International Journal of Industrial Engineering & Production Management (2013)

February 2013, Volume 23, Number 4
pp. 389-400

http://IJIEPM.iust.ac.ir/

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.*

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

نشـــریه بین المللـــی مهنـــدسی صنـــایع و مدیـــریت تولیـــد

شمـــاره ۴، جلـــد ۲۳، اسفند ۱۳۹۱
صفحـــــه ۴۰۰-۳۹۰

نشریه بین المللی مهندسی صنایع و مدیریت تولید
دانشگاه علم و صنعت ایران

ISSN: 2008-4870

http://IJIEPM.iust.ac.ir/

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

قاسم مصلحی*، علی حکیمیان و مصطفی ابویی اردکان

کلمات کلیدی

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

چکیده:

در این مقاله مسئله زمان‌بندی فلوشاپ دو ماشین با در نظر گرفتن ورود غیر همزمان و با هدف کمینه‌سازی تعداد کارهای دارای دیرکرد بررسی شده است. در ابتدا پیچیدگی مسأله بررسی و ثابت شده که مسأله 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 می‌دهد که رویه شاخه‌وکران مسائل با ابعاد ۲۸ فعالیت در گروه High و ۲۰ فعالیت در گروه Low را در زمان منطقی و به طور کامل حل می‌کند، که این امر کارآیی حد بالا، حدود پایین و اصول غلبه ارائه شده برای مسئله را نشان می‌دهد. همچنین نشان داده شد که متوسط نسبت جواب بهینه به الگوریتم ابتکاری با هدف $\sum(1\text{-}U_i)$ حداکثر ۱/۱۱ برابر می‌باشد که در مقایسه با الگوریتم‌های ارائه شده در تحقیقات مرتبط با کارهای دارای دیرکرد نسبت کوچکی می‌باشد. این نسبت نشان دهنده کارایی بالای الگوریتم ابتکاری است. با توجه به کارآیی بالای الگوریتم ابتکاری، مسائل نمونه با ابعاد بزرگ نیز حل و نتایج آن ارائه شده است.

۱. مقدمه

در بسیاری از صنایع برآورد کردن زمان‌های تحویل امری حیاتی می‌باشد. لذا معیارهای وابسته به تاخیر همانند تعداد کارهای دارای دیرکرد از اهمیت ویژه‌ای برخوردار هستند و به عنوان معیارهای ارزیابی زمان‌بندی‌ها مورد استفاده قرار می‌گیرند [۱].

در این مقاله مسئله زمان‌بندی فلوشاپ دو ماشین با زمان‌های ورود غیر همزمان و با هدف حداقل کردن تعداد کارهای دارای دیرکرد در حالت فلوشاپ جایگشتی[۲] مورد بررسی قرار می‌گیرد. در این حالت هر کاری که روی ماشین اول زودتر پردازش شود، روی ماشین دوم نیز زودتر پردازش می‌شود. 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 می‌توان در کارخانه‌های نساجی و یا کارخانه‌های تولید آینه یافت. در فرایند تولید پارچه، پارچه‌ها (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 سیستم‌ها، دانشگاه صنعتی اصفهان moslehi@cc.iut.ac.ir

علی حکیمیان، دانشجوی کارشناسی ارشد، دانشکده مهندسی صنایع و سیستم‌ها، دانشگاه صنعتی اصفهان a.hakimian@in.iut.ac.ir

مصطفی ابویی اردکان، دانشجوی دکتری، دانشکده مهندسی صنایع و سیستمها، دانشگاه صنعتی اصفهان m.abouei@in.iut.ac.ir

[2]. *Permutation Flowshop*

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 برروی حداقل سازی زمان اتمام کل[1] تمرکز داشته‌اند [۲و۳] و تاکنون بررسی‌های کمتری برای حداقل سازی تعداد کارهای دارای دیرکرد برروی مسئله فلوشاپ انجام پذیرفته است. پیچیدگی حداقل سازی تعدادکارهای 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[2] $O(nLogn)$ 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 دوماشین 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 ‎($F2|d_j=D|\Sigma U_j$)‎ ارائه کردند که از $O(nD^2)$ می‌باشد.

