

ارائه‌ی الگوریتم شبیه‌سازی تبرید پارتو برای نوع دوم مسئله بالانس خط مونتاژ چندهدفه با در نظر گرفتن زمان‌های آماده‌سازی وابسته به توالی

سید علیرضا سید علاقه‌بند (دانشجوی دکتری)

مژده ربانی* (استادیار)

گروه مدیریت صنعتی، واحد یزد، دانشگاه آزاد اسلامی، یزد، ایران

محمد میرابی (استادیار)

گروه مهندسی صنایع، دانشگاه میبد، میبد، ایران

حسن دهقان دهنوی (استادیار)

گروه مدیریت صنعتی، واحد یزد، دانشگاه آزاد اسلامی، یزد، ایران

مهندسی صنایع و مدیریت شریف، زمستان ۱۴۰۲ (۱۴)
دوری ۳۹، شماره ۲، صص. ۱۳-۳، (پژوهشی)

در حال حاضر، مسائل بالانس خط مونتاژ، به ویژه مسائل چندهدفه، به‌طور فزاینده‌ای مورد بررسی و مطالعه قرار گرفته‌اند. در پژوهش حاضر، به بهینه‌سازی هم‌زمان دو هدف کمیته‌سازی زمان سیکل و کمیته‌سازی مجموع زمان‌های آماده‌سازی در مسائل بالانس خط مونتاژ پرداخته شده است. هدف اول زمانی مورد استفاده قرار می‌گیرد که یک سازمان موجود، به دنبال تولید بیشتر محصولات در یک شیفت کاری است، به‌طوری‌که سازمان علاقه‌ای به تغییر تعداد ایستگاه‌های کاری ندارد. هدف دوم عموماً زمانی ارزش می‌یابد که طولانی شدن زمان‌های آماده‌سازی، باعث افزایش هزینه‌های تعمیرات و نگهداری ماشین‌آلات و همچنین خستگی بیشتر اپراتورها می‌شود. روش‌های قطعی با توجه به مفروضات مسئله کارآمد نیستند. بنابراین، الگوریتم شبیه‌سازی تبرید پارتو در این پژوهش مورد استفاده قرار گرفته است تا مجموعه جواب پارتو را تولید کند. نتایج محاسباتی، نشان‌دهنده کارایی قابل قبول الگوریتم پیشنهادی هستند.

alagheband@aut.ac.ir
moz.rabbani@iaui.ac.ir
mirabi@meybod.ac.ir
h.dehghan@iauyazd.ac.ir

واژگان کلیدی: الگوریتم شبیه‌سازی تبرید پارتو، بالانس خط مونتاژ، زمان‌بندی، زمان‌های آماده‌سازی وابسته به توالی.

۱. مقدمه

مسائل مرتبط با برنامه‌ریزی و بهبود عملکرد خطوط مونتاژ را مسائل بالانس خط مونتاژ^۱ می‌نامند. این دسته از مسائل طی سالیان اخیر از اهمیت خاصی برخوردار بوده و توجه بسیاری را به خود جلب کرده‌اند. هدف اصلی از طرح مسائل بالانس خط، اختصاص فعالیت‌ها به ایستگاه‌ها است به‌طوری‌که روابط تقدم و تأخر رعایت شده و معیارهای اندازه‌گیری عملکرد خط مونتاژ (بسته به نوع مسئله) بهینه گردند. بیابار در مرور خود بر مسائل بالانس خط، این مسائل را به دو دسته عمده تقسیم کرد:^[۱] مسائل بالانس خط مونتاژ ساده^۲ که فرضیات ساده‌ساز بسیاری را شامل می‌شوند و از این رو در محیط واقعی نمی‌توان با اطمینان بالایی به نتایج آنها استناد

* نویسنده مسئول

تاریخ: دریافت ۱۳۹۹/۸/۲۶، اصلاحیه ۱۴۰۱/۳/۱۶، پذیرش ۱۴۰۱/۵/۲۵

استناد به این مقاله:

سیدعلاقه‌بند، سید علیرضا، ربانی، مژده، میرابی، محمد، و دهقان دهنوی، حسن، ۱۴۰۲. ارائه‌ی الگوریتم شبیه‌سازی تبرید پارتو برای نوع دوم مسئله بالانس خط مونتاژ چندهدفه با در نظر گرفتن زمان‌های آماده‌سازی وابسته به توالی. مهندسی صنایع و مدیریت شریف، ۳۹(۲)، صص. ۱۴-۳. DOI:10.24200/J65.2022.55826.2127

هدف مسئله است.^[۴] سید علاقه‌بند و همکاران، نوع دوم مسائل بالانس خط عمومی را با در نظر گرفتن زمان‌های آماده‌سازی وابسته به توالی مورد بررسی قرار دادند. (GALBPS-II) آنها برای مسئله خود مدل ریاضی ارائه دادند و با توجه به Np-hard بودن مدل، از الگوریتم شبیه‌سازی تیرید برای حل آن بهره بردند.^[۴] این پژوهش به بررسی همزمان اهداف کمیته‌سازی زمان سیکل و کمیته‌سازی مجموع زمان‌های آماده‌سازی می‌پردازد. هدف اول زمانی مورد استفاده قرار می‌گیرد که یک سازمان موجود، به دنبال تولید بیشتر محصولات در یک شیفت کاری است، در حالی که سازمان علاقه‌ای به تغییر تعداد ایستگاه‌های کاری ندارد. این امر به مدیر سازمان کمک می‌کند تا بهره‌وری تولید خود را بهبود بخشد. هدف دوم، که بیشتر به آن به صورت بسیار محدود پرداخته شده است، عموماً زمانی ارزش می‌یابد که طولانی شدن زمان‌های آماده‌سازی، عموماً باعث افزایش هزینه‌های تعمیرات و نگهداری ماشین‌آلات و همچنین خستگی بیشتر اپراتورها می‌شود. این امر به این دلیل است که مدت زمان زیادی برای باز و بسته کردن قطعات دستگاه جهت شروع فعالیت جدید لازم است. هرچه فعالیت‌های آماده‌سازی دارای پیچیدگی بیشتر یا طولانی‌تر باشند، هزینه‌های نگهداری و تعمیرات دستگاه در بلندمدت افزایش خواهند یافت. از دیگر فواید کمیته‌سازی زمان‌های آماده‌سازی، می‌توان به کاهش هزینه‌ها، انتقال سریع، افزایش رقابت‌پذیری، تولید ناب، تحویل سریع‌تر و رضایت‌مندی بیشتر مشتریان اشاره کرد.^[۵]

ادامه این پژوهش به صورت زیر تدوین شده است:

بخش دوم به مرور پژوهش‌های پیشین موضوع بالانس خط و زمان‌های آماده‌سازی در این باب می‌پردازد؛ بخش سوم، مدل ریاضی مربوط به مسئله را تبیین شده است؛ در بخش چهارم، روش حل قطعی برای مدل ریاضی را تشریح شده است؛ در بخش پنجم به معرفی اجزای الگوریتم شبیه‌سازی شده تیرید پارتو پرداخته شده است؛ در بخش ششم نتایج محاسباتی جمع‌آوری شده است؛ بخش هفتم به جمع‌بندی پژوهش و ارائه مسیریایی برای تحقیقات آتی در این زمینه اختصاص دارد.

۲. پیشینه پژوهش

سلوین برای اولین بار، مسائل بالانس خط مونتاژ را به صورت نظری مطرح نمود^[۶] و تا به امروز، مطالعات بی‌شماری روی این موضوع صورت گرفته است. مطالعات مروری جامعی نیز روی مسائل بالانس خط و روش‌های حل آنها ارائه شده است.^[۷،۸] بوینس و همکاران، طبقه‌بندی بسیار جامعی برای مسائل بالانس خط انجام دادند.^[۹] آنها مسائل بالانس خط را از جنبه‌های ایستگاه‌های کاری، اهداف و گراف تقدم طبقه‌بندی نمودند که دارا بودن هر یک از این جنبه‌ها، مسئله بالانس خط عمومی را در پی خواهد داشت. در مطالعه‌های دیگر، بکر و شول به موارد متعدد عمومی‌سازی مسائل خطوط مونتاژ شامل خطوط U شکل و دوسویه،^۵ ایستگاه‌های موازی، انتخاب تجهیزات و زمان‌های تصادفی اشاره کردند.^[۲]

مسائل بالانس خط به صورت چندهدفه طی سال‌های اخیر از توجه بیشتری برخوردار شده‌اند. پژوهش‌های صورت گرفته در این زمینه اهداف متعددی را در نظر گرفته‌اند و به علت پیچیدگی فضای حل، روش‌های حل مبتنی بر الگوریتم‌های فراابتکاری ارائه داده‌اند. چن و همکاران، برای حل مسئله چندهدفه بالانس خط از الگوریتم ژنتیک ترکیبی استفاده کرده‌اند و برای ارزیابی کروموزوم‌ها از تابع جمع وزنی استفاده کردند.^[۱۰] این الگوریتم و حالات ارتقاء یافته آن همچنان در مسائل

بالانس خط مونتاژ چندهدفه به طور مکرر مورد استفاده قرار گرفته‌اند.^[۱۱] علاوه بر الگوریتم ژنتیک، الگوریتم‌های چندهدفه دیگری نیز در پژوهش‌های صورت گرفته یافت می‌شود. برای مثال می‌توان به الگوریتم کلونی زنبور عسل^۶ و یا الگوریتم تکامل تفاضلی^۷ اشاره کرد.^[۱۲،۱۳]

در بسیاری از مقالات، نتیجه طرح مسئله چندهدفه، دستیابی به مجموعه پارتو نبوده است. برای مثال در،^[۱۳] هدف کمیته‌سازی زمان سیکل به همراه دو هدف زمان بیکاری کمیته و هموارسازی بیشینه بررسی شده است که با استفاده از وزن‌دهی، عملاً مسئله به یک مسئله تک‌هدفه تبدیل شده است و مجموعه پارتو ارائه نشده است. این نکته به خصوص در مسائل چندهدفه با راه‌حل مبتنی بر الگوریتم SA بیشتر مشاهده می‌شود. برای مثال، بایکاسوگلو، الگوریتم چندهدفه SA را برای حل مسئله خط مونتاژ مستقیم و U شکل ارائه داد. وی بین دو هدف، کمیته‌سازی تعداد ایستگاه‌ها و بیشینه‌سازی همواری آنها، تقدم قرار داد و برای حل مسئله از برنامه‌ریزی آرمانی^۸ استفاده کرد.^[۱۴]

