

## حل مسئله سیستم تولید سلولی پویای چندهدفه فازی با استفاده از یک الگوریتم ترکیبی از NSGA-II و شبیه‌سازی تبرید تدریجی

علیرضا علی‌نژاد\*، سجاد ثابت\*\*، مصطفی اختیاری\*\*\*

### چکیده

سیستم‌های تولید سلولی یکی از مهم‌ترین روش‌های تولید به‌شمار می‌روند. در این مقاله، با توجه به انعطاف‌پذیری در پردازش عملیات، تخصیص ماشین به هر عملیات و تعریف فازی پارامترهای هزینه، یک مدل چندهدفه برای تشکیل یک سیستم تولید سلولی در شرایط پویا ارائه می‌شود. در مدل پیشنهادی، اهداف حداقل‌سازی هزینه‌های سیستم تولید و حداقل‌سازی واریانس هزینه‌های فازی در نظر گرفته می‌شوند. همچنین، به‌منظور دستیابی به جواب‌های مؤثر، یک الگوریتم ترکیبی از NSGA-II و شبیه‌سازی تبرید تدریجی پیشنهاد می‌شود و طراحی آزمایشات تاگوچی برای تنظیم پارامترهای این الگوریتم به‌کار می‌رود. برای بررسی کارایی الگوریتم پیشنهادی، نتایج به‌دست‌آمده از حل چندین مسئله نمونه مختلف، براساس شاخص‌های زمان CPU، فاصله نسل‌ها، فاصله‌بندی و کیفیت جواب‌ها با نتایج الگوریتم اولیه NSGA-II مقایسه و تحلیل می‌شود.

**کلیدواژه‌ها:** زمان‌بندی سیستم تولید سلولی؛ تئوری فازی؛ الگوریتم NSGA-II؛  
شبیه‌سازی تبرید تدریجی.

---

تاریخ دریافت مقاله: ۹۳/۷/۲۵، تاریخ پذیرش مقاله: ۹۳/۹/۱۶.

استادیار، دانشگاه آزاد اسلامی، قزوین.

دانشجوی کارشناسی ارشد، دانشگاه آزاد اسلامی، قزوین.

دانشجوی دکترای، دانشگاه شهید بهشتی (نویسنده مسئول).

## ۱. مقدمه

تولیدکنندگان همواره به دنبال کاهش هزینه‌های تولید هستند. یکی از روش‌های کاهش هزینه‌ها، استفاده از سیستم‌های تولید سلولی است. به منظور تشکیل این قبیل سیستم‌ها، اولین قدم طراحی سلول تولید است. تولید سلولی یک استراتژی تولیدی است که از فناوری گروهی<sup>۱</sup> گرفته شده است و ابزاری کارآمد برای افزایش بهره‌وری به شمار می‌رود. تولید سلولی انعطاف‌پذیری را به طرز چشمگیری در مراکز تولیدی افزایش می‌دهد. روش کار در این نوع سیستم به این صورت است که ماشین‌آلات براساس شباهت در عملیات یا اندازه و یا نوع محصول، در حداقل دو گروه تقسیم‌بندی و در سلول‌ها جایابی می‌شوند. برخی از مزایای سیستم‌های تولید سلولی عبارت‌اند از: ۱. تسهیل در برنامه‌ریزی؛ ۲. کاهش جابه‌جایی مواد؛ ۳. مدیریت بهتر و کنترل راحت‌تر سیستم؛ ۴. کاهش تعداد نیروی انسانی؛ ۵. کنترل موجودی ساده‌تر [۲۷]؛ ۶. کاهش زمان راه‌اندازی [۳]؛ ۷. کاهش ابزارآلات مورد نیاز [۲۷] و ۸. کاهش فضای مورد نیاز [۲۸].

