ارائه حدود پایین جدید روی مقدار بهینه زمان انجام کل کارها در یک سیستم تک ماشینه‌ی پردازش‌گر انباشته

علی حسین زاده کاشان[۱]; بهروز کریمی[۲*]

چکیده

در این مقاله زمان‌بندی یک ماشین پردازش‌گر انباشته با هدف حداقل‌سازی زمان انجام کل کارها (C_{max}) بررسی شده است. منظور از یک ماشین پردازشگر انباشته، ماشینی است که قابلیت انجام عملیات هم‌زمان روی گروهی از کارها را در قالب یک دسته یا انباشته دارد. البته با اعمال این محدودیت که مجموع اندازه کارهایی که در یک انباشته باهم می‌آیند از ظرفیت ماشین (B) بیشتر نباشد. برای هر یک از کارها دو عامل اندازه و زمان پردازش مفروض است. زمان انجام عملیات ماشین بر روی یک انباشته برابر با زمان عملیات مورد نیاز کاری است که در میان کارهای متعلق به آن انباشته بزرگ‌ترین زمان پردازش را دارد. برای این مساله، دو روش جدید تولید حد پایین روی مقدار بهینه تابع هدف با نام‌های LB_2 و LB_3 ارائه شده و ثابت می‌شود که نسبت به تنها حد پایین موجود در ادبیات موضوع مساله (LB_1) عملکرد بهتری دارند. هم‌چنین ثابت می‌شود که عملکرد LB_3 حداقل به خوبی عملکرد LB_2 است.

کلمات کلیدی : زمان‌بندی، ماشین پردازش‌گر انباشته، حد پایین، زمان انجام همه کارها

New Lower Bounds for the Optimal Makespan on a Single Batch Processing Machine

A. Husseinzadeh Kashan, B. Karimi

ABSTRACT

This paper considers minimizing makespan (C_{max}) on a single batch-processing machine. A batch-processing machine can process a group of jobs simultaneously, as long as the total size of jobs in the batch does not exceed the machine capacity (B). For each job, we assume a specific job size and job processing time. The processing time of a batch is just the longest processing time of all jobs in the batch. We introduce two new procedures for obtaining lower bounds of the optimal makespan, entitled LB_2 and LB_3, respectively. We prove that both of the new bounds are tighter than the only existing bound called LB_1. We also prove that LB_3 is at least as tight as LB_2.

KEYWORDS : Scheduling, batch-processing machine, lower bounds, makespan

---

تاریخ دریافت مقاله:۱۳۸۷/۹/۱۸

تاریخ اصلاحات مقاله: ۱۳۸۹/۹/۱۷

[۱] دکتری صنایع؛ دانشگاه صنعتی امیرکبیر: A.kashani@aut.ac.ir

[۲ *] نویسنده مسئول و دانشیار دانشکده صنایع؛ دانشگاه صنعتی امیرکبیر: B.karimi@aut.ac.ir

۱- مقدمه

در دو دهه اخیر مسائل مربوط به تعیین توالی و زمان‌بندی عملیات در سیستم‌های تولید انباشته‌ای مورد توجه قرار گرفته‌اند. به‌طوری‌که امروزه زمان‌بندی انباشته‌ها در سیستم تولید، یکی از سیاست‌های معمول در بسیاری از صنایع است. به طور کلی مدل‌های زمان‌بندی در تولید انباشته‌ای، بر حسب نوع انباشته‌سازی به دو دسته طبقه‌بندی می‌شوند. در نوع اول، سفارشات (که به آنها "کار" گفته می‌شود) در صورتی در قالب یک انباشته تولید می‌شوند که از نظر راه‌اندازی ماشین دارای نیازمندی‌های مشابه باشند. همچنین در هر لحظه، عملیات ماشین فقط روی یک سفارش انجام می‌شود. ولی طی دو دهه اخیر شکل دیگری از مدل‌های زمان‌بندی تولید انباشته‌ای مورد توجه محققین فعال در زمینه نظریه تعیین توالی عملیات و زمان‌بندی قرار گرفته است که شامل نوع دوم انباشته‌سازی است. در این نوع تولید انباشته‌ای، چندین کار به طور هم‌زمان تحت عملیات ماشین قرار می‌گیرند. اما ظرفیت ماشین محدود بوده و گردآوری کارها (که هر یک ممکن است دارای نیازمندی‌های ظرفیتی متفاوت باشند) در قالب انباشته‌ها باید به گونه‌ی مؤثر صورت پذیرد. به طور خاص به مدل‌های زمان‌بندی که با این نوع محیط تولیدی سروکار دارند، مدل‌های زمان‌بندی ماشین‌های پردازش‌گر انباشته[۱] گفته می‌شود.

در ادبیات زمان‌بندی، مثال‌های فراوانی از ایستگاه‌های تولیدی که در آن از ماشین‌های پردازش‌گر انباشته استفاده می‌شود آورده شده‌اند. برای مثال، عملیات فر درون سوز[۲] در تولید بوردهای مدار مجتمع، در داخل فرهایی انجام می‌شود که چندین گروه از بوردها را می‌توانند در خود جای داده و تحت عملیات حرارتی قرار دهند [۱۸]. در اینجا فرها نقش ماشین‌های پردازش‌گر انباشته را دارند.

در سال‌های اخیر مدل‌های زمان‌بندی ماشین‌های پردازش‌گر انباشته با فرض وجود نیازمندی‌های ظرفیتی (اندازه) متفاوت برای کارها مورد توجه قرار گرفته‌اند. در عمل ممکن است تعابیر متفاوتی از اندازه یک کار وجود داشته باشد. از طرفی ممکن است اندازه یک کار بعنوان عاملی درنظر گرفته شود که معرف ابعاد فیزیکی آن است. از سوی دیگر ممکن است اندازه یک کار معرف نیازمندی عملیاتی آن باشد. به عنوان مثال می‌توان مقدار انرژی مصرفی غیرمستقیم به‌وسیله یک کار در انباشته را در صورتی که مقدار انرژی صرف شده برای انباشته محدود باشد، به عنوان یک ویژگی از کار و در قالب اندازه کار درنظر گرفت. در مسائل زمان‌بندی با فرض وجود نیازمندی‌های ظرفیتی متفاوت برای کارها، فرض می‌شود که شکستن یک کار و تخصیص آن به بیش از یک انباشته مجاز نیست. چنین فرضی با توجه به درنظرگرفتن یکپارچگی فیزیکی کارها مطرح می‌شود. حتی اگر کارها از نظر فیزیکی قابل شکست باشند، ممکن است انقطاع مجاز نباشد. بعنوان مثال با درنظرگرفتن چنین سیاستی، ردیابی سفارشات مشتریان و ارائه خدمات پس از فروش می‌تواند با سهولت بیشتری انجام گیرد [۸]. نکته قابل توجه این است که خانواده مسائل زمان‌بندی ماشین‌های پردازش‌گر انباشته با کارهای با اندازه نامساوی در طبقه مسائل NP-hard قرار می‌گیرد [۲۳]. زیرا این مسائل به طور غیرمستقیم شامل مساله تخصیص کارها به انباشته‌ها هستند که جزء سخت‌ترین مسائل موجود در ادبیات مسائل ترکیباتی است.

به طور خاص، مساله در نظر گرفته شده در این مقاله شامل زمان‌بندی یک ماشین پردازش‌گر انباشته با فرضیات زیر است:

– تعداد n کار برای زمان‌بندی بر روی یک ماشین پردازش‌گر انباشته وجود دارد که همگی در لحظه صفر در دسترس هستند. فرض کنید مجموعه J شامل اندیس همه کارهای در دسترس باشد.