حریری و پتس [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[3] برروی یک فلوشاپ دو ماشین را بررسی‌کردند. 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 ین [۱۰]، مسئله حداقل کردن توام تعداد کارهای دارای دیرکرد و حداقل کردن مجموع زودکردهای وزن‌دار را در یک ماشین بررسی‌کردند. آنها برای حل مسئله از روش شاخه و کران استفاده‌کرده و برای حل مسائل بزرگ از یک الگوریتم ابتکاری با رتبه $O(n^3)$ بهره جستند.

باپتیست و همکاران [۱۱] یک الگوریتم شاخه وکران 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.

سادیکف [۱۲] یک روش شاخه و کنترل[4] 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 وکران که در آن یک ساده‌سازی جانشینی که منجر به یک مسئله کوله پشتی چند انتخابه[5] می‌شود پیشنهاد دادند. آنها از همین روش برای حداقل کردن وزنی تعداد کارهای دارای دیرکرد برروی پردازشگرهای موازی نیز استفاده کردند [۱۴].

هی و همکاران [۱۵] رویکردی دوگانه از الگوریتم حریصانه روبه جلو[6] را برای حداقل کردن تعداد کارهای دارای دیرکرد در مسئله تک ماشین ارائه کردند. آنها برای هر کار یک ضرب العجل[7] نیز در نظرگرفتند. الگوریتم ارائه شده از رتبه $O(nlogn)$ 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 کردندکه زمان ورود کارها متفاوت باشد.

رویز تورس و سنتنو [۱۷] ، مسئله حداقل کردن تعداد کارهای دارای دیرکرد برروی یک فلوشاپ جایگشتی و با منابع ثانویه[8] 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

1.Makespan
2.Moore
3.sum of maximum Earliness and Tardiness
4.Branch-and-Check
5. Multiple-choice
6.Forwards Greedy Algorithms
7.Deadline
8.Secondary Resource

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 برروی 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 برروی فلوشاپ جایگشتی با دو ماشین و زمان‌های ورود غیر همزمان کارها در ادبیات موضوع مشاهد 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 برروی 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 برروی دو ماشین پردازش شوند.

تمام کارها ابتدا بر روی ماشین اول و سپس بر روی ماشین دوم پردازش می‌شوند. همچنین هرکاری که برروی 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[1] گفته می‌شود. هدف پیدا کردن یک توالی از کارها می‌باشد که در آن تعداد کارهایی که دارای دیرکرد هستند حداقل شوند. این 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 $P2|r_j|\Sigma U_j$ 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 $m=1,2$

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

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

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

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

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

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

$$C_{2[j]}=Max(C_{1[j]}+P_{2[j]}, C_{2[j-1]}+P_{2[j]}) \quad (۱)$$

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

$$T_{[j]}=Max\{0, C_{2[j]}-d_{J[j]}\} \quad (۲)$$

U_j: اگر $d_j \geq C_{2j}$ برابر با ۰ و در غیراین‌صورت برابر با ۱ $j=1,2,3,\dots,n$

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 iام توالی

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

[1]. Release Time

[2]. Tardiness

$$NT(\sigma_L)=\sum\nolimits_{j\in\sigma_L} U_j \qquad (۱)$$

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

$$NT(\sigma'_L)=\sum\nolimits_{j\notin\sigma_L} U_j \qquad (۲)$$

$\sigma'_{L\text{-MOORE } m}$: 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. باپتیست 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.

۲-۱. مدل برنامه ریزی خطی

در این بخش مدل برنامه‌ریزی خطی عدد صحیح مختلط مسئلهٔ 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 jام پردازش شود یک و در غیر این صورت صفر است. $\quad i,j=1,2,3,\ldots,n$

M : یک عدد بزرگ

۲-۱-۱. مدل برنامه‌ریزی عدد صحیح مختلط

$$\text{Minimize } NT=\sum\nolimits_{j=1}^{n} U_{[j]} \qquad (۳)$$

s.t.

