شمـــاره ۲، جلـــد ۲۰، تابستـــــان ۱۳۸۸

نشـــریه بین المللـــی مهنـــدسی صنایـــع و مدیـــریت تولیـــد

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

ISSN: 2008-4870

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

# Minimizing Makespan in Job Shop Scheduling Problem with Parallel Machines

**S. H. Zegordi*, M.A. BeheshtiNia & N. Jafari Hajagha**

***S. H. Zegordi,*** *Associate professor, Industrial Engineering Department, Tarbiat Modares University*
***M.A. BeheshtiNia,*** *Ph.D. Industrial Engineering Department, Tarbiat Modares Universit*
***N. Jafari Hajagha**.,. M.Sc student, Industrial Engineering Department, Tarbiat Modares University*

## Keywords

Flexible Job Shop,
Parallel Machines,
Mathematical Modeling,
Heuristic Algorithms

## ABSTRACT

*In this paper the problem of job shop scheduling with parallel machines in each stages is discussed. The objective is to minimize the maximum completion time (makespan). This problem is a combination of two classic problems of job shop and parallel machines which in this case parallel machines has been used as kind of flexibility in the job shop problem. The review of literature has shown that this problem has not been discussed yet. After presenting the mathematical mode, heuristic algorithms are used for solving this NP-hard problem. Regarding this, five algorithms are presented and a lower bound is developed. Finally all these algorithms have been analyzed. According to results the proposed algorithm of H2 works better than the others when there are few jobs. As the number of jobs increases H1 is more efficient than H2 asymptotically. Also the efficiency of H3 algorithm is the worst among the rest.*



# کمینه‌سازی حداکثر زمان تکمیل در مسأله زمانبندی تولید کارگاهی با ماشینهای موازی

**سید حسام‌الدین ذگردی، محمد علی بهشتی‌نیا و ناصر جعفری حاج‌آقا**

## کلمات کلیدی

تولید کارگاهی
انعطاف‌پذیر ،
ماشینهای موازی،
مدلسازی ریاضی،
الگوریتمهای ابتکاری

## چکیده:

در این مقاله، مسأله زمانبندی تولید کارگاه منعطف (Flexible Job Shop) با تعریف جدیدی از انعطاف پذیری مورد بررسی قرار می‌گیرد. در این نوع انعطاف پذیری برای مساله فرض می‌شود که در هر ایستگاه عملیاتی چند ماشین وجود دارند که کارها در هر ایستگاه می‌توانند به یکی از آنها تخصیص داده شود. تابع هدف کمینه‌سازی بازه ساخت (make span) است. ابتدا مدل ریاضی مساله ارائه شده و سپس NP-hard بودن مسأله نشان داده می‌شود. بعلت NP-hard بودن مساله استفاده از روشهای دقیق برای حل آن در زمان چندجمله‌ای ممکن نیست و باید از الگوریتمهای ابتکاری برای حل آن استفاده نمود. به این منظور دو الگوریتم ابتکاری به نامهای H1 و H2 به

**تاریخ وصول:** ۸۶/۱۰/۱۲
**تاریخ تصویب:** ۸۷/۸/۲۲

**دکتر سید حسام‌الدین ذگردی**، دانشیار بخش مهندسی صنایع، دانشکده فنی و مهندسی دانشگاه تربیت مدرس، zegordi@modares.ac.ir.
**محمد علی بهشتی‌نیا**، دانشجوی دکتری مهندسی صنایع، دانشکده فنی و مهندسی دانشگاه تربیت مدرس،beheshtinia@modares.ac.ir .
**ناصر جعفری حاج‌آقا**، دانشجوی کارشناسی ارشد مهندسی صنایع، دانشکده فنی و مهندسی دانشگاه تربیت مدرس، n_jafari@modares.ac.ir.

ترتیب برای مسائل با ابعاد بزرگ و معمولی برای حل مساله ارائه می‌شود. بعلت اینکه این مسأله تا کنون در ادبیات موضوع مورد مطالعه قرار نگرفته است، معیار مناسبی برای ارزیابی الگوریتم‌های ارائه شده وجود ندارد. بنابراین بمنظور ارزیابی الگوریتمهای ارائه شده، سه الگوریتم ابتکاری با نامهای H3، H4 و H5 و همچنین یک کران پایین برای آن ارائه می‌شود و نتایج الگوریتمهای H1 و H2 با آنها مقایسه شده است. نتایج محاسبات نشان می‌دهد که الگوریتم پیشنهادی H2 برای مسائل با ابعاد کوچک، جوابهای بهتری را نسبت به الگوریتمهای دیگر ارائه می‌دهد. اما در مسائل با ابعاد بزرگ H1 به طور مجانبی کاراتر از H2 است. همچنین کارائی الگوریتم H3 پایین‌تر از سایر الگوریتمها است.

## ۱. مقدمه

زمانبندی تولید کارگاهی[1] (JS) یکی از مهمترین مباحث مدیریت تولید است [۱] که شاخه‌ای از زمانبندی تولید محسوب شده و جزء پیچیده ترین مباحث بهینه‌سازی ترکیبی است [۲]. در مسأله JS فرض می‌شود که n کار وجود دارند که باید توسط m ایستگاه کاری مورد پردازش قرار گیرند. مسیر پردازش کارها مشخص بوده و حداقل برای دو کار مسیر پردازش کارها متفاوت از یکدیگر است [۳]. در مساله تولید کارگاهی (JS) فرض بر این است که در هر ایستگاه عملیاتی یک ماشین وجود دارد که تمام کارهای که باید در آن ایستگاه پردازش شوند، توسط آن ماشین مورد پردازش قرار می‌گیرند. تولید کارگاهی انعطاف‌پذیر[2] (FJS) توسعه مسأله JS است. در مساله FJS انواع متفاوتی از انعطاف پذیری را می‌توان تعریف نمود. از انواع انعطاف پذیری می‌توان انعطاف پذیری در مسیر پردازش کارها را نام برد که در آن برای هر کار چند مسیر پردازش متفاوت وجود دارد و هر کار می‌تواند یکی از این چند مسیر را انتخاب کند. همچنین می‌توان انعطاف پذیری در نوع عملیات ماشینها را نام برد که در آن یک ماشین خاص می‌تواند در صورت نیاز دو یا چند عملیات متفاوت از هم را انجام دهد. در این مقاله تعریف جدیدی از انعطاف پذیری برای مساله FJS ارائه شده است.

مساله تولید کارگاهی منعطف اولین بار در سال ۱۹۹۰ توسط براکر و اسچلای [۴] بررسی گردید. آنها حالت وجود ماشینهای چند منظوره را مورد بررسی قرار داده و برای مساله با دو کار، الگوریتمی چندجمله‌ای را ارائه نمودند. هارنیک و همکاران [۵] نیز حالت وجود ماشینهای چند منظوره در محیط تولید کارگاهی را مورد بررسی قرار داده‌اند. ماسترولیلی و گمبردلا [۶] یک روش جستجوی همسایگی را برای مساله تولید کارگاهی منعطف ارائه داده‌اند.

غدجتی[۷] یک رویکرد ترکیبی از روشهای ابتکاری بر مبنای الگوریتم ژنتیک (GA) برای حل مسأله FJS بمنظور کمینه سازی حداکثر زمان تکمیل کارها ($C_{\max}$) ارائه نموده است. یکی از ویژگیهای تحقیق وی در نظر گرفتن محدودیتهای تقدمی بین عملیات کارها، به دو صورت خطی و غیرخطی است. کاسم و همکارانش [۸] نخستین بار مسأله FJS را در حالت چندهدفی مورد بررسی قرار داده‌اند، که در آن روش ارائه شده ترکیبی از منطق فازی و الگوریتم‌های تکاملی است و انعطاف‌پذیری مسأله تنها به انعطاف‌پذیری عملیات محدود می‌شود. علاوه بر این براندیمارته نیز مساله FJS را با در حالت چندهدفه بررسی نموده‌است. وی از روش جستجو ممنوع[3] برای حل مساله استفاده نموده است.

لی و همکارانش [۹] یک الگوریتم ژنتیک برای حل مسأله مشابه با FJS در زنجیره تأمین ارائه کرده‌اند در مدل ارائه شده فرض وجود منابع بیرونی، فرض آلترناتیو ماشین برای هر عملیات و توالی عملیات چندگانه را برای هر قطعه در نظر گرفته است.