– با شروع عملیات ماشین روی یک انباشته، توقف عملیات امکان‌پذیر نبوده و پیش از اتمام عملیات ماشین هیچ کاری نمی‌تواند به انباشته اضافه و یا از آن خارج شود.

– برای هر کار i، مقدار کمیتی معادل s_i واحد از منابع محدود (ظرفیت ماشین) مورد نیاز است.

– زمان شروع و پایان عملیات تمام کارهایی که متعلق به یک انباشته هستند با یکدیگر یکسان است. زمان پردازش انباشته نیز برابر با زمان پردازش کاری است که در بین سایر کارهای متعلق به انباشته طولانی‌ترین زمان پردازش را دارد.

– حداکثر ظرفیت ماشین برای انجام عملیات همزمان روی کارها برابر B است. به عبارت دیگر مجموع اندازه کارهای متعلق به یک انباشته نباید از مقدار B بیشتر شود.

ماشین‌های پردازش‌گر انباشته در مقایسه با سایر عملیات، زمان انجام عملیات طولانی‌تری دارند و بنابراین ایستگاه گلوگاه تولید هستند. بنابراین میزان بهره‌برداری بالا از این نوع ماشین‌ها می‌تواند منجر به افزایش توان عملیاتی کل سیستم شود. از آنجا که میزان بهره‌برداری بالاتر می‌تواند به معنی کاهش زمان انجام عملیات کلیه کارها (یا به طور معادل مقدار C_{max}) به‌وسیله ماشین(ها) باشد، بنابراین معیار عملکرد حداقل‌سازی C_{max} است.

در ادامه، به مساله با فرضیات فوق با عنوان *BPM-CMAX*

اشاره خواهد شد. مساله *BPM-CMAX* نخستین بار در سال ۱۹۹٤ بررسی شد [۲۳]. پس از آن نیز این مساله توجه عده زیادی از محققین را به خود جلب نمود. اما تلاش‌های صورت گرفته، فقط محدود به ارائه روش‌های حل بوده و در رابطه با ارزیابی عملکرد واقعی این روش‌ها تلاش چندانی انجام نشده است. از این‌رو در این مقاله به جبران ضعف موجود در زمینه ارزیابی عملکرد الگوریتم‌های ارائه شده برای مساله *BPM-CMAX* خواهیم پرداخت. این کار به طور خاص از طریق ارائه روش‌های کارا برای تولید حدود پایین قوی روی مقدار بهینه C_{max} انجام خواهد شد. منظور از یک حد پایین روی مقدار بهینه تابع هدف (z) در یک مساله حداقل‌سازی، مقداری است مانند l که همواره شرط $l \le z$ را برقرار می‌سازد. از مهم‌ترین کاربردهای حدود پایین عبارتند از:

– استفاده از حدود پایین در بدنه الگوریتم‌های دقیق مانند روش شاخه و حد.

– استفاده از حدود پایین در بدنه مدل‌های ریاضی به عنوان یک محدودیت.

– استفاده از حدود پایین در ارزیابی میزان نزدیکی جواب‌های بدست آمده از الگوریتم‌های مختلف به جواب بهینه.

– استفاده از حدود پایین توسعه داده شده برای یک مساله در ایجاد حدود پایین برای مسائل پیچیده و کلی‌تر.

مقدار بهینه تابع هدف با نماد C^* نشان داده می‌شود. همچنین از نماد C^A برای اشاره به مقدار حد پایین حاصل از روش A استفاده می‌شود. عبارات C_Q^A یا C_Q^* به ترتیب نشان دهنده مقدار بهینه تابع هدف و یا مقدار حد پایین حاصل از روش A، به ازای مجموعه کارهایی است که اندیس آنها متعلق به Q می‌باشد.

پیش از ارائه حدود پایین برای مساله *BPM-CMAX* فرض می‌شود که به ازای هر نمونه مساله ورودی، نخست هر کار y متعلق به J' با شرط $J' = \{y \mid B - s_y < \min_{i \in J}\{s_i\}, y \in J\}$ را در یک انباشته مجزا قرار داده و آن را از مجموعه کارها (J) حذف می‌نماییم. به این دلیل که کارهای متعلق به J' نمی‌توانند با هیچ یک از کارهای موجود در یک انباشته گروه‌بندی شوند. بنابراین داریم $C^* = \sum_{j \in J'} p_j + C^*_{J \setminus J'}$. بدون اینکه از کلیت کار کاسته شود در ادامه فرض می‌شود هیچ کاری وجود ندارد که در مجموعه J' قرار گیرد.

در بخش دوم مقاله به مرور مقالات مرتبط با موضوع پرداخته و در بخش سوم به معرفی تنها حد پایین ارائه شده برای مساله *BPM-CMAX* خواهیم پرداخت. در این بخش همچنین دو حد پایین جدید برای مقدار بهینه تابع هدف ارائه می‌شود. در بخش چهارم نیز کیفیت حدود پایین سنجیده و مقایسه می‌شوند. بخش پنجم به جمع‌بندی و نتیجه‌گیری می‌پردازد.

۲- مرور ادبیات

در سال‌های اخیر مسائل زمان‌بندی ماشین‌های پردازش‌گر انباشته به طور گسترده مورد توجه محققین فعال در زمینه زمان‌بندی و توالی عملیات قرار گرفته‌اند. در مرجع [۲۰] مروری جامع بر ادبیات مسائل زمان‌بندی ماشین‌های پردازش‌گر انباشته انجام گرفته است. در این بخش به مرور مقالاتی می‌پردازیم که فرضیات درنظر گرفته شده در آنها نزدیک به فرضیات مساله *BPM-CMAX* است؛ یعنی مقالاتی که در آنها اندازه کارها یکسان در نظر گرفته شده است.

نخستین تلاش صورت گرفته در زمینه زمان‌بندی ماشین‌های پردازش‌گر انباشته با فرض وجود کارهای با اندازه نامساوی را می‌توان به دابسون و نامبیمادوم [۸] نسبت داد. اما این مدل‌ها به طور جدی پس از اینکه اوزوی [۲۳] محل کاربرد آنها را در صنایع تولید نیمه‌هادی‌ها یافت، مورد توجه محققین قرار گرفتند. در مرجع [۲۳]، حداقل‌سازی C_{max} و ΣC_i روی یک ماشین پردازش‌گر انباشته مورد توجه بوده است. در مرجع [۱۰] نیز دو الگوریتم ابتکاری برای مساله *BPM-CMAX* ارائه شده که یکی بر مبنای اصلاح الگوریتم معرفی شده در مرجع [۲۳] و دیگری بر مبنای تعیین اقلام یک انباشته با استفاده از حل یک مساله کوله‌پشتی عمل می‌نماید. ژانگ و همکاران [۲۵] به تحلیل عملکرد الگوریتم‌های ارائه شده در مرجع [۲۳] برای مساله حداقل‌سازی C_{max} در بدترین حالت پرداخته و یک الگوریتم تقریبی جدید ارائه کرده‌اند. حسین زاده کاشان و همکاران [۱۳] فرضیات مساله معرفی شده در مرجع [۲۵] را تعمیم داده و یک الگوریتم ابتکاری با نسبت عملکرد معلوم ارائه داده‌اند. آنها همچنین به ارائه یک الگوریتم تقریبی برای حالتی که اندازه و زمان پردازش کارها موافق یکدیگر هستند، پرداخته‌اند. در مرجع [۹]، محققان یک الگوریتم شاخه و حد برای مساله *BPM-CMAX* ارائه کرده‌اند که قابلیت حل مسائل با ابعاد به نسبت بزرگ را داراست. اخیراً نیز مولفان مقاله حاضر روشی بر مبنای شاخه و قیمت را برای حل این مساله ارائه کرده‌اند [۲۲] که قابلیت رقابت با روش شاخه و حد [۹] را دارد.

