

ارائه مدلی برای زمانبندی خط تولید کارگاهی همراه با یک مرحله مونتاژ موازی با هدف کمینه‌سازی حداکثر دوره ساخت

فاطمه دانش آموز^۱، مونا جباری^۲، پرویز فتاحی^{۳*}

۱. دانشجوی دکتری مهندسی صنایع، دانشگاه بوعلی‌سینا، همدان

۲. دانشجوی کارشناسی‌ارشد مهندسی صنایع، دانشگاه بوعلی‌سینا، همدان

۳. دانشیار گروه مهندسی صنایع، دانشگاه بوعلی‌سینا، همدان

خلاصه

در این تحقیق مسئله زمانبندی تولید کارگاهی با در نظر گرفتن یک مرحله مونتاژ موازی با هدف حداقل کردن زمان تکمیل محصولات در نظر گرفته شده است. ابتدا به منظور تشریح مسئله، یک مدل برنامه‌ریزی خطی ارائه شده است. برای بررسی صحت عملکرد مدل و همچنین بررسی میزان کارایی الگوریتم در حل مسائل با اندازه کوچک‌تر چند نمونه مسئله با مدل ارائه شده توسط LINGO حل شده است. با توجه به این که مسئله مورد بررسی از نوع NP-hard می‌باشد برای حل آن در ابعاد متوسط و بزرگ از الگوریتم بهینه‌سازی انبوه ذرات (PSO) در سه حالت استفاده شده است. همچنین به منظور بررسی کارایی الگوریتم پیشنهادی مثال‌های عددی مورد استفاده قرار گرفته است و سپس نتایج الگوریتم‌های مورد نظر با الگوریتم ژنتیک (GA) مقایسه گردیده است. نتایج نشان می‌دهد که الگوریتم پیشنهادی توانایی دستیابی به نقاط بهینه و نزدیک به بهینه برای مسائل در اندازه‌های مختلف را دارا می‌باشد.

اطلاعات مقاله

تاریخچه مقاله:

دریافت ۱۳۹۲/۱۰/۱۵

پذیرش ۱۳۹۳/۵/۲۱

کلمات کلیدی:

زمانبندی

تولید کارگاهی

مونتاژ موازی

روش بهینه‌سازی انبوه ذرات

۱- مقدمه^۱

زمانبندی تولید کارگاهی (JSP) یکی از پیچیده‌ترین مباحث بهینه‌سازی ترکیبیاتی و شاخه‌ای از زمانبندی تولید است که به صورت زمانبندی n کار روی m ماشین می‌باشد. در این مسئله هر کار دارای تعدادی عملیات است که باید به ترتیب انجام گردند و هدف از زمانبندی این مسئله، تعیین توالی عملیات‌ها برای هر ماشین به نحوی است که یک تابع هدف از قبل مشخص شده مانند دوره ساخت بهینه گردد.

از طرفی در بسیاری از کارگاه‌ها و سیستم‌های تولیدی بعد از این که قطعات و مجموعه‌ها آماده شدند باید با یکدیگر مونتاژ گردند تا تبدیل به یک محصول کامل شوند. مسائل زمانبندی که در آنها

مراحل ساخت قطعات و مونتاژ محصولات همزمان با هم در نظر گرفته می‌شود، کاربرد زیادی در صنعت دارد. چرا که بهینه کردن توالی انجام کارها در یک مرحله از تولید بدون در نظر گرفتن پارامترها و ظرفیت‌های مرحله بعدی، تضمینی برای کوتاه شدن زمان کل تولید و استفاده بهینه از ماشین‌آلات و سایر ظرفیت‌های موجود نخواهد بود.

طبق نتایج مطالعه آماری انجام شده توسط ریزمن و همکاران [۲] از میان ۱۸۴ مقاله بررسی شده، تنها ۵ مقاله به مسائل دنیای واقعی پرداخته‌اند. با توجه به مطرح شدن این مشکل در سطح گسترده در سال‌های اخیر، هدف اصلی بسیاری از محققان طرح مسائل عملی و کاربردی و تلاش برای کاهش شکاف عمیق بین روش‌های تئوری و کاربردی عملی این علم است. لذا پرداختن به مسائل زمانبندی که در صنعت کاربرد عملی دارند از ضرورت‌ها و نیازهای امروز محسوب می‌شود. این مهم موجب شده است تا طی دو دهه اخیر محققین در

* نویسنده مسئول. پرویز فتاحی

تلفن: ۰۸۱۳۸۲۵۷۴۱۰؛ پست الکترونیکی: Fattahi@basu.ac.ir

نظر گرفتند.

یوکویاما [۱۷] نیز یک سیستم تولید ترکیبی را مورد بررسی قرار داد که در آن قطعات مورد نیاز هر محصول در یک کارگاه جریانی دو ماشینی با زمان مستقل پردازش شده و پس از آماده شدن قطعات هر محصول، این قطعات در مرحله پایانی مونتاژ می‌شدند. که در این مقاله تابع هدف موردنظر کمینه کردن جمع وزنی زمان تکمیل محصولات بود. سان و همکاران [۱۸] سیستم تولید دومرحله‌ای را که مرحله اول آن شامل دو ماشین موازی و مرحله دوم آن مونتاژ بود، بررسی کردند که تابع هدف در نظر گرفته شده در کار آن‌ها حداقل کردن C_{max} بود. همچنین سانگ و کیم [۱۹]، مسئله زمانبندی مونتاژ دومرحله‌ای چند ماشینه را با هدف کمینه کردن جمع زمان‌های تکمیل کارها در نظر گرفتند. در مدل پیشنهادی آن‌ها مرحله اول شامل دو ماشین مستقل که هر کدام نوع خاصی از اجزا را تولید می‌کنند (سیستم ثابت) و مرحله دوم متشکل از دو ماشین یکسان و مستقل که موازی با هم کار می‌کنند، می‌باشد. همچنین آن‌ها کمینه کردن جمع زمان اتمام کارها در مرحله دوم را به عنوان معیار زمانبندی در نظر گرفتند. فتاحی و همکاران [۲۰] مسئله زمانبندی جریان کارگاهی ترکیبی مونتاژ دو مرحله‌ای را مورد مطالعه قرار دادند. آن‌ها یک مسئله زمانبندی برای یک سیستم تولید دو مرحله‌ای شامل عملیات ماشینی و عملیات مونتاژ را مطرح کردند. در این سیستم، یک تعداد از انواع مختلف تولید شده‌اند و هر محصول به‌وسیله یک مجموعه از چندین قطعه مونتاژ شده است. در مسئله مورد نظر مرحله اول، یک جریان کارگاهی ترکیبی و مرحله دوم یک ماشین مونتاژ و هدف در نظر گرفته شده، کمینه‌سازی زمان تکمیل تمام محصولات می‌باشد. همچنین آن‌ها چهار الگوریتم بر پایه الگوریتم جانسون به‌منظور زمانبندی محصولات پیشنهاد کردند.

واضح است که مسئله مورد بررسی در این تحقیق با افزودن مرحله مونتاژ موازی و پیچیده‌تر شدن ساختار مسئله نیز قطعا NP-hard خواهد بود. با نتیجه‌گیری NP-hard بودن مسئله مورد نظر، حل آن توسط رویکردهای دقیق و در ابعاد واقعی عملا امکان‌پذیر نبوده و در نتیجه استفاده از رویکردهای نوین و فراابتکاری و البته غیر دقیق اجتناب‌ناپذیر می‌باشد. برخلاف مسائل بهینه‌سازی قدیمی، هدف اصلی از این محاسبات، حل مؤثر مسائل پیچیده در دنیای واقعی است. در حقیقت در الگوریتم‌های فراابتکاری، تضمینی برای یافتن راه‌حل بهینه سراسری وجود ندارد و هدف یافتن راه‌حل‌های با کیفیت بالا در زمان معقول می‌باشد.

کائو و زاهارا [۲۱] ادعا کردند که در میان الگوریتم‌های تکاملی موجود، بهترین شاخه شناخته شده، الگوریتم ژنتیک است. GA، یک روش جستجوی احتمالی براساس انتخاب طبیعی، ژنتیک و تکامل می‌باشد [۲۲]. همچنین کندی و ابرهات [۲۳] یک روش تکاملی جدید ارائه کردند که از مزایای تعامل بین ذرات، بهره می‌برد. رفتار حیوانات مانند پرندگان به‌وسیله این الگوریتم، که الگوریتم بهینه‌سازی انبوه ذرات نامیده می‌شود، شبیه‌سازی شده است. اگرچه

ارائه و توسعه مدل‌های زمانبندی، مراحل تولید که عموماً شامل دومرحله پردازش و مونتاژ می‌باشد را، با هم در نظر بگیرند. از طرفی در بسیاری از سیستم‌های تولید با توجه به وجود امکانات ممکن است مرحله مونتاژ منحصر به فرد نباشد و ایستگاه‌های موازی برای آن در نظر گرفته شود.

مسئله زمانبندی تولید کارگاهی^۱ بیش از ۴۰ سال پیش توسط مان [۳] و واگنر [۴] مطرح شد. از آنجایی که مسئله تولید کارگاهی دارای پیچیدگی بالایی است به‌عنوان یک مسئله بهینه‌سازی NP-hard در ادبیات زمانبندی تولید شناخته شده است [۵،۶] و بدین ترتیب جهت حل آن از الگوریتم‌های فراابتکاری استفاده می‌شود. براساس یک مرور سریع می‌توان برخی از ادبیات مطرح شده در زمینه زمانبندی سیستم‌های تولید کارگاهی را بصورت زیر طبقه‌بندی کرد: روش شاخه و کران^۲ [۷]، روش‌های هیورستیک برپایه قاعده اولویت [۸]، روش‌های متاهيورستیک: الگوریتم انجماد تدریجی^۳ [۹]، الگوریتم جست‌وجوی ممنوع^۴ [۱۰]، الگوریتم ژنتیک^۵ [۱۱] و الگوریتم بهینه‌سازی انبوه ذرات^۶ [۱۲].

