

## حل یک مسأله زمانبندی چند هدفه جدید در سیستم تولید سلولی با استفاده از یک الگوریتم تلفیقی

یوسف قلی پورکنعانی<sup>\*</sup>، رضا توکلی مقدم<sup>۲</sup>، مجتبی طبری<sup>۳</sup>، یاسر جعفری زرندینی<sup>۴</sup>، میربهادر قلی آریانزاد<sup>۵</sup>

۱- دانشجوی دکتری مهندسی صنایع، دانشکده فنی و مهندسی، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی

۲- استاد گروه مهندسی صنایع، پردیس دانشکده‌های فنی، دانشگاه تهران

۳- استادیار گروه مدیریت، واحد قائم‌شهر، دانشگاه آزاد اسلامی

۴- دانش آموخته کارشناسی ارشد گروه مهندسی صنایع، واحد تهران جنوب، دانشگاه آزاد اسلامی

۵- استاد گروه مهندسی صنایع، دانشگاه علم و صنعت ایران

### چکیده

در این مقاله، یک مدل ریاضی جدید برای مسأله زمانبندی چند هدفه در سیستم تولید سلولی ارائه می‌شود که هدف آن کمینه‌سازی حداکثر زمان تکمیل کارها، هزینه زودکرد و هزینه دیرکرد است. به علت پیچیدگی این مسأله، یک الگوریتم تلفیقی بر پایه الگوریتم ژنتیک و بهینه‌سازی ذرات انبوه برای حل آن در زمان قابل قبول پیشنهاد می‌گردد. همچنین، از یک الگوریتم تکاملی چند هدفه معروف به نام الگوریتم ژنتیک مرتب شده غیر مغلوب (NSGA-II) برای مقایسه و نشان دادن کارایی الگوریتم تلفیقی پیشنهادی استفاده می‌شود. در ادامه، نتایج حاصل از خروجی الگوریتم‌ها با هم مقایسه و سپس تحلیل می‌گردند. در خاتمه، نتیجه‌گیری و پیشنهادهایی برای تحقیقات آتی ارائه می‌شود.

واژه‌های کلیدی: مسأله زمانبندی چندهدفه، سیستم تولید سلولی، الگوریتم ژنتیک، بهینه‌سازی ذرات انبوه،

NSGA-II.

## ۱. مقدمه

در جهان رقابتی حاضر، تعیین توالی و زمانبندی مؤثر محصولات تولیدی ضرورتی برای بقا در فضای بازار است. زمانبندی ابزاری برای استفاده بهینه از منابع در دسترس است. منابع و کارها در زمانبندی ممکن است انواع گوناگونی داشته باشد و با توسعه جهان صنعتی، منابع مربوطه بحرانی تر می شود (آدامز و همکاران، ۱۹۸۸). زمانبندی این منابع به افزایش کارایی و بهره برداری از ظرفیت، کاهش زمان مورد نیاز برای تکمیل کارها و نهایتاً افزایش سوددهی یک سازمان منجر می شود. زمانبندی مؤثر منابع مانند ماشین ها و نیروی انسانی در محیط رقابتی امروز یک باید است (توکلی مقدم و همکاران، ۲۰۱۰).

اهمیت توالی کارها بر روی گروهی از ماشین ها، با هدف تمرکز بر هزینه ها از آن جهت مورد توجه است که در محیط کسب و کار امروز، رقابت شرکت های تولیدی از طریق قابلیت آن ها برای پاسخگویی سریع به تغییرات سریع در زمینه تجاری و تولید محصولات با کیفیت بالاتر و هزینه های کمتر تعیین می شود. شرکت های تولیدی در تلاش هستند تا این قابلیت ها را از طریق اتوماسیون و مفاهیم جدید مدیریت تولید، مانند تولید به موقع<sup>۱</sup> (JIT) به دست آورند. این مفاهیم به بسیاری از شرکت ها در به دست آوردن سود اقتصادی کمک کرده است. در سیستم های تولید بموقع کارها نباید زودتر و نه دیرتر تکمیل شوند که به مسایل با هزینه های زودکرد و دیرکرد و تخصیص موعدهای تحویل منجر می شود. در یک بازار بسیار رقابتی دیرکرد تحویل محصول با توجه به موعد تحویل آن ها یک مقیاس عملکرد

بسیار مهم برای واحدهای تولیدی در بازار رقابت است، همچنین، زودکرد کارها موجب افزایش هزینه نگهداری می شود.

این مقاله، به حل مسأله تولید سلولی چند هدفه با اهداف کمینه کردن حداکثر زمان تکمیل کارها، کمینه کردن هزینه دیرکرد و کمینه کردن هزینه زودکرد با استفاده از الگوریتم تلفیقی بر پایه بهینه سازی ذرات انبوه<sup>۲</sup> (PSO) و الگوریتم ژنتیک<sup>۳</sup> (GA) می پردازد. مسأله زمانبندی تولید سلولی عبارت است از یافتن توالی بهینه انجام عملیات کارهای مختلف و مرتبط با هر ماشین بر روی آن ماشین در سلول ها و تعیین توالی سلول ها. مسأله زمانبندی تولید سلولی یک مسأله چندجمله ای نامعین سخت (NP-hard) است و جزو سخت ترین و مطرح ترین مسایل بهینه سازی ترکیبی است (لوگندران و اسریسکاندرجاه، ۱۹۹۳). به سبب پیچیدگی ذاتی مسایل بهینه سازی ترکیبی و بخصوص مسأله زمانبندی تولید سلولی، استفاده از روش های ابتکاری برای حل چنین مسایلی، بهبودهای موثری در تولید جواب های قابل قبول ایجاد کرده است؛ چرا که با افزایش ابعاد مسأله، عملاً روش های سنتی تعیین جواب بهینه به دلیل زمان بر بودن کارایی خود را از دست می دهند.

ساختار این مقاله به شرح زیر است. در بخش دوم به مرور ادبیات مسأله مورد نظر می پردازد. بخش سوم به معرفی کامل مسأله مورد نظر از جمله تشریح مسأله، مفروضات، متغیرهای تصمیم گیری و معرفی مدل پیشنهادی می پردازد. در بخش چهارم به ارائه الگوریتم تلفیقی (الگوریتم ژنتیک - بهینه سازی

<sup>۲</sup> Particle Swarm Optimization

<sup>۳</sup> Genetic Algorithm

<sup>۱</sup> Just-In-Time

زمانبندی تولید سلولی با زمان راه‌اندازی خانواده قطعات وابسته به توالی بکار برد. ایشان در این مقاله نشان می‌دهد عموماً بهبود ایجاد شده توسط زمانبندی غیر ترتیبی در برابر زمانبندی ترتیبی برای معیار کارایی مبتنی بر زمان تحویل بهتر است. هندی‌زاده و همکاران (۲۰۰۸) یک روش فراابتکاری مبتنی بر الگوریتم جستجوی ممنوع برای زمانبندی قطعات و کارها در جهت کمینه کردن زمان کل ساخت در سیستم تولید سلولی با توجه به زمان‌های آماده‌سازی خانواده قطعات وابسته به توالی ارائه کرده‌اند. در این مقاله، کارایی و اثربخشی روش فراابتکاری جستجوی ممنوع پیشنهاد شده در برابر سایر الگوریتم‌های ابتکاری و فراابتکاری معرفی شده برای حل این مسأله مقایسه شد و نتایج محاسباتی نشان داد که الگوریتم جستجوی پراکنده پیشنهادی در کمینه‌سازی زمان کل ساخت در یک زمان پردازش معقول، کاملاً اثربخش است.

