



Minimizing the Number of Tardy Jobs in a Two-Machine flowshop problem with Non-Simultaneous Job Entrance

G. Moslehi*, A. hakimian & M. Abouei Ardakan

Ghasem Moslehi, Professor Isfahan University of Technology, moslehi@cc.iut.ac.ir

Ali hakimian, M.Sc student Isfahan University of Technology, a.hakimian@in.iut.ac.ir

Mostafa Abouei Ardakan, PhD student Isfahan University of Technology, m.abouei@in.iut.ac.ir

Keywords

Two machine Flowshop,
Number of tardy, Non-
Simultaneous Job Entrance,
Branch and Bound Algorithm,
Heuristic Algorithm

ABSTRACT

In this paper, minimizing the number of tardy jobs in two-machine flowshop scheduling with non-simultaneous job entrance is discussed. It is proven that the complexity of the problem is NP_hard. Therefore, a heuristic algorithm is proposed to solve the large scale problems. Besides, an exact branch and bound algorithm with utilizing heuristic algorithm as upper bound proposed to achieve optimal solution. Computational results demonstrate that branch and bound method solves problems with 28 jobs in the set High and 20 jobs in the set Low in a reasonable time. Results show the capability of the proposed upper bound, lower bounds and dominance rules. Also, it is shown that the average ratio of optimal solution to the heuristic one with the objective $\sum(1-U_i)$ is at most 1.11 which is smaller in contrast with other researches in the literature. This ratio proves efficacy of the proposed heuristic algorithm. Finally, according to efficiency of the presented approach, sample problems with large dimensions were solved and their results were displayed.

© 2013 IUST Publication, IJIEPM. Vol. 23, No. 4, All Rights Reserved

*
Corresponding author. Ghasem Moslehi
Email: moslehi@cc.iut.ac.ir



حداقل کردن تعداد کارهای دارای دیرکرد در مسئله دو ماشین با ورود غیر همزمان

قاسم مصلحی*، علی حکیمیان و مصطفی ابویی اردکان

چکیده:

کلمات کلیدی

در این مقاله مسئله زمان بندی فلو شاپ دو ماشین با در نظر گرفتن ورود غیر همزمان و با هدف کمینه سازی تعداد کارهای دارای دیرکرد بررسی شده است. در ابتدا پیچیدگی مسئله بررسی و ثابت شده که مسئله NP-hard است. بنابراین برای حل مسئله فوق یک الگوریتم ابتکاری که قابلیت حل مسائل با ابعاد خیلی بزرگ را دارد، ارائه شده است. همچنین به منظور حل بهینه مسئله از روش شاخه و کران با در نظر گرفتن الگوریتم ابتکاری به عنوان حد بالا بهره گرفته شده است. نتایج محاسباتی نشان می دهد که رویه شاخه و کران مسائل با ابعاد ۲۸ فعالیت در گروه High و ۲۰ فعالیت در گروه Low را در زمان منطقی و به طور کامل حل می کند، که این امر کارایی حد بالا، حدود پایین و اصول غلبه ارائه شده برای مسئله را نشان می دهد. همچنین نشان داده شد که متوسط نسبت جواب بهینه به الگوریتم ابتکاری با هدف $\sum (1-U_i)$ حداکثر ۱/۱۱ برابر می باشد که در مقایسه با الگوریتم های ارائه شده در تحقیقات مرتبط با کارهای دارای دیرکرد نسبت کوچکی می باشد. این نسبت نشان دهنده کارایی بالای الگوریتم ابتکاری است. با توجه به کارایی بالای الگوریتم ابتکاری، مسائل نمونه با ابعاد بزرگ نیز حل و نتایج آن ارائه شده است.

فلوشاپ دو ماشین
تعداد کارهای دارای دیرکرد
ورود غیر همزمان
الگوریتم شاخه و کران
الگوریتم ابتکاری

۱. مقدمه

در بسیاری از صنایع برآورد کردن زمان های تحویل امری حیاتی می باشد. لذا معیارهای وابسته به تاخیر همانند تعداد کارهای دارای دیرکرد از اهمیت ویژه ای برخوردار هستند و به عنوان

معیارهای ارزیابی زمان بندی ها مورد استفاده قرار می گیرند [۱]. در این مقاله مسئله زمان بندی فلو شاپ دو ماشین با زمان های ورود غیر همزمان و با هدف حداقل کردن تعداد کارهای دارای دیرکرد در حالت فلو شاپ جایگشتی^۲ مورد بررسی قرار می گیرد. در این حالت هر کاری که روی ماشین اول زودتر پردازش شود، روی ماشین دوم نیز زودتر پردازش می شود. به عبارت دیگر هیچ کاری مانند J وجود ندارد که پردازش آن روی ماشین اول بعد از i باشد ولی پردازش آن روی ماشین دوم قبل از i صورت پذیرد. کاربردهایی از این مسئله را می توان در کارخانه های نساجی و یا کارخانه های تولید آینه یافت. در فرایند تولید پارچه، پارچه ها (کارها) بر روی دو ماشین رنگ زنی و خشک کن و به صورت

تاریخ وصول: ۹۰/۲/۱۸

تاریخ تصویب: ۹۰/۵/۳۰

*نویسنده مسئول مقاله: قاسم مصلحی، استاد، دانشکده مهندسی صنایع و سیستم ها، دانشگاه صنعتی اصفهان moslehi@cc.iut.ac.ir
علی حکیمیان، دانشجوی کارشناسی ارشد، دانشکده مهندسی صنایع و سیستم ها، دانشگاه صنعتی اصفهان a.hakimian@in.iut.ac.ir
مصطفی ابویی اردکان، دانشجوی دکتری، دانشکده مهندسی صنایع و سیستم ها، دانشگاه صنعتی اصفهان m.abouei@in.iut.ac.ir

².Permutation Flowshop

سادیکف [۱۲] یک روش شاخه و کنترل^۴ را برای حل مسئله حداقل سازی وزنی تعداد کارهای دارای دیرکرد بر روی تک ماشین و با محدودیت زمان در دسترس بودن ارائه کردند. آنها در تحقیق خود از برنامه‌ریزی اعداد صحیح برای حل روش شاخه و کنترل استفاده کرده‌اند و جواب‌های امکان ناپذیر را توسط الگوریتم جدا کرده‌اند. هلاه و بولفین [۱۳] نیز برای حل همین مسئله یک روش شاخه و کران که در آن یک ساده‌سازی جانشینی که منجر به یک مسئله کوله پشتی چند انتخابه^۵ می‌شود پیشنهاد دادند. آنها از همین روش برای حداقل کردن وزنی تعداد کارهای دارای دیرکرد بر روی پردازشگرهای موازی نیز استفاده کردند [۱۴].

هی و همکاران [۱۵] رویکردی دوگانه از الگوریتم حریصانه روبه جلو^۶ را برای حداقل کردن تعداد کارهای دارای دیرکرد در مسئله تک ماشین ارائه کردند. آنها برای هر کار یک ضرب العجل^۷ نیز در نظر گرفتند. الگوریتم ارائه شده از رتبه $O(n \log n)$ می‌باشد. بولفین و هلاه [۱] با استفاده از مسئله کوله پشتی در حدود روش شاخه و کران مسئله حداقل سازی وزنی تعداد کارهای دارای دیرکرد در یک فلوشاپ دو ماشین، توانستند مسائلی تا ۱۰۰ کار حل نمایند. لودری و همکاران [۱۶] الگوریتمی ابتکاری برای حداقل سازی تعداد کارهای دارای دیرکرد در یک فلوشاپ پویا ارائه کرده‌اند. آنها فرض کردند که زمان ورود کارها متفاوت باشد.

