



A Bi-Criterion Hybrid Flow Shop Time Scheduling: Balancing the Performance and Total Completion Times

Y. Zare Mehrjerdi^{*}, A.A. Ghasemi Gajvan & M. Shahmohammadi

Yahia Zare Mehrjerdi, Associate professor of Industrial Engineering, Yazd University

Ali Akbar Ghasemi Gajvan, Industrial Engineering, Yazd University, Ali_Gajvan464@yahoo.com

Mohsen Shahmohammadi, Industrial Engineering, Yazd University, Mohsen.e.Shahmohammadi@gmail.com

Keywords

Hybrid flow shop,
Genetic algorithm,
Bicriterion Programming

ABSTRACT

Today, the flow shop environment is the most customary production area in industry. But, the need to increase the capacity or the capacity balance in different production processes has caused that in some working stations there exists more than one machine. This type of flow shops are usually named hybrid flow shop, multi processes flow shop, flexible flow shop or flow shop with parallel machines. In this article, the time scheduling of hybrid flow shop with the assumption of machine similarity is studied using a bi-criterion programming problem. Taking the fact that the proposed problem is an NP-hard type problem a genetic algorithm is developed and a sample problem is solved for illustration purposes.

^{*} **Corresponding author. Yahia Zare Mehrjerdi**
Email: Yazm2000@yahoo.com



موازنه دو معیار عملکردی دامنه عملیات و مجموع زمانهای تکمیل در مسئله زمانبندی "جریان کارکردی" ترکیبی

یحیی زارع مهرجردی^{*}، علی‌اکبر قاسمی گجوان و محسن شاه‌محمدی

کلمات کلیدی

جریان کارکردی ترکیبی،
الگوریتم ژنتیک،
دامنه عملیات،
مجموع زمانهای تکمیل

چکیده:

امروزه محیط جریان کارکردی، یکی از رایج‌ترین محیط‌های تولیدی در صنعت می‌باشد. لیکن نیاز به افزایش ظرفیت و یا بالانس ظرفیت در مراحل مختلف تولیدی، سبب شده است که در بعضی از ایستگاه‌های کاری، بیش از یک ماشین وجود داشته باشد. از این محیط‌های توسعه یافته معمولاً به نام‌های جریان کارکردی ترکیبی، جریان کارکردی چند پردازنده، جریان کارکردی انعطاف‌پذیر و یا جریان کارکردی با ماشینهای موازی یاد می‌شود. در این مقاله، مسئله زمانبندی جریان کارکردی ترکیبی با فرض همانند بودن ماشینها، تحت دو هدف دامنه عملیات و مجموع زمانهای تکمیل مورد بررسی قرار گرفته است. با توجه به NP-hard بودن مسئله، برای حل از الگوریتم ژنتیک استفاده شده است.

۱. مقدمه

زمانبندی^۱ کارها نوعی تصمیم‌گیری است که نقش تعیین‌کننده‌ای را در واحدهای صنعتی و خدماتی بازی می‌کند و عبارت است از تخصیص منابع به کارها، در طول یک دوره زمانی معین که بهینه‌سازی یک یا چند هدف را دنبال می‌کند [۱]. این مقاله در ارتباط با مسائل زمانبندی صنعتی است. این مسائل با توجه به تعداد، نحوه آرایش و خصوصیات ماشینها به چندین دسته تقسیم می‌شوند که در این مقاله مسئله زمانبندی در محیط جریان کارکردی ترکیبی^۲ مورد بحث قرار گرفته است.

در مسائل زمانبندی جریان کارکردی کلاسیک، مجموعه‌ای از کارها که دارای چندین عملیات می‌باشند، با عبور از میان تعدادی مراحل، در یک مسیر یکسان، مورد پردازش قرار می‌گیرند. در این حالت فرض می‌شود که در هر مرحله تنها یک

ماشین وجود دارد. امروزه در برخی شرکتها و کارخانجات، نیاز به افزایش ظرفیت و یا بالانس نمودن ظرفیت مراحل مختلف، سبب شده است که در بعضی مراحل بیش از یک ماشین وجود داشته باشد. از این محیط‌های توسعه یافته معمولاً به نام‌های جریان کارکردی ترکیبی، جریان کارکردی چند پردازنده^۳، جریان کارکردی انعطاف‌پذیر^۴ و یا جریان کارکردی با ماشینهای موازی^۵ یاد می‌شود (شکل ۱). مسئله زمانبندی جریان کارکردی ترکیبی را می‌توان حالت تعمیم یافته‌ای از دو نوع مسئله زمانبندی خاص قلمداد کرد: مسئله زمانبندی ماشینهای موازی^۶ و مسئله زمانبندی جریان کارکردی^۷.

مسائل زمانبندی جریان کارکردی ترکیبی از جهات مختلفی مورد تقسیم‌بندی قرار گرفته‌اند ولی عمده‌ترین تقسیم‌بندی، مربوط به مشخصات ماشینها در این نوع مسائل است. در این زمینه مسائل را به سه گروه تقسیم می‌کنند:

- جریان کارکردی ترکیبی با ماشینهای کاملاً یکسان^۸: زمان پردازش هر کار در هر مرحله، به ماشینی که به آن اختصاص داده می‌شود، وابسته نیست.

تاریخ وصول: ۹۱/۹/۱۰

تاریخ تصویب: ۹۲/۱۲/۱

^{*}نویسنده مسئول مقاله: دکتر یحیی زارع مهرجردی، دانشیار دانشکده

مهندسی صنایع، دانشگاه یزد، Yazm2000@yahoo.com

علی‌اکبر قاسمی گجوان، دانشکده مهندسی صنایع، دانشگاه یزد،

Ali_Gajvan464@yahoo.com

محسن شاه‌محمدی، دانشکده مهندسی صنایع، دانشگاه یزد،

Mohsen.e.Shahmohammadi@gmail.com

2. Scheduling

3. Hybrid Flow Shop (HFS)

4. Multi-processor Flow Shop

5. Flexible Flow Shop

6. Flow Shop with Parallel machines

7. Parallel Machine Scheduling (PMS)

8. Flow Shop Scheduling (FSS)

9. Identical

الگوریتم ژنتیک نتیجه گرفته اند که الگوریتم ژنتیک سه بعدی ارائه شده، جوابهای بهتری را تولید کرده است [۱۵].

سووندی و المغربی مسئله را در حالت دو مرحله ای و با هدف مینیمم سازی دامنه عملیات، بررسی نموده اند. آنها سه الگوریتم ابتکاری برای مسئله فوق ارائه داده اند که یکی از آنها در حالت مستقل بودن زمان شروع پردازش در مرحله دوم، از زمان تکمیل کار در مرحله اول و دو تای دیگر در حالت وابسته بودن، مسئله را مورد بررسی قرار داده اند.

همچنین با توسعه چندین حد پایین، عملکرد الگوریتم ها را در دو حالت متقارن بودن و نامتقارن بودن زمانهای پردازش بررسی نموده و بهترین آنها را مشخص نموده اند [۱۶]. کپیاریسیس و کولاماس با بررسی الگوریتمی که توسط سووندی و المغربی برای مسئله جریان کارکردی ترکیبی دو مرحله ای ارائه کرده اند، نشان دادند که حد بدترین حالتی که توسط سووندی ارائه شده، برای حالتی که سرعت ماشینها حدوداً برابر است، مطلوب بوده و وقتی اختلاف سرعت ماشینها زیاد باشد، کارکرد مطلوبی ندارد. لذا آنها یک حد بدترین حالت، برای برطرف کردن این مشکل ارائه کرده اند [۱۷].

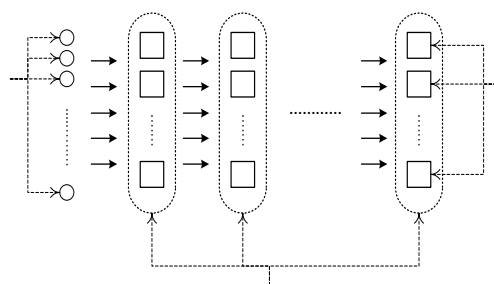
سواستیانف در مقاله خود، که هدف کمینه سازی دامنه عملیات بوده است، با استفاده از تکنیک جمع برداری سه الگوریتم ابتکاری چند جمله ای توسعه داده است. وی نشان داده است که الگوریتمهای توسعه داده شده در مقاله او، برای حالت جریان کارکردی معمولی بهترین الگوریتمهای تخمینی هستند [۱۸]. ریان و آرتیا یکی از ساده ترین حالات جریان کارکردی ترکیبی را که در آن، مرحله اول دارای یک ماشین و در مرحله دوم دو ماشین همانند وجود دارد، با هدف کمینه سازی دامنه عملیات، مورد بررسی قرار داده اند.

آنها ضمن اثبات این مطلب که مسئله مورد نظر از نوع NP-Complete است و همچنین فرموله کردن مسئله بصورت یک برنامه ریزی پویا، الگوریتمی ابتکاری را که الهام گرفته از قاعده جانسون است را ارائه نموده اند [۱۹]. دسوکی و همکارانش با فرض یکسان بودن کارها، مسئله را با هدف کمینه سازی دامنه عملیات، مورد بررسی قرار داده اند. آنها برای حالتی که مسئله دارای سه مرحله باشد، یک الگوریتم انشعاب و تحدید ارائه نموده و همچنین برای مسئله کلی، الگوریتمی ابتکاری توسعه داده اند [۲۰]. هانگ و لی مسئله را در حالت دو مرحله ای بودن و در یک محیط تکنولوژی گروهی، با هدف کمینه سازی دامنه عملیات، مورد بررسی قرار داده است. آنها مسئله را با فرض وجود تنها یک ماشین در مرحله اول و وجود چندین ماشین در مرحله دوم و همچنین گروهبندی بودن کارها و محدودیت زمان آماده سازی برای هر گروه، مورد مطالعه قرار داده اند. آنها برای حل این

جریان کارکردی ترکیبی با ماشینهای همانند^۱: زمان پردازش هر کار در هر مرحله، به ماشینی که به آن اختصاص داده می شود، وابسته است.

