

یک الگوریتم ژنتیک چندگانه برای حل مساله زمان بندی در محیط کارگاهی چندگانه با

محدودیت های کار شبانه و کار هم زمان : بررسی موردی از صنعت ترانسفورماتور

خلاصه

این مقاله به حل مساله زمان بندی کارگاهی در یک دنیای واقعی می پردازد و الگوریتم جدیدی را به عنوان راه حل ارائه می دهد. ابتدا مشخصات اختصاصی کار هم زمان و شبانه را در فرآیند تولید ترانسفورماتور، مورد بررسی قرار می دهیم. برای حل مساله در یک زمان منطقی و قابل قبول، الگوریتم ژنتیک را پیشنهاد می شود. این الگوریتم، با هدف به حداقل رساندن تاخیر کل، روش الگوریتم اکتشافی (NEH) Nawaz-Enscore-Ham، که یک الگوریتم جستجوی محلی (بهینه سازی) است را با قانون تخصیص دستگاه، تلفیق می کند. نتایج آزمایشگاهی نشان می دهد که، این الگوریتم پیشنهادی از الگوریتم NEH "که یک الگوریتم ژنتیک ساده است" و پنج قانون توزیع امکانات بر حسب عملکرد تاخیر کلی متوسط و شاخص انحراف نسبی، بهتر عمل می کند. الگوریتم پیشنهادی بر حسب راندمان و توانایی که دارد رقابتی تر جلوه می کند.

کلید واژه ها: حل مساله زمان بندی در محیط کارگاهی مختلط، کار شبانه، کار هم زمان، الگوریتم ژنتیک، جستجوی محلی، صنعت ترانسفورماتور.

1. معرفی:

حل مساله زمان بندی در محیط کارگاهی در بسیاری از سیستم های تولید معمول است. در محیط های خاصی، دستگاه های موازی از کپی های متعددی تشکیل شده و در پایگاه های متعددی دسته بندی می شوند. برای این محیط های تولید، مدل حل مساله زمان بندی در محیط کارگاهی به روش قدیمی نامناسب است، زیرا برخی پایگاه ها

از دستگاه‌های موازی استفاده می‌کند. این نوع مسائل می‌توانند تحت روش حل مساله زمان‌بندی کارگاهی چندگانه (HFSP) تعریف شوند.

حل مساله محیط کارگاهی چندگانه تعمیم و توسعه ای از سیستم تولید در محیط کارگاهی قدیمی است. این سیستم شامل دو یا چند جایگاه به طور ترتیبی و و یک یا چند دستگاه موازی در هر جایگاه است. مثال های حل مساله محیط کارگاهی چندگانه شامل تولید کفپوش، صنعت بطری شیشه ای و غیره هستند (Lopez & Roubellat 2008).

در این نوع کارگاه ها مشکل اصلی، تخصیص وظیفه به دستگاه‌ها در هر جایگاه و توالی مشاغل اختصاص داده شده به دستگاه‌ها است. (HFSP) ها به طور گسترده ای مورد مطالعه و بررسی قرار گرفته اند، اگر چه اغلب موارد از نوع ان پی سخت¹ می‌باشند (Linn & Zhanh,1999)..

این مقاله روی مساله زمان‌بندی در محیط کارگاهی چندگانه با دو دو محدودیت اختصاصی متمرکز شده است، در نظر گرفتن تیم کاری شبانه و روزانه، کار هم‌زمان انواع ترتیب های خاص. انگیزه تحقیق ما سیستم تولید ترانسفورماتور و تعداد شرایط قابلیت استفاده در بین انواع مختلف تولیدات و دستگاه‌هاست. در این مورد، راه حل ممکن که تاخیر کلی را به حداقل می‌رساند (یعنی زمان کلی که با آن ترتیب پردازش به تاخیر می‌افتد) به طرز حیاتی مهم است، زیرا هزینه خسارت مشاغل دارای دیرکرد اثر تعیین کننده ای روی شرکت دارد. علاوه بر خصوصیات (HFSP) عمومی، محدودیت‌هایی به لحاظ وجود زمان انتظار بین مراحل متوالی یک کار و لحاظ کردن کار شبانه یا کار هم‌زمان در هر مرحله، وجود دارد.

این مقاله به شکل زیر سازمان‌دهی شده است. در بخش آینده، تحقیقات قبلی در حل مساله زمان‌بندی کارگاهی چندگانه، مورد بررسی قرار می‌گیرد. این مساله و محدودیت های سیستم تولید ترانسفورماتور در بخش 3 تعریف می‌شود. سپس الگوریتم ژنتیک چندگانه برای حل این مساله در بخش 4 ارائه می‌شود. در بخش 5 نتایج بدست

¹چند جمله ای حل نشدنی در زمان بر حسب اندازه ورودی مساله (سخت) (Np-hard=non-deterministic Polynomial-time hard)

آمده برای تصدیق شیوه عملکرد انجام شده، خلاصه می‌شود. نهایتاً، نتایج و زمینه‌های مطلوب برای تحقیقات مطرح می‌شوند.

2. بررسی نوشته‌های وابسته به حرفه علمی.

Arthanary و Ramamurthy (1971) روش (HFSP) حل مساله زمان‌بندی در محیط چندگانه را مطرح نموده و الگوریتم شاخه و حد را ارائه کردند. Morris و Kochar در سال (1987) الگوریتم ابتکاری^۲ (اکتشافی) برای به حداقل رساندن میانگین زمان جریان را برای مساله خط جریان تغییرپذیر با بافرهای محدود، توسعه دادند. آنها مساله را به دو زیر مساله تقسیم کردند، توالی نقطه ورود و دیسپاچینگ^۳. HFSP دو مرحله‌ای برای NP-hard به وسیله Gupta (1988) مطرح شده است. بعداً (Gupta, Harriri & Potts) در سال 1997 نشان دادند که HFSP دو مرحله‌ای غیرپیشگرا به ظن قوی NP-hard است.

روش دقیق مبتنی بر مدل کردن ریاضی برای یافتن راه حل بهینه HFSP، می‌تواند عملکرد قوی‌تری را نسبت به روش ابتکاری تضمین نماید. (Fattahi, Hosseini, Jolai, and Tavakkoli-Moghaddam (2014) الگوریتم حد و شاخه را توسعه دادند که زمان تنظیم و تجمیع عملکردها را برای به حداقل رساندن زمان اتمام آخرین کار (Makespan) در نظر می‌گرفت. (Yu و Sun) در سال 2015 HFSP دو مرحله‌ای با محدودیت دسته‌ای و زمان پردازش متغیر از طریق روش آزاد سازی لاگرانژین را بررسی کردند. در واقع، به خاطر ماهیت NP-hard بودن آنها، روش‌های دقیق تنها به مسائلی با اندازه کوچک قابل اعمال هستند. بنابراین، روش‌های ابتکاری برای بدست آوردن تقریب‌های مطلوب در زمان قابل قبول به کار گرفته می‌شوند (Ribas, Leisten & Framinan 2010). مثال‌هایی از الگوریتم ابتکاری^۴ عبارتند از جستجوی همسایگی، شبیه سازی تبردی و الگوریتم ژنتیک.