وجود تقاضای متفاوت از دوره‌ای به دوره دیگر موجب شده است که تولیدکنندگان به جای استفاده از استراتژی ساخت برای انبار<sup>۲</sup>، به مهندسی در سفارش<sup>۳</sup> روی آورند. با توجه به تغییر در چرخه عمر محصولات و یا تقاضای آن‌ها، می‌توان این موضوع را در سیستم‌های تولید سلولی نیز در نظر گرفت. در شرایط پویا، تقاضا می‌تواند در هر دوره متفاوت باشد؛ بنابراین؛ دوره برنامه‌ریزی نیز کوتاه‌تر خواهد بود. همچنین، در شرایط پویا، مواد در جریان ساخت و موجودی نیمه‌ساخته در هر دوره متفاوت است و اگر چیدمان مناسبی وجود نداشته باشد، هزینه جابه‌جایی افزایش خواهد یافت. برای کاهش جریان جابه‌جایی مواد، تسهیلات موجود براساس حجم تقاضا در هر دوره مجدداً پیکربندی<sup>۴</sup> می‌شوند. مدل‌هایی که تقاضای متفاوت و یا تنوع را در نظر می‌گیرند و در آن‌ها، برنامه‌ریزی در یک افق برنامه‌ریزی چنددوره‌ای صورت می‌گیرد، سیستم‌های تولید سلولی پویا<sup>۵</sup> نامیده می‌شوند [۴].

## ۲. مبانی نظری و پیشینه تحقیق

نخستین بار، کاسیاک (۱۹۸۷) یک مسئله تولید سلولی را در قالب یک مدل کلی معرفی کرد؛ به طوری که پردازش یک عملیات با بیش از یک ماشین نیز قابل انجام بود [۲۱]. سلیمان‌پور و همکاران (۲۰۰۴) یک مدل برنامه‌ریزی عدد صحیح چندهدفه را برای طراحی سیستم تولید

---

1. Group Technology  
2. Make to Stock  
3. Engineering to Order  
4. Configuration  
5. Dynamic Cellular Manufacturing System

سلولی ارائه کردند [۲۹]. مدل ریاضی پیشنهادی آنان دارای چندین ویژگی مهم بود؛ از جمله تقاضای ماشین‌آلات، هزینه، زمان پردازش، توالی عملیات، سرمایه‌گذاری در خرید ماشین‌آلات، حجم تولید و ظرفیت ماشین‌آلات. یاسودا و همکاران (۲۰۰۵) الگوریتم ژنتیک گروه‌بندی را برای مسائل چندهدفه فرم‌دهی سلولی پیشنهاد دادند [۳۵]. در مدل پیشنهادی آنان، به زمان پردازش، نیاز تولید و زمان در دسترس هر ماشین به منظور حداقل‌سازی اهداف جریان‌های داخل سلولی و واریانس بارگذاری سلول‌ها در دوره‌های مفروض توجه شده است. داس و همکاران (۲۰۰۷) مدلی دوهدفه را برای طراحی و تحلیل سیستم تولید سلولی با در نظر گرفتن قابلیت اطمینان ماشین‌آلات ارائه کردند [۱۱]. آرامون بجستانی و همکاران (۲۰۰۹) نیز از جست‌وجوی پراکنده چندهدفه برای مسئله فرم‌دهی سلولی در شرایط پویا استفاده کردند و در نهایت، نتایج آن را با الگوریتم چندهدفه ژنتیک مقایسه نمودند [۱]. رنجبر بورانی و همکاران (۲۰۱۰) از الگوریتم جست‌وجوی پراکنده مبتنی بر روش تاپسیس، در طراحی سیستم تولید سلولی چندهدفه استفاده کردند [۲۴]. آنان در این پژوهش، مدلی با اهداف حداقل‌سازی هزینه‌ها، حداکثرسازی سطح به‌کارگیری ماشین‌آلات و حداقل‌سازی میزان انحراف در به‌کارگیری سلول‌ها را ارائه کردند. قطب‌الدینی و همکاران (۲۰۱۱) نیز یک مدل عدد صحیح مختلط چندهدفه را برای مسئله طراحی فرم‌دهی پویا، با توجه به شرایط بحرانی دنیای واقعی در تولید ناب پیشنهاد دادند [۱۴]. مدل پیشنهادی آنان مجموع هزینه‌های متغیر و ثابت ماشین‌آلات و همچنین هزینه‌های جابه‌جایی داخلی و خارجی سلول را حداقل می‌کند. برای این منظور، مدل آنان هزینه نیروی کار و خرید ماشین‌آلات را حداقل می‌کند و نرخ به‌کارگیری منابع انسانی را افزایش می‌دهد. چاتوپادای و همکاران (۲۰۱۳) به‌طور کمی-تحلیلی میزان تأثیر الگوریتم ژنتیک و شبکه‌های عصبی بر روش‌های فرم‌دهی سلولی در سیستم‌های تولید سلولی را مرور کردند [۸]. کیا و همکاران (۲۰۱۴) نیز یک مدل برنامه‌ریزی عدد صحیح مختلط را برای طرح جانمایی چندطبقه‌ای سیستم‌های تولید سلولی در محیط پویا ارائه کردند که در نهایت با استفاده از الگوریتم ژنتیک آن را حل نمودند [۱۹].