همچنین تحقیقات محدودی در زمینه دستیابی به جواب بهینه برای مسئله تولید کارگاهی همراه با یک مرحله مونتاژ انجام شده است. اولین تحقیق که به یک مسئله زمانبندی تولید دومرحله‌ای پرداخته است به کمتر از ۲۰ سال قبل باز می‌گردد که این تحقیق توسط لی و همکاران در سال ۱۹۹۳ انجام گردید [۱۳]. پاتس و همکاران [۱۴] نیز دو سال بعد این مسئله را بررسی کرده و آن را به حالت عمومی‌تری تعمیم دادند که در مسئله مورد بررسی ایشان m ماشین پردازش در مرحله ساخت و یک ماشین جهت مونتاژ در نظر گرفته شده بود. حریری و پاتس [۱۵] نیز مسئله قبل را در نظر گرفتند و یک حد پایین برای آن توسعه دادند و چندین رابطه چیرگی را برقرار ساختند. همچنین، آن‌ها یک الگوریتم شاخه و کران در ترکیب با حد پایین و روابط چیرگی ارائه دادند. چنگ و وانگ [۱۶] به بررسی مسئله زمانبندی قطعات تولیدی و مونتاژی در جریان کارگاهی با دو ماشین پرداختند. در مدل مورد بررسی آن‌ها هر کار شامل یک جزء اختصاصی مخصوص آن کار و یک جزء عمومی مختص تمام کارها می‌باشد. هر دو جزء اختصاصی و عمومی در مرحله اول تولید می‌شوند. در حالی که اجزا اختصاصی منحصراً پردازش می‌شوند، اجزاء عمومی در بسته‌هایی تولید شده و زمان آماده‌سازی برای تشکیل یک بسته تولیدی اجزاء عمومی نیاز است. عملیات مونتاژ هر کار در ماشین مرحله دوم صورت می‌گیرد و زمانی می‌تواند آغاز شود که هر دو جزء یک کار آماده و در دسترس باشند. همچنین آن‌ها کمینه‌سازی زمان اتمام کارها را به عنوان هدف در

1. Job shop scheduling
2. Branch and Bound (BB)
3. Simulated Annealing (SA)
4. Tabu Search (TS)
5. Genetic Algorithm (GA)
6. Particle swarm optimization (PSO)

تحلیل ارائه می‌گردد. نتیجه‌گیری و پیشنهاد تحقیقات آتی نیز در بخش پنجم ارائه می‌گردد.

۲- تعریف مسئله و مدل ریاضی

به‌طور کلی تولید محصولات در یک سیستم تولید دو مرحله‌ای مورد بررسی در این مقاله را می‌توان به‌صورت تعیین توالی انجام کارها در یک سیستم تولیدی، که مرحله اول آن تولید کارگاهی و مرحله دوم آن یک ایستگاه با خط مونتاژ موازی می‌باشد، تعریف کرد. بدیهی است که مونتاژ هر محصول در مرحله دوم زمانی می‌تواند آغاز گردد که تمام قطعات موردنیاز آن در مرحله اول پردازش شده و آماده باشند. در تولید محصولات در سیستم تولید مورد بررسی، در مرحله اول مواد و قطعات اولیه در یک سیستم تولید کارگاهی پردازش شده و قطعات موردنیاز محصولات آماده می‌شود سپس بعد از تکمیل قطعات مورد نیاز هر محصول، عملیات مونتاژ بر روی ماشین‌های موازی جهت تکمیل محصولات آغاز می‌گردد.

در این قسمت یک مدل ریاضی به‌منظور شناسایی و همچنین استفاده از آن برای حل مسئله ارائه می‌گردد. مرحله اول، مسئله زمانبندی تولید کارگاهی است که به صورت زمانبندی J قطعه ماشین‌ها، که شامل m ماشین M_1, M_2, \dots, M_m است، می‌باشد. هر قطعه دارای h_j عملیات می‌باشد که بایستی به‌ترتیب انجام گردند. اندیس p نشان‌دهنده محصول، اندیس j نشان‌دهنده قطعه، اندیس h نشان‌دهنده عملیات و اندیس i نشان‌دهنده ماشین‌ها در مرحله اول می‌باشد. همچنین عملیات h از قطعه j از محصول p با نماد $O_{j,p,h}$ ($h=1, \dots, h_j; j=1, \dots, n; p=1, \dots, P$) نشان داده می‌شود که از قبل تعیین شده است و با زمان پردازش $PS_{j,p,h}$ انجام می‌گردد.

مرحله دوم، مسئله زمانبندی مونتاژ موازی است که به‌صورت زمانبندی p محصول با زمان مونتاژ A_p روی مجموعه M' از ماشین‌ها، که شامل m' ماشین $M'_1, M'_2, \dots, M'_{m'}$ که به‌صورت موازی قرار گرفته‌اند، می‌باشد.

فرضیاتی که در مراحل مدلسازی در نظر گرفته شده است، عبارت است از:

- ✓ تمامی ماشین‌آلات در شروع افق برنامه‌ریزی در دسترس می‌باشند. $\forall i, r_i = 0$
- ✓ تقاضای محصولات نهایی مشخص و تمامی قطعات در زمان صفر آماده پردازش هستند. $\forall j, r_j = 0$
- ✓ قطع کار مجاز نمی‌باشد. بدین معنی که هر عملیات شروع شده، باید بدون وقفه به اتمام برسد.
- ✓ تعدادی محصول می‌بایست تولید شود و هر محصول برای تکمیل شدن نیازمند یک سری قطعات می‌باشد.
- ✓ قطعات هر محصول خاص همان محصول می‌باشد.

PSO و GA وجه اشتراک‌های زیادی دارند ولی برخلاف PSO، حافظه‌دار است. به‌دلیل همین توانایی، جواب‌های خوب می‌توانند در PSO حفظ و به وسیله ذرات به کار برده شوند و درست شبیه به رفتار دسته پرندگان، ذرات در PSO قادرند که اطلاعات را در میان خود به اشتراک بگذارند [۲۱].

PSO به دلیل قابلیت جستجوی سراسری در فضای پیوسته بسیار مورد توجه می‌باشد و به همین دلیل بسیاری از محققان از این الگوریتم برای جستجو در فضای حل پیوسته استفاده می‌کنند اما این الگوریتم تنها در شمار کمی از مقالات برای حل در فضای گسسته به کار برده شده است [۲۴]. در نتیجه در سال‌های اخیر تلاش‌های بسیاری برای تغییر و بهبود PSO، مانند [۲۵-۳۰]، و روش‌های تکاملی دیگر [۳۱، ۳۲]، برای حل مسائل بهینه‌سازی پیوسته شده است. اما با این وجود در ادبیات، PSO برای بهینه‌سازی مسائل زمانبندی تولید کارگاهی چندان پرکاربرد نمی‌باشد.

همان‌طور که ذکر گردید در اغلب تحقیقات انجام شده در حوزه زمانبندی صرفاً به یک مرحله تولید (مرحله اول) به‌صورت جداگانه پرداخته شده و با توجه به معیاری مشخص، بهترین زمانبندی برای انجام کارها در آن مرحله خاص ارائه شده است. حال آن‌که بیش‌تر صنایع تولیدی به صورت دو مرحله‌ای کار می‌کنند که بر اساس آن، قطعات طی مرحله اول پردازش و تولید می‌شوند و در مرحله دوم با مونتاژ قطعات، محصول نهایی شکل می‌گیرد. چنان‌چه این مراحل با هم در نظر گرفته نشوند و زمانبندی کارها فقط در یک مرحله بهینه گردد، با توجه به وابستگی مراحل تولید چه بسا نتایج مطلوب حاصل نگردد. همچنین نظر به این‌که در شرکت‌های بزرگ تولیدی مانند ایران خودرو و ... خطوط مونتاژ به صورت موازی می‌باشد پژوهش در زمینه ارائه و بهینه‌سازی مدلی جامع و نیز ارائه راهکارهایی جهت تسهیل بهینه‌سازی این خطوط، امری مفید و ضروری به نظر می‌رسد. بنابراین چنانچه با توجه به وجود امکانات در سیستم‌های تولیدی و به‌منظور کاربردی کردن مسئله، مرحله دوم (مونتاژ قطعات) به صورت موازی انجام گردد، این مسئله یک مسئله تولید کارگاهی با عملیات مونتاژ موازی خواهد بود. این مسئله تعمیم یافته تولید کارگاهی با عملیات مونتاژ می‌باشد که در آن هر محصول می‌تواند توسط یکی از ماشین‌ها مونتاژ شود. بررسی تحقیقات گذشته نشان داد که مسئله زمانبندی تولید کارگاهی همراه با یک مرحله مونتاژ موازی تاکنون مورد مطالعه قرار نگرفته است که در این مقاله، مسئله موردنظر با هدف مینیمم کردن حداکثر دوره ساخت مورد بررسی قرار می‌گیرد. در این مقاله ابتدا مدل ریاضی به‌منظور تشریح مسئله و استفاده از آن برای ارزیابی کارایی الگوریتم‌های پیشنهادی ارائه می‌گردد. سپس الگوریتم‌های پیشنهادی با استفاده از آزمایشات عددی مورد تحلیل قرار می‌گیرد.