² الگوریتم جستجوی کاشف

³ هماهنگ کردن تولید و مصرف

روش‌های ابتکاری برای حل محدودیت‌های HFSP که ناشی از کاربردهای عملی هستند، طراحی شده اند. در سال 1975 (Holland) ابتدا الگوریتم ژنتیک را در کتاب خود "تطبیق سیستم‌های طبیعی و مصنوعی" بیان کرد. در الگوریتم‌های ژنتیک مرسوم برای ایجاد تغییرات کوچک در کروموزوم از جهش استفاده شد که در جوامع مختلف مورد استفاده قرار گرفت. برخلاف الگوریتم‌های ژنتیک مرسوم، (Tsujiura & Gen 1999) یک عملگر جهش با روش جستجوی همسایگی برای تعیین راه‌حل‌های نزدیک به بهینگی، پیشنهاد کردند. (Botta-Genoulaz 2000) الگوریتم ابتکاری مبتنی بر روش دسته بندی بر حسب نزدیک‌ترین موعد تحویل با قوانین تخصیص اولین دستگاه در دسترس و آخرین دستگاه مشغول به کار برای HSFP، ارائه کرد. (Engin, Ceran & Yilmaz 2011) یک الگوریتم ژنتیک موثر برای زمان‌بندی کارگاهی چندگانه با وظایف چند پردازنده ای معرفی نمودند. (Liao, 2012) الگوریتم ژنتیک موثر برای زمان‌بندی کارگاهی چندگانه با وظایف چند پردازنده ای معرفی نمودند. (Tjandradjaja & Chung 2012) الگوریتم بهینه‌سازی ازدحام ذرات (PSO) را برای HFSP با کمترین طول زمان انجام کار واقعی را معرفی نمودند. آنها یک الگوریتم PSO سازگار شده با گلوگاه ابتکاری و شبیه سازی تبرییدی را برای کمک به خارج شدن از حد مطلوب توسعه دادند. (Bozejko, Prempera & Smutniki 2013) الگوریتم جستجوی ممنوعه را برای HFSP طراحی نمودند که از خطوط تولید مکانیزه نتیجه شده بود. (Costa, Cappadonna & Fichera 2014) الگوریتم ژنتیک را برای HFSP، با دسته کردن موازی و محدودیت‌های قابلیت انتخاب مطرح کردند. (Li, Pan & Wang 2014) برای به حداقل رساندن زمان اتمام کار HFSP، الگوریتم جستجوی همسایگی را با دو روش بهینه کردن واکنش شیمیایی و تخمین توزیع تلفیق نمودند. (Rossi, Pandolfi & Lanzetta 2014) قوانین تنظیم (تاسیس) پویا را برای HFSP با دسته بندی موازی دستگاه‌ها توسعه دادند. آنها الگوریتم جستجوی کاشف مبتنی بر نسبت بحرانی بین زمان‌های تاسیس و پردازش برای به حداقل رساندن زمان اتمام کار و تعداد کارهای با تاخیر، معرفی نمودند.

هنوز هم دو مشکل مربوط به تحقیقات زمان‌بندی کارگاهی وجود دارد. اولین مشکل پیچیدگی بیش از حد مربوط به اندازه مسائل موجود در جهان واقعی است. متأسفانه، گرچه روش‌های دقیقی مانند MILP یعنی برنامه ریزی پویا می‌توانند یک راه حل بهینه بیابند، اما اغلب این راه‌حل‌ها به دلیل زمان فوق‌العاده طولانی که برای محاسبه مسائل

بزرگ لازم است، غیرعملی هستند. بنابراین روش‌های ابتکاری مانند الگوریتم ژنتیک به مسائل پیچیده‌تر اعمال می‌گردند. اگرچه، زمان اجرا و کیفیت راه حل با نوع طراحی الگوریتم متفاوت خواهد بود. بنابراین، نیاز قابل توجهی به روش‌های ابتکاری یا فوق-ابتکاری احساس می‌شود.

مشکل دوم تعیین محدودیت‌های مختلف در صنعت و لحاظ کردن آنها در الگوریتم است. در مسائل جهان واقعی، یک محیط کارگاهی مرسوم تک-دستگاه در هر یک از طبقه‌ها به ندرت وجود دارد. عموماً، دستگاه‌های متعددی با قابلیت‌های مختلف در طبقه‌های موازی قرار خواهد داشت تا ظرفیت و موازنه حجم کار را افزایش دهند (Naderi, Gohari & yazdani 2014). اگرچه حجم وسیعی از مقالات تحقیقی قدیمی برای HFSP ها در سیستم‌های تولید وجود دارد، اما فرضیات و گمانه زنی‌ها زمانی آغاز می‌شوند که توسعه الگوریتم برای آنها به معنای وجود محدودیت در آنها باشد (Ruiz, Vazquez-Rodriguez 2010). در نتیجه، در نظر گرفتن محدودیت‌های آنها، مانند دستگاه‌های موازی ناهم‌بسته و شایستگی، یک قدم مهم برای افزایش کاربردپذیری آنها در زمینه مربوطه بوده، بنابراین برای تحقیقات بیشتر حائز ارزش است.

محدودیت‌های تحقیقات قبلی نسبت به این دو مشکل اهمیت مطالعه و تحقیق در مورد روش چندگانه برای HFSP ها را برجسته می‌کند. در این تحقیق، بخش سوم با نشان دادن محدودیت‌های معین در کارخانه تولید ترانسفورماتور، به مشکل دوم می‌پردازد. بخش چهارم با توضیح الگوریتم چندگانه که به طور موثری یک الگوریتم ژنتیک را با روش‌های ابتکاری ترکیب می‌کند، و مشکل اول را بررسی می‌نماید.

3. تعریف مساله.

با در نظر گرفتن افزایش جو رقابتی بازار و نیاز به عرضه کردن محدوده‌ای از ولتاژها و خازن‌های لازم، فرآیند زمان‌بندی بایستی انواع مختلفی از ترانسفورماتورها را در بر بگیرد. به علاوه، تعدادی از دستگاه‌های موازی (محیط های کاری و کوره‌های خشک کننده) در هر مرحله از پردازش وجود دارند که هر یک شامل محدودیت‌های مخصوص به خود هستند. تمام روند تولید مولدها در شکل (1) خلاصه شده است.

