نشریه پژوهش‌های مهندسی صنایع در سیستم‌های تولید

نشریه پژوهش‌های مهندسی صنایع در سیستم‌های تولید

IERPS

ISSN: 2345-2269

سال چهارم، شماره نهم، ویژه نامه یازدهمین و دوازدهمین کنفرانس بین المللی مهندسی صنایع، صفحه ۲۷۱-۲۷۹

www.ier.basu.ac.ir

حل مساله زمانبندی جریان کارگاهی برگشت‌پذیر بدون وقفه

سعید طسوجی حسن‌پور [۱]، محمدرضا امین ناصری [۲،*]، ابوالفضل آدرسی [۳]

۱. دانشجوی دکتری ، دانشکده مهندسی صنایع و سیستم‌ها، دانشگاه تربیت مدرس، تهران.

۲. دانشیار، دانشکده مهندسی صنایع و سیستم‌ها، دانشگاه تربیت مدرس، تهران.

۳. کارشناسی ارشد مهندسی صنایع، دانشگاه صنعتی خواجه نصیرالدین طوسی، تهران.

اطلاعات مقاله

تاریخچه مقاله:

دریافت ۱۳۹۴/۱۱/۰۷

پذیرش ۱۳۹۵/۱۲/۱۱

کلمات کلیدی:

جریان کارگاهی بدون وقفه

جریان کارگاهی برگشت‌پذیر

الگوریتم ژنتیک

الگوریتم شبیه‌سازی تبرید

خلاصه

در این مقاله زمانبندی مساله جریان کارگاهی برگشت‌پذیر بدون وقفه با هدف کمینه‌سازی حداکثر زمان تکمیل کارها بررسی می‌شود. محیط های جریان کارگاهی برگشت‌پذیر به نوعی چیدمان جریان کارگاهی گفته می‌شود که در آن حداقل یک کار می‌بایست یک یا چند مرحله را بیش از یکبار ملاقات کند. در مسائل جریان کارگاهی بدون وقفه مراحل انجام یک کار بر روی ماشین ها از ابتدا تا انتها بدون وقفه انجام می‌شوند. ادغام هردوی این خصوصیات در بسیاری از صنایع مانند صنایع رباتیک دارد کاربرد دارد که در ادبیات بصورت مجزا مورد بررسی قرار نگرفته است. در این مقاله برای مساله زمانبندی جریان کارگاهی برگشت‌پذیر بدون وقفه مدل ریاضی ارائه شده است. برای مسایل با ابعاد کوچک نتایج حاصل از نرم‌افزار GAMS با نتایج حاصل از الگوریتم‌های ژنتیک و شبیه‌سازی تبرید مقایسه گردیده است و برای ابعاد بزرگ نتایج حاصل از الگوریتم‌های ژنتیک و شبیه‌سازی تبرید با یکدیگر مقایسه گردیده‌اند. نتایج محاسباتی نشان داد که در کل الگوریتم SA نسبت به GA، از لحاظ بدست آوردن جواب بهینه یا نزدیک به بهینه الگوریتمی کاراتر می‌باشد.

۱- مقدمه

بسیاری از چیدمان‌های کارخانجات به شکل جریان کارگاهی و تولید کارگاهی هستند که کارها در آنها بدون اینکه از یک مرحله یا ایستگاه کاری بیش از یکبار عبور نمایند پردازش می‌شوند. هرچند در برخی از صنایع مانند تولید نیمه هادی ها، طراحی محصول ممکن است به گونه‌ای باشد که کارها در کارگاه دوباره بر روی ماشین‌ها از ابتدا به چرخش درآیند و یا از برخی از مراحل یا ایستگاههای کاری بیش از یکبار عبور نمایند. بطور کلی جریان کارگاهی برگشت‌پذیر به نوعی چیدمان جریان کارگاهی گفته می‌شود که در آن حداقل یک کار می‌بایست یک یا چند مرحله را بیش از یکبار ملاقات کند. در صنایع تولیدی نمونه‌های فراوانی در زمینه جریانهای برگشت‌پذیر وجود دارد. به عنوان مثال، لیتوگرافی نوری یکی از گامهای پیچیده در ساخت ویفر، نوعی تراشه کامپیوتر و در تولید نیمه هادی‌ها می‌باشد. لیتوگرافی نوری یک فرآیند بصری است که به منظور نگاشت چندین لایه از طرحهای مداری بر روی ویفر سیلیکونی مورد استفاده قرار می‌گیرد که در طی این فرآیند چندین بار می‌تواند از این مرحله عبور نماید. مثال دیگر مونتاژ و تست کردن مدارهای الکترونیکی که بر روی همدیگر قرار می‌گیرند، می‌باشد. هروقت که یک مدار جدید به مجموعه اضافه می‌شود، می‌بایست از یک سری از ماشین‌آلات مجددا عبور نماید. از نمونه‌های دیگر در این زمینه می‌توان به برد مدارهای چاپی (PCB)، کارگاههای چرخه ای دو ماشینه، پردازش سیگنالها و برنامه‌ریزی تولید برای تسهیلات هاب محور اشاره نمود. در مسائل جریان کارگاهی بدون وقفه مراحل انجام یک کار بر روی ماشین‌ها از ابتدا تا انتها بدون وقفه انجام می‌شوند، یعنی وقتی پردازش کاری بر روی ماشین اول شروع شد، باید تا وقتی که پردازشش بر روی آخرین

* نویسنده مسئول: محمدرضا امین ناصری

تلفن: ۸۲۸۸۳۳۴۴-۰۲۱؛ پست الکترونیکی: amin_nas@modares.ac.ir

ماشین تکمیل می‌شود بصورت پیوسته و بدون وقفه روی ماشین‌ها حرکت کند. دو دلیل اصلی بروز این گونه مسائل در محیط‌های تولیدی شامل ماهیت فرایندها و عدم وجود انبار در بین ایستگاه‌ها یا ماشین آلات می‌باشد. بعضی از فرایندها، برای رسیدن به نتیجه دلخواه و جلوگیری از تغییرات نامطلوب در مواد، دما یا خصوصیات دیگر مواد (مثلا چسبندگی) نیاز به انجام کارها به صورت پشت سرهم و بدون وقفه دارند.