سازماندهی سایر بخش‌های مقاله به‌ترتیب زیر است: در بخش دوم مسئله مورد بررسی تشریح می‌شود. در بخش سوم الگوریتم‌های پیشنهادی ارائه می‌گردد. در بخش چهارم آزمایشات عددی و تجزیه و

$O_{j,p,h}$ زمان شروع عملیات	$t_{j,p,h}$	✓ زمان پردازش قطعات و مونتاژ محصولات قطعی و معین می‌باشد.
S_p زمان شروع عملیات مونتاژ		✓ زمان آماده‌سازی معلوم می‌باشد و با زمان پردازش در نظر گرفته شده است.
$Tm_{i,k}$ زمان شروع به کار ماشین i در نوبت k در مرحله ۱		✓ زمانی که تولید همه اجزاء یک محصول در مرحله اول به پایان رسید، وارد مرحله مونتاژ می‌شوند. حداقل زمان شروع عملیات مونتاژ در مرحله مونتاژ برابر با حداکثر زمان اتمام تمام اجزاء آن محصول بر روی ماشین‌های مرحله اول می‌باشد.
$Sm_{i',k'}$ زمان شروع به کار ماشین i' در نوبت k' در مرحله ۲		✓ ماشین‌ها نمی‌توانند در یک زمان بیش از یک عملیات را انجام دهند و قطع کار مجاز نمی‌باشد (ماشین‌ها در هر زمان یا بر روی یک قطعه/محصول عملیات انجام می‌دهند و یا بیکار هستند).
$x_{i,j,p,h,k}$ روی ماشین i در نوبت k انجام گردد برابر ۱ و در غیر اینصورت ۰ می‌باشد.		✓ در هر زمان هر قطعه فقط بر روی یک ماشین پردازش می‌شود (هر کار در هر زمان یا در حال پردازش بر روی یک ماشین است یا در صف انتظار ماشینی برای پردازش می‌باشد).
$Z_{i',p,k'}$ روی ماشین i' در نوبت k' برابر ۱ و در غیر این صورت ۰ می‌باشد.		پارامترهای این مدل عبارتند از:
با داشتن پارامترهای بیان شده، مسئله به صورت زیر مدل‌سازی می‌گردد:		
$Min Z = (C_{max})$	(۱)	p تعداد محصولات ($p=1,2,3,\dots,P$)
<i>Subject to:</i>		n تعداد قطعات ($j=1,2,3,\dots,n_p$)
$c_{max} \geq C_p \quad \forall p$	(۲)	m تعداد ماشین‌ها در مرحله تولید کارگاهی ($i=1,2,3,\dots,m$)
$t_{j,p,h} + ps_{j,p,h} \leq t_{j,p,h+1}$	(۳)	m' تعداد ماشین‌های موازی در مرحله مونتاژ ($i'=1,2,3,\dots,m'$)
$\forall p, j; h = 1, 2, \dots, h_j - 1$		n_p تعداد قطعات زیر محصول
$Tm_{i,k} + ps_{j,p,h}x_{i,j,p,h,k} \leq Tm_{i,k+1}$	(۴)	k_i تعداد عملیات‌های اختصاص یافته به ماشین ($k=1,2,3,\dots,k_i$)
$\forall p, j, h, i; k = 1, 2, \dots, k_i - 1$		h_j تعداد عملیات‌های قطعه j ($h=1,2,3,\dots,h_j$)
$Tm_{i,k} \leq t_{j,p,h} + (1 - x_{i,j,p,h,k}) \cdot L$	(۵)	$O_{j,p,h}$ عملیات h از قطعه j از محصول p
$\forall p, j, h, k, i$		$ps_{j,p,h}$ زمان پردازش عملیات $O_{j,p,h}$
$Tm_{i,k} + (1 - x_{i,j,p,h,k}) \cdot L \geq t_{j,p,h} \quad \forall p, j, h, k, i$	(۶)	A_p زمان مونتاژ محصول p
$\sum_p \sum_j \sum_h x_{i,j,p,h,k} = 1 \quad \forall k, i$	(۷)	L یک عدد بزرگ
$\sum_k x_{i,j,p,h,k} = a_{i,j,p,h} \quad \forall p, j, h, i$	(۸)	یک متغیر ۰ و ۱ است به طوری که اگر عملیات غیراین صورت ۰ می‌باشد.
$t_{j,p,h} + ps_{j,p,h} \leq F_p \quad \forall p, j, h$	(۹)	متغیرهای این مدل عبارتند از:
$F_p \leq S_p \quad \forall p$	(۱۰)	C_{max} زمان حداکثر دوره ساخت
$A_p + S_p \leq C_p \quad \forall p$	(۱۱)	F_p زمان ساخت محصول P در مرحله اول
$Sm_{i',k'} + A_p \cdot Z_{i',p,k'} \leq Sm_{i',k'+1}$	(۱۲)	C_p زمان تکمیل نهایی محصول P
$\forall p, i'; k'=1,2,3,\dots,k'_{i'} - 1$		
$Sm_{i',k'} \leq S_p + (1 - Z_{i',p,k'}) \cdot L \quad \forall p, k', i'$	(۱۳)	

منظور صرفه‌جویی در زمان و برای حل مسائل سایز بزرگ از الگوریتم فراابتکاری بهینه‌سازی انبوه ذرات برای حل آن استفاده شده است.

الگوریتم بهینه‌سازی انبوه ذرات یک الگوریتم بهینه‌سازی فراابتکاری است که به‌طور عمده از فرآیند تکاملی در طبیعت الهام گرفته شده است. PSO یک تکنیک بهینه‌سازی است که توسط راسل ابرهارت و جیمز کندی در سال ۱۹۹۵ ارائه شد و از رفتار اجتماعی پرندگان یا ماهی‌ها در پیدا کردن غذا الهام گرفته شده است [۲۳].

در PSO هر جواب مسئله، موقعیت یک ذره در فضای جستجو است که بصورت تصادفی در فضای مسئله حرکت می‌کند و تغییر مکان هر ذره در فضای جستجو تحت تأثیر بهترین جوابی که تاکنون برای هر ذره به‌طور جداگانه بدست آمده است و $P\text{-Best}$ نامیده می‌شود و بهترین مقداری که تاکنون توسط تمام ذره‌ها در میان جمعیت به‌دست آمده است و $G\text{-Best}$ نامیده می‌شود، می‌باشد. بنابراین چگونگی جستجوی هر ذره با توجه به موقعیت ذرات دیگر است و هر ذره از برتری و عدم برتری خود نسبت به ذرات همسایه و کل گروه آگاه است و با توجه به آن در فضای مسئله حرکت می‌کند. بدین ترتیب می‌توان موقعیت و سرعت ذره را به‌صورت زیر نشان داد:

$$V_i(t) = W \times V_i(t-1) + c_1 \times r_1 \times (pbest_i - X_i(t-1)) + c_2 \times r_2 \times (gbest - X_i(t-1)) \quad (19)$$

$$X_i(t) = X_i(t-1) + V_i(t) \quad (20)$$

بنابراین در هر مرحله هر ذره i مقداری در جهت سرعت قبلی $V_i(t-1)$ ، مقداری در جهت بهترین موقعیت تجربه شده و مقداری در جهت بهترین ذره حرکت می‌کند که برآیند این سه سرعت، سرعت مرحله جدید می‌باشد. و همچنین موقعیت جدید از حاصل جمع موقعیت فعلی $X_i(t-1)$ با سرعت جدید ایجاد می‌شود. که در آن w وزن اینرسی، r_1 و r_2 دو عدد تصادفی که دارای توزیع یکنواخت در بازه $(0,1)$ هستند و به ذرات این اجازه را می‌دهند که در گام‌های تصادفی در محدوده بین $G\text{-Best}$ و $P\text{-Best}$ حرکت کنند و c_1 و c_2 که به ترتیب باعث می‌شوند ذره به سمت $P\text{-Best}$ بهترین موقعیتی که تاکنون کسب کرده و $G\text{-Best}$ بهترین موقعیتی که تاکنون کل ذرات کسب کرده‌اند حرکت کنند، می‌باشند.

۳-۱- الگوریتم بهینه‌سازی انبوه ذرات پیشنهادی

ساختار کلی الگوریتم

ساختار کلی الگوریتم پیشنهادی همانند الگوریتم حرکت توده ذرات

$$Sm_{i',k'} + (1 - Z_{i',p,k'}) \cdot L \geq S_p \quad \forall p, k', i' \quad (14)$$

$$\sum_{i'} \sum_{k'} Z_{i',p,k'} = 1 \quad \forall p \quad (15)$$

$$x_{i,j,p,h,k} \in \{0,1\} \quad \forall p, j, h, k, i \quad (16)$$

$$Z_{i',p,k'} \in \{0,1\} \quad \forall p, k', i' \quad (17)$$

$$C_p \geq 0 \quad \forall p \quad (18)$$

معادله (۱) تابع هدف مدل که حداقل سازی زمان تکمیل کارها می‌باشد را نشان می‌دهد. محدودیت (۲) مقدار تابع هدف را تعریف می‌نماید. محدودیت‌ها (۳)، محدودیت‌های پیشنیازی را معرفی می‌نماید. این محدودیت زمان شروع هر عملیات را به اتمام عملیات قبلی از همان کار محدود می‌نماید. محدودیت (۴)، این الزام را به‌وجود می‌آورد که، در صورتی یک ماشین می‌تواند به کار در نوبت $k+1$ پردازد که کار آن ماشین در نوبت k به اتمام رسیده باشد. محدودیت (۵) و (۶) این الزام را به‌وجود می‌آورند که در صورتی یک عملیات می‌تواند شروع شود که هم محدودیت پیشنیازی (عملیات قبلی آن به اتمام رسیده باشد) و هم محدودیت ماشین (ماشین مورد نظر بیکار باشد) رعایت شده باشند. محدودیت (۷) موجب می‌گردد که به هر ماشین در هر نوبت، تنها یک عملیات از یک کار اختصاص یابد. محدودیت (۸) موجب می‌گردد که برای اجرای یک عملیات از ماشینی اختصاص یافته به آن عملیات استفاده گردد. محدودیت (۹) بزرگترین زمان پردازش قطعات یک محصول را نشان می‌دهد. محدودیت (۱۰) حداقل زمان شروع مرحله مونتاژ را نشان می‌دهد. محدودیت (۱۱) بیانگر زمان تکمیل محصولات است. محدودیت (۱۲) این الزام را به‌وجود می‌آورد که، در صورتی یک ماشین می‌تواند به کار در نوبت $k+1$ پردازد که کار آن ماشین در نوبت k به اتمام رسیده باشد. محدودیت (۱۳) و (۱۴) این الزام را به‌وجود می‌آورند که در صورتی یک عملیات می‌تواند شروع شود که هم محدودیت پیشنیازی (عملیات قبلی آن به اتمام رسیده باشد) و هم محدودیت ماشین (ماشین مورد نظر بیکار باشد) رعایت شده باشند. محدودیت‌های (۱۵) موجب می‌گردد که به هر ماشین در هر نوبت، تنها یک عملیات از یک کار اختصاص یابد. محدودیت‌های (۱۶) و (۱۷) متغیرهای مسئله را به‌صورت ۰ و ۱ تعریف می‌کند. محدودیت (۱۸) مشخص می‌کند که زمان تکمیل هیچ قطعه‌ای نمی‌تواند منفی باشد.