مساله زمان‌بندی یک محیط کارگاهی چندگانه HFSP با m مرحله است. هر مرحله شامل دستگاه‌های متعددی است که به صورت موازی کار می‌کنند، اما جریان کارها از طریق مراحل یک جهته صورت می‌گیرد. در برخی مراحل تنها یک دستگاه وجود دارد، اما طبقه پایانی بایستی دارای دستگاه‌های مختلفی باشد. نوع دستگاه‌های موازی می‌تواند یکسان، یک شکل یا نامرتب باشد عمل انتخابی مربوط به یک دوره خاص از پردازش، وابسته به دستگاه‌های انتخابی است.

با توجه به نوشته مشهور سه-زمینه‌ای (Pinedo 2008) مساله تولید ترانسفورماتور می‌تواند با رابطه زیر بیان شود

$$FH2, \left((RM^{(k)})_{k=1}^4 \right) |r_j| \sum T_j \quad \text{: (Ruiz, Vazquez-Rodriguez 2010)}$$

نوع دستگاه‌های موازی از نوع دستگاه‌های موازی نامرتب بوده و زمان پردازش به دستگاه تخصیص یافته وابسته است. در کاربردهای عملی خاص با پردازش پیوسته عملکرد، مانند تولید در کارخانه پلاستیک، یک فضای ذخیره واسطه محدود بین مراحل (طبقات) وجود دارد (Moradinasab, Shafaei, Rabiee & Ramazani 2013). در این حالت تعداد عملکردها (وظائف) در ذخیره واسطه بایستی برای کاهش هزینه سرمایه گذاری به حداقل برسد. برای برآوردن این هدف، ضروری است که صف انتظار بین دو مرحله (طبقه) متوالی تحت قوانین FIFO عمل کند، که این قوانین مفروض در ذیل آمده‌اند.

1. تعداد طبقات و تعداد دستگاه‌ها در هر طبقه بایستی قبلاً مشخص شده باشد. تعداد وظایف (عملکردها) و زمان پردازش آنها نیز بایستی از قبل مشخص شود.

2. هر دستگاه در هر زمان تنها می‌تواند یک پردازش را انجام دهد. معافیت از پیش مجاز نمی‌باشد.

3. تمامی دستگاه‌ها برای کل دوره زمان‌بندی بایستی در دسترس بوده و هیچ دستگاه خرابی وجود نداشته باشد.

4. هدف به حداقل رساندن دیرکرد کل است، که دیرکرد کل از رابطه زیر تعریف می‌شود:

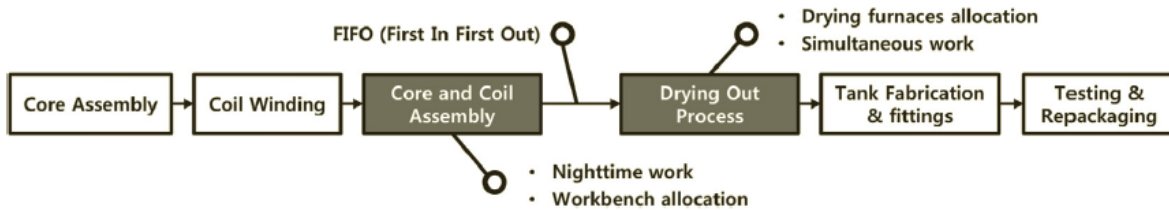
$$\text{Total Tardiness} = \sum_{i=1}^n \max(0, C_i - d_i)$$

که C_i زمان تکمیل وظیفه i ام، d_i موعد تحویل وظیفه i ام و n تعداد وظایف است.

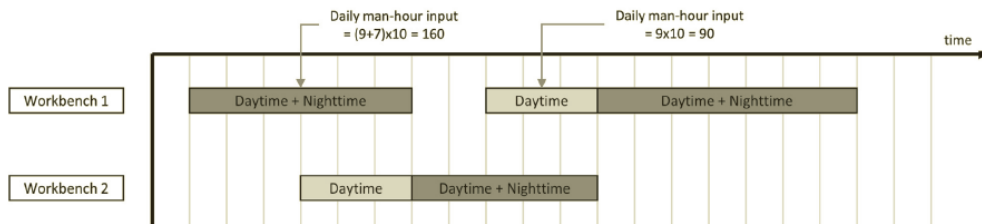
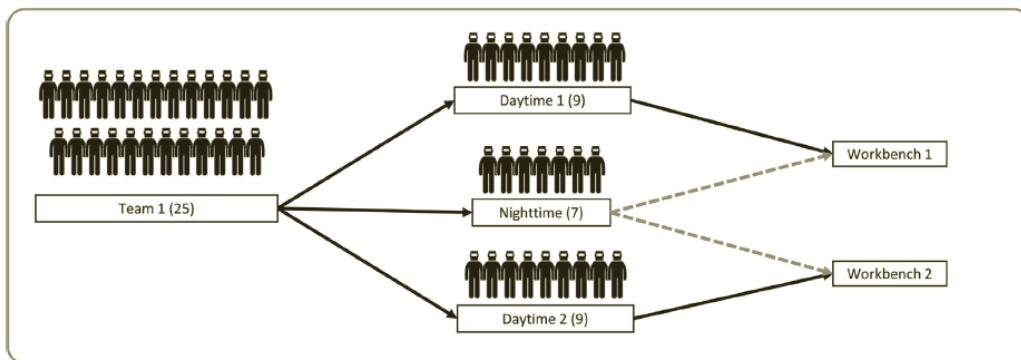
3.1. محدودیت‌های مشخص.

3.1.1 کار شبانه.

همانگونه در شکل (2) نشان داده شده، گروه کاری می تواند به سه زیر گروه تقسیم شود، دو گروه برای شیفت کار روزانه و یک گروه برای شیفت کار شبانه. در یک کارگاه تولید ترانسفورماتور، گروه کاری قابل تقسیم عموماً دارای دو میز کار برای پردازش عملیات مرحله (1) خود هستند، یعنی، هر گروه میز کاری مخصوص به خود را دارد.



شکل (1): فرآیند تولید ترانسفورماتور



شکل (2): مثال از کار شیفت شبانه.