در این مقاله، مساله زمانبندی جریان کارگاهی برگشت‌پذیر بدون وقفه با در نظر گرفتن برگشت به عقب در هر توالی برای کارها مورد بررسی قرار گرفته است و از الگوریتمهای فراابتکاری ژنتیک و شبیه سازی تبرید برای حل آن بهره گرفته شده است. سایرقسمتهای مقاله به شرح زیر می‌باشد:

در بخش دوم مرور ادبیات پیرامون موضوع مورد بررسی در این مقاله آورده شده است. در بخش سوم مقاله به تعریف مساله مورد بررسی و مدلسازی آن پرداخته شده است. بخشهای چهارم و پنجم این مقاله مربوط به حل مساله پیشنهادی با استفاده از الگوریتمهای فراابتکاری ژنتیک و شبیه سازی تبرید می‌باشد. بخش ششم مربوط به نتایج بدست آمده حاصل از اجرای الگوریتمهای فراابتکاری و همچنین اعتبار سنجی الگوریتمهای پیشنهادی می‌باشد. در پایان، در بخش هفتم نتیجه گیری و پیشنهاد برای پژوهش های آتی ارائه می‌گردد.

۲- مرور ادبیات

اخیرا مقالات متعددی در زمینه زمانبندی جریان کارگاهی برگشت‌پذیر و جریان کارگاهی بدون وقفه به چاپ رسیده است. هرچند که در نظر گرفتن هردوی این موارد با توجه به کاربردهای فراوانی که در زمینه صنایع رباتیک دارد، مورد توجه محققان واقع نشده است. جریانهای کارگاهی رباتیک بصورت گسترده در صنایع استیل و الکترونیک مورد استفاده قرار می‌گیرند که در آنها با توجه به ویژگی‌های خود تکنولوژی، بعد از اینکه پردازش بر روی یک ماشین به اتمام رسید، بلافاصله باید از آن ماشین جدا شده و بدون وقفه به ماشین بعدی در فرآیند منتقل شود. در غیر اینصورت ممکن است که قطعات معیوبی به وجود آیند.

هال و اسریسکاندراجا [۱] در سال ۱۹۹۶ مروری جامع بر روی مسائل زمانبندی بدون وقفه انجام دادند و پیشنهاداتی جهت انجام به عنوان تحقیقات آتی ارائه دادند. الدواسیان و الله وردی [۲] در سال۲۰۰۳ الگوریتم های فرا ابتکاری بر پایه شبیه سازی تبرید و ژنتیک را به منظور حداقل کردن حداکثر زمان تکمیل کارها در مسئله جریان کارگاهی بدون وقفه ارائه دادند. گوپتا و استفورد [۳] در سال ۲۰۰۶ یک مرور جامع روی مسائل زمان بندی جریان کارگاهی بدون وقفه طی پنجاه سال گذشته را انجام داده اند. چوی و کیم [۴] مساله کمینه سازی حداکثر زمان تکمیل کارها را مدنظر قرار دادند و چندین الگوریتم ابتکاری کارا نیز برای مسائل مورد نظر پیشنهاد نمودند. چن و همکاران [۵] نیز یک الگوریتم ژنتیک ترکیبی را برای همان مساله پیشنهاد کردند. جینگ و همکاران [۶] مساله زمانبندی جریان کارگاهی برگشت پذیر دو ماشینه را با هدف کمینه سازی حداکثر زمان تکمیل کارها مورد بررسی نموده و چندین روش ابتکاری برای حل مساله توسعه دادند. هوانگ و همکاران [۷] مساله زمانبندی جریان کارگاهی دو مرحله ای برگشت پذیر با مطالعه موردی کارگاه ویفر سازی (نوعی تراشه کامپیوتر) را مورد مطالعه قرار دادند. آنها از یک الگوریتم بهینه سازی ذرات گسسته برای کمینه سازی دیرکرد و زودکرد مساله مورد بررسی استفاده نمودند. لی و همکاران [۸] یک الگوریتم یادگیرنده ی افزایشی مبتنی بر جمعیت را برای مساله ی زمانبندی جریان کارگاهی برگشت پذیر m ماشینه با درنظر گرفتن تابع هدف حداکثر زمان تکمیل کارها را ارائه نمودند. جینگ و همکاران [۹] یک الگوریتم ابتکاری را برای حل مساله جریان کارگاهی برگشت پذیر با استفاده از تکنیک k-insertion برای کمینه سازی جریان کل ارائه نمودند. چوی و کیم [۱۰] از روش شاخه و کران و نیز سه روش ابتکاری جهت حل مساله با تابع هدف مجموع دیرکرد استفاده نمودند.

همانطوری که بیان شد، مسایل زمانبندی بدون انتظار در آن دسته از محیط های تولیدی رخ می‌دهد که در آن یک کار می‌بایست از آغاز تا پایان بر روی یک ماشین یا در بین ماشینها بدون وقفه مورد پردازش قرار گیرد. اولین کارها در زمینه ی زمانبندی بدون وقفه را می‌توان به کارهای آرتاناری در سال های ۱۹۷۱ و ۱۹۷۴ مربوط دانست [۱۱]. تحقیقات بسیاری برای حل مساله زمانبندی جریان کارگاهی بدون وقفه با در نظر گرفتن معیارهای متفاوتی مانند حداکثر زمان تکمیل کارها و زمان در جریان کل صورت گرفته که به ارائه الگوریتم های ابتکاری و فرا ابتکاری زیادی منجر گشته است. از جمله آخرین کارها در این حوزه می‌توان به مقاله آدرسی و همکاران [۱۲] اشاره نمود که آنها مسئله زمانبندی گروهی در جریان کارگاهی منعطف بدون وقفه را مورد بررسی قرار داده و برای حل آن الگوریتم‌های فراابتکاری ژنتیک و شبیه سازی تبرید را پیشنهاد داده اند. با توجه به تحقیقات انجام شده، ضرورت انجام تحقیق در زمینه زمانبندی در جریان کارگاهی برگشت پذیر بدون وقفه احساس می‌شود.

۳- تعریف مسئله مورد بررسی

در مساله زمان‌بندی جریان کارگاهی (FSS) فرض بر این است که مجموعه‌ای از کارها $J=\{1,.....,n\}$ وجود دارد که می‌بایستی بر روی مجموعه از ماشین‌ها $M=\{1,.....,m\}$ پردازش شود. در این مساله m ماشین به صورت سری قرار دارند که هر یک از کارها باید بر روی این ماشین‌ها پردازش شوند و مسیر پردازش همه کارها یکسان است بدین معنا که هر کار ابتدا بر روی ماشین ۱، سپس روی ماشین ۲ و به همین ترتیب بر روی بقیه ماشین های تا ماشین

m پردازش می‌شود. برای مساله فرض جایگشت در نظر گرفته شده که این فرض باعث می‌شود تا کارها با یک ترتیب مشابه بر روی هر یک از ماشین‌ها پردازش شوند. مساله زمان بندی جریان کارگاهی با فرض وجود جایگشت با نام مساله زمان‌بندی جریان کارگاهی جایگشتی (PFSS) شناخته می‌شود. فرض دیگری که برای مساله در نظر گرفته شده محدودیت زمان انتظار است که این فرض در محیط های چند ماشینه نظیر جریان کارگاهی و تولید کارگاهی می‌تواند رخ دهد. این فرض باعث می‌شود تا پردازش کارها بر روی ماشین‌ها بدون وقفه و زمان انتظار بین ماشین‌ها انجام شود. علاوه بر فرض‌هایی که در بالا اشاره شد این فرض نیز وجود دارد که کارها بتوانند بعد از انجام پردازش بر روی چند ماشین، به چند مرحله عقب‌تر برگردند و عمل پردازشی خود را انجام دهند. مساله با این فرضیات مساله زمان‌بندی جریان کارگاهی جایگشتی(PFSS) برگشتی و زمان انتظار صفر (بدون وقفه) نامیده می‌شود.

