نشریه پژوهش‌های مهندسی صنایع در سیستم‌های تولید

ISSN: 2345-2269

نشریه پـــژوهـــش‌های مهنـــدســـی صنـــایع در سیستـــم‌هـــای تولیـــد

سال پنجم، شماره دهم، بهار و تابستان ۱۳۹۶، صفحـــــه ۱۷–۲۹

www.ier.basu.ac.ir

DOI: 10.22084/ier.2017.11693.1534

حل مسأله زمان‌بندی چندعاملی در محیط جریان کارگاهی با در نظر گرفتن اثر زمانی و رد کردن کارها با استفاده از یک الگوریتم فراابتکاری

مجید حسین‌زاده[۱]، راشد صحرائیان[۲،*]

۱. کارشناس ارشد مهندسی صنایع، گروه مهندسی صنایع، دانشگاه شاهد، تهران، ایران.

۲. دانشیار گروه مهندسی صنایع، دانشکده فنی و مهندسی، دانشگاه شاهد، تهران، ایران.

اطلاعات مقاله

تاریخچه مقاله:

دریافت ۱۳۹۵/۰۷/۰۴

پذیرش ۱۳۹۶/۰۳/۲۸

کلمات کلیدی:

زمان‌بندی

عامل اثر زمانی

رد کردن

مدل برنامه‌ریزی عدد صحیح مختلط

خلاصه

در این پژوهش یک مسأله زمان‌بندی چندعاملی، در محیط جریان کارگاهی مورد بررسی قرار گرفته است. مسأله زمان‌بندی چندعاملی، زیرمجموعه‌ای از مسائل زمان‌بندی چند هدفه است که در آن هر عامل، دارای مجموعه‌ای از کارها است و هدف آن، بهینه کردن تابع هدف مربوط به خود است. جهت واقعی‌تر کردن مسأله، دو مفروض کاربردی «اثر زمانی» و «رد کردن» در نظر گرفته شده است. یک مدل برنامه‌ریزی عدد صحیح مختلط برای مسأله ارائه شده است. همچنین با توجه به پیچیدگی مدل و عدم توانایی روش‌های حل دقیق در حل مسائل با ابعاد بزرگ، الگوریتم فراابتکاری ژنتیک مبتنی بر مرتب‌سازی نامغلوب پیشنهاد شده است. راه‌حل‌های حاصل از این الگوریتم و روش دقیق محدودیت جزئی تعمیم‌یافته، با هم مقایسه شده است و نتایج به دست آمده، عملکرد آن را تأیید می‌نماید.

۱– مقدمه

زمان‌بندی نوعی فرآیند تصمیم‌گیری است که با هدف بهینه‌سازی یک و یا چند هدف انجام می‌گیرد و در بازار رقابتی کنونی به یک نیاز اساسی برای بقای کارخانجات و محیط‌های تولیدی تبدیل شده است [۱]. در مسائل زمان‌بندی کاربردی، ممکن است انواع متفاوتی از مشتریان وجود داشته باشد. برخی از مشتریان، نمی‌توانند تأخیر را تحمل کنند؛ به همین دلیل ممکن است هزینه بیشتری را برای تحویل به‌موقع کالای خود بپردازند. اما برخی مشتریان، به هزینه بیشتر از زمان اهمیت می‌دهند؛ بنابراین یک زمان‌بندی خوب باید به طور همزمان، تقاضاهای مختلف انواع مشتریان را برآورده سازد. در مسائل زمان‌بندی کلاسیک، اغلب فرض می‌شود که همه کارها متعلق به یک منبع (مشتری) هستند و هدف مسأله، پیدا کردن یک زمان‌بندی و توالی از کارها به منظور بهینه کردن توابع هدف موردنظر است. دسته‌ای از مسائل زمان‌بندی نیز وجود دارد که در آن‌ها کارها منبع یکسانی ندارند. به هر یک از این مشتری‌ها یک عامل[۱] و به مسأله مورد نظر، زمان‌بندی چندعاملی گفته می‌شود.

در مسأله زمان‌بندی چندعاملی، هر عامل دارای مجموعه‌ای از کارها است و می‌خواهد تابع هدف مربوط به خود را با توجه به این کارها بهینه نماید. هدف این مسأله، یافتن توالی از کارهای همه عامل‌ها و به دست آوردن یک راه‌حل تعادلی[۲] بین اهداف عامل‌ها است. ضرورتی که در پژوهش مسأله زمان‌بندی چندعاملی وجود دارد، این است که وجود فقط یک یا دو مشتری (عامل) در مسأله نمی‌تواند پاسخگوی مسائل دنیای واقعی باشد، زیرا در این مسائل حالت‌هایی وجود دارد که تولیدکننده با چندین مشتری مختلف با اهداف گوناگون سروکار دارد و باید اهمیت همه آن‌ها را یکسان در نظر بگیرد. مسأله زمان‌بندی با عامل، زیرشاخه‌ای از زمان‌بندی چند

* نویسنده مسئول: راشد صحرائیان

تلفن: ۵۱۲۱۲۰۴۰-۰۲۱؛ پست الکترونیکی: Sahraeian@shahed.ac.ir

1. Agent
2. Trade-off

هدفه است و نقش فراوانی در مسائل زمان‌بندی واقعی و آکادمیک ایفا می‌کند. از کاربردهای این زمان‌بندی می‌توان به صنعت حمل‌ونقل ریلی، هوایی و سازمان‌های پروژه محور اشاره داشت [۲].

مسأله‌ای که در این پژوهش مورد بررسی قرار خواهد گرفت، مسأله زمان‌بندی چندعاملی در محیط جریان کارگاهی[1] نام دارد. برای واقعی‌تر کردن مدل مسأله، دو مفروض کاربردی اثر زمانی[2] و رد کردن[3] کارها به آن افزوده شده است. منظور از اثر زمانی این است که زمان پردازش کارها، به زمان شروع آن‌ها روی ماشین وابسته است و هر چه کار دیرتر شروع شود، زمان انجام آن به‌صورت خطی افزایش می‌یابد. صنایع نورد فولاد، فعالیت‌های تعمیر و نگهداری و فعالیت‌های اورژانسی از جمله کاربردهای اثر زمانی در دنیای واقعی است [۳]. همچنین زمانی که مفروض رد کردن کارها در مسأله وجود دارد، به این معنی است که تصمیم‌گیرنده می‌تواند بنا به دلایلی مانند افزایش سود، برخی از کارهایی را که ایجاد تأخیر می‌کنند، پردازش نکند و به جای آن هزینه‌ای تحت عنوان هزینه رد کردن بپردازد. یکی از کاربردهای مهم رد کردن، در واحدهای صنعتی دارای برون‌سپاری است. طبق ادبیات، برون‌سپاری می‌تواند باعث کاهش هزینه‌های کنترل عملیات، آزاد کردن منابع داخلی برای اهداف دیگر و کاهش ریسک شود [۴]. بر اساس دانش ما، مسأله زمان‌بندی چندعاملی با در نظر گرفتن اثر زمانی و رد کردن مورد پژوهش قرار نگرفته است. با توجه به کاربردهای ذکر شده برای دو مفروض فوق و مسأله زمان‌بندی با عامل، ترکیب این مدل‌ها منطقی و انگیزه‌بخش برای ما بوده است. صنایع ظروف چینی که با مشتری‌های متفاوتی سر و کار دارند، مثالی واقعی از مدل مورد بررسی در این پژوهش است. در این صنایع، جهت ساخت ظروف، از خاک رس استفاده می‌شود. سفت شدن این خاک با گذشت زمان، موجب افزایش زمان پردازش ظروف تولیدی می‌شود. همچنین به دلیل پایین بودن ظرفیت تولیدی، برنامه‌ریز می‌تواند تعدادی از سفارشات شرکت را به‌صورت برون‌سپاری و با صرف هزینه‌ای به کارخانجات مشابه دیگر واگذار کند.

مسأله زمان‌بندی با عامل اولین بار توسط بیکر و اسمیت [۵] و آگنتیس و همکاران [۶] معرفی شده و پس از آن این مسأله به یک موضوع جذاب برای پژوهشگران تبدیل شده است [۷]. چوی [۸] مسأله زمان‌بندی دو عاملی روی یک ماشین را ارائه داده است که هدف آن، حداقل کردن جمع وزنی زمان اتمام کارها برای هر دو عامل و محدودیت حداقل کردن زمان اتمام همه کارها برای عامل دوم است؛ همچنین تابع هدف عامل دوم باید از یک مقدار از پیش تعریف شده‌ای کمتر باشد. در این مقاله، همچنین یک محدودیت در نظر گرفته شده است که زمان پردازش واقعی کارها با توجه به جایگاهشان در توالی افزایش می‌یابد که آن را اثر فرسودگی نامیده است. نویسنده ابتدا از الگوریتم شاخه و کران (B&B) برای تولید راه‌حل بهینه استفاده کرده است، اما چون B&B برای مسائل با اندازه بزرگ زمان‌بر است، الگوریتم شبیه‌سازی تبریدی (SA) را پیشنهاد داده است. طی مقایسه‌ای که بین جواب حاصل از این دو الگوریتم صورت گرفته، مشاهده شده است که جواب الگوریتم SA نزدیک به جواب بهینه و با زمانی بسیار سریع‌تر از B&B است. چنگ [۹] نیز چندین مسأله زمان‌بندی دو هدفه روی تک ماشین را بررسی کرده است. هدف مسأله عبارت است از: حداقل کردن تابع هدف عامل اول با این فرض که تابع هدف عامل دوم کمتر از یک مقدار Q باشد. توابع هدف مسائل مذکور شامل حداکثر زود کرد، جمع وزنی حداکثر زود کرد و جمع وزنی حداکثر زود کرد و دیرکرد (تأخیر) است.