جریان کارکردی ترکیبی با ماشینهای نامرتبط^۲: علاوه بر وابسته بودن زمان پردازش هر کار به ماشینی که به آن اختصاص داده می شود، سرعت هر ماشین نیز به کاری که به آن تخصیص داده می شود وابسته است.

اولین مقاله در زمینه زمانبندی جریان کارکردی ترکیبی در سال ۱۹۷۰ ارائه شد [۳] و با گذشت زمان علاقه به مطالعه پیرامون این نوع مسئله در حال افزایش بوده و در این راستا مقالات بسیاری به رشته تحریر درآمده است. با این وجود در بسیاری از این مقالات، مسئله زمانبندی جریان کارکردی ترکیبی با فرض کاملاً یکسان بودن ماشینها، مورد بررسی قرار گرفته است [۴ تا ۸]. روئیز و رودریگوئز در پژوهشی با بررسی بیش از ۲۰۰ مقاله در این زمینه، بیان داشته اند که از مجموع مقالات مورد بررسی، ۸۴ درصد آنها با فرض کاملاً یکسان بودن ماشینها بوده اند [۹]. همچنین بررسی مسئله زمانبندی جریان کارکردی ترکیبی، با فرض نامرتبط بودن ماشینها نیز در سالهای اخیر بیشتر مورد توجه قرار گرفته و مقالاتی در این زمینه به چاپ رسیده است [۱۰ تا ۱۴]. از بین سه گروهی که پیش از این تشریح شد، گروه دوم، یعنی جریان کارکردی ترکیبی با ماشینهای همانند، کمتر مورد توجه قرار گرفته است. به طوری که در پژوهش روئیز و رودریگوئز درصد مقالاتی که در این زمینه بوده اند، تنها ۴ درصد ذکر شده است [۹].



شکل ۱. محیط جریان کارکردی ترکیبی

امین ناصری و بهشتی این مسأله را در حالتی که تولید به صورت دسته ای است، با هدف کمینه سازی دامنه عملیات، مورد مطالعه قرار داده اند. آنها در این راستا چهار الگوریتم ابتکاری H1، H2، H3 و H4 را توسعه و در نهایت یک الگوریتم ژنتیک سه بعدی ارائه داده اند و با مقایسه نتایج الگوریتم های ابتکاری با نتایج

1. Uniform
2. Unrelated

$$\min \left[\alpha \sum C_j + (1-\alpha) C_{\max} \right] \quad (۱)$$

$$\min \sum C_j + w C_{\max} \quad (۲)$$

در دو روش فوق با در نظر گرفتن یک ضریب، به عنوان درجه اهمیت هر یک از معیارهای عملکردی، مسئله حل می‌گردد. در این گونه روشها احتمال دارد جواب بهینه باعث مینیمم شدن یکی از معیارها گردد ولی برای معیار دیگر جواب چندان خوبی حاصل نشود. این مسئله در مواقعی که برای یکی از معیارها یک حداکثر مقدار در نظر باشد، مطلوب نیست. در این گونه موارد بهتر است یکی از معیارها به عنوان یک محدودیت در نظر گرفته شده و مسئله را با توجه به معیار دیگر بهینه نماییم. در زمینه مسئله مورد بررسی می‌توان به یکی از روشهای زیر عمل نمود:

$$\begin{aligned} \min C_{\max} \\ \text{s.t. } \sum C_j \leq nF \end{aligned} \quad (۳)$$

$$\begin{aligned} \min \sum C_j \\ \text{s.t. } C_{\max} \leq D \end{aligned} \quad (۴)$$

$$\begin{aligned} \min C_{\max} \\ \text{s.t. } \sum C_j = \sum C_j^* \end{aligned} \quad (۵)$$

$$\begin{aligned} \min \sum C_j \\ \text{s.t. } C_{\max} = C_{\max}^* \end{aligned} \quad (۶)$$

در روش اول فرض می‌شود برای مسئله مورد بررسی، مقداری مانند F به عنوان حداکثر مقدار میانگین زمان در جریان کارها، در نظر بوده و با توجه به آن قصد مینیمم سازی دامنه عملیات را داریم. همچنین در حالت دوم مقداری مانند D به عنوان یک ضرب العجل در نظر بوده و قصد مینیمم سازی مجموع زمانهای تکمیل را داریم. حالات سوم و چهارم نیز حالت خاصی از دو حالت اول هستند که مقدار بهینه یک هدف به عنوان محدودیت آن هدف در نظر گرفته شده است.

با توجه به اینکه مسئله زمانبندی جریان کارکردی ترکیبی با ماشین‌های همانند و هدف کمینه سازی دامنه عملیات $(C_{\max} || FQ_m)$ شدیداً NP-Hard می‌باشد [۱۶]، لذا مسئله‌ای که در این مقاله مورد بررسی قرار خواهد گرفت $[(FQ_m | D | \sum C_j) \& (FQ_m | F | C_{\max})]$ نیز شدیداً NP-Hard می‌باشد. لذا حل دقیق مسئله به علت پیچیدگی و زمانبر

مسئله ۲ الگوریتم ابتکاری و ۸ قاعده برای تعیین توالی کارها توسعه داده اند [۲۱].

۱-۱. شرح مسئله

تعداد n کار وجود دارد ($j = 1, 2, \dots, n$) که با گذشتن از m مرحله ($s = 1, 2, \dots, m$) مورد پردازش قرار می‌گیرند. در هر مرحله s ، تعداد L_s ماشین وجود داشته ($k = 1, 2, \dots, L_s$) و مسیر فرآیند برای همه کارها مشابه است.

هر کار در هر مرحله تنها توسط یک ماشین مورد پردازش قرار گرفته و هر ماشین نیز در هر لحظه تنها یک کار را مورد پردازش قرار می‌دهد. زمان آماده سازی ماشین، زمان حمل و نقل بین مراحل و زمان دسترسی برای همه کارها صفر فرض شده و هیچگونه وقفه‌ای جایز نمی‌باشد.

بعلاوه فرض شده که ماشینها در هر مرحله همانند هستند، یعنی ماشینها یکسان و با سرعتهای متفاوت می‌باشند. V_{ks} نشان دهنده سرعت k امین ماشین در مرحله s ام می‌باشد. سرعت یک ماشین نشان دهنده مقدار فرآیندی است که در واحد زمان، بر روی یک کار انجام می‌شود. به عبارت دیگر، ماشینی با سرعت V_{ks} می‌تواند V_{ks} واحد از فرآیند را در واحد زمان انجام دهد. این بدین معنا است که: $(p_{jks} = p_{js} / V_{ks})$ که p_{js} نشان دهنده زمان فرآیند کار j ام در مرحله s ام و p_{jks} زمان پردازش کار j ام در مرحله s ام است، اگر توسط ماشین k ام پردازش شود.

در اغلب مقالاتی که در زمینه مسائل زمانبندی ارائه شده است، پژوهشگران تنها یک معیار عملکردی را به عنوان هدف در نظر گرفته اند. اما در واقعیت، حالاتی وجود دارد که تنها پرداختن به یک معیار و عدم دخالت اهداف دیگر، منطقی به نظر نمی‌رسد. در این مقاله، زمانبندی با توجه به دو معیار عملکردی دامنه عملیات (C_{\max}) و مجموع زمانهای تکمیل $(\sum C_j)$ ، صورت پذیرفته است.

دو هدف مذکور به نوعی در تقابل با یکدیگر هستند. بدین صورت که اگر هدف کمینه سازی دامنه عملیات باشد، سعی می‌شود که هیچ کاری زمان تکمیل خیلی طولانی نداشته باشد، هر چند که زمان تکمیل اغلب کارها نسبتاً طولانی باشد. در مقابل اگر هدف کمینه سازی مجموع زمانهای تکمیل باشد، سعی می‌شود که اغلب کارها دارای زمان تکمیل کوتاهتری باشند، هر چند که زمان تکمیل یک کار خیلی طولانی شود. در نتیجه ما به دنبال برنامه زمانبندی هستیم که به طور همزمان برای هر دو هدف مطلوب باشد. در این راستا تابع هدف مسئله را می‌توان به یکی از صورت های زیر در نظر گرفت:

$$\sum_{k=1}^{L_s} \sum_{i=0}^n x_{ijks} = 1, \quad \forall s, j \quad (8)$$

$$\sum_{k=1}^{L_s} \sum_{j=1}^{n+1} x_{ijks} = 1, \quad \forall s, i \quad (9)$$

$$\sum_{i=0}^n x_{i(n+1)ks} = 1, \quad \forall s, k \quad (10)$$

$$\sum_{j=1}^{n+1} x_{0jks} = 1, \quad \forall s, k \quad (11)$$

$$\sum_{i=0}^n x_{ijks} = \sum_{i=1}^{n+1} x_{jiks}, \quad \forall s, k, j \quad (12)$$

$$C_{js} \geq C_{is} + \sum_{k=1}^{L_s} \frac{P_{js}}{V_{ks}} \left(\sum_{i=1}^{n+1} x_{ijks} \right) + \left(\sum_{k=1}^{L_s} x_{ijks} - 1 \right) M, \quad \forall s, i, j; i \neq j \quad (13)$$

$$C_{js} \geq C_{j(s-1)} + \sum_{k=1}^{L_s} \frac{P_{js}}{V_{ks}} \left(\sum_{i=1}^{n+1} x_{ijks} \right), \quad \forall s, j \quad (14)$$

$$C_{js} \geq 0, \quad \forall s, j \quad (15)$$

$$C_{\max} \geq C_{jm}, \quad \forall j \quad (16)$$

$$C_{j0} = 0, \quad \forall j \quad (17)$$

$$x_{ijks} \in \{0, 1\}, \quad \forall i, j, k, s \quad (18)$$

$$x_{iiks} = 0, \quad \forall i, k, s \quad (19)$$

در این مدل رابطه (۱) هدف مسئله را نشان می دهد. محدودیتهای (۲) و (۳) تضمین می کنند که به هر موقعیت از توالی کارها در هر مرحله، تنها یک کار تخصیص یابد. محدودیتهای (۴) و (۵) اطمینان می دهند که در موقعیتهای ابتدایی و انتهایی، در هر توالی از کارها، بر روی هر یک از ماشینهای هر یک از مراحل، تنها یک کار تخصیص داده شود. محدودیت (۶) باعث ایجاد یک توالی نامتناقض و سازگار در هر مرحله می شود. محدودیت (۷) بیانگر این مطلب است که اگر کار i در یک مرحله، به ماشین یکسانی تخصیص داده شوند، زمان

بودن دارای توجیه نمی باشد. لذا در این مقاله جهت حل مسئله، از الگوریتم ژنتیک^۱ استفاده شده است.