مانکن و همکاران (۲۰۰۸) در یک کارخانه مونتاژ قطعات الکترونیکی یک روش ابتکاری سه مرحله‌ای برای زمانبندی مونتاژ گروهی با خطوط مونتاژ موازی و مجزا با معیار کمینه‌سازی هزینه راه‌اندازی ارائه کرده‌اند. فرآیند الگوریتم ابتکاری، شامل قدم‌های تخصیص، توالی و زمانبندی با یک رویکرد بهینه‌سازی برای هر قدم است. برای پیچیده‌ترین مرحله، یعنی مرحله توالی رویه‌ای با نام رویه جستجوی تطابقی تصادفی توانمند<sup>۴</sup> (GRASP) ارائه شده است. با توجه به هزینه‌های راه‌اندازی وابسته به توالی و کاربرد رویه جستجوی تطابقی تصادفی توانمند نتایج تجربی یک کاهش ۴۹

ذرات انبوه) پیشنهادی برای حل مسأله مورد نظر پرداخته می‌شود. نتایج محاسباتی در بخش پنجم و سپس نتیجه‌گیری در بخش ششم آورده می‌شود.

## ۲. مرور ادبیات موضوع

یوشیدا و هیتومی (۱۹۷۹) الگوریتمی برای حل بهینه مسأله زمانبندی فلوشاپ دو ماشینیه ارائه داده‌اند که در آن زمان راه‌اندازی در نظر گرفته شده است. یانگ و لیاو (۱۹۹۶) مسأله زمانبندی سلولی را با دو سلول و حرکت بین سلولی در نظر گرفتند و آن را با استفاده از تکنیک شاخه و حد و یک الگوریتم ابتکاری حل نمودند. سلیمانپور و همکاران (۲۰۰۴) مسأله زمانبندی سلولی را با احتساب زمان راه‌اندازی در نظر گرفتند و الگوریتم حلی ارائه نمودند. در این مقاله یک سیاست دو مرحله‌ای در نظر گرفته شد: در مرحله اول توالی کارها در هر سلول و در مرحله دوم توالی سلول‌ها با توجه به ماشین‌های گلوگاه<sup>۱</sup> و مقایسات جفتی مشخص شد. در نهایت، مدل خود را از طریق الگوریتم دو مرحله‌ای SVS حل کرده است و نتایج به‌دست آمده را با الگوریتم LN-PT مقایسه کرده و به این نتیجه رسیده است که الگوریتم SVS کارتر از الگوریتم LN-PT است. لو و یوان (۲۰۰۷) مسأله دسته‌بندی تک ماشینیه با زمان‌های راه‌اندازی خانواده قطعات مساوی را به منظور کمینه کردن حداکثر دیرکرد را در نظر گرفته و حل کردند.

لین و همکاران، (۲۰۰۹) سه روش فراابتکاری برجسته (یعنی الگوریتم ژنتیک، شبیه‌سازی تبرید<sup>۲</sup> (SA) و جستجوی ممنوع<sup>۳</sup> (TS)) برای مسأله

<sup>۱</sup> Bottleneck

<sup>۲</sup> Simulated Annealing

<sup>۳</sup> Tabu Search

<sup>۴</sup> Greedy Randomized Adaptive Search Procedure

درصدی در هزینه‌های راه‌اندازی را نشان می‌دهد. توکلی مقدم و همکاران، (۲۰۱۰) روش فراابتکاری جستجوی پراکنده را برای مسأله زمانبندی سیستم تولید سلولی چند معیاره با معیارهای هزینه‌های راه‌اندازی، ساخت، جابجایی بین سلولی بررسی کردند و نشان دادند که روش جستجوی پراکنده در کمینه سازی معیارهای فوق در یک زمان پردازش معقول، کاملاً اثربخش است.

جرالد و همکاران (۲۰۰۵) به زمانبندی در سیستم تولیدی انعطاف پذیر<sup>۱</sup> (FMS) با به کارگیری الگوریتم بهینه سازی ذرات انبوه پرداختند. همچنین، برای مقایسه کارایی این الگوریتم از سایر تکنیک‌های فرا-ابتکاری از قبیل الگوریتم ژنتیک، شبیه‌سازی تبرید و الگوریتم ممیتیک<sup>۲</sup> استفاده نمودند. توابع هدف در نظر گرفته شده شامل کمینه ساختن زمان بیکاری ماشین و کمینه ساختن هزینه جریمه کل است که برای برآورده نمودن موعد تحویل محصول در نظر گرفته شده است. نتایج حاصل از اعمال و پیاده سازی الگوریتم بهینه‌سازی ذکر شده، حاکی از این است که الگوریتم بهینه‌سازی ذرات انبوه نسبت به سایر الگوریتم‌ها جواب بهتر و در زمانی مناسبتر ارائه داده است. تاسگترین و همکاران (۲۰۰۷) از الگوریتم بهینه‌سازی ذرات انبوه برای مدل تولید خطی استفاده نمودند. آنها با فرموله کردن مسأله به صورت یک مدل دودویی، ابتدا مدل پیشنهادی را توسط الگوریتم بهینه‌سازی ذرات انبوه دودویی حل کرده و سپس نتایج حاصل را با جواب‌های الگوریتم ژنتیک مقایسه نمودند. در ۲۰ مورد مسأله حل شده، الگوریتم بهینه‌سازی ذرات انبوه موفق به یافتن ۱۹ مورد از

جواب‌های بهینه گردید در حالیکه الگوریتم ژنتیک تنها توانست ۱۴ مورد از جواب‌های بهینه را به دست آورد. ونگ و لیو (۲۰۰۸) به ارایه روشی برای زمانبندی در سیستم تولید انعطاف پذیر با وجود محدودیت منابع بر مبنای الگوریتم بهینه‌سازی ذرات انبوه و الگوریتم شبیه‌سازی تبرید پرداختند. در مقاله مذکور ابتدا یک مسأله زمانبندی در سیستم تولید انعطاف پذیر با وجود محدودیت منابع در نظر گرفته شد و از آن یک تابع هزینه ایجاد گردید. سپس یک الگوریتم تلفیقی از الگوریتم بهینه‌سازی ذرات انبوه و الگوریتم شبیه‌سازی تبرید برای دستیابی به جواب بهینه پیاده شد. نتایج الگوریتم شبیه‌سازی تبرید نشان می‌دهد که الگوریتم پیشنهادی توانایی گریز از بهینگی نسبی و جهت‌گیری به سوی بهینه کلی را دارد و در مقایسه با سایر روش‌ها کاراست. لطفی و همکاران (۲۰۰۸) به زمانبندی سیستم تولیدی توسط یک الگوریتم ژنتیک جدید که از الگوریتم بهینه‌سازی ذرات انبوه و همچنین از یک الگوریتم ژنتیک معمولی الهام گرفته شده، برای شناسایی پتانسیل مفاهیم الگوریتم بهینه‌سازی ذرات انبوه در بهبود عملکرد الگوریتم ژنتیک، پرداختند. در الگوریتم ژنتیک تغییر یافته، کروموزوم‌های اولیه، مانند ذرات در الگوریتم بهینه‌سازی ذرات انبوه ایجاد می‌شوند و برای نسل‌های بعدی از عملگرهای الگوریتم ژنتیک معمولی استفاده می‌شود. همچنین، همانند الگوریتم بهینه‌سازی ذرات انبوه، هریک از کروموزوم‌ها بهترین تجربه شخصی خود و بهترین تجربه کلیه کروموزوم‌ها را حفظ می‌نمایند. نتایج به دست آمده نشان می‌دهد که الگوریتم ارایه شده از عملکرد بهتری برخوردار است.