زمان‌های آماده‌سازی وابسته به توالی پیشین در مباحث زمان‌بندی تولید^۹ مورد بررسی قرار گرفته است.^[۱۵] آندرس و همکاران برای اولین بار، مسئله بالانس خط مونتاژ با زمان‌های آماده‌سازی را به طور جدی مورد بررسی قرار دادند. آنها برای مسئله جدید GALBPS-I مدل ریاضی ارائه دادند و با توجه به پیچیدگی مسئله، پس از ارائه روش‌های ابتکاری، الگوریتم فراابتکاری (GRASP)^{۱۰} استفاده کردند تا مسئله را در مدت زمان مناسب حل کنند.^[۲] مارتینو و پاسور، چند روش ابتکاری برای حل مسئله GALBPS-I مشابه ارائه کردند.^[۱۶] شول و همکاران، نمونه ویرایش شده‌ای از مسئله^[۲] را مورد بررسی قرار دادند و علاوه بر زمان آماده‌سازی بین فعالیت‌ها (فوروارد)، زمان آماده‌سازی فعالیت آخر یک ایستگاه و فعالیت اول همان ایستگاه را متفاوت در نظر گرفتند. بدین صورت که اگر دو فعالیت در ابتدا و انتهای ایستگاه کاری قرار گیرند، زمان آماده‌سازی آن با حالتی که دو فعالیت پشت سرهم قرار گیرند متفاوت خواهد بود. آنها برای این مسئله جدید تعریف شده، مدل ریاضی ارائه کرده و برای حل مسئله، الگوریتم فراابتکاری ارائه کردند.^[۱۷] اسماعیل بیگی و همکاران، مسئله^[۱۷] را با دو فرمول‌بندی جدید، یکی بر اساس ایستگاه کاری و دیگری بر اساس زمان‌بندی، توسعه دادند.^[۱۸] اخیراً دلیس و همکاران، زمان‌های آماده‌سازی را در خطوط U شکل دوطرفه مورد بررسی قرار دادند و برای حل مسئله طرح شده، الگوریتم کلونی مورچگان توسعه دادند.^[۱۹] هدف کمیته‌سازی مجموع زمان‌های آماده‌سازی در مسائل زمان‌بندی تولید پیشتر مطرح بوده است،^[۲۰] اما تاکنون در پژوهش‌های صورت گرفته مسائل بالانس خط مونتاژ مورد بررسی قرار نگرفته است. با توجه به اهمیت روزافزون کمیته‌سازی مجموع زمان‌های آماده‌سازی، پژوهش حاضر قصد دارد این هدف را با هدف سنتی مسائل بالانس خط و کمیته‌سازی زمان سیکل، به طور همزمان مورد بررسی قرار دهد.

۳. تعریف مسئله

همانطور که پیشتر اشاره شد، این مسئله، حالت خاصی از مسئله نوع دوم بالانس خط مونتاژ ساده است که در آن، زمان‌های آماده‌سازی وابسته به توالی برای هر زوج فعالیت در نظر گرفته می‌شود و هدف، کمیته‌سازی همزمان زمان سیکل کل خط و مجموع زمان‌های آماده‌سازی می‌باشد. شکل ۱، گراف تقدم مربوط به آزمون Jackson را نشان می‌دهد که در آن، زمان‌های اجرای هر فعالیت در گوشه سمت چپ آن‌ها قید شده‌اند. ماتریس زمان‌های آماده‌سازی مربوطه در جدول ۱ نشان

مسئله مورد نظر بدین صورت تعریف می شود.

$$\text{Min}(z) = ct \quad (1)$$

$$\text{Min}(z') = st \quad (2)$$

با توجه به محدودیت های زیر:

$$\sum_{j=1}^m \sum_{s=1}^{Nm_j} x_{ijs} = 1 \quad (\forall i) \quad (3)$$

$$\sum_{i=1}^N x_{ijs} \leq 1 \quad (\forall j, \forall s = 1 \dots Nm_j) \quad (4)$$

$$\sum_{i=1}^N x_{ij,s+1} \leq \sum_{i=1}^N x_{ijs} \quad (\forall j, \forall s = 1 \dots Nm_j - 1) \quad (5)$$

$$\sum_{j=1}^m \sum_{s=1}^{Nm_j} (NT_m \cdot (j+1) + s) \cdot x_{ijs} \leq \quad (6)$$

$$\sum_{j=1}^m \sum_{s=1}^{Nm_j} (NT_m \cdot (j+1) + s) \cdot x_{kjs} \quad (\forall (i, k) \in P) \quad (6)$$

$$\sum_{i=1}^N \sum_{s=1}^{Nm_j} t_i \cdot x_{ijs} + \sum_{i=1}^N \sum_{k=1 \& i \neq k}^N tsu_{ik} \cdot z_{ikj} \leq ct \quad (\forall j = 1 \dots m) \quad (7)$$

$$\sum_{j=1}^m \sum_{i=1}^N \sum_{k=1 \& i \neq k}^N tsu_{ik} \times z_{ikj} = st \quad (8)$$

$$x_{ijs} + x_{kjs+1} \leq 1 + z_{ikj} \quad (\forall j, \forall s = 1 \dots Nm_j, \forall (i, k) | (i \neq k) \wedge (k \notin PT_i)) \quad (9)$$

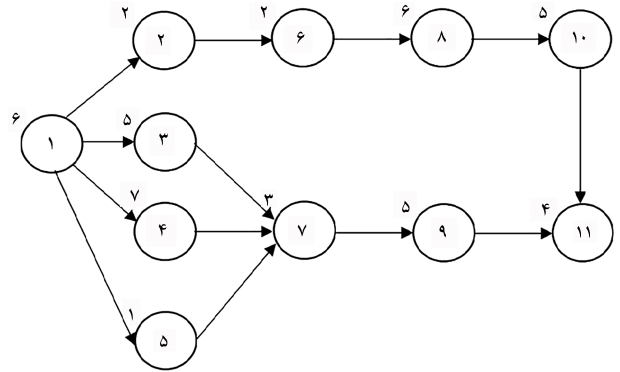
$$x_{ijs} - \sum_{\substack{k=1 \\ (i \neq k \& k \notin PT_i)}}^N x_{kjs+1} \leq w_{ij} \quad (\forall j, \forall s = 1 \dots Nm_j - 1) \quad (10)$$

$$w_{ij} + x_{kjs} \leq 1 + z_{ikj} \quad (\forall j, \forall (i, k) | (i \neq k) \wedge (i \notin PT_k)) \quad (11)$$

تابع هدف ۱ و ۲ زمان سیکل و مجموع زمان های آماده سازی را کمینه می کنند. محدودیت ۳ مبین این موضوع است که هر فعالیت فقط به یک مکان در فقط یک ایستگاه می تواند اختصاص یابد. با توجه به محدودیت ۴، در هر مکان از هر ایستگاه، بیش از یک فعالیت نمی تواند اختصاص یابد. محدودیت ۵ اطمینان حاصل می کند که فعالیت ها، مکان های هر ایستگاه را به صورت صعودی اشغال کنند. براساس محدودیت ۶، اختصاص فعالیت ها با توجه به گراف تقدم صورت می گیرد، هم از لحاظ تخصیص فعالیت ها به ایستگاه ها و هم از لحاظ ترتیب آنها در ایستگاه ها. محدودیت ۷ نشان می دهد که زمان پردازش هر ایستگاه از زمان سیکل خط که باید کمینه گردد کمتر می باشد. محدودیت ۸ به محاسبه مجموع زمان های آماده سازی کل خط می پردازد. با توجه به محدودیت ۹، متغیر z_{ikj} زمانی ۱ است که کار i به مکان s و کار k به مکان $s+1$ در ایستگاه j اختصاص می یابند. با توجه به محدودیت ۱۰، متغیر w_{ij} زمانی ۱ است که کار i به آخرین مکان ایستگاه j اختصاص یابد. بالاخره، با توجه به محدودیت ۱۱، متغیر z_{ikj} زمانی ۱ است که کار k به اولین مکان و کار i به آخرین مکان ایستگاه j اختصاص یابند.

داده شده اند که مقادیر آن به صورت تصادفی از بازه صفر و یک چهارم کمینه زمان اجرای فعالیت ها تولید شده اند. شکل ۲ دو جواب شدنی از آزمون Jackson با تعداد ایستگاه های مساوی با ۵ را نشان می دهد که در آن، ایستگاه های کاری با خطوط نقطه چین از هم جدا شده اند و مقادیر زمان سیکل و مجموع زمان آماده سازی در دو جهت متفاوت با یکدیگر حرکت کرده اند.

با استفاده از نمادهای استفاده شده در مقالات [۳۳] (بخش نمادها)، مدل ریاضی



شکل ۱. گراف تقدم آزمون Jackson.

جدول ۱. زمان های آماده سازی وابسته به توالی برای آزمون Jackson.

	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱
۱	۰	۰.۲۵	۰.۱۷	۰.۰۸	۰.۰۲۱	۰.۰۸	۰.۰۲۱	۰.۰۸	۰.۱۷	۰.۲۵	۰
۲	۰.۱۳	۰	۰.۱۲	۰.۰۹	۰.۰۳	۰.۰۸	۰.۱۲	۰.۰۹	۰.۱۲	۰	۰.۱۳
۳	۰.۱۸	۰.۲	۰	۰.۲۳	۰.۰۵	۰.۰۶	۰.۰۵	۰.۲۳	۰	۰.۲	۰.۱۸
۴	۰.۲۵	۰.۱	۰.۰۹	۰	۰.۰۹	۰.۱۷	۰	۰.۰۹	۰.۰۹	۰.۱	۰.۲۵
۵	۰.۲	۰.۰۵	۰.۱۹	۰.۰۱	۰.۱۴	۰	۰.۰۱	۰.۱۹	۰.۰۵	۰.۰۲	۰.۱۳
۶	۰.۰۷	۰.۱۹	۰.۲۲	۰.۱۲	۰.۰۷	۰	۰.۰۷	۰.۱۲	۰.۲۲	۰.۱۹	۰.۰۷
۷	۰.۱۸	۰.۱۵	۰.۰۸	۰.۰۲	۰.۱۴	۰.۰۱	۰.۱۴	۰.۰۲	۰.۰۸	۰.۱۵	۰.۱۸
۸	۰.۰۱	۰.۰۲	۰.۱۱	۰.۰۹	۰.۱۳	۰.۰۸	۰.۱۳	۰.۰۹	۰.۱۱	۰.۰۲	۰.۰۱
۹	۰.۱	۰.۰۱	۰.۰۲	۰.۱۱	۰.۱۳	۰.۰۲	۰.۱۱	۰.۰۲	۰.۱۱	۰.۰۱	۰.۱۲
۱۰	۰.۰۳	۰.۲۲	۰.۱	۰.۰۳	۰.۰۶	۰.۰۶	۰.۰۳	۰.۰۳	۰.۱	۰.۲۲	۰.۰۳
۱۱	۰.۱۷	۰.۱۵	۰.۲۴	۰.۰۳	۰.۱۷	۰.۰۹	۰.۱۷	۰.۰۳	۰.۲۴	۰.۱۵	۰.۱۷

زمان سیکل: ۱۰،۲۶											زمان آماده سازی: ۱،۱۸													
۹،۳					۸،۱۹					۱۰،۱۱					۱۰،۲۶					۹،۳۲				
۱	۲	۵	۶	۸	۱۰	۳	۴	۷	۹	۱۱	۱	۲	۵	۶	۸	۱۰	۳	۴	۷	۹	۱۱			
زمان سیکل: ۱۱،۲۱											زمان آماده سازی: ۱،۰۸													
۱۰،۴۴					۶					۱۰،۱۱					۱۱،۲۱					۹،۳۲				
۱	۲	۶	۸	۳	۱۰	۵	۴	۷	۹	۱۱	۱	۲	۶	۸	۳	۱۰	۵	۴	۷	۹	۱۱			

شکل ۲. دو جواب شدنی با مقادیر توابع هدف ناهمسو.