رویز تورس و سنتنو [۱۷]، مسئله حداقل کردن تعداد کارهای دارای دیرکرد بر روی یک فلوشاپ جایگشتی و با منابع ثانویه^۸ را مورد بررسی قرار دادند. آنها برای تخصیص منابع ثانویه و تخصیص کارها به توالی از الگوریتم‌های ابتکاری مانند جستجو همسایگی و SA استفاده کردند. رویز تورس و همکاران [۱۸] در سال ۲۰۱۰ از روشی که در سال ۲۰۰۸ استفاده کرده بودند بار دیگر برای حل مسئله حداقل کردن تعداد کارهای دارای دیرکرد در یک فلوشاپ با منابع و عملیات انعطاف پذیر بهره بردند.

چوی و کیم [۱۹]، فلوشاپ دو ماشینی را در نظر گرفتند که در آن هر کار دومرتبه به فلوشاپ وارد می‌شود. یعنی هر کار باید یک مرتبه روی ماشین یک و سپس ماشین دو پردازش شود و دوباره تکرار شود. آنها میزان مجموع دیرکرد را در این مسئله با استفاده از روش شاخه و کران حداقل کردند. چوی و لی [۲۰]، مسئله حداقل کردن تعداد کارهای دارای دیرکرد در یک هیبرید فلوشاپ دو مرحله‌ای را با استفاده از روش شاخه و کران حل کردند. آنها برای حل مسئله در اندازه‌های بزرگ یک الگوریتم ابتکاری دو

متوالی پردازش می‌شوند. در فرایند نقره‌کاری برای تولید آینه نیز از دو ماشین متوالی استفاده می‌شود. ماشین اول کار پوشش دادن پشت آینه‌ها (کارها) توسط رنگ را بر عهده دارد و ماشین دوم عملیات خشک کردن را انجام می‌دهد.

اکثر پژوهش‌ها در مسائل فلوشاپ بر روی حداقل سازی زمان اتمام کل^۱ تمرکز داشته‌اند [۳ و ۲] و تاکنون بررسی‌های کمتری برای حداقل سازی تعداد کارهای دارای دیرکرد بر روی مسئله فلوشاپ انجام پذیرفته است. پیچیدگی حداقل سازی تعداد کارهای دارای دیرکرد در یک فلوشاپ دو ماشین حتی موقعی که تمام کارهای دارای زمان تحویل یکسان باشند نیز NP-hard است [۴]. حالت بدون وزن و تک ماشین این مسئله، با الگوریتم مور^۲ $O(n \log n)$ قابل حل می‌باشد [۵] و الگوریتم‌های بهینه فقط برای حالت‌های خاص توسعه داده شده‌اند.

لین [۶] یک الگوریتم از $O(n^2)$ را برای حل حالت‌های خاص مسئله فلوشاپ دو ماشین وقتی که همه کارها زمان‌های پردازش یکسان روی ماشین اول داشته و زمان‌های تحویل با زمان‌های پردازش سازگاری دارند ارائه کرد. لاولر و مور [۷] یک الگوریتم بهینه برنامه‌ریزی پویا را برای همین مسئله با موعد تحویل یکسان $D \sum_{j=1}^n d_j = D$ ارائه کردند که از $O(nD^2)$ می‌باشد.

حریری و یتس [۸] مسئله حداقل کردن تعداد کارهای دارای دیرکرد در فلوشاپ جایگشتی با M ماشین را با الگوریتم شاخه و کران حل کردند، در این مسئله کلیه کارها در زمان صفر در دسترس هستند. آنها مسائل با ابعاد حداکثر ۲۵ کار را برای دو و سه ماشین حل کردند. مصلحی و همکاران [۹]، مسئله حداقل کردن مجموع حداکثر دیرکردها و زودکردها^۳ بر روی یک فلوشاپ دو ماشین را بررسی کردند. آنها برای حل مسئله از روش شاخه و کران استفاده کردند و برای افزایش کارایی روش شاخه و کران از لم‌هایی برای یافتن جواب‌های غالب استفاده کردند.

وان و یین [۱۰]، مسئله حداقل کردن توام تعداد کارهای دارای دیرکرد و حداقل کردن مجموع زودکردهای وزن‌دار را در یک ماشین بررسی کردند. آنها برای حل مسئله از روش شاخه و کران استفاده کرده و برای حل مسائل بزرگ از یک الگوریتم ابتکاری با رتبه $O(n^3)$ بهره جستند.

باپتیست و همکاران [۱۱] یک الگوریتم شاخه و کران را برای مسئله حداقل سازی تعداد کارهای دارای دیرکرد در تک ماشین و با محدودیت زمان در دسترس بودن ارائه کردند. در این تحقیق، حد پایین از ساده سازی لاگرانژی به دست آمده و با ارائه یک الگوریتم شاخه و کران قادر به حل مسئله تا اندازه ۲۰۰ کار شدند.

⁴.Branch-and-Check

⁵. Multiple-choice

⁶.Forwards Greedy Algorithms

⁷.Deadline

⁸.Secondary Resource

¹.Makespan

².Moore

³.sum of maximum Earliness and Tardiness

مسئله با فرض ورود غیر همزمان کارها تاکنون به صورت بهینه در بیشتر از یک ماشین حل نشده است، در این مقاله این مسئله بر روی فلوشاپ دو ماشین و به صورت بهینه حل و الگوریتمی ابتکاری نیز برای حل مسائل با اندازه بزرگ ارائه می‌شود. این مسئله از این به بعد با نماد $P2|r_j|\sum U_j$ نمایش داده می‌شود. نمادهای مورد نیاز این مسئله در زیر بیان شده است:

N : مجموعه کارها، $N = \{1, 2, 3, \dots, n\}$ که n تعداد کل کارها است.

m : اندیس ماشین $m=1, 2$

T_j : زمان در دسترس کار j $j=1, 2, 3, \dots, n$

P_{mj} : زمان پردازش کار j روی ماشین m $j=1, 2, 3, \dots, n$ & $m=1, 2$

$P_{m[j]}$: زمان پردازش کار j در موقعیت m روی ماشین m $j=1, 2, 3, \dots, n$ & $m=1, 2$

d_j : موعد تحویل کار j $j=1, 2, 3, \dots, n$

C_{mj} : زمان تکمیل کار j روی ماشین m $j=1, 2, 3, \dots, n$ & $m=1, 2$

$C_{m[j]}$: زمان تکمیل کار j در موقعیت m روی ماشین m $j=1, 2, 3, \dots, n$ & $m=1, 2$

$$C_{2[j]} = \max(C_{1[j]} + P_{2[j]}, C_{2[j-1]} + P_{2[j]}) \quad (1)$$

$$T_{[j]}: \text{میزان دیرکرد}^2 \text{ کار } j \text{ در موقعیت } m \quad j=1, 2, 3, \dots, n$$

$$T_{[j]} = \max\{0, C_{2[j]} - d_{[j]}\} \quad (2)$$

$$U_j: \text{اگر } C_{2j} \geq d_j \text{ برابر با } 0 \text{ و در غیراین صورت برابر با } 1 \quad j=1, 2, 3, \dots, n$$

σ_L : توالی جزئی از کارهای چیده شده در گره L

σ'_L : مجموعه کارهای چیده نشده در گره L
 $J[i]$: کار چیده شده در موقعیت i ام توالی

$N_T(\sigma_L)$: تعداد کارهای دارای دیرکرد در توالی جزئی σ_L که از رابطه زیر به دست می‌آید

مرحله‌ای معرفی کردند. فلوشاپ آنها شامل دو مرحله بود که در مرحله اول M_1 ماشین موازی و یکسان و مرحله دوم M_2 ماشین موازی و یکسان وجود داشت.