<sup>1</sup> Flexible Manufacturing System

<sup>2</sup> Memetic Algorithm

### ۳. مسأله زمانبندی سیستم تولید سلولی چند هدفه

#### ۳-۱. تشریح مسأله

در یک مسأله زمانبندی سیستم تولید سلولی،  $m$  ماشین و  $n$  قطعه موجود است. باید توجه داشت که این  $m$  ماشین در  $c$  سلول مستقر بوده؛ از هر کدام یک عدد موجود است. هر کار یا قطعه دارای یک فرآیند خاص است و توالی انجام عملیات مختلف به منظور تکمیل آن کار مشخص است. هر عملیات یک قطعه بر روی هر ماشین، دارای زمان خاص خود است. در این مسأله، عملیات مربوط به پردازش هر قطعه بر روی ماشین‌های مورد نیاز برای پردازش در سلولی که ماشین مستقر است، زمانبندی می‌گردد. هدف از حل این مسأله مشخص کردن توالی انجام کارها در هر سلول و همچنین تعیین توالی سلول‌هاست. به طوریکه: (۱) کاهش حرکت بین سلولی و (۲) کاهش هزینه زودکرد قطعات و (۳) کاهش هزینه دیرکرد قطعات، بهینه شود. در این مسأله  $n$  قطعه وجود دارد که هر یک از آن‌ها در یک زمان واحد باید روی  $m$  ماشین یا کمتر پردازش شوند. هر کار از یک توالی از پیش تعیین شده‌ای پیروی کرده و زمان پردازش مشخصی دارد. ترتیب ماشین‌ها برای کارهای مختلف متفاوت است. برای درک بیشتر مفهوم توالی بین سلول‌ها مفروضات در نظر گرفته شده برای این مدل عبارتند از:

- زمان عملیات تمامی قطعات روی هر نوع ماشین قطعی است.
- تعداد قطعات مشخص و ثابت است.
- تعداد ماشین‌های موجود در سیستم مشخص و ثابت است.

- تعداد سلول‌های موجود در سیستم مشخص و ثابت است.

- ماشین‌های موجود در هر سلول مشخص است.
- هر نوع ماشین می‌تواند تنها یک نوع عملیات را انجام دهد و هر عملیات نیز تنها می‌تواند توسط یک ماشین انجام شود.
- مسافت بین سلول‌ها ثابت نیست.
- زمان شکست برای ماشین‌ها نخواهیم داشت.
- راندمان ماشین‌ها ۱۰۰٪ است.
- تمامی ماشین‌ها در ابتدای دوره‌ها برای استفاده در دسترس هستند.
- زمان‌های راه‌اندازی برای هر یک از ماشین‌ها صفر است.

#### ۳-۲. اندیس‌ها

$j$ : اندیس ماشین،  $j = \{1, \dots, M\}$

$i$ : اندیس قطعات،  $i = \{1, \dots, P\}$

$c$ : اندیس سلول،  $c = \{1, \dots, C\}$

$k$ : اندیس توالی قطعات،  $k = \{1, \dots, K\}$

$b$ : اندیس توالی سلول،  $b = \{1, \dots, KC\}$

#### ۳-۳. پارامترها

$t_{ij}$ : زمان لازم برای پردازش قطعه  $i$  بر روی ماشین  $j$

۱ اگر قطعه  $i$  به ماشین  $j$  جهت پردازش نیاز داشته باشد،  
 ۰ در غیر اینصورت. }  $a_{ij}$

۱ اگر ماشین  $j$  به سلول  $c$  تعلق داشته باشد،  
 ۰ در غیر این صورت. }  $m_{jc}$

$T_{ic}$ : زمان حرکت بین سلولی برای قطعه  $i$  در سلول  $c$

$L_i$ : هزینه دیرکرد قطعه  $i$

$E_i$ : هزینه زودکرد قطعه  $i$   
 $d_i$ : زمان تحویل قطعه  $i$   
 ۳-۴. متغیرهای تصمیم‌گیری

۱ } اگر قطعه  $i$  در توالی  $k$ ام سلول  $c$  واقع شده باشد،  
 $Z_{ikc}$  } در غیر اینصورت.

$C_{kijcb}$ : زمان تکمیل قطعه واقع در توالی  $k$ ام روی ماشین  $j$  در سلول  $c$  که سلول  $c$  در توالی  $b$  واقع است.

$C_i$ : زمان تکمیل قطعه  $i$   
 $C_{max}$ : حداکثر زمان تکمیل کارها

۱ } اگر قطعه  $i$  به سلول  $c$  اختصاص یابد،  
 $X_{ic}$  } در غیر این صورت.

۱ } اگر سلول  $c$  به توالی  $b$  اختصاص یابد،  
 $Y_{cb}$  } در غیر این صورت.

### ۳-۵. مدل ریاضی

مدل ریاضی سه هدفه پیشنهادی به صورت زیر ارائه می‌شود:

$$\min Z_1 = C_{max} \quad (1)$$

$$\min Z_2 = \sum_{i=1}^p E_i \cdot \max\{0, d_i - C_i\} \quad (2)$$

$$\min Z_3 = \sum_{i=1}^p L_i \cdot \max\{0, C_i - d_i\} \quad (3)$$

S.t.

$$\sum_{c \in C} x_{ic} = 1 \quad ; \forall i \in P \quad (4)$$

$$\sum_c y_{cb} = 1 \quad ; \forall b \in K_c \quad (5)$$

$$\sum_b y_{cb} = 1 \quad ; \forall c \in C \quad (6)$$

$$\sum_k z_{ikc} = x_{ic} \quad ; \forall i \in P \ \& \ \forall c \in C \quad (7)$$

$$\sum_i z_{ikc} \leq 1 \quad ; \forall k \in K \ \& \ \forall c \in C \quad (8)$$

$$C_{11c1} = \max(\forall i \in P : t_{i1} \times z_{i1c} \times y_{c1}) \quad ; \forall c \in C \quad (9)$$

$$C_{k1c1} = c_{k-1,1,c,1} + \max(\forall i \in P : t_{i1} \times z_{i,k,C} \times y_{c1}) \quad ; \quad \forall k \geq 2 \in K \ \& \ c \in C \quad (10)$$

$$C_{11Cb} = \max(\forall n \in C : C_{K,1,n,b-1}) + \max(\forall i \in P : t_{i1} \times z_{i,1,C} \times y_{Cb}) \quad ; \quad \forall b \geq 2 \in K_c \ \& \ c \in C \quad (11)$$

$$C_{k1cb} = C_{k-1,1,c,b} + \max(\forall i \in P : t_{i1} \times z_{ikC} \times y_{cb}) \quad ; \quad \forall b \geq 2 \in K_c \ \& \ c \in C \ \& \ k \geq 2 \in K \quad (12)$$

$$C_{1jcl} = C_{1,j-1,c,1} + \max(\forall i \in P : t_{ij} \times z_{i1c} \times y_{cl}) \quad ; \forall j \geq 2 \in M \ \& \ \forall c \in C \quad (13)$$

$$C_{1jcb} = \max(\max(\forall n \in C : C_{K,j,n,b-1}), C_{1,j-1,c,b}) + \max(\forall i \in P : t_{ij} \times z_{i1c} \times y_{cb}) \quad ; \quad (14)$$

$$\forall j \geq 2 \in M \ \& \ \forall c \in C \ \& \ b \geq 2 \in K_c$$

$$C_{kijcb} = \max(C_{k,j-1,c,b}, C_{k-1,j,c,b}) + \max(\forall i \in P : t_{ij} \times z_{ikc} \times y_{cb}) \quad ; \quad (15)$$

$$\forall j \geq 2 \in M \ \& \ k \geq 2 \in K \ \& \ c \in C \ \& \ b \in K_c$$

$$C_i = \max(\forall k \in K, j \in M, c \in C \& b \in K_c : c(k, j, c, b) \times z_{ikc} \times y_{cb} + \sum_{c=1}^C (\sum_{j=1}^M a_{ij} \times |a_{ij} - m_{jc}|) \times x_{ic} \times T_{ic}) ; \forall i \in P \quad (16)$$

$$C_{\max} = \max(C_{k,j,c,b}) + \sum_{i=1}^P \sum_{c=1}^C (\sum_{j=1}^M a_{ij} \times |a_{ij} - m_{jc}|) \times x_{ic} \times T_{ic}; \quad \forall k \in K, j \in M, c \in C \& b \in K_c \quad (17)$$

$$x_{ic} \& y_{cb} \& z_{ikc} \text{ binary} ; \forall i \in P, k \in K, c \in C \& b \in K_c \quad (18)$$

$$C_{k,j,c,b}, C_{\max} \geq 0 ; \forall j \in M, k \in K, c \in C \& b \in K_c \quad (19)$$