برای حل مسئله، از روش محدودیت (تبدیل هدف زمان آماده‌سازی به محدودیت و حل مسئله تک‌هدفه) استفاده شد. دلیل تعیین هدف زمان آماده‌سازی برای تبدیل شدن به محدودیت این بوده که برای هر ترتیب فعالیت‌ها، تخصیصی صورت گیرد که کمترین زمان سیکل را داشته باشد و بنابراین زمان سیکل می‌بایست به‌عنوان هدف باقی بماند:

$$\sum_{j=1}^m \sum_{i=1}^N \sum_{k=1 \& i \neq k}^N t_{s u_{i k}} \times z_{i k j} \leq f \quad (12)$$

۴. روش حل قطعی

کدینگ مدل ارائه شده در بخش قبل در محیط برنامه ۱۰/۱ CPLEX انجام شد و توسط یک سری از مسائل نمونه اجرا گردید. این مسائل نمونه، شامل ماتریس‌های زمان‌های آماده‌سازی ساخته شده آزمون Jackson، توسط پژوهش حاضر و آزمون‌های مربوط به زمان‌های پردازش و گراف‌های تقدم موجود در سایت اینترنتی بالانس خط مونتاژ^{۱۱} است.

با وجود بررسی بسیار روش‌های قطعی برای حل مسئله دوهدفه، بسیاری از جواب‌های به دست آمده با توجه به فرض پژوهش غیرقابل قبول بودند. فرض در نظر گرفته شده در این مسئله را می‌توان بدین صورت توضیح داد: برای هر جواب (ترتیبی از فعالیت‌ها)، بسته به چگونگی تخصیص فعالیت‌ها به ایستگاه‌ها، زمان‌های سیکل و زمان‌های آماده‌سازی بی‌شماری یافت می‌شود و این تحقیق توانایی گزارش کلیه جواب‌های پارو به خصوص در مسائل نمونه بزرگ را ندارد. با توجه به این موضوع و همچنین نبود روشی در پژوهش‌های صورت گرفته برای یافتن کمترین زمان آماده‌سازی یک جواب، برای هر ترتیب از فعالیت‌ها، تنها جوابی گزارش می‌شود که گروه‌بندی فعالیت‌ها در ایستگاه‌ها منجر به کمترین مقدار زمان سیکل برای آن ترتیب شود. به عبارت دیگر، تخصیص فعالیت‌ها به ایستگاه‌ها در صورتی مطلوب است که کمترین مقدار ممکن برای زمان سیکل به دست آید. این در حالی است که با وزنی کردن مسئله یا حتی محدود کردن زمان‌های آماده‌سازی، جواب‌هایی بدست می‌آیند که پس از رمزگشایی، کمترین زمان سیکل ممکن را ندارند و بنابراین، گروه‌بندی آن‌ها مطلوب نیست.

شکل ۳، جواب به دست آمده برای آزمون Jackson با تعداد ایستگاه‌های مساوی با ۶ نشان می‌دهد. زمان پردازش ایستگاه‌ها در بالای هر یک از آنها مشاهده

تخصیص فعالیت‌ها به ایستگاه‌ها با حل مدل ریاضی

زمان سیکل: ۱۰،۱۴؛ زمان آماده‌سازی: ۰،۵۹

۶	۱۰،۱۴	۸،۱۹	۱۰،۱۱	۸،۱۵	۴
۱	۲	۵	۴	۶	۸
				۱۰	۳
					۷
					۹
					۱۱

تخصیص فعالیت‌ها به ایستگاه‌ها با فرض کمینه‌سازی زمان سیکل

زمان سیکل: ۹،۳۲؛ زمان آماده‌سازی: ۱۰،۴

۹،۳	۷	۸،۱۹	۸	۸،۲۳	۹،۳۲
۱	۲	۵	۴	۶	۸
				۱۰	۳
					۷
					۹
					۱۱

شکل ۳. دو نوع تخصیص برای یک جواب.

می‌شود. روش قطعی برای جواب به دست آمده با محدودیت زمان آماده‌سازی کمتر از ۰،۷۵، زمان سیکل ۱۰،۱۴ و زمان آماده‌سازی ۰،۵۹ را به دست آورده است. اما با استفاده از اصل کمینه‌سازی زمان سیکل برای هر ترتیب فعالیت، زمان سیکل ۹،۳۲ و زمان آماده‌سازی ۱،۴ به دست می‌آید که با توجه به فرض پژوهش حاضر، تخصیص شدنی، تخصیص دوم است.

۵. الگوریتم شبیه‌سازی تیرید پارو^{۱۲}

مسئله بالانس خط مونتاژ حتی در مقیاس ساده آن، حالت خاصی از مسئله بسته‌بندی اقلام در ظروف (بین پکینگ)^{۱۳} است که در دسته مسائل Np-hard قرار دارد.^[۲۱] بنابراین، برای حل حالت‌های پیچیده‌ترین مسئله، باید روش‌های ابتکاری را در نظر داشت. الگوریتم شبیه‌سازی تیرید، یک روش متداول مبتنی بر جست‌وجوی همسایگی محسوب می‌شود. ایده اصلی این الگوریتم از رونه فیزیکی ذوب فلزات و نحوه سرد شده آنها برای رسیدن به پایداری الگو برگرفته شده است. این روش از یک جواب اولیه آغاز کرده و در یک دمای ثابت، به صورت متوالی در بین همسایگی‌ها با توجه به ساختار همسایگی‌های از پیش تعیین شده حرکت می‌نماید تا به شرایط توقف الگوریتم رسیده و متوقف شود. در طول اجرای برنامه، بهترین جواب به دست آمده ذخیره شده و در انتها به‌عنوان جواب همیشه الگوریتم نمایش داده می‌شود.

الگوریتم شبیه‌سازی تیرید پارو که برای اولین بار توسط کزبازک و جازکیویکز مطرح شد، یک روش فراابتکاری چندهدفه است که از مبانی الگوریتم تک‌هدفه شبیه‌سازی تیرید استفاده می‌کند.^[۲۲] در عین حال، این الگوریتم ویژگی‌های خاصی نیز دارد که طبیعت چندهدفه الگوریتم را ایجاد می‌کنند. اجرای الگوریتم و پارامترهای آن در ادامه مورد بررسی قرار خواهند گرفت.

۱.۵. بررسی بهینگی در مسئله دوهدفه مورد بررسی

در مسائل چندهدفه، مهم‌ترین پیش‌شرط برای داشتن مجموعه جواب‌های پارو، وجود اهداف ناسازگار است. بررسی مسئله مورد مطالعه، نشان داد که کاهش یک هدف، افزایش هدف دیگر را در پی خواهد داشت. بنابراین، مجموعه‌ای از جواب‌های غیرمغلوب توسط این روش‌ها تولید شد (شکل ۲). به صورت شهودی، می‌توان این ناسازگاری را به صورت زیر تحلیل کرد:

- زمان‌های آماده‌سازی جزئی جدانشدنی از زمان سیکل تولید هستند و هر دو هدف از جنس زمان هستند. اما زمانی‌که از منظر تولیدی به مساله بنگریم، بر خلاف زمان اجرای فعالیت‌ها که جزئی جداناپذیر از زمان سیکل خط هستند، زمان‌های آماده‌سازی لزوماً بین همه فعالیت‌ها اعمال نمی‌شوند و هر دو فعالیتی که پس از یکدیگر می‌آیند، زمان آماده‌سازی آنها جزئی از زمان تولید به حساب می‌آید؛
- کمینه‌سازی زمان آماده‌سازی نیازمند قرار گرفتن فعالیت‌های با زمان آماده‌سازی کم در کنار یکدیگر است. این عمل باعث می‌شود در اکثر اوقات، فعالیت‌ها به صورت نامتقارن (بخش فعالیت‌های طولانی در چند ایستگاه خاص و بقیه فعالیت‌ها در ایستگاه‌های دیگر) به ایستگاه‌ها تخصیص یابند و اختلاف زمان پردازش ایستگاه‌ها افزایش یابد که این موضوع باعث افزایش زمان سیکل خط می‌شود؛
- زمان‌های آماده‌سازی در طول خط باعث خستگی و سردرگمی اپراتورها می‌شوند. برخلاف کمینه‌سازی زمان سیکل که باعث افزایش بهره‌وری و خرسندی تولیدکننده

جواب‌های تعریف شده در این مسئله، به صورت آرایه‌هایی با اندازه N هستند که جایگشت N فعالیت، خانه‌های آرایه را پر می‌کند. در این پژوهش، سه روش برای ساخت جواب اولیه بررسی می‌شود. دو روش اول بر اساس ایستگاه عمل می‌کنند؛ به نحوی که کلیه فعالیت‌های قابل تخصیص می‌بایست ابتدا ایستگاه نیمه پر k را پر کنند پیش از اینکه به ایستگاه $k + 1$ تخصیص یابند. در روش اول، فعالیت‌های بین مجموعه فعالیت‌های قابل تخصیص (فعالیت‌هایی که فعالیت‌های پیش‌نیازی آنها وارد شده‌اند)، به ایستگاه اختصاص می‌یابد که در بین فعالیت‌های دیگر، بیشترین مقدار «مجموع زمان پردازش و آماده‌سازی» را داشته باشد (SH-max-ts) در روش دوم، برعکس روش اول، فعالیت با کمترین مقدار «مجموع زمان پردازش و آماده‌سازی» اختصاص داده می‌شود (SH-min-ts). برای پرکردن ایستگاه‌های کاری، می‌بایست یک زمان سیکل نظری در نظر گرفته شود تا فعالیت‌ها پس از پر شدن هر ایستگاه، به ایستگاه بعدی اختصاص یابند. زمان سیکل نظری از رابطه زیر به دست می‌آید:

$$C_{thws} = \frac{\sum_{i=1}^N \left(t_i + \min_{\forall k, i \neq k} (tsu_{ik}) \right)}{m} \quad (15)$$

روش سوم، تولید جواب بصورت تصادفی و سپس شدنی^{۱۷} کردن آن است. شدنی کردن یک جواب در بخش ۸.۵ تشریح خواهد شد. پس از شدنی کردن جواب، زمان سیکل کمینه و مجموع زمان‌های آماده‌سازی برای آن محاسبه خواهند شد.