این مساله طبق نماد گذاری گراهام و همکاران [۱۳] به صورت $FSS \mid pmu, rcrc, no-wait \mid C_{\max}$ نمایش داده می‌شود. تابع هدف در نظر گرفته شده برای مساله مورد بررسی کمینه سازی بیشینه زمان تکمیل کارها بر روی ماشین‌ها در نظر گرفته شده است.

علاوه بر فرضیاتی که قبلا اشاره شد فرضیات زیر نیز برای مساله مورد بررسی در نظر گرفته شده است.

- همزمان دو عمل از یک کار قابل انجام نیست.
- انقطاع وجود ندارد، یعنی یک کار تا زمان تکمیل شدن پردازش خود بر روی ماشین مربوطه باقی می‌ماند.
- در هنگام انجام کارها انصراف وجود ندارد، بدین معنی که اگر یک عملیات از کاری مورد پردازش قرار گرفت، عملیات‌های بعدی کار نیز حتما می بایستی پردازش شوند.
- مدت زمان پردازش هر کار مستقل از ترتیب انجام کارها است.
- مدت زمان آماده سازی، مستقل از ترتیب انجام کارها است و در مدت زمان پردازش کارها در نظر گرفته شده است.
- مدت زمان حمل و نقل بین ماشین‌ها ناچیز است.
- ماشین‌ها ممکن است بیکاری داشته باشند.
- هر ماشین بیش از یک کار را به طور همزمان پردازش نمی-کند.
- محدودیت‌های فنی شناخته شده‌اند و تغییر ناپذیرند.
- حالت تصادفی وجود ندارد. بدین معنی که زمان‌های پردازش، زمان‌های آماده سازی، زمان‌های ورود قطعات و تعدادکارها مقادیری قطعی دارند.
- ماشین‌ها در دوره برنامه‌ریزی، پیوسته در دسترس هستند.

۴- مدل‌سازی مسئله

نمادهای بکار گرفته شده برای مدل سازی مساله در زیر معرفی شده‌اند.

اندیس‌ها

i: اندیس کارها $i = \{1, 2, \ldots, n\}$

j: اندیس عمل‌ها $j = \{1, 2, \ldots, n_i\}$

k: اندیس ماشین‌ها $k = \{1, \ldots, m\}$

h: اندیس توالی کارها بر روی هر یک از ماشین‌ها $h = \{1, 2, \ldots, h_k\}$

پارامترها

p_{ij}: مدت زمان پردازش عمل j از کار i .

a_{jk}: اگر عمل j از کارها به ماشین k احتیاج داشته باشد برابر ۱ در غیر این‌صورت برابر ۰ است.

Re_j: اگر عمل بعد از عمل j برگشت به عقب داشته باشد برابر با ۱ و در غیر اینصورت برابر با ۰ است.

SRe_j: اگر عمل بعد از عمل j برگشت به عقب داشته باشد برابر با شماره ماشینی است که عمل بعدی باید بر روی آن پردازش شود و در غیر اینصورت برابر با ۰ خواهد بود.

متغیرها

$C_{\max}$: بیشینه زمان تکمیل کارها

s_{ij}: زمان شروع پردازش عمل j از کار i

pb_{kh}: زمان پردازش کاری که در اولویت h ماشین k قرار دارد.

sb_{kh}: زمان شروع پردازش کاری که در اولویت ماشین k اختصاص داده شده است.

r_{ijk}^{h}: اگر عمل j از کار i که به ماشین k احتیاج دارد و در جایگاه h قرار گیرد برابر ۱، در غیر این‌صورت برابر ۰ است.

$$\text{Minimize} \quad C_{max} \tag{۱}$$

$$\sum_{h} r_{ijk}^{k} = a_{jk\,;\,\forall i,j,k} \tag{۲}$$

$$\sum_{i}\sum_{j} r_{ijk}^{h} \le 1;\ \forall k, h \tag{۳}$$

$$r_{i,j-1,k-1}^{h} = r_{ijk}^{h};\ \forall i, j > 1, k > 1, h \qquad \mathrm{Re}(j-1) \ne 1 \tag{۴}$$

$$r_{i,j-1,k-1}^{h} = r_{ijk}^{h};\ \forall i, j, k > 1, k, h \qquad \mathrm{Re}(j-1) = 1, S\,\mathrm{Re}(j) = k \tag{۵}$$

$$p_{ij} * r_{ijk}^{h} \le pb_{k,h};\ \forall i, j, k, h \tag{۶}$$

$$s_{i,j-1} + \sum_{k}\sum_{l}\sum_{m} p_{i,j-1,k} * r_{i,j-1,k}^{ih} = s_{i,j;}\ \forall i, j > 1 \tag{۷}$$

$$sb_{k,h-1} + pb_{k,h-1} \le sb_{k,h};\ \forall k,\ h > 1 \tag{۸}$$

$$s_{ij} \le (1 - r_{ijk}^{h}) * M + sb_{kh};\ \forall i, j, k, h \tag{۹}$$

$$sb_{kh} \le (1 - r_{ijk}^{h}) * M + s_{ij};\ \forall i, j, k, h \tag{۱۰}$$

$$C_{\max} \ge s_{ij} + \sum_{k}\sum_{h} p_{ij} * r_{ijk}^{h};\ \forall i, j \tag{۱۱}$$

$$r_{ijk}^{h} = \{0,1\}, s_{i,j} \ge 0, sb_{kh} \ge 0,\ \forall i, j, k, h \tag{۱۲}$$

در این مدل رابطه (۱)، تابع هدف مساله است که بیشینه

زمان تکمیل کارها را کمینه می‌کند. محدودیتهای (۲) و (۳) تضمین می‌کنند عمل‌های هر یک از کارها بر روی هر یک از ماشین‌ها با توجه به میسر پردازش کارها در یک جایگاه از توالی کار قرار گیرد. محدودیت های (۴) و (۵) به منظور رعایت فرض جایشگت مساله جریان کارگاهی به مدل اضافه شده است. محدودیت (۶) برای تعیین زمان پردازش کاری که در جایگاه ماشین مربوطه قرار گرفته است به مدل اضافه شده است. محدودیت (۷) زمان شروع عمل‌هایی که در مسیر پردازشی هر کار قرار دارند را تنظیم می‌کند، به بیان دیگر پردازش یک عمل از کار تا زمانی که پردازش عمل‌های قبلی آن به پایان نرسیده، شروع نخواهد شد. محدودیت (۸) برای تعیین زمان شروع کارها بر روی ماشین‌ها به مدل اضافه شده و بیان می‌کند تا زمانی که پردازش کار قبلی به اتمام نرسیده است، پردازش کار جاری نمی‌تواند شروع شود. محدودیت‌های (۹) و (۱۰) نیز برای تنظیم زمان شروع عمل-های هر کار و زمان شروع کارها بر روی ماشین‌ها به مدل اضافه شده است. محدودیت (۱۱) برای محاسبه بیشینه زمان تکمیل کارها به مدل اضافه شده است. محدودیت (۱۲) ماهیت متغیرهای مدل را تعیین می‌کند.