حل مسائل زمان‌بندی ماشین‌های پردازش‌گر انباشته با فرض وجود کارهای با اندازه نامساوی، با استفاده از الگوریتم‌های فرا ابتکاری نیز مورد توجه محققین قرار گرفته

است. در مرجع [۲۱] یک الگوریتم شبیه‌سازی بازپخت برای مساله *BPM-CMAX* ارائه شده و نشان داده شده است که روش معرفی شده در مقایسه با جواب‌های حاصل از حل مدل ریاضی مساله با استفاده از نرم افزار سیپلکس[۳] نتایج بهتری را در زمان کمتر بدست می‌دهد. در مرجع [۶] نیز یک الگوریتم ژنتیک ساده برای مساله *BPM-CMAX* ارائه شده و عملکرد آن با الگوریتم شبیه‌سازی بازپخت ارائه شده در مرجع [۲۱] مقایسه شده است. برای این مساله، در مرجع [۱۵] مولفان مقاله یک الگوریتم ژنتیک بر مبنای نمایش کروموزومی بر اساس کلیدهای تصادفی و یک الگوریتم ژنتیک ترکیبی که از یک نمایش کروموزومی ابداعی بهره می‌برد را ارائه کرده‌اند. نتایج محاسبات نشان می‌دهد که الگوریتم ژنتیک ترکیبی ارائه شده کارا بوده و جواب‌های بسیار بهتری را نسبت به الگوریتم شبیه‌سازی بازپخت ارائه شده در مرجع [۲۱] و الگوریتم ژنتیک مبتنی بر نمایش کروموزومی بر اساس کلیدهای تصادفی می‌دهد. مساله حداقل‌سازی هم‌زمان C_{max} و T_{max} در مرجع [۱۶] مورد مطالعه قرار گرفته است. برای این مساله مولفین یک الگوریتم ژنتیک دو هدفه ترکیبی ارائه کرده‌اند. نتایج محاسبات آنها نشان می‌دهد که این الگوریتم قابلیت یافتن جواب‌های نزدیک به جواب‌های بهینه پارتو را دارد. اخیرا نیز محققین از روش‌های فرا ابتکاری نسل جدید مانند جستجوی آزاد [۲۷] و تکامل تفاضلی [۲۶] برای حل مساله *BPM-CMAX* بهره برده‌اند. تعدادی از محققین به مطالعه مساله حداقل‌سازی C_{max} روی ماشین‌های پردازش‌گر انباشته موازی [۴]، [۵]، [۱۴]، [۲۴] و هم‌چنین روی یک سیستم فلوشاپ از ماشین‌های پردازش‌گر انباشته [۷]، [۱۲]، [۱۹] با فرض وجود کارهای با اندازه نامساوی پرداخته‌اند.

همان‌طور که گفته شد مسائل زمان‌بندی با فرض وجود کارهای با اندازه نامساوی اولین‌بار در مرجع [۸] معرفی شدند. از جمله فرضیات مدل زمان‌بندی در نظر گرفته شده در آنجا، وجود خانواده‌های ناسازگار کارها است. با هدف حداقل‌سازی Σw_iC_i، در مرجع [۸] نشان داده شده است که این مساله NP-hard بوده و یک مدل برنامه‌ریزی عدد صحیح ترکیبی برای مساله ارائه شده است. در مرجع [۲]، برای این مساله یک روش شاخه و کران و در مرجع [۱۷] تعدادی روش ابتکاری و یک الگوریتم ژنتیک بر مبنای نمایش کروموزومی بر اساس کلیدهای تصادفی ارائه شده است. در مرجع [۱۱] نیز یک چارچوب بر مبنای بهینه‌سازی کولنی مورچگان، در دو نسخه متفاوت ارائه شده است.

۳- حدود پایین برای مساله BPM-CMAX

در این بخش به معرفی روش‌های تولید حد پایین روی مقدار بهینه C_{max} می‌پردازیم. ابتدا به معرفی تنها روش موجود برای تولید حد پایین (LB_1) که در مرجع [۲۳] ارائه شده است پرداخته و سپس به معرفی رویه‌های جدید تولید حدود پایین می‌پردازیم. تعدادی از محققین [۲۳]، [۱۰]، [۱۵] از LB_1 برای سنجش کیفیت جواب‌های حاصل از الگوریتم‌های خود استفاده کرده‌اند و یا از آن در بدنه الگوریتم‌های تولید حدود پایین برای سایر مسائل زمان‌بندی استفاده کرده‌اند [۱۲]، [۱۴] و یا از آن در بدنه الگوریتم شاخه و حد استفاده کرده‌اند [۹].

الگوریتم LB_1

ابتدا همه کارها را به ترتیب نزولی زمان پردازش آنها مرتب کنید. با شروع از ابتدای لیست، نخستین کار را در تنها انباشته‌ای که به طور کامل پر نشده (ظرفیت آن کمتر از B است) قرار داده (اگر همه انباشته‌ها به حد بالای ظرفیت خود رسیده‌اند یک انباشته جدید تشکیل دهید) و چنانچه فضای لازم برای قرار دادن کار در انباشته مذکور وجود ندارد، قسمتی از کار را که ظرفیت خالی انباشته را به طور کامل اشغال می‌سازد درون انباشته قرار دهید. سپس یک انباشته جدید تشکیل داده و باقیمانده کار خرد شده را در آن قرار دهید. این فرآیند را تا زمانی‌که هیچ کاری در لیست باقی نماند تکرار نمایید.

مبنای الگوریتم فوق بر اساس آزادسازی است. بدین ترتیب که در آن فرض مجاز نبودن شکست کارها به اجزای با اندازه کوچکتر، آزاد شده است. زمان اجرای الگوریتم فوق از مرتبه $n\log n$ است. این پیچیدگی زمانی فقط به دلیل مرتب‌سازی کارهاست. اجرای سایر قسمت‌های الگوریتم در زمان $O(n)$ امکان‌پذیر است. اخیراً ثابت شده که نسبت عملکرد الگوریتم فوق در بدترین حالت برابر ۰/۵ است[۴] [۱].

۳-۱- ارائه حدود پایین جدید برای مساله *BPM-CMAX*

وقتی اندازه کارها نسبت به ظرفیت ماشین کوچک است، عملکرد متوسط LB_1 خوب تلقی می‌شود. در واقع ثابت شده که هر چه اندازه کارها نسبت به ظرفیت ماشین کوچکتر شود عملکرد LB_1 در بدترین حالت بهتر می‌شود [۱]. همان‌طور که در مرجع [۱۵] گزارش شده است، برای بسیاری از مسائلی که در آنها اندازه کارها نسبت به ظرفیت ماشین کوچک است، مقدار حد پایین تولید شده به‌وسیله LB_1 با مقدار بهینه C_{max} برابر است. اما از عملکرد LB_1 زمانی کاسته می‌شود که اندازه کارها به نسبت بزرگ باشد. از این نقطه ضعف می‌توان برای تولید حدود پایین جدید استفاده کرد.