در این مسئله، تصمیم‌گرفته شد که دو عامل برای جست‌وجوی فضای حل در نظر گرفته شود و هر یک به سوی یکی از اهداف متمایل شود. آزمایش‌ها نشان داد که تعداد بیشتر عامل‌ها به همان نسبت زمان اجرای برنامه را افزایش خواهد داد و بهبود چندانی در کیفیت مجموعه پارتو ایجاد نخواهد کرد. برای تولید هر یک از این عامل‌ها، روش‌های مختلفی وجود دارد. با توجه به آزمایشات متعدد، مشخص شد که عامل اول که گرایش بیشتری به سوی زمان سیکل دارد، توسط روش SH-max-ts تولید شود. برای تولید عامل دوم، آزمایش‌ها نشان داد که تفاوت چندانی بین روش‌های مختلف از لحاظ کیفیت نهایی پارتو وجود ندارد. بنابراین از روش تصادفی برای تولید آن استفاده شد.

با توجه به آزمایشات صورت گرفته، بردار وزنی برای عامل اول به صورت (۰٫۲ و ۰٫۸) و برای عامل دوم (۰٫۸ و ۰٫۲) تعیین شد (عدد اول وزن زمان سیکل و عدد دوم مربوط به زمان آماده‌سازی است). برای به‌روز کردن وزن عامل‌ها پس از هر بار تکرار الگوریتم، ضریب دفع روی ۰٫۰۰۰۳۵ تعیین گردید. لازم به ذکر است که مقادیر تعیین شده برای پارامترهای الگوریتم، پس از انجام مکرر الگوریتم روی مقادیر مختلف پارامترها و یافتن مقادیر مناسب‌تر و کاراتر برای پارامترها تعیین شده است.

۴.۵. نرمال‌سازی توابع هدف

در الگوریتم PSA، برای محاسبه احتمال پذیرش جواب‌ها و تعیین فاصله بین دو عامل، از مقادیر هدف نرمال شده می‌بایست استفاده کرد. در این مسئله، برای نرمال‌سازی مقادیر هدف، دو روش مورد بررسی قرار گرفته است. در روش اول، هر هدف را می‌توان بر بازه تقریبی آن هدف یا R_z (بازه بین بیشینه و کمینه هدف) تقسیم کرد و اگر مقدار این بازه نامعلوم یا غیرقابل محاسبه باشد، مقدار تقریبی بیشینه هدف را به‌عنوان R_z می‌توان در نظر گرفت. در روش دوم از رابطه زیر استفاده می‌شود که طی آن، مقادیر هدف در بازه $[1, 0]$ قرار می‌گیرند.

می‌شود، کمینه‌سازی زمان‌های آماده‌سازی باعث خرسندی اپراتورها و کارگران درگیر در واحد تولیدی می‌شوند. ممکن است برای تولیدکننده، هدف افزایش بهره‌وری خط بسیار مهم‌تر از افزایش رضایت و بهره‌وری کارکنان باشد، اما تولیدکنندگان حاضرند تا حدی هرچند اندک، زمان سیکل را قربانی زمان‌های آماده‌سازی کنند تا هم میزان بازو بسته شدن دستگاه‌ها و هم درگیری فیزیکی اپراتورها در طی فرایند تولید کاهش یابد. بنابراین، ارائه مجموعه‌ای از جواب‌های شدنی با زمان سیکل و زمان آماده‌سازی متفاوت می‌تواند دست تولیدکننده را در انتخاب گزینه مناسب جهت تولید، باز نگه دارد.

۲.۵. احتمال پذیرش جمعی

در مسائل چندهدفه، هنگام حرکت از جواب G به جواب G' ، یکی از حالات زیر اتفاق می‌افتد:

- G' بر G غالب است یا با آن مساوی است؛
- G' مغلوب G است؛
- هیچ‌یک بر دیگری غالب نمی‌باشد.

در حالت اول، G' جای G را به‌عنوان جواب کنونی می‌گیرد. اما در دو حالت دیگر، G' بر G غالب نیست و با احتمالی کمتر از یک به‌عنوان جواب کنونی پذیرفته یا رد می‌شود تا از افتادن الگوریتم در بهینه محلی جلوگیری به عمل آید. بنابراین، با جمع کردن اهداف با استفاده از تابع وزنی چیبیچف،^{۱۴} احتمال انتخاب جواب جدید با استفاده از روابط ۱۳ یا ۱۴ محاسبه می‌شود که در آنها، δ_z ضریب وزنی برای هدف z ، Δ_G و مجموعه ضرایب وزنی برای جواب G و G' مقدار نرمال هدف z برای جواب G است. نرمال‌سازی اهداف در بخش ۵.۴ توضیح داده شده است.

$$P_1(G, G', \Delta_G, T_I) = \min \left\{ 1, \exp \left(- \max_{j=1, \dots, J} \frac{\delta_j [z_j(G') - z_j(G)]}{T_I} \right) \right\} \quad (13)$$

$$P_2(G, G', \Delta_G, T_I) = \min \left\{ 1, \exp \left(- \sum_{j=1}^J \frac{\delta_j [z_j(G') - z_j(G)]}{T_I} \right) \right\} \quad (14)$$

۳.۵. جواب‌های اولیه (عامل‌ها)^{۱۵}

هنگام شروع اجرای الگوریتم PSA، گروهی از جواب‌های اولیه تولید می‌شوند تا به الگوریتم امکان بازدید محیط وسیع‌تری از فضای حل را بدهند. برای هر یک از جواب‌ها (که اصطلاحاً به آن‌ها، عامل گفته می‌شود)، یک مجموعه ضرایب وزنی اختصاص می‌یابد. نکته اینجا است که هر ضریب مربوط به یک هدف برای آن جواب است و بدیهی است که ضریب بزرگ‌تر وزن هدف z ، جواب را به سوی مناطقی از فضای حل مسئله با مقدار هدف z بهتر سوق می‌دهد. بنابراین، برای هر جواب، ضرایب وزنی طوری انتخاب می‌شوند که هر یک از جواب‌ها را به سوی نقاط مختلف فضای حل حرکت دهند. مجموع ضرایب وزنی برای هر جواب، یک است. این ضرایب در طول اجرای الگوریتم با استفاده از ضریب دفع^{۱۶} α دستکاری می‌شوند تا بدین ترتیب، جواب‌ها در طول اجرای الگوریتم بتوانند محیط وسیع‌تری از فضای حل را پوشش دهند.

• دمای نهایی: باید آنقدر پایین باشد تا در انتهای برنامه به جواب‌های نامرغوب اجازه جایگزین شدن با جواب کنونی ندهد؛

• تعداد دمای مشاهده شده: مسلماً افزایش تعداد دماها باعث بهبود کیفیت جواب بهینه می‌شود. اما از سوی دیگر، زمان اجرای برنامه را طولانی‌تر می‌کند. در نتیجه، می‌بایست بین کیفیت جواب مورد انتظار و زمان برنامه مقایسه‌ای صورت گرفته و تعداد دمای مشاهده شده را به دست آورد؛

• روند کاهشی دما: برای کنترل مکانیزم پذیرش، مشخص کردن این نکته بسیار مهم است که با چه روندی دما را باید کاهش داد. به نحوی که جواب کنونی در بیشتر زمان اجرای الگوریتم به سوی جواب‌های نامرغوب متمایل نشود.

از آنجا که هر یک از عامل‌ها به سوی یکی از هدف‌ها متمایل هستند و اهداف، حتی با وجود نرمال‌سازی، مقیاس‌ها و رفتارهای متفاوتی در هر مسئله نمونه دارند، می‌بایست برای هر عامل تنظیمات مختلف دما انجام داد. مقادیر بهینه دمای اولیه و نهایی برای هر عامل با توجه به نوع مسئله نمونه، در بخش نمادها مشاهده می‌شود. دماهای به دست آمده برای هر دو عامل، بر اساس آزمایش‌های متعدد به دست آمده است. در این جدول، مقدار بیشینه تقریبی زمان سیکل و CT_{Max} مقدار بیشینه تقریبی زمان سیکل و ST_{Min} و ST_{Max} به ترتیب مقادیر تقریبی بیشینه و کمینه زمان راه‌اندازی کل خط برای مسئله نمونه خاص می‌باشند که در بخش نرمال‌سازی به آنها اشاره شد.

تعداد دماهای مشاهده شده بیشتر بستگی به اندازه مسئله نمونه دارد و در این الگوریتم، عدد 500 برای مسائل کوچک و 3000 برای مسائل بزرگ تعیین شده است. در پژوهش‌های صورت گرفته، برای روند کاهشی، از دو روش خطی و نمایی استفاده شده است و در این مسئله، نوع نمایی روند کاهشی، نشان داد که نتایج بهتری نسبت به نوع خطی بدست می‌دهد:

$$A = \frac{(T_0 - T_f)(N + 1)}{N}$$

$$T_i = \frac{A}{i + 1} + B$$

$$B = T_0 - A \quad i = 1, \dots, N \quad (17)$$

در این فرمول، T_0 ، T_f و N به ترتیب، دمای اولیه، دمای نهایی و تعداد دمای مشاهده شده هستند. همچنین برای محاسبه احتمال پذیرش جواب‌های جدید، از رابطه ۱۴ استفاده شده است.

۷.۵. تعداد جست‌وجوی همسایگی در یک دما

در هر دما، تعدادی از همسایه‌های جواب کنونی جست‌وجو می‌گردد. بیشتر پژوهش‌های صورت گرفته در رابطه با شبیه‌سازی تبرید، تعداد جست‌وجوها در هر دما را یکسان در نظر می‌گیرند. این در حالی است که در صورت ثابت بودن تعداد جست‌وجوها در هر دما، حدود نیمی از جست‌وجوها با احتمال قبولی قابل توجه جواب‌های نامرغوب انجام خواهد شد. در همین راستا، در این مسئله، علاوه بر لحاظ کردن تعداد ثابت جست‌وجو، دو روند خطی و نمایی نیز برای تعیین NNS پیشنهاد شده است. روابط مربوط به هر کدام از این روندها در جدول ۲ نشان داده شده است. I_f ، I_r و N به ترتیب، تعداد جست‌وجوها در دمای T_i ، تعداد جست‌وجوها در دمای T_f و تعداد دماهای مشاهده شده هستند.

در این رابطه، z_j^{max} و z_j^{min} به ترتیب، کمینه و بیشینه تقریبی برای هدف z می‌باشند.

$$\bar{z}_j = \frac{z_j - z_j^{min}}{z_j^{max} - z_j^{min}} \quad (16)$$

در این مسئله، گرچه هر دو هدف از جنس زمان هستند، اما اندازه آنها هم‌مقیاس نبوده و در بیشتر حالات، زمان سیکل، مقادیر بزرگ‌تری نسبت به زمان آماده‌سازی خط دارد. دو روش متفاوت برای نرمال‌سازی هر یک از دو هدف استفاده شده است. برای نرمال‌سازی زمان سیکل، مقادیر این هدف بر بیشینه تقریبی مقدار زمان سیکل برای آن مسئله نمونه خاص تقسیم می‌شوند. نرمال‌سازی زمان آماده‌سازی براساس رابطه ۱۶ صورت می‌گیرد که در آن، مجموع ماکزیمم زمان‌های آماده‌سازی مربوط به هر N فعالیت و z_j^{min} مجموع $N - m + 1$ زمان‌های آماده‌سازی کمینه به دست آمده از N فعالیت است.