موشیف و ساریگ [۲۱]، مسئله فلوشاپی را مطالعه کردند که در آن کارها نیاز به دو عملیات داشتند. عملیات اول بر روی تک ماشین مرحله اول (ماشین بحرانی) و عملیات دوم بر روی یکی از M ماشین مرحله دوم صورت می‌پذیرد. آنها برای حل مسئله مورد نظر یک الگوریتم برنامه‌ریزی پویای چند جمله‌ای کاذب و یک الگوریتم ابتکاری موثر ارائه کردند. هدف این مقاله حداقل کردن وزنی تعداد کارهای دارای دیرکرد بود.

همانطور که مشاهده شد، مسئله حداقل سازی تعداد کارهای دارای دیرکرد بر روی فلوشاپ جایگشتی با دو ماشین و زمان‌های ورود غیر همزمان کارها در ادبیات موضوع مشاهده نگردید. لذا در این مقاله این مسئله مورد بررسی قرار می‌گیرد و برای حل سریع آن یک روش ابتکاری و برای پیدا کردن جواب بهینه یک روش شاخه و کران ارائه می‌شود.

ادامه مقاله به این صورت سازماندهی شده است. در بخش دوم مسئله مورد نظر معرفی می‌شود، در بخش سوم رویکرد حل مسئله بیان می‌شود، در بخش چهارم نتایج محاسباتی ارائه می‌شود و در پایان در بخش پنجم نتیجه‌گیری و تحقیقات آینده ارائه می‌شوند.

۲. تعریف و پیچیدگی مسئله

در این بخش، مسئله حداقل سازی تعداد کارهای دارای دیرکرد بر روی فلوشاپ جایگشتی با دو ماشین و ورود غیر همزمان کارها به همراه پارامترها، متغیرها و فرضیات مسئله ارائه می‌شود. همانطور که در مقدمه اشاره شد، این مسئله با تاکید بر روی دو ماشین در ادبیات موضوع مشاهده نشده است، لذا در این مقاله مسئله مذکور مورد بررسی قرار می‌گیرد. مجموعه N شامل n کار مستقل $N = \{1, 2, 3, \dots, n\}$ می‌باشد. این کارها باید بر روی دو ماشین پردازش شوند.

تمام کارها ابتدا بر روی ماشین اول و سپس بر روی ماشین دوم پردازش می‌شوند. همچنین هرکاری که بر روی ماشین اول زودتر پردازش شود بر روی ماشین دوم هم زودتر پردازش می‌شود. به عبارت دیگر برنامه کاری دو ماشین مشابه هستند.

کارها دارای ورود غیر همزمان هستند و هر کار در زمان خاصی برای انجام پردازش در اختیار می‌باشد که به آن زمان در دسترس^۱ گفته می‌شود. هدف پیدا کردن یک توالی از کارها می‌باشد که در آن تعداد کارهایی که دارای دیرکرد هستند حداقل شوند. این

².Tardiness

¹.Release Time

$$\sum_{i=1}^n x_{i[j]} = 1 \quad j = 1, 2, 3, \dots, n \quad (5)$$

$$C_{1[j]} \geq \sum_{i=1}^n r_i x_{i[j]} + \sum_{i=1}^n p_{1i} x_{i[j]} \quad j = 1, 2, 3, \dots, n \quad (6)$$

$$C_{1[j]} \geq C_{1[j-1]} + \sum_{i=1}^n p_{2i} x_{i[j]} \quad j = 1, 2, 3, \dots, n \quad (7)$$

$$C_{2[j]} \geq C_{1[j]} + \sum_{i=1}^n p_{2i} x_{i[j]} \quad j = 1, 2, 3, \dots, n \quad (8)$$

$$C_{2[j]} \geq C_{2[j-1]} + \sum_{i=1}^n p_{2i} x_{i[j]} \quad j = 1, 2, 3, \dots, n \quad (9)$$

$$T_{[j]} \geq 0 \quad j = 1, 2, 3, \dots, n \quad (10)$$

$$T_{[j]} \geq C_{2[j]} - \sum_{i=1}^n d_i x_{i[j]} \quad j = 1, 2, 3, \dots, n \quad (11)$$

$$T_{[j]} \leq M \cdot U_{[j]} \quad j = 1, 2, 3, \dots, n \quad (12)$$

$$x_{i[j]} \in \{0, 1\} \quad i, j = 1, 2, 3, \dots, n \quad (13)$$

$$U_{[j]} \in \{0, 1\} \quad j = 1, 2, 3, \dots, n \quad (14)$$

رابطه (۶) تضمین می‌کند که کلیه کارها حتماً یک مرتبه در یکی از n موقعیت قرار بگیرند. رابطه (۷) نشان می‌دهد که هر موقعیت مانند j حتماً باید شامل یک کار باشد. محدودیت (۸) و (۹)، زمان تکمیل کارها را بر روی ماشین اول تعیین می‌کند. محدودیت (۱۰) و (۱۱) معین کننده‌ی زمان تکمیل کارها بر روی ماشین دوم می‌باشند. روابط (۱۲) تا (۱۴) حدود متغیر $T_{[j]}$ را تعیین می‌نمایند. محدودیت‌های (۱۵) تا (۱۶) شرایط متغیرهای مساله را نشان می‌دهند.

از آنجا که اثبات شد این مدل NP-hard است، لذا برای حل مدل از روش شاخه و کران استفاده شده است. در بخش بعدی رویکرد حل مسئله توضیح داده می‌شود.

۳. رویکرد حل

از آنجائی که پیچیدگی مسئله $P2|r_j|\Sigma U_j$ به صورت NP-hard است، لذا منطقی است که برای حل بهینه مسئله از روش شاخه و کران استفاده نمود. در روش شاخه و کران از حدود بالا و پایین

$$NT(\sigma_L) = \sum_{j \in \sigma_L} U_j \quad (1)$$

$N_T(\sigma'_L)$: تعداد دیرکرد برای کارهایی که در توالی جزئی σ_L قرار ندارند:

$$NT(\sigma'_L) = \sum_{j \notin \sigma_L} U_j \quad (2)$$

σ'_L -MOORE m : توالی جزئی حاصل از مجموعه σ'_L با استفاده از الگوریتم مور بر روی ماشین m ، $m=1, 2$.

در این مقاله فرض شده است که تمام کارها بدون انقطاع انجام می‌شوند. همچنین بیکاری عمدی برای ماشین مجاز می‌باشد، به عبارت دیگر می‌توان انجام کاری را که در دسترس است به صورت عمدی به تعویق انداخت و ماشین برای مدتی بیکار بماند تا کار دیگری برای پردازش برسد.