پارک و همکاران [۱۰] برای حل مساله تولید کارگاهی یک روش ابتکاری با تکیه بر الگوریتم ژنتیک منفرد و موازی ارائه نموده‌اند. براکر و نیر [۱۱] در سال ۱۹۹۸ به بررسی مساله تولید کارگاهی چند حالته پرداخته[4] و یک الگوریتم جستجو ممنوع برای حل آن ارائه نموده‌اند. در سال ۱۹۹۹براکر و همکارانش [۱۲] مساله تولید کارگاهی انعطاف‌پذیر با ماشین‌های چند منظوره[5] را مورد بررسی قرار دادند. این مسأله، حالت خاصی از کار کارگاهی انعطاف‌پذیر است که در آن، زمان پردازش هر عملیات به ماشینی که آن عملیات را انجام می‌دهد، بستگی ندارد.

کیم و همکارانش [۱۳] یک الگوریتم تکاملی برای یکپارچه‌سازی برنامه‌ریزی فرآیند و زمانبندی JS ارائه داده‌اند، که از ویژگیهای این تحقیق می‌توان به جستجوی موازی چندگانه و در نظر گرفتن سه نوع انعطاف‌پذیری (انعطاف‌پذیری عملیات، توالی عملیات، و فرآیند) در برنامه‌ریزی فرآیند اشاره کرد. اسکریچ و همکارانش [۱۴] انعطاف‌پذیری در مسأله JS را تنها به انعطاف‌پذیری در عملیات محدود کرده‌اند. تابع هدف مساله آنها کمینه سازی دیرکردها است که با استفاده از الگوریتم TS به حل مسأله پرداخته شده است. ژیا و وو [۲] مسأله FJS مطالعه شده توسط کاسم و همکارانش [۸] را با روشی جدید مورد مطالعه قرار دادند. آنها یک رویکرد ترکیبی از PSO[6] و SA بوجود آوردند که PSO درتخصیص عملیات به ماشینها مورد استفاده قرارگرفته و از SA برای تعیین توالی عملیات استفاده شده است.

---

[1] *Job shop*

[3] *Tabu Search*
[4] *Multi-mode*
[5] *Multi Purpose Machine*
[6] *Particle Swarm Optimization*

در این مقاله نوع جدیدی از انعطاف پذیری تعریف می‌شود که در آن در هر ایستگاه چند ماشین برای پردازش کارها وجود دارد و هر کار در هر ایستگاه می‌تواند یکی از ماشینها را برای پردازش انتخاب کند. (بر خلاف حالت معمولی مساله تولید کارگاهی که در آن فرض می‌شود در هر مرحله تنها یک ماشین وجود دارد و تمام عملیات حتما باید توسط آن ماشین پردازش شوند). حل مسأله زمانبندی FJS شامل دو زیر مسأله مسیریابی (تخصیص ماشین به عملیات) و زمانبندی عملیات (تعیین توالی عملیات) می‌باشد [۱۵].

این نوع انعطاف پذیری قبلا در مورد مساله تولید جریانی[1] بررسی شده است. مساله تولید جریانی منعطف[2] مطالعه شده‌است اما تا کنون مساله تولید کارگاهی بررسی نشده است. در ادامه این بخش مرور ادبیات مربوط به مساله تولید جریانی منعطف بیان می‌گردد که نوآوری ارائه شده را در مسائل تولید جریانی مورد بررسی قرار داده‌اند.

ریانه و همکارانش [۱۶] مسأله تولید جریانی منعطف (FFS) سه مرحله‌ای را با ساختار ویژه یک ماشین در مراحل اول و سوم و دو ماشین در مرحله دوم با هدف کمینه‌سازی حداکثر زمان تکمیل کارها[3] ($C_{\max}$) مورد بررسی قرار داده‌اند. برای این مسأله آنها یک مدلسازی ریاضی و دو رویکرد ابتکاری ارائه کرده‌اند. درهمان سال نوویکی و اسموتنیکی [۱۷] یک الگوریتم جستجوی ممنوع برای حل مسأله تولید جریانی با ماشینهای موازی ارائه کردند. کاری مشابه با کارهای قبلی نیز توسط دسوکی و همکارانش [۱۸] انجام شده است آنها مسأله زمانبندی تولید جریانی با ماشینهای موازی یکنواخت را با فرض یکسان بودن کارها با هدف کمینه سازی $C_{\max}$ مورد بررسی قرار داده و نشان دادند. مسأله با دو مرحله در زمان چند جمله‌ای قابل حل است و برای بیش از دو مرحله NP-hard است. همچنین بر مبنای تجزیه مسأله سه مرحله‌ای به سه مرحله جداگانه و استفاده از روش شاخه وکران برای مرحله دو و روشهای ابتکاری برای مراحل یک و سه، به یک الگوریتم بهینه دست یافته‌اند و یک الگوریتم ابتکاری نیز برای مسأله سه مرحله‌ای ارائه کرده‌اند

لو[۱۹] با استفاده از یک الگوریتم ابتکاری بر مبنای الگوریتم شبیه‌سازی تبرید[4] (SA) به حل مسأله زمانبندی تولید جریانی چند مرحله‌ای با ماشینهای موازی متفاوت پرداخته است که در آن هدف کمینه‌سازی کل زمان جاری است. کیپرسز و کولاماس [۱۵] مسأله زمانبندی تولید جریانی انعطاف‌پذیر با ماشینهای موازی یکنواخت را با تابع هدف کمینه‌سازی $C_{\max}$ مورد مطالعه قرار داده‌اند و دسته‌ای از هیوریستیکهای موجود برای مسأله جریان‌کارگاهی را بسط داده و اثبات کرده‌اند هیوریستیک مبتنی بر روش جمع برداری، هنگامی که تعداد کارها بسیار زیاد باشد بطور مجانبی بهینه است.

[1] Flow shop
[2] Flexible Flow Shop
[3] Makespan
[4] Simulated Annealing

کرز و آسکین [۲۰-۲۱] مساله جریان کاری منعطف را با در نظر گرفتن زمانهای آماده سازی وابسته به توالی، مورد مطالعه قرار داده‌اند. آنها سه دسته از الگوریتمهای ابتکاری را نخست برای حل مساله مزبور ارائه داده‌اند که الگوریتمهای دسته اول بر مبنای تخصیص ساده کارها به ماشینها، دسته دوم بر مبنای روشهای حل مساله فروشنده دوره‌گرد و دسته سوم بر مبنای قاعده جانسون استوار هستند[ ۲۰]. آنها مجدداً به بررسی همین مساله پرداخته‌اند و در آن یک الگوریتم ابتکاری بر مبنای الگوریتم ژنتیک (RKGA) برای حل مسأله ارائه داده‌اند[۲۱].

همانگونه که ذکر گردید در این مقاله تعریف جدیدی از انعطاف پذیری برای مساله تولید کارگاهی (JS) ارائه شده است که در آن فرض می‌شود که در هر ایستگاه عملیاتی ماشینهای مختلفی وجود دارند و هر عملیات در هر ایستگاه عملیاتی می‌تواند روی یکی از ماشینهای آن مرحله مورد پردازش قرار گیرد. علاوه بر آن فرض می‌شود در هر مرحله سرعت پردازش ماشینها می‌تواند متفاوت باشد. این مسأله زمانبندی ترکیب دو مسأله سنتی زمانبندی در محیط کارگاهی(JS) و محیط ماشینهای موازی (PM) است و ما آن را با کارگاهی با JSPM نشان می‌دهیم.

## ۲. تعریف مسأله و مدلسازی ریاضی

### ۲-۱. تعریف مسأله

در مسأله زمانبندی تولید کارگاهی با ماشینهای موازی (JSPM) در نظر گرفته شده در این تحقیق فرض می‌شود تعداد $n$ کار و $m$ ایستگاه عملیاتی وجود دارد که در هر ایستگاه یک عملیات خاص انجام می‌شود. فرض می‌شود هر کار باید از تمام $m$ مرحله عملیاتی عبور کند ولی مسیر طی کردن ایستگاهها برای هر کار می‌تواند متفاوت از کارهای دیگر باشد. در هر ایستگاه عملیاتی چند ماشین بطور موازی برای پردازش کارها وجود دارند که سرعت آنها با یکدیگر متفاوت است و هر کار در هر ایستگاه عملیاتی می‌تواند روی یکی از آنها مورد پردازش قرار گیرد. همچنین فرض می‌شود هر کار در هر مرحله نمی‌تواند بیش از یکبار مورد پردازش قرار گیرد. ترتیب حرکت هر کار بین مراحل مختلف مسیر ساخت نامیده می‌شود که این مسیر برای کارهای مختلف متفاوت ولی از پیش تعیین شده است. سایر فرضیات مساله به صورت زیر هستند:

۱- هر ماشین در هر لحظه تنها می تواند یک کار را مورد پردازش قرار دهد.
۲- هر کار در هر ایستگاه به منظور پردازش تنها باید به یک ماشین تخصیص یابد.
۳- همه ماشینها و کارها از زمان صفر در دسترس هستند.
۴- ماشینها همیشه در دسترس هستند و خراب نمی شوند.
۵- قطع کردن عملیات[5] مجاز نیست.