هدف اصلی مقاله موازنه دو معیار عملکردی دامنه عملیات و مجموع زمانهای تکمیل در مسئله زمانبندی "جریان کارکردی" ترکیبی با استفاده از روش تصمیم گیری چند هدفه (MCDM) است بطوری که این حوزه تحقیق که کمتر مورد توجه واقع می شود بتواند از نو احیا گردد. در این تحقیق از روش الگوریتم ژنتیک با کروموزوم های یک بعدی و دوبعدی استفاده شده است. این مقاله دارای ساختار زیر است: مساله اصلی در بخش دو فرمولبندی شده است در حالی که الگوریتم ژنتیک موضوع بخش سوم است. نتایج محاسباتی ما در بخش چهار آورده شده است. نتیجه گیری و تحقیقات آتی موضوع بخش پنجم است.

۲. فرمولبندی مسئله

در این بخش مسئله مورد بحث، بصورت یک مدل برنامه ریزی خطی عدد صحیح مختلط^۲ صفر و یک، فرموله می شود.

۱-۲. شرح نمادگذاریها

n : تعداد کارها

m : تعداد مراحل

$i, j = 1, 2, 3, \dots, n$: اندیس کارها

$s = 1, 2, 3, \dots, m$: اندیس مرحله

L_s : تعداد ماشینهای موازی در مرحله s ام

$k = 1, 2, 3, \dots, L_s$: اندیس ماشین ها

V_{ks} : سرعت ماشین k ام در مرحله s ام

P_{js} : زمان پردازش کار j ام در مرحله s ام

M : یک عدد بزرگ که از مدت زمانبندی کوچکتر نیست.

C_{js} : زمان تکمیل کار j ام در مرحله s ام

C_{\max} : بزرگترین زمان تکمیل همه کارها در آخرین مرحله

X_{ijks} : اگر در مرحله s ام، کار j ام بر ماشین k ام و بلافاصله پس

از کار i ام تخصیص و زمانبندی شده باشد، مقدار آن برابر یک و

در غیر این صورت برابر صفر است.

۲-۲. مدلسازی ریاضی

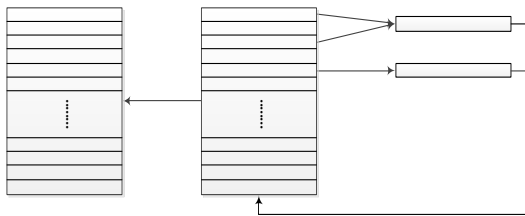
مسئله مورد بررسی را می توان بصورت زیر فرموله نمود:

$$\begin{cases} \text{Min } C_{\max} \\ \sum C_j \leq nF \end{cases} \quad \text{or} \quad \begin{cases} \text{Min } \sum C_j \\ C_{\max} \leq D \end{cases} \quad (7)$$

1. Genetic Algorithm (GA)

2. 0-1 mixed integer linear programming

فرآیند انتخاب، گلچین نمودن کروموزوم های برانزنده در میان والدین و فرزندان است بطوریکه تعداد کروموزومهای منتخب برابر با اندازه جمعیت اولیه باشد. فرآیند انتخاب بر اساس مقدار برانزندی هر کروموزوم صورت می گیرد که این مقدار با استفاده از تابع برانزندی^۷ بدست می آید. این روند تا رسیدن به شرایط توقف مسأله تکرار می شود. شکل ۲ ساختار کلی یک الگوریتم ژنتیک را نشان می دهد.



شکل ۲. ساختار کلی یک الگوریتم ژنتیک

۳-۱. نمایش

اولین گام در بکارگیری و پیاده سازی یک الگوریتم ژنتیک، نمایش جواب های مسأله بصورت یک کروموزوم است. در این مقاله دو روش مختلف از نحوه نمایش جواب ها بکار گرفته شده است.

۳-۱-۱. روش اول (کروموزوم یک بعدی)

یکی از رایج ترین روش های نمایش جواب در حل مسائل جریان کارکردی که بصورت گسترده ای مورد استفاده قرار گرفته است، در نظر گرفتن ترتیب کارها به عنوان یک کروموزوم می باشد. بطور مثال یک کروموزوم برای یک مسأله فرضی با ۶ کار، می تواند بصورت $[6|2|4|3|5|1]$ که معرف ترتیب پردازش کارها می باشد، در نظر گرفته شود. در رابطه با مسأله جریان کارکردی ترکیبی، کروموزوم می تواند به عنوان ترتیب کارها در مرحله اول در نظر گرفته شود.

۳-۱-۲. روش دوم (کروموزوم دو بعدی)

در این روش، کروموزوم را در دو بعد تعریف می کنیم. یک بعد نشانگر مراحل و بعد دیگر نشان دهنده ترتیب کارها در مراحل مختلف می باشد. به طور مثال یک کروموزوم برای یک مسأله فرضی با ۵ کار و ۴ مرحله می تواند به صورت زیر نشان داده شود:

تکمیل کار j که بلافاصله پس از کار i مورد پردازش قرار می گیرد، بایستی حداقل به اندازه زمان پردازش کار j در مرحله مورد نظر بیشتر از زمان تکمیل کار i در آن مرحله باشد. محدودیت (۸) اطمینان می دهد که زمان شروع پردازش هر کار، در هر مرحله، نمی تواند کوچکتر از زمان تکمیلش، در مرحله قبل باشد. محدودیت (۹) شرط نامنفی بودن زمانهای تکمیل کار در هر مرحله را برقرار نموده و محدودیت (۱۰) بیانگر این مطلب است که دامنه عملیات، از هیچ یک از زمانهای تکمیل در مرحله آخر، کوچکتر نمی باشد.

محدودیت (۱۱) نشان دهنده این مطلب است که زمان دسترسی به همه کارها صفر است (همه کارها در زمان صفر در دسترس هستند). محدودیتهای (۱۲) و (۱۳) نیز معرف متغیرهای مسأله می باشند.

مسأله فوق را می توان توسط یکی از نرم افزارهای حل مسائل برنامه ریزی ریاضی حل نمود. البته زمان اجرای مدل، تنها برای مسائلی با ابعاد کوچک (بطور مثال ۷ کار و ۴ مرحله) قابل قبول می باشد و با بزرگ شدن مسأله، زمان اجرا، بسیار طولانی خواهد بود.

۳. الگوریتم ژنتیک

الگوریتم ژنتیک یکی از تکنیکهای جستجو می باشد که از تکامل ژنتیکی و انتخاب طبیعی الهام گرفته شده است. این الگوریتم در طیف گسترده ای از مسائل مورد استفاده قرار گرفته است، بویژه در مسائل بهینه سازی ترکیبی که در مدت زمانی قابل قبول، جوابهای نزدیک به جواب بهینه، ایجاد می کند. الگوریتم ژنتیک برای اولین بار در سال ۱۹۷۵ توسط جان هالند مطرح گردید و پس از آن توسط گلدبرگ توسعه داده شد.

ساختار کلی یک الگوریتم ژنتیک بدین صورت است که پس از مشخص نمودن یک مکانیزم مناسب جهت تبدیل هر جواب مسأله به یک کروموزوم^۱، مجموعه ای از کروموزومها به عنوان یک جمعیت اولیه^۲ (معمولاً بصورت تصادفی) ایجاد می شوند. سپس با استفاده از عملگرهای ژنتیک^۳، اقدام به ایجاد کروموزومهای جدید، موسوم به فرزند^۴ می شود.

اصلی ترین عملگرهای ژنتیک، تقاطع^۵ و جهش^۶ می باشند. بعد از تولید یکسری کروموزوم جدید (فرزند)، بایستی برانزنده ترین کروموزومها جهت ایجاد نسل جدید، انتخاب شوند. عبارتی دیگر

1. Chromosomes
2. Initial population
3. Genetic operators
4. Offspring
5. Cross over
6. Mutation

7. Fitness function

۳-۲-۱. زمانبندی پیشرو

زمانبندی پیشرو برای یک مرحله مانند s ($s=1,2,\dots,m$) به شرح ذیل می باشد [۱۵]:

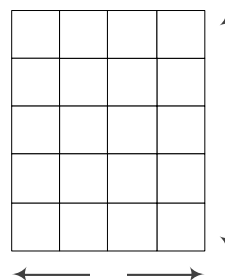
گام ۰: ترتیب کارها در مرحله s را وارد کنید (π_s) و قرار دهید $i=0$

گام ۱: قرار دهید $i=i+1$

گام ۲: کاری که در i امین موقعیت از π_s قرار دارد را به یکی از L_s ماشین موجود در مرحله s ، که توانایی انجام سریعتر این پردازش را دارد، اختصاص دهید.

گام ۳: اگر $i=n$ برو به گام ۴، در غیر اینصورت برو به گام ۱.

گام ۴: زمانبندی در مرحله s تمام شده است.



در نظر گرفتن این شیوه نمایش این مزیت را دارد که هنگام انجام عمل تقاطع، ترتیب کارها در مراحل مختلف، دستخوش تغییر قرار گرفته و تنها به ترتیب کارها در مرحله اول (روش اول نمایش) بسنده نمی شود.