آخر سلول قبلی و زمان پردازش قطعه در توالی فعلی روی ماشین اول قرار می‌دهد. معادله (۱۲) زمان تکمیل اختصاص داده شده به توالی  $k$  روی ماشین اول در سلولی که در سایر توالی‌ها غیر از توالی اول قرار دارند را برابر با مجموع زمان تکمیل توالی قبلی و زمان پردازش قطعه مشخص شده در توالی فعلی روی ماشین اول قرار می‌دهد. رابطه (۱۳) زمان تکمیل قطعه اختصاص داده شده به توالی اول روی ماشین  $z$  (غیر از ماشین اول) در سلولی که به توالی اول اختصاص دارد را برابر با مجموع زمان تکمیل این قطعه روی ماشین  $1-z$  در همان توالی و زمان پردازش قطعه در همان توالی روی ماشین  $z$  قرار می‌دهد.

معادله (۱۴) زمان تکمیل قطعه اختصاص داده شده به توالی اول روی ماشین  $z$  (غیر از ماشین اول) در سلولی که به توالی غیر از توالی اول اختصاص داده شده است را برابر مجموع بیشترین زمان تکمیل این قطعه روی ماشین قبلی ( $j-1$ ) در سلول فعلی و زمان تکمیل توالی آخر سلول قبلی ( $b-1$ ) روی ماشین  $z$  با زمان پردازش قطعه توالی اول در سلول فعلی ( $b$ ) روی ماشین  $z$  قرار می‌دهد. رابطه (۱۵) زمان تکمیل قطعه اختصاص داده شده به توالی  $k$  (غیر از توالی اول) روی ماشین  $z$  (غیر از ماشین اول) در سلول  $c$  که به توالی  $b$  اختصاص دارد را برابر با مجموع بیشترین زمان تکمیل توالی قبلی ( $k-1$ ) روی

تابع هدف اول (۱)، حداکثر زمان تکمیل کارها را کمینه می‌کند. تابع هدف دوم (۲)، هزینه زودکرد تولید قطعات و تابع هدف سوم (۳)، هزینه دیرکرد تولید قطعات را کمینه می‌کند.

معادله (۴) هر یک از قطعات را به یک سلول تخصیص می‌دهد. رابطه (۵) تضمین می‌کند که به هر توالی فقط یک سلول اختصاص یابد. معادله (۶) تضمین می‌کند که هر سلول تنها به یک توالی اختصاص یابد. رابطه (۷) هر یک از قطعات تخصیص داده شده به هر سلول را به یک توالی در آن سلول اختصاص می‌دهد. معادله (۸) تضمین می‌کند که هر توالی در یک سلول حداکثر به یک قطعه اختصاص یابد. رابطه (۹) زمان تکمیل قطعه اختصاص داده شده به اولین توالی بر روی اولین ماشین در سلولی که در توالی اول قرار گرفته را برابر با مجموع زمان پردازش قطعه مشخص شده روی ماشین اول در سلول مشخص شده قرار می‌دهد.

معادله (۱۰) زمان تکمیل قطعه اختصاص داده شده به توالی  $k$  ( $k \geq 2$ ) روی ماشین اول در سلولی که در توالی اول قرار گرفته را برابر با مجموع زمان تکمیل در توالی قبلی و زمان پردازش در توالی فعلی قرار می‌دهد. رابطه (۱۱) زمان تکمیل قطعه اختصاص داده شده به توالی اول روی ماشین اول در سلول‌هایی که در سایر توالی‌ها ( $b \geq 2$ ) غیر از توالی اول قرار دارند را برابر با مجموع زمان تکمیل در توالی

قطعات بر روی ماشین پردازش می‌شوند. برای درک بیشتر از مفهوم توالی بین سلول‌ها در این قسمت از مقاله، مثالی ارائه می‌شود. یک شرکت تولیدی که شامل واحدهای مختلفی است، در واحد رنگ آن شرکت، چهار نوع قطعه تولیدی رنگ می‌شود. این واحد دارای دو سلول تولیدی بوده که پنج ماشین آن در این دو سلول مستقر شده‌اند. جدول شماره ۱ ماتریس مسیر ساخت قطعات بوده که نشان دهنده ماشین‌های مورد نیاز برای پردازش بر روی قطعات است. همان‌طور که مشخص است، قطعه ۱ به ماشین‌های ۱، ۲ و ۳ نیاز دارد. جدول شماره ۲ زمان پردازش هر یک از قطعات را نشان می‌دهد. همان‌طور که مشخص است، زمان پردازش قطعه ۱ روی ماشین ۱، ۲ و ۳ به ترتیب ۴، ۵ و ۱۲ است. و در نهایت، تعلق ماشین‌های واحد رنگ شرکت به هر یک از دو سلول در جدول شماره ۳ آمده است.

ماشین  $k$  و زمان تکمیل توالی فعلی  $(k)$  روی ماشین قبلی  $(j-1)$  با زمان پردازش قطعه مشخص شده در توالی فعلی روی ماشین  $k$  قرار می‌دهد.

معادله (۱۶) زمان تکمیل هر قطعه را محاسبه می‌کند. رابطه (۱۷)  $C_{max}$  را برابر با حداکثر زمان تکمیل قرار می‌دهد. رابطه (۱۸) متغیرهای ۰ و ۱ را معرفی می‌کند و در نهایت، رابطه (۱۹) متغیرهای غیر منفی را معرفی می‌کند.

همان‌طور که اشاره شد، تابع هدف اول مدل کمینه‌سازی حداکثر زمان تکمیل کارهاست و هدف در مسأله زمانبندی سلولی تعیین توالی قطعات در سلول‌ها و تعیین توالی سلول‌هاست. بدین معنی که در شرایطی که چند قطعه به یک ماشین اختصاص می‌یابند، ابتدا قطعه‌ای روی آن ماشین پردازش می‌شود که سلول آن دارای شماره توالی کمتری بوده و توالی آن قطعه در سلول خود نیز دارای کمترین شماره توالی باشد و متناسب با این رویکرد سایر

جدول ۱. ماتریس مسیر ساخت قطعات  $(a_{ij})$

ماشین	۱	۲	۳	۴	۵
قطعه ۱	۱	۱	۱	۰	۰
قطعه ۲	۱	۱	۰	۰	۰
قطعه ۳	۰	۰	۱	۱	۱
قطعه ۴	۰	۰	۱	۱	۰

جدول ۲. زمان پردازش قطعات بر روی ماشین‌ها  $(t_{ij})$

ماشین	۱	۲	۳	۴	۵
قطعه ۱	۴	۵	۱۲	۰	۰
قطعه ۲	۶	۴	۰	۰	۰
قطعه ۳	۰	۰	۷	۶	۱۴
قطعه ۴	۰	۰	۵	۳	۰