پژوهش‌هایی که ذکر شد، همگی در محیط تک ماشینی انجام شده است. لی [۱۰] در مقاله خود مسأله زمان‌بندی در محیط جریان کارگاهی با دو عامل را مورد مطالعه قرار داده است. تابع هدف عامل اول، حداقل کردن زمان اتمام آخرین کار و تابع هدف عامل دوم حداقل کردن مجموع تأخیرها است. هدف مسأله پیدا کردن توالی‌های غیر مسلط از کل کارها با در نظر گرفتن هر دو تابع هدف است. برای حل این مسأله یک الگوریتم جستجوی همسایگی متغیر (VNS) با روابط همسایگی مختلف ارائه شده است. جواب‌های غیر مسلط تولیدشده توسط این الگوریتم با دو الگوریتم ممتیک[4] و NSGA-II مقایسه شده است و مشاهده می‌شود که این الگوریتم، جواب‌های خوبی ایجاد می‌کند. لی و همکاران [۱۱] یک مسأله زمان‌بندی جریان کارگاهی با دو عامل را در نظر گرفته‌اند که هدف عامل اول حداقل کردن مجموع زمان اتمام کارها و هدف عامل دوم صفر بودن تعداد کارهای به تأخیر افتاده است. در این مقاله از الگوریتم B&B و SA برای به دست آوردن جواب بهینه استفاده شده است.

آنچه که لازم به ذکر است، تعداد مطالعاتی که در زمینه زمان‌بندی با بیشتر از دو عامل انجام شده، بسیار کم است و از این تعداد می‌توان به مقاله شیائو و همکاران [۱۲] اشاره نمود. در این پژوهش یک مسأله زمان‌بندی تک ماشینی سه عاملی با زمان آزادسازی و فعالیت تعمیر و نگهداری ارائه شده است. هدف عامل اول حداقل کردن زمان اتمام کل کارها است؛ حداکثر تأخیر کارهای عامل دوم نباید از یک مقدار ثابت و محدود بیشتر شود و عامل سوم نیز می‌خواهد فعالیت‌های تعمیر و نگهداری در یک فاصله زمانی مشخص که به آن پنجره تعمیر و نگهداری گفته می‌شود، انجام شود. یک حد پایین برای شتاب دادن به الگوریتم شاخه و کران و جلوگیری از ایجاد شاخه‌های غیرضروری پیشنهاد شده است و نتایج محاسباتی مقاله نشان می‌دهد که این حد پایین عملکرد قابل قبولی داشته است. همچنین نویسندگان این مقاله از الگوریتم ژنتیک برای به دست آوردن جواب اولیه‌ی (نزدیک به بهینه) الگوریتم شاخه و کران استفاده کرده‌اند.

1. Flow Shop
2. Deteriorating Jobs
3. Rejection
4. Memetic Algorithm

کاناتور و گوپتا [۱۳]، جز اولین محققینی هستند که در زمینه زمان پردازش وابسته به زمان، کار کرده‌اند. آن‌ها در مقاله خود مسأله حداقل سازی زمان انجام آخرین کار با محدودیت اثر زمانی روی تک ماشین را بررسی کرده‌اند. در این مقاله زمان پردازش کار به‌صورت زیر تعریف شده است:

$$y_i = p_i + b_i \qquad (۱)$$

که در آن، y_i زمان واقعی پردازش، p_i زمان نرمال پردازش و b_i جریمه کار i است و به‌صورت زیر محاسبه می‌شود.

$$b_i = \max\{0, v_i(s_i - d_i)\} \qquad (۲)$$

در رابطه (۲)، v_i نرخ اثر زمانی است و به شرایط محیط، ماشین‌آلات و... بستگی دارد. همچنین d_i دیرترین زمان شروع کار i است و اگر کار در زمان d_i شروع نشود، زمان پردازش آن با نرخ v_i افزایش می‌یابد که s_i نامیده می‌شود. در اکثر مطالعات انجام شده در ادبیات مسأله زمان‌بندی با در نظر گرفتن اثر زمانی، فرض شده است که زمان پردازش واقعی یک کار، تابعی از زمان شروع آن است، اما در پژوهش چنگ و همکاران [۱۴] یک مدل جدید از اثر زمانی در نظر گرفته شده است که در آن زمان پردازش واقعی یک کار تابعی از زمان پردازش کارهایی است که قبل از آن پردازش شده‌اند و از رابطه (۳) محاسبه می‌شود.

$$p_{j[r]} = p_j\left(1 - \frac{\sum_{l=1}^{r-1} p_{[l]}}{\sum_{l=1}^{n} p_l}\right)^a = p_j\left(\frac{\sum_{l=r}^{n} p_{[l]}}{\sum_{l=1}^{n} p_l}\right)^a \qquad (۳)$$

که در آن، p_j زمان پردازش نرمال کار j، $p_{[l]}$ زمان پردازش نرمال کار قرار گرفته در موقعیت lام توالی و a شاخص اثر زمانی و یک مقدار نامثبت ($a \le 0$) است. دراین پژوهش ثابت شده است که مسأله زمان‌بندی تک ماشینی با تابع هدف حداقل کردن زمان اتمام آخرین کار و مجموع زمان اتمام کارها، با در نظر گرفتن مدل اثر زمانی ذکر شده نیز در زمان جند جمله‌ای قابل حل است. مسأله زمان‌بندی چند عاملی روی تک ماشین با در نظر گرفتن اثر زمانی در مقاله وو و همکاران [۳] مورد بررسی قرار گرفته است. تابع هدف عامل اول، حداقل کردن حداکثر تأخیر وزنی و تابع هدف عامل دوم جمع وزنی مجموع تأخیرها است. برای حل این مسأله الگوریتم دقیق شاخه و کران ارائه شده است و برای کم کردن تکرارهای این الگوریتم و به دست آوردن جواب اولیه آن (حد بالا) یا همان جواب نزدیک به بهینه از الگوریتم‌های فراابتکاری مورچگان و شبیه‌سازی تبرید استفاده شده است. در پژوهش یین و همکاران [۱۵] ترکیب‌های مختلفی از اهداف عامل‌ها در مسأله زمان‌بندی چند عاملی با در نظر گرفتن اثر زمانی مورد پژوهش قرار گرفته است. برای هر کدام از مسائل طرح شده، پیچیدگی محاسباتی و قابلیت حل آن برای پیدا کردن جواب بهینه عامل اول تحت این شرط که حداکثر زمان زودکرد عامل دوم از یک حد بالا کمتر باشد، بررسی شده است. زمان پردازش واقعی کار j مربوط به عامل X از رابطه (۴) به دست می‌آید.

$$p_j^X = a_j^X(a + bt) \qquad (۴)$$

که در آن، p_j^X و a_j^X به ترتیب زمان پردازش واقعی و نرمال کار jام عامل X و a و b هر دو ثابت هستند ($a > 0$ و $b \ge 0$). لی و همکاران [۱۶] نیز مفروض اثر زمانی را در مسأله زمان‌بندی چند عاملی مورد بررسی قرار داده اند. زمان پردازش واقعی کارها از رابطه (۵) محاسبه می‌شود.

$$p_j = \alpha_j + \beta t \qquad (۵)$$

که در آن، α_j زمان پردازش نرمال کار j ، t زمان شروع آن کار و β نرخ اثر زمانی است. در این پژوهش تابع هدف عامل اول، حداقل کردن جمع وزنی زمان اتمام کارها با مجاز نبودن هیچ کار به تاخیر افتاده برای عامل دوم است. همچنین یک الگوریتم شاخه و کران و سه الگوریتم ابتکاری برای پیدا کردن جواب بهینه و نزدیک به بهینه پیشنهاد شده است.