۳-۲-۲. الگوریتم ابتکاری H2

این الگوریتم که توسط امین ناصری و بهشتی نیا [۱۵] ارائه گردیده است، مسئله اصلی را به m مسئله زمانبندی ماشین های موازی، تجزیه می کند. در این الگوریتم ابتدا یک زمان پردازش مجازی برای هر کار محاسبه شده و سپس یک ترتیب از کارها برای اولین مرحله، بر اساس مرتب سازی صعودی زمانهای پردازش مجازی، مشخص می شود. ترتیب کارها در مراحل بعد، بر اساس قاعده FCFS بدست آورده شده و همچنین در تمامی مراحل، تخصیص کارها به ماشین ها، بر اساس زمانبندی پیشرو صورت می پذیرد. الگوریتم فوق شامل گامهای زیر است توانایی ایجاد جواب های با کیفیت را دارد [۱۵]:

گام اول:

برای هر کار j زمان پردازش مجازی (\tilde{p}_j) را به صورت زیر محاسبه نمایید:

$$\tilde{p}_j = \sum_{s=1}^m \frac{P_{js}}{\sum_{k=1}^{L_s} V_{ks}}, \quad \forall j=1,2,\dots,n$$

قرار دهید $s=1$.

گام دوم: اگر $s=1$ آنگاه ترتیب کارها در این مرحله را (π_1) با مرتب سازی صعودی آنها بر اساس زمانهای پردازش مجازیشان، بدست آورید. در غیر این صورت ترتیب کارها در مرحله s (π_s) را با مرتب سازی صعودی کارها، بر اساس زمان تکمیلشان در مرحله $s-1$ بدست آورید. اگر زمان تکمیل برای برخی کارها برابر بود، آنها را بر اساس مرتب سازی صعودی زمانهای پردازش مجازیشان مرتب کنید.

گام سوم: کارها را با استفاده از زمانبندی پیشرو (بر اساس π_s) به ماشینهای مرحله s اختصاص دهید.

گام چهارم: قرار دهید $s=s+1$.

۳-۲. جمعیت اولیه

جمعیت اولیه در الگوریتم ژنتیک که تعداد آن بر اساس پارامتر اندازه جمعیت (Popsiz) مشخص می شود، معمولاً بصورت تصادفی تولید می شود. در این مقاله جهت تقویت جمعیت اولیه، نیمی از آن را $(Popsiz/2)$ بصورت تصادفی و مابقی را بر اساس الگوریتم ابتکاری H2 که در ادامه شرح داده خواهد شد، ایجاد می نمایم. بدین صورت که با انجام $[Popsiz/2]$ مرتبه عمل جهش (به روش SM) بر روی جواب حاصله از الگوریتم H2، مابقی جمعیت تولید خواهد شد.

در اغلب الگوریتم های ابتکاری که در زمینه حل مسأله زمانبندی جریان کارکردی ترکیبی ارائه شده، روند کار از دو بخش تشکیل شده است. ابتدا برای هر مرحله، ترتیبی از کارها به دست آورده شده و سپس کارها در آن مرحله، بر اساس یک قاعده، به ماشینها اختصاص داده می شوند. لذا دو موضوع مهم در این الگوریتم ها عبارتند از:

(۱) مشخص کردن ترتیب کارها برای هر مرحله

(۲) نحوه تخصیص کارها به ماشینها در هر مرحله، بر اساس ترتیب بدست آمده

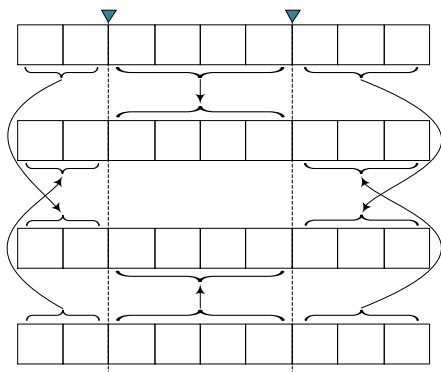
حالت خاصی از این روش بدین صورت است که بوسیله یک قاعده خاص ترتیب کارها در مرحله اول مشخص شده و ترتیب کارها در مراحل بعد، بر اساس قاعده FCFS^۱ بدست آورده می شود. نحوه تخصیص کارها به ماشینها نیز، در تمامی مراحل بر اساس زمانبندی پیشرو صورت می پذیرد. در الگوریتم H2 نیز از این رویکرد استفاده شده است. در ادامه ابتدا گامهای زمانبندی پیشرو بیان شده و سپس به شرح الگوریتم H2 خواهیم پرداخت.

1. First-Come First-Served

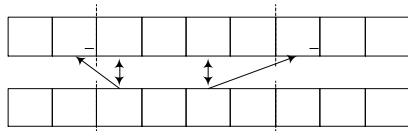
بسیاری از مقالات به عنوان بهترین روشها معرفی شده اند، به شرح ذیل استفاده می شود.

۳-۵-۱. تقاطع نگاشت ناقص^۱ (PMX)

روش PMX احتمالاً رایجترین روش تقاطع، در زمینه مسئله مورد بررسی باشد. فرض کنید با مسئله ای روبرو هستید که دارای ۹ کار می باشد. ابتدا دو کروموزوم والد انتخاب شده، سپس دو موقعیت از کروموزوم ها جهت برش، به تصادف انتخاب می شود. در اینجا دو نقطه ۲ و ۷ به تصادف انتخاب شده است. حال برای ایجاد اولین فرزند، بخشی که بین دو نقطه برش قرار دارد، از والد ۱ و دو بخش ابتدایی و انتهایی، از والد ۲ انتخاب می شوند. برای فرزند ۲ این روند معکوس می شود.



همانطور که مشاهده می شود دو فرزند تولید شده، جوابی شدنی برای مسئله نمی باشند (تکراری بودن برخی کارها و نبود بعضی دیگر). در فرزند ۱، کارهای تکراری (۳ و ۵) که در دو طرف نقاط برش قرار دارند را، با کارهایی از فرزند ۲ که دارای موقعیت یکسانی با کارهای تکراری که در قسمت میانی نقاط برش در فرزند ۱ (۳ و ۵) قرار دارند (۲ و ۷)، جایگزین می شوند.



برای فرزند ۲ نیز مشابه همین عمل صورت می پذیرد.

۳-۵-۲. تقاطع دو نقطه ای^۲ (TP)

در این روش نیز پس از انتخاب کروموزوم های والد، دو نقطه تقاطع به تصادف انتخاب می شوند. سپس برای تولید فرزند ۱، در

گام پنجم: اگر $s > m$ آنگاه به گام ششم بروید در غیر اینصورت بروید به گام دوم.

گام ششم: الگوریتم را خاتمه دهید.

۳-۳. ارزیابی

در هر مرتبه از تولید نسل جدید، کروموزوم ها توسط یک معیار برازندگی، مورد ارزیابی قرار می گیرند. در بسیاری از کاربردهای بهینه سازی، تابع برازندگی بر اساس تابع هدف مسئله اصلی بنا نهاده می شود. با توجه به کمینه سازی بودن هدف در مسئله اصلی، مقدار برازندگی بایستی با معکوس تابع هدف مسئله اصلی، متناسب باشد بطوریکه کروموزوم شایسته تر دارای مقدار برازندگی بزرگتری باشد:

$$\text{Fitness}(\text{chrom}_i) = \frac{1}{f(\text{chrom}_i)}, \quad i = 1, 2, \dots, \text{Popsiz}$$

در رابطه فوق $\text{Fitness}(\text{chrom}_i)$ مقدار برازندگی و $f(\text{chrom}_i)$ مقدار تابع هدف کروموزوم i ام می باشد.

با توجه به اینکه در این مقاله هدف کمینه سازی دامنه عملیات و همچنین مجموع زمانهای تکمیل است، لذا مقدار برازندگی برای هر کروموزوم، با توجه به نوع هدف مسئله، به یکی از شکلهای زیر می باشد:

$$\text{Fitness}(\text{chrom}_i) = \frac{1}{C_{\max}(\text{chrom}_i)}, \quad i = 1, 2, \dots, \text{Popsiz}$$

$$\text{Fitness}(\text{chrom}_i) = \frac{1}{\sum C_j(\text{chrom}_i)}, \quad i = 1, 2, \dots, \text{Popsiz}$$

۳-۴. انتخاب

در این مقاله، عمل انتخاب به این روش صورت می پذیرد که پس از انجام عمل تقاطع و جهش و محاسبه مقدار برازندگی برای هر کروموزوم، کروموزومها بر اساس مقدار برازندگی، به صورت نزولی مرتب می شوند. تعداد $\text{Popsiz} * 0.3$ کروموزوم (۳۰ درصد جمعیت جدید) از بالای لیست انتخاب می شوند. ۷۰ درصد باقیمانده نیز به تصادف از کروموزومهای باقیمانده انتخاب می شوند.

۳-۵. تقاطع

تقاطع و جهش دو جزء بسیار مهم در یک الگوریتم ژنتیک می باشند. عمل تقاطع در الگوریتم ژنتیک، جهت مبادله اطلاعات میان کروموزوم ها، در هنگام ایجاد کروموزوم های فرزند، مورد استفاده قرار می گیرد. در این مقاله از دو روش تقاطع که در

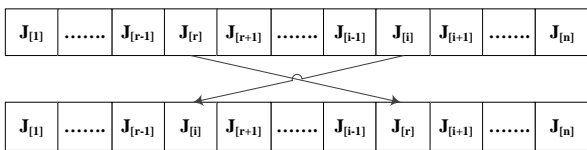
1. Partially Mapped crossover
2. Two Point crossover

۳-۶. جهش

مهمترین دلیل برای استفاده از این عملگر، فرار از نقاط بهینه محلی می باشد. همچنین ایجاد تفاوت و تمایز بین جوابهای موجود در جمعیت از دیگر دلایل استفاده از آن است. در این مقاله دو روش تبادل جفتی^۱ و حرکت انتقالی^۲ که از مؤثرترین روشها در این زمینه هستند، مورد استفاده قرار گرفته اند.

