

## تعیین توالی خطوط مونتاژ مدل ترکیبی تحت عدم قطعیت با زمان‌های پردازش تصادفی با استفاده از الگوریتم جست‌وجوی هارمونی بهبود یافته

پرویز فتاحی<sup>\*</sup>، آرزو عسکری<sup>۱</sup>، بهنام عسکری<sup>۲</sup>

۱- دانشیار، دانشگاه الزهراء، گروه مهندسی صنایع، تهران، ایران

۲- کارشناسی ارشد، دانشگاه بوعلی سینا، گروه مهندسی صنایع، همدان، ایران

۳- دانشجوی کارشناسی ارشد، دانشگاه آزاد اسلامی واحد ملایر، گروه مهندسی صنایع، ملایر، ایران

رسید مقاله: ۱۰ خرداد ۱۳۹۶

پذیرش مقاله: ۳۰ آبان ۱۳۹۶

### چکیده

در سال‌های اخیر با پیشرفت روزافزون تکنولوژی، تنوع‌طلبی مشتریان و افزایش رقابت جهانی سیستم‌های سنتی جوابگوی نیازهای مشتریان نیستند. شرکت‌ها برای برآوردن تقاضای مشتریان، پیشی گرفتن از رقبای و بقای هر چه طولانی‌تر در عرصه تولید نیازمند سیستم‌های نوین تولیدی هستند تا هم‌زمان با افزایش کیفیت محصولات تولیدی، هزینه تولید و قیمت محصولات تولیدی را نیز کاهش دهند. یک سیستم نوین تولیدی، خط مونتاژ مدل ترکیبی<sup>۱</sup> (MMAL) است. در این نوع خط مونتاژ که به‌عنوان حالت خاصی از خطوط تولید شناخته شده است، مدل‌های مختلف و متنوعی از یک محصول پایه مونتاژ می‌شود. در این مقاله به بررسی این نوع خطوط با هدف کمینه‌سازی هزینه کارگر کمکی و بیکاری اپراتور پرداخته شده در شرایطی که پارامتر زمان‌های پردازش به‌صورت تصادفی در نظر گرفته شده است. برای مدل‌سازی تصادفی از رویکرد محدودیت-شانس بهره برده شده است. برای اعتبارسنجی مدل نیز، در ابتدا با نرم‌افزار گمز حل شده و در نهایت با توجه به NP-hard بودن مسأله، الگوریتم جست‌وجوی هارمونی بهبود یافته<sup>۲</sup> (IHS) برای حل مدل استفاده شده است. در انتها نیز با توجه به نتایج به‌دست آمده، می‌توان نتیجه گرفت که جواب‌های به‌دست آمده از الگوریتم IHS در مقایسه با جواب دقیق حاصل از نرم‌افزار گمز از خطای محاسباتی کمی برخوردار است و جواب‌های قابل قبولی را ارائه می‌دهد و این نشان‌دهنده کارایی خوب الگوریتم می‌باشد.

**کلمات کلیدی:** تعیین توالی خطوط مونتاژ، زمان پردازش تصادفی، محدودیت شانس، الگوریتم جست‌وجوی هارمونی بهبود یافته.

\* عهده دار مکاتبات

آدرس الکترونیکی: p.fattahi@alzahra.ac.ir

<sup>1</sup> Mixed Model Assembly Line

<sup>2</sup> Improved Harmony Search

## ۱ مقدمه

تعیین توالی<sup>۳</sup> و زمان‌بندی<sup>۴</sup> نوعی فرآیند تصمیم‌گیری است که نقشی حیاتی در صنایع تولیدی و خدماتی ایفا می‌کند. این فرآیند تصمیم‌گیری نقش بسیار مهمی در کاهش هزینه‌ها، افزایش بهره‌وری، افزایش رضایت مشتری و به‌طور کلی افزایش مزیت رقابتی شرکت‌ها دارد، به گونه‌ای که در محیط رقابتی کنونی تعیین توالی و زمان‌بندی مؤثر تبدیل به یک ضرورت برای بقا در بازار و دنیای تجارت شده است. در ابتدا تولیدکنندگان تنها قادر به تولید محصولات در یک مدل<sup>۵</sup> و به صورت انبوه بودند که با توجه به نیازهای مشتری امروزه استفاده از این نوع خط چندان به‌صرفه نخواهد بود، از این‌رو گرایش تولیدکنندگان به تولید محصولات متنوع برای تطابق با نیازهای مشتریان و در نتیجه استفاده از خطوط مونتاژ مدل ترکیبی<sup>۶</sup> (MMAL) بیش‌تر شده است. دلایل عمده استفاده از این نوع خط را می‌توان این‌گونه ذکر کرد: انعطاف‌پذیری زیاد، نرخ استفاده از قطعات و توانایی پاسخگویی سریع به تقاضاهای متنوع مشتریان بدون نیاز به نگهداری موجودی انبوه از محصولات [۱].

درواقع خط مونتاژ مدل ترکیبی، نوعی خط تولید است که مدل‌های گوناگونی از محصولات را روی آن مونتاژ می‌گردد. این نوع خط مونتاژ، با تولید متنوع و با واکنش سریع نسبت به تغییرات ناگهانی تقاضا، توان پاسخ‌گویی به نیازهای متغیر مشتریان را دارد. برای حل مسایل خط مونتاژ مدل ترکیبی، ابتدا باید به مسایلی همچون تعیین زمان سیکل، تعداد و توالی ایستگاه‌ها و به‌طور کلی بالانس خط<sup>۷</sup> مونتاژ پرداخت و سپس توالی تولید مدل‌ها در خط مونتاژ را بررسی نمود. در اینجا ما فرض می‌کنیم بالانس خط مونتاژ انجام شده و مساله موردبررسی، تعیین توالی تولید محصولات است. از سویی دیگر، در مدل‌های سنتی زمان‌بندی فرض می‌شود که مقادیر تمام پارامترهای مربوط به کارها و ماشین‌ها ثابت بوده و از قبل معلوم هستند؛ اما بسیاری از محیط‌های تولیدی در دنیای واقعی با عدم اطمینان و حالت‌های تصادفی مواجه هستند. در اغلب مسایل تعیین توالی خطوط مونتاژ موجود نیز فرض بر این قرار گرفته است که پارامترهای هدف و یا محدودیت‌های مساله دقیق و قطعی هستند؛ باین‌حال، در واقعیت همیشه مقداری عدم قطعیت وجود دارد. در این پژوهش پارامتر زمان انجام عملیات<sup>۸</sup> به‌صورت غیرقطعی در نظر گرفته شده است. اگر عملیات مونتاژ با دستگاه‌های پیشرفته‌تر / روبات و یا اپراتورهای متخصص و بانگیزه انجام گیرد، زمان‌های انجام کار ممکن است ثابت باشند؛ باین‌حال در دنیای واقعی، به‌ویژه در خطوط مونتاژ دستی، وظایف ممکن است زمان‌های اجرای متفاوتی که به‌عنوان یک توزیع احتمال معرفی می‌شود، داشته باشند. تنوع در زمان‌های انجام کار ممکن است ناشی از خرابی‌های دستگاه، بی‌انگیزگی اپراتورها، فقدان آموزش، اپراتورهایی که واجد شرایط نیستند، کارهای پیچیده، محیط و غیره باشد [۲]. در نتیجه در این مقاله زمان‌های پردازش و انجام

<sup>3</sup> Sequencing

<sup>4</sup> Scheduling

<sup>5</sup> Single model

<sup>6</sup> Mixed Model Assembly Line

<sup>7</sup> Balancing

<sup>8</sup> Operation time

عملیات به صورت غیرقطعی و تصادفی در نظر گرفته شده است. روش‌های مختلف و متنوعی توسط محققین برای مقابله با پارامترهای تصادفی مطرح شده که در اینجا از رویکرد محدودیت-شانس<sup>۹</sup> استفاده شده است.

برنامه‌ریزی با قیود احتمالی یا همان محدودیت شانس، نوعی روش بهینه‌سازی تصادفی است که اولین بار توسط چارلز و همکاران [۳]، معرفی گشت. این روش برای حل مسایل بهینه‌سازی با پارامترهای تصادفی در محدودیت‌ها و گاهی نیز در تابع هدف، مناسب است. در این روش، تضمین می‌شود که محدودیت‌ها با یک احتمال مشخص شده یا سطح اطمینانی که از یک توزیع جمععی/چگالی احتمال شناخته شده برای متغیرهای غیرقطعی استفاده می‌کند، برآورد شود. برنامه‌ریزی محدودیت شانس برای مدل‌هایی که لازم است تصمیمات (بهینه) قبل از تحقق اثرات تصادفی گرفته شود، قابل اجراست [۴].

همان‌طور که ذکر شد، خطوط مونتاژ مدل ترکیبی یکی از ملزومات بازار رقابتی امروزه است. محققین زیادی از دهه‌ها قبل بر روی این نوع از خطوط، تحقیقاتی انجام داده‌اند که برخی به صورت تک هدفه و برخی دیگر به صورت چندهدفه بوده‌اند که در ادامه به تعدادی از این تحقیقات اشاره شده است، هم‌چنین در انتها تعدادی از تحقیقات که به صورت غیرقطعی هستند نیز آورده شده است.