مفهوم رد کردن اولین بار توسط بارتال و همکاران [۱۷] در سال ۲۰۰۰ ارائه شد. هدف این مقاله حداقل سازی زمان اتمام آخرین کار و جریمه پرداختی به دلیل رد کردن بعضی از کارها، روی ماشین‌های موازی یکسان است. شبتای و همکاران [۴] نیز مسأله زمان‌بندی حداقل کردن زمان اتمام آخرین کار (F_1) و جریمه پرداختی برای کارهای رد شده (F_2) بر روی تک ماشین را بررسی کرده‌اند ($\min(F_1 + F_2)$) و به دنبال نقاط بهینه پارتو برای تابع هدف مربوطه می‌باشند. در مقاله فنگ و همکاران [۱۸] چندین مسأله زمان‌بندی دو عاملی با مفروض رد کردن روی تک ماشین بررسی شده است. برای عامل اول، هدف مسأله، حداقل کردن یک تابع هدف مشخص f، مربوط به کارهای پذیرفته‌شده به علاوه کل هزینه پرداخت شده برای کارهای رد شده است. همچنن تابع هدف عامل دوم، نباید از مقدار Q بیشتر باشد.

پژوهش‌هایی نیز وجود دارد که دو مفروض رد کردن و اثر زمانی را همزمان در مسأله تک عاملی در نظر گرفته‌اند. پژوهش لی و یوآن [۱۹] از این قبیل مسائل است. در این مقاله هدف مسأله، حداقل کردن زمان اتمام آخرین کار و جمع وزنی زمان اتمام کارها به همراه حداقل کردن جریمه پرداختی برای کارهای رد شده، است.

همانطور که در مقالات مروری [۵] و [۱۱] مشاهده شد، مقالاتی که در زمینه چند عاملی کار شده، به علت سادگی، یا در محیط تک ماشینی، یا دو هدفه و یا بدون در نظر گرفتن مفروضات واقعی از قبیل رد کردن کارها، اثر زمانی، زمان آماده‌سازی و... انجام شده است. در حالی که این مسأله، در محیط چند ماشینی بسیار کم مورد توجه قرار گرفته و طی مطالعات انجام شده این نتیجه حاصل شد که مسأله زمان‌بندی بیشتر از دو عامل در محیط چند ماشینی و با در نظر گرفتن دو مفروض تعریف شده، مورد پژوهش قرار نگرفته است. همچنین در بیشتر موارد، تابع هدف مربوط به یک عامل یا هر دو عامل محدود شده است. به بیان بیشتر، هدف این مسائل، عبارت است از حداقل کردن تابع هدف یکی از عامل‌ها

با این شرط که تابع هدف عامل دیگر از یک حد بالا تجاوز نکند که این شرط تا حدودی از واقعیت به دور است. حالت واقعی‌تر این مسأله به گونه‌ای است که همه عامل‌ها از اهمیت یکسانی برای تصمیم‌گیرنده برخوردار باشند و حداقل کردن تابع هدف فقط یک عامل خیلی منصفانه و جامع نیست؛ بنابراین در این پژوهش، مسأله‌ای بررسی می‌شود که هدف آن به دست آوردن راه‌حلی است که در آن اهمیت همه عامل‌ها در نظر گرفته شده است. لازم به ذکر است که در ادبیات این مسأله، الگوریتم زمان چندجمله‌ای و روش‌های برنامه‌ریزی ریاضی، رویکردهای اصلی حل مسأله زمان‌بندی با عامل هستند و در موارد کمی، از برخی از الگوریتم‌های فراابتکاری مانند VNS، TNS[1] و SA استفاده شده است. همچنین مدل‌سازی خوبی برای این مسائل صورت نگرفته است که این نقیصه در پژوهش حاضر نیز برطرف شده است.

با توجه به مطالب گفته شده، نوآوری اصلی انجام شده در این پژوهش، در نظر گرفتن محیط جریان کارگاهی با چند عامل، اضافه کردن دو مفروض کاربردی اثر زمانی و رد کردن کارها و مدل‌سازی آن است. همچنین یک الگوریتم NSGA-II برای حل مسأله در ابعاد بالا ارائه شده است.

۲- شرح مسأله

مسأله زمان‌بندی محیط جریان کارگاهی با چند عامل، متشکل از n کار $J_1, J_2, \ldots, J_n$ و m ماشین $M_1, M_2, \ldots, M_m$ است و همه کارها برای پردازش شدن، باید یک مسیر یکسان را روی این ماشین‌ها طی کنند. هر ماشین در یک زمان خاص فقط یک کار را می‌تواند پردازش کند. در یک مسأله سه عاملی، کارها یا متعلق به عامل اول، یا عامل دوم و یا عامل سوم هستند. برای سهولت و درک بهتر راه‌حل‌ها، این‌طور فرض می‌شود که n_1 کار اول، متعلق به عامل اول است، n_2 کار دوم، کارهای عامل دوم و $n - n_1 - n_2$ کار بعدی هم متعلق به آخرین عامل است. پر واضح است، یک برنامه‌ریزی خوب برای مسأله مورد بررسی باید کارهای همه عامل‌ها را مورد توجه قرار دهد و چون این کارها بر هم تأثیرگذار هستند، نمی‌توان کارهای هر عامل را به صورت مستقل برنامه‌ریزی کرد. لازم به ذکر است که در این مسأله، تصمیم‌گیرنده می‌تواند، کارها را با پرداخت هزینه مربوط به آن‌ها، رد کند. همچنین زمان پردازش واقعی کارها در این پژوهش، از رابطه زیر به دست می‌آید [۲۰]:

$$p_{ji} = a_{ji} + bt_{ji} \qquad (۶)$$

که در آن p_{ji}، a_{ji} و t_{ji} به ترتیب زمان پردازش واقعی، زمان پردازش نرمال و زمان شروع کار j روی ماشین i، و b نرخ اثر زمانی است.

۲-۱- مدل ریاضی

در این بخش، مدل برنامه‌ریزی عدد صحیح مختلط پیشنهادی تشریح خواهد شد. نوآوری‌های صورت گرفته در این مدل، اضافه کردن مسأله چند عاملی، اثر زمانی و رد کردن کارها به مدل‌های کلاسیک جریان کارگاهی است. قبل از ارائه مدل، ابتدا لازم است که مفروضات و محدودیت‌هایی که در آن لحاظ شده است، ذکر شود:

۱- توالی کارها روی ماشین‌های مختلف، یکسان است.

۲- تا زمانی که پردازش یک کار تمام نشده است، نمی‌توان آن را قطع کرد.

۳- در هر زمان، هر ماشین می‌تواند فقط یک کار را پردازش کند و یک کار نمی‌تواند همزمان روی دو ماشین انجام شود.

۴- تا زمانی که پردازش عملیات یک کار روی ماشین قبلی تمام نشود، عملیات بعدی آن نمی‌تواند شروع شود.

۵- برای تعریف بهتر مدل، یک کار مجازی تعریف می‌شود که زمان پردازش آن روی همه ماشین‌ها صفر است و حتما دارای یک کار پسنیازی خواهد بود.

۶- برای زیباتر شدن مدل، کارهای عامل دوم قابل رد شدن نیستند.

پارامترها و متغیرهای تصمیم مدل نیز به صورت زیر تعریف می‌شوند:

Q	تعداد عامل‌ها
m	تعداد کل ماشین‌ها
n	تعداد کل کارها
q	زیروند عامل‌ها: $q = \{1, \ldots, Q\}$
j, k	زیروند کارها: $j = \{1,2, \ldots, n\}$ و $k = \{1,2, \ldots, n\}$
i	زیروند ماشین‌ها: $i = \{1,2, \ldots, m\}$
AG_q	تعداد کارهای عامل q
a_{ji}	زمان پردازش نرمال کار j روی ماشین i
tc	هزینه هر واحد تأخیر
uc	هزینه هر واحد کار به تأخیر افتاده
M	یک عدد مثبت بزرگ
ρ	یک عدد کوچک
d_j	زمان تحویل کار j
e_j	هزینه رد کردن کار j
b	نرخ اثر زمانی
f_q	تابع هدف عامل q
p_{ji}	متغیر پیوسته زمان واقعی پردازش کار j روی ماشین i
t_{ji}	متغیر پیوسته زمان شروع کار j روی ماشین i
C_{ji}	متغیر پیوسته زمان اتمام کار j روی ماشین i
T_j	متغیر پیوسته مدت زمان تأخیر کار j
C_{max}	متغیر پیوسته زمان اتمام آخرین کار $(C_{max} = \max(C_{jm}))$
U_j	متغیر صفر و یک که مقدار آن یک است، اگر کار j تأخیر داشته باشد و در غیر اینصورت صفر است. $U_j = \begin{cases} 1 & C_{jm} > d_j \\ 0 & o.w \end{cases}$
y_j	متغیر صفر و یک که مقدار آن یک است، اگر کار j رد شود و در غیر اینصورت صفر است.

1. Two-phase Neighborhood Search Algorithm

X_{kj} متغیر صفر و یک که مقدار آن یک است، اگر کار j بلافاصله بعد از کار k پردازش شود و در غیر اینصورت صفر است.