فرض بیکاری عمدی موقعی که کارها دارای زمان ورود غیر همزمان هستند می‌تواند باعث کاهش تعداد کارهای دارای دیرکرد شود هرچندکه حل مسئله را دشوارتر می‌کند. باپتیسست و همکاران نشان دادند که مسئله $1|r_j|\Sigma U_j$ مسئله NP-hard است [۱۱]. از آنجا که این مسئله تقلیل یافته مسئله $P2|r_j|\Sigma U_j$ می‌باشد، بنابراین آن هم حداقل دارای پیچیدگی NP-hard است.

۱-۲. مدل برنامه ریزی خطی

در این بخش مدل برنامه‌ریزی خطی عدد صحیح مختلط مسئله مورد نظر ارائه می‌شود. در ابتدا متغیرها و پارامتر مورد استفاده در مدل شرح داده می‌شوند. سپس مدل طراحی شده ارائه و تابع هدف و محدودیت‌های مدل تشریح می‌شوند.

$x_{i[j]}$ اگر کار i در موقعیت j ام پردازش شود
 $i, j = 1, 2, 3, \dots, n$
 یک و در غیر این صورت صفر است.

M : یک عدد بزرگ

۱-۲-۱. مدل برنامه‌ریزی عدد صحیح مختلط

$$\text{Minimize } NT = \sum_{j=1}^n U_{[j]} \quad (3)$$

s.t.

$$\sum_{j=1}^n x_{i[j]} = 1 \quad i = 1, 2, 3, \dots, n \quad (4)$$

نسبت به دو الگوریتم دیگر کارایی بهتری داشته، از آن به عنوان الگوریتم برتر و همچنین به عنوان حد بالا استفاده شده است.

۳-۲. الگوریتم شاخه و کران

برای به دست آوردن جواب بهینه مسئله $P2|rj|\Sigma U$ از روش شاخه و کران استفاده شده است. در این الگوریتم هر گره از درخت جستجو یک توالی جزئی σ_L را نشان می‌دهد. علاوه بر این هر گره شامل تعداد کارهای زمان‌بندی شده، تابع هدف توالی جزئی چیده شده $NT(\sigma_L)$ ، یک تخمین برای تابع هدف کارهای چیده نشده (تخمینی از NT کارهای چیده نشده) و یک حد پایین به صورت جمع تابع هدف تخمینی برای کارهای چیده نشده و تابع هدف توالی جزئی چیده شده می‌باشد.

برای شاخه زدن از روش جستجوی عمقی^۳ استفاده شده و در آن از سه حد پایین و از الگوریتم H_1 به عنوان حد بالا استفاده شده است. همچنین چندین اصل غلبه برای افزایش کارایی روش ارائه شده است.

در سطح k ام الگوریتم با توالی جزئی کارهای $J[k] = \{j_1, \dots, j_k\}$ ، شاخه‌های جدید، از طریق اضافه شدن کارهای زمان‌بندی نشده به آخر توالی σ_L به وجود می‌آیند که این کارها می‌بایستی شرایط زمان ورود و اصول غلبه را ارضا کرده باشند. هنگامی که هیچ کاری برای برنامه‌ریزی وجود نداشته باشد، توالی σ_L یک زمان‌بندی کامل خواهد بود و گره مد نظر به آخرین سطح خود رسیده است. در طی فرایند جستجو میزان تابع هدف برای بهترین زمان‌بندی یافت شده تاکنون با حد پایین مقایسه و به هنگام می‌شود.

لم ۱: یک حد پایین برای مسئله $P2|rj|\Sigma U$ به صورت زیر است:

$$LB_1 = NT(\sigma_L) + NT(\sigma'_{L-MOORE2}) \quad (15)$$

که در آن $NT(\sigma_L)$ تعداد کارهای دارای دیرکرد در توالی جزئی σ_L و $NT(\sigma'_{L-MOORE2})$ تعداد کارهای دارای دیرکرد در توالی $\sigma'_{L-MOORE2}$ می‌باشد. که این توالی بر اساس الگوریتم مور و از روی ماشین دوم به دست می‌آید.

اثبات: در گره‌ای مثل L ، کارهای چیده شده با σ_L و کارهای چیده نشده با σ'_L نشان داده می‌شوند. حد پایین LB_1 شامل دو قسمت می‌باشد.

قسمت اول تعداد کارهای دارای دیرکرد از توالی جزئی چیده شده σ_L می‌باشد و با $NT(\sigma_L)$ نشان داده می‌شود. قسمت دوم تخمینی

مناسب و اصول غلبه کارایی استفاده شده، تا بتوان با قطع^۱ کردن سریع شاخه‌ها در زمان مناسب به جواب بهینه دست یافت. در ادامه یک الگوریتم ابتکاری معرفی می‌شود که به علت عملکرد خوب آن، به عنوان حد بالا در روش شاخه و کران نیز استفاده شده است.

۳-۱. الگوریتم ابتکاری H_1

اجرای الگوریتم مور- هاجسون برای مسئله $1||\Sigma U$ منجر به جواب بهینه می‌شود [۵]. لذا با الهام از این الگوریتم، برای مسئله $P2|rj|\Sigma U$ یک الگوریتم ابتکاری به نام H_1 ارائه شده است. قدم-های الگوریتم H_1 به صورت زیر است:

(۱) ابتدا کارها به صورت زودترین موعد تحویل (EDD^*) مرتب شوند.

(۲) سپس کارها براساس ترتیب EDD و با توجه به زمان در دسترس هر کار (T_j) ، به ماشین اول و دوم تخصیص داده می‌شوند و زمان تحویل هر کار محاسبه شود.

(۳) اگر اولین کار دیرکردار در موقعیت i باشد، از ابتدای ترتیب تا موقعیت i هر دفعه یکی از کارها حذف شده و میزان صرفه جویی در زمان ختم کار در موقعیت i به دست آورده شود. کاری که حذف آن منجر به بیشترین کاهش در زمان ختم کار در موقعیت i بر روی ماشین دوم می‌شود در آخر توالی قرار گرفته و این کار دارای دیرکردار است. هر موقع کاری به آخر توالی برده می‌شود از تعداد کل کارها یکی کم می‌شود. (یعنی اولین کار حذف شده در آخر توالی قرار می‌گیرد و دومین کار حذف شده در مرحله بعدی در موقعیت $n-1$ و به همین ترتیب محل کارهای دارای دیرکرد بعدی مشخص می‌شود.)

(۴) برای بقیه کارها مجدداً قدم های ۲ و ۳، مادامی که کار دیرکردار یافت شود اجرا می‌شوند.

نتایج عددی نشان می‌دهد که این الگوریتم کارایی خوبی دارد. دو الگوریتم ابتکاری دیگر به نام‌های H_2 و H_3 نیز مورد بررسی قرار گرفته‌اند. در H_2 ، به جای کار حذف شده در قدم سوم الگوریتم H_1 ، کار با بیشترین زمان پردازش روی ماشین اول و دوم حذف می‌شود.