همچنین، هر یک از میزها می تواند به گروه شیفت شب کار تعلق یابد. در فرآیند تولید ترانسفورماتور، هر میز کاری که در دسترس و آماده باشد به گروه شب کار اختصاص داده می شود. اهمیت اختصاص دادن میز کارها به گروه های کاری جهت حصول اطمینان از انجام و تکمیل به موقع کارها است. زمان های پردازش کار بسته به اینکه به گروه روز کار یا شب کار اختصاص یافته است، می تواند متفاوت باشد، زیرا زمان پردازش و تولید یک ترانسفورماتور بر مبنای

تولید در واحد زمان کار، محاسبه می‌شود. در حالی که سیستم شب‌کاری برای برآوردن موعد تحویل مطلوب، انتخاب قابل انعطاف تری را با اختصاص نیروی که به کار مورد نظر، اتخاذ می‌کند. انتخاب این سیستم تخصیص انعطاف پذیر، بخاطر هزینه خسارت بسیار زیاد تحویل سفارش‌های با تاخیر و مسائل مربوط به حجم کار نامتعادل، بسیار اهمیت دارد.

3.1.2. کار همزمان.

در فرآیند تولید ترانسفورماتور، کارگران بایستی یک کوره خشک کنی آماده را با اندازه گیری طول، عرض و ارتفاع ترانسفورماتور محاسبه و تعیین کنند. برخی ترانسفورماتورها در صف انتظار به اندازه ای کوچکند که می‌توانند با یک دستگاه معین مطابق شوند، بنابراین برخی دستگاه‌ها می‌توانند دو عملکرد را برای بکارگیری کوره‌های خشک‌کنی و کاهش زمان انجام پروژه همزمان انجام دهند، که در شکل (3) نشان داده شده است. در این مورد، برای افزایش بهره‌برداری از کوره‌های خشک‌کنی و کاهش زمان کل پردازش، سفارش‌ها بایستی همزمان به انجام برسند. همچنین، این امر هزینه کلی نیروی الکتریسته مصرفی و استفاده از نیروی کار را کاهش داده و پردازش سفارش‌های بیشتری را امکان‌پذیر می‌کند.

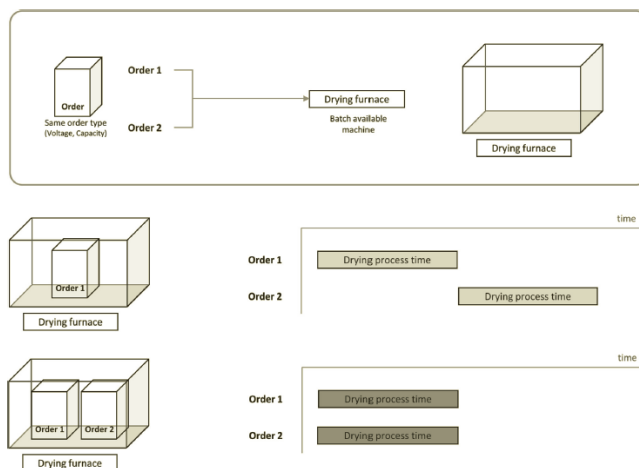
3.2. یک مثال توضیحی و روشن کننده.

برای توضیح مفاهیم شرح داده شده در شکل (4)، یک نمونه مثال کوچک از این مساله را ارائه می‌کنیم. مرحله اول شامل محدودیت کار شیفت شبانه است و مرحله دوم کار همزمان را مورد ملاحظه قرار می‌دهد. مرحله (1) شامل سه میز کار موازی و دو گروه کاری است. هر گروه کاری بایستی یک میز کار را برای پردازش یک سفارش در مرحله اول، اختصاص دهد. روند کاری گروه (1) می‌تواند به گروه‌های کار روزانه (1)، کار روزانه (2) و کار شبانه تقسیم شود. ولی گروه (2) امکان تقسیم ندارد. جدول (1) شرایط در دسترس بودن سفارش دستگاه، جدول (2) موعد تحویل، ساعات کل کاری لازم برای مرحله (1) و زمان دسترسی هر سفارش را نشان می‌دهد. جدول (3) تخصیص نیروی کار به هر گروه را نشان می‌دهد.

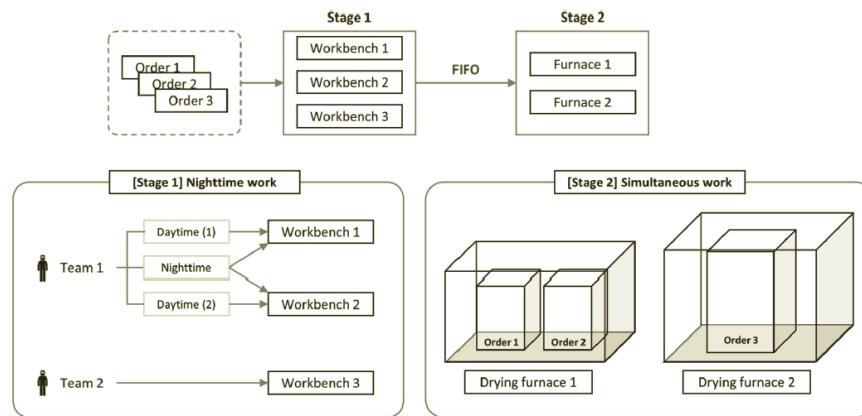
مرحله (2) ترکیبی از دو دستگاه موازی است. این دو دستگاه می‌توانند دو سفارش را بطور هم‌زمان پردازش نمایند، همچنین دارای ولتاژ و ظرفیت مشابه بوده و به اندازه کافی کوچک هستند تا بتوانند به راحتی با هر دستگاهی سازگار شوند. از جدول (1) داریم، سفارش‌های (1) و (2) توسط کوره‌های (1) و (2) بعنوان یک کار هم‌زمان پردازش شوند.

دسترس پذیری بودن کار هم‌زمان با علامت \odot در جدول (1) بیان شده است. جدول (2) زمان پردازش هر محصول در مرحله (2) را نشان می‌دهد.

با توجه به تاخیر کلی، زمان بندی دوم، اجازه می‌دهد تا کارها به طور هم‌زمان در مرحله (2) انجام شوند، بنابراین بهتر از زمان بندی اول است (شکل 5 و 6 را ببینید). به علاوه زمان بندی دوم می‌تواند، هزینه بهره‌برداری از کوره دوم را کاهش دهد، زیرا دو سفارش مختلف بطور هم‌زمان پردازش می‌شوند. بنابراین بایستی، محدودیت‌های مشخص و دقیق مساله را، برای توسعه یک الگوریتم زمان بندی موثر در نظر داشته باشیم،



شکل (3): مثالی از کار هم‌زمان



شکل (4): نمایش محیط کارگاهی چندگانه در مساله مثال دومرحله ای

Order	Stage 1			Stage 2	
	Workbench 1	Workbench 2	Workbench 3	Fumace 1	Fumace 2
1	○	○	X	⊙	⊙
2	○	○	X	⊙	⊙
3	○	○	X	X	○
4	X	X	○	X	○

جدول (1): ماتریس شرایط دسترس پذیری دستگاه-سفارش.