درنهایت مدل مسأله به صورت زیر نوشته می‌شود:

$$\min(f_1, f_2, f_3)$$

$$f_1 = \sum_{j\in AG_1} tc * T_j + (1 - y_j) * e_j$$

$$f_2 = C_{max\,j\in AG_2} \quad (۷)$$

$$f_3 = \sum_{\in AG_3} uc * U_j + (1 - y_j) * e_j$$

$s.t$

$$\sum_{k=0, j\neq k}^{n} X_{kj} = y_j \qquad \forall j \quad (۸)$$

$$\sum_{j=1, j\neq k}^{n} X_{kj} \le y_j \qquad \forall k = \{1,2,\dots,n\} \quad (۹)$$

$$\sum_{j=1}^{n} X_{0j} = 1 \quad (۱۰)$$

$$X_{kj} + X_{jk} \le 1 \qquad \forall j,k = \{1,2,\dots,n-1\}, j > k \quad (۱۱)$$

$$p_{ji} = (a_{ji} + bt_{ji})y_j \qquad \forall j,i \quad (۱۲)$$

$$t_{ji} \ge C_{ki} + (X_{kj} - 1)M \qquad \forall j,i,k,j\neq k \quad (۱۳)$$

$$t_{ji} \ge C_{j,i-1} + p_{ji} \qquad \forall j, i > 1 \quad (۱۴)$$

$$C_{j1} \ge p_{j1} \qquad \forall j \quad (۱۵)$$

$$C_{ji} \ge C_{ki} + p_{ji} + (X_{kj} - 1)M \qquad \forall j,i,k \quad (۱۶)$$

$$C_{ji} \ge C_{j,i-1} + p_{ji} \qquad \forall j, i > 1 \quad (۱۷)$$

$$T_j \ge C_{j,m} - d_j \qquad \forall j \in AG_1 \quad (۱۸)$$

$$C_{max} \ge C_{jm} \qquad \forall j \in AG_2 \quad (۱۹)$$

$$C_{ji} \le d_j + MU_j \qquad \forall j \in AG_3, i \quad (۲۰)$$

$$y_j = 1 \qquad \forall j \in AG_2 \quad (۲۱)$$

$$C_{ji} \ge 0 \qquad \forall j,i \quad (۲۲)$$

$$T_j \ge 0 \qquad \forall j \in AG_1 \quad (۲۳)$$

$$X_{kj} \in \{0,1\} \qquad \forall j,k,j\neq k \quad (۲۴)$$

$$y_j \in \{0,1\} \qquad \forall j \quad (۲۵)$$

رابطه (۷) توابع هدف مسأله را نشان می‌دهد که عبارت‌اند از:

f_1: حداقل کردن مجموع هزینه تأخیر کارهای پردازش شده و میزان جریمه کارهای رد شده برای عامل اول.

f_2: حداقل کردن زمان اتمام آخرین کار عامل دوم.

f_3: حداقل کردن مجموع هزینه تعداد کارهای به تأخیر افتاده و هزینه رد کردن کارها برای عامل سوم.

همچنین لازم به ذکر است که این مدل یک حالت خاص از مسأله چند عاملی است و می‌توان برای بیشتر از سه تابع هدف هم آن را تعمیم داد که تابع هدف آن به صورت زیر نوشته می‌شود.

$$\min(f_1, f_2, \dots, f_Q)$$

محدودیت (۸) تضمین می‌کند که هر کاری که تصمیم به پردازش آن گرفته شود ($y_j = 1$)، حتماً باید یک پیشنیاز داشته باشد و اگر تصمیم بر رد کردن آن باشد ($y_j = 0$)، آن کار هیچ پیشنیازی هم نخواهد داشت. رابطه (۹) محدودیت پسنیازی است و تضمین می‌کند که هر کاری که تصمیم به پردازش آن گرفته شود ($y_j = 1$)، حداکثر یک کار پسنیازی دارد و اگر تصمیم بر رد کردن آن باشد ($y_j = 0$)، آن کار هیچ پسنیازی ندارد. معادله (۱۰) تأکید دارد که کار مجازی حتماً باید یک پسنیاز داشته باشد. محدودیت (۱۱) نشان می‌دهد که یک کار نمی‌تواند در یک زمان هم پیشنیاز و هم پسنیاز یک کار دیگر باشد. معادله (۱۲) زمان پردازش واقعی یک کار را توسط رابطه (۶) محاسبه می‌کند. اگر $y_j = 1$ باشد، p_{ji} مقدار می‌گیرد و در غیر اینصورت صفر خواهد بود. همانطور که مشاهده می‌شود این رابطه غیرخطی است و در ادامه چگونگی تبدیل آن به یک محدودیت خطی توضیح داده خواهد شد. روابط (۱۳) و (۱۴) چگونگی به دست آوردن زمان شروع پردازش یک کار روی یک ماشین را نشان می‌دهد (زمان شروع یک کار روی یک ماشین خاص یا برابر زمان اتمام کار پیشنیازش در همان ماشین یا مساوی زمان انجام عملیات خود در ماشین قبلی است). (۱۵) تضمین می‌کند که زمان انجام هر کار در ماشین اول باید از زمان پردازش آن کار بیشتر باشد. محدودیت (۱۶) اطمینان حاصل می‌کند که زمان اتمام هر کار روی هر ماشین باید از زمان اتمام کار پیشنیازی آن بیشتر باشد. رابطه (۱۷) بر این اصل تأکید دارد که زمان اتمام هر کار در ماشین i باید بزرگتر از زمان اتمام عملیات آن در ماشین $i-1$ به علاوه زمان پردازش روی ماشین i باشد (تا عملیات قبلی یک کار تمام نشده است، عملیات بعدی آن شروع نخواهد شد). (۱۸) مدت زمان تأخیر کارهای عامل دوم را محاسبه می‌کند (m بیانگر آخرین ماشین است). محدودیت (۱۹) متغیر C_{max} عامل دوم را تعریف می‌کند. رابطه (۲۰) نحوه محاسبه U_j را برای عامل سوم نشان می‌دهد. معادله (۲۱) بیانگر این مطلب است که همه کارهای عامل دوم باید انجام شوند و قابل رد شدن، نیستند. روابط (۲۲) تا (۲۵) نیز متغیرهای تصمیم مسأله را تعریف می‌کند.

خطی کردن مدل:

معادله (۱۲) را دوباره در نظر بگیرید.

$$p_{ji} = (a_{ji} + bt_{ji})y_j = a_{ji}y_j + bt_{ji}y_j \quad (۲۶)$$

همانطور که مشاهده می‌شود، به علت وجود ضرب متغیر باینری و پیوسته، قسمت دوم معادله‌ی (۲۶) غیرخطی است. طبق آنچه که در کتاب چن و همکاران [۲۱] گفته شده است، یک رابطه غیرخطی به صورت زیر، خطی سازی می‌شود.

اگر t_{ji} یک متغیر پیوسته و y_j متغیر صفر و یک باشد، داریم:

$$p_{ji} = (a_{ji} + bt_{ji})y_j = a_{ji}y_j + bt_{ji}y_j = a_{ji}y_j + bR_{ji} \quad (۲۷)$$

$$R_{ji} \le t_{ji} \quad (۲۸)$$

$$R_{ji} \ge t_{ji} - M(1 - y_j) \quad (۲۹)$$

$$R_{ji} \le My_j \quad (۳۰)$$

حال به جای محدودیت (۹) در مدل بالا، روابط (۲۷) تا (۳۰) جایگزین می‌شود.

۳- روش‌های حل مسأله

۳-۱- روش محدودیت جزئی تعمیم‌یافته

از آنجایی که مدل مسأله حاضر، سه هدفه است، برای حل آن از روش محدودیت جزئی تعمیم‌یافته که جز روش‌های کلاسیک حل مسائل چندهدفه است، استفاده شده است. روش محدودیت جزئی، اولین بار در سال ۱۹۷۱ توسط هایمس و همکاران ارائه شده است. رویکرد این روش به گونه‌ای است که مسأله چند هدفه را تبدیل به مسأله تک هدفه می‌کند؛ به این صورت که یکی از توابع هدف موجود ($f_q(x), q = \{1,2, \dots, Q\}$) انتخاب و حداقل سازی شده و سایر توابع هدف به محدودیت‌هایی با حد بالا تبدیل می‌شوند. روش اپسیلون محدودیت، مجموعه‌ای از راه‌حل‌های تعادلی را تحویل تصمیم‌گیرنده می‌دهد و سپس تصمیم‌گیرنده می‌تواند با توجه به شرایط، یک یا چند تا از این جواب‌ها را انتخاب نماید. مدل کلی این روش برای پژوهش حاضر، به صورت زیر تعریف می‌شود [۲۲]:

$$\begin{aligned} & Min\ f_1(x) + \rho(f_2(x) + f_3(x)) \\ & \text{s.t} \\ & g(x) \le 0 \\ & f_2(x) \le \varepsilon_2 \\ & f_3(x) \le \varepsilon_3 \\ & x \in X \end{aligned} \tag{۳۱}$$

که در آن $g(x)$ شامل همه محدودیت‌های اصلی مسأله و ε_2 و ε_3 حدود بالای به دست آمده برای توابع هدف دوم و سوم هستند. همچنین لازم به ذکر است که برای حل مدل فوق به روش محدودیت جزئی، تابع هدف عامل اول به عنوان تابع هدف اصلی انتخاب شده و توابع هدف عامل‌های دوم و سوم، به محدودیت‌های مسأله افزوده شده‌اند.