میلتنبرگ [۵] مساله تعیین توالی خطوط مدل ترکیبی را با برنامه‌ریزی عدد صحیح غیرخطی با هدف کمینه‌سازی انحراف کل میزان تولید واقعی از میزان تولید موردنظر فرموله کرد. تحت این فرض که تمام مدل‌ها نیاز به تعداد و ترکیبی مشابه از قطعات دارند، میلتنبرگ اشاره کرد که به حداقل رساندن تنوع در میزان تولید محصولات پایانی (مساله سطح-محصول) با حداقل رساندن تنوع در نرخ استفاده از قطعات (مساله سطح-قطعه) به دست می‌آید. با توجه به پیچیدگی مساله دو هیورستیک برای حل پیشنهاد شد و آزمایش‌های محاسباتی محدودی برای مقایسه عملکرد نسبی آن‌ها انجام شد. میلتنبرگ و همکاران [۶] نیز یک الگوریتم برنامه‌ریزی پویا برای حل مساله ارایه شده توسط میلتنبرگ [۵]، که برای بسیاری از مسایل تولید با تقاضای زیاد و تعداد محصولات کم عملکرد مناسبی داشت، ارایه دادند. بارد و همکاران [۷] تعدادی از مدل‌های کمینه‌سازی طول خط و زمان عملکرد<sup>۱۰</sup> را در هر دو ایستگاه‌های باز و بسته در خطوط مونتاژ متحرک مدل ترکیبی توسعه دادند. آن‌ها دریافتند که خط مونتاژ با ایستگاه باز می‌تواند مشارکت بیش تری در هدف کمینه‌سازی طول خط و زمان عملکرد داشته باشد. سارکر و پان [۸] مساله تعیین توالی خط مونتاژ مدل ترکیبی را برای دو حالت ایستگاه باز و بسته با تابع هدف کمینه نمودن مجموع هزینه‌های ناشی از بیکاری و اضافه‌بار کاری بررسی نمودند. دو مدل برنامه‌ریزی عدد صحیح مختلط برای این دو حالت پیشنهاد شده است. این دو مدل، پارامترهای خط از جمله طول ایستگاه و زمان سیکل یا همان فاصله زمانی ثابت بین ورود مدل‌های مختلف به خط را به صورت بهینه به دست می‌آورند. نتایج نشان می‌دهد که حداقل مجموع هزینه‌ها در یک سیستم با ایستگاه باز، کم‌تر از حداقل مجموع هزینه‌ها در یک سیستم با ایستگاه بسته است. کیم و همکاران [۹]، نیز از الگوریتم ژنتیک برای حل مساله تعیین توالی خطوط مونتاژ مدل ترکیبی با ایستگاه‌های ترکیبی که در آن ایستگاه‌های باز و بسته درهم آمیخته و زمان راه‌اندازی وابسته به توالی است، بهره بردند. تابع هدف در مساله آن‌ها کمینه کردن

<sup>9</sup> Chance-constraint

<sup>10</sup> throughput time

کل طول یک خط است. عملکرد الگوریتم پیشنهادی با الگوریتم‌های هیوریستیک و روش شاخه و کران مقایسه شد. نتایج نشان داد که الگوریتم ژنتیک زمان محاسباتی کم‌تری دارد و جواب بسیار نزدیک به جواب بهینه است. ژیاو و اوهنو [۱۰]، مساله تعیین توالی خط مونتاژ مدل ترکیبی را هنگامی که توقف نقاله جایز است مورد بررسی قرار دادند. آن‌ها تحقیقات خود را در بررسی کارخانه خودروسازی توپوتا قرار دادند. آن‌ها در این مقاله، مساله تعیین توالی با هدف کمینه ساختن مجموع زمان توقف نوار نقاله را فرمول‌بندی نمودند. مشخصات متعددی از قبیل پیچیدگی محاسباتی مساله تعیین توالی، شرایط لازم و کافی برای توقف نوار نقاله، حد بالا و پایین تابع هدف و شرایط کافی برای بهینه بودن یک راه‌حل، تشریح شده است.

برای حل مساله نیز یک الگوریتم ابتکاری ارائه نمودند. بویسن و همکاران [۱۱]، به مساله نوسان نرخ تولید و چگونگی ارتباط آن با خطوط مونتاژ مدل ترکیبی در دنیای واقعی پرداختند. برای تسهیل تأمین مواد مورد نیاز در خط اصلی مونتاژ بر اساس نگرش JIT، مساله تعیین توالی را به صورت "زمان‌بندی مسطح" مورد بررسی قرار دادند که هدف هموار نمودن نرخ مصرف قطعات ناشی از توالی موجود در خط مونتاژ است. فتاحی و صالحی [۱۲] مساله تعیین توالی را با فاصله زمانی متغیر بین ورود محصولات در خط مونتاژ در نظر گرفتند. مساله مورد نظر دو مساله بهینه‌سازی را شامل می‌شد (مساله تعیین توالی و مساله فاصله زمانی بین ورود محصولات). آن‌ها الگوریتم متاهوریستیک ترکیبی بر اساس روش انجماد تدریجی و یک روش هیوریستیک را توسعه دادند. در نهایت نیز آزمایش‌های عددی برای بررسی عملکرد و کارایی الگوریتم پیشنهادی ارائه گشت که نتایج حاکی از عملکرد خوب الگوریتم تحت فرضیات ارائه شده بود. در ادامه به تعدادی از تحقیقاتی که به صورت چندهدفه انجام شده، اشاره شده است.

هیون و همکاران [۱۳] الگوریتم ژنتیک را برای حل مساله تعیین توالی خطوط مونتاژ مدل ترکیبی در حالت چندهدفه پیشنهاد دادند. مساله پیشنهادی آن‌ها شامل سه هدف است: (۱) کمینه‌سازی اضافه‌بار کاری، (۲) ثابت نگه‌داشتن نرخ مصرف قطعات و (۳) کمینه‌سازی هزینه راه‌اندازی کل. آن‌ها پس از مدل‌سازی مساله با اهداف پیشنهادی به صورت چندهدفه، یک الگوریتم ژنتیک برای یافتن راه‌حل بهینه پارتو یا نزدیک به پارتو برای این مساله طراحی کردند. یک مکانیسم ارزیابی و انتخاب جدید در الگوریتم ژنتیک، بنام لایه تورفتگی مکعبی پارتو<sup>۱۱</sup> ارائه شد. در نهایت عملکرد الگوریتم پیشنهادی با سه الگوریتم ژنتیک موجود مقایسه شد و مشاهده گشت الگوریتم پیشنهادی عملکرد بهتری بخصوص در مسایل با ابعاد بزرگ و مسایلی با نوسان زیاد در هزینه راه‌اندازی دارد. پانام بالان و همکارانش [۱۴] به بررسی الگوریتم ژنتیک در MMAL و مقایسه هیوریستیک پیشنهادی خود با الگوریتم ژنتیک پرداختند.

آن‌ها مساله تعیین توالی خطوط مونتاژ را به صورت چندهدفه و با اهداف: (۱) کمینه نمودن میزان اضافه‌بار کاری با ثابت نگه‌داشتن نرخ مصرف قطعات، (۲) کمینه نمودن تغییرپذیری در مصرف قطعات، (۳) کمینه نمودن مجموع هزینه‌های راه‌اندازی مورد بررسی قرار دادند.

در این تحقیق آن‌ها عملکرد مکانیسم انتخاب، فضای لایه تورفتگی پارتو و مقدار تابع هدف برازندگی اسکالر را با در نظر گرفتن هر سه تابع هدف مذکور مقایسه نمودند. در نهایت نتایج حاکی از عملکرد بهتر الگوریتم ژنتیک با مکانیسم لایه

<sup>11</sup> Pareto stratum-niche cubicle

تورفتگی پارتو نسبت به سایر مکانیسم‌های انتخابی است. بولات و همکاران [۱۵] الگوریتم‌هایی برای تعیین توالی با زمان واقعی کارها در خطوط مونتاژ مدل ترکیبی ارائه دادند. آن‌ها اهداف کمینه‌سازی هزینه راه‌اندازی کل و هزینه اضافه‌بار کاری کل را به‌طور هم‌زمان در نظر گرفتند. سپس یک فرمول‌بندی کلی از مساله تعیین توالی ارائه دادند. یک روش شاخه و کران و دو الگوریتم هیوریستیک برای تعیین توالی که اهداف موجود را کمینه می‌کند، نیز گسترش دادند. بارد و همکارانش [۱۶]، مدلی با دو هدف را برای حل مساله تعیین توالی در خط مونتاژ مدل ترکیبی ارائه دادند که قادر به بررسی توازن میان طول خط مونتاژ و نرخ استفاده از قطعات می‌باشد. مدل حاصل یک مدل غیرخطی عدد صحیح ترکیبی است که هم با روش هیوریستیک و هم با روش شاخه و کران حل شده است. نتایج حاصل شامل تعداد ایستگاه‌های موجود در خط، تعداد مدل‌ها و تعداد کل قطعات لازم برای مونتاژ است. زمان محاسباتی در هیوریستیک منطقی بوده ولیکن در روش شاخه و کران به‌طور قابل‌ملاحظه‌ای افزایش داشته است. مک‌مولن [۱۷]، یک الگوریتم هیوریستیک برای تعیین توالی خطوط مونتاژ مدل ترکیبی محصولات در سیستم JIT، زمانی که راه‌اندازی لازم است، ارائه داد. وی از الگوریتم جستجوی ممنوع<sup>۱۲</sup> (TS) برای تعیین توالی با اهداف کمینه‌سازی نرخ استفاده از قطعات و حداقل کردن راه‌اندازی بهره برد. مک‌مولن و فریتزر [۱۸]، مدل استفاده‌شده توسط مک‌مولن [۱۷] را در نظر گرفتند. برای حل نیز، هیوریستیک مبتنی بر الگوریتم انجماد تدریجی را که به‌طور هم‌زمان، راه‌اندازی و نرخ استفاده از قطعات را در تعیین توالی خطوط مونتاژ مدل ترکیبی در سیستم JIT بهینه می‌کند، ارائه دادند.