$$\sum_{j=1}^{n} x_{i[j]} = 1 \qquad i = 1,2,3,\ldots,n \qquad (۴)$$

$$\sum_{i=1}^{n} x_{i[j]} = 1 \qquad j = 1,2,3,\ldots,n \qquad (۵)$$

$$C_{1[j]} \geq \sum_{i=1}^{n} r_i x_{i[j]} + \sum_{i=1}^{n} p_{1i} x_{i[j]} \qquad j = 1,2,3,\ldots,n \qquad (۶)$$

$$C_{1[j]} \geq C_{1[j-1]} + \sum_{i=1}^{n} p_{1i} x_{i[j]} \qquad j = 1,2,3,\ldots,n \qquad (۷)$$

$$C_{2[j]} \geq C_{1[j]} + \sum_{i=1}^{n} p_{2i} x_{i[j]} \qquad j = 1,2,3,\ldots,n \qquad (۸)$$

$$C_{2[j]} \geq C_{2[j-1]} + \sum_{i=1}^{n} p_{2i} x_{i[j]} \qquad j = 1,2,3,\ldots,n \qquad (۹)$$

$$T_{[j]} \geq 0 \qquad j = 1,2,3,\ldots,n \qquad (۱۰)$$

$$T_{[j]} \geq C_{2[j]} - \sum_{i=1}^{n} d_i x_{i[j]} \qquad j = 1,2,3,\ldots,n \qquad (۱۱)$$

$$T_{[j]} \leq M.U_{[j]} \qquad j = 1,2,3,\ldots,n \qquad (۱۲)$$

$$x_{i[j]} \in \{0,1\} \qquad i,j = 1,2,3,\ldots,n \qquad (۱۳)$$

$$U_{[j]} \in \{0,1\} \qquad j = 1,2,3,\ldots,n \qquad (۱۴)$$

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 کنندهٔ زمان تکمیل کارها بر روی ماشین دوم می‌باشند. روابط (۱۲) تا (۱۴) حدود متغیر $T_{[j]}$ را تعیین می‌نماید. 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

مناسب و اصول غلبه کارایی استفاده شده، تا بتوان با قطع[1] 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||ΣUj منجر به جواب بهینه می‌شود [۵]. لذا با الهام از این الگوریتم، برای مسئله P2|rj|ΣUj یک الگوریتم ابتکاری به نام H_1 ارائه شده است. قدم‌های الگوریتمH_1 به صورت زیر است:

۱) ابتدا کارها به صورت زودترین موعد تحویل (EDD[2]) مرتب شوند.

۲) سپس کارها براساس ترتیب EDD و با توجه به زمان در دسترس هر کار (r_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 می شود 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 می شود.)

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

نتایج عددی نشان می‌دهد که این الگوریتم کارائی 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 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|ΣUj از روش شاخه وکران 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 $N_T(\sigma_L)$، یک تخمین برای تابع هدف کارهای چیده نشده (تخمینی از N_T 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[3] 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 کارائی روش ارائه شده است.

در سطح kام الگوریتم با توالی جزئی کارهای $\sigma_L = J_{[1]},\dots,J_{[k]}$، 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|ΣUj به صورت زیر است:

$$LB_1= NT(\sigma_L)+ NT(\sigma'_{L\text{-}MOORE2}) \qquad (۱۵)$$

که در آن $NT(\sigma_L)$ تعداد کارهای دارای دیرکرد در توالی جزئی σ_L و $NT(\sigma'_{L\text{-}MOORE2})$ تعداد کارهای دارای دیرکرد در توالی $\sigma'_{L\text{-}MOORE2}$ 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 وکارهای 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

---
[1]. *fatom*
[2]. *Earliest Due Date*
[3]. *Back-Tracking*

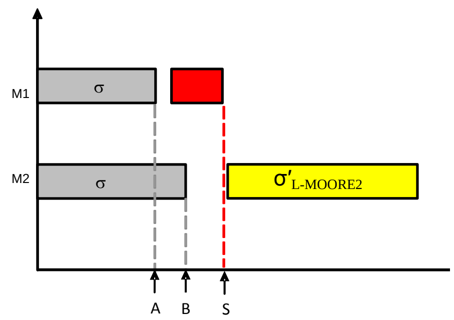
از تعداد کارهای دارای دیرکرد از کارهای چیده نشده σ'_L است که با $NT(\sigma'_{L\text{-}MOORE2})$ نشان داده می‌شود.

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.

$$S= \text{Max}\left\{\text{Min}_{i\in\sigma'_L}\{\text{Max}\{A, r_i\}+P_{1i}\}, B\right\} \qquad (۱۶)$$

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 لم۱

لم ۲ : یک حد پایین برای مسئله P2|rj|ΣUj به صورت زیر است:

$$LB_2= NT(\sigma_L)+ NT(\sigma'_{L\text{-}MOORE1}) \qquad (۱۷)$$

$NT(\sigma_L)$ 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.