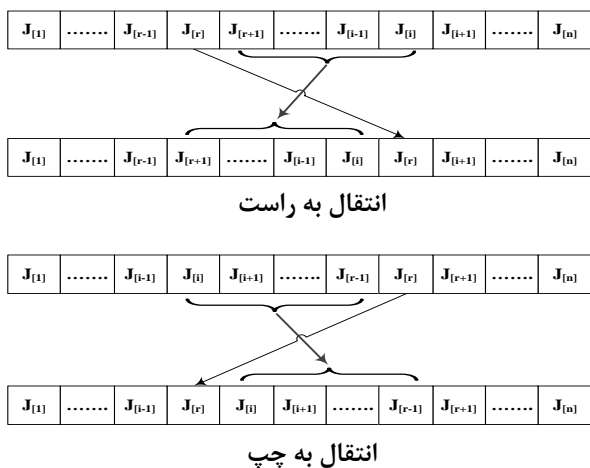
۳-۶-۱. تبادل جفتی (PI)

در این ساختار جستجوی همسایگی ابتدا دو موقعیت از برنامه زمانبندی (i و r) بصورت تصادفی انتخاب می شوند ($i \neq r, 1 \leq i, r \leq n$). سپس جای کارهایی که در این دو موقعیت قرار دارند ($J_{[i]}$ و $J_{[r]}$) را با هم عوض می کنیم.



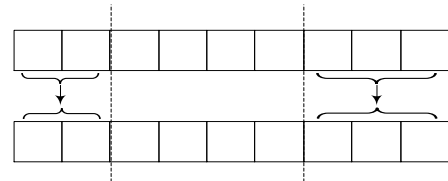
۳-۶-۲. حرکت انتقالی (SM)

در این روش ابتدا یک موقعیت از برنامه زمانبندی (r) بصورت تصادفی انتخاب می شود. سپس کاری که در این موقعیت قرار دارد ($J_{[r]}$) را به موقعیت دیگری (i) که آن هم بصورت تصادفی انتخاب شده است، منتقل می شود. لازم به ذکر است که موقعیت تمام کارهای بین این دو موقعیت، بدون تغییر در توالی آنها، به اندازه یک واحد به سمت چپ یا راست شیفت می شوند. این روش، اگر $1 \leq r < i \leq n$ باشد، انتقال به راست (RSM) و در صورتیکه $1 \leq i < r \leq n$ باشد، انتقال به چپ (LSM) نامیده می شود.

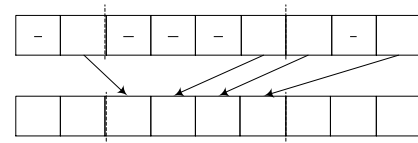


- 1. Pairwise Interchange (PI)
- 2. Shift Move (SM)

گام اول عناصر موجود در دو طرف نقاط تقاطع را از والد ۱، در فرزند ۱ وارد می کنیم.



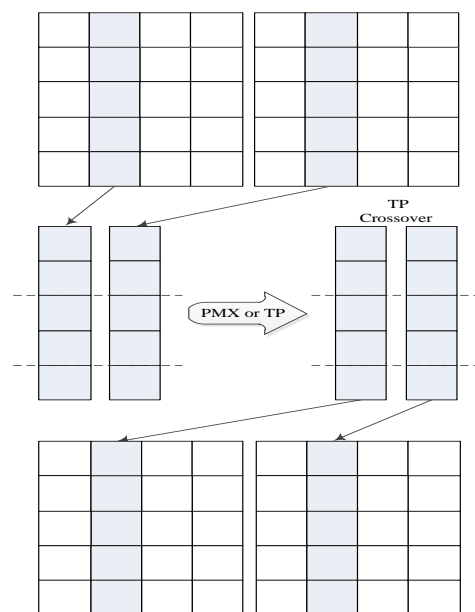
در گام بعد، موقعیت های بین دو نقطه تقاطع، در فرزند ۱، عناصری از کروموزوم والد ۲ بصورتی پر می شوند که عنصری تکراری، وارد فرزند ۱ نشود.



برای تولید فرزند ۲ کافی است جای دو کروموزوم والد عوض شود.

۳-۵-۳. تقاطع برای حالت کروموزوم دو بعدی

برای حالتی که کروموزوم ها به صورت دو بعدی تعریف می شوند، روش کار بدین صورت است که پس از انتخاب دو کروموزوم والد (به صورت تصادفی)، یک مرحله به تصادف انتخاب می گردد. سپس بین دو ترتیب (ترتیب کارها در مرحله انتخابی در کروموزومها)، به یکی از روشهای PMX و یا TP عمل تقاطع در یک بعد را انجام می دهیم. سپس دو کروموزوم فرزند حاصل از تقاطع یک بعدی را جایگزین دو مرحله ای که از کروموزوم های والد انتخاب شده اند، می کنیم.



۳-۷. شرایط خاتمه

پس از اینکه الگوریتم بارها تکرار شود، زمانی فرا می رسد که در جوابهایی که در جمعیت قرار دارند، تغییر محسوسی مشاهده نمی شود. در این هنگام است که الگوریتم ژنتیک پایان می یابد زیرا با تکرارهای بیشتر، میزان بهبودی که در مقدار تابع برازندگی ایجاد می شود، ناچیز است. شرایطی که برای خاتمه الگوریتم مورد استفاده قرار گرفته شده است، بسیار متنوع است اما از متداولترین شرایطی که در این زمینه در نظر گرفته شده و در این مقاله مورد استفاده قرار گرفته اند، می توان از (۱) تعداد نسلی که مقدار هدف بهبود نمی یابد و (۲) تعداد کل تکرار تولید نسل، نام برد.

۴. نتایج محاسباتی

در این بخش با طراحی یک آزمایش و حل مثال های عددی متفاوت کارایی الگوریتم های ژنتیک ارائه شده در بخش های گذشته مورد آزمایش قرار خواهد گرفت و بهترین ترکیب الگوریتم ژنتیک معرفی خواهد شد.

با توجه به اینکه در هر یک از سه بخش ۳-۱، ۳-۵ و ۳-۶ که به ترتیب عبارتند از نحوه نمایش، تقاطع و جهش، دو روش ارائه شده است، تعداد حالات مختلفی که می توان الگوریتم را ایجاد نمود برابر ۸ حالت می باشد (۲*۲*۲). به طور مثال GA.P.P بدین معنی است که در الگوریتم کروموزوم به صورت یک بعدی تعریف شده، تقاطع از نوع PMX بوده و جهش به صورت PI صورت گرفته شده است و یا 2GA.T.S حالتی را نشان می دهد که کروموزوم بصورت دو بعدی، تقاطع بصورت TP و جهش از نوع SM می باشد. این اطلاعات با جزئیات بیشتر در زیر جداول ۱ تا ۵ آورده شده است. در مسائلی که جهت انجام آزمایش از آنها استفاده خواهد شد، تعداد مراحل را ۵، ۱۰ و ۲۰ در نظر گرفته

شده است. همچنین تعداد کارها ۱۰، ۲۰، ۵۰ و ۱۰۰ لحاظ شده اند. تعداد ماشین در هر مرحله بصورت یکنواخت در بازه [۱،۴]، سرعت هر ماشین یکنواخت در بازه [۰/۵، ۱/۵] و زمان پردازش کارها در هر مرحله نیز بصورت یکنواخت در فاصله [۱۰، ۱۰۰] در نظر گرفته شده است.

با توجه به حالات در نظر گرفته شده برای تعداد کارها (۴ حالت) و تعداد مراحل (۳ حالت)، ۱۲ مسئله با ابعاد مختلف تولید شده و جهت دقت بیشتر نتایج، هر یک از مسئله ها را تحت هر یک از ۸ سناریو، به تعداد ۱۰ مرتبه اجرا شده است. همانطوری که قبلاً نیز اشاره گردید، مسئله مورد بررسی را یکبار با هدف کمینه سازی دامنه عملیات و با محدودیت میانگین زمان در جریان (F) و یکبار با هدف کمینه سازی مجموع زمانهای تکمیل و با محدودیت حداکثر زمان انجام کارها و یا به عبارتی وجود یک ضرب العجل (D) در نظر گرفته خواهد شد.

در حالتی که دامنه عملیات به عنوان هدف در نظر گرفته شود، می بایست مقداری را برای محدود نمودن مجموع زمانهای تکمیل انتخاب نمود. در این مقاله این مقدار بدین صورت برآورد گردیده است که مسئله را ۱۰۰۰ مرتبه، با ترتیب اولیه تصادفی زمانبندی نموده و میانگین بیشترین و کمترین مجموع زمان های تکمیل، به عنوان کران بالای مجموع زمانهای تکمیل، انتخاب شده است. در حالت دوم که مجموع زمانهای تکمیل به عنوان هدف مسئله است نیز به همین روال عمل می شود. در جدول ۱ نتایج مربوط به دامنه عملیات، در حالتی که مجموع زمانهای تکمیل با محدودیت همراه می باشند، آورده شده است. در جدول ۲ نیز نتایج مربوط به مجموع زمانهای تکمیل، هنگامی که با محدودیت زمان تکمیل کارها روبرو هستیم، نشان داده شده است.