مسائل زیادی با روش پیشنهادی حل شد و مقادیر تابع هدف با جواب به‌دست آمده با الگوریتم جستجوی ممنوع مقایسه گشت. نتایج حاکی از برتری روش انجماد تدریجی نسبت به جستجوی ممنوع بود. منصوری [۱۹]، یک الگوریتم ژنتیک چندهدفه برای حل مساله تعیین توالی خط مونتاژ مدل ترکیبی تحت سیستم JIT ارائه نمود. هدف وی بهینه‌سازی نوسان‌های نرخ تولید و تعداد راه‌اندازی‌ها به‌طور هم‌زمان بود که این دو هدف ارتباط معکوس دارند؛ بنابراین بهینه‌سازی هم‌زمان هر دو تابع هدف مشکل بوده و از این‌رو مساله NP\_hard است. الگوریتم ژنتیک چندهدفه<sup>۱۳</sup> (MOGA) سعی در یافتن راه‌حل بهینه پارتو محلی و یا مرز غیر مؤثر محلی (مرزی که راه‌حل‌های موجود در آن منجر به کمینه ساختن هم‌زمان نوسان نرخ‌های تولید و تعداد راه‌اندازی‌ها می‌شود) دارد. عملکرد الگوریتم پیشنهادی با سه الگوریتم جستجوی دیگر با طرح شمارش کلی مقایسه شده است. هم‌چنین در مقایسه با سه الگوریتم ابتکاری دیگر از نظر کیفیت جواب‌ها در مدت‌زمان معقولی از پردازش بهتر عمل نموده است. توکلی مقدم و رحیمی واحد [۲۰]، مدل ریاضی جدید از خطوط مونتاژ مدل ترکیبی، برای یافتن بهترین توالی مدل‌های تولیدی در سیستم JIT ارائه دادند. اهداف آن‌ها حداقل کردن کل اضافه‌بار کاری، مجموع تغییرپذیری نرخ تولید و مجموع هزینه‌های آماده‌سازی بود. با توجه به NP\_hard بودن مساله از الگوریتم ممتیک برای حل استفاده کردند. برای ارزیابی عملکرد الگوریتم پیشنهادی، مسایل متعددی طراحی شد که نتایج حاکی از برتری این الگوریتم در کیفیت جواب و زمان محاسبات است. ربانی و رحیمی واحد [۲۱]، یک روش برنامه‌ریزی هدف فازی برای حل مساله

<sup>12</sup> Tabu Search

<sup>13</sup> Multi-Objective Genetic Algorithm

خطوط مونتاژ مدل ترکیبی چندهدفه در سیستم تولید JIT ارایه نمودند. آن‌ها سه هدف را به‌طور هم‌زمان در نظر گرفتند: حداقل کردن کل اضافه‌بار کاری، مجموع تغییرپذیری نرخ تولید و مجموع هزینه‌های آماده‌سازی. از آنجا که اهداف موجود متضاد هستند، آن‌ها روش برنامه‌ریزی هدف فازی را پیشنهاد نمودند. این روش بر اساس درجه مطلوبیت تصمیم‌گیرنده و تلرانس مدنظر روی مقادیر هدف ساخته شده است. برای نشان دادن رفتار مدل پیشنهادی مثال‌هایی حل شده و نتایج محاسباتی گزارش شده است. در ادامه نمونه‌هایی از تحقیقات غیرقطعی در این زمینه آمده است.

اوزکان و همکاران [۲]، مساله بالانس و تعیین توالی خطوط مونتاژ مدل ترکیبی U شکل را با زمان انجام کار<sup>۱۴</sup> احتمالی در نظر گرفتند. آن‌ها برای حل مساله از الگوریتم ژنتیک استفاده نمودند. برای ارزیابی عملکرد الگوریتم پیشنهادی، مطالعه‌ای بر روی هر دو نسخه قطعی و غیرقطعی زمان انجام کار انجام شد و نتایج محاسباتی حاکی از برتری الگوریتم پیشنهادی بود. همچنین از رویکرد فاصله اطمینان برای حل پارامتر تصادفی با توزیع نرمال بهره بردند.

دانگ و همکاران [۲۲]، مساله بالانس خط و مساله تعیین توالی مدل را به‌طور مشترک در خطوط U-شکل در نظر گرفتند و مدل برنامه‌ریزی احتمالی ۱-۰ را برای آن ارایه دادند. در این مدل زمان انجام کارها، متغیرهای احتمالی در نظر گرفته شده‌اند که دارای توزیع نرمال هستند و اهداف در این مساله کمینه ساختن زمان اضافه‌بار کاری<sup>۱۵</sup> مورد انتظار برای ترکیب داده شده از زمان چرخه و تعداد ایستگاه‌های کاری است. برای حل مساله الگوریتم مبتنی بر انجماد تدریجی توسعه داده شد که برای کمینه ساختن انحراف مطلق از حجم کار<sup>۱۶</sup> استفاده شده است.

اما تأکید اصلی این پژوهش، بررسی مساله توالی خطوط مونتاژ مدل ترکیبی در شرایط عدم قطعیت می‌باشد. با بررسی مطالعات انجام شده مشاهده شد، زمان انجام عملیات در بسیاری از مطالعات خطوط مونتاژ به‌صورت قطعی در نظر گرفته شده است، در صورتی که به علت وجود عوامل غیرقطعی فراوانی که در شرایط واقعی وجود دارد، مانند سطح توانایی اپراتورها، بی‌انگیزگی، پیچیدگی کارها، خرابی ماشین، تأخیر در تحویل مواد اولیه و غیره، برخی فرضیات مدل‌های قطعی ممکن است مصداق عینی نداشته باشند. محققین برای مقابله با این عدم قطعیت‌ها از برخی مدل‌های غیرقطعی مانند برنامه‌ریزی فازی، احتمالی و غیره استفاده کرده‌اند. رویکرد برنامه‌ریزی تصادفی که از توزیع‌های احتمالی برای داده‌های غیرقطعی استفاده می‌کند، در سال‌های اخیر توجه زیادی را به خود جلب کرده است. هدف از این مطالعه، ارایه یک الگوریتم فرا ابتکاری برای زمان‌بندی خطوط مونتاژ مدل ترکیبی در شرایط عدم قطعیت می‌باشد.

همان‌طور که ذکر شد در این مقاله مساله تعیین توالی خطوط مونتاژ مدل ترکیبی با هدف مجموع کمینه‌سازی هزینه کارگر کمکی و هزینه بیکاری اپراتور موردنظر است؛ درحالی که زمان‌های پردازش به‌صورت غیرقطعی و تصادفی می‌باشد. از رویکرد محدودیت شانس برای مدل‌سازی تصادفی و از الگوریتم فرا ابتکاری جست‌وجوی هارمونی بهبودیافته برای حل استفاده شده است. در ادامه، در بخش ۲ شرح مساله و مدل‌سازی ریاضی در دو حالت قطعی و تصادفی آمده است. در بخش

<sup>14</sup> Task time

<sup>15</sup> work overload time

<sup>16</sup> Workloads

۳ الگوریتم فرا ابتکاری توضیح داده شده و همین طور بخش ۴ شامل نتایج عددی می باشد. در انتها نیز نتیجه گیری و منابع ذکر شده است.

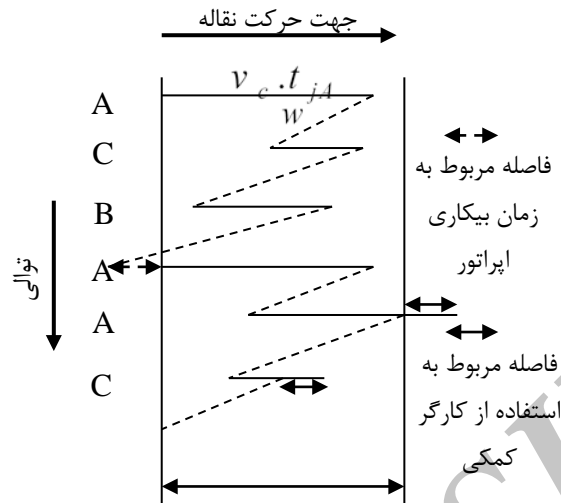
## ۲ شرح مساله

درواقع یک خط مونتاژ مدل ترکیبی (MMAL) شامل مجموعه ای از ایستگاه های کاری است که به صورت سریالی پشت سرهم قرار گرفته اند و نوار نقاله ای که با سرعت ثابت حرکت می کند [۱۲]، هم چنین می تواند مدل های متنوعی از محصولات مختلف را در طول یک شیفت و یا یک روز کاری مونتاژ کند. مدل های مختلف به طور متوالی بر روی نوار نقاله با طول زمان چرخه مشخص برای مونتاژ محصولات نیمه تمام، قرار می گیرند. در اینجا فرض بر این قرار گرفته که نوار نقاله با سرعت ثابت و از چپ به راست حرکت کرده و محصولات نیمه تمام را از طریق ایستگاه های کاری برای تکمیل تدریجی عملیات مونتاژ برای تولید محصول نهایی حرکت می دهد و هم چنین فرض شده که ایستگاه ها همه از نوع بسته هستند ایستگاه نوع بسته دارای مرز است و اپراتورها مجاز به عبور از مرزهای کاری خود نیستند. بر اساس این فرضیات نقطه شروع هر ایستگاه صفر و نقطه پایان  $L_j$  می باشد. ایستگاه های کاری قبلاً به طور مناسبی بالانس شده اند و زمان های پردازش نیز پارامتری تصادفی با توزیع نرمال در نظر گرفته شده است. فاصله زمانی بین ورود محصول ها ثابت می باشد و زمان های حرکت اپراتور در نظر گرفته نشده است. اپراتور به محض اتمام کار به ابتدای ایستگاه بازگشته و منتظر محصول بعدی می ماند، سرعت راه رفتن اپراتور نیز برابر سرعت نقاله است و هم چنین از الگوی زودترین زمان شروع استفاده شده است.