۵- حل مسئله زمانبندی جریان کارگاهی برگشت پذیر بدون وقفه

پن و چن [۷] نشان دادند که مساله زمانبندی جریان کارگاهی برگشت‌پذیر با در نظر گرفتن تابع هدف کمینه‌سازی حداکثر زمان تکمیل کارها حتی برای مسایل با دو ماشین نیز Np-hard می‌باشد. می‌توان نشان داد که مساله مدنظر در این مقاله نیز Np-hard است. بنابراین برای حل مساله علاوه بر حل دقیق برای ابعاد کوچک، الگوریتم‌های فراابتکاری ژنتیک و شبیه‌سازی تبرید نیز برای حل مساله در ابعاد کوچک و بزرگ ارائه شده است. در ادامه به معرفی اجزای الگوریتم‌های فراابتکاری پیشنهادی پرداخته می‌شود.

۵-۱- حل مسئله مورد نظر با استفاده از الگوریتم ژنتیک

اولین گام در این الگوریتم ارتباط دادن مساله اصلی با ساختار الگوریتم ژنتیک (GA) است. برای مساله مورد بررسی در این مقاله با توجه به ویژگی جایگشت که برای مساله وجود دارد، کروموزومی که برای نمایش جواب مساله بکار می‌رود طولی برابر با تعداد کارهای مساله و مستقل از تعداد ایستگاه های کاری دارد. بعبارتی کروموزوم با توالی از شماره کارها به‌صورت $\sigma=([1],[2],....,[n])$ نمایش داده می‌شود که شماره هر کار تنها یکبار در هر کروموزوم تکرار می‌شود. اولویت پردازش هر یک از کارها بر روی ماشین‌ها به ترتیب ظاهر شدن آن‌ها بر روی کروموزوم از چپ به راست می‌باشد و یا بعبارتی اولویت پردازش عمل‌های کارها بر روی هر یک از ماشین‌ها بر اساس زودترین رخداد آن‌ها در بردار توالی σ است. در شکل ۱ کروموزومی برای یک مساله با ۶ کار نشان داده شده است. در این کروموزوم ترتیب پردازش کارها بر روی ماشین‌ها به این صورت است که ابتدا کار ۱ بر روی همه ماشین‌ها پردازش می‌شود و سپس کار ۶ با توجه به زمان اتمام پردازش کار ۱ بر روی هر یک از ماشین‌ها و این ویژگی که کار در بین هیچ دو ماشینی دارای زمان انتظار نمی‌باشد بر روی همه ماشین‌ها با توجه به ترتیب پردازش بر روی همه ماشین‌ها پردازش می‌شود. این رویه تا پردازش و تکمیل همه کارها بر روی ماشین‌ها ادامه می‌یابد.

۱	۶	٤	٥	۲	۳

شکل(۱): نحوه نمایش کروموزوم

این نحوه کدبندی باعث می‌شود تا هر جایگشتی از ژن‌های کروموزوم به یک زمان‌بندی قابل قبول برای مساله بر روی هر یک از ماشین‌ها تبدیل شود.

نقش تابع برازندگی، نشان دادن میزان برازندگی هر کروموزوم است. تابع برازندگی پایه و اساس فاز انتخاب را تشکیل می‌دهد. برای مساله مورد بررسی در این مقاله از آنجایی که تابع هدف مساله کمینه سازی می‌باشد، میزان برازندگی هر کروموزوم در هر نسل برابر با انحراف مقدار تابع هدف متناظر با کروموزوم، با بدترین تابع هدف در آن نسل به اضافه ۱ در نظر گرفته می‌شود.

از جمله متداول‌ترین روش‌های انتخاب می‌توان چرخ رولت، تورنمنت، تصادفی و رتبه‌بندی را نام برد. در الگوریتم GA پیشنهادی از روش انتخاب چرخ رولت استفاده شده است.

از جمله متداول‌ترین روش‌های انجام تقاطع می‌توان به تقاطع دو نقطه‌ای اشاره نمود که در این مقاله مورد استفاده قرار گرفته است و به شرح زیر صورت می‌پذیرد:

- ژن‌های بین دو نقطه شکست والد ۱ را در فرزند ۲ کپی کنید.
- ژن‌های بین دو نقطه شکست والد ۲ را در فرزند ۱ کپی کنید.
- ژن‌های متناظر با ژن‌های کپی شده در فرزند ۱ را از والد ۱ حذف کنید.
- ژن‌های متناظر با ژن‌های کپی شده در فرزند ۲ را از والد ۲ حذف کنید.
- ژن‌های باقی مانده از والد ۱ را در ژن‌های تهی فرزند ۱ قرار دهید.
- ژن‌های باقی مانده از والد ۲ را در ژن‌های تهی فرزند ۲ قرار دهید.

در مساله مورد بررسی، برای اجرای عملگر جهش دو ژن از کروموزوم که دارای مقدارهای متفاوتی هستند، به‌صورت تصادفی انتخاب و مقادیر آن‌ها با یکدیگر جابجا می‌شود.

به منظور جستجوی دقیق‌تر در اطراف جواب‌های خوب (پرقدرت سازی) و همچنین جستجو در بخش‌های بازدید نشده فضای حل (متنوع سازی)، از یک جستجوی محلی در هر تکرار الگوریتم استفاده می‌شود. برای این منظور به ازای هر کروموزومی که به نسل بعدی راه می‌یابد، یک جستجوی محلی در اطراف

کروموزوم انجام می‌شود.

آماده سازی الگوریتم یعنی تعیین تعداد جمعیت اولیه، تعداد نسل‌ها، درصد تقاطع و جهش و باز تولید (درصد نخبه) که این مقادیر برای مسائل با اهداف مختلف، متفاوت است. این آماده سازی‌ها برای الگوریتم در بخش تنظیم پارامتر برای مساله مورد بررسی انجام خواهد شد. شرایط خاتمه در نظر گرفته شده برای الگوریتم را رسیدن به تعداد معینی از نسل‌ها در نظر گرفته شده است.