۳-۲ الگوریتم پیشنهادی

با توجه به مدل ریاضی که در بخش گذشته ارائه شد و محدودیت‌های متعدد آن، حل مسئله به‌وسیله روش‌های حل دقیق و نرم‌افزار بهینه‌سازی موجود، بسیار سخت به نظر می‌رسد. بنابراین به

1. Personal-Best

2. Global- Best

ماشین‌ها در زیرمسئله مونتاژ را نشان می‌دهد. شکل زیر نحوه نمایش جواب را نشان می‌دهد.

۲	۴	۱	۳	۱	۳	۲	۴	۲	۱
---	---	---	---	---	---	---	---	---	---

مرحله ۲ (مونتاژ موازی) | مرحله ۱ (تولید کارگاهی)

شکل (۱): نحوه نمایش جواب‌ها

در این حالت به تعداد عملیات‌های هر قطعه عدد مربوط به آن کار در جواب تکرار می‌شود. به‌عنوان مثال در شکل (۱) قطعه ۱ دارای ۲ عملیات می‌باشد که به ترتیب در اولویت‌های سوم و پنجم قرار دارند. همچنین این چهار قطعه ۲ محصول را ایجاد می‌کنند که ۲ خانه آخر در کروموزوم نشان‌دهنده این است که محصول اول روی ماشین دوم و محصول دوم روی ماشین اول مونتاژ می‌شود.

نحوه رمزگشایی جواب‌ها

به‌منظور رمزگشایی جواب‌ها با دو زیرمسئله مواجه خواهیم شد. در ارتباط با زیرمسئله تولید کارگاهی بهترین رویه رمزگشایی رویه فعال می‌باشد که روش‌های مختلفی برای این نوع رمزگشایی وجود دارد. این رویه از رمزگشایی در حین از دست ندادن بهترین جواب، موجب کوچکتر شدن فضای جواب می‌شود. در این مقاله جهت رمزگشایی جواب‌ها از رویه رمزگشایی فعال گلیفرو تامسون [۳۳]، تحت عنوان GT، استفاده شده است.

همچنین در ارتباط با زیرمسئله مونتاژ همان‌طور که اشاره شد، توالی محصولات بدین‌صورت خواهد بود که محصولاتی با زودترین زمان شروع به مونتاژ (پردازش عملیات قطعات مربوط به آن زودتر تمام می‌شود)، در اولویت بالاتری جهت مونتاژ قرار می‌گیرند. همچنین تخصیص عملیات‌ها به ماشین‌ها با توجه به مطالب اشاره شده صورت می‌گیرد.

تطبيق الگوریتم انبوه ذرات با فضای گسسته

الگوریتم توده ذرات برای حل مسائل با فضای پیوسته طراحی شده است، اما به‌دلیل کارایی بالای این الگوریتم تاکنون مسائل گسسته زیادی نیز با آن حل شده است. در الگوریتم پیشنهادی بعد از هر بار جابه‌جایی ذرات، ابتدا هر کدام از آنها به یک جواب گسسته قابل قبول تبدیل می‌شوند و سپس مقدار تابع هدف برای آنها محاسبه می‌شود. بدین‌منظور برای هر ذره به ترتیب از چپ به راست هر کدام از درایه‌ها به نزدیک‌ترین عدد صحیح ممکن تبدیل می‌شود. شکل (۲) یک ذره بعد از جابجایی و همچنین تبدیل آن به یک جواب گسسته قابل قبول را نشان می‌دهد.

برای تولید جواب قابل قبول در این مرحله همان‌طور که در شکل مشخص است نزدیک‌ترین عدد صحیح به عدد ۴.۸ عدد صفر است، اما چون قطعه صفر نداریم، عدد ۴.۸ به ۱ تبدیل می‌شود. همچنین نزدیک‌ترین عدد صحیح به عدد ۳.۵۵ عدد ۴ است اما چون

می‌باشد. بنابراین جهت شروع به کار الگوریتم، پس از تولید جمعیت اولیه برای هر کدام از ذرات یک سرعت اولیه در نظر گرفته می‌شود و بنابر فرمول عنوان شده به جستجو در فضای جواب می‌پردازد.

تولید جمعیت اولیه

در مسئله موردنظر تولید جمعیت اولیه دارای دو بخش است. به عبارت دیگر برای هر عضو از جمعیت باید ساختار توالی آن در زیرمسئله تولید کارگاهی و همچنین نحوه چیدمان محصولات در زیرمسئله مونتاژ مشخص شود. در این حالت برای زیرمسئله تولید کارگاهی از سه رویه و برای زیرمسئله مونتاژ نیز از یک رویه استفاده شده است.

چیدمان تولید کارگاهی

به‌منظور تولید جمعیت اولیه برای هر عضو از جمعیت در زیرمسئله تولید کارگاهی از سه رویه زیر استفاده شده است:

۱. **بیشترین کار باقیمانده:** در این شیوه هر کار که مجموع زمان پردازش عملیات‌های باقیمانده آن، بیشتر باشد دارای اولویت بالاتری است و اولین عملیات قابل پردازش آن با توجه به روابط پیش‌نیازی، در زودترین اولویت موجود قرار می‌گیرد. سپس بعد از تعیین اولویت هر عملیات، مجموع زمان پردازش عملیات‌های باقیمانده کار مربوط به آن بروز می‌شود.

۲. **بیشترین تعداد عملیات باقیمانده:** در این روش هر کار که عملیات‌های باقیمانده آن بیشتر باشد، دارای اولویت بالاتری است و اولین عملیات قابل پردازش آن با توجه به روابط پیش‌نیازی، در زودترین اولویت موجود قرار می‌گیرد. سپس بعد از تعیین اولویت هر عملیات، تعداد عملیات‌های باقیمانده کار مربوط به آن بروز می‌شود.

۳. **توالی تصادفی:** در این شیوه هر عملیات با رعایت شرط پیش‌نیازی به‌صورت تصادفی در یکی از اولویت‌های در دسترس قرار می‌گیرد.

چیدمان خطوط مونتاژ

توالی در خطوط مونتاژ موازی بدین‌صورت خواهد بود که هر محصول که زودتر آماده مونتاژ شود، (پردازش عملیات قطعات مربوط به آن زودتر تمام شود) مونتاژ آن نیز زودتر شروع گردد. تولید جمعیت اولیه در زیرمسئله مونتاژ محصولات به ترتیب، تصادفی انتخاب می‌شوند. سپس بعد از انتخاب هر محصول، محصول به ماشین با کمترین بار کاری اختصاص می‌یابد و بار کاری ماشین‌ها بعد از تخصیص هر محصول بروز می‌شود و این روند تا اختصاص آخرین محصول انتخابی ادامه خواهد داشت.

نحوه نمایش جواب‌ها

همان‌طور که اشاره شد هر عضو از جمعیت دارای دو بخش است که بخش اول توالی عملیات‌ها روی ماشین‌ها در زیرمسئله تولید کارگاهی را نشان می‌دهد و بخش دوم تخصیص محصولات به

۳-۲- الگوریتم توده ذرات با رویکرد بیش‌ترین زمان پردازش^۱ پردازش^۱ (LPT) در خطوط مونتاژ

تمامی مراحل در این الگوریتم مشابه حالت قبل است با این تفاوت که در این حالت نمایش جواب فقط شامل زیر مسئله تولید کارگاهی می‌شود و توالی و تخصیص محصولات در خطوط مونتاژ با رویکرد LPT انجام خواهد شد. شکل (۳) نحوه نمایش جواب در الگوریتم با رویکرد LPT را در مقایسه با الگوریتم پیشنهادی در بخش قبل نشان می‌دهد.

۲	۴	۱	۳	۱	۳	۲	۴
---	---	---	---	---	---	---	---

مرحله (تولید کارگاهی)

شکل (۳): نحوه نمایش جواب‌ها در الگوریتم پیشنهادی با رویکرد LPT

در رویکرد LPT برای حالت مونتاژ موازی ابتدا زمان مونتاژ هر محصول را با زودترین زمان شروع ممکن آن جمع می‌کنیم و بدین‌صورت زمان مونتاژ محصولات تعدیل می‌گردد. سپس برای تعیین توالی در این حالت، اولویت با عملیات با بیشترین زمان مونتاژ تعدیل شده است. در این روش تعیین تخصیص بدین‌صورت خواهد بود، که هر محصول از میان خطوط مونتاژی که زمان شروع ممکن آن‌ها از زودترین زمان ممکن شروع محصول کوچک‌تر باشد، خط مونتاژی را انتخاب می‌کند که زمان شروع ممکن آن کوچک‌تر باشد.

۳-۳- الگوریتم توده ذرات با کم‌ترین زمان پردازش^۲ (SPT) در خطوط مونتاژ

این حالت مشابه با حالت قبل می‌باشد با این تفاوت که در تعیین توالی در این حالت، اولویت با عملیات با کمترین زمان پردازش تعدیل شده است.

۳-۴- الگوریتم ژنتیک

تولید جمعیت اولیه، نحوه نمایش جواب (کروموزوم) مورد استفاده و نحوه محاسبه تابع هدف در الگوریتم ژنتیک مقایسه‌ای مورد استفاده، همانند الگوریتم توده ذرات پیشنهادی می‌باشد که در ادامه به توضیح دیگر قسمت‌های الگوریتم می‌پردازیم.

انتخاب والد

در این روش به هر یک از اعضای گروه، قطعه‌ای از یک چرخ رولت مدور اختصاص داده می‌شود. قطعه اختصاص داده شده متناسب با میزان برزندگی عضو می‌باشد. بدین معنی که هر چه برزندگی یک عضو بیشتر باشد میزان اختصاص فضا به این عضو بیشتر است. اعضا به طور تصادفی از این چرخ رولت انتخاب می‌شوند. در نتیجه هر چه برزندگی یک عضو بیشتر باشد، شانس انتخاب آن نیز بیشتر می‌شود.

عدد ۴ دو بار ایجاد شده و قطعه چهار نیز فقط دو عملیات دارد بنابراین عدد ۳.۵۵ به نزدیک‌ترین عدد صحیح ممکن یعنی ۳ تبدیل می‌شود. و بدین صورت جواب گسسته قابل قبولی ارائه می‌گردد.