[5] Preemption

۶- انبار میانی بین ایستگاهها وجود دارد و ظرفیت آن نامحدود است.
۷- زمانهای پردازش عملیات قطعی و معلوم هستند.
۸- زمانهای آماده سازی بین عملیات ناچیز است یا جزئی زمانهای پردازش هستند.
۹- زمان حمل و نقل بین ایستگاههای عملیاتی قابل چشم پوشی است.

### ۲-۲. مدل ریاضی

در این قسمت مدل ریاضی عدد صحیح مختلط برای مساله مورد بررسی ارائه می‌گردد. نمادهای مورد استفاده برای پارامترهای مساله عبارتند از:

$n$ : تعداد کارها
$j=1,...,n$ : شاخص کار
$J_j$ : $j$ امین کار
$m$ : تعداد مراحل پردازش
$i=1,...,m$ : شاخص عملیات(چون فرض شده است که هر کار باید از تمام مراحل بگذرد، بنابراین هر کار دارای m عملیات است).
$k=1,...,m$ : شاخص مرحله (چون فرض شده است که هر کار باید از تمام مراحل بگذرد، بنابراین حداکثر مقدار k برابر m است).
$M_k$ : $k$ امین مرحله پردازش
$M_{k,r}$ : $r$ امین ماشین موازی در مرحله $k$ ام
$o_{i,j}$ : $i$ امین عملیات کار $j$
$St_{i,j}$ : مرحله پردازش عملیات $o_{i,j}$
$l_k$ : تعداد ماشینهای موازی در مرحله $k$ ام
$r=1,...,l_k$ : شاخص ماشین
$S_{k,r}$ : سرعت ماشین $M_{k,r}$ (بر حسب ماشین-ساعت در هر ساعت). به عبارت دیگر ماشین $M_{k,r}$ در هر واحد زمانی می‌تواند $S_{k,r}$ ماشین-ساعت را پردازش نماید.
$P_{i,j}$ : زمان پردازش عملیات $o_{i,j}$ (بر حسب ماشین-ساعت)
$L$ : یک مقدار ثابت بسیار بزرگ

همچنین نمادهای مورد استفاده برای متغیرهای تصمیم مساله بصورت زیر هستند:

$C_{i,j}$ : زمان تکمیل عملیات $o_{i,j}$
$C_{\max}$ : حداکثر زمان تکمیل کارها (یا Makespan)
$c_{i,j}$ : زودترین زمان تکمیل عملیات $o_{i,j}$
$avail_{k,r}$ : زمان دسترسی به ماشین $M_{k,r}$
$X_{i,j,r}=\begin{cases}1\\0\end{cases}$ : اگر کار $j$ ام روی ماشین $r$ ام مرحله $St_{i,j}$ پردازش شود برابر ۱ و در غیر اینصورت برابر صفر در نظر گرفته می‌شود.
$Y_{a,b,i,j}=\begin{cases}1\\0\end{cases}$ : اگر عملیات $o_{a,b}$ زودتر از عملیات $o_{i,j}$ صورت پذیرد برابر ۱ و در غیر اینصورت برابر صفر در نظر گرفته می‌شود.

$$\min\ Z=C_{\max} \tag{۱}$$

$$C_{i,j}\ge C_{(i-1),j}+\sum_{r=1}^{l_{St_{i,j}}}(X_{i,j,r}\times P_{i,j}/S_{St_{i,j},r}) \quad \forall i:1\le j\le n,\ \forall j:2\le i\le m. \tag{۲}$$

$$C_{1,j}\ge \sum_{r=1}^{l_{St_{1,j}}}(X_{1,j,r}\times P_{1,j}/S_{St_{1,j},r}) \quad \forall j:1\le j\le n. \tag{۳}$$

$$\sum_{r=1}^{l_{St_{i,j}}}X_{i,j,r}=1 \quad \forall i:1\le j\le n,\ \forall j:1\le i\le m. \tag{۴}$$

$$C_{i,j}\ge C_{a,b}+P_{i,j}/S_{St_{i,j},r}-L\times(3-Y_{a,b,i,j}-X_{i,j,r}-X_{a,b,r}) \quad \forall j,b\in\{1\le j,b\le n\mid j\ne b\},\ \forall r\in\{1,\dots,l_{St_{i,j}}\} \tag{۵}$$

$$C_{a,b}\ge C_{i,j}+P_{a,b}/S_{St_{a,b},r'}-L\times(2+Y_{a,b,i,j}-X_{i,j,r}-X_{a,b,r}) \quad \forall i,a\in\{1\le i,a\le m\mid St_{i,j}=St_{a,b}\},\ \forall r'\in\{1,\dots,l_{St_{a,b}}\} \tag{۶}$$

$$C_{\max}\ge C_{m,j} \quad \forall j:1\le j\le n \tag{۷}$$

### ۳-۲. شرح مدل

رابطه (۱) تابع هدف مسأله را که کمینه‌سازی حداکثر زمان تکمیل برای کارها است را نشان می‌دهد. مجموعه محدودیت (۲) عدم تداخل زمانی برای مجموعه توالی عملیات کارها را تضمین می‌کند. به عبارتی مجموعه این محدودیتها باعث می‌شود هر عملیات از یک توالی در صورتی شروع شود که عملیات متقدم آن تکمیل شده باشد. مجموعه محدودیت (۳) زمان تکمیل اولین عملیات برای هرکاری را بزرگتر یا مساوی طول زمان پردازش آن بر روی ماشین

مربوطه قرار می‌هد. مجموعه محدودیت (۴) هر کار را در هر مرحله دقیقاً روی یک ماشین قرار می‌دهد و برای اینکه مجموعه عملیاتی که روی یک ماشین انجام می‌شوند تداخل زمانی نداشته باشند، مجموعه محدودیتهای (۵) و (۶) در مدل گنجانده شده است. مجموعه محدودیت (۷) نیز حداکثر زمان تکمیل کارها را محاسبه می‌کند.

### ۲-۴. NP-hard بودن مسأله

هوگوین و همکارانش اثبات کرده‌اند که مسأله زمانبندی تولید جریانی با ماشینهای موازی یکسان ( $FP_m \| C_{\max}$ ) در حالت خاص $FP2 \| C_{\max}$ با $l_1 = 1, l_2 = 2$ یا $l_1 = 2, l_2 = 1$ حتی زمانی که اجازه قطع پردازش عملیات را نیز داریم قویاً NP-hard است [۸]. بنابراین از آنجایی که مسأله زمانبندی تولید کارگاهی با ماشینهای موازی بسط مسأله زمانبندی تولید جریانی با ماشینهای موازی یکنواخت ($FQ_m \| C_{\max}$) است و مسأله $FQ_m \| C_{\max}$ بسط $FP_m \| C_{\max}$ است، قویاً NP-hard خواهد بود.

## ۳. حل مساله

از آنجا که مسأله JSPM جزء مسائل NP-hard است، در نتیجه الگوریتمی که بتواند در زمان چندجمله ای به جواب بهینه مسأله با اندازه بزرگ حتی متوسط دست یابد وجود نخواهد داشت و الگوریتمی که بتواند به جواب بهینه برسد، زمان بسیار زیادی را لازم دارد. در این مقاله به منظور حل مساله الگوریتم ابتکاریH1 برای مسائل با ابعاد بزرگ و الگوریتم ابتکاری H2 برای مسائل با ابعاد کوچک و متوسط ارائه شده است. به علت جدید بودن مساله در ادبیات موضوع روش حلی برای مساله مذکور ارائه نشده است و معیار مناسبی برای ارزیابی الگوریتمهای ارائه شده وجود ندارد. بنابراین الگوریتمهای H3، H4 و H5 (بر پایه قواعد زمانبندی) برای حل مساله ارائه گردیده اند که با الگوریتمهای H1 و H2 مقایسه می‌شوند. به منظور ارزیابی بیشتر الگوریتمهای ارائه شده یک کران پایین نیز در بخش بعد توسعه داده می‌شود. مسأله زمانبندی JSPM به دو زیر مسأله، مسیریابی و تعیین توالی عملیات تقسیم می‌شود که در زیر مسأله مسیریابی، عملیات به یکی از ماشینها که توانایی پردازش آن را دارد تخصیص داده می‌شوند و در زیر مسأله تعیین توالی عملیات، توالی انجام عملیات مشخص می‌شود. در الگوریتم H1 به صورت یکپارچه دو زیر مسأله حل می‌شود ولی در الگوریتمهای H2, H3, H4, H5 تعیین توالی در مرحله اول انجام می‌گیرد و در مرحله بعدی زیر مسأله مسیریابی حل می‌شود.