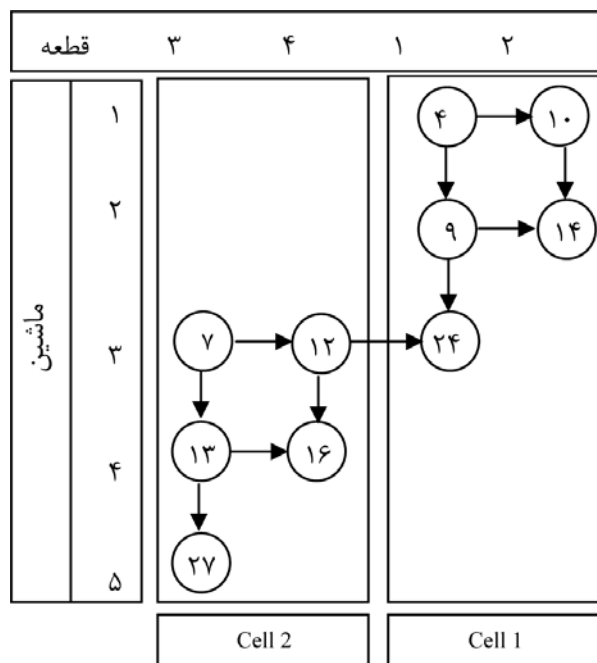
جدول ۳. ماتریس تخصیص ماشین‌ها به سلول‌ها ( $m_{jc}$ )

ماشین	۱	۲	۳	۴	۵
سلول ۱	۱	۱	۰	۰	۰
سلول ۲	۰	۰	۱	۱	۱

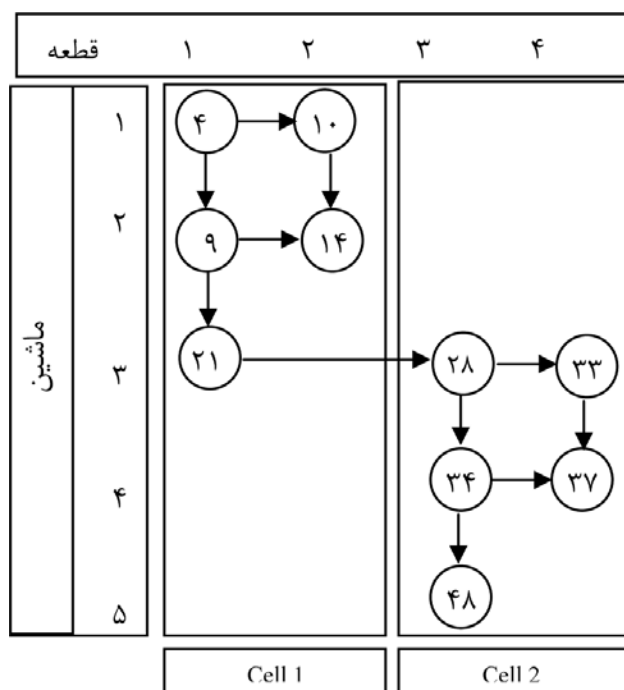
قطعات ۱ و ۲ به سلول ۲ و قطعات ۳ و ۴ به سلول ۱ اختصاص یافته‌اند و نهایت توالی سلول‌ها بدین صورت است که سلول ۲ در توالی اول بوده و سلول ۱ در توالی دوم. حال در صورتی که توالی سلول‌ها تغییر یابد؛ یعنی سلول ۲ در توالی اول قرار گیرد و سلول ۱ در توالی دوم، مدت زمان کل ساخت ۴۸ می‌شود. نمایی شماتیک از زمانبندی صورت گرفته برای این توالی در شکل ۲ آمده است.

با توجه به مسأله مطرح شده و حل مدل با در نظر گرفتن تابع هدف اول که حداقل‌سازی حداکثر زمان تکمیل کارها است، بهترین توالی به دست آمده با استفاده از نرم افزار Lingo 11 در شکل ۱ نشان داده شده است.

مدت زمان کل ساخت برای این مسأله ۲۷ بوده ( $C_{max} = 27$ ) و با توجه به تخصیص قطعات، یک حرکت بین سلولی وجود دارد. با توجه به نتایج حل،



شکل ۱. شمایی از توالی قطعات و سلول‌ها در شرایط اول از آزمون ارابه شده



شکل ۲. شمایی از توالی قطعات و سلول‌ها در شرایط دوم از آزمون ارایه شده

ژنتیک و جستجوی همسایگی متغیر ترکیب شده، برای حل مسأله زمانبندی سلولی پیشنهاد شده است. شکل ۳ ساختار کلی طراحی شده برای الگوریتم بهینه‌سازی ذرات انبوه چند هدفه را نشان می‌دهد.

۴. طراحی الگوریتم بهینه‌سازی ذرات انبوه برای حل مدل پیشنهادی

در این مقاله، یک الگوریتم بهینه‌سازی ذرات انبوه چند هدفه مبتنی بر آرشیو پارتو که با عملگرهای

The pseudo-code of our algorithm is as follows:  
 {Initialize search parameters  
 Determine the approximate dynamic ideal point  
 Generate  $N$  initial particles with elite tabu search  
 Evaluate the initial particles to get the local best  $P_i$  and the global best  $P_g$   
 Initialize the adaptive Pareto archive set so that it is empty  
**While** a given maximal number of iterations is not achieved.  
   Improve particles by VNS  
   Update the adaptive Pareto archive set  
   Update particle position  $X$  according to the  
     
$$s_i^{k+1} = (p_i^k - s_i^k) + (p_g^k - s_i^k) + \bar{s}_i^k$$
  
   Evaluate updated particles to get the new  $P_i$  and  $P_g$   
   Update the value of the dynamic ideal point  
**End while**}

شکل ۳. ساختار کلی الگوریتم پیشنهادی

#### ۴-۱. نحوه نمایش جواب‌ها

در تمام الگوریتم‌های فراابتکاری، به دلیل نیاز به حل شدن در شروع کار، لازم است حل شدن را طبق ساختار مشخصی ذخیره کنیم که به این ساختار، نحوه نمایش جواب می‌گویند. در اینجا برای نمایش جواب از ساختار ماتریسی استفاده شده است. نمایش جواب در این مسأله، شامل دو ماتریس است که یکی برای نمایش توالی سلول‌ها و دیگری برای نمایش توالی کارها در سلول‌ها طراحی شده است. شکل شماره‌های ۲ و ۳ به ترتیب ماتریس سطری (برداری) و ماتریس دو بعدی را نشان می‌دهد. در شکل ۴،  $C_j$  نشان دهنده شماره سلولی است که در خانه  $j$ ام توالی زمانبندی شده است. در شکل شماره ۵،  $A_{ij}$  شماره کاری است که در سلول  $i$ ام در خانه  $j$ ام توالی زمانبندی شده است.

$C_1$	$C_2$	$C_3$		$C_j$		$C_{c-1}$	$C_c$
-------	-------	-------	--	-------	--	-----------	-------

شکل ۴. ماتریس برداری نمایش توالی سلول‌ها

$A_{11}$	$A_{12}$		$A_{1j}$		
$A_{i1}$			$A_{ij}$		
$A_{c1}$					

شکل ۵. ماتریس دو بعدی نمایش توالی کارها در

سلول‌ها

#### ۴-۲. نحوه تولید جواب‌های اولیه

در این مقاله، از یک روش جستجوی ممنوع نخبه‌گرا<sup>۱</sup> برای تولید جواب‌های آغازین استفاده می‌شود. هدف اصلی به کارگیری این روش، تولید جواب‌های اولیه‌ای است که از نظر کیفیت<sup>۲</sup> جواب و پراکندگی<sup>۳</sup>، از سطح مطلوب مناسبی برخوردار باشند. جواب آغازین می‌تواند هر یک از جواب‌های حاصل از بهینه‌سازی هر یک از توابع هدف در زمان متوقف شدن نرم افزار لینگو باشد (نقطه آرمانی پویا). روش پیشنهادی در ابتدا، جواب موجود را در یک فهرست مجازی ذخیره کرده، سپس با استفاده از یک رویکرد حرکت مناسب به جستجوی جواب مطلوب دیگری که شرط (شرایط) پذیرش را دارا باشد، در مجاورت جواب می‌پردازد. این فرایند تا دستیابی الگوریتم به شرط پایانی ادامه پیدا می‌کند. رویکرد حرکت مورد استفاده، عملگر شیفت به راست است، به این صورت که دو اندیس بصورت تصادفی تولید شده و توالی بین این دو اندیس به سمت راست به صورت چرخشی شیفت داده می‌شوند. رویکرد حرکت طراحی شده از استراتژی هوشمند روش جستجوی ممنوع، مبتنی بر پرهیز از بازگشت به جواب‌هایی که در تکرارهای قبلی به دست آمده‌اند، از طریق بکارگیری یک فهرست مجازی حافظه به نام فهرست ممنوع استفاده می‌کند. فهرست ممنوع مورد استفاده در اینجا، به مسیر جستجو وابسته بوده، از تکرار یک حرکت در طول دوره ممنوعیت<sup>۴</sup> جلوگیری می‌کند. برای دستیابی به جواب‌های مطلوب از نظر کیفیت و