۲.۳۶	۲	۲
۱.۱۴	۱	۱
۱.۶۹	۲	۲
۰.۴۸	۰	۱
۳.۰۸	۳	۳
۳.۵۹	۴	۴
۳.۹۶	۴	۴
۳.۵۵	۴	۳
۲.۲۱	۲	۲
۱.۲۵	۱	۱

تبدیل جواب از حالت پیوسته به گسسته (جواب نشدنی) جواب شدنی

شکل (۲): تبدیل جواب پیوسته به یک جواب گسسته قابل قبول

ساختارهای همسایگی

در این الگوریتم در هر مرحله بعد از به‌روزرسانی P-Best و G-Best دو ساختار همسایگی بر روی G-Best اعمال می‌شود.

➤ ساختار همسایگی توالی

این ساختار روی زیرمسئله تولید کارگاهی اعمال می‌شود. بدین‌صورت که از میان بلوک‌های بحرانی با بیش از یک عملیات، یکی را انتخاب و با احتمالی برابر با عملیات ابتدایی یا انتهایی آن را برمی‌گزینیم. اگر عملیات ابتدایی انتخاب شد با عملیات دوم یا سوم بلوک جابجا می‌شود و چنانچه عملیات انتهایی انتخاب شد با عملیات ۱-k یا ۲-k بلوک جابجا می‌شود. در این قسمت k تعداد عملیات‌های بلوک انتخابی را نشان می‌دهد. چنانچه بلوک بحرانی انتخابی دارای دو عملیات باشد، جای آن دو عملیات باهم جابجا می‌شود. این ساختار همسایگی می‌تواند باعث کاهش تعداد مسیرهای بحرانی زیرمسئله تولید کارگاهی و متعاقباً کاهش تابع هدف دوره ساخت شود.

➤ ساختار همسایگی تخصیص

این ساختار همسایگی بر روی زیرمسئله مونتاژ اعمال می‌شود. بدین‌صورت که از میان عملیات‌های مربوط به خط مونتاژ با بار کاری بالاتر، یک محصول به تصادف انتخاب می‌شود و به تصادف به یکی دیگر از خطوط مونتاژ تخصیص می‌یابد. این ساختار همسایگی می‌تواند باعث تعدیل بار کاری در خطوط مونتاژ و کاهش تابع هدف دوره ساخت شود.

1. Longest Processing Time
2. Shortest Processing Time

ترکیب و ایجاد جمعیت جدید

نمی‌توان کل این چهار کروموزوم را به جمعیت برگرداند و در نتیجه باید دو کروموزوم حذف گردد. بنابراین فرزندان تولید شده و والدین بر اساس تابع هدف مرتب می‌شوند و ۷۰٪ از بهترین‌ها به مرحله بعد می‌روند و برای ۳۰٪ دیگر جمعیت از کل جمعیت باقی‌مانده انتخاب می‌شوند.

۴- نتایج محاسباتی

در این قسمت به کمک نتایج محاسباتی، عملکرد و کارایی الگوریتم‌های موردنظر بررسی و مقایسه می‌گردد. مدل ریاضی مسئله در نرم‌افزار LINGO کدنویسی و حل شده است. الگوریتم‌های فراابتکاری پیشنهادی با استفاده از نرم‌افزار MATLAB (R2007a) کدنویسی و اجرا شده است. با توجه به این‌که اطلاعات مربوط به مسائل حل شده در مقالات مشابه برای مسئله موردنظر وجود ندارد، و به منظور اعتبارسنجی مدل از نمونه‌های تصادفی برای آن استفاده شده است که نحوه تولید آن‌ها در جدول (۱) آورده شده است. همچنین هر نمونه به‌صورت a-b-c-d-e نمایش داده می‌شود که a نشان‌دهنده تعداد قطعات، b نشان‌دهنده تعداد محصولات، c نشان‌دهنده تعداد ماشین در مرحله اول، d نشان‌دهنده تعداد ماشین‌ها در مرحله دوم و e نشان‌دهنده حداکثر عملیات روی یک قطعه می‌باشد.

جدول (۱): نحوه تولید داده برای نمونه‌های تصادفی

مقادیر	نمادها	پارامترها
{۲,۳,...,۵۰}	p	تعداد محصولات
{۵,۴,...,۱۰۰}	n	تعداد قطعات
{۵,۴,...,۲۰}	m	تعداد ماشین‌آلات در مرحله ۱
{۲,۳,۴}	m'	تعداد ماشین‌آلات در مرحله ۲
یکنواخت پیوسته	ps	زمان پردازش
یکنواخت پیوسته	A_p	زمان مونتاژ

علاوه بر این تنظیم مناسب پارامترها و انتخاب درست عملگرها در الگوریتم‌های فراابتکاری تأثیر بسزایی بر عملکرد و کارایی آن‌ها دارد که باید با توجه به شرایط مسئله مورد بررسی، تعیین گردد. با این وجود بیشتر محققین این تنظیمات را با استفاده از مقادیر مرجع که در مقالات مشابه پیشنهاد شده انجام می‌دهند. در این تحقیق از روش تاگوچی جهت تنظیم پارامترها یا الگوریتم استفاده شده است. در جدول (۲) پارامترهای قابل کنترل الگوریتم بهینه‌سازی انبوه ذرات و تعداد سطوح مناسب آن‌ها آورده شده‌است، در این الگوریتم ۷ پارامتر مورد بررسی قرار می‌گیرد. برای الگوریتم PSO برای هر پارامتر دو سطح معرفی شده‌است که آرایه مناسب آن $Lg(2^7)$ می‌باشد. همچنین نتایج حاصل از تنظیم پارامترها به روش تاگوچی در شکل (۷) آورده شده است.

در این مسئله ترکیب شامل دو قسمت می‌باشد. ابتدا برای قسمت اول کروموزوم که توالی عملیات‌های زیرمسئله تولید کارگاهی را نشان می‌دهد یک عدد تصادفی در بازه $[1, H]$ ، که H مجموع تعداد عملیات‌های زیرمسئله است، تولید می‌کنیم و آن را A می‌نامیم. از میان عملیات‌های زیرمسئله، A عملیات به تصادف انتخاب می‌کنیم. ترتیب این عملیات‌ها در والد اول براساس والد دوم تغییر کرده و قسمت اول فرزند اول را تولید می‌کند و همچنین ترتیب این عملیات‌ها در والد دوم براساس والد اول تغییر کرده و قسمت اول فرزند دوم را تولید می‌کند.

مشابه با حالت قبل برای زیرمسئله مونتاژ نیز، یک عدد تصادفی در بازه $[1, P]$ ، که P تعداد محصولات زیرمسئله است تولید می‌کنیم و آن را B می‌نامیم. از میان محصولات زیرمسئله، B محصول به تصادف انتخاب می‌کنیم. سپس خط مونتاژ تخصیص یافته این محصولات در والد اول براساس والد دوم تغییر کرده و قسمت دوم فرزند اول را تولید می‌کند و همچنین خط مونتاژ تخصیص یافته این محصولات در والد دوم براساس والد اول تغییر کرده و قسمت دوم فرزند دوم را تولید می‌کند.

همان‌طور که در شکل (۴) مشاهده می‌شود عملیات اول قطعات ۲ و ۵ و ۶ و عملیات دوم قطعات ۸ و ۱ در زیرمسئله تولید کارگاهی و محصول اول و دوم نیز در زیرمسئله مونتاژ انتخاب شده‌اند. در این شکل چگونگی ایجاد فرزندان نیز نشان داده شده است.

۱	۲	۳	۴	۵	۸	۷	۶	۳	۴	۲	۵	۸	۱	۷	۶	۱	۲	۲	۱
والد اول																			
۸	۵	۱	۳	۷	۴	۶	۲	۱	۳	۲	۵	۸	۴	۶	۷	۲	۲	۱	۱
والد دوم																			
۱	۵	۳	۴	۶	۸	۷	۲	۳	۴	۲	۵	۱	۸	۷	۶	۱	۲	۱	۱
فرزند اول																			
۸	۲	۱	۳	۷	۴	۵	۶	۸	۳	۲	۵	۱	۴	۶	۷	۲	۲	۲	۱
فرزند دوم																			

شکل (۴): کروموزوم‌های والد و فرزند

جهش

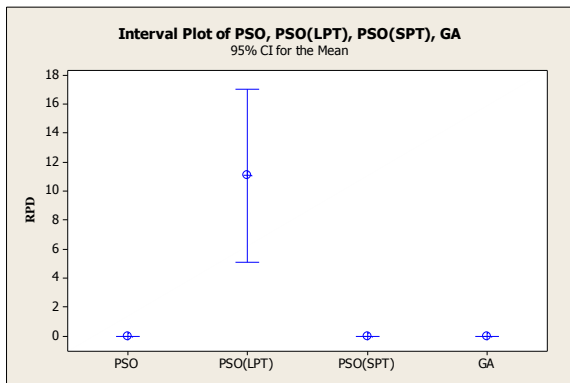
به منظور اجرای رویه جهش، دو ژن متفاوت به طور تصادفی انتخاب می‌شوند سپس ارزش‌های مرتبط با آن‌ها با یکدیگر تعویض می‌گردند. در این حالت عددی برابر p در بازه $[0, 1]$ تولید می‌شود چنان‌چه این مقدار کمتر از $1/n$ ، که در آن n نشان‌دهنده ژن‌های کروموزوم می‌باشد، باشد این دو مقدار با یکدیگر تعویض می‌گردند و کروموزوم‌ها بعد از تعویض این دو ژن جهش می‌یابند.