با مطرح کردن تئوری مجموعه‌های فازی توسط زاده (۱۹۶۵)، تاکنون از این تئوری در مسائل سیستم‌های تولید سلولی نیز استفاده شده است. ژو و وانگ (۱۹۸۹) از ریاضیات فازی در تشکیل خانواده‌های قطعات در تکنولوژی گروهی استفاده کردند [۳۴]. توکلی مقدم و آریانژاد (۲۰۰۷) یک مدل برنامه‌ریزی مختلط عدد صحیح خطی فازی، با تقاضای فازی و تغییر در تقاضا را ارائه و آن را با الگوریتم ژنتیک حل کردند [۳۱]. بولیف و آتیف (۲۰۰۸) نیز یک الگوریتم ژنتیک فازی را برای مسئله دوهدفه فرم‌دهی سلولی در شرایط پویا با در نظر گرفتن دو استراتژی

فعال و غیرفعال ارائه کردند [۶]. در استراتژی غیرفعال و در افق برنامه‌ریزی که به صورت چنددوره‌ای است، شکل چیدمان ماشین‌آلات حفظ می‌شود؛ اما در استراتژی فعال، ترکیب ماشین‌آلات می‌تواند متفاوت باشد. آریکان و گانگور (۲۰۰۹) مدلی را برای طراحی چیدمان تولید سلولی با استفاده از برنامه‌ریزی پارامتریک چندهدفه فازی ارائه کردند [۲]. کیا و همکاران (۲۰۱۱) مدلی را به منظور طراحی چیدمان سیستم تولید سلولی پویا، با انتخاب مسیر و پیکربندی مجدد و با رویکردی جدید از برنامه‌ریزی خطی فازی ارائه کردند [۱۸].

فرض بیشتر مدل‌هایی که در زمینه سیستم‌های تولید سلولی ارائه شده‌اند، این بوده است که پردازش هر عملیات فقط به یک شکل و با یک ماشین قابلیت برنامه‌ریزی دارد [۱۸]. در این مقاله، با توسعه مدل طراحی چیدمان سیستم تولید سلولی در شرایط پویا [۱۷]، یک مدل ریاضی غیرخطی ارائه می‌شود که علاوه بر تعیین طراحی چیدمان در هر دوره و با توجه به امکان وجود چند گزینه برای تولید، تخصیص مناسب برای پردازش هر عملیات تعیین می‌شود. با توجه به غیرقطعی بودن زمان پردازش هر عملیات، هزینه انجام آن‌ها را نمی‌توان به صورت قطعی تعیین کرد؛ در نتیجه، برای اینکه مدل مورد نظر به واقعیت نزدیک‌تر شود، هزینه‌های انجام عملیات به صورت پارامترهای فازی در نظر گرفته شده است. از آنجا که برای حل مدل ریاضی، هزینه‌های فازی را فازی‌زدایی می‌کنیم، مجموع واریانس هزینه‌های فازی تخصیص داده شده در انجام عملیات توسط تابع هدف دیگری مدل‌سازی می‌شود؛ بنابراین، در این مقاله، یک مدل ریاضی با اهداف حداقل‌سازی هزینه‌ها و حداقل‌سازی واریانس هزینه‌های فازی پیشنهاد می‌شود. برای حل این مسئله، الگوریتمی ترکیبی از الگوریتم‌های NSGA-II و شبیه‌سازی تبرید تدریجی ارائه و روش حل پیشنهادی با الگوریتم اولیه NSGA-II، براساس شاخص‌های مطرح در الگوریتم‌های فراابتکاری چندهدفه، مقایسه و تحلیل می‌شود.