به عبارت دیگر کاری با بیشترین $P_{1j} + P_{2j}$ حذف می‌شود. در الگوریتم H_3 کار با بزرگترین زمان پردازش روی ماشین دوم حذف شده و به آخر توالی برده می‌شود. از آنجا که الگوریتم H_1

^۱fatom

^۲.Earliest Due Date

^۳. Back-Tracking

اثبات: مشابه لم ۱. □

لم ۳: برای به دست آوردن یک حد پایین در مسئله $P2|r_j|\Sigma U_j$ می توان به صورت زیر عمل کرد:

تعداد کارهای دارای دیرکرد در توالی جزئی چیده شده σ_L همانند لم‌های قبل محاسبه شود. برای تخمین تعداد کارهای دارای دیرکرد در مجموعه σ'_L به صورت زیر عمل شود.

$$LB_3 = NT(\sigma_L) + NT(\sigma'_L) \quad (18)$$

$$NT(\sigma'_L) = \sum_{i \in \sigma'_L} \delta(T'_{[i]}) \quad (19)$$

که $\delta(T'_{[i]})$ برابر است با،

$$\delta(T'_{[i]}) = \begin{cases} 0 & \text{if } T'_{[i]} = 0 \\ 1 & \text{if } T'_{[i]} > 0 \end{cases} \quad (20)$$

اگر کارهای مجموعه σ'_L یکبار بر اساس زمان‌های پردازش ماشین اول و یکبار هم بر اساس ماشین دوم و به ترتیب از زمان شروع A و B چیده شوند برای هر موقعیت دو زمان تکمیل به دست می‌آید که حداکثر بین هر دو، به عنوان زمان تکمیل کار در موقعیت مربوطه در نظر گرفته می‌شود و به صورت $C_{[i]}$ نمایش داده می‌شود. از طرف دیگر اگر کارهای مجموعه σ'_L به ترتیب EDD چیده شوند و موعد تحویل به دست آمده به صورت $d_{[i]}$ نمایش داده شود می‌توان دیرکرد هر موقعیت را به صورت زیر محاسبه نمود،

$$T'_{[i]} = \text{MAX} \{0, C_{[i]} - \text{Min}_{j \geq i} (d_{[j]})\} \quad (21)$$

هرگاه برای یک موقعیت $\delta(T'_{[i]}) = 1$ شود، به تعداد کارهای دارای دیرکرد تخمینی یکی اضافه می‌شود و زمان ختم آن موقعیت با زمان تحویل بعدی که بزرگتر می‌باشد مقایسه می‌شود و این عمل برای تمام مواعدهای تحویل ادامه می‌یابد. □ لازم به یادآوری است که حد پایین مسئله، یعنی LB با توجه به سه حد پایین گفته شده بر اساس رابطه (۲۴) به دست می‌آید.

$$LB = \text{MAX} \{LB_1, LB_2, LB_3\} \quad (22)$$

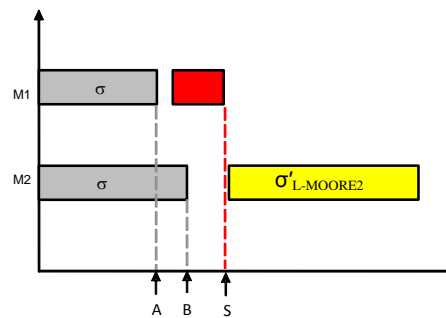
لم ۴: اگر در گره L کار i از مجموعه کارهای چیده نشده برای چیدن در انتهای σ_L انتخاب شود و کار i در این موقعیت دارای دیرکرد شود، جواب بهینه‌ای وجود دارد که در آن کار i در آخر توالی قرار می‌گیرد.

از تعداد کارهای دارای دیرکرد از کارهای چیده نشده σ'_L است که با $NT(\sigma'_{L-MOORE2})$ نشان داده می‌شود.

این تخمین با استفاده از روش مور-هاجسون در تک ماشین به دست می‌آید و برای این منظور فقط از زمان پردازش کارها روی ماشین دوم یعنی P_{2i} استفاده می‌شود. فرض کنید زمان در دسترس بودن برای شروع مجموعه σ'_L روی ماشین اول A و روی ماشین دوم B باشد. زودترین زمان شروع کارهای مجموعه σ'_L روی ماشین دوم برابر با S است که از رابطه (۱۸) به دست می‌آید.

$$S = \text{Max} \{ \text{Min}_{i \in \sigma'_L} \{ \text{Max} \{ A, r_i \} + P_{1i} \}, B \} \quad (16)$$

از آنجا که در این روش فقط ماشین دوم در نظر گرفته می‌شود و اثر ماشین اول در ایجاد بیکاری روی ماشین دوم نادیده گرفته شده است، بنابراین تعداد کارهای دارای دیرکرد به دست آمده از روش مور-هاجسون یک حد پایین برای تعداد کارهای دارای دیرکرد مجموعه σ'_L است. با جمع این مقدار از تعداد کارهای دارای دیرکرد با تعداد کارهای دارای دیرکرد مجموعه σ_L ، یک حد پایین برای مسئله به دست می‌آید که با LB_1 نمایش داده می‌شود. □ شکل ۱ نمایی از این حد پایین را نشان می‌دهد.

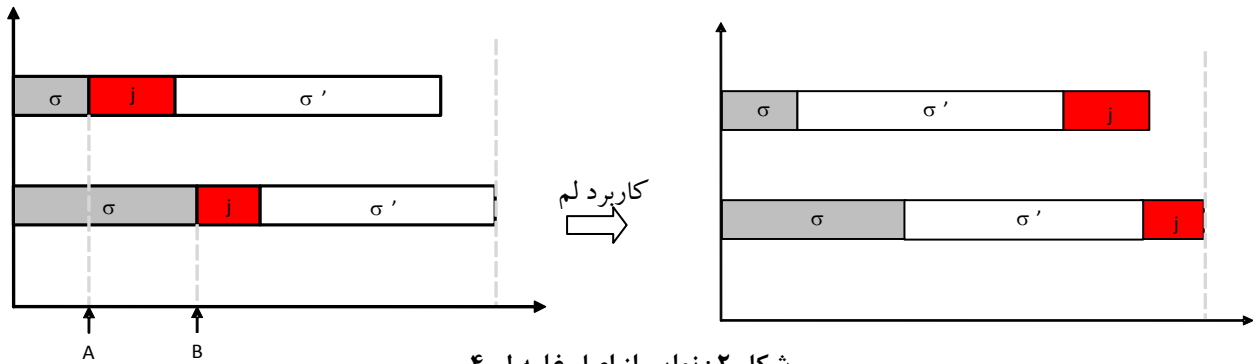


شکل ۱: نمایی از حد پایین لم ۱

لم ۲: یک حد پایین برای مسئله $P2|r_j|\Sigma U_j$ به صورت زیر است:

$$LB_2 = NT(\sigma_L) + NT(\sigma'_{L-MOORE1}) \quad (17)$$

$NT(\sigma_L)$ مشابه لم ۱ است که از روی ماشین اول به دست می‌آید. کارهای مجموعه σ'_L بر اساس الگوریتم مور-هاجسون از زمان شروع A چیده می‌شوند. همچنین برای به دست آوردن آخرین کار چیده شده در مجموعه σ'_L مقدار کوچکترین مدت زمان پردازش روی ماشین دوم از مجموعه σ'_L به آن اضافه می‌شود.