اثبات: مشابه لم ۱. □

لم ۳: برای به دست آوردن یک حد پایین در مسئله P2|rj|ΣUj 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.

$$LB_3= NT(\sigma_L) +NT(\sigma'_L) \qquad (۱۸)$$

$$NT(\sigma'_L) =\sum_{i\notin\sigma_L}\delta(T'_{[i]}) \qquad (۱۹)$$

که $\delta(T'_{[i]})$ برابر است با،

$$\delta(T'_{[i]}) = \begin{cases}0 & \text{if } T'_{[i]} = 0\\ 1 & \text{if } T'_{[i]} > 0\end{cases} \qquad (۲۰)$$

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 $d_{[j]}$ نمایش داده شود می‌توان دیرکرد هر موقعیت را به صورت زیر محاسبه نمود،

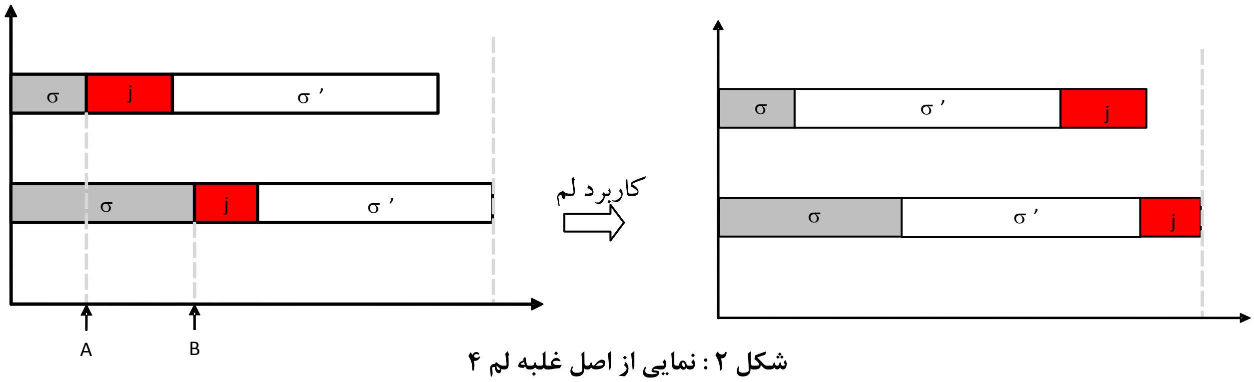
$$T'_{[i]}=MAX\{0, C_{[i]} - Min_{j\geq i}(d_{[j]})\} \qquad (۲۱)$$

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 موعدهای 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.

$$LB=MAX\{LB_1, LB_2, LB_3\} \qquad (۲۲)$$

لم ۴: اگر در گره **L** کار **i** از مجموعه کارهای چیده نشده برای چیدن در انتهای $\boldsymbol{\sigma_L}$ انتخاب شود وکار **i** در این موقعیت دارای دیرکرد شود، جواب بهینه‌ای وجود دارد که در آن کار **i** در آخر توالی قرار می‌گیرد.



شکل ۲ : نمایی از اصل غلبه لم ۴

اثبات: واضح است که با انتقال کار دیرکرددار 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.

$$R_i=min_{k\epsilon\sigma'}\{r_k\} \qquad (۲۳)$$

$$r_i \geq A \;\; ,r_j \geq r_i+P_{1i} \qquad (۲۴)$$

$$r_i + P_{1i} + P_{2i} \leq r_j+P_{1j} \qquad (۲۵)$$

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 ij و ji بعد از σ_L به صورت روابط (۲۸) و (۲۹) محاسبه می‌شود.

$$C_2\,(\sigma_L ij) = r_j+ P_{1j} + P_{2j} \qquad (۲۶)$$

$$C_2\,(\sigma_L ji)= r_j+ P_{1j} + Max\,(P_{1i}, P_{2j}) + P_{2i} \qquad (۲۷)$$

به سادگی دیده می‌شود که $C_2(\sigma_L ji) > C_2\,(\sigma_L ij)$، بنابراین کار i باید قبل از j پردازش شود. □

لم ۶: اگر دو کار i , j در دو توالی $\sigma_L ij$ و $\sigma_L ji$ 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.