۵-۲- تنظیم پارامتر الگوریتم ژنتیک

در این مقاله مسائل نمونه در دو دسته کوچک و بزرگ تقسیم بندی شده و مورد آزمایش قرار گرفتند. مسایل با تعداد ۴، ۶ و ۸ کار به عنوان مسائل با ابعاد کوچک و مسائل با تعداد ۲۰، ۳۰ و ۴۰ کار به عنوان مسائل با ابعاد بزرگ انتخاب گردیده‌اند. برای تنظیم پارامترهای مربوط به الگوریتم ژنتیک مورد استفاده در این مقاله از روش تاگوچی استفاده شده است که پارامترهای بهینه برای مسائل کوچک و بزرگ در جداول (۱) و (۲) آورده شده اند:

جدول (۱): تنظیم پارامترهای مربوط به مسائل کوچک (الگوریتم ژنتیک)

جمعیت اولیه	تعداد نسل	درصد تقاطع	درصد جهش	درصد بازتولید	تعداد جمعیت جستجوی محلی
۱۰۰	۵۰	۰.۸	۰.۱۳	۰.۰۷	۵

جدول (۲): تنظیم پارامترهای مربوط به مسائل بزرگ (الگوریتم ژنتیک)

جمعیت اولیه	تعداد نسل	درصد تقاطع	درصد جهش	درصد بازتولید	تعداد جمعیت جستجوی محلی
۲۵۰	۱۵۰	۰.۸	۰.۱	۰.۱	۷

۵-۳- حل مسئله مورد نظر با استفاده از الگوریتم شبیه سازی تبرید

الگوریتم شبیه سازی تبرید (SA) یک تکنیک حل قوی می‌باشد که برای مسائل بهینه‌سازی یک و چند هدفه، جواب‌های بسیار خوبی تولید می‌کند. این الگوریتم با ایجاد و ارزیابی جواب‌های متوالی، به‌صورت گام به‌گام به سمت جواب بهینه حرکت می‌کند. برای حرکت، یک همسایگی جدید به‌صورت تصادفی ایجاد و ارزیابی می‌شود. در این روش به بررسی نقاط نزدیک به نقطه داده شده در فضای جستجو پرداخته می‌شود. در صورتی که نقطه‌ی جدید، نقطه بهتری باشد به عنوان نقطه جدید در فضای جستجو انتخاب می‌شود و اگر بدتر باشد بر اساس یک تابع احتمال انتخاب می‌شود.

۵-۴- تنظیم پارامتر الگوریتم شبیه سازی تبرید

از جمله پارامترهای تأثیر گذار در این الگوریتم می‌توان به دمای اولیه و نهایی، تابع تبرید، تعداد جواب‌های پذیرفته شده و درصد خطا برای انجام تست تعادل در هر دما را نام برد. رویه تنظیم پارامتر این الگوریتم در دو مرحله انجام می‌شود. ابتدا مقادیر اولیه‌ای با استفاده از سعی و خطا برای هر یک از پارامترهای الگوریتم در نظر گرفته می‌شود. سپس در مرحله اول دو پارامتر درصد خطا (ε) و تعداد جواب‌های پذیرفته شده (N_k) جهت انجام تست تعادل در هر دما تنظیم شده و مقادیر بدست آمده برای هر یک از پارامترها در الگوریتم تثبیت می‌شود. در نهایت در مرحله دوم دمای اولیه و نهایی به همراه ثابت بولتزمن تنظیم می‌شود.

جدول (۳): مقادیر اولیه پارامترهای الگوریتم SA

پارامتر	دمای اولیه	مقدار کاهش تابع تبرید	دمای نهایی	ثابت بولتزمن
مقادیر اولیه پارامترها	۱۰۰	۰.۸	۱	۱

در این مرحله ترکیبات مختلف از پارامترهای ε و N_k برای الگوریتم در نظر گرفته شده است. نتایج محاسباتی بدست آمده حاصل از ۴ بار اجرای الگوریتم به ازای مقادیر مختلف از پارامترها، برای ۴ مساله در تکرار اول تنظیم پارامتر نشان داد که بهترین ترکیب پارامترهای ε و N_k برای مسایل در ابعاد کوچک با مقادیر ۴، ۶ و ۸ کار به ترتیب برابر با ۰.۰۰۸ و ۳ می‌باشد. این در حالی است که برای مسایل در ابعاد بزرگ با تعداد ۲۰، ۳۰ و ۴۰ کار بهترین ترکیب پارامترهای ε و N_k به ترتیب برابر با ۰.۰۰۳ و ۱۰ می‌باشد.

بعد از تنظیم پارامترهای ε و N_k در مرحله دوم به تنظیم دمای اولیه و نهایی و ثابت بولتزمن برای الگوریتم پرداخته شد. برای دمای اولیه ۳ مقدار ۵۰، ۱۰۰ و ۱۵۰، برای ثابت بولتزمن نیز ۳ مقدار ۱، ۲ و۳ و برای دمای نهایی نیز مقادیر ۱ و ۰.۲ در نظر گرفته شد. نتایج محاسباتی نشان داد که بهترین دمای اولیه برای مسایل در ابعاد کوچک برابر با ۵۰ و برای مسایل در ابعاد بزرگ ۱۰۰ می‌تواند در نظر گرفته شود. همچنین بهترین مقدار بدست آمده از آزمایشات برای دمای نهایی و ثابت بولتزمن برابر با ۱ می‌باشد.

از آنجایی که الگوریتم SA بسته به سختی مسائل در برخی از دماها به دلیل متعادل نشدن و قابل قبول نبودن تست تعادل متوقف نمی‌شود، لذا علاوه بر شرط توقف دمای نهایی، یک شرط توقف نیز بر روی تعداد کل جواب‌های پذیرفته شده و تعداد کل همسایگی‌های ایجاد شده در طول اجرای الگوریتم در نظر گرفته شده تا متوقف شدن الگوریتم تضمین شود.

۶- نتایج محاسباتی و اعتبار سنجی الگوریتمهای پیشنهادی

ارزیابی کارایی روش‌های حل ارائه شده به دو صورت انجام می‌گیرد. در حالت اول، جواب بدست آمده از روش‌های حل ارائه شده با جواب بهینه بدست آمده از مدل ریاضی در محدودیت زمانی ۳۶۰۰ثانیه برای مسائل در ابعاد کوچک مورد مقایسه قرار می‌گیرد. در حالت دوم برای مسائل در ابعاد بزرگ از آنجایی که جواب مساله با استفاده از مدل ریاضی در یک زمان معقول بدست نمی‌آید به

ارزیابی روش‌های حل پیشنهادی با یکدیگر پرداخته می‌شود.