جدول ۱. نتایج بدست آمده برای دامنه عملیات (C_{max})

تعداد کارها (n)	تعداد مراحل (m)	$F = \frac{\sum C_i}{n}$	GA.P.P	GA.P.S	GA.T.P	GA.T.S	2GA.P.P	2GA.P.S	2GA.T.P	2GA.T.S
۱۰	۵	۵۶۰/۸	۷۵۶/۷	۷۵۶/۷	۷۵۶/۷	۷۵۶/۷	۷۵۹/۴	۷۵۷/۶	۷۵۹/۴	۷۵۶/۷
	۱۰	۹۱۱/۲	۱۱۶۱/۰	۱۱۴۷/۹	۱۱۵۴/۳	۱۱۵۶/۳	۱۱۷۶/۸	۱۱۷۳/۷	۱۱۷۲/۵	۱۱۷۸/۰
۲۰	۲۰	۱۴۳۵/۹	۱۵۷۳/۸	۱۵۷۳/۸	۱۵۷۳/۸	۱۵۷۳/۸	۱۵۷۹/۴	۱۶۰۱/۴	۱۵۹۰/۴	۱۵۹۳/۱
	۵	۶۸۳/۴۵	۹۰۶/۱	۹۰۵/۷	۹۰۹/۰	۹۰۶/۷	۹۱۵/۲	۹۱۳/۶	۹۱۸/۲	۹۰۷/۵
۵۰	۱۰	۱۱۴۲/۷۵	۱۶۶۹/۹	۱۶۶۹/۶	۱۶۶۹/۳	۱۶۶۸/۷	۱۶۷۹/۴	۱۶۷۶/۸	۱۶۸۰/۸	۱۶۸۳/۸
	۲۰	۲۱۷۲/۷۵	۲۹۴۱/۱	۲۹۴۷/۴	۲۹۴۷/۸	۲۹۵۵/۳	۲۹۹۰/۸	۲۹۸۴/۳	۳۰۰۴/۲	۳۰۰۶/۸
۱۰۰	۵	۱۲۲۹/۱۴	۲۱۶۳/۷	۲۱۶۳/۲	۲۱۶۳/۲	۲۱۶۳/۷	۲۱۶۴/۸	۲۱۶۴/۰	۲۱۶۵/۳	۲۱۶۸/۶
	۱۰	۲۹۰۴/۴	۴۷۸۰/۹	۴۷۷۷/۲	۴۷۷۴/۵	۴۷۷۷/۰	۴۷۹۲/۱	۴۷۹۳/۵	۴۷۹۷/۲	۴۸۰۷/۰
۱۰۰	۲۰	۳۰۹۵/۶	۴۸۱۷/۱	۴۸۱۷/۱	۴۸۲۲/۰	۴۸۱۷/۱	۴۸۴۳/۸	۴۸۹۴/۹	۴۸۴۹/۵	۴۸۵۲/۴
	۵	۱۰۷۳/۶	۱۶۹۶/۵	۱۶۹۲/۹	۱۶۹۴/۲	۱۶۹۱/۳	۱۷۱۲/۰	۱۷۲۳/۳	۱۷۲۷/۱	۱۷۲۸/۱
۱۰۰	۱۰	۲۰۳۸/۱	۳۳۴۸/۶	۳۳۴۵/۶	۳۳۴۸/۶	۳۳۵۲/۴	۳۳۸۲/۴	۳۳۸۲/۶	۳۳۹۱/۱	۳۳۷۵/۲
	۲۰	۶۰۳۹/۶	۱۰۲۷۲	۱۰۲۶۵	۱۰۲۷۸	۱۰۲۸۶	۱۰۳۲۷	۱۰۳۵۵	۱۰۳۷۹	۱۰۳۵۶

G.A.P.P = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع PMX وجهش به صورت PI است.
 G.A.P.S = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع PMX وجهش به صورت SM است.
 G.A.T.P = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع TP وجهش به صورت PI است.
 G.A.T.S = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع TP وجهش به صورت SM است.
 2G.A.P.P = در الگوریتم ژنتیک کروموزوم از نوع دو بعدی، تقاطع از نوع PMX وجهش به صورت PI است.
 2G.A.P.S = در الگوریتم ژنتیک کروموزوم از نوع دو بعدی، تقاطع از نوع PMX وجهش به صورت SM است.
 2G.A.T.P = در الگوریتم ژنتیک کروموزوم از نوع دو بعدی، تقاطع بصورت TP و جهش از نوع SM است.
 2G.A.T.S = حالتی را نشان می دهد که کروموزوم بصورت دو بعدی، تقاطع بصورت TP و جهش از نوع SM است.

جدول ۲. نتایج بدست آمده برای مجموع زمانهای تکمیل

تعداد کارها (n)	تعداد مراحل (m)	C _{max}	G.A.P.P	G.A.P.S	G.A.T.P	G.A.T.S	2G.A.P.P	2G.A.P.S	2G.A.T.P	2G.A.T.S
۱۰	۵	۸۷۶/۹۴۵۷	۴۸۲۵/۴	۴۸۲۵/۴	۴۸۲۵/۴	۴۸۲۵/۴	۴۸۲۲/۹	۴۸۲۰/۵	۴۸۲۰/۹	۴۸۲۰/۸
	۱۰	۱۳۲۰/۴	۸۱۱۹/۰	۸۰۸۳/۹	۸۰۹۴/۷	۸۱۱۲/۸	۸۱۷۵/۲	۸۱۵۸/۰	۸۱۰۰/۴	۸۱۲۰/۰
۲۰	۲۰	۱۸۴۰/۴	۱۲۸۵۸/۰	۱۲۸۵۴/۰	۱۲۸۵۳/۰	۱۲۸۵۹/۰	۱۳۰۶۸/۰	۱۳۰۴۴/۰	۱۳۰۳۷/۰	۱۲۹۸۱/۰
	۵	۱۰۹۰/۵	۱۱۴۶۴/۰	۱۱۴۱۹/۰	۱۱۴۵۰/۰	۱۱۴۶۱/۰	۱۱۵۹۳/۰	۱۱۵۳۰/۰	۱۱۴۶۸/۰	۱۱۵۳۱/۰
	۱۰	۱۸۲۶/۹	۱۸۵۵۰/۰	۱۸۵۱۸/۰	۱۸۴۹۸/۰	۱۸۴۹۳/۰	۱۸۵۵۰/۰	۱۸۸۰۷/۰	۱۸۷۶۱/۰	۱۸۷۱۸/۰
	۲۰	۳۲۶۵/۳	۳۶۸۷۴/۰	۳۶۸۸۲/۰	۳۶۹۶۴/۰	۳۶۸۷۶/۰	۳۷۵۹۵/۰	۳۷۵۴۲/۰	۳۷۵۵۱/۰	۳۷۹۴۲/۰
۵۰	۵	۲۲۶۷/۳	۴۶۶۸۰/۰	۴۶۷۲۲/۰	۴۶۷۰۰/۰	۴۶۷۲۱/۶	۴۸۰۰۰/۰	۴۸۱۳۸/۰	۴۷۶۳۹/۰	۴۸۵۲۶/۰
	۱۰	۵۰۸۸/۳	۱۱۷۲۶۰/۰	۱۱۷۳۰۰/۰	۱۱۷۲۳۰/۰	۱۱۷۱۵۰/۰	۱۱۹۹۸۰/۰	۱۲۱۴۱۰/۰	۱۱۹۴۸۰/۰	۱۲۰۲۳۰/۰
	۲۰	۵۱۷۴/۵	۱۲۳۱۸۰/۰	۱۲۳۴۶۰/۰	۱۲۳۲۵۰/۰	۱۲۳۱۰۰/۰	۱۲۷۲۰۰/۰	۱۲۹۷۱۰/۰	۱۲۷۴۷۰/۰	۱۳۰۳۱۰/۰
۱۰۰	۵	۱۹۰۲/۳	۹۰۲۵۶/۰	۹۰۴۴۳/۰	۹۰۱۷۲/۰	۹۰۲۷۷/۰	۹۰۸۲۱/۰	۹۳۳۰۸/۰	۹۳۳۱۱/۰	۹۲۲۸۳/۰
	۱۰	۳۵۹۲/۵	۱۶۶۴۵۰/۰	۱۶۶۳۴۰/۰	۱۶۶۱۶۰/۰	۱۶۶۹۸۰/۰	۱۶۹۰۸۰/۰	۱۷۰۶۶۰/۰	۱۷۱۵۰۰/۰	۱۷۰۵۳۰/۰
	۲۰	۱۰۸۶۸/۰	۴۸۳۴۷۰/۰	۴۸۴۱۸۰/۰	۴۹۰۰۷۰/۰	۴۷۹۸۷۰/۰	۴۹۳۷۳۰/۰	۵۰۰۳۶۰/۰	۵۰۱۸۴۰/۰	۵۰۴۵۷۰/۰

G.A.P.P = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع PMX وجهش به صورت PI است.
 G.A.P.S = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع PMX وجهش به صورت SM است.
 G.A.T.P = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع TP وجهش به صورت PI است.
 G.A.T.S = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع TP وجهش به صورت SM است.
 2G.A.P.P = در الگوریتم ژنتیک کروموزوم از نوع دو بعدی، تقاطع از نوع PMX وجهش به صورت PI است.
 2G.A.P.S = در الگوریتم ژنتیک کروموزوم از نوع دو بعدی، تقاطع از نوع PMX وجهش به صورت SM است.
 2G.A.T.P = در الگوریتم ژنتیک کروموزوم از نوع دو بعدی، تقاطع از نوع TP وجهش به صورت PI است.
 2G.A.T.S = حالتی را نشان می دهد که کروموزوم بصورت دو بعدی، تقاطع بصورت TP و جهش از نوع SM است.