به گونه‌ای که بتواند در یک محیط واقعی پیاده شود. الگوریتم ابتکاری پیشنهاد شده، در بخش بعدی مورد بررسی قرار خواهد گرفت.

4. الگوریتم مطرح شده.

اکنون روش تحلیل الگوریتم ژنتیک چندگانه (HGA) پیشنهادی را ارائه می‌کنیم. کل چارچوب HGA در ابتدای بحث توضیح داده شده و سپس روال آن برای HFS به تفصیل توضیح داده شد. الگوریتم پیشنهادی به شرح زیر خلاصه شده است شکل (7) را ببینید.

به هنگام در نظر گرفتن HFSP در جهان واقعی، مشکل اصلی تعیین لیست وظائف در در نقطه ورودی و اختصاص این وظایف به دستگاه‌های مختلف است. لیست کارها در مرحله الگوریتم ژنتیک و مرحله جستجوی محلی تعیین شده است. در الگوریتم ژنتیک چندگانه، روش جستجوی همسایگی را (نوعی از روش جستجوی محلی) با جهش، crossover و انتخاب حلقه‌های الگوریتم ژنتیک ترکیب می‌کنیم. اگر بهترین راه حل در جمعیت، در مراحل با آستانه N بهبودی از خود نشان ندهد، مرحله عملیات الگوریتم ژنتیک متوقف شده و به مرحله عملیات جستجوی

محلی بازمی‌گردیم. الگوریتم ژنتیک چندگانه پیشنهادی، الگوریتم ژنتیک معمولی را برای جستجوی کلی جمعیت انتخاب شده، اعمال می‌کند، در حالی که جستجوی همسایگی، جستجوی محلی را برای هر یک از کروموزوم‌ها انجام می‌دهد.

اختصاص عملی این کارها به دستگاه‌های در دسترس، در مرحله رمز گشایی کروموزوم اجرا شده است. زمان‌بندی دقیق و میزان سازگاری توسط قانون تخصیص دستگاه، محاسبه شده است. هدف سه مرحله قبلی، محاسبه تابع واقعی است، در حالی که تابع در مرحله رمز گشایی کروموزوم‌ها، زمان‌بندی واقعی را توضیح می‌دهد.

4.1. مرحله الگوریتم NEH.

الگوریتم ابتکاری (Nawaz-Enscore-Ham) یک روش بهینه برای تبدیل مساله زمان‌بندی کارگاهی با به حداقل رساندن زمان اتمام کار می‌باشد (Ruiz, Maroto 2005). اگر موعد تحویل در نظر گرفته شود، راه‌های مختلفی برای طبقه بندی کارها وجود دارد. اما اگر کارها بر اساس زودترین موعد تحویل طبقه بندی شود، این روش NEH_{edd} نامیده می‌شود (Vallada, Ruiz & Minella 2008). در این مرحله، راه‌حل اولیه برای الگوریتم ژنتیک از طریق NEH_{edd} تولید می‌شود.

الگوریتم NEH

مرحله اول: لیست کارها بر اساس موعد تحویل غیر-افزایشی مرتب نمایید.

مرحله دوم: دو کار را در نظر بگیرید و و آن‌ها را برای به حداقل رساندن تاخیر کلی، زمان‌بندی کنید.

مرحله سوم: برای $k=3$ تا n مرحله چهارم را انجام دهید.

مرحله چهارم: کار شماره k را داخل زمان‌بندی قرار دهید تا تاخیر کلی نسبی بین k مقدار موجود را به حداقل برسانید.

Order	Release time	Due date	Required man-hour at stage 1	Process time at stage 2
1	4	15	580	5
2	3	15	580	5
3	5	18	720	6
4	7	24	1400	7

جدول (2): ماتریس اطلاعات سفارش.

4.2. مرحله الگوریتم ژنتیک

4.2.1. نمایندگی کروموزوم

یک کروموزوم دنباله ای از کارهایی را نشان می دهد که بایستی در مرحله اول برای زمان بندی در نظر گرفته شوند. این نمایش مبتنی بر تبدیل کار، به صورت گسترده ای در مسائل زمان بندی مورد استفاده قرار می گیرد.

4.2.2. اجتماع اولیه

هر کروموزوم اولیه به صورت تصادفی از اپراتورهای جهش و crossover مبتنی بر قوانین 5 گانه دیسپاچینگ تولید می شود (زودترین موعد تحویل EDD، slack، نسبت بحرانی، COVERT و MDD). این قوانین در عمل و به عنوان دنباله اولیه الگوریتم های ابتکاری استفاده می شوند (Tari & Olfat 2013). دنباله کارها بر اساس ترتیب غیر-افزایشی هر شاخص قوانین، تعیین می شود.

اپراتورهای جهش و crossover می توانند از تنوع در یک اجتماع پشتیبانی کرده و باعث شوند الگوریتم های چندگانه از مینیمم های محلی اجتناب کنند. به علاوه، از مشابه شدن بیش از حد راه حل ها به هم، جلوگیری نمایند.

4.2.3. Crossover

روش crossover دو نقطه ای به الگوریتم پیشنهادی اعمال شد، تا این اطمینان حاصل شود که حداقل سه ژن بین هر جفت از کروموزوم های با هم مبادله می شوند (Korytokowski, Wisniowski & Rymaszewski, 2013). یک مثال کوتاه و مختصر در شکل (8) نشان داده شده است.

4.2.4. جهش

از روش تبادل جهش، در الگوریتم پیشنهادی استفاده شده است. زیرا تنوع بیشتری نسبت به دیگر اپراتورهای تولید می کند (Fang, Lu & Li 2009). در روش جهش تبدالی، دو ژن به تصادف انتخاب شده و با هم مبادله می شوند. مثالی از اجرای جهش تبدالی در شکل (9) نشان داده شده است.

4.3. مرحله جستجوی محلی.

اصلی ترین مزیت چندگانه کردن الگوریتم ژنتیک با یک الگوریتم جستجوی محلی، همگرا شدن به یک نقطه بهینه است. همچنین جستجوی محلی برای انتخاب راه حل های باقی مانده از اجتماع های قبلی به کار می رود. در این مرحله فرآیند جستجوی محلی اعمال شده و می تواند تحت مراحل زیر نوشته شود.

فرآیند جستجوی محلی

قدم اول: یک $seed\ solution^5$ را در نظر گرفته و عنوان S به آن بدهید (یعنی مجموعه ای صفات یا پارامترها که برای تعریف جمعیت اولیه کاربرد دارد).