۵.۵. مکانیزم جست‌وجوی همسایگی

برای اجرای جست‌وجوی محلی، روش‌های مختلفی در پژوهش‌های صورت گرفته بررسی شده است. سه نوع جست‌وجوی محلی عبارتند از:

- عملگر یک نقطه‌ای (SPO):^{۱۸} جای یک فعالیت که به صورت تصادفی انتخاب شده عوض می‌شود و در مکان دیگری بین دو فعالیت دیگر قرار می‌گیرد؛
- عملگر جابجایی:^{۱۹} جای دو فعالیت که به صورت تصادفی انتخاب شده‌اند، عوض می‌شود؛
- عملگر چند نقطه‌ای (MSPO):^{۲۰} دو فعالیت را به طور مجزا انتخاب کرده و جای آنها را در آرایه تغییر می‌دهد.

با توجه به ناکارآمدی ساختارها در مقاطعی از کار الگوریتم و افتادن عامل‌ها در بهینه محلی، ممکن است لازم باشد ساختار جست‌وجوی همسایگی تغییر کند. بنابراین، از ساختار محلی متغیر که به صورت زیر صورت تعریف می‌گردد، استفاده می‌شود:

- اگر عملگر i نتواند جواب بهینه به دست آمده توسط الگوریتم را پس از 5 مرتبه کاهش دما بهبود دهد (واریانس بهترین مقادیر توابع هدف پس از 5 تکرار الگوریتم نزدیک به صفر باشد)، عملگر $i + 1$ مورد استفاده قرار می‌گیرد. در این الگوریتم، ساختار متغیر ((SPO - Swap - MSPO)) به عنوان ساختار جست‌وجوی همسایگی انتخاب گردید و ابتدا الگوریتم با ساختار SPO شروع به فعالیت خواهد کرد. پس از عملگر MSPO، ساختار دوباره تکرار می‌شود. پاسخ به دست آمده ممکن است محدودیت تقدم را رعایت نکند. بنابراین، از مکانیزم ترمیم جواب‌های غیرقابل قبول که در بخش ۸.۵ اشاره شده است برای موجه ساختن جواب‌ها استفاده می‌شود.

۶.۵. تنظیمات دما

مقدار دما در طول برنامه باید به گونه‌ای تعیین شود که در ابتدا جواب‌های نامرغوب با احتمال بالا پذیرفته شوند و هرچه به انتهای برنامه نزدیک می‌شویم، این احتمال کاهش یابد. در همین راستا، می‌بایست موارد زیر را برای کنترل دما به طور مناسب تخمین زد:

- دمای اولیه: باید آنقدر بالا باشد تا کل فضای حل در ابتدای برنامه، شانس یکسانی برای انتخاب شدن داشته باشند؛

جدول ۲. دمای اولیه و دمای نهایی برای عامل‌های مختلف.

عامل	واریانس موجود در زمان انجام فعالیت‌ها	دمای اولیه	دمای نهایی
متماثل به زمان سیکل بهتر	پایین	$\frac{1}{5} CT_{max}$	$\frac{0.1}{CT_{max}}$
	بالا	$\frac{200}{CT_{max}}$	$\frac{2.2}{CT_{max}}$
متماثل به زمان آماده‌سازی بهتر	پایین	۰/۱	$\frac{0.2}{ST_{max} - ST_{min}}$
	بالا	۰/۱	$\frac{0.1}{ST_{max} - ST_{min}}$

روندهای مختلف برای تعیین NNS:

$$I_i = I_0 + \left[i * \frac{I_f - I_0}{N} \right] \quad i = 1 \dots N \quad (18) \quad \text{خطی}$$

$$I_i = a \times e^{c(i+1)} \quad a = e^{(\ln(I_0) - c)} \quad c = \frac{\ln(\frac{I_f}{I_0})}{N} \quad i = 1 \dots N \quad (19) \quad \text{نمایی}$$

اگرچه بنا به آزمایش‌های انجام شده بر روی هر دو روند تفاوت چندانی بین کیفیت نتایج نهایی الگوریتم مشاهده نشد، از نظر زمان اجرای الگوریتم، روند نمایی زمان بسیار کمتری نسبت به روند خطی و تعداد ثابت جست‌وجو مصرف می‌کند. در نتیجه، روند نمایی بین ۱۰۰ تا ۱۲۰۰ جست‌وجوی همسایگی (به ترتیب برای دمای اولیه و دمای نهایی) در الگوریتم بکار گرفته شد.

عملگر ساخت ماتریس M
 برای i از ۱ تا N انجام بده (N تعداد فعالیت‌ها)
 برای j از ۱ تا N انجام بده
 اگر فعالیت i باید بلافاصله پیش از فعالیت j پایان یابد آنگاه $M_{jk} = 1$ و در غیر این صورت $M_{jk} = 0$
 پایان حلقه
 پایان حلقه
 برای i از ۱ تا N انجام بده
 برای j از ۱ تا N انجام بده
 اگر $M_{ij} = 1$ باشد آنگاه برای k از ۱ تا N انجام بده $M_{jk} \otimes M_{ik} = M_{jk}$ پایان شرط
 پایان حلقه
 پایان حلقه
 برای i از ۱ تا N انجام بده
 $x = 0$
 برای i از ۱ تا N انجام بده
 اگر $M_{ij} = 1$ باشد آنگاه به x یک واحد اضافه شود، پایان شرط
 پایان حلقه
 $M_{N+1,j} = x$
 پایان حلقه
 نمایش بده M

0	0	1
0	0	1
1	1	1

شکل ۴. شبه کد عملگر ساخت ماتریس M .

۸.۵. ترمیم جواب‌های غیر قابل قبول

در الگوریتم پیشنهادی، جواب اولیه تولید شده تصادفی و با جواب ساخته شده توسط جست‌وجوی همسایگی ممکن است با توجه به محدودیت‌های تقدم فعالیت‌ها به یکدیگر غیر قابل قبول باشند. بنابراین، می‌بایست با استفاده از مکانیزم ترمیم جواب‌ها، آنها را به جواب‌های قابل قبول تبدیل کرد. مکانیزم استفاده شده در این مسئله، مکانیزم ارائه شده در پژوهش [۲۳] می‌باشد. بر این اساس، ابتدا ماتریس تقدم M می‌بایست برای چک کردن قابل قبول بودن جواب ساخته شود. در این ماتریس، اگر کار i قبل از کار k بیاید، $M_{ik} = 1$ بوده و در غیر این صورت، صفر است. دقت شود که در واقع، فعالیت i ، فعالیت پیش‌نیاز k محسوب می‌شود و نباید لزوماً بلافاصله پس از فعالیت k بیاید. شبه کد ساخت ماتریس M در شکل ۴ نشان داده شده است.

در حلقه اول عملگر، اگر فعالیت i بلافاصله قبل از فعالیت k در گراف تقدم انجام گیرد $M_{ik} = 1$ بوده و در غیر این صورت صفر خواهد شد. در حلقه دوم، اگر فعالیت i فقط و فقط قبل از k در گراف تقدم بیاید، $M_{ik} = 1$ بوده و در غیر این صورت صفر خواهد شد. در نهایت، حلقه سوم، کلیه فعالیت‌های پیش‌نیاز فعالیت k را شمرده و عدد مربوط را در سطر آخر ماتریس ذخیره می‌کند. حال با استفاده از ماتریس M ، ترمیم جواب‌ها با استفاده از تابع زیر (شکل ۵) صورت می‌گیرد.

این عملگر، با چک کردن تمامی فعالیت‌های آرایه x ، اولین فعالیت قابل قبول a را به عنوان فعالیت منتخب وارد آرایه جدید PS می‌کند. پس از آن، با به‌روز کردن ماتریس M' ، فعالیت a را از گراف تقدم حذف می‌کند تا فعالیت‌هایی که پیش از آن می‌بایست منتظر انجام فعالیت a باشند، بتوانند به عنوان فعالیت بعدی انتخاب شوند. اولویت تخصیص فعالیت‌های کاندید به ترتیب آنها در جواب x بستگی دارد.

عملگر ترمیم جواب نشدنی آرایه X

M' را به عنوان یک نسخه از M می‌کن
 $j = 0$ // نشان برای آرایه RX
 $i = 1$ // شمارنده تعداد تکرارها
 تکرار کن
 $a = X_i$

اگر $M'_{N+1,a} = 0$ باشد و a در آرایه RX قرار نگرفته باشد آنگاه

به X یک واحد اضافه شود

$$RX_j = a$$

M' را به‌روز کن: برای k از ۱ تا N انجام بده: اگر $M'_{ik} = 1$ است آنگاه $M'_{ik} = 0$ شود و از $M'_{N+1,k}$ شود

یک واحد کسر گردد؛ انتهای شرط

در آرایه X ، فعالیت a حذف گردد

انتهای شرط

به شمارنده i یک واحد اضافه شود

اگر $i > N$ شد آنگاه $i = 1$ شود؛ پایان شرط

تا زمانی که $j = N$ گردد

جواب RX را بعنوان جواب شدنی نمایش بده

شکل ۵. شبه کد عملگر ترمیم جواب‌های غیر قابل قبول.

۹.۵. ارزیابی جواب‌ها

نحوه تعریف جواب در این مسئله به نحوی است که برای ارزیابی جواب‌ها، می‌بایست فعالیت‌ها را به‌طور مناسبی به ایستگاه‌ها تخصیص داد تا کمترین زمان سیکل ممکن به دست آید. برای محاسبه زمان سیکل در این مسئله، به علت پایین بودن زمان حل و کیفیت بهتر جواب‌ها، از روش متداول ارائه شده در پژوهش [۹] استفاده شده است. شکل ۶، شمای کلی فرایند محاسبه زمان سیکل را نشان می‌دهد.

عملگر ارزیابی-کن-آرایه X

تولید می‌کند. از این رو، در ادامه، الگوریتمی برای مقایسه نتایج معرفی خواهد شد.

۱: زمان سیکل C را زمان سیکل کمینه تئوریک در نظر بگیر (مجموع زمان‌های انجام فعالیت‌ها تقسیم بر تعداد ایستگاه‌های کاری).

۲: بر اساس ترتیب فعالیت‌ها در آرایه X فعالیت‌ها را به $m-1$ ایستگاه کاری اول اختصاص بده بنحویکه زمان اجرای هر ایستگاه از C کوچکتر باشد، و فعالیت‌های باقیمانده را به ایستگاه آخر اختصاص بده.