انتخاب جمعیت

جایجایی آخرین مرحله از چرخه تولید مثل است. دو والد از یک جمعیت با اندازه ثابت، انتخاب شده و دو فرزند به‌وجود آورده‌اند. اما

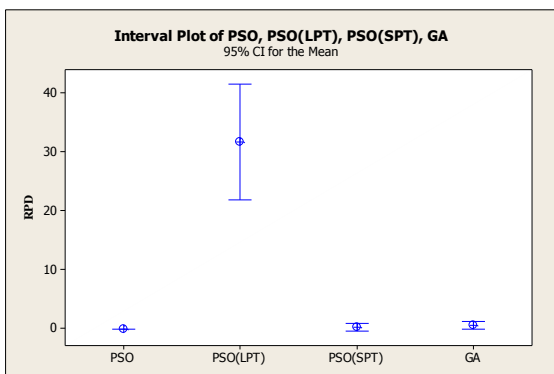
دست می‌یابد. علاوه بر این برای مسائل با ابعاد بزرگ (جدول ۵) الگوریتم‌های PSO پیشنهادی و PSO با ساختار SPT به نتایج بهتری نسبت به سایر الگوریتم‌ها دست می‌یابند علاوه بر این مشاهده می‌شود که الگوریتم PSO با ساختار SPT در مدت زمان کم‌تر به جواب موردنظر دست می‌یابد. همچنین باتوجه به شکل (۶) عملکرد ضعیف الگوریتم PSO(LPT) نسبت به سایر الگوریتم‌ها قابل مشاهده است.

با توجه به مقادیر RPD در مسائل با ابعاد متوسط (شکل ۵)، برتری الگوریتم‌های PSO پیشنهادی، PSO با ساختار SPT و GA نسبت به الگوریتم PSO با ساختار LPT قابل مشاهده است. از طرفی با توجه به این‌که LINGO در محدوده زمانی مناسب به جواب بهینه دست نیافته و الگوریتم‌ها PSO پیشنهادی و PSO با ساختار SPT و GA در مدت زمان کم‌تری به جواب‌های بهتری نسبت به LINGO رسیده‌اند، کارایی بالای روش‌های حل پیشنهادی قابل بیان است.



شکل (۵): نمودار میانگین RPD با بازه اطمینان ۹۵٪ برای الگوریتم‌های PSO پیشنهادی با GA در مسائل متوسط

با توجه به مقادیر RPD با بازه اطمینان ۹۵٪ در شکل (۶)، قابل مشاهده است که الگوریتم PSO پیشنهادی از نظر آماری بهتر از الگوریتم‌های دیگر عمل می‌کند.



شکل (۶): نمودار میانگین RPD با بازه اطمینان ۹۵٪ برای الگوریتم‌های PSO پیشنهادی با GA در مسائل بزرگ

جدول (۲): فاکتورها و مقادیر مناسب آن‌ها برای الگوریتم PSO

Factor	Description	Level
A	Inertia weight {[.4,.9], [0.729,1]}	2
B	C1,C2{2,1.499}	2
C	Swarm size{100,200}	2
D	Variable interval{[-4,4],[-5,5]}	2
E	Vmax,Vmin{[-4,4],[-5,5]}	2
F	Iteration{200,400}	2
G	Random number{.5,.7}	2

در نهایت عملکرد روش‌های حل ارزیابی و با یکدیگر مقایسه می‌شود. برای ارزیابی روش‌های حل پیشنهادی، نمونه‌ها به سه گروه با اندازه کوچک، اندازه متوسط و اندازه بزرگ تقسیم شده‌اند. برای مقایسه روش‌های حل، مقادیر به دست آمده از هر روش به معیار عملکرد (RPD) با استفاده از رابطه زیر تبدیل شده است [۱].

$$RPD = \frac{Alg_{sol} - Min_{sol}}{Min_{sol}} \times 100 \quad (21)$$

که Alg_{sol} تابع هدف حاصل از حل یک نمونه با استفاده از الگوریتم موردنظر و Min_{sol} مینیمم مقدار تابع هدف حاصل از حل آن نمونه با استفاده از روش‌های حل مورد مقایسه می‌باشد. مقادیر موجود در ستون $Imp\%$ درصد بهبود جواب اولیه را نشان می‌دهد که از رابطه زیر محاسبه می‌شود.

$$IMP = \frac{Alg_{initial sol} - Alg_{final sol}}{Alg_{initial sol}} \times 100 \quad (22)$$

که $Alg_{initial sol}$ مقدار تابع هدف جواب اولیه الگوریتم و $Alg_{final sol}$ میزان تابع هدف جواب نهایی الگوریتم را نشان می‌دهد. در ستون $CPU\ time(s)$ زمان محاسباتی هر روش درج شده است.

۴-۱- تجزیه و تحلیل

برای مسائل با ابعاد کوچک جواب بهینه توسط LINGO به دست آمد و در جدول ثبت گردید. در مسائل اندازه کوچک (جدول ۳) با توجه به مقادیر RPD نشان داده می‌شود که الگوریتم‌های PSO پیشنهادی، PSO با ساختار SPT، PSO با ساختار LPT و GA به جواب بهینه، مشابه با جواب حاصل از LINGO دست می‌یابند و این در حالی است که زمان محاسباتی حاصل از حل این الگوریتم‌ها به مراتب کم‌تر از LINGO می‌باشد. همچنین در مسائل با ابعاد متوسط با توجه به مقادیر RPD (شکل ۵)، برتری الگوریتم‌های PSO پیشنهادی، PSO با ساختار SPT و GA نسبت به PSO با ساختار LPT قابل مشاهده است که مقادیر RPD نزدیکی الگوریتم‌ها به بهترین جواب بدست آمده از سایر الگوریتم‌ها را نشان می‌دهد. همچنین با توجه به جواب مشابه حاصل از الگوریتم‌های PSO پیشنهادی، PSO با ساختار SPT و GA (جدول ۴) در صورت اهمیت زمان محاسباتی الگوریتم ژنتیک در ارجحیت است. همچنین می‌توان در شکل (۱۰) مشاهده کرد که الگوریتم PSO با ساختار SPT در مدت زمان کم‌تری به نتایج حاصل از الگوریتم PSO پیشنهادی

داشت که الگوریتم PSO با ساختار SPT از عملکرد بهتری نسبت به الگوریتم PSO پیشنهادی برخوردار است که ساختار SPT هر چند حداقل زمان را بر روی ماشین‌های موازی تضمین نمی‌کند اما منجر به ساختاری خوب و قابل قبولی می‌گردد که نسبت به حالت تصادفی در PSO پیشنهادی، از لحاظ زمانی بهتر عمل می‌کند.

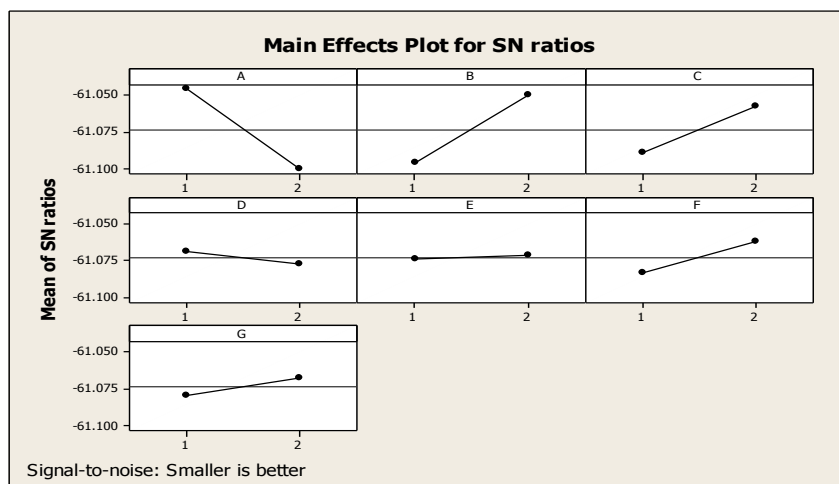
۵- نتیجه‌گیری

در این تحقیق حل مسئله تولید کارگاهی همراه با یک مرحله مونتاژ موازی مورد بررسی قرار گرفت. به‌علت پیچیدگی بالای مسئله مورد بررسی برای حل مسئله به‌صورت دقیق، از الگوریتم‌های فراابتکاری جهت حل مسئله استفاده شد. همچنین نتایج محاسباتی عملکرد مناسب الگوریتم‌های PSO پیشنهادی و PSO با ساختار SPT را نسبت به سایر الگوریتم‌ها در کسب جواب‌های با کیفیت، و برتری PSO با ساختار SPT نسبت به PSO پیشنهادی از نظر زمان محاسباتی را نشان داد. علاوه بر این، الگوریتم فراابتکاری GA با زمان‌های محاسباتی کمتر نسبت به سایر الگوریتم‌ها به جواب‌های قابل قبول دست یافت. اما طی انجام محاسبات الگوریتم PSO(LPT) در تمام ابعاد مسئله به نتایج محاسباتی ضعیف‌تری دست پیدا کرد که با توجه به زمان ورود نابرابر این موضوع چندان دور از انتظار نیست. همچنین عملکرد خوب الگوریتم‌های PSO پیشنهادی، PSO(SPT) و GA به مرور و با افزایش ابعاد مسئله قابل مشاهده بود. محورهای متعددی جهت تحقیقات آتی قابل پیشنهاد می‌باشد از مهم‌ترین آن‌ها در نظر گرفتن جریان محموله در صورت در نظر گرفتن تعداد بیش از یک محصول و در نظر گرفتن عدم قطعیت و حالت‌های احتمالی برای پارامترهای مسئله است. همچنین می‌توان از الگوریتم‌های فراابتکاری دیگری برای حل مدل استفاده کرد.