جدول ۳. درصد انحراف از بهترین جواب (هدف کمینه سازی دامنه عملیات)

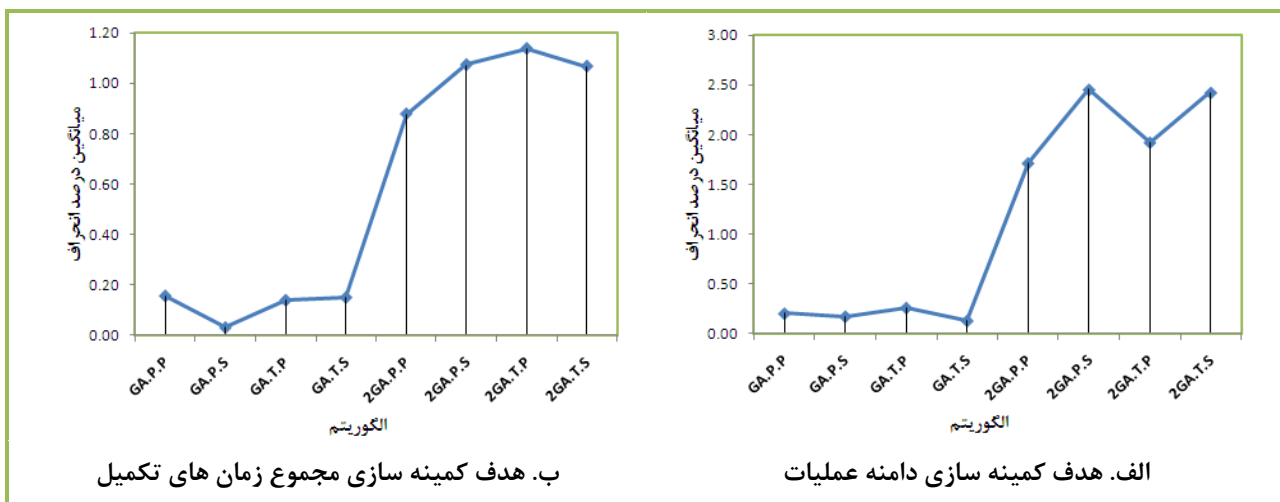
تعداد کارها (n)	تعداد مراحل (m)	MIN C _{max}	G.A.P.P	G.A.P.S	G.A.T.P	G.A.T.S	2G.A.P.P	2G.A.P.S	2G.A.T.P	2G.A.T.S
۱۰	۵	۷۵۶/۷	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۳۶	۰/۱۲	۰/۳۶	۰/۰۰
	۱۰	۱۱۴۷/۹	۱/۱۴	۰/۰۰	۰/۵۶	۰/۷۳	۲/۵۲	۲/۲۵	۲/۱۴	۲/۶۲
۲۰	۲۰	۱۵۷۳/۸	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۳۶	۱/۷۵	۱/۰۵	۱/۲۳
	۵	۹۰۵/۷	۰/۰۴	۰/۰۰	۰/۳۶	۰/۱۱	۱/۰۵	۰/۸۷	۱/۳۸	۰/۲۰
	۱۰	۱۶۶۸/۷	۰/۰۷	۰/۰۵	۰/۰۴	۰/۰۰	۰/۶۴	۰/۴۹	۰/۷۳	۰/۹۰
	۲۰	۲۹۴۱/۱	۰/۰۰	۰/۲۱	۰/۲۳	۰/۴۸	۱/۶۹	۱/۴۷	۲/۱۵	۲/۲۳
۵۰	۵	۲۱۶۳/۲	۰/۰۲	۰/۰۰	۰/۰۰	۰/۰۲	۰/۰۷	۰/۰۴	۰/۱۰	۰/۲۵
	۱۰	۴۷۷۴/۵	۰/۱۳	۰/۰۶	۰/۰۰	۰/۰۵	۰/۳۷	۰/۴۰	۰/۴۸	۰/۶۸
	۲۰	۴۸۱۷/۱	۰/۰۰	۰/۰۰	۰/۱۰	۰/۰۰	۰/۵۵	۱/۶۲	۰/۶۷	۰/۷۳
۱۰۰	۵	۱۶۹۱/۳	۰/۳۱	۰/۰۵	۰/۱۷	۰/۰۰	۱/۲۲	۱/۸۹	۲/۱۲	۲/۱۸
	۱۰	۳۳۴۵/۶	۰/۰۹	۰/۰۰	۰/۰۹	۰/۲۰	۱/۱۰	۱/۱۱	۱/۳۶	۰/۸۸
	۲۰	۱۰۲۶۵	۰/۰۷	۰/۰۰	۰/۱۳	۰/۲۰	۰/۶۰	۰/۸۸	۱/۱۱	۰/۸۹

G.A.P.P = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع PMX و جهش به صورت PI است.
 G.A.P.S = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع PMX و جهش به صورت SM است.
 G.A.T.P = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع TP و جهش به صورت PI است.
 G.A.T.S = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع TP و جهش به صورت SM است.
 2G.A.P.P = در الگوریتم ژنتیک کروموزوم از نوع دو بعدی، تقاطع از نوع PMX و جهش به صورت PI است.
 2G.A.P.S = در الگوریتم ژنتیک کروموزوم از نوع دو بعدی، تقاطع از نوع PMX و جهش به صورت SM است.
 2G.A.T.P = در الگوریتم ژنتیک کروموزوم از نوع دو بعدی، تقاطع از نوع TP و جهش به صورت PI است.
 2G.A.T.S = در الگوریتم ژنتیک کروموزوم از نوع دو بعدی، تقاطع بصورت TP و جهش از نوع SM است.

جدول ۴. درصد انحراف از بهترین جواب (هدف کمینه سازی مجموع زمان های تکمیل)

تعداد کارها (n)	تعداد مراحل (m)	MIN C _{max}	G.A.P.P	G.A.P.S	G.A.T.P	G.A.T.S	2G.A.P.P	2G.A.P.S	2G.A.T.P	2G.A.T.S
۱۰	۵	۴۸۲/۵	۰/۱۰	۰/۱۰	۰/۱۰	۰/۱۰	۰/۰۵	۰/۰۰	۰/۰۱	۰/۰۱
	۱۰	۸۰۸۳/۹	۰/۴۳	۰/۰۰	۰/۱۳	۰/۳۶	۱/۱۳	۰/۹۲	۰/۲۰	۰/۴۵
	۲۰	۱۲۸۵۳	۰/۰۴	۰/۰۱	۰/۰۰	۰/۰۵	۱/۶۷	۱/۴۹	۱/۴۳	۱/۰۰
۲۰	۵	۱۱۴۱۹	۰/۳۹	۰/۰۰	۰/۱۳	۰/۳۶	۱/۱۳	۰/۹۲	۰/۲۰	۰/۴۵
	۱۰	۱۸۴۹۳	۰/۳۱	۰/۱۴	۰/۰۳	۰/۰۰	۰/۳۱	۱/۷۰	۱/۴۵	۱/۲۲
	۲۰	۳۶۸۷۴	۰/۰۰	۰/۰۲	۰/۲۴	۰/۰۱	۱/۹۶	۱/۸۱	۱/۸۴	۲/۹۰
۵۰	۵	۴۶۶۸۰	۰/۰۰	۰/۰۹	۰/۰۴	۰/۰۹	۲/۸۴	۳/۱۲	۲/۰۳	۳/۹۵
	۱۰	۱۱۷۱۵۰	۰/۰۹	۰/۱۳	۰/۰۷	۰/۰۰	۲/۴۲	۲/۶۴	۱/۹۹	۲/۶۳
	۲۰	۱۲۳۱۰۰	۰/۰۶	۰/۲۹	۰/۱۲	۰/۰۰	۳/۳۳	۵/۳۷	۳/۵۵	۵/۸۶
۱۰۰	۵	۹۰۱۷۲	۰/۰۹	۰/۳۰	۰/۰۰	۰/۱۲	۰/۷۲	۲/۴۸	۲/۳۷	۲/۳۴
	۱۰	۱۶۶۱۶۰	۰/۱۷	۰/۱۱	۰/۰۰	۰/۴۹	۱/۷۶	۲/۷۱	۳/۲۱	۲/۶۳
	۲۰	۴۷۹۸۷۰	۰/۷۵	۰/۹۰	۲/۱۳	۰/۰۰	۲/۸۹	۴/۲۷	۴/۵۸	۵/۱۵

G.A.P.P = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع PMX و جهش به صورت PI است.
 G.A.P.S = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع PMX و جهش به صورت SM است.
 G.A.T.P = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع TP و جهش به صورت PI است.
 G.A.T.S = در الگوریتم ژنتیک کروموزوم از نوع تک بعدی، تقاطع از نوع TP و جهش به صورت SM است.
 2G.A.P.P = در الگوریتم ژنتیک کروموزوم از نوع دو بعدی، تقاطع از نوع PMX و جهش به صورت PI است.
 2G.A.P.S = در الگوریتم ژنتیک کروموزوم از نوع دو بعدی، تقاطع از نوع PMX و جهش به صورت SM است.
 2G.A.T.P = در الگوریتم ژنتیک کروموزوم از نوع دو بعدی، تقاطع از نوع TP و جهش به صورت PI است.
 2G.A.T.S = در الگوریتم ژنتیک کروموزوم از نوع دو بعدی، تقاطع بصورت TP و جهش از نوع SM است.



شکل ۳. میانگین درصد انحراف از بهترین جواب

$$PIM = \left(\frac{\text{Result } H_2 - \text{ResultGA}}{\text{Result } H_2} \right) \times 100\%$$

با توجه به نتایج جدول ۳ الگوریتم GA.P.S توانسته است به طور متوسط بیش از ۱۰٪ دامنه عملیات مسئله را کاهش دهد.

۵. نتیجه گیری و کارآیندگان

موضوع مهم در مسائل مربوط به ماشینهای موازی، تصمیم گیری در رابطه با نحوه تخصیص کارها به ماشینهای مختلف است. در حالیکه در مسائل مربوط به جریان کارکردی، تصمیم گیری درباره ترتیب حرکت کارها در کارگاه، مسئله اساسی می باشد. بنابراین در مسائل HFS تصمیمات اساسی در رابطه با (۱) تخصیص دادن و (۲) زمانبندی نمودن کارها، بر روی ماشینها، در هر مرحله است [۲]. در این مقاله مساله جریان کارکردی ترکیبی را بوسیله یک مدل برنامه ریزی دوهدفه معرفی و برای حل آن از روش الگوریتم ژنتیک استفاده شده است. برای حل مساله از کروموزوم های یک بعدی و دو بعدی استفاده کرده ایم. در کروموزوم های دو بعدی یک بعد نشانگر مراحل و بعد دیگر نشان دهنده ترتیب کارها در مراحل مختلف است. لازم بذکر است که در این مقاله برای حل مساله از روش اپسیلون-کانسترنیت استفاده شده است.