۳: بار کاری W_i برای کلیه ایستگاه‌های کاری از ۱ تا m و بار کاری باقی‌مانده PW_i برای کلیه ایستگاه‌های کاری از ۱ تا $m-1$ بصورت مقابل تعریف می‌شود. همان‌زمان اجرای ایستگاه Z است و PW_Z زمان اجرای ایستگاه است وقتی که اولین فعالیت تخصیص یافته به ایستگاه $Z+1$ به انتهای ایستگاه Z اضافه گردد.

$$c = \min \{PW_1, PW_2, \dots, PW_m\} \text{ و } c_m = \max \{W_1, W_2, \dots, W_m\} \quad ۴$$

۵: اگر $c_m > c$ باشد آنگاه به خط ۲ برگرد.

۶: c_m زمان سیکل بهینه آرایه X است.

شکل ۶. شبیه‌کد محاسبه زمان سیکل.

الگوریتم شبیه‌سازی تیرید پارتو

برای راز ۱ تا ۲ انجام بده // تولید دو جواب اولیه $||$

آر را با یکی از روش‌های تولید جواب اولیه ایجاد کن

ترمیم جواب نشدنی آرایه آر

C_{avg} و CV_{avg} را با عملگر ارزیابی-کن-آرایه آر محاسبه کن

وزن اولیه به جواب آر بده

پایان حلقه

تولیدکننده به شاخص توقف نرسیده ایم انجام بده

برای راز ۱ تا ۲ انجام بده

وزن اولیه جواب آر را به روز کن

برای i از ۱ تا NNS انجام بده

NX_i - همسایه آر که با یک ساختار همسایگی NSS بدست آمده است

ترمیم جواب نشدنی آرایه NX_i

C_{avg} و CV_{avg} را با عملگر ارزیابی-کن-آرایه NX_i محاسبه کن

اگر NX_i بر NX غالب نیست آنگاه مجموعه پارتو را با NX_i به روز کن، انتهای شرط

p - مجموع وزنی $C_{avg} - CV_{avg}$ و $CV_{avg} - CV_{avg}$ یک عدد رندوم است

اگر $r < \min(1, \exp(-p/Temp))$ باشد آنگاه مقادیر آر، CV_{avg} و CV_{avg} را با NX_i و CV_{avg} جایگزین کن، انتهای شرط

جایگزینی کن، انتهای شرط

انتهای حلقه

انتهای حلقه

ساختار همسایگی NSS و تعداد جستجوی در هر دما NNS را به روز کن

دمای $Temp$ را به روز کن

انتهای شرط

شکل ۷. شبیه‌کد شبیه‌سازی تیرید پارتو برای مسئله مورد بررسی.

برای ارزیابی جواب‌ها، ذکر این نکته ضروری است که روش ارزیابی جواب‌ها مبتنی بر کمینه‌سازی زمان سیکل و بیکاری خط است. پس از تخصیص فعالیت‌ها به ایستگاه‌ها، برای محاسبه زمان آماده‌سازی کل خط، مجموع زمان پردازش خط از مجموع زمان پردازش فعالیت‌ها کسر می‌گردد.

شکل ۷، شمای کلی الگوریتم شبیه‌سازی تیرید پارتو برای مسئله مورد مطالعه را نشان می‌دهد.

۶. نتایج محاسباتی

الگوریتم شبیه‌سازی تیرید پارتو، در محیط دلفی پاسکال ۱۱ تحت سیستم عامل مایکروسافت ویندوز ویستا کد شد و روی یک رایانه با CPU دو هسته‌ای ۱/۵ گیگاهرتز و رم ۲ گیگابایت اجرا شد. به علت جدید بودن مسئله مطرح شده، روش حالی وجود نداشت تا بتوان نتایج آن را با نتایج الگوریتم PSA مقایسه کرد. از طرفی، همان‌طور که پیشتر اشاره شد، روش قطعی از تولید مجموعه جواب‌های پارتو ناتوان است و با توجه به فرضیات در نظر گرفته شده، جواب‌های غیرقابل قبول

۱.۶. مسائل نمونه

به علت وجود زمان‌های آماده‌سازی، به جز مسائل نمونه موجود در [۳] که در دسترس ما نبود، مسائل نمونه دیگری برای زمان‌های آماده‌سازی در دست نیست. لذا برای آزمایش الگوریتم پیشنهادی، مسائل نمونه با در نظر گرفتن اندازه، پیچیدگی گراف و واریانس زمان‌های آماده‌سازی ساخته شد. از این رو، چهار مسئله نمونه از مجموعه مسائل نمونه معروف SALBP براساس اندازه و پیچیدگی گراف تقدم انتخاب شد که این مسائل در سایت اینترنتی بالانس خط مونتاژ موجود است:

- Sawyer: اندازه (۳۰ کوچک فعالیت) و پیچیدگی گراف پایین (۴۴/۸۳):
- Lutz۱: اندازه کوچک (۳۲ فعالیت) و پیچیدگی گراف بالا (۸۳/۴۷):
- Arcus۱: اندازه بزرگ (۸۳ فعالیت) و پیچیدگی گراف بالا (۵۹/۰۹):
- Arcus۲: اندازه بزرگ (۱۱۱ فعالیت) و پیچیدگی گراف بالا (۴۰/۳۸):

برای هر مسئله، زمان پردازش فعالیت‌ها و گراف تقدم ثابت نگاه داشته شد و دو مقدار نمونه برای تعداد ایستگاه‌ها تولید شد که این مقادیر به صورت تصادفی و در بازه m_{min} و m_{max} تولید شده اند (m_{min} و m_{max} به ترتیب کمترین تعداد ایستگاه‌ها و بیشترین تعداد ایستگاه‌ها براساس اطلاعات موجود در سایت اینترنتی بالانس خط هستند). برای هر مسئله نمونه، سطوح مختلف زمان‌های آماده‌سازی بصورت زیر در نظر گرفته شد:

- برای زمان‌های آماده‌سازی با واریانس پایین، ماتریس زمان‌های آماده‌سازی به صورت تصادفی از بازه $[\min t_i \times 0.25, \max t_i]$ تولید شده‌اند؛
 - برای زمان‌های آماده‌سازی با واریانس بالا، ماتریس زمان‌های آماده‌سازی به صورت تصادفی از بازه $[\min t_i \times 0.75, \max t_i]$ تولید شده‌اند.
- بنابراین، با توجه به ترکیبات مختلف اندازه، پیچیدگی گراف، واریانس زمان‌های آماده‌سازی و تعداد ایستگاه‌ها، ۱۶ مسئله نمونه مورد بررسی قرار می‌گیرد.

۲.۶. الگوریتم شبیه‌سازی تیرید چندهدفه

الگوریتم شبیه‌سازی تیرید چندهدفه بکار گرفته شده ۲۲ حالت میانی دو الگوریتم SA و PSA است که عموماً در مسائل چندهدفه حل شده توسط الگوریتم شبیه‌سازی تیرید از این نوع استفاده شده است. تفاوت عمده بین الگوریتم‌های PSA و MoSA عبارتند از:

- MoSA با یک جواب اولیه کار می‌کند؛
- در طول اجرای الگوریتم، وزن‌های مربوط به هر هدف ثابت در نظر گرفته می‌شوند و برای هر زوج وزن، یک مجموعه پارتو به دست می‌آید.

الگوریتم MoSA همچنین می‌تواند از وزن‌های متغیر در طول اجرای الگوریتم استفاده کند؛ اما استفاده از روندهای متغیر معرفی شده در ادبیات این الگوریتم توانست مجموعه پارتو با کیفیت مناسب تولید کند. (برای توضیحات بیشتر در مورد این الگوریتم به [۲۴] مراجعه کنید).

مشکل اصلی استفاده از این الگوریتم، این است که برای هر مسئله نمونه، الگوریتم می‌بایست بارها با وزن‌های مختلف اجرا شود و هر بار دمای الگوریتم بازید و تغییر داده شود که این موضوع، استفاده از آن را زیر سوال می‌برد. استفاده از

زمانی که دو مجموعه جواب پارتو ادغام می‌شوند، برخی از جواب‌ها مغلوب می‌شوند و باقی‌جاوب‌ها، مجموعه جدیدی از جواب‌های غیرمغلوب را تشکیل می‌دهند. مجموعه‌ای که درصد بیشتری از اعضای آن در مجموعه جدید باشد، نشان‌دهنده عملکرد بهتر الگوریتم تولیدکننده آن مجموعه می‌باشد. رابطه مربوط به این معیار به صورت زیر است. در این رابطه، $e_i = 1$ خواهد بود اگر i امین جواب مجموعه پارتو در پارتوی ادغام موجود باشد. در غیر این صورت، $e_i = 0$ خواهد بود. مسلماً مقادیر بالای PSN مطلوبیت بیشتری خواهد داشت.

$$PSN = \frac{\sum_{i=1}^n e_i}{n} \quad (23)$$

معیار چهارم، تعداد اعضای مجموعه پارتوی حاصل از اجرای هر الگوریتم را نشان می‌دهد. مشخصاً، هرچه تعداد اعضای مجموعه‌ای بیشتر باشد، گزینه‌های بیشتری را به تصمیم‌ساز ارائه داده و مطلوب‌تر است.

معیار پنجم یا همان رابطه اسکات، یکنواختی پراکندگی جواب‌ها (فاصله بردارهای مجاور در فضای حل) را ایجاد می‌کند. اما رابطه کلاسیک اسکات، برای محاسبه فاصله جواب‌ها از رابطه اسکالار استفاده می‌کند. این در حالی است که در بسیاری از مسائل مانند مسئله فعلی، روابط جمعی به علت یکسان نبودن مقیاس‌ها معنادار نیستند. بنابراین، به جای محاسبه فاصله نقاط به صورت جمعی، از فاصله اقلیدسی استفاده می‌شود. در رابطه اسکات که به صورت زیر تعریف می‌شود، مقادیر i و j برابر $1, 2, \dots, n$ بوده، \bar{c} میانگین مقادیر c_i است و n تعداد بردارها در مجموعه جواب پارتو است. در این روش، $MSS = 0$ به این مفهوم است که همه اعضا به صورت یکنواخت و مجزا از هم پراکنده شده‌اند.

$$MSS = \sqrt{\frac{\sum_{i=1}^{n-1} (\bar{c} - c_i)^2}{n-2}} \quad (24)$$

$$c_i = \sqrt{(f_{1i} - f_{1(i+1)})^2 + (f_{2i} - f_{2(i+1)})^2} \quad \bar{c} = \frac{\sum_{i=1}^{n-1} c_i}{n-1}$$

۴.۶. مقایسه الگوریتم‌ها

جدول ۳، مقادیر به دست آمده میانگین معیارهای اندازه‌گیری عملکرد برای مجموعه‌های پارتوی حاصل از دو الگوریتم را نشان می‌دهد. نزدیکی مقادیر MID نشان می‌دهد که نقاط مجموعه‌های پارتو نزدیک به یکدیگر هستند و از لحاظ کیفیت، اختلاف چندانی بین آنها وجود ندارد. گرچه با توجه به مقادیر PSN، تعداد بیشتری از نقاط مجموعه‌های پارتو حاصل از MoSA در مجموعه غیرمغلوب کلی (حاصل از ادغام دو مجموعه پارتو) نسبت به PSA وجود دارد، پراکندگی بیشتر جواب‌ها (SNS) و تعداد جواب‌های بیشتر مجموعه پارتو (ONVG) حاصل از PSA نشان‌دهنده فراهم نمودن گزینه‌های بیشتر و گوناگون‌تر برای تصمیم‌گیر است. مقادیر نزدیک MSS نیز نشان‌دهنده یکنواختی یکسان پراکندگی جواب‌ها در دو الگوریتم است.