مدل ریاضی توسعه داده شده برای مساله توسط نرم افزار GAMS مدل شده و توسط حل کننده CPLEX حل شده است. همچنین برای کد نویسی روش‌های حل ارایه شده از زبان برنامه-نویسی C++ استفاده شده است. همه محاسبات بر روی یک رایانه با مشخصات IntelCore2DuoP7550CPU,2.26GHz، 4GBRAM انجام شده است. در ادامه نتایج محاسباتی مربوط به مسائل در ابعاد کوچک و سپس برای مسائل در ابعاد بزرگ مورد تحلیل و بررسی قرار می‌گیرد.

جدول ۴ نتایج محاسباتی مرتبط با ابعاد کوچک یعنی مسائل با تعداد ۴، ۶ و ۸ کار است. همانطور که از جدول می‌توان استنباط نمود الگوریتم ژنتیک قادر به پیدا کردن ۵ جواب به صورت بهینه بوده است. این الگوریتم همچنین جواب یک مساله را با توجه به آن که مدل ریاضی قادر به پیدا کردن جواب بهینه برخی مسایل در محدودیت ۳۶۰۰ ثانیه نبوده بهتر از مدل ریاضی پیدا نموده است. این در حالی است که الگوریتم شبیه سازی تبرید قادر به پیدا کردن ۵ مساله به صورت بهینه بوده و جواب یک مساله را بهتر از مدل ریاضی پیدا کرده است. همچنین از این جدول می‌توان نتیجه گرفت که در ۴ مساله از ۱۸ مساله، الگوریتم‌های SA و GA جواب یکسانی را پیدا کرده‌اند و از بین مسائلی که دو الگوریتم جواب یکسانی بدست نیاورده اند در ۸ مساله جواب SA بهتر از GA و در مابقی مسائل، یعنی ۵ مساله GA بهتر از SA عمل کرده است.

خلاصه‌ی نتایج بدست‌آمده برای مسائل در ابعاد کوچک در جدول ۵ گزارش شده است. در این جدول در ستون‌های اول الی آخر به ترتیب میانگین درصد خطای هر یک از روش‌های حل نسبت به جواب‌های بدست آمده از GAMS، تعداد جواب‌های بهینه بدست آمده از هر یک از الگوریتم‌ها و میانگین زمان‌های حل توسط هر یک از روش‌های حل و GAMS بر حسب ثانیه گزارش شده است. از لحاظ زمان محاسباتی نیز می‌توان به این نتیجه رسید که متوسط زمان حل مسایل توسط مدل ریاضی برابر با ۷۴۴.۱۹۳ ثانیه بوده، این در حالی است که متوسط زمان حل مسائل برای الگوریتم‌های SA و GA به ترتیب برابر با ۰.۲۲۱ و ۰.۲۰۳ ثانیه می‌باشد.

جدول‌های شماره ۶ و ۷ به ترتیب نشانگر نتایج محاسباتی ابعاد بزرگ و خلاصه نتایج حاصل از مسائل با ابعاد بزرگ می‌باشند. از لحاظ متوسط درصد اختلاف جواب‌های بدست آمده از الگوریتم‌ها با جواب‌های بدست آمده از مدل ریاضی می‌توان نتیجه گرفت که الگوریتم‌های SA و GA به طور متوسط به ترتیب دارای ۰.۰۰۴۱ و ۰.۰۱۴۴ درصد خطا نسبت به جواب بدست آمده از مدل ریاضی می‌باشند. همچنین در حالت‌هایی که جواب بدست آمده از الگوریتم‌ها بهتر از GAMS بوده است، جواب‌های SA و GA به ترتیب ۰.۰۱۵۹ و ۰.۰۱۵۶ درصد بهتر از جواب بدست آمده از مدل ریاضی بوده است. همان‌طور که در جدول ۶ قابل مشاهده می‌باشد در تمامی موارد ابعاد بزرگ، همواره جواب‌های بدست آمده از الگوریتم SA کیفیت بهتری نسبت به GA داشته‌اند. جواب‌های بدست آمده از الگوریتم SA در مسایل ابعاد بزرگ از کیفیت بهتری برخوردارند ولی از لحاظ زمان محاسباتی زمان بیشتری را برای بدست آوردن جواب نیاز دارند. میانگین زمان محاسباتی الگوریتم SA برابر ۳.۱۸۳۴ می باشد، این در حالیست که میانگین زمان محاسباتی برای GA برابر با ۰.۲۳۱۵ است. این مقادیر نشان دهنده کارایی بالای الگوریتم SA نسبت به GA هم از لحاظ کیفیت و هم از لحاظ زمان محاسباتی می‌باشد.

روشی که برای اعتبارسنجی الگوریتم پیشنهادی در این مقاله استفاده شده این است که در مساله پیشنهادی این مقاله به مقایسه جوابهای بدست آمده از هر دو الگوریتم پرداختیم که بدین منظور با توجه به غیرنرمال بودن توزیع جوابهای بدست آمده حاصل از الگوریتم های پیشنهادی، از آزمون ناپارامتری کروسکال والیس برای مشخص کردن برابری میانگین جوابهای حاصل از دو الگوریتم پیشنهادی در سطح اطمینان ۹۵٪ با استفاده از نرم افزار مینی تب ۱۶ استفاده نمودیم که جوابهای بدست آمده حاصل از انجام این آزمون (مقدارP-Value حاصل از آزمون بزرگتر از ۰.۰۵ شده است) نشان دهنده برقراری فرض صفر آزمون (برابری میانگین جوابهای بدست آمده حاصل از الگوریتم های پیشنهادی) در مقابل فرض یک آزمون (عدم برابری میانگین جوابهای بدست آمده حاصل از الگوریتم های پیشنهادی) می باشد. جوابهای حاصل از انجام آزمون کروسکال والیس را می توان در شکل (۲) مشاهده نمود:

Kruskal-Wallis Test on باسخ

فاکتور	N	Median	Ave Rank	Z
1	36	2774	37.3	0.34
2	36	2757	35.7	-0.34
Overall	72		36.5	

H = 0.12 DF = 1 P = 0.731
H = 0.12 DF = 1 P = 0.731 (adjusted for ties)

شکل (۲): نتایج حاصل از آزمون کروسکال والیس

۷- نتیجه گیری و پیشنهاد برای پژوهش های آتی

در این مقاله الگوریتمهای فراابتکاری ژنتیک و شبیه سازی تبرید برای حل مساله جریان کارگاهی برگشت پذیر بدون وقفه پیشنهاد شده است. مدل ریاضی توسعه داده شده برای مساله توسط نرم افزار GAMS مدل شده و توسط حل کننده CPLEX حل شده است. برای کد نویسی روش های حل ارائه شده از زبان برنامه نویسی C++ استفاده شده است. همچنین برای مقایسه دو الگوریتم از نظر برابری میانگین جوابهای بدست آمده از هر روش از آزمون ناپارامتری کروسکال والیس استفاده شده است. جواب های بدست آمده نشان دهنده کیفیت بهتر جوابهای الگوریتم شبیه سازی تبرید پیشنهادی در حل مساله مذکور مخصوصا در حل مسائل با ابعاد بزرگ می باشد.