قدم دوم: یک مجموعه همسایگی N ایجاد کنید. این مجموعه از مجموعه S با مبادله کردن تمام جفت کارهای همسایه انجام می شود.

قدم سوم: در مجموعه همسایگی N ایجاد شده توسط $seed\ solution$ ، یک زمان بندی n انتخاب کرده و معیار تابع برازش (fitness-function) را برای آن محاسبه کنید..

قدم چهارم: اگر تمامی راه حل های همسایگی مجموعه S امتحان شوند، راه حل همسایگی با کمترین مقدار fitness-function و نسبت پیشرفت را جستجو کنید. و اگر هیچ راه حل همسایگی نبود تا راه حل کلی را اصلاح کرده و آن را ترقی بخشد، آنگاه فرآیند را به پایان برسانید. در غیر این صورت، $seed\ solution$ را با راه حل همسایگی دارای کمترین معیار تابع برازش جایگزین کرده و به مرحله سوم بازگردید.

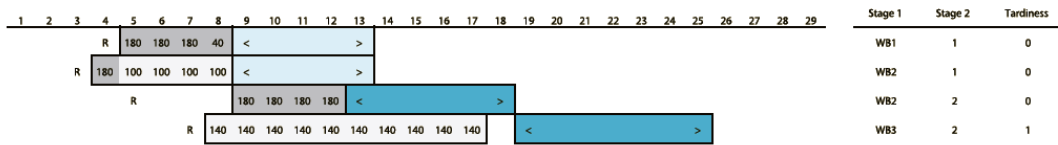
	Team 1		Team 2	
	Daytime (1)	Daytime (2)	Nighttime	Daytime Only
	Workbench 1	Workbench 2	Workbench 1, 2	Workbench 3
Number of workers	10	10	8	14
Total	28		14	

جدول (3): اختصاص نیروی کار گروه (2) و گروه (2) در مرحله یک.

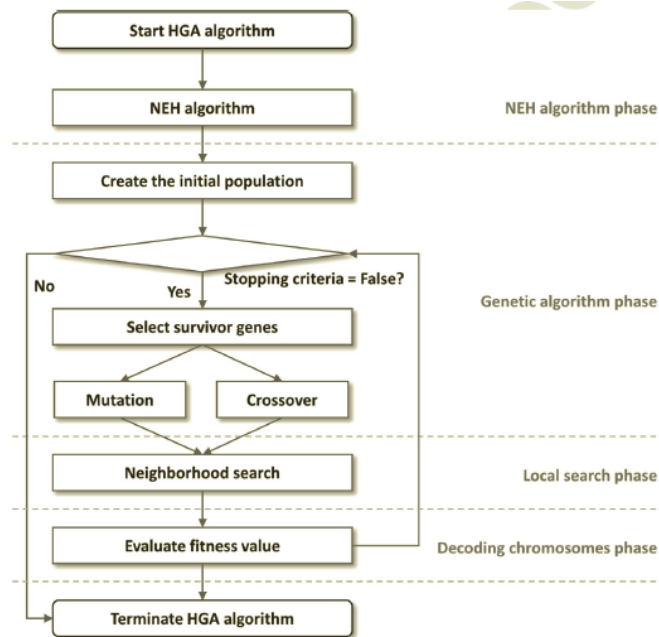
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	Stage 1	Stage 2	Tardiness	
R				180	180	180	40	<																							WB1	1	0
R			180	100	100	100	100	<																							WB2	2	0
R									180	180	180	180	<																		WB2	2	1
R																															WB3	2	2

⁵ The **SEED** solution consists of an integrated set of services, solutions and tools for the creation, management and delivery of special actions.

شکل (5): اولین زمان‌بندی ممکن برای نمونه مساله .



شکل (6): دومین زمان‌بندی برای مساله نمونه.



شکل (7): چارچوب کلی الگوریتم ژنتیک چندگانه.

4.4. مرحله رمزگشایی کروموزوم.

برحسب فهرست کارها در نقطه ورود، زمان‌بندی دقیق کار، با زمان‌های آغاز و پایان، با استفاده از قانون تخصیص دستگاه تعیین می‌شود.

4.4.1. قانون تخصیص دستگاه.

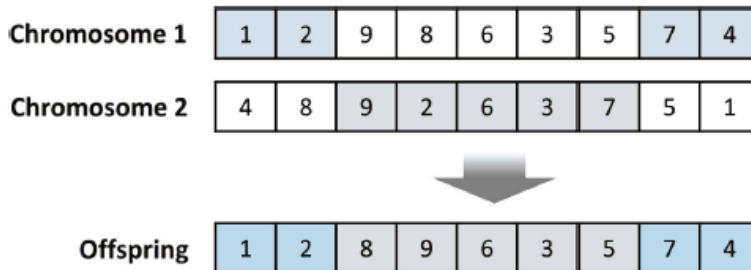
الگوریتم پیشنهادی دنباله ای از کارها را تولید می‌کند، اما اطلاعاتی از وظائف اختصاص داده شده به دستگاه به دست نمی‌دهد. بنابراین قانون اختصاص دستگاه نیاز دارد تا یک زمان‌بندی دقیق از نمایش کروموزوم تولید نماید. در این مقاله، قانون امتیازدهی دستگاه ارائه شده است.

اگر دستگاه‌ها ناهم بسته نباشند (مرتبط باشند)، به این معناست که کارها داری زمان پردازشی هستند که به دستگاه مورد استفاده وابسته است. بنابراین، کاری بایستی به دستگاه اختصاص داده شود که وظیفه پردازش خود را در کوتاه‌ترین زمان به اتمام برساند. در این حالت، قانون امتیاز دهی دستگاه، تمایل دارد تا زمان دیرکرد کلی و تعداد کارهای انجام شونده را، بخصوص برای یک HFS با کار شبانه و همزمان، به حداقل برساند (Jun& Park 2013). شاخص امتیاز دهی دستگاه از روابط زیر محاسبه می شود:

$$Machine\ Score_{k,s} = \alpha \times Uniqueness_{k,s} + (1 - \alpha) \times Usefulness_{k,s}$$

$$Uniqueness_{k,s} = \frac{\sum_t (a_{k,s,t} \times \prod_{l \neq k} (1 - a_{l,s,t}))}{\sum_t a_{k,s,t}}$$

$$Usefulness_{k,s} = \frac{v_{k,s}}{\max_g (v_{g,s})}$$



شکل (8): اپراتور دو نقطه ای crossover.



شکل (9): اپراتور جهش تبدیلی.