فرض کنید کارهای در دسترس برای زمان‌بندی را در قالب دو مجموعه تفکیک نماییم. مجموعه کارهایی که ابعاد آنها بزرگتر از نصف ظرفیت ماشین یعنی $B/۲$ است و مجموعه کارهایی که ابعاد آنها کوچکتر از نصف ظرفیت ماشین است. بدیهی است که هیچ دو کاری که اندازه آنها بزرگتر از نصف ظرفیت ماشین است نمی‌توانند در قالب یک انباشته به‌وسیله ماشین پردازش شوند، زیرا در این‌صورت محدودیت ظرفیت ماشین نقض می‌شود. بنابراین، برای هر یک از این نوع کارها یک انباشته جداگانه مورد نیاز است. همچنین کاری با اندازه ε ($B/\varepsilon \le ۲$) نیز نمی‌تواند با کاری با اندازه بزرگتر از B-ε در یک انباشته قرار گیرد. بنابراین، برای رسیدن به یک حد پایین روی مقدار بهینه تابع هدف می‌توان به ازای مقادیر مختلف ε ($B/\varepsilon \le ۲$) فقط اجازه داد کارهایی که اندازه آنها در دامنه $[\varepsilon, B-\varepsilon]$ قرار دارد قابل شکست به اجزایشان باشند. فرض کنید $S(u,v)=\{g \mid u < s_g \le v,\ g\in J\}$ دلالت بر مجموعه کارهایی داشته باشد که اندازه آنها بزرگتر از u و کوچکتر یا مساوی با v است. به طور مشابه می‌توان تعریف کرد: $\bar{S}(u,v)=\{g \mid u \le s_g \le v,\ g\in J\}$. رویه زیر با نام NLB یک حد پایین معتبر را روی مقدار بهینه $C_{\max}$ می‌دهد.

الگوریتم NLB

۱- به ازای یک مقدار $\varepsilon \in [0, B/۲]$، یک حد پایین روی مقدار بهینه Cmax را به صورت زیر محاسبه کنید:

$$C^{NLB^{\varepsilon}}=\sum\nolimits_{j\in S(B-\varepsilon,B)} p_j + C^{LB_1}_{\bar{S}(\varepsilon,B-\varepsilon)}$$

که در آن $C^{LB_1}_{\bar{S}(\varepsilon,B-\varepsilon)}$ مقدار حد پایین بدست آمده به‌وسیله LB_1، بر اساس مجموعه کارهایی است که متعلق به مجموعه $\bar{S}(\varepsilon,B-\varepsilon)$ هستند.

۲- با تکرار گام ۱ به ازای همه مقادیر ε، بزرگترین مقدار بدست آمده برای $C^{NLB^{\varepsilon}}$، یک حد پایین روی مقدار بهینه $C_{\max}$ است. **یعنی:**

$$C^{NLB}=\max_{\varepsilon\in[0,B/2]}\{C^{NLB^{\varepsilon}}\}. \qquad (۱)$$

لم ۱. C^{NLB} یک حد پایین روی مقدار بهینه $C_{\max}$ است. همچنین $C^{NLB} \ge C^{LB_1}$.

اثبات. به ازای یک مقدار مفروض ε، از آنجا که هیچ یک از کارهایی که اندازه آنها در بازه $[\varepsilon, B$-$\varepsilon]$ است نمی‌توانند در انباشته‌ای قرار گیرند که کاری با اندازه بزرگتر از B-ε در آن قرار دارد، داریم:

$$C^{NLB^{\varepsilon}}=\sum\nolimits_{j\in S(B-\varepsilon,B)} p_j + C^{LB_1}_{\bar{S}(\varepsilon,B-\varepsilon)} \le \sum\nolimits_{j\in S(B-\varepsilon,B)} p_j + C^{*}_{\bar{S}(\varepsilon,B-\varepsilon)} = C^{*}_{\bar{S}(\varepsilon,B)}$$

از آنجا که $\bar{S}(\varepsilon,B)\subseteq J$، پس $C^{*}_{\bar{S}(\varepsilon,B)} \le C^{*}$. در نتیجه $C^{NLB}\le C^{*}$. همچنین به ازای $\varepsilon=0$ داریم $C^{NLB^{0}}=C^{LB_1}$ که دلالت بر $C^{LB_1}\le C^{NLB}$ دارد.

از آنجا که هیچ دو کاری که اندازه آنها در بازه $[B/۲, B$-$\varepsilon]$ قرار می‌گیرد نمی‌توانند در یک انباشته باشند، هنگامی که $C^{LB_1}_{\bar{S}(\varepsilon,B-\varepsilon)} < \sum\nolimits_{j\in S(B/2,B-\varepsilon)} p_j$، ممکن است بتوان مقدار C^{NLB} را قوی‌تر نمود. داریم:

$$\begin{aligned}
C^{LB_2} &= \max_{\varepsilon\in[0,B/2]}\left\{\begin{matrix}\sum\nolimits_{j\in S(B-\varepsilon,B)} p_j + \\ \max\left\{\sum\nolimits_{j\in S(B/2,B-\varepsilon)} p_j, C^{LB_1}_{\bar{S}(\varepsilon,B-\varepsilon)}\right\}\end{matrix}\right\} \\
&= \max_{\varepsilon\in[0,B/2]}\left\{\begin{matrix}\sum\nolimits_{j\in S(B/2,B)} p_j + \\ \max\left\{0, C^{LB_1}_{\bar{S}(\varepsilon,B-\varepsilon)} - \sum\nolimits_{j\in S(B/2,B-\varepsilon)} p_j\right\}\end{matrix}\right\} \\
&= \max\{C^{*}_{S(B/2,B)}, \max_{\varepsilon\in[0,B/2]}\{C^{NLB^{\varepsilon}}\}\} \qquad (۲)
\end{aligned}$$

که در آن $C^{*}_{S(B/2,B)}=\sum\nolimits_{j\in S(B/2,B)} p_j$.

در نتیجه از لم ۱ می‌توان دریافت که $C^{LB_1}\le C^{LB_2}$. همچنین از آنجا که در گام ۲ در الگوریتم NLB فقط باید به تعداد مقادیر متمایز $\varepsilon\in[0, B/۲]$، مقدار $C^{NLB^{\varepsilon}}$ را محاسبه نمود، می‌توان نتیجه گرفت که پیچیدگی محاسباتی C^{LB_2} در بدترین حالت برابر با $O(n^2\log n)$ است.

همان‌طور که دیده شد، C^{LB_2} از طریق تفکیک مجموعه کارهای در دسترس در قالب دو مجموعه بدست آمد. بدین ترتیب ممکن است بتوان از ایده تفکیک کارها در قالب سه مجموعه برای ارائه یک حد پایین دیگر (و در صورت امکان قوی‌تر) به صورت رابطه (۳) بهره برد:

$$C^{LB_3}=\max\left\{C^{*}_{S(B/3,B)}, \max_{\varepsilon\in[0,B/3]}\{C^{NLB^{\varepsilon}}\}\right\} \qquad (۳)$$

که در آن $C^{*}_{S(B/3,B)}$ مقدار بهینه $C_{\max}$ در زیر مساله‌ای از مساله اصلی است که در آن فقط کارهای با اندازه بزرگتر از $B/۳$ در نظر گرفته شده‌اند.

خوشبختانه، این موضوع به صورت زیر امکان‌پذیر است. بنابر تعریف، مجموعه نخست شامل کارهایی است که اندازه آنها بزرگتر از $B/۲$ هستند. مجموعه دوم نیز به وسیله کارهایی ساخته می‌شود که اندازه آنها در بازه $[B/۳, B/۲)$ قرار می‌گیرد. در پایان، مجموعه سوم شامل کارهایی است که در مجموعه اول و دوم قرار نمی‌گیرند. دوباره هر یک از کارهایی که در

مجموعه نخست قرار می‌گیرند باید در یک انباشته جداگانه قرار گیرند. البته کارهایی که در مجموعه دوم قرار دارند می‌توانند حداکثر با یک کار از مجموعه نخست که اندازه آن در بازه $[B/۲, ۲B/۳)$ قرار دارد، در یک انباشته گروه‌بندی شوند. فرض کنید H مجموعه کارهایی باشد که عضو مجموعه نخست بوده و می‌توانند با یکی از کارهای متعلق به مجموعه دوم در یک انباشته قرار گیرند. با توجه به تفکیک‌بندی کارها به صورت فوق می‌توان $C^*_{S(B/3,B)}$ را به صورت رابطه (٤) ارائه نمود:

$$C^*_{S(B/3,B)} = \sum_{j\in S(B/2,B)} p_j + C^*_X - \sum_{j\in H} p_j \qquad (٤)$$