<sup>1</sup> Elite Tabu Search

<sup>2</sup> Quality

<sup>3</sup> Diversity

<sup>4</sup> Tabu Tenure

پراکنندگی در مسیر جستجو، از رابطه (۲۰) استفاده می‌گردد.

$$\xi = \sum_{i=1}^k \frac{|f_i - F_i|}{w_i} \quad (20)$$

$f_i$ : مقدار  $i$ امین تابع هدف توالی تولید،  $F_i$ : مقدار  $i$ امین مؤلفه نقطه آرمانی پویا و  $w_i$ : وزن تخصیص یافته به  $i$ امین تابع هدف است. هدف از به کارگیری تابع فوق بر پایه این حقیقت استوار است که اگر تابع فوق به ازای یک مجموعه از وزن‌های  $w_i$  برای یک جواب کمینه گردد، آن جواب متعلق به جواب‌های پارتو است.

برای تعیین شرط پذیرش جواب جدید، متغیر  $\eta$  به صورت زیر تعریف می‌گردد:

$$\eta = \xi_B - \xi_A \quad (21)$$

در رابطه (۲۱)،  $A$  جواب موجود و  $B$  جوابی است که توسط روش جستجوی ممنوع و با به کارگیری رویکرد حرکت به‌دست آمده است. بنابراین شرایط پذیرش بصورت یکی از موارد زیر است:

(۱) اگر  $\eta \leq 0$  و حرکت در فهرست ممنوع یافت نشود، جواب  $B$  جایگزین جواب  $A$  می‌گردد.

(۲) اگر  $\eta \leq 0$  و حرکت در فهرست ممنوع یافت شود، از استراتژی تنفس استفاده شده و جواب  $B$  جایگزین جواب  $A$  می‌گردد.

(۳) اگر  $\eta \geq 0$  و حرکت در فهرست ممنوع یافت نشود، جواب  $B$  در صورتی جایگزین جواب  $A$  می‌شود که توسط آن مغلوب نشده باشد.

(۴) اگر  $\eta \geq 0$  و حرکت در فهرست ممنوع یافت شود، جواب  $A$  بدون تغییر باقی خواهد ماند.

استراتژی تنفس<sup>۱</sup> در این مقاله به این صورت پیاده‌سازی شده است که اگر جواب به‌دست آمده توسط هیچ یک از جواب‌های پذیرفته شده مغلوب نشود، پذیرفته خواهد شد.

روش جستجوی ممنوع نخبه‌گرا پس از  $\alpha \times N$  تکرار متوقف می‌گردد. پس از اجرای الگوریتم جستجوی ممنوع نخبه‌گرا،  $\alpha \times N$  جواب پراکنده که تا حد امکان به مرز بهینه نزدیک شده‌اند، وجود دارد. از طرفی تعداد جواب‌ها در هر تکرار روش الگوریتم بهینه‌سازی ذرات انبوه چند هدفه، برابر با  $N$  است. بنابراین باید از بین  $\alpha \times N$  جواب موجود،  $N$  جواب به عنوان توالی‌های تولید اولیه الگوریتم بهینه‌سازی ذرات انبوه انتخاب گردند. در این مقاله، انتخاب جمعیت اولیه جواب‌ها، بر اساس یک روش سریع مرتب کردن جواب‌های غیرمغلوب که توسط دب (۲۰۰۲) شرح داده شده است، صورت می‌گیرد. این روش به این صورت عمل می‌کند که  $\alpha \times N$  جواب موجود با الگوریتمی که توسط دب (۲۰۰۲) طراحی شده است، مرتب و سطح‌بندی می‌شوند؛ شماره هر سطح، کیفیت جواب‌های موجود در آن را نشان می‌دهد. در نهایت  $N$  جواب که دارای بالاترین کیفیت و پراکنندگی هستند، مطابق با قانون دب انتخاب می‌شوند.

برای به روز رسانی مقادیر مؤلفه‌های نقطه آرمانی پویا، فرآیند زیر در انتهای هر تکرار روش پیشنهادی انجام می‌شود:  $N$  مقدار به‌دست آمده برای هر یک از سه تابع هدف، جداگانه و به صورت صعودی مرتب می‌شوند. اگر کوچکترین مقدار به‌دست آمده برای تابع هدف  $i$ ام از مؤلفه متناظرش در نقطه آرمانی پویا

<sup>۱</sup> Aspiration

کوچکتر باشد، این مقدار جایگزین می‌گردد و در غیر این صورت مؤلفه مورد نظر تغییر نمی‌کند.

#### ۳-۴. آرشیو جواب‌های پارتو

از آنجا که در حل مسایل چندهدفه به دلیل وجود تناقض بین اهداف، جواب واحدی که همه اهداف در آن بهینه باشند، وجود ندارد، در نهایت مجموعه‌ای از جواب‌های غالب به عنوان جواب‌های بهینه (نزدیک بهینه)، ارایه خواهند شد که به آنها آرشیو جواب‌های پارتو گویند. در اینجا که از یک روش مبتنی بر آرشیو پارتو برای حل استفاده شده است، کیفیت جواب‌های موجود در آرشیو، بسیار اهمیت دارد. لذا در این مقاله، این آرشیو در هر تکرار الگوریتم پیشنهادی به روز خواهد شد. روش به روز کردن آن در اینجا به گونه‌ای طراحی شده است که در آن جواب‌هایی که دارای کمترین فاصله با مرز بهینه و همچنین بالاترین پراکندگی هستند، در این آرشیو نگهداری می‌شوند. برای پیاده سازی این تابع از روش تخصیص برازندگی دانه دانه ریز شده<sup>۱</sup> همراه با روش تخمین چگالی و یک تابع بهبود یافته برش<sup>۲</sup> آرشیو پارتو استفاده شده است (زیتزلر و همکاران، ۲۰۰۱). در این روش، سائز آرشیو ثابت بوده، به همه جواب‌ها، چه جواب‌های غالب و چه مغلوب، اهمیت داده می‌شود.

#### ۴-۴. تعیین $p_i$ و $p_g$ اولیه

برای هر کدام از جواب‌ها بهترین جوابی را که تا به حال به آن رسیده، به عنوان  $p_i$  تعیین می‌کنیم و  $p_g$

بهترین نقطه یا جوابی است که همه جواب‌ها یا ذرات به آن رسیده اند.

#### ۵-۴. روش بهبود جواب‌ها به وسیله جستجوی

##### همسایگی متغیر (VNS)

هر کدام از ذرات یا جواب‌ها که در جمعیت جواب‌ها موجود است، توسط این رویکرد بهبود داده می‌شوند. این رویکرد با یک جواب اولیه شروع می‌کند که در این کار برای هر کدام از جواب‌ها اجرا می‌شود. از آنجایی که روش جستجوی همسایگی متغیر هر جواب ورودی را بواسطه ساختارهای مختلف جستجوی همسایگی بهبود می‌دهد، چهار ساختار جستجوی همسایگی (NSS) برای رویکرد جستجوی همسایگی متغیر تعریف می‌شود:

۱- در جستجوی همسایگی اول، دو اندیس (برای هر دو ماتریس) بطور تصادفی ایجاد شده و برای ماتریس اول (یک بعدی) خانه با اندیس اول قبل از خانه با اندیس دوم درج می‌شود و برای ماتریس دوم (دو بعدی) ستون با اندیس اول قبل از ستون با اندیس دوم درج می‌شود.