### ۳. روش شناسی پژوهش

روش شناسی این پژوهش، در ابتدا ارائه یک مدل ریاضی چند هدفه فازی در خصوص سیستم‌های تولید سلولی پویا و سپس پیشنهاد یک الگوریتم فراابتکاری ترکیبی به منظور حل آن است.

**مدل پیشنهادی.** در این بخش، پس از مروری بر مفاهیم بهینه‌سازی چندهدفه و تئوری اعداد فازی، مدل پیشنهادی این مقاله ارائه می‌شود.

**بهینه‌سازی چندهدفه.** مسئله بهینه‌سازی چندهدفه با  $T$  تابع هدف را می‌توان به صورت رابطه

۱ نوشت:

$$\min \mathbf{y} = f(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_T(\mathbf{x})]$$

s.t.

$$g_i(\mathbf{x}) \geq 0, \quad i = 1, \dots, D.$$

رابطه (۱)

به طوری که  $\mathbf{x} = (x_1, \dots, x_n)^T$  بردار متغیرهای تصمیم،  $\Theta$  فضای جواب مسئله و  $\mathbf{x} \in \Theta \subset \mathbb{R}^n$  است. همچنین،  $\mathbf{y} \in Y$  بردار هدف و  $Y$  فضای هدف است.  $g_i$  برای  $i = 1, \dots, D$  نیز محدودیت‌های مسئله به‌شمار می‌رود.

غالباً مفاهیم اساسی زیر در مورد مدل ۱ کاربرد بسیار دارند [۱۶]:

بردارهای تصمیم  $x_1, x_2 \in \Theta$  را در نظر بگیرید:

- تعریف ۱:  $x_1$  بر بردار تصمیم  $x_2$  غالب است ( $x_2 < x_1$ )؛ اگر و تنها اگر  $f_t(x_1) \leq f_t(x_2)$  (برای  $t = 1, \dots, T$ ) و حداقل یک  $t \in \{1, \dots, T\}$  وجود داشته باشد که رابطه  $f_t(x_1) < f_t(x_2)$  برقرار باشد. اندیس  $t$  شماره تابع هدف را برای اهداف اول تا  $T$ ام در مدل ۱ نشان می‌دهد.

- تعریف ۲:  $x_2$  بهینه پارتو (غیرمسلط شده) نامیده می‌شود؛ اگر و تنها اگر هیچ جوابی وجود نداشته باشد که بر آن غالب باشد.

- تعریف ۳:  $P_S = \{x_1 \in \Theta \mid \exists x_2 \in \Theta \text{ s.t. } x_2 < x_1\}$  مجموعه بهینه پارتو همه بردارهای تصمیم بهینه پارتو است.

- تعریف ۴:  $P_F = \{f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_T(\mathbf{x})) \mid \mathbf{x} \in P_S\}$  مرز بهینه پارتو تمامی مقادیر توابع هدف متناظر با بردارهای تصمیم در مجموعه  $P_S$  را ارائه می‌کند. زمانی که یک بردار تصمیم روی کل فضای جواب غیر غالب است، آنگاه این بردار، بهینه پارتو خواهد بود.

**تئوری فازی.** در یک فرآیند تصمیم‌گیری، اغلب تصمیم‌گیران با تردیدها و عدم قطعیت‌هایی روبه‌رو می‌شوند. به منظور رفع ابهام و عدم قطعیت‌های موجود، تئوری مجموعه‌های فازی توسط زاده (۱۹۶۵) معرفی شد. پیش از ارائه مدل پیشنهادی، برخی از مهم‌ترین عملیات‌های استاندارد در ادبیات تئوری فازی مرور می‌شوند.