۳-۲- الگوریتم NSGA-II

در این پژوهش یک الگوریتم ژنتیک چند هدفه مبتنی بر مرتب‌سازی نامغلوب، برای حل مسأله در ابعاد بالا ارائه شده است. الگوریتم NSGA-II یکی از پرکاربردترین و قدرتمندترین الگوریتم‌های تکاملی موجود بری حل مسائل بهینه‌سازی چند هدفه است و کارایی آن در حل مسائل مختلف به اثبات رسیده است. همچنین این روش کاربردهای فراوانی در حل مسائل زمان‌بندی چند هدفه داشته که از آن جمله می‌توان به پژوهش‌های سیرو و همکاران [۲۳] و آصفی و همکاران [۲۴] اشاره داشت.

نحوه عملکرد این الگوریتم، به این صورت است که ابتدا یک جمعیت اولیه P_0 به صورت تصادفی تولید می‌شود. رتبه‌بندی جبهه‌ها انجام می‌شود و به کروموزوم‌های هر جبهه (سطح)، برازشی معادل رتبه آن تخصیص داده می‌شود. با اعمال عملگرهای تزویج و جهش، جمعیت Q_0 تولید می‌شود. شبه کد این الگوریتم در شکل (۱) نشان داده شده است [۲۵].

> دو جمعیت والدین و فرزندان را ادغام کنید: $R_t = P_t \cup Q_t$
> رتبه‌بندی غیر مسلط روی جمعیت R_t را اعمال کنید و کروموزوم‌ها را در جبهه‌های F_i ($i = 1,2, \dots$) دسته‌بندی کنید.
> جمعیت جدید $P_{t+1} = \emptyset$ را ایجاد کنید. شمارنده i را مساوی یک فرض کنید.
> تا زمانی که $|P_{t+1}| + |F_i| < N$، عملیات $P_{t+1} = P_{t+1} \cup F_i$ و $i = i + 1$ را انجام دهید.
> رتبه‌بندی بر اساس ازدحام را انجام دهید و جواب‌های دارای بیشترین پراکندگی را از جواب‌های مرتب شده در F_i در P_{t+1} قرار دهید.
> کروموزوم‌های جدید در جمعیت Q_{t+1} را با اعمال عملگرهای انتخاب مسابقه‌ای (بر اساس فاصله ازدحام)، تزویج و جهش بر روی جمعیت P_{t+1} تولید کنید.

شکل (۱): شبه کد الگوریتم NSGA-II

۳-۲-۱- ساختار نمایش جواب

نحوه نمایش جواب یکی از مولفه‌های مهم در الگوریتم‌های فراابتکاری است. ساختار نمایش جواب، در مسائل مختلف، متفاوت است و متناسب با ویژگی‌های مسأله مورد بررسی، تعریف می‌شود.

کروموزوم: برای مسأله مورد بررسی در این پژوهش، نمایش جواب به صورت یک رشته به طول تعداد کارها در نظر گرفته شده و هر ژن نشان دهنده یک کار در توالی مورد نظر است. به این دلیل که در این مسأله، مفروض رد کردن کارها در نظر گرفته شده، هر ژن از این نمایش جواب، عددی بین صفر و یک است که اگر ژن مورد نظر صفر باشد، به این معنی است که کار مربوط به آن رد شده است و در ترتیب قرار نمی‌گیرد. برای کارهایی که قرار است پردازش شوند، ترتیب قرار گرفتن در توالی به صورت کوچکترین عدد بزرگ‌تر از صفر تا بزرگترین عدد داخل رشته است. فرض کنید رشته نشان داده شده در شکل (۲) یک نمایش جواب برای این مسأله باشد.

1	2	3	4	5	6
0	0	0.75	0.52	0.86	0

شکل (۲): کروموزوم ایجاد شده برای نماش جواب

در این کروموزوم، کارهای ۱،۲،۶ به این دلیل که عدد متناظر آن در رشته صفر است، رد خواهند شد. برای باقی کارها توالی به دست آمده به صورت زیر خواهد بود.

$$۴ \rightarrow ۳ \rightarrow ۵$$

شکل (۳): توالی به دست آمده از شکل (۲)

عدد متناظر کار ۴ در رشته نشان داده شده، کوچکترین عدد در میان دو عدد دیگر است و به همین دلیل به عنوان اولین کار پردازش خواهد شد. پس از آن کار ۳ با عدد متناظر ۰/۷۵ پردازش می‌شود و آخرین کار قرار گرفته در توالی، کار ۵ خواهد بود.

عملگر تزویج: جهت اعمال عملگر تزویج در این پژوهش، از

تزویج یکنواخت استفاده شده است. نحوه انجام عملگر تزویج یکنواخت به این صورت است که ابتدا یک رشته به ابعاد کروموزوم، ایجاد شده و سپس فرزند اول با نسبت α و $1-\alpha$ از دو والد ایجاد می‌شود. فرزند بعدی نیز به همین ترتیب ولی با نسبت $1-\alpha$ و α خواهد شد. در شکل (۳)، چگونگی اعمال این عملگر نشان داده شده است.

والد 1	0.4387	0.3815	0.7655	0.7951	0.1868	0.4897
والد 2	0.4455	0.6463	0.7093	0.7546	0.2768	0.6797
α	0.6551	0.16261	0.1189	0.4983	0.9597	0.34033
فرزند 1	0.4411	0.6032	0.7160	0.7748	0.1904	0.6150
فرزند 2	0.443	0.42461	0.7588	0.77500	0.2724	0.5548

شکل (۳): عملگر تزویج یکنواخت

به عنوان مثال در شکل (۳)، مقدار اولین ژن فرزند اول، به صورت زیر محاسبه می‌شود.

$$y_1 = (0.6551 * 0.438) + (1 - 0.6551) * 0.4455 = 0.4411$$

عملگر جهش: برای انجام عملگر جهش در این مسأله، یک کروموزوم انتخاب می‌شود و تعدادی از ژن‌هایش، با توجه به نرخ از پیش تعیین شده، برابر صفر قرار گرفته و یا مجددا مقدار تصادفی بین صفر و یک می‌گیرد.

پس از اینکه عملگرهای تزویج و جهش بر روی کروموزوم‌ها انجام شد، با استفاده از کلید تصادفی، رشته‌ها از حالت پیوسته به گسسته تبدیل می‌شوند. همچنین لازم به ذکر است که جهت تنظیم پارامترهای الگوریتم ژنتیک چند هدفه پیشنهادی، از روش تاگوچی استفاده شده است. نحوه به دست آوردن مقادیر بهینه پارامترها با استفاده از روش تاگوچی به این صورت است که ابتدا برای یک مثال خاص، به تعداد اجراهایی که تاگوچی تعیین می‌کند، مسأله با الگوریتم مورد نظر حل می‌شود. در مرحله بعد معیارهای ارزیابی عملکرد الگوریتم‌های چند هدفه برای هر مثال، به دست آورده می‌شود. همچنین به علت مهم بودن زمان حل الگوریتم‌ها در مسائل زمان‌بندی، زمان نیز به عنوان معیار ارزیابی عملکرد به مسأله اضافه می‌شود. سپس این معیارها را با استفاده از روش RPD[1] بی مقیاس کرده و جمع وزنی آن‌ها به نرم‌افزار داده می‌شود. بعد از اجرای نرم‌افزار مینی تب[2]، سطوح بهینه پارامترهای الگوریتم به دست می‌آید. پارامترهای انتخابی و سطوح بهینه آن‌ها در جدول (۱) نشان داده شده است.

به علت متفاوت بودن ارزیابی مسائل چند هدفه با مسائل تک هدفه، ابتدا لازم است که معیارهای ارزیابی عملکرد روش‌های حل مسائل چند هدفه بیان شود که این معیارها، عبارت‌اند از:

1. Relative Percentage Deviation
2. Minitab

۱. تعداد نقاط غیر مسلط پیدا شده (NPS): این معیار، تعداد جواب‌های پیدا شده توسط الگوریتم را نشان می‌دهد [۱۰].

۲. معیار همگرایی (CM): این معیار، جهت ارزیابی عملکرد یک مجموعه غیر مسلط Ω_j نسبت به یک مجموعه Ω^* مرجع استفاده می‌شود و هر چه کمتر باشد، نشان دهنده عملکرد خوب الگوریتم پیشنهادی است.