در اینجا از استراتژی MPS نیز استفاده شده است. استراتژی حداقل مجموعه قطعات<sup>۱۷</sup> به طور گسترده در خطوط مونتاژ مدل ترکیبی مورد پذیرش قرار گرفته است. تعداد محصول هایی که در یک سیکل تولید باید تعیین توالی شود، توسط برداری به نام حداقل مجموعه قطعات مشخص می گردد. اگر تقاضای مدل  $i$ ام برابر  $d_i$  باشد و  $i = 1, \dots, m$  و در مجموع  $m$  مدل داشته باشیم، در این صورت بردار حداقل مجموعه قطعات به صورت  $(\frac{D_1}{h}, \frac{D_2}{h}, \dots, \frac{D_M}{h}) = (d_1, d_2, \dots, d_M)$  است.  $h$  بزرگ ترین تقاضای مدل  $m$  است. به عنوان مثال، اگر تقاضاهای مدل های ترکیبی ۳۰۰ واحد از مدل A، ۵۰۰ واحد از مدل B و ۲۰۰ واحد از مدل C باشد، تعیین توالی برای قطعات کاری همه این ۱۰۰۰ مدل بسیار دشوار خواهد بود؛ بنابراین برای حل این مشکل ۳ واحد از محصول A، ۵ واحد از محصول B و ۲ واحد از محصول C در ابتدا زمان بندی خواهد شد. ادامه یک چنین توالی در حداقل مجموعه قطعات در ۱۰۰ تکرار تقاضای کل برای مدل های مختلف در همه دوره های کاری را ارضا خواهد کرد.

شکل ۱ که در ابتدا در مقاله هیون و همکاران [۱۳]، ذکر شد، زمان های بیکاری و استفاده از کارگر کمکی را در خطوط مونتاژ با ایستگاه های بسته نشان می دهد. در این ایستگاه سه نوع محصول (A, B, C) مونتاژ می شوند و سرعت نقاله ثابت و برابر  $V_c$  در نظر گرفته شده و  $t_{mj}$  زمان پردازش محصول  $m$  در ایستگاه  $j$  می باشد [۲۳].

<sup>17</sup> Minimum part set (MPS)



شکل ۱. عملیات‌های انجام شده در یک ایستگاه برای مسأله مورد بررسی

در جدول ۱ نیز نمادها و پارامترهای مورد استفاده در مسأله مورد بررسی ذکر شده است.

جدول ۱. معرفی متغیرها و پارامترها

اندیس‌ها	تعاریف
$i$	توالی محصولات، $i = \{1, 2, \dots, I\}$
$j$	ایستگاه، $j \in \{1, 2, \dots, J\}$
$m$	مدل، $m \in \{1, 2, \dots, M\}$
پارامترها	تعاریف
$J$	تعداد کل ایستگاه‌ها.
$M$	تعداد مدل‌ها.
$\gamma$	فاصله زمانی بین ورود محصول توالی $i-1$ ام و $i$ ام (این فاصله زمانی برای تمامی ایستگاه‌ها برابر می‌باشد).
$I$	تعداد کل توالی‌ها، $I = \sum_{m=1}^M d_m$
$d_m$	تقاضا برای مدل $m$ در بردار حداقل مجموعه قطعات.
$V_c$	سرعت نقاله، (فوت / دقیقه).
$L_j$	طول ایستگاه $j$ ام، (فوت).
$t_{mj}$	زمان لازم برای انجام عملیات بر روی محصول مدل $m$ ام در ایستگاه $j$ ام.
$\mu_{mj}$	مقدار میانگین پارامتر زمان پردازش $t_{mj}$ محصول مدل $m$ ام در ایستگاه $j$ ام.
$\sigma_{mj}^2$	واریانس پارامتر تصادفی زمان پردازش $t_{mj}$ محصول مدل $m$ ام در ایستگاه $j$ ام.
$C_U$	هزینه زمان لازم برای تکمیل عملیات ناتمام، (دلار / دقیقه).
$C_{ID}$	هزینه زمان بیکاری، (دلار / دقیقه).



متغیرها	تعاریف
$x_{im}$	متغیری باینری و برابر ۱ است اگر محصول توالی $i$ ام از نوع مدل $m$ باشد در غیر این صورت برابر ۰ است.
$Z_{ij}$	موقعیت مکانی شروع به کار بر روی محصول توالی $i$ ام در ایستگاه $j$ ام
$U_{ij}$	زمان لازم برای تکمیل عملیات ناتمام ایستگاه $j$ ام بر روی محصول توالی $i$ ام.
$ID_{ij}$	زمان بیکاری اپراتور ایستگاه $j$ ام برای توالی $i$ ام.

## ۲-۱ مدل ریاضی به صورت قطعی

در این مقاله تابع هدف مورد بررسی کمیته سازی مجموع هزینه کارگر کمکی و هزینه بیکاری اپراتور می باشد. با توجه به این که طبق فرضیات توقف نقاله مجاز نیست، زمانی که کارگر عادی مشغول در خط موفق نشود کار را تا انتهای خط به اتمام برساند و در واقع خط دچار اضافه بار کاری شود از کارگر کمکی بهره می گیرد که هدف اول در رابطه با این موضوع است. هیون و همکاران [۱۳] برای این منظور از معادله زیر استفاده کردند:

$$\min \sum_{j=1}^J C_U \left( \sum_{i=1}^I U_{ij} + z_{(I+1)j} / v_c \right)$$

هدف دوم سعی در کمیته سازی بیکاری اتفاق افتاده برای هر اپراتور در هر ایستگاه کاری دارد که به صورت زیر فرموله شده است:

$$\min \sum_{j=1}^J \sum_{i=1}^I (C_{ID} \cdot ID_{ij})$$

مدل کلی و محدودیت های مساله نیز به شرح زیر می باشد:

$$\min Z = \sum_{j=1}^J C_U \left( \sum_{i=1}^I U_{ij} + z_{(I+1)j} / v_c \right) + \sum_{j=1}^J \sum_{i=1}^I (C_{ID} \cdot ID_{ij})$$

st :

$$\sum_{m=1}^M x_{im} = 1, \quad i = 1, \dots, I, \quad (1)$$

$$\sum_{i=1}^I x_{im} = d_m, \quad m = 1, \dots, M, \quad (2)$$

$$z_{\backslash, j+1} = \sum_{l=1}^j L_l, \quad j = 1, \dots, J-1, \quad (3)$$

$$z_{i+1, j} = z_{ij} + v_c \cdot \left( \sum_{m=1}^M x_{im} t_{mj} - U_{ij} - \gamma + ID_{i+1, j} \right), \quad i = 1, \dots, I-1, \quad j = 1, \dots, J, \quad (4)$$

$$U_{ij} \geq (z_{ij} + v_c \cdot \sum_{m=1}^M x_{im} t_{mj} - \sum_{l=1}^j L_l) / v_c, \quad i = 1, \dots, I-1, \quad j = 1, \dots, J, \quad (5)$$

$$U_{ij} \geq [z_{ij} + v_c \cdot \sum_{m=1}^M x_{im} t_{mj} - (\sum_{l=1}^{j-1} L_l + v_c \cdot \gamma)] / v_c, \quad j = 1, \dots, J, \quad (6)$$

$$ID_{ij} \geq \left( \sum_{l=1}^{j-1} L_l - (z_{i-1,j} + v_c \sum_{m=1}^M x_{i-1,m} t_{mj} - v_c U_{i-1,j} - v_c \cdot \gamma) \right) / v_c \quad ID_{ij} \geq 0, \quad i = 2, \dots, I-1, \quad j = 1, \dots, J, \quad (7)$$

$$x_{im} \in \{0, 1\}; \quad (8)$$

$$U_{ij} \geq 0, z_{1,j} = 0, z_{ij} \geq 0; \quad i = 1, \dots, I, \quad j = 1, \dots, J, \quad m = 1, \dots, M, \quad (9)$$

محدودیت (۱) اطمینان می‌دهد که تنها یک محصول به هر موقعیت مکانی در یک توالی تخصیص داده شده است. تقاضا برای هر مدل  $m$  برابر  $d_m$  است. برای تأمین مقدار تقاضا، تعداد  $d_m$  تا از کل توالی‌ها باید از مدل  $m$  باشد؛ لذا برای تأمین تقاضای موجود برای هر مدل محدودیت (۲) در نظر گرفته شده است. محدودیت (۳) بیانگر این است که کار بر اولین محصول از هر چرخه باید از مرز سمت چپ ایستگاه شروع شود. محدودیت (۴) موقعیت شروع کارگر در هر ایستگاه  $j$  برای محصول  $(i+1)$  در توالی را نشان می‌دهد. محدودیت (۵) نشان‌دهنده زمان اضافه‌بار کاری تأمین محصول از ایستگاه  $j$  در یک توالی است. محدودیت (۶) زمان اضافه‌بار کاری برای آخرین محصول  $I$  در ایستگاه و در یک توالی است. محدودیت (۷) نشان‌دهنده زمان بیکاری برای محصول  $i$  از ایستگاه  $j$  است.  $Z_{1,j}$  نیز بیانگر این است که ایستگاه از نوع بسته می‌باشد.