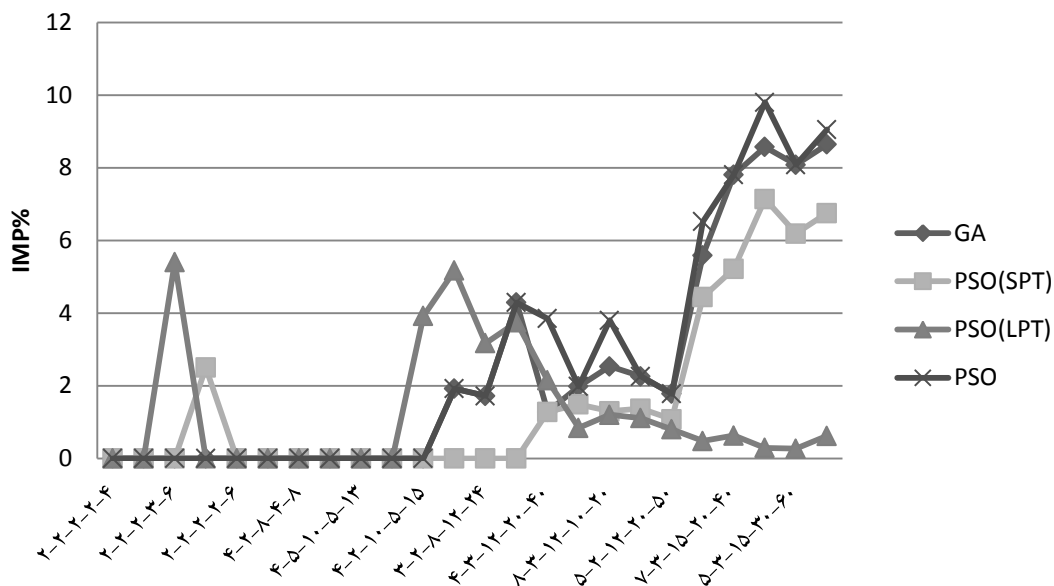
در شکل (۸) عملکرد الگوریتم‌های پیشنهادی در بهبود جواب اولیه مورد مقایسه قرار گرفته است که در این شکل مقادیر بالای بهبود نشان‌دهنده عملکرد و کارایی مناسب در بهبود جواب اولیه می‌باشد. همان‌گونه که در شکل مشخص می‌باشد، میزان بهبود جواب اولیه در الگوریتم PSO پیشنهادی با افزایش ابعاد مسئله، در اغلب موارد بیش‌تر از سایر روش‌ها می‌باشد. همچنین قابل مشاهده است که با افزایش ابعاد مسئله الگوریتم PSO با ساختار LPT نسبت به سایر الگوریتم‌ها از کارایی پایین‌تری برخوردار است. مقادیر عدم بهبود در مسائل با ابعاد متوسط نشان‌دهنده این موضوع است که الگوریتم‌ها در همان تکرار اول به مقدار بهینه دست می‌یابند و همان‌طور که انتظار می‌رود به مرور الگوریتم‌ها با کاوش بیش‌تر به مقدار بهتری نسبت به مقدار اولیه خود دست می‌یابند.

در شکل (۹) جواب‌های حاصل از حل الگوریتم‌های ارائه شده در تمامی مسائل با یکدیگر مقایسه گردیده است. همان‌طور که در شکل نشان داده شده است الگوریتم PSO با ساختار LPT با افزایش در ابعاد مسئله عملکردی ضعیف‌تر نسبت به سایر الگوریتم‌های ارائه شده، داشته است. همچنین با توجه به جواب‌های حاصل از الگوریتم PSO پیشنهادی با الگوریتم PSO با ساختار SPT می‌توان گفت که الگوریتم PSO با ساختار SPT مشابه الگوریتم PSO پیشنهادی عمل می‌کند. علاوه بر این با توجه به شکل (۹) افزایش فاصله بین بهترین جواب‌های بدست آمده و جواب حاصل از الگوریتم GA با افزایش در ابعاد مسئله دیده می‌شود.

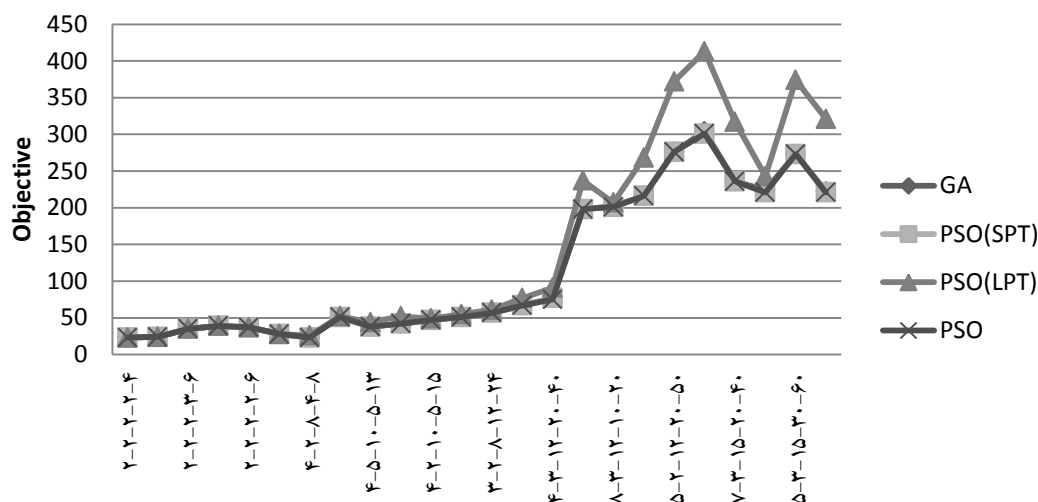
در شکل (۱۰) زمان‌های محاسباتی حاصل از الگوریتم‌های ارائه شده مورد مقایسه قرار گرفته است که با توجه به این شکل، الگوریتم GA کم‌ترین زمان محاسباتی و الگوریتم PSO پیشنهادی بیش‌ترین زمان محاسباتی در بین الگوریتم‌ها را دارا می‌باشد. همچنین با توجه به عملکرد مشابه بین الگوریتم‌های PSO پیشنهادی و PSO با ساختار SPT در رسیدن به جواب (شکل ۸) و کم‌تر بودن زمان محاسباتی حاصل از الگوریتم PSO با ساختار SPT، می‌توان بیان



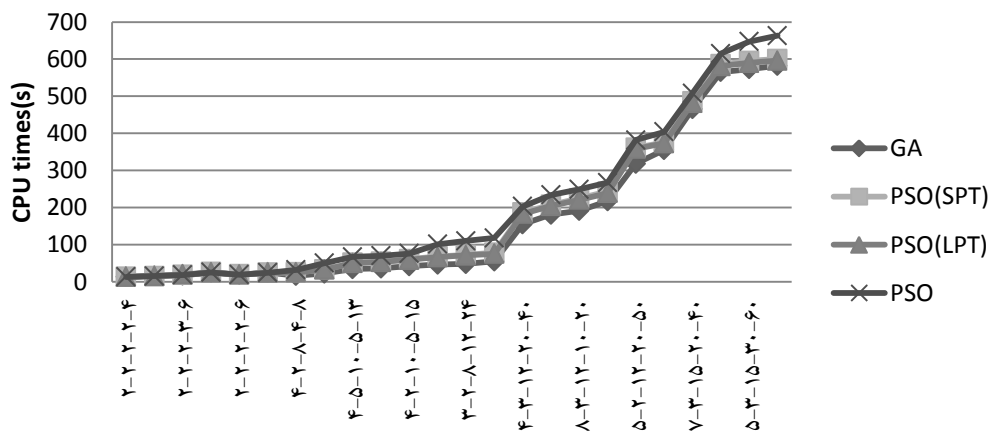
شکل (۷): نمودار میانگین مقادیر SN ratio برای هر سطح از فاکتورها در الگوریتم PSO



شکل (۸): درصد بهبود جواب‌های اولیه الگوریتم‌های PSO پیشنهادی با GA



شکل (۹): مقایسه جواب‌های حاصل از الگوریتم‌های PSO پیشنهادی با GA



شکل (۱۰): زمان‌های محاسباتی الگوریتم‌های PSO پیشنهادی با GA

جدول (۳): نتایج محاسباتی حل نمونه‌های اندازه کوچک

Instance	LINGO		PSO			PSO (LPT)			PSO (SPT)			GA		
	CPU time(s)	RPD%	CPU time(s)	IMP%	RPD%	CPU time(s)	IMP%	RPD%	CPU time(s)	IMP%	RPD%	CPU time(s)	IMP%	RPD%
۴-۲-۲-۲-۲	۳	.	۱۳	.	.	۱۳	.	.	۱۳	.	.	۱۳	.	.
۴-۲-۳-۲-۲	۱۲	.	۱۵	.	.	۱۵	.	.	۱۵	.	.	۱۶	.	.
۶-۳-۲-۲-۲	۲۱۰	.	۱۸	.	.	۲۰	۵.۴	.	۱۸	.	.	۱۷	.	.
۶-۳-۲-۳-۲	۵۰۲	.	۲۵	.	.	۲۵	.	.	۲۶	۲.۵	.	۲۵	.	.
۶-۲-۲-۲-۲	۱۰۸	.	۱۹	.	.	۱۹	.	.	۲۰	.	.	۱۹	.	.
۶-۲-۳-۲-۳	۲۰۵	.	۲۴	.	.	۲۴	.	.	۲۴	.	.	۲۵	.	.

جدول (۴): نتایج محاسباتی حل نمونه‌های اندازه متوسط

Instance	LIN GO	PSO		PSO (LPT)			PSO (SPT)			GA				
		RPD%	CPU time(s)	IMP%	RPD%	CPU time(s)	IMP%	RPD%	CPU time(s)	IMP%	RPD%			
۸-۴-۸-۲-۴	.	۳۲	.	.	.	۱۳۰۴۳	.	.	۲۵	.	.	۱۶	.	.
۸-۴-۸-۲-۵	.	۵۰	.	.	.	۳۳	.	۱.۹۶۱	۳۳	.	.	۲۲	.	.
-۵-۱۰-۵-۴ ۱۳	۵.۲۶	۶۷	.	.	۵۰	.	۱۵.۷۸۹	۵۱	.	.	۳۴	.	.	
۱۳-۶-۸-۲-۴	۹.۵۲	۷۰	.	.	۵۳	.	۲۳.۸۱۰	۵۴	.	.	۳۶	.	.	
-۵-۱۰-۲-۴ ۱۵	-	۷۶	.	.	۵۹	۳.۹۲	۴.۲۵۵	۶۱	.	.	۴۲	.	.	
۱۸-۶-۸-۳-۴	-	۱۰۱	۱.۹۲	.	۶۶	۵.۱۷	۷.۸۴	۶۸	.	.	۴۶	۱.۹۲	.	
-۱۲-۸-۲-۳ ۲۴	-	۱۱۰	۱.۷۲	.	۷۱	۳.۱۷	۷.۰۲	۷۲	.	.	۴۸	۱.۷۲	.	
-۱۰-۱۰-۲-۴ ۲۰	-	۱۱۸	۴.۲۹	.	۷۵	۳.۷۵	۱۴.۹۳	۷۸	.	.	۵۵	۴.۲۹	.	