کـه در آن C^*_X مقـدار بهینـه Cmax در زیـر مسـاله‌ای از مسـاله اصلی است که در آن فقط کارهایی کـه انـدیس‌شـان متعلـق بـه مجموعه $X = S(B/3, B/2) \cup H$ است مورد نظـر قـرار دارنـد. بنـابراین بـا معرفـی چگـونگی محاسـبه C^*_X مـی‌تـوان بـه مقدار C^{LB_3} رسید. C^{LB_3} به صورت رابطه (٥) نیز قابل معرفـی است:

$$C^{LB_3} = \max_{\varepsilon\in[0,B/3]} \left\{ \begin{array}{l} \sum_{j\in S(B-\varepsilon,B)} p_j + \\ \max\left\{ \begin{array}{l} \sum_{j\in S(B/2,B-\varepsilon)} p_j + C^*_X - \sum_{j\in H} p_j \\ , C^{LB_1}_{\bar{S}(\varepsilon,B-\varepsilon)} \end{array} \right\} \end{array} \right\}$$

$$= \max_{\varepsilon\in[0,B/3]} \left\{ \begin{array}{l} \sum_{j\in S(B/2,B)} p_j + C^*_X - \sum_{j\in H} p_j + \\ \max\left\{ 0, \left(\begin{array}{l} C^{LB_1}_{\bar{S}(\varepsilon,B-\varepsilon)} - \sum_{j\in S(B/2,B-\varepsilon)} p_j \\ - C^*_X + \sum_{j\in H} p_j \end{array} \right) \right\} \end{array} \right\} \qquad (٥)$$

حال نشان خواهیم داد که چگونه مساله زمان‌بندی کارهای متعلق به مجموعه X و مقدار بهینه تابع هدف آن یعنی C^*_X از طریق تبدیل به یک مساله شناخته شده در نظریه گراف قابل حل است. ابتدا باید توجه داشت که به طور کلی، حداکثر دو کار متعلق به X می‌توانند با یکدیگر در یک انباشته گروه‌بندی شوند. در نتیجه یادآور می‌شویم که کارهای متعلق به X که اندازه آنها در بازه $[B/۳, B/۲)$ قرار می‌گیرد (کارهای متعلق به مجموعه دوم) فقط می‌توانند حداکثر با یک کار دیگر متعلق به X که آن نیز از مجموعه دوم آمده در یک انباشته گروه‌بندی شوند و یا می‌توانند حداکثر با یک کار متعلق به X که از مجموعه H آمده گروه‌بندی شوند. همچنین هیچ دو کار متعلق به X که از مجموعه H آمده باشند نمی‌توانند در یک انباشته قرار گیرند. به عبارت دیگر می‌توان این‌طور در نظر گرفت که این کارها با یکدیگر ناسازگار هستند. به طور مشابه نیز می‌توان رابطه سازگاری بین دو کار را در قالب امکان قرارگیری آنها در یک انباشته تعریف نمود. برای تعریف رابطه سازگاری و ناسازگاری میان کارهای متعلق به X می‌توان از گراف $G_X(V,E)$ استفاده نمود که در آن V مجموعه گره‌های گراف است که به هر گره یک کار متعلق به X نظیر می‌شود. مجموعه E نیز مجموعه یال‌های گراف است. هنگامی که دو کار (گره) i و j در مجموعه X با یکدیگر سازگار باشند یعنی $s_i + s_j \le B$، آنگاه یک یال بین گره‌های نظیر آنها وجود دارد. بنابر تعریف، همه کارهای متعلق به مجموعه H با یکدیگر ناسازگار هستند و بنابراین مجموعه گره‌های نظیر آنها تشکیل یک مجموعه استوار[۵] با اندازه $|H|$ را می‌دهند (منظور از یک مجموعه استوار در یک گراف، مجموعه‌ای از گره‌ها است که همگی ایزوله بوده و هیچ یالی بین آنها نیست). همچنین همه کارهای متعلق به مجموعه دوم، دو به دو سازگار بوده و بنابراین مجموعه گره‌های نظیر آنها تشکیل یک خوشه[۶] با اندازه $|S(B/۳, B/۲)|$ می‌دهند (منظور از یک خوشه در یک گراف مجموعه‌ای از گره‌ها است که در آن هر جفت گره با یک یال به یکدیگر متصل هستند). همچنین وقتی یک کار متعلق به X که اندیس آن از مجموعه H آمده با کاری دیگر متعلق به X که اندیس آن در مجموعه دوم قرار دارد سازگار باشد یک یال بین گره‌های نظیر آنها در گراف G_X وجود دارد. در نظریه گراف، گرافی نظیر G_X که از یک خوشه و از یک مجموعه استوار تشکیل شده است، گراف شکسته[۷] گفته می‌شود.

با توجه به اینکه حداکثر دو کار متعلق به X می‌توانند در یک انباشته قرار گیرند، می‌توان برای هر یک از کارهای متعلق به X یک اندازه واحد در نظر گرفت (اندازه کار $i \in X$ را از s_i به یک کاهش داد) و ظرفیت انباشته را از B به ۲ کم کرد. همچنین، این محدودیت نیز باید اعمال شود که کارهایی که با یکدیگر ناسازگارند نباید در یک انباشته قرار گیرند. بنابراین مساله یافتن زمان‌بندی بهینه کارهای متعلق به X (با مقدار تابع هدف C^*_X) با فرضیات بخش [۱-مقدمه]، به مساله بخش‌بندی گراف G_X با خوشه‌هایی (انباشته‌هایی) با اندازه حداکثر ۲ و با هدف حداقل‌سازی مجموع زمان پردازش خوشه‌ها (انباشته‌ها) تبدیل می‌شود. با این تبدیل می‌توان مقدار C^*_X را از طریق حل بهینه مساله بخش‌بندی با خوشه‌ها[۸] در زمان $O(|X|^3)$ از طریق الگوریتم زیر موسوم به $MWMA$ بدست آورد [۳].

الگوریتم *MWMA*

۱. یک گراف وزنی جدید $G^v_X(V,E)$ بسازید که در آن هر یال که گره‌های i و j را به هم متصل می‌سازد دارای وزن $\min\{p_i, p_j\}$ باشد.

۲. حداکثر تطابق وزنی[9] را در G_X^v بیابید و مقدار آن را برابر با $W(X)$ قرار دهید.

۳. انباشته‌ها را به ترتیب زیر تشکیل دهید:

– هر دو گره‌ای که یال بین آنها در مجموعه تطابق تشکیل شده در گام ۲ قرار می‌گیرد را در یک انباشته قرار دهید.

– هر یک از گره‌های باقی‌مانده را در یک انباشته مجزا قرار دهید.

۴. مقدار C_X^* برابر است با مجموع زمان‌های پردازش همه کارهای متعلق به مجموعه X منهای ارزش تطابق وزنی بدست آمده در گام ۲.

قضیه ۱. به ازای هر نمونه مساله از *BPM-CMAX* که در آن اندازه همه کارها بزرگتر از $B/3$ است، زمان‌بندی بهینه در زمان $O(n^3)$ به‌وسیله الگوریتم *MWMA* بدست می‌آید.

با اینکه امکان‌پذیر است به طریقی مشابه روند فوق را برای دستیابی به C^{LB_4}، C^{LB_5} و غیره توسعه داد، اما حل بهینه زیر مسائل بدست آمده که در واقع مسائل بخش‌بندی گراف با خوشه‌هایی با حداکثر اندازه ۳، ۴ و یا بیشتر هستند در زمان چند جمله‌ای امکان‌پذیر نیست.