## ۲-۲ استفاده از رویکرد محدودیت-شانس در مدل‌سازی تصادفی مساله

با توجه به فرضیات و در مساله پیشنهادی زمان‌های پردازش،  $t_{mj}$ ، پارامتری تصادفی و مستقل در نظر گرفته شده که از توزیع نرمال پیروی می‌کند، با میانگین  $\mu_{mj}$  و واریانس  $\sigma_{mj}^2$ ، به گونه‌ای که  $(t_{mj} \sim N(\mu_{mj}, \sigma_{mj}^2))$  با سطح اطمینان  $\alpha$  ( $0 \leq \alpha \leq 1$ ) لازم به ذکر است که مقادیر  $\alpha$  بستگی به نظر تصمیم‌گیرنده<sup>۱۸</sup> (DM) دارد.

همان‌طور که در قسمت‌های قبل ذکر شد، برای برخورد با پارامتر تصادفی مساله و مدل کردن مساله در شرایط عدم قطعیت از رویکرد محدودیت-شانس استفاده شده است. قابل ذکر است که در روش محدودیت-شانس مقدار  $\alpha = 1$  معادل حالت قطعی است. در ادامه بحث، با توجه به مدل پیشنهادی و استفاده از روش محدودیت-شانس، محدودیت‌های (۴)–(۷) پس از اعمال تغییرات به شکل زیر در مدل گسترش داده شده است.

$$z_{i+1,j} = z_{ij} + v_c \cdot \left( \sum_{m=1}^M x_{im} \cdot \mu_{mj} + Z_{1-\alpha} \sqrt{\sum_{m=1}^M \sigma_{mj}^2 x_{im}^2} - U_{ij} - \gamma + ID_{i+1,j} \right), \quad i = 1, \dots, I-1, \quad j = 1, \dots, J, \quad (10)$$

$$U_{ij} \geq \frac{\left( z_{ij} + v_c \cdot \left( \sum_{m=1}^M x_{im} \cdot \mu_{mj} + Z_{1-\alpha} \sqrt{\sum_{m=1}^M \sigma_{mj}^2 x_{im}^2} \right) - \sum_{l=1}^j L_l \right)}{v_c}, \quad i = 1, \dots, I-1, \quad j = 1, \dots, J, \quad (11)$$

$$U_{ij} \geq \left( z_{ij} + v_c \cdot \left( \sum_{m=1}^M x_{im} \cdot \mu_{mj} + Z_{1-\alpha} \sqrt{\sum_{m=1}^M \sigma_{mj}^2 x_{im}^2} \right) - \left( \sum_{l=1}^{j-1} L_l + v_c \cdot \gamma \right) \right) / v_c, \quad j = 1, \dots, J, \quad (12)$$

<sup>18</sup> Decision maker

$$ID_{ij} \geq \left( \sum_{l=1}^{j-1} L_l - \left( z_{i-1,j} + v_c \cdot \left( \sum_{m=1}^M x_{i-1,m} \cdot \mu_{mj} + Z_{1-\alpha} \sqrt{\sum_{m=1}^M \sigma_{mj}^2 \cdot x_{i-1,m}^2} \right) - v_c U_{i-1,j} - v_c \cdot \gamma \right) \right) / v_c \quad (13)$$

$, i = 2, \dots, I-1 \quad , j = 1, \dots, J$

با توجه به رابطه زیر تعداد کل توالی‌های ممکن برای یک مساله مشخص از رابطه زیر به دست می‌آید [۱۳]:

$$\text{تعداد توالی تولید} = \frac{\left( \sum_{m=1}^M d_m \right)!}{\prod_{m=1}^M (d_m)!} \quad (14)$$

با توجه به این رابطه، به راحتی می‌توان دریافت که با افزایش بعد مساله، زمان مورد نیاز برای به دست آوردن توالی‌های تولید بهینه به صورت نمایی افزایش خواهد یافت. از طرف دیگر فرض مبتنی بر غیرقطعی بودن زمان پردازش نیز بر پیچیدگی مساله می‌افزاید؛ بنابراین در مسایل با ابعاد بالا نمی‌توان از روش‌های بهینه‌سازی کلاسیک استفاده کرد.

### ۳ الگوریتم جست‌وجوی هارمونی بهبود یافته<sup>۱۹</sup>

الگوریتم جست‌وجوی هارمونی<sup>۲۰</sup> (HS)، یکی از ساده‌ترین و جدیدترین روش‌های فرا ابتکاری است که در فرایند جست‌وجوی جواب شدنی بهینه در مسایل بهینه‌سازی، از فرایند نواختن هم‌زمان گروه ارکستر موزیک الهام گرفته شده است. به عبارت دیگر میان پیدا کردن یک حل بهینه در مساله پیچیده و فرایند اجرای موزیک تشابه وجود دارد. این روش حل را ابتدا گیم و همکاران [۲۴] در سال ۲۰۰۱ ارائه کردند. مطابق با منطق این روش فرا ابتکاری، تلاش برای به دست آمدن هماهنگی در یک فرایند موزیک، مشابه پیدا کردن یک حل بهینه در مسایل بهینه‌سازی است. HS مانند الگوریتم ژنتیک، جزء الگوریتم‌های بهبوددهنده است. به عبارت دیگر با نسلی از بردارهای حل شروع و برای ایجاد نسل‌های جدید از فرایند انتخاب استفاده می‌شود؛ اما برخلاف الگوریتم ژنتیک (که در آن از دو کروموزوم برای تولید کروموزوم یا بردار حل جدید استفاده می‌شود) در این روش از همه بردارهای حل موجود در حافظه برای تولید<sup>۲۱</sup> حل جدید استفاده می‌شود. از مزایای این الگوریتم، همگرایی سریع آن به دلیل ساختار مناسب آن است و از معایب آن افتادن در بهینه محلی به دلیل جستجو با تنوع کم در تکرارهای پایانی الگوریتم است، که راه‌حل برای رفع آن استفاده از تکنیک فاز شروع دوباره و تغییر در قواعد الگوریتم به خصوص در تکرارهای پایانی است. در این راستا برای بهبود الگوریتم از روش پیشنهادی مهدوی و همکاران [۲۴] استفاده شده است.

این الگوریتم از پنج گام تشکیل شده است:

(۱) مقداردهی اولیه مساله بهینه‌سازی و پارامترهای اولیه

<sup>19</sup> Improved Harmony Search (IHS)

<sup>20</sup> Harmony search algorithm

<sup>21</sup> Improve

۲) مقداردهی حافظه هارمونی

۳) ایجاد یک هارمونی جدید بهبود یافته

۴) به‌روز کردن حافظه هارمونی

۵) معیار توقف

در ادامه توضیح گام‌های این الگوریتم آمده است:

### ۳-۱ مقداردهی اولیه مساله بهینه‌سازی و پارامترهای اولیه

در این مرحله، مساله بهینه‌سازی به صورت زیر تعریف می‌شود:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x_i \in X_i = 1, 2, \dots, N \end{aligned} \quad (15)$$

$f(x)$  همان تابع هدف و  $x$  مجموعه‌ای از متغیرهای تصمیم‌گیری است.  $N$  تعداد متغیرهای تصمیم است.  $x_i$  مجموعه‌ای از طیف ممکن از ارزش‌ها برای هر متغیر تصمیم‌گیری است که در آن  $x_i^L \leq x_i \leq x_i^U$  است که  $x_i^L$  و  $x_i^U$  مرزهای بالا و پایین برای هر متغیر تصمیم‌گیری است. فرض کنید  $k$  هارمونی توسط  $N$  نوازنده ساخته شده است، در واقع فرض شده است اندازه حافظه نوازندگان  $^{22}$  (HMS)، برابر  $k$  هارمونی است. پس با توجه به رابطه (۱۶) یک ماتریس با  $k$  سطر (تعداد هارمونی‌هایی که نوازندگان به خاطر می‌سپارند) و  $N+1$  ستون که در آن  $N$  تعداد نوازنده (تعداد متغیرهای تأثیرگذار در مساله) و یک ستون برای ارزش آن هارمونی ( $f(x)$ ) در نظر گرفته شده است (در مسایل تک هدفه). به ماتریس حاصل حافظه هارمونی  $^{23}$  (HM) می‌گویند.

$$HMS \text{ (Harmony Memory Size)} = k \quad (16)$$

در ادامه، پارامترهای کلیدی الگوریتم، اندازه حافظه هارمونی HMS، نرخ در نظر گرفتن حافظه هارمونی  $^{24}$  HMCR، نرخ تنظیم زیروبمی  $^{25}$  PAR، ماکزیمم تعداد بهبود  $^{26}$  NI و پهنای باند  $^{27}$  BW نیز مقداردهی می‌شوند [۲۵]. در شکل ۲ ماتریس حافظه هارمونی آمده است.

### ۳-۲ مقداردهی حافظه هارمونی

حافظه هارمونی مکانی است که بردارهای جواب (مجموعه متغیرهای تصمیم)، در آن ذخیره شده‌اند. HM شبیه به تعداد جمعیت در دیگر الگوریتم‌های تکاملی است. ماتریس HM در ابتدا به عنوان راه‌حل‌های تولیدی تصادفی مشخص شده

<sup>22</sup> Harmony Memory Size

<sup>23</sup> Harmony Memory

<sup>24</sup> Harmony memory consideration rate

<sup>25</sup> Pitch adjusting rate

<sup>26</sup> Maximum improvisation

<sup>27</sup> Band width

به عنوان HMS در محدوده حد بالا و پایین، پر می شود و هم چنین با مقدار تابع متناظر با هر بردار تصادفی مقدار  $(f(x))$  محاسبه می شود.

$$\begin{bmatrix} x_1^1 & x_2^1 & \dots & x_n^1 & f(x^1) \\ x_1^2 & x_2^2 & \dots & x_n^2 & f(x^2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \dots & x_n^{HMS} & f(x^{HMS}) \end{bmatrix}$$

شکل ۲. ماتریس حافظه هامونی

### ۳-۳ ایجاد یک هارمونی جدید بهبود یافته

هر هارمونی جدید که تولید می شود، می تواند از تغییر در نت های هارمونی های قبلی ایجاد شود. به این معنا که نوازنده نت های خود را کمی تغییر می دهد. بردار هارمونی جدید  $X' = (x_1', x_2', \dots, x_N')$  در الگوریتم بر اساس سه قاعده تولید شود: (۱) در نظر گرفتن نت های موجود در حافظه، (۲) انتخاب تصادفی و (۳) تنظیم زیروبمی.