### ۱-۳. الگوریتم H1

بمنظور حل مساله در ابعاد بزرگ الگوریتم ابتکاری H1 ارائه می‌گردد. این الگوریتم اولویت بندی کارها با مرتب کردن صعودی کارها بر اساس تاریخ تکمیلشان بدست می‌آورد. در این حالت کاری دارای اولویت است که زمان تکمیل کمتری دارد (ECT[1]). این الگوریتم توانایی حل مسأله به صورت یکپارچه را دارد یعنی به طور همزمان از اطلاعات مربوط به کارها و ماشینها استفاده می‌کند. گامهای اجرای این الگوریتم در زیر آورده شده است.

**گام صفر:** قرار دهید (i=1) و مجموعه عملیات قابل زمانبندی را به شکل زیر تشکیل دهید.

$$B := \{o_{1,j} \mid 1 \le j \le n\}$$

**گام یک:** برای هر یک از عملیات متعلق به مجموعه $B$ زودترین زمان تکمیل ($c_{i,j}$) را به روش زیر بدست آورید:

اگر i=1 آنگاه

$$c_{i,j} = \min\{avail_{st_{i,j},r} + (P_{i,j} / S_{st_{i,j},r}),\ r = 1,\dots,l_k\}$$

اگر i≠1 آنگاه:

$$c_{i,j} = \min\{\max(avail_{st_{i,j},r}, c_{i-1,j}) + (P_{i,j} / S_{st_{i,j},r}),\ r = 1,\dots,l_k\}$$

که $avail_{k,r}$ زمان آماده بودن ماشین $r$ در مرحله $k$ است.

**گام دو:** اولین عملیات ($o^*_{i,j}$) متعلق به مجموعه $B$ که دارای زودترین زمان تکمیل است را انتخاب کنید و آنرا به ماشینی تخصیص دهید که زودترین زمان تکمیل روی آن بدست آمده است.

$$o^*_{i,j} = \{o_{i,j} \in B \mid c_{i,j} = \min\{c_{i',j'} \in B\}\} \qquad (۸)$$

**گام سه:** عملیات زمانبندی شده $o^*_{i,j}$ را از مجموعه $B$ حذف کنید.

$$B := B \setminus \{o^*_{i,j}\}$$

**گام چهار**: عملیات $o^*_{i+1,j}$ را که پس نیاز عملیات $o^*_{i,j}$ است را در صورت وجود به مجموعه $B$ اضافه کنید.

$$B := B \cup \{o^*_{i+1,j}\} \qquad (۹)$$

**گام پنج:** اگر $B$ ناتهی بود به گام یک برگردید. در غیر اینصورت به الگوریتم خاتمه دهید.

### ۲-۳. الگوریتم H2

این الگوریتم برای حل مساله آن را به دو زیر مساله تعیین توالی کارها و تخصیص کارها به ماشینها تقسیم می‌کند. تعیین توالی عملیات با مرتب کردن نزولی کارها بر پایه مقدار زمان باقیمانده از

[1] *Earliest Completion Time*

تکمیل نهاییشان (MWR[1]) صورت می‌پذیرد. بر اساس اولویت تعیین شده هر عملیات نیز به ماشینی تخصیص می یابد که زودتر از سایر ماشینها بتواند آنرا تکمیل کند.
در مرحله نخست برای تعیین توالی عملیات از وزنهای ویژه‌ای بر مبنای ساختار مسأله برای عملیات استفاده می شود. گامهای این الگوریتم در زیر آورده شده است.

**گام صفر:** وزن مجازی مربوط به هر عملیات را به طریق زیر بدست آورید.

$$w_{i,j} = \left(l_{st_{i,j}}\right) \Bigg/ \left(\sum_{r=1}^{l_k} S_{st_{i,j},r}\right) \qquad (۱۰)$$

**گام یک:** زمان پردازش مجازی مربوط به هر یک از عملیات را از فرمول زیر محاسبه کنید.

$$P'_{i,j} = P_{i,j} \times w_{i,j} \qquad (۱۱)$$

**گام دو:** قرار دهید (i=1) و مجموعه عملیات قابل زمانبندی را به شکل زیر تشکیل دهید.

$$A := \{o_{1,j} \mid 1 \le j \le n\} \qquad (۱۲)$$

**گام سه:** برای هر یک از عملیات متعلق به A مجموع زمانهای پردازش باقیمانده مجازی را از فرمول زیر محاسبه کنید.

$$WR'_j = P'_{i,j} + P'_{i+1,j} + \ldots + P'_{m,j} \qquad (۱۳)$$

**گام چهار:** از مجموعه A کاری را که دارای بیشترین مقدار $WR'_j$ است را انتخاب و در مجموعه‌ای بنام B قرار دهید. اگر چند کار دارای مقدار بیشینه $WR'_j$ هستند، همه آنها را در مجموعه B قرار دهید.

$$B = \{o_{i,j} \in A \mid WR'_j = \max\{WR'_{j'} \mid o_{i',j'} \in A\}\} \qquad (۱۴)$$

**گام پنج:** گام یک و دو الگوریتم H1 را اجرا کنید.

**گام شش:** عملیات زمانبندی شده $o^*_{i,j}$ را از مجموعه A حذف کنید.

$$A := A \setminus \{o^*_{i,j}\} \qquad (۱۵)$$

**گام هفت:** عملیات $o^*_{i+1,j}$را که پس نیاز عملیات $o^*_{i,j}$ است را در صورت وجود به مجموعه A اضافه کنید.

$$A := A \cup \{o^*_{i+1,j}\} \qquad (۱۶)$$

**گام هشت:** اگر A ناتهی بود به گام سه برگردید. در غیر اینصورت به الگوریتم خاتمه دهید.

### ۳-۳. الگوریتم H3

این الگوریتم بر پایه قاعده اولویت بندی LWR[2] توالی عملیات را تعیین می‌کند. بر اساس اولویت تعیین شده هر عملیات نیز به ماشینی تخصیص می یابد که زودتر از سایر ماشینها بتواند آنرا تکمیل کند. تفاوت این الگوریتم با الگوریتم H2 تنها در گام چهار است که این گام نیز به صورت زیر در این الگوریتم پی گرفته می‌شود.

**گام چهار:** از مجموعه A کاری را که دارای کمترین مقدار $WR'_j$ است را انتخاب و در مجموعه‌ای بنام B قرار دهید. اگر چند کار دارای مقدار کمینه $WR'_j$ هستند، همه آنها را در مجموعه B قرار هید.:

$$B = \{o_{i,j} \in A \mid WR'_j = \min\{WR'_{j'} \mid o_{i',j'} \in A\}\} \qquad (۱۷)$$

### ۴-۳. الگوریتم H4

این الگوریتم بر پایه قاعده اولویت بندی SPT[3] توالی کارها را تعیین می‌کند. بر اساس اولویت تعیین شده هر عملیات نیز به ماشینی تخصیص می یابد که زودتر از سایر ماشینها بتواند آنرا تکمیل کند.
به منظور تعیین توالی عملیات از وزنهای ویژه‌ای بر مبنای ساختار مساله استفاده می‌شود. گامهای این الگوریتم به شرح زیر است:

**گام صفر:** وزن مجازی مربوط به هر یک از عملیات را به طریق زیر بدست آورید.

$$w_{i,j} = l_{st_{i,j}} \Bigg/ \sum_{r=1}^{l_k} S_{st_{i,j},r} \qquad (۱۸)$$

**گام یک:** زمان پردازش مجازی مربوط به هر یک از عملیات را از فرمول زیر محاسبه کنید.

$$P'_{i,j} = P_{i,j} \times w_{i,j} \qquad (۱۹)$$

**گام دو:** قرار دهید (i=1) و مجموعه عملیات قابل زمانبندی را به شکل زیر تشکیل دهید.

$$A := \{o_{1,j} \mid 1 \le j \le n\} \qquad (۲۰)$$

**گام سه:** از مجموعه A کاری را که دارای کمترین مقدار زمان پردازش مجازی است را انتخاب و در مجموعه‌ای بنام B قرار دهید. اگر چند کار دارای مقدار کمینه زمان پردازش مجازی هستند، همه آنها را در مجموعه B قرار هید.:

$$B = \{o_{i,j} \in A \mid P'_{i,j} = \min\{P'_{i',j'} \mid o_{i',j'} \in A\}\} \qquad (۲۱)$$