شکل ۲: نمایی از اصل غلبه لم ۴

$$C_1(\sigma_{Lji}) \geq C_1(\sigma_{Lij}) \quad (28)$$

$$C_2(\sigma_{Lji}) > C_2(\sigma_{Lij}) \quad (29)$$

اثبات: با توجه به شرایط ذکر شده، زمان تکمیل توالی جزئی σ_{Lij} بر روی هر دو ماشین بدتر از زمان تکمیل توالی جزئی σ_{Lji} نمی‌باشد. بنابراین توالی σ_{Lij} سبب به تعویق افتادن شروع کارهای بعدی نمی‌شود و با توجه به این که تابع هدف حداقل-کردن تعداد کارهای دارای دیرکرد یک تابع منظم است، توالی σ_{Lij} موجب افزایش تعداد کارهای دارای دیرکرد نمی‌شود. □

۴. نتایج محاسباتی

الگوریتم شاخه و کران و الگوریتم ابتکاری در زبان برنامه نویسی Visual Basic 6 کدنویسی شده‌اند و بر روی یک کامپیوتر با 2 GB of RAM اجرا شده‌اند. برای بررسی کارایی روش ارائه شده مسائلی به صورت تصادفی و با استفاده از روش اشاره شده در مرجع [۱۶] تولید شده‌اند.

زمان‌های پردازش از توزیع یکنواخت گسسته در فاصله [۱۷۵ و ۱] تولید شده‌اند. مقادیر موعد تحویل از توزیع یکنواخت گسسته در بازه (α, β) به دست می‌آید که مقدار P از رابطه (۳۲) زیر حاصل می‌شود.

$$P = (1/n) * [\sum_{i=1}^n \sum_{j=1}^2 P_{ij}] + (n-1) \max\{\sum_{i=1}^n P_{i1}, \sum_{i=1}^n P_{i2}\} \quad (30)$$

مقادیر α و β که به ترتیب حد پایین و حد بالای مواعدهای تحویل را نشان می‌دهد از دو مجموعه $\alpha \in \{0.4, 0.6\}$ و $\beta \in \{0.8, 1\}$ به دست می‌آیند.

زمان‌های در دسترس بودن یعنی Γ_i از توزیع یکنواخت گسسته $U(P\delta, P\epsilon)$ تولید شده که برای δ مقدار صفر و برای ϵ مقادیر $\{0.05, 0.25\}$ در نظر گرفته شده‌اند. برای داده‌ها دو گروه Low و High در نظر گرفته شده‌است. پارامترهای Low به

اثبات: واضح است که با انتقال کار دیرکردار به آخر، زمان ختم بقیه کارها افزایش نمی‌یابد و حتی ممکن است کاهش هم پیدا کند. بنابراین با توجه به این که تابع هدف حداقل‌کردن تعداد کارهای دارای دیرکرد یک تابع منظم است، موجب افزایش تعداد کارهای دارای دیرکرد نمی‌شود. □

لم ۵: با برقراری شروط ۲۵ تا ۲۷ حتماً یک جواب بهینه وجود دارد که کار j باید بعد از کار i پردازش شود.

$$R_i = \min_{k \in \sigma'} \{r_k\} \quad (23)$$

$$r_i \geq A, r_j \geq r_i + P_{Ii} \quad (24)$$

$$r_i + P_{Ii} + P_{2i} \leq r_j + P_{Ij} \quad (25)$$

اثبات: هنگامی که کارهای مجموعه σ_L چیده شده باشند و دو کار دارای کوچکترین زمان ورود بوده و روابط (۲۵) تا (۲۷) را نیز برآورده کنند مسلماً کار i باید قبل از j انجام شود چون انجام و یا عدم انجام کار i در زمان شروع کار j تاثیری ندارد و بنابراین برای کاهش تعداد کارهای دارای دیرکرد، کار i حتماً باید قبل از کار j انجام شود. برای این منظور زمان‌های تکمیل دو توالی جزئی σ_{Lij} و σ_{Lji} بعد از σ_L به صورت روابط (۲۸) و (۲۹) محاسبه می‌شود.

$$C_2(\sigma_{Lij}) = r_j + P_{Ij} + P_{2j} \quad (26)$$

$$C_2(\sigma_{Lji}) = r_j + P_{Ij} + \max(P_{Ii}, P_{2j}) + P_{2i} \quad (27)$$

به سادگی دیده می‌شود که $C_2(\sigma_{Lji}) > C_2(\sigma_{Lij})$ ، بنابراین کار i باید قبل از j پردازش شود. □

لم ۶: اگر دو کار i, j در دو توالی σ_{Lij} و σ_{Lji} دیرکرد نداشته باشند و روابط (۳۰) و (۳۱) برقرار باشد، آنگاه جواب بهینه‌ای وجود دارد که در آن کار j بعد از i پردازش می‌شود.

صورت $(\alpha=0.6, \beta=0.8, \varepsilon=0.05)$ و High به صورت $(\alpha=0.4, \beta=1, \varepsilon=0.25)$ می‌باشد. مسائل گروه High دارای شرایط بسیار پویاتری نسبت به مسائل گروه Low می‌باشند. به عبارت دیگر مسائل گروه Low مسائل سخت‌تری نسبت به مسائل گروه High می‌باشند. جدول ۱ تفاوت‌های بین دو گروه مسئله را نشان می‌دهد.

جدول ۱. تفاوت‌های بین دو گروه مسئله تولید شده [۱۶]

پارامتر تاثیرگذار	گروه High	گروه Low
r_j	کارها در یک بازه زمانی بزرگ که طولانی‌تر از زمان شروع برنامه‌ریزی شده می‌باشد در دسترس قرار می‌گیرد.	کارها در یک بازه زمانی کوچک و نزدیک به زمان شروع برنامه‌ریزی شده در دسترس قرار می‌گیرند.
d_j	بازه زمان‌های تحویل باز	بازه زمان‌های تحویل فشرده
d_j	زمان‌های تحویل متفاوت	زمان‌های تحویل مشابه
تعداد کارهای دارای دیرکرد	تعداد کارهای دارای دیرکرد بیشتر	تعداد کارهای دارای دیرکرد کمتر

شده‌اند. بولفین و هلاه [۱۵] جهت بررسی عملکرد الگوریتم ابتکاری از شاخص Z^{opt} / Z^{heu} استفاده نمودند که در آن Z^{opt} و Z^{heu} به ترتیب جواب بهینه و جواب الگوریتم ابتکاری با هدف $\sum (1-U_i)$ می‌باشند. با توجه به این که حداقل مقدار این شاخص در تمام دسته‌ها برابر با ۱ می‌باشد بنابراین در ستون "انحراف جواب الگوریتم ابتکاری از جواب بهینه" مقادیر متوسط و حداکثر شاخص ارائه شده است.