جدول (۴): جوابهای بدست آمده برای مسائل با ابعاد کوچک

m×n	جوابهای GAMS	متوسط جوابهای GA	متوسط جوابهای SA	بهترین جواب GA	بهترین جواب SA	زمان GAMS	زمان GA	زمان SA
۳×۴	۷۱۸	۷۱۸	۷۲۰.۵	۷۱۸	۷۱۸	۰.۹۲۸	۰.۱۸۲	۰.۱۹
۵×۴	۷۷۳	۷۷۳	۷۷۳	۷۷۳	۷۷۳	۱.۱۵۴	۰.۱۸۸	۰.۲۲۵
۷×۴	۸۲۶	۸۲۶	۸۲۶	۸۲۶	۸۲۶	۱.۳۵۴	۰.۱۹۳	۰.۱۲۹
۱۰×۴	۸۵۷	۸۵۷	۸۵۷	۸۵۷	۸۵۷	۲.۸۸۲	۰.۱۹	۰.۲۵۸
۱۵×۴	۱۱۸۰	۱۱۸۰	۱۱۸۳.۵	۱۱۸۰	۱۱۸۰	۶.۹۶۷	۰.۱۸	۰.۱۵۶
۲۰×۴	۱۴۴۸	۱۴۴۸	۱۴۴۸	۱۴۴۸	۱۴۴۸	۱۶.۷۸	۰.۲۰۴	۰.۴۶
۳×۶	۹۷۳	۹۷۷	۹۸۶.۲۵	۹۷۳	۹۸۶.۵	۷.۹۵۳	۰.۱۸۷	۰.۱۷۸
۵×۶	۱۱۶۶	۱۱۶۸.۵	۱۱۶۶	۱۱۶۶	۱۱۶۶	۱۲.۱۹۸	۰.۱۹۴	۰.۱۵
۷×۶	۱۳۳۶	۱۳۳۶.۲۵	۱۳۳۸.۷۵	۱۳۳۶	۱۳۳۶	۱۶.۵۶	۰.۱۹۷	۰.۱۴۹
۱۰×۶	۱۲۲۰	۱۲۳۰.۵	۱۲۲۱.۷۵	۱۲۲۰	۱۲۲۰	۶۹.۰۱۵	۰.۱۸۱	۰.۲۱۸
۱۵×۶	۱۷۶۳	۱۷۷۶.۷۵	۱۷۸۲	۱۷۶۳	۱۷۶۳	۱۱۹.۱۹۲	۰.۲۱۳	۰.۲۵۷
۲۰×۶	۱۷۹۸	۱۸۱۲	۱۸۱۲	۱۷۹۸	۱۷۹۸	۲۸۶.۴۸۲	۰.۲۰۱	۰.۲۸۴
۳×۸	۱۳۲۰	۱۳۲۴	۱۳۲۰.۷۵	۱۳۲۰	۱۳۲۰	۶۴۲.۹۸۷	۰.۲۲۶	۰.۱۶۲
۵×۸	۱۴۶۴	۱۴۸۰.۲۵	۱۴۶۴.۵	۱۴۷۹	۱۴۶۴	۱۳۴۰.۳۷	۰.۲۵۸	۰.۱۵۳
۷×۸	۱۲۷۳	۱۳۷۸	۱۲۸۱.۵	۱۳۶۵	۱۲۷۳	۹۵۳.۴۴۱	۰.۱۹۹	۰.۲۵۵
۱۰×۸	۱۳۱۴	۱۴۴۲	۱۳۲۱	۱۴۳۸	۱۳۱۴	۲۷۱۷.۲۱۵	۰.۲۱۸	۰.۱۹۹
۱۵×۸	۱۷۵۴	۱۷۸۷.۲۵	۱۷۵۹	۱۷۷۰	۱۷۵۴	۳۶۰۰	۰.۲۲۹	۰.۲۷۶
۲۰×۸	۲۳۲۲	۲۲۸۵.۷۵	۲۲۸۵	۲۲۸۲	۲۲۷۶	۳۶۰۰	۰.۲۲	۰.۲۸۱

جدول (۵): خلاصه نتایج بدست آمده برای مسائل با ابعاد کوچک

میانگین درصد خطا نسبت به GAMS		تعداد جواب‌های بهینه			میانگین زمان حل		
SA	GA	SA	GA	GAMS	SA	GA	
۰.۰۰۳۴	۰.۰۱۴۳	۵	۶	۷۴۴.۱۹۳	۰.۲۲۱	۰.۲۰۳	

جدول (۶): جوابهای بدست آمده برای مسائل با ابعاد بزرگ

m×n	متوسط جوابهای GA	متوسط جوابهای SA	بهترین جواب GA	بهترین جواب SA	زمان GA	زمان SA
۳×۲۰	۳۲۶۲.۷۵	۳۲۲۸	۳۲۱۶	۳۲۲۵	۰.۱۹۳	۰.۱۵۷
۵×۲۰	۳۴۷۷.۵	۳۳۸۹	۳۴۳۴	۳۳۸۸	۰.۱۹۶	۰.۳۰۲
۷×۲۰	۳۴۵۱.۵	۳۲۳۹	۳۳۴۷	۳۲۳۲	۰.۲۴۶	۰.۲۶۹
۱۰×۲۰	۳۵۵۰.۵	۳۴۷۷	۳۵۳۱	۳۴۶۵	۰.۱۹۲	۰.۴۰۶
۱۵×۲۰	۴۵۵۲	۴۳۱۷	۴۵۳۲	۴۲۹۵	۰.۲۰۴	۱.۸۵۴
۲۰×۲۰	۴۵۶۰	۴۱۵۱.۵	۴۵۳۷	۴۱۱۷	۰.۲۰۶	۳.۰۵
۳×۳۰	۴۶۸۶.۲۵	۴۵۸۷.۵	۴۵۵۸	۴۵۷۸	۰.۱۷۴	۰.۳۱۲
۵×۳۰	۴۸۰۲.۵	۴۵۶۴.۲۵	۴۸۰۱	۴۵۳۲	۰.۲۲۱	۰.۵۸۶
۷×۳۰	۵۲۵۰.۷۵	۵۰۷۱	۵۲۵۰	۵۰۴۶	۰.۲۰۶	۲.۴۳
۱۰×۳۰	۵۵۸۹.۵	۵۲۷۷	۵۵۸۹	۵۲۷۲	۰.۲۳۱	۲.۶۰۶
۱۵×۳۰	۶۱۴۹	۵۵۷۱.۷۵	۶۱۰۶	۵۵۳۷	۰.۲۲۸	۵.۴۶۹
۲۰×۳۰	۶۳۸۱.۷۵	۵۸۷۰	۶۳۴۹	۵۷۴۳	۰.۲۲۲	۷.۹۵
۳×۴۰	۶۴۴۸.۵	۶۳۵۳.۵	۶۳۱۲	۶۳۰۶	۰.۲۰۱	۰.۷۶۷
۵×۴۰	۶۶۲۷	۶۳۰۶.۲۵	۶۶۰۹	۶۲۸۵	۰.۲۵۴	۰.۷۸۹
۷×۴۰	۶۷۳۱.۲۵	۶۳۸۷	۶۶۵۹	۶۳۵۰	۰.۲۱۷	۲.۲۷
۱۰×۴۰	۶۹۵۲.۵	۶۵۷۶.۲۵	۶۹۴۲	۶۵۶۴	۰.۲۱۸	۴.۱۸۵
۱۵×۴۰	۷۵۱۶	۶۸۰۸	۷۵۱۴	۶۷۶۲	۰.۲۰۳	۱۰.۹۵
۲۰×۴۰	۸۳۲۴	۷۷۵۵	۸۲۳۷	۷۶۸۷	۰.۲۳۱	۱۲.۹۵