[1] *Most Work Remaining*

[2] *Least Work Remaining*

[3] *Shortest Processing Time*

**گام چهار:** گام یک و دو الگوریتم H1 را اجرا کنید.

**گام پنج:** عملیات زمانبندی شده $o^*_{i,j}$ را از مجموعه A حذف کنید.

$$A := A \setminus \{o^*_{i,j}\} \qquad (۲۲)$$

**گام شش:** عملیات $o^*_{i+1,j}$ را که پس نیاز عملیات $o^*_{i,j}$ است را در صورت وجود به مجموعه A اضافه کنید.

$$A := A \cup \{o^*_{i+1,j}\} \qquad (۲۳)$$

**گام هفت:** اگر A ناتهی بود به گام سه برگردید. در غیر اینصورت به الگوریتم خاتمه دهید.

### ۳-۵. الگوریتم H5

این الگوریتم بر پایه قاعده اولویت بندی LPT[1] توالی عملیات را تعیین می‌کند. بر اساس اولویت تعیین شده هر عملیات نیز به ماشینی تخصیص می یابد که زودتر از سایر ماشینها بتواند آنرا تکمیل کند. تفاوت این الگوریتم با الگوریتم H4 تنها در گام سه است که این گام به صورت زیر در این الگوریتم دنبال می‌شود

**گام سه:** مجموعه عملیاتی از A را که بیشترین زمان پردازش مجازی را دارند در مجموعه B قرار دهید.

$$B = \{o_{i,j} \in A \mid P'_{i,j} = \max\{P'_{i',j'} \mid o_{i',j'} \in A\}\} \qquad (۲۴)$$

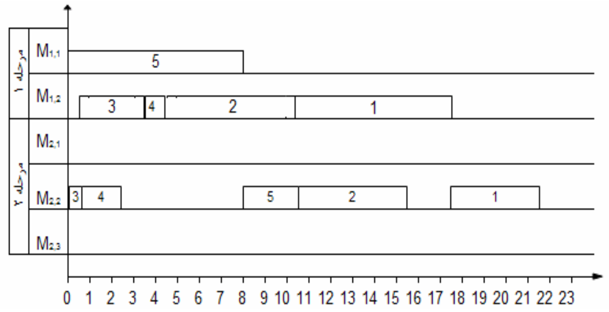
### ۳-۶. مثال عددی

بمنظور تبیین بیشتر در این بخش یک مثال عددی ذکر شده و توسط هر یک از پنج الگوریتم ارائه شده حل می‌گردد. فرض کنید ۵ کار و دو مرحله عملیاتی وجود دارند. در مرحله اول دو ماشین وجود دارند که سرعتهای آنها به ترتیب برابر ۱ و ۲ است. در مرحله دوم نیز سه ماشین وجود دارند که سرعتهای آنها به ترتیب برابر ۱، ۴ و ۲ است. همچنین زمان پردازش کارها و توالی ساخت آنها از جدول ۱ تبعیت می‌کند. با حل این مساله توسط هر یک از ۵ الگوریتم ابتکاری ارائه شده، نتایج نشان داده شده در شکلهای ۱ تا ۵ بدست می آید.

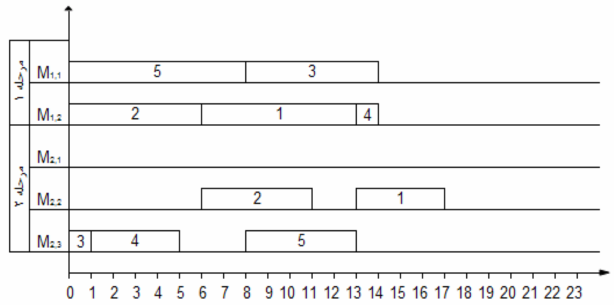
**جدول۱. داده‌های مربوط به مثال عددی**

| کار | مرحله۱ | مرحله۲ | توالی‌ساخت |
|---|---|---|---|
| ۱ | ۱۴ | ۱۶ | ۱→۲ |
| ۲ | ۱۲ | ۲۰ | ۱→۲ |
| ۳ | ۶ | ۲ | ۲→۱ |
| ۴ | ۲ | ۸ | ۲→۱ |
| ۵ | ۸ | ۱۰ | ۱→۲ |

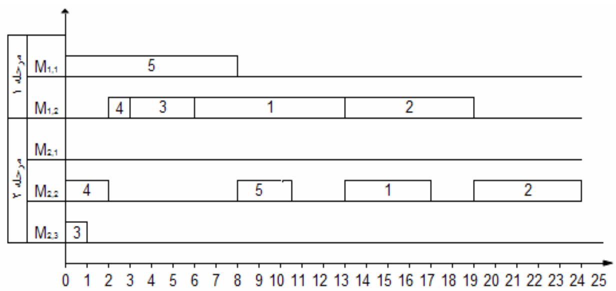
[1] *Longest Processing Time*



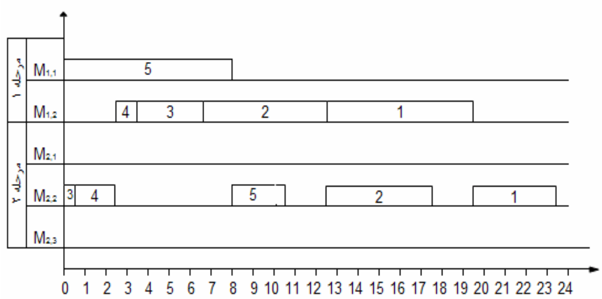

**شکل ۱. حل مثال عددی توسط الگوریتم H1**



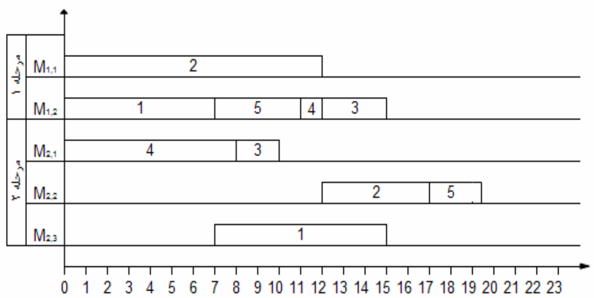

**شکل ۲. حل مثال عددی توسط الگوریتم H2**




**شکل ۳. حل مثال عددی توسط الگوریتم H3**




**شکل ۴. حل مثال عددی توسط الگوریتم H4**




**شکل ۵. حل مثال عددی توسط الگوریتم H5**

## ۴. محاسبه یک کران پایین برای مسأله

در این قسمت دو کران پایین برای مساله توسعه داده می‌شود. یکی از این کرانها مبتنی بر کار و دیگری مبتنی بر ماشین نامگذاری شده است. بدیهی است که در مورد هر مساله هر یک از این دو کران که مقدار بزرگتری دارد، می تواند بعنوان کران پایین اصلی مساله مورد استفاده قرار گیرد. در محاسبه کران مبتنی بر هرکار فرض می‌شود که تنها یک کار برای زمانبندی آماده است در نتیجه انتظار در صف برای این کار در هر مرحله برابر صفر خواهد شد و در هر مرحله روی سریعترین ماشین مرحله مورد نظر پردازش خواهد شد. اگر فرض شود که $T_{i,j}$ مدت زمانی باشد که عملیات $o_{i,j}$ برای پردازش در مرحله $St_{i,j}$ در صف باقی مانده است و در آن مرحله روی ماشین $r'$ پردازش شود در این صورت می‌توان کران مبتنی بر کار را به طریق زیر بدست آورد.

$$C_{m,j} = \sum_{i=1}^{m} (T_{i,j} + P_{i,j} \big/ S_{St_{i,j},r'}) \quad (۲۵)$$
$$\Rightarrow C_{m,j} \ge \sum_{i=1}^{m} (P_{i,j} \big/ S_{St_{i,j},r'})$$

از طرفی چون $S_{st_{i,j},r'} \le (\max_r S_{st_{i,j},r})$ پس می‌توان نتیجه گرفت که $P_{i,j} \big/ (\max_r S_{st_{i,j},r}) = \min_r (P_{i,j} \big/ S_{st_{i,j},r})$، در نتیجه:

$$P_{i,j} \big/ (\max_r S_{St_{i,j},r}) \le P_{i,j} \big/ S_{St_{i,j},r} \quad (۲۶)$$
$$\Rightarrow LB_j = \sum_{i=1}^{m} \min_r (P_{i,j} \big/ S_{St_{i,j},r})$$
$$\le \sum_{i=1}^{m} (P_{i,j} \big/ S_{St_{i,j},r'}) \le C_{m,j} \le C_{\max}$$

اگر برای تک‌تک کارها این کران را محاسبه شود در نهایت بیشینه آنها کران مبتنی بر کار برای مسأله تحقیق خواهد بود، یعنی:

$$LB_J = \max_j \{\sum_{i=1}^{m} (\min_r (P_{i,j} \big/ S_{St_{i,j},r}))\} \quad (۲۷)$$