همچنین برای بررسی کارایی روش‌های ارائه شده مسائلی با تعداد کار ۸ تا ۳۴ و از هر مسئله ۲۰ نمونه تولید شده‌اند و در نهایت در این تحقیق ۳۴۰ نمونه مسئله حل و برای هر مسئله محدودیت زمانی برابر ۳۶۰۰ ثانیه در نظر گرفته شده‌است و در این مدت زمان اگر جواب بهینه توسط روش شاخه و کران به دست نیاید روند اجرای الگوریتم شاخه و کران قطع می‌شود. جداول ۲ و ۳ نتایج هر دو گروه را نشان می‌دهد. در جدول ۲ مسائل در گروه High و در جدول ۳ مسائل برای گروه Low حل

جدول ۲. نتایج برای الگوریتم شاخه و کران (B&B) و ابتکاری - گروه High

انحراف جواب الگوریتم ابتکاری از جواب بهینه	تعداد مسائل حل نشده توسط B&B	تعداد گره‌های طی شده			زمان‌های اجرا (ثانیه)			تعداد کارها	
		حداکثر	متوسط	حداقل	حداکثر	متوسط	حداقل		
حداکثر	متوسط	حداکثر	متوسط	حداقل	حداکثر	متوسط	حداقل		
۱,۳۳	۱,۱	۰	۱۰۰	۴۰	۱۰	۰,۰۰	۰,۰۰	۰,۰۰	۸
۱,۲۵	۱,۰۶	۰	۶۶۶۱	۵۲۲	۱۵	۰,۰۹	۰,۰۱	۰,۰۰	۱۲
۱,۱۸	۱,۰۹	۰	۸۶۱۹۹	۵۷۵۵	۴۳	۱,۶۴	۰,۱۲	۰,۰۰	۱۶
۱,۱۳	۱,۰۸	۰	۵۸۲۵۵۸	۳۶۲۳۸	۲۱	۱۳,۴۸	۰,۸۵	۰,۰۰	۲۰
۱,۲۲	۱,۰۹	۰	۶۴۳۴۲۷۸	۳۲۶۱۹۹	۴۹	۱۸۶,۲۵	۹,۴۵	۰,۰۰	۲۴
۱,۲۸	۱,۱۱	۰	۷۹۰۵۲۹۶۹	۱۱۴۰۴۰۲۹	۵۴۷	۲۶۰۰,۵۲	۲۵۰,۳۱	۰,۰۳	۲۶
۱,۱۸	۱,۰۸	۰	۸۷۸۲۵۹۴۱	۱۱۷۶۱۰۱۹	۴۶۶	۲۵۰۷,۰۲	۳۲۹,۰۸	۰,۰۱	۲۸
۱,۲۲	۱,۱	۱	۱۰۳۵۴۲۲۹۰	۶۳۱۸۹۲۰	۱۲۶۶	۳۱۸۲,۰۶	۱۹۵,۱۹	۰,۰۳	۳۰
۱,۱۲	۱,۰۸	۱	۳۳۴۰۶۹۳۴	۴۶۲۱۴۲۱	۹۲۱	۹۳۲,۰۲	۱۵۲,۰۱	۰,۰۵	۳۲
۱,۱۸	۱,۰۷	۷	۵۴۴۸۶۶۵۷	۱۵۶۰۴۴۴۱	۷۴۲	۲۳۰۹,۳۷	۵۸۹,۴۷	۰,۰۳	۳۴

جدول ۳. نتایج برای الگوریتم شاخه و کران (B&B) و ابتکاری - گروه Low

انحراف جواب الگوریتم ابتکاری از جواب بهینه	*تعداد مسائل حل نشده توسط B&B	تعداد گره های طی شده			زمان های اجرا (ثانیه)			تعداد کارها
		حداکثر	متوسط	حداقل	حداکثر	متوسط	حداقل	
حداکثر	متوسط							
۱,۴۰	۱,۰۸	۰	۵۱۶	۱۶۲	۱۱	۰,۰۲	۰	۸
۱,۱۳	۱,۰۷	۰	۵۹۱۰	۱۴۰۸	۱۳	۰,۱۱	۰,۰۳	۱۲
۱,۱۸	۱,۰۶	۰	۴۲۰۸۱۳۱	۳۹۳۹۲۸	۳۱	۹۳,۱۹	۸,۳۹	۱۶
۱,۱۳	۱,۰۴	۰	۱۰۶۴۴۵۹۱۱	۱۰۶۷۳۹۶۵	۲۵	۳۴۵۰,۹۳	۳۰۶,۲۳	۲۰
۱,۱۱	۱,۰۷	۷	۴۷۱۰۰۸۸۵	۱۲۰۷۰۳۹۲	۳۶	۱۴۰۰,۴۷	۳۷۰,۳۶	۲۴
۱,۲۲	۱,۰۸	۱۳	۲۷۲۴۱۳۳۷	۶۱۹۹۶۱۷	۴۲	۷۹۶,۱۶	۱۷۶,۶۳	۲۶
۱,۱۴	۱,۰۵	۱۴	۱۴۶۳۲۲۷۵	۲۷۸۴۹۳۸	۴۰	۴۵۱,۰۹	۸۵,۳۱	۲۸

جدول شماره ۴. نتایج برای الگوریتم ابتکاری -

مسائل گروه High

تعداد کارها	زمان های اجرا (ثانیه)		
	حداقل	متوسط	حداکثر
۲۰۰	۰,۰۸	۰,۱۳	۰,۱۷
۴۰۰	۰,۷۰	۱,۰۵	۱,۳۳
۶۰۰	۲,۳۱	۳,۰۳	۳,۶۴
۸۰۰	۵,۵۳	۷,۱۰	۸,۶۴
۱۰۰۰	۱۱,۱۶	۱۳,۹۸	۱۶,۳۷

جدول شماره ۵. نتایج برای الگوریتم ابتکاری -

مسائل گروه Low

تعداد کارها	زمان های اجرا (ثانیه)		
	حداقل	متوسط	حداکثر
۲۰۰	۰,۲۲	۰,۲۵	۰,۳۰
۴۰۰	۱,۶۴	۱,۸۵	۲,۰۶
۶۰۰	۵,۶۰	۶,۱۳	۶,۵۵
۸۰۰	۱۲,۹۷	۱۴,۴۶	۱۹,۴۲
۱۰۰۰	۲۵,۵۵	۳۲,۰۸	۴۳,۴۶

۵. نتیجه گیری و پیشنهادات

در این مقاله مسئله $P2/r_j|\Sigma U_j$ یعنی زمان بندی فلو شاپ جایگشتی دو ماشین با زمان های ورود غیر همزمان و هدف حداقل کردن تعداد کارهای دارای دیرکرد مورد بررسی قرار گرفت. در ابتدا نشان داده شد که این مسئله NP-hard است. از این رو یک الگوریتم شاخه و کران و یک الگوریتم ابتکاری برای حل آن ارائه شد.

در جداول ۲ و ۳ متوسط شاخص Z^{opt} / Z^{heu} حداکثر ۱/۱۱ برابر می باشد و در مقایسه با تحقیقات دیگر مربوط به تعداد کارهای دیرکردار [۲۲]، عدد کوچکی است که نشان دهنده ی کارایی بالای الگوریتم ابتکاری است.

نتایج نشان می دهد که الگوریتم شاخه و کران پیشنهادی قادر است مسائلی را تا اندازه ۲۸ کار در گروه High و ۲۰ کار در گروه Low را بطور کامل در زمان منطقی حل نماید و نیز الگوریتم ابتکاری در اندازه های کوچک قادر است تا اکثر مسائل را به طور بهینه حل نماید و در اندازه های متوسط با تقریب مناسبی می تواند مسائل را حل نماید.

علاوه بر این، نتایج نشان می دهد که با بزرگ شدن بازه r_j و d_j زمان حل و تعداد گره های حل شده در روش شاخه و کران کاهش می یابد. از آنجا که مسئله حداقل کردن تعداد کارهای دارای دیرکرد بر روی فلو شاپ دو ماشین تا کنون با هیچ فرضی با روش های بهینه حل نشده است برای مقایسه نتایج می توان به کار حریری و پتس اشاره کرد که آنها همین مسئله را بر روی یک فلو شاپ M ماشین و بدون فرض دسترسی حل کردند. آنها توانستند برای فلو شاپ با ۲ ماشین تعدادی از مسائل با ۲۵ کار و برای فلو شاپ با ۳ ماشین تعداد اندکی از مسائل با ۲۵ کار را حل کنند.