۲- در جستجوی همسایگی دوم، چهار اندیس بطور تصادفی تولید شده و دو اندیس اول قبل از دو اندیس دوم درج می‌شوند.

۳- در جستجوی همسایگی سوم، شش اندیس بطور تصادفی تولید شده و سه اندیس اول قبل از سه اندیس دوم درج می‌شوند.

۴- در جستجوی همسایگی چهارم، دو اندیس چندین بار تولید می‌شود و در هر تکرار خانه‌های این دو اندیس جابه‌جا می‌شوند.

<sup>1</sup> Fine-Grained

<sup>2</sup> Truncation Method

$s_i^k$ : نشان دهنده عملگر جهش<sup>۳</sup> بر روی جواب است. در این مقاله، برای پیاده‌سازی این عملگر از یک جستجوی همسایگی تکرار شونده به همراه جهش هدایت شده استفاده شده است (رحیمی واحد و همکاران، ۲۰۰۸).

۱+ : عملگر انتخاب جواب از بین سه جواب  $p_i^k - s_i^k$  و  $p_g^k - s_i^k$  و  $s_i^k$  است. در پایان هر تکرار، PSO مقادیر  $p_i$  و  $p_g$  به روز رسانی می‌شوند.

## ۵. نتایج محاسباتی

در این بخش، روش پیشنهادی بر روی مسایل با اندازه‌های کوچک و بزرگ اجرا گردیده است. برای اثبات کارایی روش پیشنهادی، نتایج حاصل از آنها با جواب‌های تولید شده توسط الگوریتم ژنتیک مرتب شده غیر مغلوب (NSGA-II) مقایسه شده است.

### ۵-۱. مفروضات کلی الگوریتم‌ها

در هر دو گروه از مسایل، مفروضات کلی زیر در نظر گرفته می‌شود:

برای هر یک از مسایل تولید شده، هر یک از الگوریتم‌ها پنج بار اجرا گردیده، نتایج به صورت متوسط مقدار حاصل از این اجراها بیان می‌گردد.

- مقدار  $\alpha$  برابر با سه در نظر گرفته می‌شود.
- تعداد تکرارهای محلی (Local\_iteration) برابر با پنج در نظر گرفته می‌شود.

در رویکرد جستجوی همسایگی متغیر که در آن ساختارهای جستجوی همسایگی مختلف استفاده شده‌اند، پس از هر عملگر جستجوی همسایگی، یک عملگر آشفتگی<sup>۱</sup> بر روی جواب خروجی جستجوی همسایگی اعمال می‌شود. علت استفاده از عملگر آشفتگی این است که در بهینه محلی گیر نکنیم. این عملگر بر اساس ویژگی‌های بهترین جوابی است که تا به حال یافت شده است. این ویژگیها در یک ماتریس احتمالی ذخیره می‌شوند.

### ۴-۶. به روز رسانی مکان ذرات یا جواب‌ها

در حالت کلی، در الگوریتم بهینه‌سازی ذرات انبوه، فضای جواب باید پیوسته باشد، ولی در اینجا برای به روز رسانی مکان ذرات از عملگرهای الگوریتم ژنتیک استفاده شده است و برای همین نیازی نیست که ساختار جواب در فضای پیوسته باشد. نحوه به روز رسانی ذرات یا جواب‌ها مطابق فرمول (۲۲) صورت می‌پذیرد:

$$s_i^{k+1} = (P_i^k - s_i^k) + (P_g^k - s_i^k) + \overline{s_i^k} \quad (22)$$

$s_i^{k+1}$ : جواب نام در تکرار  $k+1$ ام.

$s_i^k$ : جواب نام در تکرار  $k$ ام.

$P_i^k$ : بهترین جوابی که جواب نام به آن رسیده است.

$P_g^k$ : بهترین جوابی که در تکرار  $k$ ام الگوریتم پیدا کرده است.

۱- : عملگر تقاطعی<sup>۲</sup> است که برای طراحی این عملگر از یک عملگر OX استفاده شده است.

<sup>1</sup> Disturb

<sup>2</sup> Crossover

<sup>3</sup> Mutation

بهینه استفاده می‌گردد. تعریف شاخص پراکندگی به صورت زیر است:

$$D = \sqrt{\sum_{i=1}^N \max(\|x_i^i - y_i^i\|)} \quad (23)$$

در رابطه (۲۳)،  $\|x_i^i - y_i^i\|$  نشان دهنده فاصله اقلیدسی بین دو جواب مجاور  $x_i^i$  و  $y_i^i$  بر روی مرز بهینه است.

### ۳-۵. نتایج مقایسه‌ای مسایل با اندازه‌های کوچک و بزرگ

مفروضات روش الگوریتم بهینه‌سازی ذرات انبوه - الگوریتم ژنتیک:

- در مسایل با اندازه‌های کوچک و بزرگ اندازه جمعیت به ترتیب برابر با ۵۰ و ۲۰۰ در نظر گرفته می‌شود.

- اندازه آرشیو پارتو در مسایل با اندازه‌های کوچک و بزرگ به ترتیب برابر با ۵۰ و ۲۰۰ فرض شده است.

- تعداد تکرار در الگوریتم ۱۰۰ بار در نظر گرفته می‌شود.

مفروضات روش الگوریتم ژنتیک مرتب شده غیر مغلوب (NSGA-II):

- تعداد تکرارهای محلی (Local\_iteration) برابر با پنج و تعداد تکرار الگوریتم ۱۵۰ در نظر گرفته می‌شود.

- نرخ عملگر جهش ۰/۲ و عملگر تقاطع ۰/۸ در نظر گرفته شده است.

- اندازه جمعیت در این روش در مسایل کوچک و بزرگ به ترتیب برابر با ۵۰ و ۲۰۰ فرض می‌شود.

- زمان‌های پردازش برای مسایل کوچک در بازه [۱,۴۰] و مسایل با اندازه بزرگ در بازه [۱,۱۰۰] بصورت تصادفی تولید می‌گردند.

- فاصله زمانی بین سلول‌ها به صورت تصادفی در  $[0.2p_{mean}, 0.3p_{mean}]$  تولید می‌شوند. که  $p_{mean}$  میانگین زمان‌های پردازش کارها بر روی ماشین هاست.

- هزینه مربوط به دیرکرد و زودکرد به طور تصادفی در بازه [۱,۲۰] تولید می‌شوند.

- موعد تحویل کارها بر اساس روش ارایه شده توسط لوکیل (۲۰۰۵) تولید گردیده است. هر کدام از مسایل با مقادیر فاکتورهای  $r, t$  که برابرند با  $r=0.6, t=0.4$  حل گردیده‌اند.

### ۲-۵. شاخص‌های مقایسه<sup>۱</sup>

برای ارزیابی کیفیت و پراکندگی الگوریتم‌های فراابتکاری چند هدفه، شاخص‌های متعدد و متنوعی وجود دارند. در این مقاله، برای انجام مقایسه دو شاخص زیر مورد توجه قرار خواهند گرفت.

(۱) **شاخص کیفیت:** این شاخص به مقایسه کیفیت جواب‌های پارتوی به‌دست آمده توسط هر روش می‌پردازد. در واقع همه جواب‌های پارتوی به‌دست آمده توسط هر دو روش را با هم سطح‌بندی کرده و مشخص می‌کند که چند درصد جواب‌های سطح یک متعلق به هر روش است. هر چه این درصد بالاتر باشد، الگوریتم، از کیفیت بالاتری برخوردار است.