برای محاسبه کران مبتنی بر ماشین، کران بدست آمده برای مسأله $FQ_m||C_{\max}$موجود در ادبیات را گسترش داده شد ( اثبات آن در [۲۲] ارائه شده است.) سپس این کران را با افزودن جمله $\max_{o_{i,j}\in o_k}(P_{i,j}) \big/ \max_r (S_{k,r})$ بهبود داده شد که در این جمله $o_k$ مجموعه عملیاتی است که توسط مرحله $k$ام پردازش می‌شوند. این جمله بیانگر این واقعیت است که چون اجازه قطع عملیات وجود ندارد، در نتیجه عملیاتی که دارای بیشترین زمان پردازش است در خوشبینانه‌ترین حالت روی سریعترین ماشین پردازش شود.

$$LB_M = \max_k \{\max\{ \sum_{o_{i,j}\in o_k} P_{i,j} \Big/ \sum_{r=1}^{l_k} S_{k,r} \;, \quad (۲۸)$$
$$\max_{o_{i,j}\in o_k}(P_{i,j}) \Big/ \max_r(S_{k,r})\}\}$$

در نتیجه کران پایین مسأله به شکل زیر خواهد بود.

$$LB = \max\{LB_J, LB_M\} \quad (۲۹)$$

## ۵. طراحی آزمایش‌های عددی

### ۵-۱. روشهای مورد مقایسه

به خاطر اینکه مسأله مورد نظر در ادبیات مورد مطالعه قرار نگرفته است به منظور ارزیابی الگوریتمهای H1 و H2 سه الگوریتم ابتکاری دیگر و همچنین یک کران پایین برای مساله توسعه داده شد. در این بخش توانایی هر یک الگوریتمهی ابتکاری ارائه شده نسبت به هم مورد مقایسه قرار می‌گیرد. برای مقایسه الگوریتمها از میانگین جوابها(Mean) ، بهترین جوابها(Best) و انحراف معیار(s.d.) پارامتر (ml) استفاده شده است که از اجرای مثالهای تصادفی ایجاد شده بدست آمده‌اند.

$$ml = \frac{(makespan - lowerbound)}{lowerbound}$$

### ۵-۲. تولید مسائل تصادفی

برای تولید مسائل تصادفی نخست پنج پارامتر از مسأله شناسایی گردید. توزیع چهار پارامتر از کار مشابه‌ای که در زمینه تولید جریانی مختلط انجام شده بود مشخص شد و توزیع پارامتر پنجم نیز به شرح زیر تعیین گردید.

تعداد کارها، تعداد مراحل پردازش، تعداد ماشینهای موجود در هر مرحله و زمان پردازش هر عملیات از مسأله مشابه [۱۸] موجود در ادبیات گرفته شده است. که در آن تعداد کارها برابر20,50,100,150,200,300 و تعداد مراحل پردازش برابر 2,4,6,8,10 است. همچنین تعداد ماشینهای موجود در هر مرحله دارای توزیع گسسته یکنواخت بین ۱ و ۵ ( $U[1,5]$) می‌باشد و زمان پردازش در نظر گرفته شده برای هر عملیات دارای توزیع گسسته یکنواخت بین ۱ و ۴۰ برابر مجموع سرعت ماشینهایی است که قابلیت پردازش عملیات مزبور را دارند (k مرحله‌ایست که عملیات مورد نظر در آن پردازش خواهد شد.

($U[1,40 \times \sum_{r=1}^{l_k} S_{k,r}]$) است. سرعت ماشینها نیز دارای توزیع گسسته یکنواخت بین ۱و۳ ( $U[1,3]$ ) در نظر گرفته شده است.

### ۵-۳. طراحی روش انجام آزمایش‌ها

حالات مختلفی که می‌توان برای تولید نمونه‌های تصادفی در نظر گرفت برابر $6\times5\times1\times1\times1=30$ حالت است که از هر اندازه مسأله ۱۰ نمونه تصادفی تولید شده و در یک فایل باینری ذخیره شدند تا با هر یک از الگوریتمهای معرفی شده اجرا شوند. برنامه

همه الگوریتمها به زبان C کد شده و با یک کامپیوتر شخصی ( CPU 1100 MHz , RAM 128 M) Pentium III شدند.
نتایج بدست آمده از اجرای الگوریتمهایH5, H4, H3, H2, H1 در شکلهای ۷،۸،۶ و جدول۲ آورده شده است. ستون اول و دوم و سوم جدول ۲ به ترتیب شماره مسأله، تعداد کارها و تعداد مراحل را در هر نمونه از مسائل نشان می‌دهد.
ستونهای چهار، پنج و شش به ترتیب میانگین جوابها(Mean) ، بهترین جوابها (Best) و انحراف معیار(s.d.) ده بار اجرای هر مسأله را با الگوریتم H1 نشان می‌دهد. به همین ترتیب ستونهای هفت تا هجده نیز به ترتیب میانگین جوابها، بهترین جوابها و انحراف معیار ده بار اجرای هراندازه از مسأله را با الگوریتم‌های H5, H4, H3, H2 نشان می‌دهد

## ۶. نتایج آزمایش‌ها

برای تجزیه وتحلیل الگوریتمها، ۱۰ نمونه تصادفی ذخیره شده در فایل باینری برای هر یک از حالات اجرا شده و آماره‌های میانگین جوابهای بدست آمده (*Mean*)، بهترین جواب بدست آمده (*Best*) و انحراف معیار (*s.d.*) پارامتر (*ml*) که انحراف نسبی از جوابهای بدست آمده است محاسبه شده است.
با توجه به داده‌های جدول۲ و شکلهای ۶، ۷ و ۸ مشاهده می‌شود هنگامی که تعداد کارها (۲۰،۵۰ و ۱۰۰ کم) باشد، الگوریتم H2 به طور نسبی از الگوریتم H1 کاراتر است ولی هنگامی که تعداد کارها (۲۰۰،۱۵۰و۳۰۰) بیشتر شود و تعداد مراحل پردازش کمتر (۲و۴و۶)، الگوریتم H1 بهتر از H2 جواب می‌دهد.

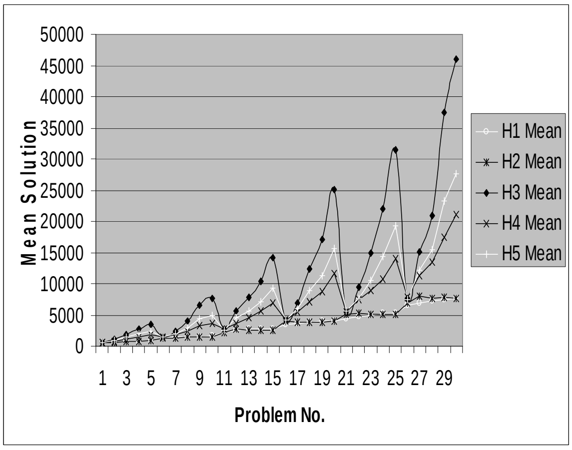
**جدول۲. مقایسه آماره‌های $C_{\max}$ الگوریتم‌ها**