در جداول ۴ و ۵ نتایج اجرای الگوریتم ابتکاری برای مسائل با اندازه بزرگ ارائه شده است. در هر مسئله تعداد ۱۰۰ نمونه اجرا شده است.

نتایج نشان می دهد که الگوریتم ابتکاری پیشنهادی قادر به حل مسائل بزرگ در زمانی کمتر از ۱۴ ثانیه برای مسائل گروه High و ۳۲ ثانیه برای مسائل گروه Low می باشد.

- [8] Hariri, A.M.A., Potts, C.N., "A Branch and Bound Algorithm to Minimize Number of Late Jobs in a Permutation Flow Shop", *European Journal of Operational Research*, Vol.38, 1989, pp.228-237.
- [9] Moslehi, G., Mirzaee, M., Vasei, M., Modarres, M., Azaron, A., "Two-Machine Flow Shop Scheduling to Minimize the Sum of Maximum Earliness and Tardiness", *Int. J. Production Economics*, Vol.122, 2009, pp.763-773.
- [10] Wan, G., Yen, B.P.C., "Single Machine Scheduling to Minimize Total Weighted Earliness Subject to Minimal Number of Tardy Jobs", *European Journal of Operational Research*, Vol.195, 2009, pp. 89-97.
- [11] Baptiste, P., Peridy, L., Pinson, E., "A Branch and Bound to Minimize the Number of Late Jobs on a Single Machine with Release Time Constraints", *European Journal of Operational Research*, Vol.144, 2003, pp.1-11.
- [12] Sadykov, R., "A Branch-and-Check Algorithm for minimizing the Weighted Number of Late Jobs on a Single Machine with Release Dates", *European Journal of Operational Research*, Vol. 189, 2008, pp.1284-1304.
- [13] Hallah, R., Bulfin, R.L., "Minimizing the Weighted Number of Tardy Jobs on a Single Machine with Release Dates", *European Journal of Operational Research*, Vol. 176, 2007, pp.727-744.
- [14] Hallah, R., Bulfin, R.L., "Minimizing the Weighted Number of Tardy Jobs on Parallel Processors", *European Journal of Operational Research*, Vol. 160, 2005, pp.471-484.
- [15] He, C., Lin, Y., Yuan, J., "A Note on the Single Machine Scheduling to Minimize the Number of Tardy Jobs with Deadlines", *European Journal of Operational Research*, Vol.201, 2010, pp.966-970.
- [16] Lodree, E., Jang, W., Klein, C.M., "A New Rule for Minimizing the Number of Tardy Jobs in Dynamic flow Shops", *European Journal of Operational Research*, Vol.159, 2004, pp.258-263.
- [17] Ruiz-Torres, A.J., Centeno, G., "Minimizing the Number of Late Jobs for the Permutation Flowshop Problem with Secondary Resources", *Computers & Operations Research*, Vol.35, 2008, pp.1227-1249.
- [18] Ruiz-Torres, A.J., Ablanedo-Rosas, J.H., Ho, J.C., "Minimizing the Number of Tardy Jobs in the flowshop Problem with Operation and Resource Flexibility", *Computers & Operations Research*, Vol. 37, 2010, pp.282-291.
- [19] Choi, S.W., Kim, Y.D., "Minimizing Total Tardiness on a Two Machine Re-Entrant Flowshop" *European*

در روش شاخه و کران سه حد پایین و سه اصل غلبه ارائه گردید. کارایی و عملکرد الگوریتم با حل تعداد ۳۴۰ مسئله نمونه مورد بررسی قرار گرفت. نتایج محاسباتی نشان داد که الگوریتم شاخه و کران مسائل با ابعاد ۲۸ فعالیت در مسائل گروه High و ۲۰ فعالیت در مسائل گروه Low را بطور کامل حل نموده است که این امر کارایی رویه ارائه شده را نشان می‌دهد. علاوه بر آن نشان داده شد که متوسط نسبت جواب بهینه به ابتکاری با هدف $\sum (1-U_i)$ حداکثر ۱/۱۱ برابر می‌باشد که بیان کننده عملکرد بالای الگوریتم ابتکاری است. از آنجا که الگوریتم ابتکاری معرفی شده از قابلیت بالایی برخوردار است و از این الگوریتم به عنوان حد بالا در روش شاخه و کران استفاده شده است، لذا پیشنهاد می‌شود برای بهبود روش شاخه و کران حدود پایین و اصول غلبه جدیدی ارائه شود. در نظر گرفتن فرض زوال پذیری کارها در مسئله مورد بررسی از مسائل جالب برای ادامه تحقیقات به حساب می‌آید. از آنجا که مسئله مورد بررسی تا کنون با هیچ روشی حل نشده است پیشنهاد می‌شود از الگوریتم‌های فراابتکاری برای حل مسئله در ابعاد بزرگ استفاده شود.

مراجع

- [1] Bulfin, R.L., M'Hallah, R., "Minimizing the Weighted Number of Tardy Jobs on a Two-Machine Flow Shop", *Computers & Operations Research*, Vol.30, 2003, pp.1887-1900.
- [2] Lageweg, B.J., Lenstra, J.K., Rinnooy Kan, A., "A General Bounding Scheme for the Permutation Flowshop Problem", *Operations Research*, Vol.26, 1978, pp.53-67.
- [3] Ignall, E., Schrage, L.E., "Applications of the Branch and Bound Technique to Some Flow Shop Problems", *Operations Research*, Vol.13, 1965, pp.400-412.
- [4] Lenstra, J.K., Rinnooy Kan, A., Brucker, P., "Complexity of Machine Scheduling Problems", *Annals of Discrete Mathematics*, Vol.1, 1968, pp.343-362.
- [5] Moore, J.M., "An n Job, One Machine Algorithm for Minimizing the Number of Late Jobs", *Management Science*, Vol.15, 1968, pp 102-109.
- [6] Lin, B., "Scheduling in the Two-Machine Flow Shop with Due Date Constraints", *International Journal of Production Economics*, Vol.70, 2001, pp.117-123.
- [7] Lawler, E.L., Moore J.M. "A Functional Equation and its Application to Resource Allocation and Sequencing Problems", *Management Science*, Vol. 15, 1968, pp.102-109

Journal of Operational Research, Vol. 199, 2009, pp.375–384.

- [20] Choi, H.S., Lee, D.H., “Scheduling Algorithms to Minimize the Number of Tardy Jobs in Two-Stage Hybrid Flow Shops”, Computers & Industrial Engineering, Vol.56, 2009, pp.113–120.
- [21] Mosheiov, G., Sarig, A., “Minimum Weighted Number of Tardy Jobs on an M-Machine Flow-shop with a Critical Machine”, European Journal of Operational Research, Vol.201, 2010, pp.404–408.
- [22] Moslehi, G., Jafari, A., “Minimizing the Number of Tardy Jobs Under Piecewise-Linear Deterioration” Computers & Industrial Engineering, Vol.59, 2010, pp. 573-584.