در این قانون، دستگاه‌های آماده به کار، برحسب ترتیب غیر افزایشی اندیس امتیاز دهی دستگاه، تخصیص می یابند. در ابتدا، دستگاه‌های معینی در مرحله یکسان در گروه طبقه بندی می‌شوند. اندیس امتیاز دهی ماشین ترکیبی از جمع وزن دار شده یکتایی گروه k ، و سودمندی در مرحله s است. α وزن منحصر به فرد یک گروه دستگاه و t نوع ترتیب را بر اساس ولتاژ و ظرفیت نشان می‌دهد. $v_{k,s}$ سرعت نسبی گروه k ام دستگاه‌ها در مرحله s است. اگر گروه دستگاه k بتواند براساس نوع ترتیب t در مرحله s پردازش کند، داریم $a_{k,s,t} = 1$ و گرنه $a_{k,s,t} = 0$.

4.4.2. تابع برازش

ارزیابی کروموزومها توسط تابع برازش انجام می‌شود. تابع برازش نقش مهمی را در انتخاب ژن‌های باقی‌مانده برای نسل بعدی، ایفا می‌کند. در این تحقیق تابع برازش بعنوان دیرکرد کلی زمان‌بندی تعریف شده است.

4.4.3. انتخاب.

انتخاب ژن‌های باقی‌مانده نسل بعدی توسط روش چرخ گردان⁶ انتخاب می‌شود (Gen & Cheng 1997).

4.4.4. معیار توقف.

بیشترین زمان محاسبه شده تحت عنوان شاخص توقف استفاده می‌شود.

5. نتایج عملی و آزمایشگاهی.

در شبیه‌سازی آزمایشی در مرحله اول 30 دستگاه و در مرحله دوم 12 دستگاه در نظر گرفته شد. هفت نوع ماشین مختلف وجود داشت. زمان پردازش کار در هر ماشین بر مبنای تولید داده قبلی تعیین شد، زمان تحویل از رابطه زیر محاسبه می‌شود.

$$d_j = r_j \times DT + \text{random}[LB(\text{Voltage}_j, \text{Capacity}_j), UB(\text{Voltage}_j, \text{Capacity}_j)]$$

که Dt پارامتر سختی موعد تحویل، r_j دسترسی قبلی کار نوع j ام می‌باشد. LB ولتاژ و ظرفیت محدوده پایین و UB ولتاژ و ظرفیت محدوده بالای کار j ام بر مبنای داده تولید شده قبلی است. تمام الگوریتم‌ها بر حسب VB.NET کدگذاری شده و با یک پردازشگر اینتل core2 با 4 گیگابایت رم و فرکانس 2.4 گیگاهرتز اجرا شده است.

تست‌های مختلف آزمایشی پایلوت برای انتخاب بهترین پارامتر برای الگوریتم ژنتیک اجرا شده است. اندازه جمعیت اولیه روی 100 تنظیم شده است و نرخ crossover و نسبت جهش روی 0.4 و 0.3 تنظیم شد. مسائل مورد آزمایش ($N=100,250,500$) بر مبنای توزیع ترتیب قبلی، تولید شدند تا الگوریتم توانایی ارزیابی عملی را داشته باشد. پارامترهای آزمایشی دقیق در جدول (4) خلاصه شده‌اند.

⁶Roulette wheel method.

انحراف معیار نسبی (RDI) برای محاسبه نتایج اعمال شد (Akshabi, Tvakkoli- Mogaddam 2014) و RPD از رابطه زیر محاسبه می شود:

$$RDI_k = \frac{F_k - Min_k}{Max_k - Min_k} \times 100$$

که F_k میزان دیرکرد کلی به دست آمده برای آزمایش k ام است و Max_k و Min_k بهترین و بدترین راه حل در آزمایش k ام هستند.

5.2. نتایج آزمایشگاهی.

برای کارآیی عملکرد روش های جستجوی احتمالاتی، آزمایش های شبیه سازی 30 بار تکرار شد. کارآیی متوسط و انحراف معیار نسبی تمامی الگوریتم ها که با 30 بار تکرار محاسبه شد، در جداول (5-7) آمده است. زمانی متوسط CPU برای محاسبه و ارزیابی هر الگوریتم نیز در جداول (5-7) نشان داده شده است. زمان CPU برای الگوریتم ژنتیک چندگانه و الگوریتم ژنتیک معمولی یکسان است، زیرا هر دو الگوریتم از شرط پایان یکسانی استفاده می کنند. کارآیی تمامی الگوریتم ها بر حسب RDI که از ANOVA یک-طرفه و نمودارهای فاصله اطمینان 95٪ استفاده می کند، تحلیل شده است.

برای مقایسه آماری کارایی متوسط الگوریتم ها تحلیل ANOVA یک طرفه RDI استفاده شد (Rabee, Rad, 2014). نتایج تحلیل ANOVA برای مسائل با ابعاد کوچک، متوسط و بزرگ در جدول (8) نشان داده شده است. نتایج نشان می دهد که مقدار P برای هر حالت نزدیک به صفر است. بنابراین، RDI های حداقل یک الگوریتم به طور قابل توجهی متفاوت است.

همچنین برای ارزیابی اهمیت نتایج، فواصل اطمینان 95٪ در شکل های (10-12) نشان داده شده است. این شکل ها نشان می دهد که اختلاف معنا داری بین متوسط کارایی الگوریتم ها بر حسب RDI وجود دارد.

Parameters	Small size	Medium size	Large size
Number of jobs	100	250	500
Number of machines	30 (Stage 1), 12 (Stage 2)		
Number of experiments	30	30	30
Maximum CPU time (s)	2000	6,000	50,000
Due date tightness	0.3	1.0	1.0
Population	100		
Genetic algorithm parameters	Mutation rate (0.3), Crossover rate (0.4)		
N_Threshold	8		

جدول (4): پارامترهای مورد آزمایش با جزئیات کامل.

Algorithm	Average objective function	Average RDI	Average computation time (s)
Genetic algorithm	1410.3	30.71377	2000
Proposed algorithm	1396.533	0.10929	2000
NEH algorithm	1396.8	1.039641	58.43179
EDD	1431.933	76.09123	0.304879
SLK	1428.933	68.64578	0.249955
CR	1428.6	65.62185	0.293273
COVERT	1427.033	68.87144	0.481561
MDD	1430.333	73.43126	0.503761

جدول (5): خلاصه نتایج برای ابعاد کوچک.