جدول (۷): خلاصه نتایج بدست آمده برای مسائل با ابعاد کوچک

درصد خطای GA نسبت به SA	میانگین زمان حل GA	میانگین زمان حل SA
۰.۰۵۴۹	۰.۲۳۱۵	۳.۱۸۳۴

حل مساله مذکور با استفاده از الگوریتم های فراابتکاری مختلف و مقایسه جوابهای بدست آمده از آنها با جوابهای این مقاله و هم چنین حل مساله مذکور به صورت جریان کارگاهی انعطاف پذیر به عنوان پژوهش های آتی پیشنهاد می‌گردد. همچنین در نظر گرفتن مطالعات موردی جهت کاربردی‌تر کردن مساله ذکر شده یکی دیگر از مسیرهای تحقیقاتی ممکن می تواند باشد. ارائه روش‌های ابتکاری جهت بدست آوردن حد پایین یا حل مساله نیز پیشنهاد می‌گردد.

مراجع

[1] Hall, N.G., Sriskandarajah, C., (1996). "A survey of machine scheduling problems with blocking and no-wait in process", Operations Research, 44: 510-525.

[2] Aldowaisan, T., Allahverdi, Ali., (2003). "New heuristics for no-wait flowshops to minimize makespan", Computers & Operations Research, 30: 1219–1231.

[3] Gupta, J.N.D., Strusevich, V.A., Zwaneveld, CM., (1997). "Two-stage no-wait scheduling models with setup and removal times separated", Compute Operation Research, 24: 1025–1031.

[4] Choi, S.W., Kim, Y.D. (2008). "Minimizing makespan on an -machine re-entrant flowshop", Computers & Operations Research, 35: 1684-1696.

[5] Chen, J.S., Pan, J.C.H., Lin, C.M., (2008). "A hybrid genetic algorithm for the re-entrant flow-shop scheduling problem", Expert Systems with Applications, 34: 570-577.

[6] Jing, C., Tang, G., Qian, X., (2008). "Heuristic algorithms for two machine re-entrant flow shop", Theoretical Computer Science, 400: 137-143.

[7] Huang, R.H., Yu, S.C., Kuo, C.W., (2014). "Reentrant two-stage multiprocessor flow shop scheduling with due windows", The International Journal of Advanced Manufacturing Technology, 71: 1263-1276.

[8] Li, Z.C., Qian, B., Hu, R., Zhang, C.S., Li, K., (2013). "A self-adaptive hybrid population-based incremental learning algorithm for M-machine reentrant permutation flow-shop scheduling", Intelligent Computing Theories, 7995: 8-20.

[9] Jing, C., Huang, W., Tang, G., (2011). "Minimizing total completion time for re-entrant flow shop scheduling problems", Theoretical Computer Science, 412: 6712-6719.

[10] Choi, S.W., Kim, Y.D., (2009). "Minimizing total tardiness on a two-machine re-entrant flow shop", European Journal of Operational Research, 199: 375-384.

[11] Pan, J.C.H., Chen, J.S., (2003). "Minimizing makespan in reentrant permutation flow-shops", Journal of Operational Research Society, 57: 642-653.

[12] Adressi, A., Hassanpour, S., Azizi, V., (2016). "Solving group scheduling problem in no-wait flexible flowshop with random machine breakdown", Decision Science Letters, 5(1): 157-168.

[13] Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, A.R., (1979). "Optimization and approximation in deterministic sequencing and scheduling: a survey", Annals of discrete mathematics, 5: 287-326.

IERPS Journal of Industrial Engineering Research in Production Systems

ISSN: 2345-2269

Journal of Industrial Engineering Research in Production Systems

Volume 4. No.9. Special Issue: 11th & 12th International Conference on Industrial Engineering, pp.271-279
www.ier.basu.ac.ir

Solving Re-entrant No-wait Flow Shop Scheduling Problem

S. Tasouji Hasanpour[1], M.R. Amin-Naseri[1,*], A. Adressi[2]

[1] Department of Industrial Engineering, Tarbiat Modares University, Tehran, Iran.
[2] Department of Industrial Engineering, Khajeh Nasir Toosi University of Technology, Tehran, Iran.

ARTICLE INFO

***Article history*:**

Received 27 January 2016
Accepted 1 March 2017

***Keywords*:**

Re-entrant Flow shop
No-wait Flow shop
Genetic Algorithm
Simulated Annealing

ABSTRACT

In this study we consider the production environment of re-entrant flow-shop (RFS) with the objective of minimizing make span of the jobs. In a RFS, at least one job should visit at least one of the machines more than once. In a no-wait flow shop-scheduling problem, when the process of a specific job begins on the first machine, it should constantly be processed without waiting in the line of any machine until its processing is completed on the last one. Integration of the properties of both of these environments, which is applied in many industries such as robotic industries, is not investigated separately. In the paper, we present a simulated annealing (SA) and a genetic algorithm (GA) based on heuristics for the problem. First, we develop the mathematical model for the problem, and then we present the suggested algorithms. For small scale, results of GA and SA are compared to GAMS. For large-scale problems, results of GA and SA are compared to each other. Computational results show that both SA ad GA algorithms perform properly but totally, SA is likely to turn out well in finding better solutions especially in large-scale problems.

* Corresponding author. Mohammad-Reza Amin-Naseri
Tel.: 021-82883344; *E-mail address: amin_nas@modares.ac.ir*