شکل ۸، نمودار جواب‌های به دست آمده از دو الگوریتم را برای مسئله ARC۳ با ۲۱ ایستگاه کاری و واریانس بالایی زمان‌های آماده‌سازی نشان می‌دهد. همان‌طور که مشخص است، الگوریتم PSA نتایج با کیفیت‌تر و متنوع‌تری نسبت به الگوریتم MoSA از خود نمایش داده است. نتایج به دست آمده در مجموع نشان می‌دهد که الگوریتم PSA با کیفیت جواب‌های قابل قبول می‌تواند گزینه‌های متعدد و با پراکندگی و یکنواختی مناسب را برای تصمیم‌گیر فراهم کند. ذکر این نکته در انتها ضروری است که الگوریتم MoSA، گرچه کارایی مناسبی دارد، به علت تعداد اجراهای بسیار برای وزن‌های متفاوت، بسیار زمان‌بر و ناکارآمد است.

MoSA در این مسئله صرفاً برای ارزیابی عملکرد الگوریتم PSA بوده است و از آنجا که ماهیت دو الگوریتم یکسان است، می‌توان از مقادیر پارامترهای PSA برای ماژول‌های MoSA استفاده کرد. این در حالی است که دمای الگوریتم MoSA، می‌بایست برای هر زوج وزن اهداف، تغییر یابد و الگوریتم برای هر زوج بطور مجزا اجرا شود. انتخاب درست دما برای هر زوج وزن نیازمند آزمایشات بسیار و وقت‌گیر است. برای هر مسئله نمونه، وزن‌های انتخاب شده از زوج (۰٫۹، ۰٫۱) شروع شده و به (۰٫۱، ۰٫۹) ختم می‌شود و وزن‌ها به اندازه ۰٫۱ در آزمایشات دیگر کم و زیاد می‌شوند. جمعاً ۹ آزمایش برای تولید مجموعه‌های پارتو انجام می‌شود و در نهایت، ۹ مجموعه تولید شده در هم ادغام می‌شوند تا یک مجموعه پارتو برای مسئله نمونه به خصوص تولید گردد.

۳.۶. معیارهای اندازه‌گیری کارایی الگوریتم

هریک از الگوریتم‌ها، مجموعه‌ای از جواب‌های غیر مغلوب یا مجموعه پارتو ایجاد می‌دهند. مقایسه جواب‌ها با توجه به دارا بودن اعداد غیر هم‌جنس، نیازمند بهره‌گیری از معیارهایی است که بتوانند کارایی الگوریتم‌ها را نسبت به یکدیگر بسنجند. معیارهای کارایی مسئله چندهدفه در این تحقیق برای انجام مقایسات از ۵ معیار اندازه‌گیری کارایی استفاده شده که این معیارها عبارتند از:

۱. MID: فاصله از نقطه ایده‌آل؛^{۲۳}
۲. SNS: پراکندگی جواب‌های پارتو؛^{۲۴}
۳. PSN: درصد جواب‌های استفاده شده در مجموعه پارتوی حاصل از ادغام؛^{۲۵}
۴. ONVG: سایز مجموعه پارتو؛^{۲۶}
- MSS: یکنواختی جواب‌های پارتو.^{۲۷}

پیش از محاسبه معیارها، با توجه به مقیاس‌های مختلف، مقادیر تابع هدف برای مسائل نمونه متفاوت و لزوم میانگین‌گیری از مقادیر برای مقایسه نهایی، ابتدا مقادیر هدف نرمال‌سازی شده سپس مورد استفاده قرار گرفته‌اند. برای نرمال‌سازی، معادل انحراف نسبی درصدی (RPD) ^{۲۸} برای هر یک از مقادیر محاسبه شده است. در ادامه، رابطه مربوطه نشان داده شده است که در آن، Alg_{sol} مقدار تابع هدف به دست آمده و Min_{sol} بهتر (کمترین) مقدار به دست آمده برای آن تابع هدف در بین کلیه نقاط دو مجموعه پارتو است.

$$RPD = \frac{Alg_{sol} - Min_{sol}}{Min_{sol}} \times 100\% \quad (20)$$

معیار اول، فاصله از نقطه ایده‌آل، به صورت زیر تعریف می‌شود. n اندازه مجموعه پارتو و d_i فاصله جواب i از نقطه ایده‌آل (نقطه (۰٫۰) است $(d_i = \sqrt{f_{1i}^2 + f_{2i}^2})$. مقادیر کمتر این معیار، نزدیکی به نقطه ایده‌آل را نشان می‌دهد و از مطلوبیت بالاتری برخوردار است.

$$MID = \frac{\sum_{i=1}^n d_i}{n} \quad (21)$$

معیار دوم، پراکندگی جواب‌های پارتو، با استفاده از رابطه زیر محاسبه می‌گردد. این رابطه، پراکندگی فاصله جواب‌ها از نقطه ایده‌آل را محاسبه می‌کند. مقادیر بیشتر این معیار، نشان‌دهنده تنوع بیشتر جواب‌ها می‌باشد که برای تصمیم‌گیر یک مزیت تلقی می‌شود.

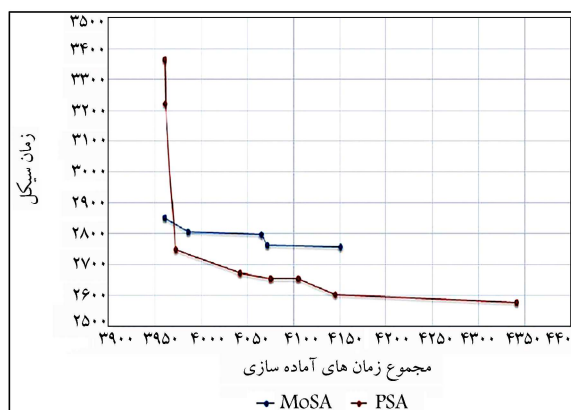
$$SNS = \sqrt{\frac{\sum_{i=1}^n (MID - d_i)^2}{n-1}} \quad (22)$$

جدول ۳. ارزیابی جواب‌های غیرمغلوب بدست آمده از الگوریتم‌ها با توجه به سائز مسئله بر حسب معیارهای مختلف.

مسئله	MID		SNS		PSN		MSS		ONVG			
	M	V	MoSA	PSA	MoSA	PSA	MoSA	PSA	MoSA	PSA		
LUTZ1	۸	H	۳۲,۱۴	۳۲,۱۴	۱,۸۵	۱,۸۵	۰	۰	۰,۱۳	۰,۱۳	۱۰	۹
		L	۱۵,۳۴	۱۵,۷۸	۰,۶۱	۰,۶۵	۰,۵	۰,۳۳	۰,۰۴	۰,۰۴	۱۲	۱۰
	۱۱	H	۳۸,۱۰	۳۸,۷۳	۱,۸۸	۱,۹۱	۰,۱۲	۰	۰,۰۵	۰,۰۶	۱۰	۶
		L	۱۳,۹۷	۱۳,۹۷	۰,۵۴	۰,۵۴	۰	۰	۰,۰۳	۰,۰۳	۹	۶
SAWYER	۸	H	۲۶,۶۹	۲۷,۹۴	۰,۷۲	۱,۲۱	۰,۶۷	۰	۰,۰۴	۰,۰۹	۴	۴
		L	۱۷,۶۱	۱۹,۱۶	۰,۸۹	۱	۰,۹	۰,۴۴	۰,۰۳	۰,۰۳	۸	۶
	۱۲	H	۳۱,۷۳	۳۰,۲۰	۲,۲۹	۱,۶۵	۰,۵۶	۰,۲۷	۰,۱۱	۰,۱۲	۸	۸
		L	۴۷,۶۵	۴۶,۸۱	۳,۲۲	۲,۸۷	۰,۸	۰,۳۳	۰,۱۴	۰,۰۸	۶	۶
ARC83	۷	H	۳,۳۰	۲,۷۱	۰,۱۹	۰,۱۴	۰,۶۷	۰,۵	۰	۰	۳	۲
		L	۱۱,۱۰	۶,۰۸	۰,۶۹	۰,۲۴	۰,۳۳	۰,۲۲	۰,۰۱	۰,۰۱	۶	۹
	۲۱	H	۷,۵۱	۱۰,۰۱	۰,۳۵	۰,۹۹	۰,۲	۰,۷۵	۰	۰,۰۶	۵	۸
		L	۳۱,۴۸	۳۰,۷۷	۱,۲۸	۱,۱۳	۰,۵۵	۰,۶۲	۰,۰۴	۰,۰۵	۱۱	۱۳
ARC111	۵	H	۵,۰۲	۷,۹۶	۰,۱۹	۱,۴۲	۱	۰,۲۵	۰,۰۲	۰,۰۴	۳	۸
		L	۶,۸۶	۶,۱۶	۰,۳۷	۱,۳۵	۰,۴	۰,۳۱	۰,۰۱	۰,۰۲	۱۰	۱۶
	۲۷	H	۱۹,۱۰	۱۸,۱۵	۰,۸	۲,۲۲	۱	۰,۵۶	۰,۰۵	۰,۰۶	۹	۱۸
		L	۴۶,۶۹	۴۶,۴۲	۱,۸۳	۴,۳۸	۰,۹۳	۰,۳۶	۰,۰۶	۰,۱۵	۱۰	۱۴
میانگین			۲۲,۱۴	۲۲,۰۶	۱,۱۱	۱,۴۷	۰,۵۴	۰,۳۱	۰,۰۵	۰,۰۶	۷,۷۵	۸,۹۴

ابتدا مدل ریاضی برای حالت دو هدفه تنظیم شد. اما مدل ریاضی از تولید جواب‌های بهینه پارتو ناکام ماند. در نتیجه، الگوریتم شبیه‌سازی تیرید پارتو برای حل مسئله به کار گرفته شد. برای اعتبارسنجی روش پیشنهادی، نیاز به مقایسه الگوریتم پیشنهادی با روشی جدید در پژوهش‌های پیشین بود. به این منظور، پس از تولید مسائل نمونه، روش پیشنهادی در حالت چند هدفه با روشی دیگر مبتنی بر الگوریتم شبیه‌سازی تیرید (که حالت میانه الگوریتم‌های تک‌هدفه و پارتو می‌باشد و MoSA نام‌گذاری شد) مقایسه شد. برای مقایسه دو الگوریتم، نیاز به معیار عملکردی برای اندازه‌گیری کارایی الگوریتم‌ها می‌باشد. در نتیجه، چند معیار جدید پیشنهاد شد و پس از مقایسه الگوریتم‌ها با استفاده از این پنج معیار، این نتیجه حاصل شد که روش پیشنهادی پژوهش حاضر دارای کارایی مناسب و قابل قبولی است.