قضیه ۲. $C^{LB_3} \geq C^{LB_2}$

اثبات. برای اثبات، حالت‌های زیر را در نظر می‌گیریم:

حالت ۱) $\varepsilon \in [0, B/3]$

در این حالت کافی است نشان دهیم:

$$\sum_{j\in S(B/2,B-\varepsilon)} p_j \le \sum_{j\in S(B/2,B-\varepsilon)} p_j + C_X^* - \sum_{j\in H} p_j.$$

از نامساوی فوق داریم $C_X^* - \sum_{j\in H} p_j \ge 0$ که همواره برقرار است زیرا $H \subseteq X$.

حالت ۲) $\varepsilon \in (B/3, B/2)$

کافی است به ازای $\varepsilon \in (B/3,B/2)$ ثابت کنیم $C^{NLB^{\varepsilon}} \le C^*_{S(B/3,B)}$. فرض کنید به ازای $B/3 \le s < B/2$، مجموعه کارهای متعلق به $S(B/2, B-s)$ که با کارهای متعلق به $S(s, B/2)$ سازگار هستند را با $h(s)$ نشان دهیم. بنابر تعریف داریم $H=h(B/3)$. فرض کنید $x(s) = S(s, B/2) \cup h(s)$. به ازای $\varepsilon \in (B/3,B/2)$ و $\sigma > 0$، می‌خواهیم ثابت کنیم:

$$C^{LB_1}_{\bar{S}(\varepsilon,B-\varepsilon)} - \sum_{j\in S(B/2,B-\varepsilon)} p_j \le C^*_{x(\varepsilon-\sigma)} - \sum_{j\in h(\varepsilon-\sigma)} p_j$$

هم‌چنین فرض می‌کنیم که شرایط زیر برقرار باشند:

۱) $\varepsilon - \sigma > B/3$.

۲) کاری وجود ندارد که اندازه آن در بازه‌های $[\varepsilon-\sigma,\varepsilon)$ و $(B-\varepsilon, B-\varepsilon+\sigma]$ قرار گیرد.

از فرض دوم نتیجه می‌شود: $\bar{S}(\varepsilon, B/2) = S(\varepsilon-\sigma, B/2)$ و $\bar{S}(B/2, B-\varepsilon) = S(B/2, B-\varepsilon+\sigma)$. بنابراین داریم $\bar{S}(\varepsilon, B-\varepsilon) = S(\varepsilon-\sigma, B-\varepsilon+\sigma)$. بر این اساس می‌توان استدلال نمود که محاسبه $C^{LB_1}_{\bar{S}(\varepsilon,B-\varepsilon)} - \sum_{j\in S(B/2,B-\varepsilon)} p_j$ و $C^*_{x(\varepsilon-\sigma)} - \sum_{j\in h(\varepsilon-\sigma)} p_j$ بر مبنای یک مجموعه یکسان از کارها صورت می‌پذیرد و بنابراین حکم ثابت است. این نتیجه‌گیری از آنجاست که در محاسبه سمت چپ نامساوی با توجه به مجاز بودن شکست کارها، از ظرفیت انباشته‌ها به طور کامل استفاده می‌شود. در حالی‌که در محاسبه سمت راست نامساوی ممکن است از ظرفیت انباشته‌ها به طور کامل استفاده نشود. حال می‌خواهیم درستی نامساوی $C^*_{x(\varepsilon-\sigma)} - \sum_{j\in h(\varepsilon-\sigma)} p_j \le C_X^* - \sum_{j\in H} p_j$ را نشان دهیم. فرض کنید s و $\hat{s}$ به گونه‌ای موجود باشند که $B/3 \le \hat{s} \le s < B/2$. بدیهی است که $S(s, B/2) \subseteq S(\hat{s}, B/2)$ و $S(B/2, B-s) \subseteq S(B/2, B-\hat{s})$. هم‌چنین داریم $h(s) \subseteq h(\hat{s})$ که نتیجه می‌دهد $x(s) \subseteq x(\hat{s})$. از گام ۴ الگوریتم *MWMA* داریم $C^*_{x(s)} = \sum_{j\in x(s)} p_j - W(x(s))$ که در آن $W(x(s))$ مقدار حداکثر تطابق وزنی محاسبه شده در گام ۲ به ازای گراف $G_{x(s)}$ است که به‌وسیله کارهای متعلق به $x(s)$ ساخته شده است. می‌خواهیم ثابت کنیم $C^*_{x(s)} - \sum_{j\in h(s)} p_j \le C^*_{x(\hat{s})} - \sum_{j\in h(\hat{s})} p_j$. در واقع معادل این نامساوی می‌تواند به صورت $W(x(s)) \ge W(x(\hat{s}))$ تعریف شود. نامساوی اخیر نیز همواره برقرار است زیرا $x(s) \subseteq x(\hat{s})$. با تعریف $s = \varepsilon - \sigma$ و $\hat{s} = B/3$ نتیجه مورد نظر بدست می‌آید.

حالت ۳) $\varepsilon = B/2$.

به ازای $\varepsilon = B/2$ داریم:

$$C^{LB_2^{B/2}} = \sum_{j\in S(B/2,B)} p_j + C^{LB_1}_{\bar{S}(B/2,B/2)}.$$

همان‌طور که دیده می‌شود $C^{LB_2^{B/2}}$ به ازای مجموعه کارهایی که اندیس آنها متعلق به $\bar{S}(B/2,B)$ است محاسبه می‌شود. بنابراین داریم $C^{LB_2^{B/2}} \le C^*_{\bar{S}(B/2,B)}$. از آنجا که $\bar{S}(B/2,B) \subseteq S(B/3,B)$ لذا $C^*_{\bar{S}(B/2,B)} \le C^*_{S(B/3,B)}$. نامساوی اخیر نیز درستی حکم را به ازای $\varepsilon = B/2$ می‌دهد.

مثال. برای آگاهی بیشتر از نحوه محاسبه حدود پایین، در این قسمت از یک مثال عددی کمک گرفته می‌شود که در آن $B=10;\ s_1=7; s_2=5; s_3=4; s_4=6; s_5=5; s_6=9; s_7=1$ و $p_1=10;\ p_2=14;\ p_3=13;\ p_4=1;\ p_5=7;\ p_6=19;\ p_7=6;$.

حل: برای محاسبه C^{LB_2} داریم:

$$\varepsilon = 0 \Rightarrow C^{NLN^0} = C^{LB_1}_{\bar{S}(0,B)} = C^{LB_1} = 49$$

$$\varepsilon = 1 \Rightarrow C^{NLN^1} = \sum_{j \in S(9,10)} P_j + C^{LB_1}_{\bar{S}(1,9)} = 0 + 49$$

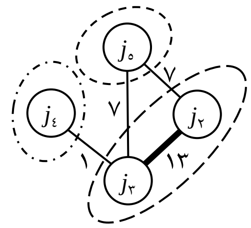
$$\varepsilon = 4 \Rightarrow C^{NLN^4} = \sum_{j \in S(6,10)} P_j + C^{LB_1}_{\bar{S}(4,6)} = 10 + 19 + 21 = 50$$

$$\varepsilon = 5 \Rightarrow C^{NLN^5} = \sum_{j \in S(5,10)} P_j + C^{LB_1}_{\bar{S}(5,5)} = 1 + 10 + 19 + 14 = 44$$

$$C^*_{S(B/2,B)} = \sum_{j \in S(5,10)} P_j = 30$$

$$C^{LB_2} = \max\{30,49,49,50,44\} = 50$$

برای محاسبه C^{LB_3} کافی است مقدار $C^*_{S(B/3,B)}$ را محاسبه نماییم. ابتدا گراف $G^v_X(V,E)$ را به شکل زیر تشکیل می‌دهیم:



می‌توان دریافت که حداکثر تطابق وزنی با مقدار ۱۳ به ازای قرار گرفتن یال j_2-j_3 در مجموعه تطابق ایجاد می‌شود. داریم $C^*_X =$ (۷+۱+۱۳+۱٤)−۱۳=۲۲. مقدار $C^*_{S(B/3,B)}$ نیز به صورت زیر محاسبه می‌شود.

$$C^*_{S(B/3,B)} = \sum_{j \in \{j_1, j_4, j_6\}} p_j + C^*_X - \sum_{j \in \{j_4\}} p_j = \text{۳۰+۲۲−۱=۵۱}$$

بنابراین داریم: $\max C^{LB_2} = \{$۵۱,٤۹,٤۹$\}$=۵۱.