(۲) **شاخص پراکندگی:** این شاخص برای تعیین میزان جواب‌های غیرمغلوب یافت شده بر روی مرز

<sup>1</sup> Comparison Metrics

در انتها، عملکرد روش الگوریتم بهینه‌سازی ذرات انبوه - الگوریتم ژنتیک (GA-PSO) چند هدفه و الگوریتم ژنتیک مرتب شده غیر مغلوب (NSGA-II) برای حل مسایل طراحی شده، تجزیه و تحلیل می‌گردد. هر کدام از مسایل کوچک و بزرگ برای هر ترکیب  $r, t$ ، پنج بار اجرا گردیده‌اند. جداول ۴ و ۵ نتایج حاصل از اجرای دو الگوریتم را نشان می‌دهد.

جدول ۴. مقایسه نتایج به دست آمده برای مسایل کوچک

شاخص پراکندگی		شاخص کیفیت		تعداد قطعه	تعداد ماشین	تعداد سلول	شماره مساله
NSGA-II	GA-PSO	NSGA-II	GA-PSO				
282.8	587.8	12.54	87.46	7	10	2	1
314.2	579.2	22.16	77.84	7	10	3	2
503.9	640.1	0	100	7	10	4	3
169.3	670.1	3.26	96.74	8	10	3	4
204.1	711.7	3.39	96.61	8	10	4	5
609.2	640.7	0	100	10	10	2	6
617.2	726.9	0	100	10	10	3	7
603.4	716.6	0	100	10	10	4	8

جدول ۵. مقایسه نتایج به دست آمده برای مسایل بزرگ

شاخص پراکندگی		شاخص کیفیت		تعداد قطعه	تعداد ماشین	تعداد سلول	شماره مساله
NSGA-II	GA-PSO	NSGA-II	GA-PSO				
1556.1	4793.1	4.84	95.16	20	20	4	1
7494.5	4744.2	0	100	20	20	8	2
1172.4	4819.9	0	100	30	10	4	3
4457.6	7699.6	0	100	40	20	8	4
3321.4	8848.8	2.14	97.86	40	30	8	5
5892.3	10758	0	100	50	20	8	6
7211.3	12639	0	100	50	30	8	7
9145.9	12947	0	100	50	40	4	8
11738	13486	0	100	50	40	8	9



## ۶. نتیجه گیری

در این مقاله، مدل ریاضی جدید چند هدفه‌ای برای مسأله زمانبندی سیستم تولید سلولی ارائه شده است. اهداف مسأله مورد نظر، شامل: حداقل کردن حداکثر زمان تکمیل کارها، حداقل کردن هزینه دیرکرد و حداقل کردن هزینه زودکرد است که تاکنون به آن پرداخته نشده است. این مسأله با توجه به ماهیت پیچیده آن در زمره مسایل چند جمله‌ای نامعین سخت (NP-hard) قرار می‌گیرد. بنابراین، الگوریتم تلفیقی بهینه‌سازی ذرات انبوه - ژنتیک چندهدفه‌ای برای حل مسأله مورد نظر ارائه گردیده است. ابتدا چندین مسأله در ابعاد کوچک و بزرگ طراحی و سپس نتایج حاصل از اجرای روش پیشنهادی بر روی این مسایل با جواب‌های تولید شده توسط روش الگوریتم ژنتیک مرتب شده غیرمغلوب (NSGA-II) از لحاظ دو شاخص کیفیت و پراکندگی مقایسه شده است. نتایج نشان می‌دهد که روش پیشنهادی تلفیقی بهینه‌سازی ذرات انبوه - ژنتیک چند هدفه، کارایی بسیار قابل قبولی برای تولید جواب‌های پارتوی با کیفیت، متنوع و دارای پراکندگی بالاتری نسبت به روش الگوریتم ژنتیک مرتب شده غیرمغلوب داشته است. به عبارت دیگر، روش پیشنهادی، توالی‌های بهینه بیشتری را در اختیار تصمیم‌گیرندگان نهایی جهت انتخاب قرار می‌دهد. در نظر گرفتن توابع هدف دیگر، در نظر گرفتن زمان‌های آماده‌سازی، زمان‌های در دسترس برای ماشین‌ها، بررسی مسأله در حالت پویا، در نظر گرفتن بریدگی کارها و یا استفاده از سایر روش‌های فراابتکاری برای حل مسأله مورد نظر برای تحقیقات آتی توصیه می‌شود.

## منابع

- Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job-shop scheduling. *Management Science*, 34(3), 391-401.
- Deb, K. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transaction on Evolutionary Computation*, 6(2), 182-197.
- Hendizadeh, S.H., Faramarzi, H., & Mansouri, S.A. (2008). Meta-heuristic for scheduling a flowline manufacturing cell with sequence dependent family setup times. *International Journal of Production Economics*, 111, 593-605.
- Jerald, J., Asokan, P., Prabaharam, G., & Saravanan, R. (2005). Scheduling optimization of flexible manufacturing systems using particle swarm optimization algorithm. *International Journal of Advanced Manufacturing Technology*, 25, 964-971.
- Lin, S.-W., Ying, K.-C., & Lee, Z.-J. (2009). Metaheuristics for scheduling a non-permutation flow line manufacturing cell with sequence dependent family setup times. *Computers & Operations Research*, 36, 1110-1121.
- Logendran, R., & Sriskandarajah, C. (1993). Two-machine group scheduling problem with blocking and anticipatory setups. *European Journal of Operational Research*, 69, 467-481.
- Lotfi, K.G., Sherif, A.M., & Ashraf, O.N. (2008). A particle swarm-based genetic algorithm for scheduling in an agile environment. *Computers and Industrial Engineering*, 55, 707-720.
- Loukil, T., Teghem, J., & Tuytens, D. (2005). Solving multi-objective production scheduling problems using metaheuristics. *European Journal of Operational Research*, 161(1), 42-61.
- Lu, L.F., & Yuan, J.J. (2007). The single machine batching problem with identical family setup times to minimize maximum lateness is strongly NP-Hard. *European Journal of Operational Research*, 177, 1302-1309.
- Monkman, S.K., Morrice, D.J., & Bard, J.F. (2008). A production scheduling heuristic

- for an electronics manufacturer with sequence-dependent setup costs. *European Journal of Operational Research*, 187, 1100-1114.
- Rahimi-vahed, A.R., Javadi, B., Rabbani, M., & Tavakoli-Moghadam, R. (2008). A Multi-objective scatter search for bi-criteria no-wait flow shop scheduling problem. *Engineering Optimization*, 40(4), 331-346.
- Solimanpur, M., Vart, P., & Shankar, R. (2004). A heuristic to minimize makespan of cell scheduling problem. *International Journal of Production Economics*, 88, 231-241.
- Tasgetiren, M.F., Yun-Chia L., Mehmet S., & Gunes G. (2007). A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. *European Journal of Operational Research*, 177, 1930-1947.
- Tavakkoli-Moghaddam, R., Javadian, N., Khorrami, A., & Gholipour-Kanani, Y. (2010). Design of a scatter search method for a novel multi-criteria group scheduling problem in a cellular manufacturing system. *Expert Systems with Applications*, 37(3), 2661-2669.
- Wang, D., & Liu, L. (2008). Hybrid particle swarm optimization for solving resource-constrained FMS. *Progress in Natural Science*, 18, 1179-1183.
- Yang, W.H., & Liao, C.-J. (1996). Group scheduling on two cells with inter-cell movement. *Computers & Operations Research*, 23, 997-1006.
- Yoshida, T., & Hitomi, K. (1979). Optimal two-stage production scheduling with setup times separated. *AIIE Transactions*, 11, 261-263.
- Zitzler, E., Laumanns, M., & Thiele, L. (2001). *Improving the strength Pareto evolutionary algorithm*. Technical Report TIK-Report 103, Swiss Federal Institute of Technology Zurich (ETH).