جدول(۱): سطوح بهینه پارامترهای الگوریتم NSG-II

پارامتر	تعریف	سطح بهینه
MaxIT	حداکثر تعداد تکرارهای الگوریتم (معیار توقف)	۱۰۰
nPop	تعداد جمعیت اولیه	۱۵۰
PCrossover	احتمال انجام تزویج	۰/۶
PMutation	احتمال انجام جهش	۰/۳
mu	نرخ جهش	۰/۰۲
sigma	اندازه گام جهش	۰/۰۷

۴- نتایج محاسباتی

در این بخش نتایج مثال‌های عددی حل شده با دو روش حل پیشنهادی ارائه می‌شود. برای تولید مقادیر پارامترهای این مسأله، به دلیل نبود داده‌های جامع در ادبیات موضوع، از روش ایجاد تصادفی اعداد در نرم‌افزار اکسل استفاده شده و در جدول (۱) نشان داده شده است. همچنین لازم به ذکر است که تمامی محاسبات مربوطه با استفاده از نرم‌افزار بهینه‌سازی GAMS و MATLAB 2014، در یک رایانه شخصی با مشخصات Intel Core i7، CPU 2.66 GHZ و RAM 6GB انجام شده است.

جدول(۲): مقادیر پارامترهای مسأله

پارامتر	تعریف	مقادیر
n	تعداد کارها	۶ و۸و۱۲
m	تعداد ماشین‌ها	۲و۳و۵
M	مقدار بزرگ	۱۰۰۰۰
ρ	مقدار بسیار کوچک	۰/۰۰۰۰۱
b	نرخ اثر زمانی	۰/۱
a_{ji}	زمان پردازش نرمال کار j روی ماشین i	$unif\_rand(1,10)$
d_j	موعد تحویل کار j	$unif\_rand(15,30)$
e_j	میزان جریمه رد کردن کار j	$unif\_rand(1,5)$
tc	هزینه هر واحد تأخیر	۲
uc	هزینه هر واحد کار به تأخیر افتاده	۳

$$CM = \frac{1}{|\Omega^*|} \sum_{y\in\Omega^*} \min\{d_{xy} | x \in \Omega_j\} \qquad (۳۲)$$

که در آن d_{xy} فاصله جواب x به دست آمده توسط الگوریتم از جواب مرجع y در فضای توابع هدف است و به صورت زیر محاسبه می‌شود:

$$d_{xy} = \sqrt{(f_1(x) - f_1(y))^2 + \cdots + (f_Q(x) - f_Q(y))^2} \qquad (۳۳)$$

همچنین Q تعداد توابع هدف را نشان می‌دهد [۱۰].

۱- معیار حداکثر گسترش یا پراکندگی (ME): این معیار که توسط زیتزلر و همکاران [۲۶] معرفی شده است، میزان گسترش جواب‌های حاصل از الگوریتم را نشان می‌دهد. بیشترین بودن این مقدار نشان دهنده عملکرد خوب الگوریتم است. معیار حداکثر گسترش از رابطه زیر به دست می‌آید.

$$ME = \sqrt{\sum_{q=1}^{Q} \max\{(f_q(x) - f_q(y))^2\} \ \forall x, y \in \Omega_j} \quad (۳۴)$$

۲- معیار فاصله (SM): این شاخص میزان یکنواختی نقاط نامغلوب را در فضای حل اندازه‌گیری می‌کند. مقادیر کم این معیار بیانگر توزیع یکنواخت‌تر جواب در پارتوی شناسایی شده است [۲۷].

$$SM = \frac{\frac{\sqrt{\sum_{i=1}^{n}(d_i - \bar{d})^2}}{n}}{\bar{d}} \quad (۳۵)$$

که در آن n تعداد جواب‌های پارتو و d_i فاصله اقلیدسی بین دو جواب پارتوی کناری در فضای حل است و از رابطه (۳۳) محاسبه می‌شود. همچنین $\bar{d}$ برابر میانگین فواصل d_iها است.

$$\bar{d} = \frac{\sum_{i=1}^{n} d_i}{n} \quad (۳۶)$$

۴-۱- مقایسه دو روش محدودیت جزئی و NSGA-II

در جدول (۲) نتایج حاصل شده از الگوریتم پیشنهادی و روش دقیق محدودیت جزئی در ۱۸ مثال عددی مختلف در اندازه‌های پایین آورده شده است. نتایج نوشته شده برای الگوریتم، بهترین مقدار از ۵ بار اجرای آن است. اعداد نوشته شده در قسمت Time، زمان‌های اجرای هر یک از این روش‌های حل را بر حسب ثانیه نشان می‌دهند. زمان حل نرم‌افزار GAMS بر روی ۱۰۰۰۰ ثانیه تنظیم شده است و اگر تا آن زمان جواب بهینه‌ای حاصل نشد، همان مقدار به دست آمده در ۱۰۰۰۰ ثانیه، به عنوان جواب این روش ثبت می‌شود. پس از مقایسه جواب‌های الگوریتم NSGA-II با روش محدودیت جزئی مشاهده می‌شود که زمان‌های حل الگوریتم فراابتکاری در ۱۰ مورد از ۱۸ مثال حل شده کمتر از روش محدودیت جزئی است (با افزایش اندازه مسأله، زمان حل NSGA-II بهتر از محدودیت جزئی شده است). در مثال‌هایی که با * مشخص شده (۶ مثال)، مشاهده می‌شود که جواب‌های حاصل از الگوریتم ژنتیک چند هدفه، مشابه جواب‌های نرم‌افزار GAMS است.

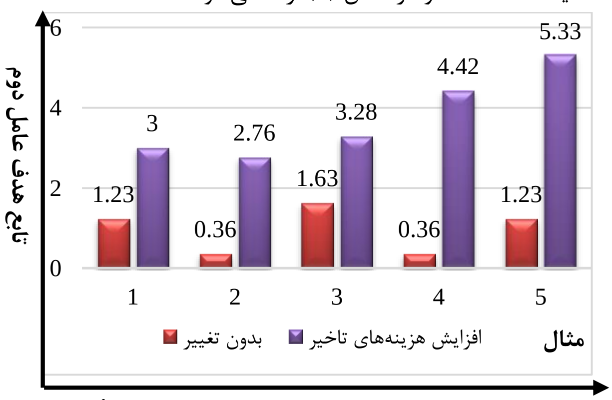
به این معنی که الگوریتم فراابتکاری توانسته تمام جواب‌های ممکن به دست آمده از روش دقیق را پیدا کند و در نتیجه تعداد نقاط غیر مسلط پیدا شده، گسترش و فاصله جواب‌ها در هر دو روش برابر و در نتیجه معیار همگرایی NSGA-II به روش محدودیت جزئی صفر است. همچنین جواب‌های به دست آمده از NSGA-II در ۹ مثال، از نظر تعداد نقاط غیر مسلط پیدا شده بهتر یا برابر روش محدودیت جزئی است که در جدول (۳) با × مشخص شده است. در انتها نتایج به دست آمده نشان می‌دهد که کیفیت جواب‌های الگوریتم NSGA-II به طور قابل قبولی برابر یا همگرا به جواب‌های حاصل از روش دقیق محدودیت جزئی است چرا که در ۱۵ مورد از مثال‌های حل شده جواب‌های به دست آمده از روش فراابتکاری برابر یا بهتر از روش دقیق بوده است که این امر نشان‌دهنده این است که NSGA-II یک روش حل مناسب برای حل مسأله زمان‌بندی چند عاملی با دو مفروض اثر زمانی و رد کردن بوده است.

۵- اعتبار و پایایی مدل

در این قسمت، به ارزیابی عملکرد مدل پیشنهادی و تحلیل رفتار مدل، تحت مقادیر مختلف پارامترها پرداخته می‌شود. برای این منظور لازم است که با تغییر مقادیر پارامترهای مهم مسأله، چگونگی پاسخ مدل به این تغییرات سنجیده شود. در ادامه تحلیل حساسیت‌های انجام شده بر روی مدل و نتایج حاصل شده از آن‌ها ارائه می‌شود.

۵-۱- تأثیر افزایش هزینه‌های تأخیر بر متوسط تعداد کارهای رد شده

در این بخش تأثیر افزایش هزینه‌های تأخیر بر متوسط تعداد کارهای رد شده در هر راه‌حل غیر مسلط بررسی خواهد شد. انتظار می‌رود که اگر هزینه‌های تأخیر افزایش یابد، تصمیم‌گیرنده، تعداد کارهای بیشتری را رد خواهد کرد. برای بررسی این موضوع، در ۵ مثال مختلف هزینه‌های تأخیر افزایش یافته، نتایج با حالت اولیه آن‌ها مقایسه شده است و در شکل (۵) ارائه می‌شود:



شکل (۴): تحلیل حساسیت افزایش هزینه‌های تأخیر

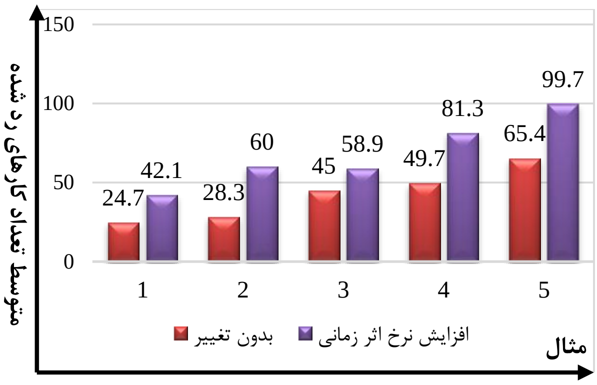
همانطور که در شکل (۵) مشاهده می‌شود، متوسط تعداد کارهای رد شده با افزایش هزینه‌های تأخیر افزایش می‌یابد و این دقیقاً همان انتظاری بود که از مدل وجود داشت.