۳- نتایج محاسباتی

برای بررسی کیفیت حدود پایین ارائه شده و انجام مقایسات با حد پایین موجود، یک برنامه کامپیوتری در نرم افزار مطلب ۷,۳,۰ نوشته شده و آزمایش‌ها روی یک کامپیوتر پنتیوم ٤ با ۲/٦٦ گیگا هرتز سرعت پردازش و ۵۱۲ مگابایت حافظه انجام شده است. برای انجام آزمایش‌های محاسباتی، ۱۰ مجموعه مساله در نظر گرفته شده که تعداد کارهای آنها از ۱۰ تا ۱۰۰ است. برای هر مجموعه، ۱۰۰ نمونه مساله به صورت تصادفی تولید شده و مقادیر حدود پایین برای آنها حساب شده‌اند. برای کلیه مسائل، اندازه کارها به طور تصادفی در بازه اعداد صحیح بین ۱ تا ۹۹ و زمان‌های پردازش نیز در بازه اعداد صحیح بین ۸۰ تا ۱۲۰ تولید شده‌اند. نتایج در جدول (۱) آمده است. در جدول (۱) عملکرد حدود پایین بر اساس کیفیت حدود و نیز زمان اجرا سنجیده شده است. برای سنجش کیفیت حدود (LB) روی هر مجموعه از مسائل، از رابطه $AVG((C^{UB} - C^{LB})/C^{LB})$ استفاده شده است که در آن $AVG(.)$ تابع میانگین است که روی نتایج حاصل از ۱۰۰ نمونه مساله تولید شده برای هر مجموعه اعمال می‌شود. از آنجا که دسترسی به جواب‌های بهینه به دلیل سرکش بودن مساله به آسانی امکان‌پذیر نیست، کیفیت حدود پایین در مقایسه با یک حد بالا (C^{UB}) سنجیده شده است. برای تولید حد بالا نیز از الگوریتم FFLPT استفاده شده است [۲۳]. به تجربه ثابت شده است که این الگوریتم بسیار سریع بوده و به طور معمول جواب‌های نزدیک به بهینه را تولید می‌نماید. البته اگرچه شاخص کیفیت گزارش شده در جدول (۱) معرف عملکرد دقیق حدود نیست اما به درستی می‌تواند روند عملکرد حدود را نشان دهد. از آنجا که عملکرد حدود بر مبنای فاصله آنها نسبت به یک نقطه مرجع واحد (حد بالا) سنجیده شده است، بنابراین صرف نظر از ماهیت نقطه مرجع، نتایج گزارش شده به طور دقیق میزان قوت هر یک از حدود را نشان می‌دهند. به عنوان شاخص دیگری از عملکرد، زمان اجرای هر یک از حدود نیز گزارش شده‌اند. قابل توجه است که هنگام محاسبه C^{LB_3}، برای حل مساله حداکثر تطابق وزنی در گام دوم الگوریتم $MWMA$ از جعبه ابزار بهینه‌سازی موجود در نرم‌افزار مطلب ۷,۳,۰ استفاده شده است که بر مبنای روش شاخه و حد عمل می‌نماید. اگرچه برای یافتن تطابق وزنی بیشینه در یک گراف الگوریتم‌های مؤثر با زمان چند جمله‌ای چنان توسعه داده شده‌اند که ظرف کسری از ثانیه می‌توانند زیر مسائل مربوطه را حل نمایند، اما این الگوریتم‌ها بسیار پیچیده بوده و پیاده‌سازی آنها دشوار است. همان‌طور که به طور نظری نیز انتظار می‌رود، در میان حدود پایین مقایسه شده، LB_1 ضعیف‌ترین و LB_3 قوی‌ترین عملکرد را از خود نشان داده‌اند. البته از نقطه نظر زمان اجرا، این مساله وارون است. همان‌طور که نتایج نشان می‌دهند عملکرد LB_2 نزدیک به ۲/۵٪ بهبود را نسبت به عملکرد LB_1 نشان می‌دهد. این مساله بین LB_2 و LB_3 نیز برقرار است. اما این‌بار LB_2 ضعیف‌تر است.

جدول (۱): مقایسه بین حدود پایین

تعداد کارها (n)	LB_1 کیفیت (%)	LB_1 زمان (ثانیه)	LB_2 کیفیت (%)	LB_2 زمان (ثانیه)	LB_3 کیفیت (%)	LB_3 زمان (ثانیه)
۱۰	۸/۹٤	۰/۰۰	٤/۹۸	۰/۰۰	٤/۳٤	۰/۰۱
۲۰	۱۲/۵۱	۰/۰۰	۸/۸۰	۰/۰۰	۵/۷٦	۰/۱۳
۳۰	۱۲/۸۲	۰/۰۰	۹/۷۳	۰/۰۱	٦/٦۹	۰/۵۰
٤۰	۱۲/٤۲	۰/۰۰	۹/۷۰	۰/۰۱	٦/۷۲	۰/٦۳
۵۰	۱۱/۹۲	۰/۰۰	۹/٦۳	۰/۰۲	٦/۷۲	۰/۷٤
٦۰	۱۱/٤۹	۰/۰۰	۹/٤۰	۰/۰۳	٦/۹۲	۰/۸۵
۷۰	۱۱/۰۵	۰/۰۰	۹/۰۵	۰/۰۵	٦/۷٦	۰/۹۷
۸۰	۱۰/۹۲	۰/۰۰	۹/۰۲	۰/۰۷	٦/٤۰	۱/۲۳
۹۰	۱۰/۵۲	۰/۰۰	۸/۷٤	۰/۹	٦/۳۵	۱/٤۰
۱۰۰	۹/۷۸	۰/۰۰	۷/۹٦	۰/۱۱	۵/٦۰	۱/۵۵
متوسط	۱۱/۲۳		۸/۷۰		۶/۲۲	

٤- جمع‌بندی و نتیجه‌گیری

در این مقاله روش‌های جدیدی برای تولید حدود پایین قوی روی مقدار بهینه زمان انجام عملیات بر روی همه کارها به‌وسیله یک ماشین پردازنده انباشته با فرض وجود کارهای با اندازه نامساوی ارائه شده و ثابت شد که عملکرد آنها حداقل به خوبی عملکرد تنها حد پایین موجود در ادبیات موضوع است. همچنین ثابت شد که به طور خاص عملکرد یکی از حدود پایین معرفی شده باید حداقل به خوبی عملکرد حد پایین دیگر باشد.

برای تحقیقات آینده، استفاده از حدود پایین ارائه شده در بدنه الگوریتم‌هایی مانند شاخه و کران و سنجش موازنه بین زمان و کیفیت قابل توجه است. همچنین توسعه حدود پایین برای مسائل زمان‌بندی در سیستم‌های تولید پیچیده که در آن ماشین‌های پردازش‌گر انباشته به صورت موازی و یا فلوشاپ فعالیت می‌کنند نیز قابل ملاحظه است.