| Problem No. | Job | Stage | H1 | | | H2 | | | H3 | | | H4 | | | H5 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Mean | Best | s.d. | Mean | Best | s.d. | Mean | Best | s.d. | Mean | Best | s.d. | Mean | Best | s.d. |
| 1 | 20 | 2 | 525.72 | 473.67 | 0.008 | 482.92 | 434 | 0.0339 | 611.18 | 531.33 | 0.0115 | 572.55 | 485 | 0.0134 | 574.7 | 435 | 0.0264 |
| 2 | 20 | 4 | 595.37 | 529.17 | 0.066 | 566.63 | 477.5 | 0.0069 | 1167.38 | 872 | 0.1753 | 774.53 | 632.83 | 0.0186 | 940.45 | 644.67 | 0.0888 |
| 3 | 20 | 6 | 742 | 648.17 | 0.0478 | 645.17 | 578.5 | 0.0352 | 1785.72 | 1388.67 | 0.2395 | 1050.28 | 864.33 | 0.1411 | 1465.75 | 1081.17 | 0.2 |
| 4 | 20 | 8 | 812.83 | 740.5 | 0.0238 | 768.35 | 713.67 | 0.1049 | 2782.03 | 1699.5 | 0.8464 | 1435.73 | 1140.17 | 0.304 | 2010.8 | 1443.67 | 0.4931 |
| 5 | 20 | 10 | 941.1 | 866.83 | 0.0203 | 860.32 | 766.83 | 0.0077 | 3367.12 | 2504.5 | 0.4721 | 1799.95 | 1476.33 | 0.0033 | 2436.65 | 1748.33 | 0.2303 |
| 6 | 50 | 2 | 1202.95 | 1010.33 | 0.0143 | 1233.18 | 1068 | 0.02 | 1401.88 | 1208.5 | 0.023 | 1380.05 | 1160 | 0.0044 | 1376.87 | 1234.5 | 0.0209 |
| 7 | 50 | 4 | 1363.97 | 1239.67 | 0.0226 | 1254.57 | 1120.33 | 0.021 | 2325.48 | 1978.33 | 0.0156 | 1807 | 1652.17 | 0.0068 | 1974.25 | 1773.33 | 0.0039 |
| 8 | 50 | 6 | 1467.73 | 1328.67 | 0.0001 | 1411.32 | 1242.33 | 0.0363 | 3964.77 | 3080.83 | 0.1816 | 2399.35 | 1985.83 | 0.0404 | 2910.57 | 2369.5 | 0.1956 |
| 9 | 50 | 8 | 1543.83 | 1381.17 | 0.0163 | 1485.63 | 1400 | 0.0465 | 6508.85 | 4407.17 | 0.888 | 3190.8 | 2560.17 | 0.2634 | 4405.67 | 3546.5 | 0.3524 |
| 10 | 50 | 10 | 1694.78 | 1571.33 | 0.0487 | 1517.4 | 1423.5 | 0.0254 | 7567.38 | 5236.83 | 0.0764 | 3577.85 | 2889.83 | 0.0574 | 4989.08 | 3782 | 0.0006 |
| 11 | 100 | 2 | 2217.9 | 2037 | 0.0043 | 2234.22 | 1969.5 | 0.0079 | 2671.3 | 2398.33 | 0.0209 | 2719.48 | 2534.5 | 0.0014 | 2532.28 | 2187.5 | 0.0339 |
| 12 | 100 | 4 | 2465.05 | 2364.33 | 0.0114 | 2767.7 | 2625.33 | 0.0034 | 5578.45 | 4335.67 | 0.0838 | 3726.2 | 3570 | 0.0082 | 4460.15 | 3605.5 | 0.0738 |
| 13 | 100 | 6 | 2654.95 | 2428.5 | 0.0183 | 2541.42 | 2356 | 0.0494 | 7754.33 | 6019.5 | 0.6826 | 4485.43 | 4032.67 | 0.079 | 5705.83 | 4524.5 | 0.528 |
| 14 | 100 | 8 | 2723.35 | 2550.33 | 0.0006 | 2613.1 | 2407 | 0.0173 | 10444.05 | 7299.83 | 0.2533 | 5649.15 | 4861.17 | 0.0667 | 7162.45 | 5447.83 | 0.1547 |
| 15 | 100 | 10 | 2805.25 | 2686.83 | 0.0306 | 2600.8 | 2389 | 0.0146 | 14218.48 | 10685.17 | 0.3704 | 6957.67 | 5888.5 | 0.0161 | 9357.05 | 7590.17 | 0.091 |
| 16 | 150 | 2 | 3374.93 | 3136.5 | 0.0103 | 3922.68 | 3245.33 | 0.03 | 4138.47 | 3763.17 | 0.0267 | 4151.17 | 3650.5 | 0.0173 | 4164.17 | 3692 | 0.0214 |
| 17 | 150 | 4 | 3584.37 | 3386 | 0.0016 | 3864.37 | 3350 | 0.0092 | 6887.32 | 5431 | 0.0581 | 5373.43 | 5001.67 | 0.0002 | 5921.98 | 4396.5 | 0.0085 |
| 18 | 150 | 6 | 3757.42 | 3651.5 | 0.011 | 3886.52 | 3601 | 0.0159 | 12437.83 | 10099.17 | 0.2692 | 7020.42 | 6565.67 | 0.0101 | 8900.57 | 7014.83 | 0.1401 |
| 19 | 150 | 8 | 3958.98 | 3900.83 | 0.0125 | 3898.38 | 3733.5 | 0.0019 | 17087.93 | 11534.33 | 0.5156 | 8702.37 | 7397.83 | 0.099 | 11361.17 | 8347 | 0.2286 |
| 20 | 150 | 10 | 4071.93 | 3881.33 | 0.0054 | 4002.3 | 3742 | 0.0225 | 25136.78 | 18414.33 | 0.0504 | 11587.95 | 9742 | 0.0028 | 15611.2 | 12152.83 | 0.0057 |
| 21 | 200 | 2 | 4421.73 | 4322.17 | 0.0074 | 5077.5 | 4056 | 0.038 | 5458.52 | 4827.67 | 0.0358 | 5436.65 | 4944.33 | 0.022 | 5550.13 | 4576.33 | 0.0352 |
| 22 | 200 | 4 | 4750.22 | 4506.83 | 0.0037 | 5240.85 | 4501 | 0.0203 | 9401.8 | 7526 | 0.0207 | 7393.7 | 6684.67 | 0.0488 | 7711.33 | 6252.17 | 0.0446 |
| 23 | 200 | 6 | 4971.75 | 4695.33 | 0.0061 | 5116.23 | 4560.33 | 0.0041 | 14826.27 | 11792.83 | 0.1753 | 8971.07 | 8294.17 | 0.001 | 10466.07 | 8775.5 | 0.0537 |
| 24 | 200 | 8 | 5196.97 | 4889.33 | 0.0024 | 5058.57 | 4670.67 | 0.0247 | 21917.6 | 14635.17 | 0.5235 | 10795.12 | 9358.33 | 0.086 | 14407.1 | 9952.33 | 0.3196 |
| 25 | 200 | 10 | 5317.92 | 5044.83 | 0.0011 | 5147.22 | 4900.67 | 0.0239 | 31408.73 | 17897.33 | 0.5772 | 14026.15 | 10922.5 | 0.1906 | 19284.42 | 13421.83 | 0.0076 |
| 26 | 300 | 2 | 6607.8 | 6295.5 | 0.0007 | 6896.67 | 6010 | 0.0363 | 7652.2 | 7135.67 | 0.0226 | 7879.13 | 7067.5 | 0.032 | 7506.37 | 6776.67 | 0.0123 |
| 27 | 300 | 4 | 6917.08 | 6606.67 | 0.0043 | 7979.82 | 6987.5 | 0.0045 | 15038.3 | 10964.67 | 0.0144 | 11190.3 | 10078.33 | 0.0182 | 12133.62 | 9430.83 | 0.0026 |
| 28 | 300 | 6 | 7283.67 | 7025 | 0.0076 | 7652.38 | 7053 | 0.0101 | 20990.86 | 15976 | 0.1965 | 13511.55 | 12019.17 | 0.0752 | 15525.53 | 12860.5 | 0.1292 |
| 29 | 300 | 8 | 7439.83 | 7228.17 | 0.0043 | 7840.67 | 7322.33 | 0.0262 | 37534.49 | 28610.84 | 0.0143 | 17511.63 | 16440.84 | 0.0168 | 23362.95 | 19267.16 | 0.0414 |
| 30 | 300 | 10 | 7634.75 | 7409.17 | 0.0072 | 7603.1 | 7060.33 | 0.0129 | 46082.6 | 27314.5 | 0.9014 | 21086.71 | 16177.33 | 0.2262 | 27727.12 | 20798.67 | 0.3214 |

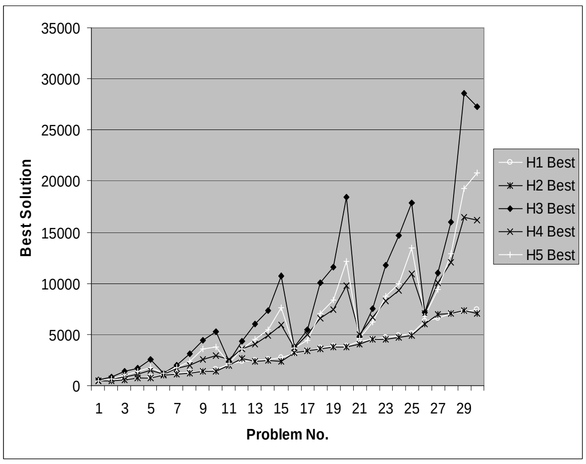
با توجه به داده‌های بدست آمده از جدول ۲، هنگامی که تعداد کارها بسیار زیاد باشد الگوریتم H1 بطور مجانبی از الگوریتم H2 کاراتر است. با توجه به نمودارها مشاهده می‌شود که کارایی سه الگوریتم دیگر به مراتب از دو الگوریتم H2 و H1 بدتر است و H3 بدترین کارایی و سپس به ترتیب H5 و H4 بدترین کارایی را بین این پنج الگوریتم دارند.

همچنین با افزایش مراحل پردازش اختلاف بین جوابهای بین الگوریتمها به شدت افزایش پیدا می‌کند ولی هنگامی که تعداد مراحل کم باشد اختلاف فاحشی میان جوابهای بین الگوریتمها وجود ندارد.