بمنظور ادامه کار، این نویسندگان برآنند تا از روش پارامتریکی-اپسیلون-کانسترنیت و باتلفیق الگوریتم ژنتیک مساله را حل و نقاط ناندامینیت را تعیین نمایند. نظر به اینکه مسایل نوع برنامه ریزی دوهدفه را می توان با استفاده از روشهای گوناگون حل کرد توصیه ما این است که محققین دیگر مساله را با استفاده از الگوریتم های متاهیوریستیکی حل و به مقایسه جواب ها بپردازند. در اغلب پژوهش های انجام شده در زمینه مسائل زمانبندی پژوهشگران تنها یک معیار عملکردی را به عنوان هدف در نظر گرفته اند ولی حالاتی وجود دارد که تنها پرداختن به یک معیار و عدم دخالت اهداف دیگر، منطقی به نظر نمی رسد. در این مقاله، زمانبندی با توجه به دو معیار عملکردی دامنه عملیات و مجموع زمان های تکمیل صورت پذیرفته است

مراجع

- [1] Chun, Y.H., "Optimal Number of Periodic Preventive Maintenance Operations Under Warranty", Reliability Engineering and System Safety, 37, 1992, pp. 223-225.
- [2] Ribas, R., Leisten, J.M., Farminan, "Review and Classification of Hybrid Flow Shop Scheduling Problems from a Production System and a Solutions Procedure perspective", Comput Oper Res 37,

جدول ۵. درصد بهبود دامنه عملیات

تعداد کارها (n)	تعداد مراحل (m)	$C_{max}(H_2)$	GA.P.P	GA.P.S
	۵	۹۲۰/۸	۷۵۶/۷	۱۷/۸٪
۱۰	۱۰	۱۳۵۸/۷	۱۱۴۷/۹	۱۵/۵٪
	۲۰	۱۹۱۵/۷	۱۵۷۳/۸	۱۷/۸٪
۲۰	۵	۱۰۹۸/۸	۹۰۵/۷	۱۷/۶٪
	۱۰	۱۷۱۹/۹	۱۶۶۹/۶	۲/۹٪
	۲۰	۳۲۸۹/۴	۲۹۴۷/۴	۱۰/۴٪
	۵	۲۲۷۰/۵	۲۱۶۳/۳	۴/۷٪
۵۰	۱۰	۵۲۱۲/۸	۴۷۷۷/۲	۸/۴٪
	۲۰	۵۱۹۸/۰	۴۸۱۷/۱	۷/۳٪
	۵	۱۹۳۶/۳	۱۶۹۲/۹	۱۲/۶٪
	۱۰۰	۳۵۶۸/۱	۳۳۴۵/۶	۶/۳٪
	۲۰	۱۰۶۳۸/۳	۱۰۲۶۵	۳/۵٪

به منظور مقایسه الگوریتم های مختلف ابتدا کمترین مقدار به دست آمده دامنه عملیات (و یا مجموع زمان تکمیل) را در هر مثال مشخص نموده و سپس درصد انحراف جواب هر یک از الگوریتم ها از این مقدار محاسبه شد. نتایج مربوط به این محاسبات را در جداول ۳ و ۴ مشاهده می کنید. شکل ۳ متوسط میانگین انحراف ایجاد شده نسبت به بهترین جواب را به ازای الگوریتم های مختلف نشان می دهد. چنانچه مشاهده می نمایید هر چند میان ۴ الگوریتم ابتدایی که همگی از نوع تک بعدی و ۴ الگوریتم بعدی که همگی الگوریتم های ژنتیک دو بعدی هستند تفاوت معناداری در میزان انحراف دارند اما در هر یک از این دو گروه میزان تفاوت این انحراف جزئی است. علاوه بر این در هر دو نمودار مشاهده می نمایید که بر خلاف آنچه تصور می شد الگوریتم های ژنتیک تک بعدی به نتایج بهتری نسبت به الگوریتم های ژنتیک دو بعدی دست یافته اند. البته احتمالاً این نتیجه به دلیل ضعیف بودن عملگر های ژنتیکی طراحی شده برای این الگوریتم است. هر چند تمام الگوریتم های تک بعدی نتایج قابل قبولی را ارائه داده اند اما در این میان الگوریتم GA.P.S توانسته است نتایج بهتری را به دست آورد و لذا به عنوان بهترین الگوریتم ژنتیک ی معرفی می گردد.

پس از انتخاب بهترین الگوریتم دامنه عملیات حاصل از این الگوریتم با دامنه عملیات حاصل از الگوریتم ابتکاری H_2 مقایسه گردید. و نتایج آن در جدول ۳ جمع آوری شد. دو ستون اول این جدول مربوط به ابعاد مسئله حل شده است، ستون سوم و چهارم به ترتیب دامنه عملیات ترتیب حاصل از الگوریتم ابتکاری H_2 و الگوریتم ژنتیک را نشان می دهد و در نهایت ستون آخر با مقایسه نتایج حاصل از دو ستون قبلی و از طریق رابطه زیر درصد بهبود دامنه عملیات مسئله را نشان می دهد.

- of Operational Research 169 (3) (2006) 781–800.
- [14] C.Y. Low, "Simulated Annealing Heuristic for Flow Shop Scheduling Problems with Unrelated Parallel Machines", Computers and Operations Research 32 (8) (2005) 2013–2025.
- [15] Amin-Naseri, M.R., Beheshti-nia, M.A., "Hybrid Flow Shop Scheduling with Parallel Batching", Int. J. Production Economics 117 (2009) 185-196.
- [16] Soewandi, H., Elmaghraby, S.E., "Sequencing on Two-Stage Hybrid Flowshops with Uniform Machines to minimize makespan", IIE Transactions 35 (5) (2003) 467–477.
- [17] Kyparisis, G.J., Koulamas, C., "A Note on Makespan Minimization in Two-Stage Flexible Flow Shops with Uniform Machines", European Journal of Operational Research 175, 2006, 1321–1327.
- [18] Sevastianov, S.V., "Geometrical Heuristics for Multiprocessor Flowshop Scheduling with Uniform Machines at Each Stage", Journal of Scheduling 5 (3), 2002, 205–225.
- [19] Riane, F., Artiba, A., Elmaghraby, S.E., "Sequencing a Hybrid Two-Stage Flowshop with Dedicated Machines", International Journal of Production Research 40 (17), 2002, 4353–4380.
- [20] Dessouky, M.M., Dessouky, M.I., Verma, S.K., "Flowshop Scheduling with Identical Jobs and Uniform Parallel Machines", European Journal of Operational Research 109, 1998, 620-631.
- [21] Huang, W., Li, S., "A Two-Stage Hybrid Flowshop with Uniform Machines and Setup Times" Mathematical and Computer Modelling 27 (2), 1998, 27-45.
- [22] McCormick, S.T., Pinedo, M.L., "Scheduling n Independent Jobs on m Uniform Machines with Both Flowtime and Makespan Objectives: A Parametric Analysis", UBC Faculty of Commerce Working Paper 90-MS-019 (1992).
- [23] Alcaraz, J., Maroto, C., Ruiz, R., "Solving the Multi-Mode Resourceconstraints Project Scheduling Problem with Genetic Algorithms", Journal of the Operational Research Society 54, 2003, 614–626.
- [24] Yaurima, V., Burtseva, L., Tchernykh, A., "Hybrid Flowshop with Unrelated Machines, Sequence-Dependent Setup Time, Availability Constraints and Limited Buffers", Computers & Industrial Engineering 56, 2009, 1452–1463.
- [3] T.B.K. Rao, "Sequencing in the Order a, b , with Multiplicity of Machines for a Single Operation", OPSEARCH: Journal of the Operational Research Society of India 7, 1970, 135–144.
- [4] Ying, K.C., "An Iterated Greedy Heuristic for Multistage Hybrid Flowshop Scheduling Problems with Multiprocessor Tasks", Journal of the Operational Research Society 60 (6), 2009, 810–817.
- [5] Haouari, M., Hidri, L., "On the Hybrid Flowshop Scheduling Problem", International Journal of Production Economics 113 (1), 2008, 495–497.
- [6] Janiak, A., Kozan, E., Lichtenstein, M., Oguz, C., "Metaheuristic Approaches to the Hybrid Flow Shop Scheduling Problem with a Cost-Related Criterion", International Journal of Production Economics 105 (2), 2007, 407–424.
- [7] Jin, Z.H., Yang, Z., Ito, T., "Metaheuristic Algorithms for the Multistage Hybrid Flowshop Scheduling Problem", International Journal of Production Economics 100 (2), 2006, 322–334.
- [8] Morita, H., Shio, N., "Hybrid Branch and Bound Method with Genetic Algorithm for Flexible Flowshop Scheduling Problem", JSME International Journal Series C-Mechanical Systems Machine Elements and Manufacturing 48 (1), 2005, 46–52.
- [9] Ruiz, R., Vazquez-Rodriguez, J.A., "The Hybrid Flow Shop Scheduling Problem", European Journal of operation research (2010); 205 : 1-18.
- [10] Chen, C.L., "Bottleneck-Based Heuristics to Minimize Total Tardiness for the Flexible Flow Line with Unrelated Parallel Machines", Computers and Industrial Engineering 56 (4), 2009, 1393–1401.
- [11] Ruiz, R., Serifoglu, F.S., Urlings, T., "Modeling Realistic Hybrid Flexible Flowshop Scheduling Problems", Computers and Operations Research 35 (4), 2008, 1151–1175.
- [12] Jenabi, M., Fatemi Ghomi, S.M.T., Torabi, S.A., Karimi, B., "Two Hybrid Metaheuristics for the Finite Horizon Elsp in Flexible Flow Lines with Unrelated Parallel Machines", Applied Mathematics and Computation 186 (1), 2007, 230–245.
- [13] Ruiz, R., Maroto, C., "A Genetic Algorithm for Hybrid Flowshops with Sequence Dependent Setup Times and Machine Eligibility", European Journal