن است J_i و J_j در توالی δ_2 دیرکردار گردد بنابراین رابطه زیر برای این توالی برقرار می‌گردد.

$$N_T(\delta_0) + N_T(\delta_{Tardy}) \leq N_T(\delta_2)$$

$$\leq N_T(\delta_0) + N_T(\delta_{Tardy}) + 2 \quad (۴۴)$$

بامقایسه روابط (۴۴) و (۴۵) رابطه زیر برقرار خواهد شد.

$$N_T(\delta_1) \leq N_T(\delta_2) \quad (۴۵)$$

باتوجه به برقراری دو رابطه $C_j(\delta_1) \leq C_i(\delta_2)$ و $N_T(\delta_1) \leq N_T(\delta_2)$ به‌طور هم‌زمان، ثابت می‌گردد که توالی δ_1 بر δ_2 غالب می‌باشد.

اثبات لم ۳:

$$S_i(\delta_1) = S_i = \text{Max}\{T, r_i\} = r_i \quad (۴۶)$$

$$C_i(\delta_1) = P_i + S_i = (a_i + b_i S_i) + S_i = a_i + (1 + b_i)S_i = a_i + (1 + b_i)r_i \quad (۴۷)$$

$$S_j(\delta_1) = S_j = \text{Max}\{C_i(\delta_1), r_j\} = \text{Max}\{a_i + (1 + b_i)r_i, r_j\} = a_i + (1 + b_i)r_i \quad (۴۸)$$

$$= a_j + (1 + b_j)S_j$$

$$= a_j + (1 + b_j)r_j$$

$$S_j(\delta_2) = S_j = \text{Max}\{T, r_j\} = r_j \quad (76)$$

$$C_j(\delta_2) = P_j + S_j = (a_j + b_j S_j) + S_j \quad (77)$$

$$= a_j + (1 + b_j)S_j = a_j + (1 + b_j)r_j$$

$$S_i(\delta_2) = S_i = \text{Max}\{C_j(\delta_2), r_i\} = \text{Max}\{a_j + (1 + b_j)r_j, r_i\} = a_j + (1 + b_j)r_j \quad (78)$$

$$C_i(\delta_2) = P_i + S_i = (a_i + b_i S_i) + S_i = a_i + (1 + b_i)S_i = a_i + (1 + b_i)(a_j + (1 + b_j)r_j) \quad (79)$$

$$a_j + (1 + b_j)r_j < a_i + (1 + b_i)(a_j + (1 + b_j)r_j) \quad (80)$$

وجود رابطه (۸۰) و در نظر گرفتن روابط (۷۵) و (۷۹) باعث برقراری رابطه زیر می‌گردد.

$$C_j(\delta_1) \leq C_i(\delta_2) \quad (81)$$

باتوجه به فرض زودکردار شدن هر دو کار J_i و J_j در توالی δ_1 رابطه زیر برقرار است.

$$N_T(\delta_1) = N_T(\delta_0) + N_T(\delta_{Tardy}) \quad (82)$$

ازطرفی ممکن است J_i و J_j در توالی δ_2 دیرکردار گردد بنابراین رابطه زیر برای این توالی برقرار می‌گردد.

$$N_T(\delta_0) + N_T(\delta_{Tardy}) \leq N_T(\delta_2) \leq N_T(\delta_0) + N_T(\delta_{Tardy}) + 2 \quad (83)$$

بامقایسه روابط (۸۲) و (۸۳) رابطه زیر برقرار خواهد شد.

$$N_T(\delta_1) \leq N_T(\delta_2) \quad (84)$$

باتوجه به برقراری دو رابطه $C_j(\delta_1) \leq C_i(\delta_2)$ و $N_T(\delta_1) \leq N_T(\delta_2)$ به‌طور هم‌زمان، ثابت می‌گردد که توالی δ_1 بر δ_2 غالب می‌باشد.