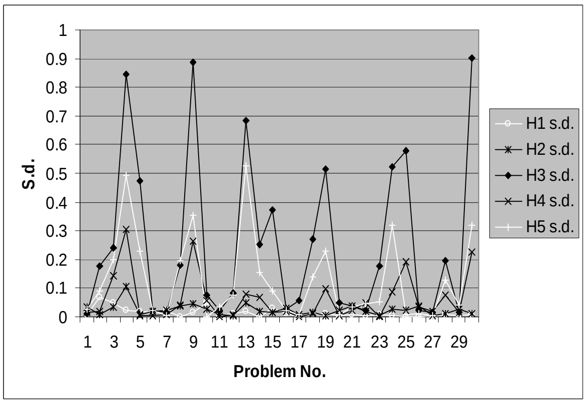
شکل ۷ نشان می‌دهد که ۸۰ درصد بهترین جوابهای بدست آمده متعلق به الگوریتم H2 است و ۲۰ درصد بقیه مربوط به الگوریتم H1 می‌باشد. نتایج تلفیقی شکلهای ۶ و ۷ را می‌توان در شکل ۸ مشاهده کرد که انحراف نسبی جوابهای بدست آمده در الگوریتم H1 در اکثر موارد پایین‌تر از انحراف نسبی بدست آمده از الگوریتم H2 است که حاکی از نزدیکی نسبی جوابهای هر گروه به میانگین بدست آمده است.




**شکل ۶. مقایسه میانگین جوابهای بدست آمده از الگوریتمها**




**شکل ۷. مقایسه بهترین جوابهای بدست آمده از الگوریتمها**




**شکل ۸. مقایسه انحراف معیارهای ( پارامتر ml ) بدست آمده از الگوریتم‌ها**

## ۷. جمع‌بندی و نتیجه‌گیری

در این مقاله یک انعطاف‌پذیری جدید در مسأله زمانبندی تولید کارگاهی (JS) که از ترکیب مساله تولید کارگاهی و ماشینهای موازی بدست می‌آید، ارائه گردید. پس از نشان دادن NP-hard بودن مساله از نظر زمان حل، برای حل این مسأله دو الگوریتم ابتکاری H1 و H2 به ترتیب برای مسائل با ابعاد بزرگ و معمولی ارائه گردید. بعلت جدید بودن مساله در ادبیات موضوع روش حلی برای مساله مذکور ارائه نشده است و معیار مناسبی برای ارزیابی الگوریتمهای ارائه شده وجود ندارد. بنابراین الگوریتمهای H3، H4 و H5 (بر پایه قواعد زمانبندی) برای حل مساله ارائه گردید که با الگوریتمهای H1 و H2 مقایسه شده‌اند. نتایج محاسبات نشان داد که برای تعداد کارهای کم (کمتر از ۱۰۰) الگوریتم H2 به طور نسبی از سایر الگوریتمها کاراتر است ولی هنگامی که تعداد کارها بسیار زیاد باشد، الگوریتم H1 به طور مجانبی کاراتر از H2 است. همچنین کارائی الگوریتم H3 به مراتب بدتر از سایر الگوریتمها می‌باشد.

همچنین بمنظور ارزیابی بیشتر الگوریتمهای ارائه شده یک کران پایین نیز برای مساله توسعه داده شده است و نتایج الگوریتمهای ارائه شده با آن مقایسه شده‌است.

حل مسأله با استفاده از روشهای فرا ابتکاری یا ارائه روشهای حل جدید و در نظر گرفتن سایر انعطاف‌پذیریها در کنار ماشینهای موازی در تولید کارگاهی می‌تواند به عنوان تحقیقات آتی در نظر گرفته شود.

## مراجع

[1] Zhou, H., Feng, Y., Han, L., "*The Hybrid Heuristic Genetic Algorithm for Job Shop Scheduling*", Computers & Industrial Engineering, Vol. 40, 2001, pp. 191-200.

[2] Xia, W., Wu, Z., "*An Effective Hybrid Optimization Approach for Multi-Objective Flexible Job-Shop Scheduling Problems*", Computers & Industrial Engineering, Vol. 48, 2005, pp. 409-425.

[3] Baker, K.R., Introduction to Sequencing and Scheduling, John Wiley, New York, 1974.

[4] Brucker, P., Schlie, R., "*Job-Shop Scheduling with Multipurpose Machines",* Computing, Vol. 45, 1990, pp. 369–375.

[5] Hurink, J., Jurisch, B., Thole, M., *"Tabu Search for the Job Shop Scheduling Problem with Multi-Purpose Machines"*, OR-Spektrum, Vol. 15, 1994, pp. 205–215.

[6] Mastrolilli, M., Gambardella, L.M., *"Effective Neighbourhood for the Flexible Job-Shop Problem*", Journal of Scheduling, Vol. 3, 2000, pp. 3–20.

[7] Ghedjati, F., "*Genetic Algorithms for the Job-Shop Scheduling Problem with Unrelated Parallel Constraints: Heuristic Mixing Method Machines and Precedence*", Computer & Industrial Engineering, Vol. 37, 1999, pp. 39-42.

[8] Kacem, I., Hammadi, S., Borne, P., *"Pareto-Optimality Approach for Flexible Job-Shop Scheduling Problems: Hybridization of eVolutionary Algorithms and Fuzzy Logic",* Mathematics and Computers in Simulation, Vol. 60, 2002, pp. 245-276.

[9] Lee, Y.H., Jeong, C.S., Moon, C., *"Advanced Planning and Scheduling with Outsourcing in Manufacturing Supply Chain*", Computer & Industrial Engineering, Vol. 43, 2002, pp. 351-374.

[10] Park, B.J., Choi, H.R., Kim, H.S., *"A Hybrid Genetic Algorithm for the Job Shop Scheduling Problems*", Computers & Industrial Engineering, Vol. 45 , 2003, PP. 597–613.

[11] Brucker, P., Neyer, J., "*Tabu Search for the Multi-Mode Jobshop Problem",* OR-Spektrum, Vol. 20, 1998, pp. 21–28.

[12] Brucker, P., Drexl, A., Mohring, R., Neumann, K., Pesch, E., *"Resource-Constrained Project Scheduling: Notation, Classification, Models, and Methods*", European Journal of Operational Research, Vol. 112 ,1999, pp. 3-41.

[13] Kim, Y.K., Park, K., Ko, J., *"A Symbiotic eVolutionary Algorithm for the Integration of Process Planning and Job Shop Scheduling*", Computers & Operations Research, Vol. 30, 2003, pp. 1151-1171.

[14] Scrich, C.A., Armentano, V.A., Laguna, M., *"Tardiness Minimization in a Flexible Job Shop: A Tabo Search Approach",* Journal of Intelligent Manufacturing, Vol. 15, 2004, pp. 103-115.

[15] Kyparisis, G.J., Koulamas, C., "*Flexible Flow Shop Scheduling with Uniform Parallel Machines*", European Journal of Operational Research, Vol. 168, 2006, pp. 985-997.

[16] Riane, F., Artiba, A., Elmaghraby, S.E., "*A Hybrid Three-Stage Flowshop Problem: Efficient Heuristics to minimize Makespan*", European Journal of Operational Research, Vol. 109, 1998, pp. 321-329.

[17] Nowicki, E., Smutniciki, C., "*The Flow Shop with Parallel Machines: A Tabu Search Approach*", European Journal of Operational Research, Vol. 106, 1998, pp. 226-253.

[18] Dessouky, M.M., Dessouky, M.I., Verma, S.K., "*Flowshop Scheduling with Identical Jobs and Uniform Parallel Machines*", European Journal of Operational Research, Vol. 109, 1998, pp. 620-631.

[19] Low, C., *"Simulated Annealing Heuristic for Flow Shop Scheduling Problem with Unrelated parallel Machines*", Computer & operation Research, Vol. 32, 2005, pp. 2013-2025.

[20] Kurz, M.E., Askin, R.G., *"Comparing Scheduling Rules for Flexible Flow Lines*", Int. J. Production Economics, Vol. 85, 2003, pp. 371-388.

[21] Kurz, M.E., Askin, R.G., *"Scheduling Flexible Flow Lines with Sequencing-Dependent Setup Times*", European Journal of Operational Research, Vol. 159, 2004, pp. 66-82.

[22] Hoogeveen, J.A., van de Velde, S.L., Veltman, B., *"Complexity of Scheduling Multiprocessor Tasks with Prespecified Processor Allocations*", Discrete Applied Mathematics, Vol. 55, 1994, pp. 259–272.

[23] Brandimarte, P., *"Routing and Scheduling in a Flexible Jobshop by Tabu Search*", Annals of Operations Research, Vol. 41, 1993, pp. 157–183.