$$C_1\,(\sigma_L ji) \geq C_1\,(\sigma_L ij) \qquad (۲۸)$$

$$C_2\,(\sigma_L ji) > C_2\,(\sigma_L ij) \qquad (۲۹)$$

اثبات: با توجه به شرایط ذکر شده، زمان تکمیل توالی جزئی $\sigma_L ji$ بر روی هر دو ماشین بدتر از زمان تکمیل توالی جزئی $\sigma_L ij$ نمی‌باشد. بنابراین توالی $\sigma_L ij$ 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 $\sigma_L ij$ 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 کارائی 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 [۱و۷۵] تولید شده‌اند. مقادیر موعد تحویل از توزیع یکنواخت گسسته در بازه (Pα , Pβ) به دست می‌آید که مقدار P از رابطه (۳۲) زیر حاصل می‌شود.

$$P=(1/n)*\left[\sum_{i=1}^{n}\sum_{j=1}^{2}Pij+(n-1)\max\{\sum_{i=1}^{n}Pi1\,,\sum_{i=1}^{n}Pi2\}\right] \qquad (۳۰)$$

مقادیر α و β که به ترتیب حد پایین و حدبالای موعدهای تحویل را نشان می‌دهد از دو مجموعه {0.6 , 0.4}∋α و {1 , 0.8}∋β به دست می‌آیند.

زمان های در دسترس بودن یعنی r_i از توزیع یکنواخت گسسته U (Pδ ,Pε) تولید شده که برای δ مقدار صفر و برای ε مقادیر {0.25 , 0.05} در نظر گرفته شده‌اند. برای داده‌ها دو گروه Low و High در نظر گرفته شده‌است. پارامترهای Low به

صورت (α=0.6 , β=0.8 , ε=0.05) و High به صورت (α=0.4 , β=1 , ε=0.25) تولید شده‌اند [۱۶]. 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]

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

همچنین برای بررسی کارائی 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 و۳ نتایج هر دو گروه را نشان می‌دهد. در جدول ۲ مسائل در گروه High و در جدول ۳ مسائل برای گروه Low حل 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} به‌ترتیب جواب بهینه و جواب الگوریتم ابتکاری با هدف $\sum(1-Ui)$ می‌باشند. با توجه به این‌که 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.

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

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

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

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

در جداول ۲ و ۳ متوسط شاخص Z^{heu} / Z^{opt} حداکثر 1/11 برابر می‌باشد و در مقایسه با تحقیقات دیگر مربوط به تعداد کارهای دیرکرددار [۲۲]، عدد کوچکی است که نشان دهنده‌ی کارآیی بالای الگوریتم ابتکاری است.

نتایج نشان می‌دهد 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 الگورریتم ابتکاری در اندازه‌های کوچک قادر است تا اکثر مسائل را به طور بهینه حل نماید و در اندازه‌های متوسط با تقریب مناسبی می‌تواند مسائل را حل نماید.

علاوه براین، نتایج نشان می‌دهد که با بزرگ شدن بازه r_j و d_j زمان حل و تعداد گره‌های حل شده در روش شاخه و کران کاهش می‌یابد. از آنجا که مسئله حداقل کردن تعداد کارهای دارای دیرکرد برروی فلوشاپ دوماشین تا کنون با هیچ فرضی با روش‌های بهینه حل نشده است برای مقایسه نتایج می‌توان به کار حریری و پتس اشاره کرد که آنها همین مسئله را برروی یک فلوشاپ M ماشین و بدون فرض دسترسی حل کردند. آنها توانستند برای فلوشاپ با ۲ ماشین تعدادی از مسائل با ۲۵ کار و برای فلوشاپ 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 می‌دهد 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 می‌باشد.

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

تعدادکارها	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. نتیجه‌گیری و پیشنهادات

دراین مقاله مسئله $P2|r_j|\Sigma U_j$ یعنی زمان‌بندی فلوشاپ جایگشتی دو ماشین با زمان‌های ورود غیر همزمان و هدف حداقل کردن تعداد کارهای دارای دیرکرد مورد بررسی قرار گرفت. در ابتدا نشان داده‌شد که این مسئله NP-hard است. از این رو یک الگوریتم شاخه‌وکران و یک الگوریتم ابتکاری برای حل آن ارائه شد.

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 شاخه‌وکران 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 کارآیی رویه ارایه 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 کننده‌ی 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

[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-Torresa, 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-Torresa, 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

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.