با وجود لحاظ کردن زمان‌های آماده‌سازی وابسته به توالی در مسائل بالانس خط مونتاژ، هنوز هم بین شرایط واقعی خط مونتاژ و مسئله مطالعه شده در این تحقیق شکاف بزرگی وجود دارد. بنابراین، می‌توان علاوه بر زمان‌های آماده‌سازی، موارد دیگری همچون خطوط U شکل یا دوسویه، موازی‌سازی، زمان‌های تصادفی و غیرقطعی و روش‌های کاری جایگزین را به این مسئله اضافه و مسئله بررسی شده را به شرایط کنونی صنعت نزدیک‌تر کرد. توسعه روش‌های قطعی مانند شاخه و حد، شامل روشی ابتکاری برای حذف



شکل ۸. مقایسه مجموعه جواب‌های بدست آمده از الگوریتم‌ها.

۷. نتیجه‌گیری و تحقیقات آتی

در این تحقیق، از روشی فراابتکاری برای حل مسائل بالانس خط مونتاژ عمومی چندهدفه با زمان‌های آماده‌سازی وابسته به توالی استفاده شده است. مسئله با هدف کمینه‌سازی همزمان زمان سیکل و زمان آماده‌سازی کل خط تعریف شد.

P : مجموعه زوج‌های (i, k) که در آن فعالیت i بلافاصله ماقبل فعالیت k در گراف بیاید؛
 PT_i : مجموعه کلیه فعالیت‌های ماقبل فعالیت i در گراف؛
 Nm_j : بیشترین تعداد فعالیت‌هایی که می‌تواند به ایستگاه j اختصاص یابد؛
 NT_m : بیشترین تعداد فعالیت‌هایی که می‌تواند به هر ایستگاهی اختصاص یابد $NT_m = \max_j \{Nm_j\}$ ؛
 tsu_{ik} : زمان آماده‌سازی هنگامی که فعالیت k دقیقاً پس از فعالیت i در یک ایستگاه انجام گیرد؛
 x_{ijs} : اگر فعالیت i به ایستگاه j در مکان s اختصاص می‌یابد مساوی با ۱، در غیر این صورت صفر $(i = 1, \dots, N; j = 1, \dots, m; s = 1, \dots, Nm_j)$ ؛
 z_{ikj} : اگر فعالیت i بلافاصله پیش از فعالیت k در ایستگاه j انجام گیرد مساوی با ۱، در غیر این صورت صفر $(\forall j, \forall (i, k) | (i \neq k))$ ؛
 w_{ij} : اگر i آخرین فعالیت اختصاص یافته به ایستگاه j باشد مساوی با ۱، در غیر این صورت صفر $(\forall i; \forall j)$.

گروه‌بندی‌های نامطلوب با فرض کمترین زمان سیکل برای یک ترتیب فعالیت‌ها، به‌عنوان یکی دیگر از موارد بالقوه برای مطالعات آتی مطرح است. استفاده از اهداف دیگر رایج مانند کاهش هزینه‌ها، افزایش هموارسازی فعالیت‌ها و مربوط بودن فعالیت‌ها به‌عنوان اهداف دیگر در کنار هدف‌های بررسی شده در این تحقیق پیشنهاد می‌شود.

نمادها

J : ایستگاه کاری؛
 S : مکان یک فعالیت در داخل ایستگاه؛
 N : مجموعه فعالیت‌ها $(i=1, \dots, N)$ ؛
 M : تعداد ایستگاه‌ها؛
 t_i : زمان پردازش فعالیت i ؛
 Ct : زمان سیکل؛

پانویس‌ها

1. Assembly Line Balancing Problems (ALBP)
2. Simple Assembly Line Balancing Problems (SALBP)
3. General Assembly Line Balancing Problems (GALBP)
4. General Assembly Line Balancing Problem With Setups (GALBPS)
5. Two-sided
6. Ant Colony Optimization (ACO)
7. Differential Evolution (DE)
8. Goal Programming
9. Production Scheduling
10. Greedy Randomized Adaptive Search Procedure
11. WWW.Assembly-Line-Balancing.De
12. Pareto Simulated Annealing (PSA)
13. Bin Packing Problem (BPP)
14. Weighted Tchebycheff Scalarizing Function
15. Agents
16. Degree of Repulsion
17. Feasible
18. Single Point Operator
19. Swap Operator
20. Multi Single Point Operator
21. Number of Neighborhood Search in Each Temperature (NNS)
22. Multi-Objective SA (MoSA)
23. Mean Ideal Distance
24. Spread of Non-Dominated Solutions
25. Percentage of Solutions Used In the Overall Non-Dominated Set
26. Overall Non-Dominated Vector Generation
27. Modified Schott's Spacing Metric
28. Relative Percentage Deviation

منابع (References)

1. Baybars, I., 1986. A survey of exact algorithms for the simple assembly line balancing problem. *Management Science*, 32(8), pp.909-932. <https://doi.org/10.1287/mnsc.32.8.909>.
2. Becker, C. and Scholl, A., 2006. A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 168, pp.694-715. <https://doi.org/10.1016/j.ejor.2004.07.023>.
3. Andres, C., Miralles, C. and Pastor, R., 2008. Balancing and scheduling tasks in assembly lines with sequence-dependent setup times. *European Journal of Operational Research*, 187, pp.1212-1223. <https://dx.doi.org/10.1016/j.ejor.2006.07.044>.
4. Seyed-Alagheband, S.A., Fatemi Ghomi, S.M.T. and Zandieh, M., 2011. A simulated annealing algorithm for balancing the assembly line type II problem with sequence-dependent setup times between tasks. *International Journal of Production Research*, 49(3), pp.805-825. <https://doi.org/10.1080/00207540903471486>.
5. Allahverdi, A. and Soroush, H.M., 2008. The significance of reducing setup times/setup costs. *European Journal of Operational Research*, 187, pp.978-984. <https://doi.org/10.1016/j.ejor.2006.09.010>.
6. Salveson, M.E., 1955. The assembly line balancing problem. *Journal of Industrial Engineering*, 6(3), pp.18-25. <https://doi.org/10.1115/1.4014559>.

7. Scholl, A. and Becker, C., 2006. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168, pp.666-693. <https://doi.org/10.1016/j.ejor.2004.07.022>.
8. Sivasankaran, P. and Shahabudeen, P., 2014. Literature review of assembly line balancing problems. *International Journal of Advanced Manufacturing Technology*, 73, pp.1665-1694. <https://doi.org/10.1007/s00170-014-5944-y>.
9. Boysen, N., Fliedner, M. and Scholl, A., 2007. A classification of assembly line balancing problems. *European Journal of Operational Research*, 183, pp.674-693. <https://doi.org/10.1016/j.ejor.2006.10.010>.
10. Chen, R.S., Lu, K.Y. and Yu, S.C., 2002. A hybrid genetic algorithm approach on multi-objective of assembly planning problem. *Engineering Applications of Artificial Intelligence*, 15, pp.447-457. [https://doi.org/10.1016/S0952-1976\(02\)00073-8](https://doi.org/10.1016/S0952-1976(02)00073-8).
11. Babazadeh, H. and Javadian, N., 2019. A novel meta-heuristic approach to solve fuzzy multi-objective straight and U-shaped assembly line balancing problems. *Soft Computing*, 23, pp.8217-8245. <https://doi.org/10.1007/s00500-018-3457-6>.
12. Saif, U., Guan, Z., Liu, W., Zhang, C. and Wang, B., 2014. Pareto based artificial bee colony algorithm for multi objective single model assembly line balancing with uncertain task times. *Computers & Industrial Engineering*, 76, pp.1-15. <https://doi.org/10.1016/j.cie.2014.07.009>.
13. Nearchou, A.C., 2008. Multi-objective balancing of assembly lines by population heuristics. *International Journal of Production Research*, 46(8), pp.2275-2297. <https://doi.org/10.1080/00207540600988089>.
14. Baykasoglu, A., 2006. Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems. *Journal of Intelligent Manufacturing*, 17, pp.217-232. <https://doi.org/10.1007/s10845-005-6638-y>.
15. Allahverdi, A., 2015. The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research*, 246(2), pp.345-378. <https://doi.org/10.1016/j.ejor.2015.04.004>.
16. Martino, L. and Pastor, R., 2010. Heuristic procedures for solving the general assembly line balancing problem with setups. *International Journal of Production Research*, 48(6), pp.1787-1804. <https://doi.org/10.1080/00207540802577979>.
17. Scholl, A., Boysen, N. and Fliedner, M., 2013. The assembly line balancing and scheduling problem with sequence-dependent setup times: Problem extension, model formulation and efficient heuristics. *OR Spectrum*, 35(1), pp.291-320. <https://doi.org/10.1007/s00291-011-0265-0>.
18. Esmailbeigi, R., Naderi, B. and Charkhgard, P., 2015. The type E simple assembly line balancing problem: A mixed integer linear programming formulation. *Computers & Operations Research*, 64, pp.168-177. <https://doi.org/10.1016/j.cor.2015.05.017>.
19. Delice, Y., Aydogan, E.K., Soylemez, İ. and Ozcan, U., 2018. An ant colony optimisation algorithm for balancing two-sided U-type assembly lines with sequence-dependent set-up times. *Sādhanā*, 43(12), pp.1-15. <http://dx.doi.org/10.1007%2Fs12046-018-0969-9>.
20. Wu, C.C. and Lee, W.C., 2006. Single machine scheduling to minimize the setup time and the earliness. *Journal of Information & Optimization Sciences*, 27(2), pp.499-510. <http://dx.doi.org/10.1080/02522667.2006.10699705>.
21. Garey, M.R. and Johnson, D.S., 1979. *Computers and Intractability: a Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Company. ISBN 0-7167-1045-5.
22. Czyzak, P. and Jaskiewicz, A., 1998. Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7, pp.34-47. [https://doi.org/10.1002/\(SICI\)1099-1360\(199801\)7:1%3C34::AID-MCDA161%3E3.0.CO;2-6](https://doi.org/10.1002/(SICI)1099-1360(199801)7:1%3C34::AID-MCDA161%3E3.0.CO;2-6).
23. Nearchou, A.C., 2007. Balancing large assembly lines by a new heuristic based on differential evolution method. *International Journal of Advanced Manufacturing Technology*, 34, pp.1016-1029. <https://doi.org/10.1007/s00170-006-0655-7>.
24. Şahin, R. and Türkbey, O., 2009. A simulated annealing algorithm to find approximate pareto optimal solutions for the multi-objective facility layout problem. *International Journal of Advanced Manufacturing Technology*, 41, pp.1003-1018. <https://doi.org/10.1007/s00170-008-1530-5>.