جدول(۳): مقایسه الگوریتم NSGA-II با روش محدودیت جزئی

$n×m$	n_1 / n_2	ε-constraint				NSGA-II				
		NPS	**ME**	**SM**	**Time**	**NPS**	**CM**	**ME**	**SM**	**Time**
۶×۲	۲ / ۲	۵	۱۷/۱۱	۰/۲۷	۶/۵	۵❋	۰	۱۷/۱۱	۰/۲۷	۳۶/۳۹
	۱ / ۲	۷	۱۸/۲۱	۰/۲۱	۸/۴	۹×	۰/۱۳	۱۸/۸۶	۰/۲۴	۳۷/۹۲
۶×۳	۲ / ۲	۴	۱۹/۰۶	۰/۳۳	۴/۵	۴❋	۰	۱۹/۰۶	۰/۳۳	۳۵/۷۲
	۱ / ۲	۳	۵۵/۷۹	۰/۷۳	۶/۶	۳❋	۰	۵۵/۷۹	۰/۷۳	۳۷/۱۴
۶×۵	۲ / ۲	۸	۲۱/۴۸	۰/۰۶۹	۷	۶	۰/۱۹	۱۹/۸۵	۰/۰۹۱	۳۶/۲۴
	۱ / ۲	۲	۱۳/۸۲	۰	۹/۸	۲❋	۰	۱۳/۸۲	۰	۳۹/۶۶
۸×۲	۳ / ۳	۱۰	۴۱/۷۷	۰/۲۲	۲۲/۸	۱۲×	۰/۱۱	۴۰/۹۴	۰/۵۷	۳۸/۷۶
	۲ / ۴	۵	۴۷/۱۸	۰/۳۹	۳۵/۳۳	۵	۰/۰۹	۴۹/۹	۰/۶۳	۴۱/۳۶
۸×۳	۳ / ۳	۷	۳۶/۸۵	۰/۳۴	۱۲۵/۱۸	۱۰×	۰/۲۱	۳۲/۶	۰/۴۴	۴۲/۴۸
	۲ / ۴	۶	۳۸/۲۴	۰/۳۵	۱۳۴/۱۸	۵	۰/۱۸	۳۶/۶۹	۰/۴۳	۴۰/۵۲
۸×۵	۳ / ۳	۳	۱۲/۷۱	۰/۲۷	۲۳۰/۲۵	۳❋	۰	۱۲/۷۱	۰/۲۷	۳۸/۹۶
	۲ / ۴	۴	۳۴/۵	۰/۸	۲۵۰/۱۲	۴❋	۰	۳۴/۵	۰/۸	۴۳/۸۹
۱۲×۲	۴ / ۴	۹	۲۳/۳۲	۰/۳	۵۵۷۰	۱۱×	۰/۱۴	۳۵/۳۲	۰/۴۶	۴۴/۲۵
	۵ / ۴	۸	۳۸/۳	۰/۱۲	۶۳۳۰	۸	۰/۰۸	۴۲/۳	۰/۳۱	۴۱/۲۷
۱۲×۳	۴ / ۴	۱۲	۵۴/۵	۰/۱	۵۸۷۰	۱۱	۰/۲۶	۵۱/۴۵	۰/۱۱	۴۵/۴۴
	۵ / ۴	۱۰	۴۰/۳۲	۰/۵۶	۵۹۶۰	۱۰	۰/۰۷	۴۲/۸	۰/۶۱	۴۶/۸۵
۱۲×۵	۴ / ۴	۱۳	۳۸/۸۳	۰/۴۳	۷۶۳۵	۱۳	۰/۲۲	۴۵/۹۶	۱/۳۸	۴۵/۱۹
	۵ / ۴	۱۴	۵۵/۷۹	۰/۷۳	۸۷۸۰	۱۶×	۰/۰۸	۵۹/۵۴	۰/۶۹	۴۷/۶۸

۵-۲- تأثیر افزایش نرخ اثر زمانی بر توابع هدف

در این بخش مشابه آنچه که در بخش قبل انجام شد، تأثیر افزایش نرخ اثر زمانی بر توابع هدف بررسی می‌شود. همانطور که در فصل پیشین گفته شد، زمان پردازش واقعی کارها از رابطه (۶) پیروی می‌کند. انتظار می‌رود با افزایش مقدار b، زمان پردازش و درنتیجه‌ی آن زمان اتمام آخرین کار افزایش یابد که این مسأله باعث افزایش مقدار تابع هدف دوم مسأله یعنی زمان ات مام آخرین کار می‌شود. شکل (۶) بیانگر این موضوع است. اعداد نشان داده شده در این شکل، نشان دهنده حداکثر مقدار ثبت شده برای تابع هدف دوم مسأله در هر مثال است.



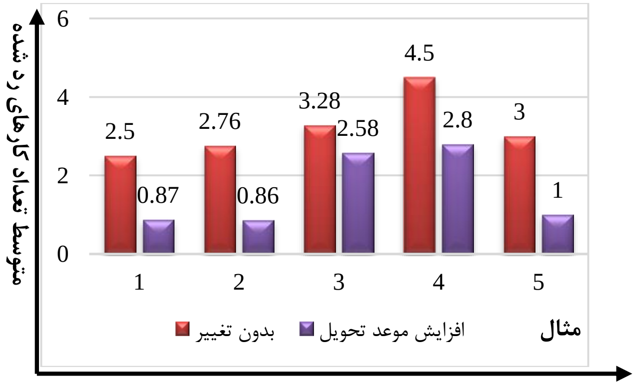
شکل (۵): تحلیل حساسیت افزایش b

۵-۳- تأثیر افزایش موعد تحویل بر متوسط تعداد کارهای رد شده

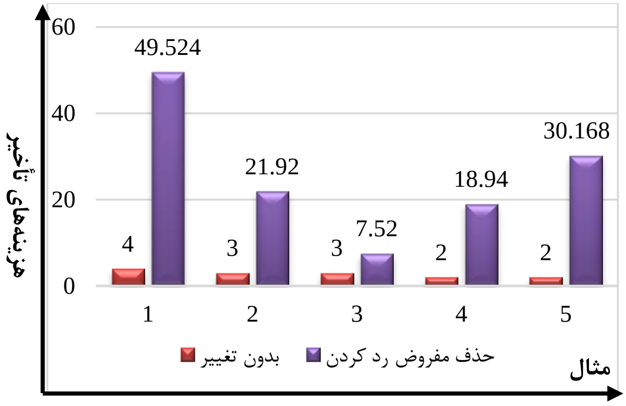
در این بخش نیز تأثیر افزایش پارامتر موعد تحویل بر متوسط تعداد کارهای رد شده بررسی می‌شود. بدیهی است که با افزایش مناسب زمان تحویل، تعداد کارهای رد شده کاهش بیابد و دلیل آن هم این است که با افزایش موعد تحویل، مدت زمان تأخیر کارها کاهش می‌یابد و نیاز به رد کردن آن‌ها کم می‌شود. شکل (۷) به خوبی این حالت را نشان می‌دهد.

۵-۴- حذف مفروض رد کردن از مسأله

تأثیر حذف محدودیت رد کردن کارها بر توابع هدفی که مقادیر تأخیر را محاسبه می‌کنند، یکی دیگر از تحلیل‌هایی است که می‌توان روی مدل مسأله انجام داد. دلیل اصلی اضافه شدن مفروض رد کردن به مسأله، کاهش هزینه‌های تأخیر است؛ زیرا وقتی تصمیم‌گیرنده با کارهای دارای تأخیر مواجه شود، می‌تواند با در نظر گرفتن هزینه تأخیر، آن‌ها را رد کند و مقدار تابع هدف مسأله را کاهش دهد. حال وقتی اجازه رد کردن کارها در مسأله وجود نداشته باشد، انتظار داریم که هزینه‌های تأخیر افزایش یابد. برای بررسی این حالت، حداقل هزینه تأخیر هر مثال در حالت وجود یا عدم وجود رد کردن با هم مقایسه می‌شوند و نتایج آن در شکل (۸) قابل ملاحظه است.