$$S_i(\delta_2) = S_i = \text{Max}\{C_j(\delta_2), r_i\} = \text{Max}\{a_j + (1 + b_j)r_j, r_i\} = a_j + (1 + b_j)r_j \quad (65)$$

$$C_i(\delta_2) = P_i + S_i = (a_i + b_i S_i) + S_i = a_i + (1 + b_i)S_i = a_i + (1 + b_i)(a_j + (1 + b_j)r_j) \quad (66)$$

$$a_j + (1 + b_j)r_j < a_i + (1 + b_i)(a_j + (1 + b_j)r_j) \quad (67)$$

وجود رابطه (۶۷) و در نظر گرفتن روابط (۶۲) و (۶۶) باعث برقراری رابطه زیر می‌گردد.

$$C_j(\delta_1) \leq C_i(\delta_2) \quad (68)$$

باتوجه به فرض زودکردار شدن هر دو کار J_i و J_j در توالی δ_1 رابطه زیر برقرار است.

$$N_T(\delta_1) = N_T(\delta_0) + N_T(\delta_{Tardy}) \quad (69)$$

ازطرفی ممکن است J_i و J_j در توالی δ_2 دیرکردار گردد بنابراین رابطه زیر برای این توالی برقرار می‌گردد.

$$N_T(\delta_0) + N_T(\delta_{Tardy}) \leq N_T(\delta_2) \leq N_T(\delta_0) + N_T(\delta_{Tardy}) + 2 \quad (70)$$

بامقایسه روابط (۶۹) و (۷۰) رابطه زیر برقرار خواهد شد.

$$N_T(\delta_1) \leq N_T(\delta_2) \quad (71)$$

باتوجه به برقراری دو رابطه $C_j(\delta_1) \leq C_i(\delta_2)$ و $N_T(\delta_1) \leq N_T(\delta_2)$ به‌طور هم‌زمان، ثابت می‌گردد که توالی δ_1 بر δ_2 غالب می‌باشد.

اثبات لم ۵:

$$S_i(\delta_1) = S_i = \text{Max}\{T, r_i\} = r_i \quad (72)$$

$$C_i(\delta_1) = P_i + S_i = (a_i + b_i S_i) + S_i = a_i + (1 + b_i)S_i = a_i + (1 + b_i)r_i \quad (73)$$

$$S_j(\delta_1) = S_j = \text{Max}\{C_i(\delta_1), r_j\} = \text{Max}\{a_i + (1 + b_i)r_i, r_j\} = r_j \quad (74)$$

$$C_j(\delta_1) = P_j + S_j = (a_j + b_j S_j) + S_j \quad (75)$$



DOI: 10.22084/IER.2021.24228.2024

A Mathematical Model and a Branch and Bound Algorithm for the Single Machine Scheduling Problem under Linear Deterioration and Release Times

A. Jafari-Nodoushan^{1*}, M. H. Dehghani Sardabadi², A. Bozorgi-Amiri³

1. Assistant Professor, Department of Industrial Engineering, Faculty of Engineering, Meybod University, Meybod, Iran
2. PhD student, Department of Industrial Engineering, Faculty of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran
3. Associate Professor, Department of Industrial Engineering, Faculty of Industrial Engineering, Campus of Technical Colleges, University of Tehran, Tehran, Iran

ARTICLE INFO

Article history:

Received 7 May 2021

Accepted 26 November 2021

Keywords:

Deteriorating Jobs
Scheduling
Number of Tardy Jobs
Release Time
Branch and Bound Algorithm

ABSTRACT

In this paper, the single machine scheduling problem with linear deteriorating jobs under release times is considered where the objective is to minimize the number of tardy jobs. The problem is proved *NP-hard* according to the literature review. At first, a mathematical model is presented to the problem and a Branch and Bound algorithm with considering dominance rules and lower bounds is supposed to solve the problem optimally. Computational results are presented in four parts to evaluate the performance of the proposed algorithm and the effect of related parameters on the algorithm. According to the variance analysis test, it was found that the efficiency of the branch and bound algorithm is high so that it is able to solve the most problems with job size 30 within a reasonable time and the average percentage of entire fathomed nodes in all the problems is at least 85.61 percentage. It was also shown the problems with larger λ and smaller deterioration rates are difficult and the average solution time of the algorithm is high for them. On the other hand, if the due date of the jobs was big or small, the problem will be simple and the solution time is less than the problems with medium due dates.

* Corresponding author. A. Jafari-Nodoushan
Tel.: 035-32322416; E-mail address: a.jafari@meybod.ac.ir