منظور از در نظر گرفتن نت های موجود در حافظه، استفاده از نت های قبلی نواخته شده موجود در حافظه می باشد، به طور نمونه در شکل زیر از نت های نوازنده اول استفاده می شود. به عبارت دیگر، مقدار متغیر تصمیم  $x_1'$  برای بردار جدید از هر یک از مقادیر در محدوده حافظه هارمونی مشخص شده  $(x_1^1 - x_1^{HMS})$ ، انتخاب می شود. مقادیر سایر متغیرها،  $(x_2', x_3', \dots, x_N')$  نیز به همین ترتیب انتخاب می شوند. HMCR که نرخ انتخاب از مقادیر قبلی ذخیره شده در حافظه HM است بین ۰ و ۱ متغیر می باشد و  $(1-HMCR)$  نرخ انتخاب تصادفی یک مقدار از محدوده مقادیر ممکن است.

$$x_i' = \begin{cases} x_i' \in (x_i^1, x_i^2, \dots, x_i^{HMS}) \rightarrow \text{with probability of HMCR} \\ x_i' \in X_i \rightarrow \text{with probability of } (1-HMCR) \end{cases} \quad (17)$$

به عنوان مثال، HMCR با مقدار ۰/۸۵ نشان می دهد که الگوریتم HS، مقدار متغیر تصمیم را از مقادیر ذخیره شده قبلی در HM با احتمال ۸۵٪ انتخاب خواهد کرد، یا از تمامی محدوده ممکن با احتمال ۱۰-۸۵٪ انتخاب می کند. سپس هر جزء به دست آمده با در نظر گرفتن حافظه، بررسی می شود که آیا نیاز به تنظیم<sup>۲۸</sup> دارد یا خیر. این عملیات توسط پارامتر PAR انجام می شود که به شرح زیر است:

$$x_i' = \begin{cases} \text{Yes} & \text{with probability PAR, so } \rightarrow x_i' = x_i' \pm \text{rand}() * bw, \\ \text{No} & \text{with probability } (1-PAR) \rightarrow \text{(the rate of doing nothing)} \end{cases} \quad (18)$$

که در اینجا،  $BW$  پهنای باند تصادفی است و همین‌طور منظور از  $\text{rand}()$  مقداری تصادفی بین ۰ و ۱ است. لازم به ذکر است این عملیات برای هر متغیر از بردار هارمونی جدید انجام می‌شود.

برای بهبود عملکرد الگوریتم جست‌وجوی هارمونی و جلوگیری از افتادن در دام بهینه محلی، الگوریتم جست‌وجوی هارمونی بهبود داده‌شده<sup>۲۹</sup> (IHS) در اینجا مورد استفاده قرار گرفته است. در این الگوریتم روش جدیدی در ایجاد بردار حل جدید ارایه شده است که دقت و همگرایی آن نسبت به روش جست‌وجوی هارمونی ساده، بهبود می‌یابد [۲۵]. نقش پارامتر  $PAR$  در افزایش تنوع در شروع جست‌وجوی الگوریتم در فضای حل و نقش  $BW$  در جست‌وجوی محلی الگوریتم برای افزایش نرخ همگرایی تأثیرگذار است. IHS مشابه HS است با این تفاوت که مقادیر پارامترهای  $PAR$  و  $BW$  به صورت پویا در تکرار جداگانه مطابق روابط ذکر شده در زیر حاصل می‌گردند. مقادیر  $PAR$  و  $BW$  در طی هر نسل به شرح زیر تغییر می‌کنند:

$$PAR(gn) = PAR_{\min} + \frac{PAR_{\max} - PAR_{\min}}{NI} \times gn \quad (19)$$

که  $PAR(g)$  نرخ تنظیم نسل فعلی است،  $PAR_{\min}$  کم‌ترین نرخ تنظیم و  $PAR_{\max}$  بیش‌ترین نرخ تنظیم است.  $g$  تعداد نسل فعلی است و  $NI$  تعداد نواختن‌ها<sup>۳۰</sup> (تکرارها) است.

$$BW(gn) = BW_{\max} \exp\left(\frac{\ln\left(\frac{BW_{\min}}{BW_{\max}}\right)}{NI} \times gn\right) \quad (20)$$

که  $BW$  نرخ پهنای باند نسل فعلی است،  $BW_{\min}$  کم‌ترین نرخ پهنای باند و  $BW_{\max}$  بیش‌ترین نرخ پهنای باند است.

در اکثر مسایل نشان داده‌شده است که عملکرد HIS به مراتب بهتر از عملکرد HS استاندارد است [۲۵]. از معایب روش HS استفاده از مقادیر ثابت برای پارامترهای  $PAR$  و  $BW$  است که تنظیم مناسب این پارامترها مشکل است. هم‌چنین از معایب دیگر HS این است که تعداد تکرارهایی که الگوریتم نیاز دارد تا حل بهینه را پیدا کند، مناسب نیست. اگر مقدار  $PAR$  کوچک و مقدار  $BW$  بزرگ باشد، عملکرد الگوریتم تضعیف شده و برای بهبود آن باید تعداد تکرارها را افزایش دهیم تا به حل بهینه دست پیدا کنیم. هرچه مقدار  $BW$  در تکرارهای ابتدایی بزرگ‌تر باشد، موجب می‌شود که الگوریتم، تنوع جست‌وجو در کل فضای حل را افزایش دهد و در تکرارهای بعدی برای جست‌وجوی محلی، مقدار کم‌تر آن مناسب است؛ بنابراین مقادیر بزرگ  $PAR$  و مقادیر کوچک  $BW$  در تکرارهای پایانی منجر به رسیدن به فضای بهینه و همگرایی به جواب بهینه می‌گردد.

<sup>29</sup> Improved Harmony Search

<sup>30</sup> Improvisation

### ۳-۴ به روز کردن حافظه هارمونی

اگر بردار هارمونی جدید از بدترین بردار هارمونی موجود در حافظه بهتر باشد بر اساس ارزیابی مقدار تابع هدف، بردار هارمونی جدید در HM گنجانده می شود و بدترین بردار هماهنگی موجود از HM حذف می شود.

### ۳-۵ معیار توقف

آخرین گام در الگوریتم جست و جوی هارمونی، این است که چک کنیم که آیا الگوریتم توانسته جواب قابل قبولی که بتواند انتظارات تصمیم گیرنده را برآورده کند، بیابد. معیارهای توقف متفاوتی در ادبیات استفاده شده است: (۱) توقف بعد از تعداد مشخصی تکرار، (۲) توقف پس از آنکه بهبودی در تابع هدف حاصل نشد و (۳) توقف پس از رسیدن به مقدار مشخصی از تابع هدف. در اینجا معیار توقف، رسیدن به تعداد مشخصی تکرار است. لازم به ذکر است که تعداد تکرار بستگی به نوع مساله و نظر تصمیم گیرنده دارد. وقتی معیار توقف ارضا شود، الگوریتم متوقف می شود؛ به عبارت دیگر، مراحل ۳ و ۴ تا توقف الگوریتم، تکرار می شوند.

### ۴ نتایج عددی

در ادامه، مثال‌هایی ارائه شده که در ابتدا توسط نرم افزار گمز حل شده است. با توجه به ماهیت NP\_hard بودن مساله و پیچیدگی بالای مدل که با افزایش تعداد مدل‌ها، توالی‌ها و ایستگاه‌ها نیز به طور فزاینده‌ای افزایش می یابد و با توجه به اینکه حل مدل توسط نرم افزارهای حل دقیق مانند گمز مستلزم صرف زمان و برخورداری از حافظه و قدرت پردازش بالا خواهد بود؛ لذا برای غلبه بر این مشکلات از الگوریتم فرا ابتکاری جست و جوی هارمونی استفاده شده است و عملکرد این دو مورد مقایسه قرار گرفته است.

جهت کد نویسی مساله و حل توسط الگوریتم فرا ابتکاری از نرم افزار MATLAB R2014a در یک لپ تاپ با پردازنده Intel® Core™ i5-4200U CPU @ 1.60GHz 2.30 GHz و حافظه 4.00GB (2.98GB usable) و سیستم عامل Microsoft Windows 8.1 استفاده شده است.

### ۴-۱ نحوه تولید مسایل آزمایشی

مسایل آزمایشی برای ارزیابی و اثربخشی الگوریتم پیشنهادی در دو اندازه کوچک و بزرگ و به صورت تصادفی تولید شده‌اند. این مسایل در تعداد مدل‌ها، تقاضا برای هر مدل و هم چنین تعداد ایستگاه‌ها باهم تفاوت دارند. زمان‌های پردازش همان‌طور که در فصول قبل توضیح داده شد از توزیع نرمال پیروی کرده و همین‌طور با توجه به منبع [۴]، می توان واریانس را نسبتی از میانگین دانست، به طوری که  $\sigma_{mj}^2 = \lambda \mu_{mj}$ . در این رابطه  $\lambda$  عدد ثابت مثبتی است که در اینجا ۵٪ در نظر گرفته شده است و هم چنین  $\mu_{mj}$  و  $\sigma_{mj}^2$  به ترتیب میانگین و واریانس پارامتر تصادفی زمان پردازش مدل  $m$  از ایستگاه  $j$  هستند. در نظر گرفتن یک چنین ارتباطی بین میانگین و واریانس در توزیع نرمال غیر معمول نبوده و [۲۶] و [۲۷] برای زمان‌های پردازش کارها با توزیع نرمال، نیز از این نسبت استفاده کرده‌اند. لازم به ذکر است مقدار  $\alpha$  توسط تصمیم گیرنده

مشخص می‌شود که در اینجا با بررسی مرور ادبیات و مطالعات انجام گرفته، مقدار  $0.975$  ( $Z_{\alpha} = 1/96$ ) قرار داده شده است.