٥- مراجع

[١] حسین زاده کاشان، علی؛ "ارائه حدود بالا و پایین برای مسائل زمان‌بندی تک ماشین و جریان کارگاهی در سیستم‌های تولید انباشته‌ای با فرض وجود نیازمندی ظرفیتی متفاوت برای کارها". رساله دکتری مهندسی صنایع، دانشگاه صنعتی امیرکبیر، ١٣٨٨.

[٢] Azizoglu, M.; Webster, S.; "Scheduling a batch processing machine with incompatible job families". Computers & Industrial Engineering, vol. 39, p.p. 325-335, 2000.

[٣] Boudhar, M.; Finke, G.; "Scheduling on a batch machine with job compatibilities", Belgian Journal of Operations Research, Statistics and Computer Science, vol. 40, p.p. 69–80, 2000.

[٤] Chang, P.Y.; Damodaran, P.; Melouk, S.; "Minimizing makespan on parallel batch processing machines", International Journal of Production Research, vol. 42, p.p. 4211-4220, 2004.

[٥] Damodaran, P., Chang, P.Y.; "Heuristics to minimize makespan of parallel batch processing machines", International Journal of Advance Manufacturing Technology, vol. 37, p.p. 1005-1013, 2008.

[٦] Damodaran, P.; Manjeshwar, P.K.; Srihari, K.; "Minimizing makespan on a batch-processing machine with non-identical job sizes using genetic algorithms", International journal of production economics, vol. 103, p.p. 882-891, 2006.

[٧] Damodaran, P.; Srihari, K.; "Mixed integer formulation to minimize makespan in flow shop with batch processing machines". Mathematical and Computer Modeling, vol. 40, p.p. 1465-1472, 2004.

[٨] Dobson, G.; Nambimadom, R.S.; "The batch loading and scheduling problem", Research Report, Simon School of Business Administration. University of Rochester, Rochester, NY, 1992.

[٩] Dupont, L.; Dhaenens-Flipo, C.; "Minimizing the makespan on a batch machine with non-identical job sizes: an exact procedure", Computers & Operations Research, vol. 29, p.p. 807-819,2002.

[١٠] Dupont, L.; Jolai Ghazvini, F.; "Minimizing Makespan on a Single Batch Processing Machine with Non identical Job Sizes", European journal of Automation Systems, vol. 32, p.p. 431-440, 1998.

[١١] Husseinzadeh Kashan, A.; Karimi, B.; "Scheduling a single batch-processing machine with arbitrary job sizes and incompatible job families: an ant colony framework", Journal of the Operational Research Society, vol. 59, p.p. 1269-1280, 2008.

[١٢] Husseinzadeh Kashan, A.; Karimi, B.; "An improved mixed integer linear formulation and several lower bounds for minimizing makespan on a flowshop with batch processing machines", International Journal of Advanced manufacturing Technology, vol. 40, p.p. 582-594, 2009.

[١٣] Husseinzadeh Kashan, A.; Karimi, B.; Fatemi Ghomi, S.M.T.; "A note on: Minimizing makespan on a single batch processing machine with non-identical job sizes", Theoretical Computer Science, vol. 410, p.p. 2754-2758, 2009.

[١٤] Husseinzadeh Kashan, A.; Karimi, B.; Jenabi, M.; "A hybrid genetic heuristic for scheduling parallel batch processing machines with arbitrary job sizes", Computers & Operations Research, vol. 35, p.p. 1084-1098, 2008.

[١٥] Husseinzadeh Kashan, A.; Karimi, B.; Jolai, F.; "Effective hybrid genetic algorithm for minimizing makespan on a single-batch-processing machine with non-identical job sizes", International Journal of Production Research, vol. 44, p.p. 2337-2360, 2006.

[١٦] Husseinzadeh Kashan, A.; Karimi, B.; Jolai, F.; "An effective hybrid multi-objective genetic algorithm for bi-criteria scheduling on a single batch processing machine with non-identical job sizes". Engineering Applications of Artificial Intelligence, vol. 23, p.p. 911-922, 2010.

[١٧] Koh, S.G.; Koo, P.H.; Kim, D.C.; Hur, W.S.; "Scheduling a single-batch-processing machine with arbitrary job sizes and incompatible job families", International Journal of Production Economics, vol. 98, p.p. 81–96, 2005.

[١٨] Lee, C.Y.; Uzsoy, R.; Martin-Vega, L.A.; "Efficient Algorithms for Scheduling Semiconductor Burn-in Operations", Operations Research, vol. 40, p.p. 764-775, 1992.

[١٩] Liao L.M.; Huang, C.J.; "Tabu search heuristic for two-machine flowshop with batch processing machines", Computers & Industrial Engineering, In Press, 2010.

[۲۰] Mathirajan, M.; Sivakumar, A.I.; “A Literature Review, Classification and Simple Meta-analysis on Scheduling of Batch Processors in Semiconductor”, International Journal of Advanced Manufacturing Technology, vol. 29, p.p. 990-1001, 2006.

[۲۱] Melouk, S.; Damodaran, P.; Cheng, P.Y.; “Minimizing make span for single machine batch processing with non-identical job sizes using simulated annealing”, International Journal of Production Economics, vol. 87, p.p. 141-147, 2004.

[۲۲] Rafiee Parsa, N.; Karimi, B.; Husseinzadeh Kashan, A.; “A branch and price algorithm to minimize makespan on a single batch processing machine with non-identical job sizes”, Computers & Operations Research, vol. 37, p.p. 1720-1730, 2010.

[۲۳] Uzsoy, R.; “A Single Batch Processing Machine with Non-identical Job Sizes”, International Journal of Production Research, vol. 32, p.p. 1615-1635, 1994.

[۲٤] Wang, H.M.; Chou, F.D.; “Solving the parallel batch-processing machines with different release times, job sizes, and capacity limits by metaheuristics”, Expert Systems with Applications, vol 37. p.p. 1510-1521, 2010.

[۲٥] Zhang, G.; Cai, X.; Lee, C.Y.; Wong, C.K.; “Minimizing makespan on a single batch processing machine with nonidentical job sizes”, Naval Research Logistics, vol. 48, p.p. 226-240, 2001.

[۲٦] Zhang, W.G.; Chen, H.P.; Lu, D.; Shao, H.; “A Novel Differential Evolution Algorithm for a Single Batch-processing Machine with Non-identical Job Sizes”, Proc. Fourth International Conference on Natural Computation, p.p. 447-451, 2008.

[۲۷] Zhang, Y.L.; Chen, H.P.; Shao, H.; Xu, R.; “Minimizing Makespan for Single Batch Processing Machine with Non-identical Job Sizes Using a Novel Algorithm: Free Search”, Proc. International Conference on Information Technology and Computer Science p.p. 180-183, 2009.

۶- زیرنویس‌ها

[۱] Batch processing machine

[۲] Burn-in oven

[۳] CPLEX

[٤] فرض کنید A یک الگوریتم تولید حد پایین برای مساله باشد. می‌توان عملکرد A را به ازای نمونه مساله Q به صورت C_Q^A/C_Q^* تعریف کرد که در آن C_Q^A مقدار حد پایین حاصل از A برای نمونه مساله Q است ($C_Q^A \le C_Q^*$). اگر به ازای هر نمونه مساله Q داشته باشیم $C_Q^A/C_Q^* \ge \rho$، که در آن ρ بزرگترین مقدار حقیقی به ازای تمامی نمونه مسائل ممکن Q است، آنگاه شاخصی از عملکرد A بدست می‌آید. بنابر تعریف، نسبت عملکرد A در بدترین حالت ($\rho(A)$) عبارت است از: $\rho(A)=\inf\{C_Q^A/C_Q^*, \forall Q\}$.

[٥] Stable set

[٦] Clique

[٧] Split graph

[٨] Partitioning with cliques

[٩] Maximum weight matching