Algorithm	Average objective function	Average RDI	Average computation time (s)
Genetic algorithm	2076.7	4.968086	6000
Proposed algorithm	2073.133	2.732215	6000
NEH algorithm	2125.167	32.21516	3027.462
EDD	2204.433	76.75813	0.557202
SLK	2206.367	77.64346	0.827727
CR	2229.933	88.46668	0.611689
COVERT	2177.133	62.98672	1.018642
MDD	2200	74.86941	1.160462

جدول (6): خلاصه نتایج ابعاد متوسط

Algorithm	Average objective function	Average RDI	Average computation time (s)
Genetic algorithm	28120.6	37.55309	50000
Proposed algorithm	27637.8	0	50000
NEH algorithm	28603.3	75.0182	42333.95
EDD	28820.6	92.26373	13.78962
SLK	28895.73	98.11295	16.4751
CR	28780.73	89.57702	13.19096
COVERT	28201	44.208	19.51381
MDD	28302.63	51.76036	14.16533

جدول (7): خلاصه نتایج ابعاد بزرگ.

Size	Source	df	SS	MS	F	P value
Small	Factor	6	200114.97	33352.495	65.132	0.000
	Error	203	103950.901	512.073		
	Total	209	304065.871			
Medium	Factor	6	231130.823	38521.804	128.3	0.000
	Error	203	60950.116	300.247		
	Total	209	292080.939			
Large	Factor	6	237214.294	39535.716	660.103	0.000
	Error	203	12158.334	59.893		
	Total	209	249372.628			

جدول (8): جدول ANOVA برای RDI مسائل کوچک، متوسط و بزرگ.

References

- Akshabi, M., Tavakkoli-Moghaddam, R., & Rahnamay-Roodposhti, F. (2014). A hybrid particle swarm optimization algorithm for a no-wait flow shop scheduling problem with the total flow time. *The International Journal of Advanced Manufacturing Technology*, 70(5-8), 1181-1188.
- Arthanari, T., & Ramamurthy, K. (1971). An extension of two machines sequencing problem. *Opsearch*, 8(1), 10-22.
- Botta-Genoulaz, V. (2000). Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness. *International Journal of Production Economics*, 64(1), 101-111.
- Bozejko, W., Pempera, J., & Smutnicki, C. (2013). Parallel tabu search algorithm for the hybrid flow shop problem. *Computers & Industrial Engineering*, 65(3), 466-474.
- Costa, A., Cappadonna, F. A., & Fichera, S. (2014). A novel genetic algorithm for the hybrid flow shop scheduling with parallel batching and eligibility constraints. *The International Journal of Advanced Manufacturing Technology*, 75(5-8), 833-847.
- Engin, O., Ceran, G., & Yilmaz, M. K. (2011). An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems. *Applied Soft Computing*, 11(3), 3056-3065.
- Fattahi, P., Hosseini, S. M. H., Jolai, F., & Tavakkoli-Moghaddam, R. (2014). A branch and bound algorithm for hybrid flow shop scheduling problem with setup time and assembly operations. *Applied Mathematical Modelling*, 38(1), 119-134.
- Feng, H., Lu, S., & Li, X. (2009). Genetic algorithm for hybrid flow-shop scheduling with parallel batch processors. Paper presented at the Information Engineering, 2009. ICIE'09. WASE International Conference on Information Engineering.
- Gen, M., & Cheng, R. (1997). Genetic algorithms and engineering design. Wiley.
- Gupta, J. N. (1988). Two-stage, hybrid flowshop scheduling problem. *Journal of the Operational Research Society*, 39(4), 359-364.
- Gupta, J., Hariri, A., & Potts, C. (1997). Scheduling a two-stage hybrid flow shop with parallel machines at the first stage. *Annals of Operations Research*, 69, 171-191.
- Holland, J. H. (1975). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. U Michigan Press.
- Jun, S., & Park, J. (2013). The development of a heuristic algorithm for the transformer production scheduling. The Journal of Korea Society for Simulation Autumn Conference, 6-8. <http://simul.hosting.bizfree.kr/files/full_kss_2013autumn.pdf> (accessed April 10, 2014).
- Kochhar, S., & Morris, R. J. (1987). Heuristic methods for flexible flow line scheduling. *Journal of Manufacturing Systems*, 6(4), 299-314.
- Korytkowski, P., Wiśniewski, T., & Rymaszewski, S. (2013). An evolutionary simulation-based optimization approach for dispatching scheduling. *Simulation Modelling Practice and Theory*, 35, 69-85.
- Li, J. Q., Pan, Q. K., & Wang, F. T. (2014). A hybrid variable neighborhood search for solving the hybrid flow shop scheduling problem. *Applied Soft Computing*, 24, 63-77.

- Liao, C.-J., Tjandradjaja, E., & Chung, T.-P. (2012). An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem. *Applied Soft Computing*, 12(6), 1755–1764.
- Linn, R., & Zhang, W. (1999). Hybrid flow shop scheduling: a survey. *Computers & Industrial Engineering*, 37(1), 57–61.
- Lopez, P., & Roubellat, F. (2008). *Production scheduling*. Wiley.
- Moradinasab, N., Shafaei, R., Rabiee, M., & Ramezani, P. (2013). No-wait two stage hybrid flow shop scheduling with genetic and adaptive imperialist competitive algorithms. *Journal of Experimental & Theoretical Artificial Intelligence*, 25(2), 207–225.
- Naderi, B., Gohari, S., & Yazdani, M. (2014). Hybrid flexible flowshop problems: Models and solution methods. *Applied Mathematical Modelling*, 38(24), 5767–5780.
- Pinedo, M. (2008). *Scheduling: Theory, algorithms, and systems* (3rd ed.). Springer.
- Rabiee, M., Rad, R. S., Mazinani, M., & Shafaei, R. (2014). An intelligent hybrid meta-heuristic for solving a case of no-wait two-stage flexible flow shop scheduling problem with unrelated parallel machines. *The International Journal of Advanced Manufacturing Technology*, 71(5–8), 1229–1245.
- Ribas, I., Leisten, R., & Framiñan, J. M. (2010). Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Computers & Operations Research*, 37(8), 1439–1454.
- Rossi, A., Pandolfi, A., & Lanzetta, M. (2014). Dynamic set-up rules for hybrid flow shop scheduling with parallel batching machines. *International Journal of Production Research*, 52(13), 3842–3857.
- Ruiz, R., & Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, 165(2), 479–494.
- Ruiz, R., & Vázquez-Rodríguez, J. A. (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research*, 205(1), 1–18.
- Sun, L., & Yu, S. (2015). Scheduling a real-world hybrid flow shop with variable processing times using Lagrangian relaxation. *The International Journal of Advanced Manufacturing Technology*, 1–10.
- Tari, F. G., & Olfat, L. (2013). Heuristic rules for tardiness problem in flow shop with intermediate due dates. *The International Journal of Advanced Manufacturing Technology*, 1–13.