در الگوریتم جست‌وجوی هارمونی، هفت پارامتر مهم اندازه حافظه (HMS)، تعداد بردارهای حل در حافظه (HMCR)، مینیمم نرخ تنظیم گام (PAR)، ماکزیمم نرخ تنظیم گام (PAR)، مینیمم پهنای باند (BW)، ماکزیمم پهنای باند (BW) و حداکثر تکرار (NI) وجود دارد. برای تنظیم پارامترهای الگوریتم که موجب افزایش کارایی الگوریتم می‌شود، برای عواملی که در بالا معرفی شد سطوحی را در نظر می‌گیریم و به ازای هر سطح از پارامترها برنامه را ۵ بار در متلب اجرا کرده و میانگین اعداد هر سطر را به‌عنوان ستون مبنا در نرم‌افزار مینی‌تب جهت تجزیه و تحلیل و محاسبات قرار می‌دهیم. عوامل و سطوح را به شکل زیر در جدول ۲ در نظر می‌گیریم:

جدول ۲. بازه مورد استفاده پارامترهای الگوریتم HS

HMS	HMCR	PAR <sub>min</sub>	PAR <sub>max</sub>	BW <sub>min</sub>	BW <sub>max</sub>	NI
۱۰	۰/۸۵	۰/۱	۰/۷	۰/۵	۳	۱۰۰
۲۰	۰/۹۳	۰/۲	۰/۹۹	۰/۴۵	۰/۹	۵۰۰
۳۰	۰/۹۹	۰/۳	۲	۰/۳	۴	۱۰۰۰

بعد از اجرای برنامه در نرم‌افزار متلب با استفاده از داده‌های به‌دست آمده تنظیم پارامتر انجام شده است. در ادامه با استفاده از نتایج به‌دست آمده و نرم‌افزار مینی‌تب با استفاده از روش فاکتوریل به تنظیم پارامتر الگوریتم پرداخته شده که نتایج ذیل در جدول ۳ برای هر عامل حاصل شده است.

جدول ۳. مقادیر مورد استفاده پارامترهای الگوریتم HS

HMS	HMCR	PAR <sub>min</sub>	PAR <sub>max</sub>	BW <sub>min</sub>	BW <sub>max</sub>	NI
۲۰	۰/۸۵	۰/۲	۰/۹۹	۰/۴۵	۰/۹	۱۰۰

#### ۴-۲ ساینز کوچک

برای تمام این مسایل، هزینه‌های زمان بیکاری، کارگر کمکی و انحراف از نرخ تولید برابر ۱ دلار، سرعت نقاله برابر ۱ متر بر دقیقه و نیز برابر ۱ در نظر گرفته شده است. میانگین پارامتر تصادفی زمان پردازش با توجه به منبع [۴] به صورت تصادفی با استفاده از توزیع یکنواخت در فاصله [۱۵, ۵] تولید شده و واریانس نیز با استفاده از رابطه  $\sigma_{mj}^2 = \lambda \mu_{mj}$ ، به‌دست آمده که  $\lambda$  نیز ۵٪ در نظر گرفته شده است. در ادامه طول ایستگاه و زمان‌های پردازش در جدول ۴ و سایر مشخصات مساله شامل تعداد ایستگاه، طول ایستگاه و میانگین و واریانس پارامتر تصادفی زمان پردازش در جدول ۵ آمده است.

برای بررسی امکان‌پذیری مدل لازم است که ابتدا با روشی دقیق حل شود که از نرم‌افزار گمز<sup>۳۱</sup> بدین منظور در این پژوهش استفاده شده است و همین‌طور برای ارزیابی کارایی الگوریتم فرا ابتکاری، نتایج حاصل از آن با نتیجه حاصل از

<sup>31</sup> GAMS software



نرم افزار گمز مقایسه شده است. از آنجا که زمان نقش مهمی در بررسی عملکرد الگوریتم دارد، یکی از معیارهای مقایسه CPU time در نظر گرفته شده که در واقع زمان حل مساله را نشان می دهد. جواب هایی حاصل از الگوریتم IHS و گمز نیز توسط معادله (۲۱) مورد مقایسه قرار گرفته اند [۱]. نتایج حل مسایل در ابعاد کوچک در جدول ۶ آمده است. لازم به ذکر است که به دلیل پیچیده بودن مساله، نرم افزار GAMS قادر به حل سایزهای بزرگ تر نمی باشد و مقایسات بین روش دقیق و الگوریتم فرا ابتکاری IHS در ابعاد کوچک مورد بررسی قرار گرفته است.

$$GAP = \frac{best\ cost\ (IHS) - best\ cost\ (GAMS)}{best\ cost\ (GAMS)} \quad (21)$$

جدول ۴. طول ایستگاه و زمان های پردازش

J	مدل			L
	۱	۲	۳	
۱	N(۶/۳، ۳/۱۵)	N(۵/۹، ۲/۹۵)	N(۱۴/۶، ۷/۳)	۲
۲	N(۱۴/۱، ۷/۰۵)	N(۷/۸، ۳/۹)	N(۶/۸، ۳/۴)	۴
۳	N(۱۱/۳، ۵/۶۵)	N(۱۰/۵، ۵/۲۵)	N(۱۴/۷، ۷/۳۵)	۲

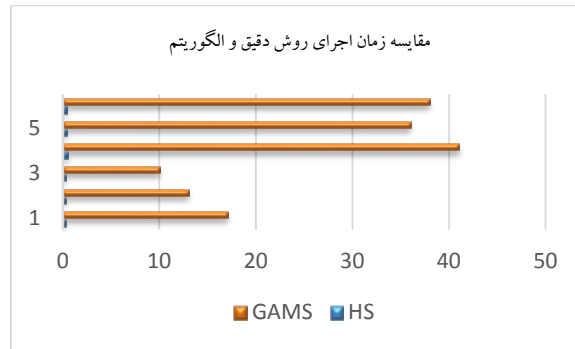
جدول ۵. مشخصات نمونه های مساله مورد استفاده در سایز کوچک

مساله	I	MPS	تعداد راه حل ممکن	$\gamma$
۱	۶	(۲، ۲، ۲)	۹۰	۶/۶
۲	۶	(۱، ۲، ۳)	۶۰	۶/۵
۳	۶	(۱، ۱، ۴)	۳۰	۶/۶
۴	۸	(۳، ۲، ۳)	۵۶۰	۶/۲
۵	۸	(۴، ۲، ۲)	۴۲۰	۶/۲
۶	۸	(۲، ۳، ۳)	۵۶۰	۶/۲

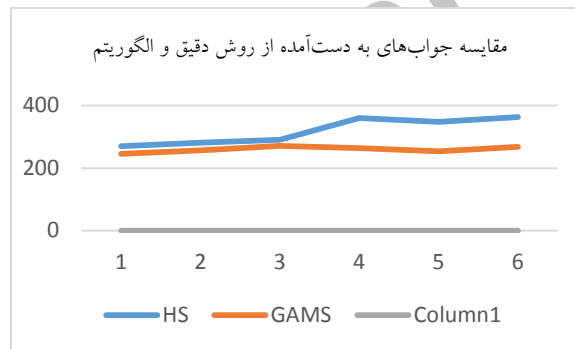
جدول ۶. نتایج حل مسایل در ابعاد کوچک

مساله	RUN time (s)		Objective function		GAP
	GAMS	HIS	GAMS	HIS	
۱	۱۷	۰/۲۷	۲۴۶	۲۷۰	۰/۰۹۸
۲	۱۳	۰/۲۶	۲۵۷	۲۸۱	۰/۰۹۳
۳	۱۰	۰/۲۶	۲۷۱	۲۹۱	۰/۰۷۴
۴	۴۱	۰/۴۴	۲۶۴	۳۶۰	۰/۳۶
۵	۳۶	۰/۳۹	۲۵۴	۳۴۸	۰/۳۷
۶	۳۸	۰/۴	۲۶۸	۳۶۳	۰/۳۵

در توضیح مقایسه دو روش همان‌طور که در جدول ۶ و شکل ۴ مشاهده می‌شود می‌توان نتیجه گرفت که جواب‌های به‌دست‌آمده از الگوریتم IHS در مقایسه با جواب دقیق حاصل از نرم‌افزار گمز از خطای محاسباتی کمی برخوردار است و جواب‌های قابل‌قبولی را نتیجه می‌دهد و این نشان‌دهنده کارایی خوب الگوریتم می‌باشد. با توجه به جدول ۶ و همان‌طور که از نتایج مشخص است و شکل ۳، الگوریتم جست‌وجوی هارمونی بهبود یافته در ابعاد کوچک از لحاظ زمان اجرا از عملکرد بسیار بهتری برخوردار است.