شکل (۶): تحلیل حساسیت افزایش d_j



شکل (۷): تحلیل حساسیت حذف مفروض رد کردن

۶- نتیجه‌گیری

مسأله زمان‌بندی چند عاملی، در سال‌های اخیر مورد توجه بسیاری از پژوهشگران قرار گرفته است. با این حال، این مسأله در محیط چند ماشینی و مفروضات واقعی مورد پژوهش قرار نگرفته است. در این پژوهش یک مسأله زمان‌بندی چند عاملی با در نظر گرفتن دو مفروض کاربردی اثر زمانی و رد کردن در محیط جریان کارگاهی، بررسی شده است یک مدل برنامه‌ریزی عدد صحیح مختلط و یک الگوریتم فراابتکاری ژنتیک چند هدفه مبتنی بر مرتب‌سازی نامغلوب برای این مسأله ارائه شده است. الگوریتم NSGA-II در مثال‌های عددی مختلف با روش حل محدودیت جزئی تعمیم‌یافته مقایسه شده و نتایج حاصل نشان دهنده همگرایی قابل قبول راه‌حل‌های این الگوریتم به جواب‌های مرجع به دست آمده از روش محدودیت جزئی، است. همچنین با انجام تحلیل حساسیت‌های مختلف روی پارامترهای مسأله، اعتبار و صحت مدل ارائه شده، مورد تأیید قرار گرفت.

برای مطالعات آتی، می‌توان مسأله مورد نظر را با مسائل زنجیره تأمین (در غالب زمان‌بندی زنجیره تأمین) ادغام و یا مفروضات واقعی دیگر، از قبیل زمان آماده‌سازی، اثر یادگیری و ... به مسأله اضافه کرد. همچنین می‌توان این مسأله را با ارائه الگوریتم‌های فراابتکاری جدید، در محیط‌های دیگر تولیدی مانند محیط تولید کارگاهی[1] در نظر گرفت.

مراجع

[1] Pinedo, M., (2015). Scheduling. Springer.
[2] Agnetis, A., Billaut, J.C., Gawiejnowicz, S., Pacciarelli, D., Souhal, A., (2014). "Multi-agent scheduling". Berlin Heidelberg: Springer Berlin Heidelberg. doi, 10(1007): 978-3.
[3] Wu, W.H., Yin, Y., Wu, W.H., Wu, C.C., Hsu, P.H. (2014). "A time-dependent scheduling problem to minimize the sum of the total weighted tardiness among two agents", Journal of Industrial and Management Optimization, 10(2): 591-611.
[4] Shabtay, D., Gaspar, N., Kaspi, M. (2013). "A

1. Job Shop

survey on offline scheduling with rejection", Journal of Scheduling, 16(1): 3-28.

[5] Baker, K.R., Smith, J.C., (2003). "A multiple-criterion model for machine scheduling", Journal of Scheduling, 6(1): 7-16.

[6] Agnetis, A., Mirchandani, P.B., Pacciarelli, D., Pacifici, A. (2004). "Scheduling problems with two competing agents", Operations Research, 52(2): 229-242.

[7] Fan, B.Q., Cheng, T.C.E. (2016). "Two-agent scheduling in a flowshop", European Journal of Operational Research, 252(2): 376-384.

[8] Choi, J.Y. (2015). "Minimizing total weighted completion time under makespan constraint for two-agent scheduling with job-dependent aging effects", Computers & Industrial Engineering, 83: 237-243.

[9] Cheng, S.R., (2014). "Some new problems on two-agent scheduling to minimize the earliness costs", International Journal of Production Economics, 156: 24-30.

[10] Lei, D. (2015). "Variable neighborhood search for two-agent flow shop scheduling problem", Computers & Industrial Engineering, 80: 125-131.

[11] Lee, W.C., Chen, S.K., Chen, C.W., Wu, C.C. (2011). "A two-machine flowshop problem with two agents", Computers & Operations Research, 38(1): 98-104.

[12] Shiau, Y.R., Lee, W.C., Kung, Y.S., Wang, J.Y. (2016). "A lower bound for minimizing the total completion time of a three-agent scheduling problem", Information Sciences, 340: 305-320.

[13] Kunnathur, A.S., Gupta, S.K. (1990). "Minimizing the makespan with late start penalties added to processing times in a single facility scheduling problem", European Journal of Operational Research, 47(1): 56-64.

[14] Cheng, T.C.E., Lee, W.C., Wu, C.C. (2010). "Single-machine scheduling with deteriorating functions for jobs procedding times", Applied Mathematical Modeling, 34(12): 4171-4178.

[15] Yin, Y., Cheng, T.C.E., Wan, L., Wu, C.C., Liu, J. (2015). "Two-agent single-machine scheduling with deteriorating jobs", Computers and Industrial Engineering, 81: 177-185.

[16] Lee, W.C., Wang, W.J., Shiau, Y.R. Wu, C.C. (2010). "A single machine scheduling problem with two agent and deteriorating jobs", Applied Mathematical Modeling, 34(10): 3098-3107.

[17] Bartal, Y., Leonardi, S., Marchetti-Spaccamela, A., Sgall, J., Stougie, L. (2000). "Multiprocessor scheduling with rejection", SIAM Journal on Discrete Mathematics, 13(1): 64-78.

[18] Feng, Q., Fan, B., Li, S., Shang, W., (2015). "Two-agent scheduling with rejection on a single machine", Applied Mathematical Modelling, 39(3): 1183-1193.

[19] Li, S., Yuan, J., (2010). "Parallel-machine scheduling with deteriorating jobs and rejection", Theoretical Computer Science, 411(40): 3642-3650.

[20] Wu, C.C., Lee, W.C., (2008). "Single-machine group-scheduling problems with deteriorating setup times and job-processing times", International Journal of Production Economics, 115(1): 128-133.

[21] Chen, D.S., Batson, R.G., Dang, Y., (2010). "Applied integer programming: modeling and solution",1st edition, John Wiley & Sons.

[22] Sadjadi, S.J., Heidari, M., Esboei, A.A., (2014). "Augmented ε-constraint method in multi objective staff scheduling problem: a case study", The International Journal of Advanced Manufacturing Technology, 70(5-8): 1505-1514.

[23] Ciro, G.C., Dugardin, F., Yalaoui, F., Kelly, R., (2016). "A NSGA-II and NSGA-III comparison for solving an open shop scheduling problem with resource constraints", IFAC-Papers Online, 49(12): 1272-1277.

[24] Asefi, H., Jolai, F., Rabiee, M., Araghi, M.T., (2014). "A hybrid NSGA-II and VNS for solving a bi-objective no-wait flexible flowshop scheduling problem", The International Journal of Advanced Manufacturing Technology, 75(5-8): 1017-1033.

[25] Zitzler, E., Deb, K., Thiele, L. (2000). "Comparison of multiobjective evolutionary algorithms: Empirical results", Evolutionary computation, 8(2): 173-195.

[26] Zitzler, E., Deb, K., Thiele, L. (2000). "Comparison of multiobjective evolutionary algorithms: Empirical results", Evolutionary computation, 8(2): 173-195.

[27] Goh, C.K., Tan, K.C., Liu, D.S., Chiam, S.C., (2010). "A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design", European Journal of Operational Research, 202(1): 42-54.

Journal of Industrial Engineering Research in Production Systems

ISSN: 2345-2269

Journal of Industrial Engineering Research in Production Systems

Spring and Summer 2017 Volume 5. No. 10. pp.17-29

www.ier.basu.ac.ir

DOI: 10.22084/ier.2017.11693.1534

Solving a Multi-Agent Scheduling Problem in a Flow Shop Environment Considering Rejection and Deteriorating Jobs Using a Meta-Heuristic Algorithm

M. Hosseinzadeh[1], R. Sahraeian[1,*]

[1] Department of Industrial Engineering, Shahed University, Tehran, Iran.

ARTICLE INFO

***Article history*:**

Received 25 September 2016
Accepted 18 June 2017

***Keywords*:**

Scheduling
Agent
Deteriorating jobs
Rejection
MIP

ABSTRACT

A multi-agent scheduling problem in a flow shop environment has been considered in this study. Multi-agent scheduling problem is a subset of multi-objective scheduling problems in which each agent has a set of jobs and its aim is to optimize its own objective function. To make the proposed problem more realistic, two practical assumptions such as deteriorating jobs and rejection has been considered. A mixed integer programming model is presented for the problem. The main contribution of the proposed model is to consider multi-agent with two mentioned assumptions. Also, due to the complexity of the model and its inability to solve large-scale problems, a meta-heuristic Non-Dominated sorting Genetic Algorithm (NSGA-II) are developed. Obtained solutions of this algorithm are compared with exact augmented ε-constraint method and the results confirm its performance.

[*] Corresponding author. Rashed Sahraeian
Tel.: 021-51212040; *E-mail address: Sahraeian@shahed.ac.ir*