جدول (۵): نتایج محاسباتی حل نمونه‌های اندازه بزرگ

Instance	PSO			PSO (LPT)			PSO (SPT)			GA		
	CPU time(s)	IMP %	RPD%	CPU time(s)	IMP%	RPD%	CPU time(s)	IMP%	RPD%	CPU time(s)	IMP%	RPD%
-۲۰-۱۲-۳-۴ ۴۰	۲۰۳	۳.۸۵	.	۱۸۳	۲.۱۵	۲۱.۳۳	۱۸۷	۱.۲۸	۲.۶۷	۱۵۵	۱.۲۸	۲.۶۷
-۱۰-۱۲-۲-۸ ۲۵	۲۳۴	۱.۹۸	.	۲۰۴	۰.۸۴	۱۹.۷۰	۲۰۷	۱.۴۹	.	۱۸۱	۱.۹۸	.
-۱۰-۱۲-۳-۸ ۲۰	۲۴۹	۳.۸۰	.	۲۲۱	۱.۲۰	۷.۸۹	۲۲۵	۱.۳۰	.	۱۹۱	۲.۵۳	۱.۳۲
-۱۵-۱۲-۲-۷ ۳۰	۲۶۷	۲.۲۶	.	۲۳۹	۱.۱۱	۲۴.۰۷	۲۴۳	۱.۳۷	.	۲۱۷	۲.۲۶	.
-۲۰-۱۲-۲-۵ ۵۰	۳۸۲	۱.۷۸	.	۳۵۸	۰.۸۰	۳۴.۸۷	۳۶۱	۱.۰۸	.	۳۱۸	۱.۷۸	.
-۲۵-۱۵-۳-۵ ۵۰	۴۰۴	۶.۵۲	.	۳۷۳	۰.۴۸	۳۷.۲۱	۳۷۵	۴.۴۴	.	۳۵۵	۵.۵۹	۱.۰۰
-۲۰-۱۵-۳-۷ ۴۰	۵۰۸	۷.۸۱	.	۴۸۲	۰.۶۳	۳۴.۳۲	۴۸۶	۵.۲۲	.	۴۶۵	۷.۸۱	.
-۵۰-۲۰-۴-۳ ۱۰۰	۶۱۵	۹.۸۰	.	۵۸۱	۰.۲۹	۵۵.۲۰	۵۸۶	۷.۱۴	.	۵۶۵	۸.۵۷	۱.۳۶
-۳۰-۱۵-۳-۵ ۶۰	۶۴۷	۸.۰۸	.	۵۹۰	۰.۲۷	۳۷.۰۰	۵۹۵	۶.۱۹	.	۵۷۳	۸.۰۸	.
-۴۰-۲۰-۴-۳ ۱۰۰	۶۶۳	۹.۰۵	.	۵۹۶	۰.۶۲	۴۵.۲۵	۶۰۰	۶.۷۵	.	۵۸۲	۸.۶۴	۰.۴۵

مراجع

- International Journal of Advance Manufacture Technology.
- [21] Kao, Y.T., Zahara, E. (2008). A hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Applied Soft Computing*, 8: 849-857.
- [22] Goldberg, D.E. (1998). *Genetic Algorithms in Search, Optimization and Machine learning*, Addison-Wesley, New York.
- [23] Eberhart, R., Kennedy, J., (1995). A new optimizer using particle swarm theory. In: *Proc of the sixth international symposium on micro machine and human science*, Nagoya, Japan; p. 39-43.
- [24] Lian, Z., Gu, X., Jiao, B. (2008). A novel particle swarm optimization algorithm for permutation flow shop scheduling to minimize makespan. *Chaos, Solitons and Fractals*, 35: 851-861.
- [25] Bo, L., Ling, W., Yi-Hui, J., Fang, T., De-Xian, H. (2005). Improved particle swarm optimization combined with chaos. *Chaos, Solitons & Fractals*, 25: 1261-1271.
- [26] Shi, X.H., Liang, Y.C., Lee, H.P., Lu, C., Wang, L.M. (2005). An improved GA and a novel PSO-GA based hybrid algorithm. *Inform Process Lett*; 93: 255-261.
- [27] Kennedy, J., Mendes, R., (2002). Population structure and particle swarm performance. *Proceedings of the 2002 congress on evolutionary computation CEC*. IEEE Press; p. 1671-1676.
- [28] Fan, H.Y. (2002). A modification to particle swarm optimization algorithm. *Eng Comput*, 19:970-989.
- [29] Robinson, J., Sinton, S., Yahya, R.S. (2002). Particle swarm, genetic algorithm, and their hybrids: Optimization of a profiled corrugated horn antenna. In: *IEEE antennas and propagation society international symposium*, San Antonio, 314-317.
- [30] He, S., Wu, Q.H., Wen, J.Y., Saunders, J.R., Paton, R.C. (2004). A particle swarm optimizer with passive congregation. *BioSystems*, 78: 135-147.
- [31] Lu, Z., Shieh, L.S., Chen, G.R. (2003). On robust control of uncertain chaotic systems: a sliding mode synthesis via chaotic optimization. *Chaos, Solitons & Fractals*, 18: 819-827.
- [32] Ji, M.J., Tang, H.W., (2004). Application of chaos in simulated annealing. *Chaos, Solitons & Fractals*; 21: 933-41.
- [۱] واحدی نوری، بهدین (۱۳۹۰). حل مسئله زمانبندی کارگاه گردش کاری با محدودیت دسترسی ماشین و اثر یادگیری. پایان‌نامه کارشناسی‌ارشد، دانشکده مهندسی، دانشگاه بوعلی سینا همدان.
- [2] Reisman, A., Kumar, A., Motwani, J. (1997). Flowshop scheduling/sequencing research: a statistical review of the literature, 1952-1994. *Engineering Management, IEEE Transactions on*, 44:316-329.
- [3] Manne, A.S. (1960). On the job shop scheduling problem. *Operation Research*, 8: 219-223.
- [4] Wagner, H. (1959). An integer linear-programming model for machinescheduling. *Naval Research logistics Quarterly*, 6: 131-140.
- [5] Gonzalez, T., Sahni, S. (1987). Flow shop and Job shop schedule: Complexity and approximation. *Operation Research*, 26: 36-52.
- [6] Garey, M.R., Johnson, D.S., Sethi, R. (1976). The Complexity of flow shop and job shop scheduling. *Mathematics of Operation Research*, 1: 117-129.
- [7] Brucker, P., Jurisch, B., Sievers, B. (1994). A branch and bound algorithm for job-shop scheduling problem. *Discrete Applied Mathematics*, 49: 105-127.
- [8] Gray, C., Hoesada, M. (1991). Matching heuristic scheduling rules for job shops to the business sales level. *Production and Inventory Management Journal*, 4: 12-17.
- [9] Laarhoven, P.J.M.V., Aarts, E.H.L., Lenstra, J.K. (1992). Job shop scheduling by simulated annealing. *Operations Research*, 40: 113-125.
- [10] Nowicki, E., Smutnicki, C. (1996). A fast taboo search algorithm for the job-shop problem. *Management Science*, 42: 797-813.
- [11] Davis, L. (1985). Job shop scheduling with genetic algorithms. *Int. Conf. Genetic Algorithms*.
- [12] Lian, Z., Jiao, B., Gu, X. (2006). A similar particle swarm optimization algorithm for job-shop scheduling to minimize makespan. *Applied Mathematics and Computation*, 183: 1008-1017.
- [13] Lee, C.Y., Cheng, T.C.E., Lin, B.M.T. (1993). Minimizing the makespan in the 3-machine assembly-type flowshop scheduling problem. *Management Science*, 39: 616-625.
- [14] Potts, C.N., Sevast'yanov, S.V., Strusevich, V. A., Van Wassenhove, L.N., Zwaneveld, C.M. (1995). The two-stage assembly scheduling problem: Complexity and approximation. *Operations Research*, 43:346-355.
- [15] Hariri, A.M.A., Potts, C.N. (1997). A branch and bound algorithm for the two-stage assembly scheduling problem, *European Journal of Operational Research*, 103: 547-556.
- [16] Cheng, T.C.E., Wang, G. (1999). Scheduling the fabrication and assembly of components in a two-machine flow shop, *IIE Transactions*, 31: 135-143.
- [17] Yokoyama, M. (2001). Hybrid flow shop scheduling with assembly operations. *International Journal of Production Economics*, 73: 103-116.
- [18] Sun, X., Morizawa, K., Nagasawa, H. (2003). Powerful heuristics to minimize makespan in fixed, 3-machine, assembly-type flow shop scheduling. *European Journal of Operational Research*, 146: 498-516.
- [19] Sung, C.S., Kim, H.A. (2008). A two-stage multiple-machine assembly scheduling problem for minimizing sum of completion times. *International Journal of Production Economics*, 113: 1038-1048.
- [20] Fattahi, P., Hosseini, S.M.H., Jolai, F. (2012). A mathematical model and extension algorithm for assembly flexible flow shop scheduling problem,



A model for job shop scheduling with a parallel assembly stage to minimize makespan

Fatemeh Daneshamoz, Mona Jabbari, Parviz Fattahi*

Department of Industrial Engineering, Collage of Engineering, Bu-Ali Sina University, Hamedan, Iran

ARTICLE INFO

Article history:

Received 5 January 2014
Accepted 12 August 2014

Keywords:

Scheduling
Job shop
Parallel Assembly
Particle Swarm Optimization.

ABSTRACT

In this research, job shop scheduling problem by considering a parallel assembly stage is studied to minimize completion time for all products. At first, a linear model is introduced to describe the problem. To confirm the accuracy of model, it is solved by LINGO software. Since this problem is classified as NP-Hard, a particle swarm optimization algorithm is suggested to solve the problem in large and medium dimensions. To analyze the efficiency of the suggested algorithm, result of the proposed algorithm are compared with the results gained from GA algorithm. This results show that suggested algorithm can reach to near-optimal solutions in various dimensions of problems.

* Corresponding author. Parviz Fattahi
Tel.: +98 81 38257410; E-mail addresses: fattahi@basu.ac.ir