شکل ۳. مقایسه زمان اجرای روش دقیق و الگوریتم



شکل ۴. مقایسه جواب‌های روش دقیق و الگوریتم

### ۳-۴ سایز بزرگ

هزینه زمان بیکاری بر اساس توزیع یکنواخت در بازه [۵, ۱۰] و هزینه کارگر کمکی بر اساس توزیع یکنواخت در بازه [۸, ۱۲] تولیدشده است. سرعت نقاله نیز یک متر بر دقیقه می‌باشد. هم‌چنین میانگین پارامتر تصادفی زمان پردازش با توجه به منبع [۴] به صورت تصادفی با استفاده از توزیع یکنواخت در فاصله [۵, ۱۵] تولیدشده و  $\lambda$  نیز ۵٪ در نظر گرفته شده است. داده‌های ورودی این مساله در جدول ۷ آمده است.

جدول ۷. مشخصات نمونه‌های مساله مورد استفاده در سایز بزرگ

مساله	I	MPS	تعداد راه حل ممکن	$\gamma$
۱	۲۰	(۴,۶,۳,۲,۲,۳)	$9/78 * 10^{12}$	۱۰
۲	۲۰	(۴,۴,۲,۳,۳,۴)	$2/44 * 10^{12}$	۱۰
۳	۲۰	(۵,۵,۴,۳,۲,۱)	$5/78 * 10^{12}$	۱۰
۴	۳۰	(۴,۴,۳,۵,۵,۳,۲,۴)	$18/51 * 10^{12}$	۱۰
۵	۳۰	(۵,۴,۴,۴,۳,۳,۲,۵)	$18/51 * 10^{12}$	۱۰
۶	۳۰	(۳,۵,۵,۲,۴,۳,۴,۴)	$18/51 * 10^{12}$	۱۰

جدول ۸. نتایج حل مسایل در ابعاد بزرگ

مساله	RUN time (s)		Objective function		GAP
	GAMS	HIS	GAMS	HIS	
۱	*	۳	*	۸۷۹۲	-----
۲	*	۳	*	۸۷۶۸	-----
۳	*	۲/۵	*	۸۶۵۳	-----
۴	*	۵	*	۱۳۰۶۶	-----
۵	*	۵	*	۱۳۱۵۵	-----
۶	*	۵	*	۱۳۲۹۴	-----

\* در این مسایل، روش حل دقیق گمز قادر به ارایه جواب نمی‌باشد.

جدول ۸ شامل نتایج حل مسایل در ابعاد بزرگ می‌باشد که نرم‌افزار گمز قادر به حل این گونه مسایل نبوده و تنها نتایج حاصل از حل توسط الگوریتم IHS آمده است.

## ۵ نتیجه گیری

در این پژوهش، مساله تعیین توالی خطوط مونتاژ مدل ترکیبی با تابع هدف کمینه‌سازی مجموع هزینه اضافه‌بار کاری و هزینه بیکاری اپراتور در شرایطی مورد مطالعه قرار گرفت که پارامتر زمان پردازش به صورت غیرقطعی و تصادفی در نظر گرفته شده است. لازم به ذکر است با توجه به بررسی مرور ادبیات، مساله مورد نظر، برای اولین بار مورد مطالعه قرار گرفته است. در ادامه برای مدل‌سازی تصادفی از رویکرد محدودیت-شانس استفاده شد و هم‌چنین الگوریتم جست‌وجوی هارمونی بهبودیافته برای حل مدل ارایه گشت. با توجه به نتایج محاسباتی مشخص شد، جواب‌های به دست آمده از الگوریتم IHS در مقایسه با جواب دقیق حاصل از نرم‌افزار گمز از خطای محاسباتی کمی برخوردار بوده و جواب‌های قابل قبولی را ارایه می‌دهد و این نشان‌دهنده کارایی خوب الگوریتم می‌باشد. با توجه به جدول ۶ و شکل ۳، الگوریتم جست‌وجوی هارمونی بهبودیافته در ابعاد کوچک از لحاظ زمان اجرا از عملکرد بسیار بهتری برخوردار است. در ابعاد بزرگ نیز، الگوریتم پیشنهادی جواب‌های قابل قبولی را در زمان معقولی ارایه داد.

استفاده از سایر روش‌های برنامه‌ریزی تصادفی، در نظر گرفتن پارامترهای کلیدی به صورت فازی و هم‌چنین استفاده از سایر الگوریتم‌های فرا ابتکاری و مقایسه عملکرد الگوریتم جست‌وجوی هارمونی بهبود یافته با دیگر الگوریتم‌های مطرح، برای تحقیقات آتی پیشنهاد می‌شود.

## منابع

- [1] Manavizadeh, N Tavakoli, L., Rabbani, M., & Jolai, F. (2013). A multi-objective mixed-model assembly line sequencing problem in order to minimize total costs in a Make-To-Order environment, considering order priority. *Journal of Manufacturing Systems*, 32(1), 124-137.
- [2] Özcan, U., Kellegöz, T., & Toklu, B. (2011). A genetic algorithm for the stochastic mixed-model U-line balancing and sequencing problem. *International Journal of Production Research*, 49(6), 1605-1626.
- [3] Charnes, A., Cooper, W. W., & Symonds, G. H. (1958). Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science*, 4(3), 235-263.
- [4] Elyasi, A., & Salmasi, N. (2013). Stochastic scheduling with minimizing the number of tardy jobs using chance constrained programming. *Mathematical and Computer Modelling*, 57(5), 1154-1164.
- [5] Miltenburg, J. (1989). Level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science*, 35(2), 192-207.
- [6] Miltenburg, J., Steiner, G., & Yeomans, S. (1990). A dynamic programming algorithm for scheduling mixed-model, just-in-time production systems. *Mathematical and Computer Modelling*, 13(3), 57-66.
- [7] Bard, J. F., DAR-ELJ, E. Z. E. Y., & SHTUB, A. (1992). An analytic framework for sequencing mixed model assembly lines. *The International Journal of Production Research*, 30(1), 35-48.
- [8] Haixu, S., & Bhaba, R. (1998). Designing a mixed-model assembly line to minimize the costs of idle and utility times. *Computers & industrial engineering*, 34(3), 609-628.
- [9] Kim, Y. K., Hyun, C. J., & Kim, Y. (1996). Sequencing in mixed model assembly lines: a genetic algorithm approach. *Computers & Operations Research*, 23(12), 1131-1145.
- [10] Xiaobo, Z., & Ohno, K. (2000). Properties of a sequencing problem for a mixed model assembly line with conveyor stoppages. *European Journal of Operational Research*, 124(3), 560-570.
- [11] Boysen, N., Flidner, M., & Scholl, A. (2009). The product rate variation problem and its relevance in real world mixed-model assembly lines. *European Journal of Operational Research*, 197(2), 818-824.
- [12] Fattahi, P., & Salehi, M. (2009). Sequencing the mixed-model assembly line to minimize the total utility and idle costs with variable launching interval. *The International Journal of Advanced Manufacturing Technology*, 45(9-10), 987-998.
- [13] Hyun, C. J., Kim, Y., & Kim, Y. K. (1998). A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines. *Computers & Operations Research*, 25(7), 675-690.
- [14] Ponnambalam, S. G., Aravindan, P., & Subba Rao, M. (2003). Genetic algorithms for sequencing problems in mixed model assembly lines. *Computers & Industrial Engineering*, 45(4), 669-690.
- [15] Bolat, A., Savsar, M., & Al-Fawzan, M. A. (1994). Algorithms for real-time scheduling of jobs on mixed model assembly lines. *Computers & operations research*, 21(5), 487-498.
- [16] Bard, J. F., Shtub, A., & Joshi, S. B. (1994). Sequencing mixed-model assembly lines to level parts usage and minimize line length. *The International Journal of Production Research*, 32(10), 2431-2454.
- [17] McMullen, P. R. (1998). JIT sequencing for mixed-model assembly lines with setups using tabu search. *Production Planning & Control*, 9(5), 504-510.
- [18] McMULLEN, P. R., & Frazier, G. V. (2000). A simulated annealing approach to mixed-model sequencing with multiple objectives on a just-in-time line. *Iie Transactions*, 32(8), 679-686.
- [19] Mansouri, S. A. (2005). A multi-objective genetic algorithm for mixed-model sequencing on JIT assembly lines. *European Journal of Operational Research*, 167(3), 696-716.
- [20] Tavakkoli-Moghaddam, R., & Rahimi-Vahed, A. R. (2006). A memetic algorithm for multi-criteria sequencing problem for a mixed-model assembly line in a JIT production system. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on* (pp. 2993-2998). IEEE.
- [21] Rabbani, M., Rahimi-Vahed, A., Javadi, B., & Tavakkoli-Moghaddam, R. (2007). A new approach for mixed-model assembly line sequencing. In *Operations Research Proceedings 2006* (pp. 169-174). Springer Berlin Heidelberg.

- [22] Dong, J., Zhang, L., Xiao, T., & Mao, H. (2014). Balancing and sequencing of stochastic mixed-model assembly U-lines to minimise the expectation of work overload time. *International Journal of Production Research*, 52(24), 7529-7548.
- [23] Tavakkoli-Moghaddam, R., & Rahimi-Vahed, A. R. (2006). Multi-criteria sequencing problem for a mixed-model assembly line in a JIT production system. *Applied Mathematics and Computation*, 181(2), 1471-1481.
- [24] Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *Simulation*, 76(2), 60-68.
- [25] Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied mathematics and computation*, 188(2), 1567-1579.
- [26] Sarin, S. C., Erel, E., & Steiner, G. (1991). Sequencing jobs on a single machine with a common due date and stochastic processing times. *European Journal of Operational Research*, 51(2), 188-198.
- [27] Cai, X., & Zhou, S. (1997). Scheduling stochastic jobs with asymmetric earliness and tardiness penalties. *Naval Research Logistics (NRL)*, 44(6), 531-557.

Archive of SID