**عملیات استاندارد ریاضی در مجموعه‌های فازی.** روابط زیر عملیات استاندارد ریاضی برای اعداد فازی مثلثی  $\tilde{A} = (a_1, a_2, a_3)$  و  $\tilde{B} = (b_1, b_2, b_3)$  را در مجموعه اعداد حقیقی نشان می‌دهند [۹]:

$$\tilde{A} + \tilde{B} = (a_1 + b_1, a_2 + b_2, a_3 + b_3) \quad \text{رابطه (۲)}$$

$$\tilde{A} - \tilde{B} = (a_1 - b_3, a_2 - b_2, a_3 - b_1) \quad \text{رابطه (۳)}$$

$$\tilde{A} \times \tilde{B} \equiv (a_1 \times b_1, a_2 \times b_2, a_3 \times b_3) \quad \text{رابطه (۴)}$$

**فازی‌زدایی.** در ادبیات، روش‌های مختلفی برای فازی‌زدایی وجود دارد. روشی که در این مقاله برای فازی‌زدایی استفاده می‌شود، روش میانگین اعداد فازی است. با توجه به تابع عضویت عدد فازی مثلثی  $\tilde{A} = (a_1, a_2, a_3)$ ، روش میانگین برای فازی‌زدایی به صورت رابطه ۵ قابل تعریف است [۱۳]:

$$A_D = \frac{\int_{-s}^{+s} u \mu_{\tilde{A}}(u) du}{\int_{-s}^{+s} \mu_{\tilde{A}}(u) du} \quad \text{رابطه (۵)}$$

به طوری که  $\mu_{\tilde{A}}(u)$  تابع عضویت عدد فازی  $\tilde{A}$  و  $[-s, +s]$  فاصله پایه‌ای عدد فازی است؛ بنابراین، میانگین عدد فازی مثلثی  $\tilde{A}$  برای فازی‌زدایی به صورت رابطه ۶ محاسبه می‌شود:

$$A_D = \frac{2a_2^3 - 3a_1a_2^2 + a_1^3}{6(a_2 - a_1)} + \frac{a_3^3 + 2a_2^3 - 3a_3a_2^2}{6(a_3 - a_2)} \quad \text{رابطه (۶)}$$

**واریانس اعداد فازی.** کارلسون و فولر (۲۰۱۱) روشی را برای محاسبه واریانس اعداد فازی ارائه کردند که در این مقاله برای محاسبه واریانس اعداد فازی استفاده می‌شود. در این روش، واریانس عدد فازی مثلثی  $\tilde{A} = (a_1, a_2, a_3)$  به صورت رابطه ۷ تعریف می‌شود:

$$\text{Var}(\tilde{A}) = \frac{1}{2} \int_0^1 \gamma (A_U(\gamma) - A_L(\gamma))^2 d\gamma \quad \text{رابطه (۷)}$$

به طوری که در رابطه ۷،  $[\tilde{A}]^\gamma = [A_L(\gamma), A_U(\gamma)]$ ، سطح  $\gamma \in [0, 1]$  مجموعه عدد فازی مثلثی  $\tilde{A}$  نام دارد و به صورت رابطه ۸ نمایش داده می‌شود:

$$[\tilde{A}]^\gamma = [a_2 - a_1(1-\gamma), a_2 + a_3(1-\gamma)] \quad \text{رابطه (۸)}$$

بنابراین، واریانس اعداد فازی مثلثی به صورت رابطه ۹ محاسبه می‌شود:

$$\text{Var}(\tilde{A}) = \frac{1}{2} \int_0^1 \gamma (a_2 + a_3(1-\gamma) - (a_2 - a_1(1-\gamma)))^2 d\gamma = \frac{(a_1 + a_3)^2}{24} \quad \text{رابطه (۹)}$$

**مدل برنامه‌ریزی فازی چندهدفه پیشنهادی.** در این بخش، یک مدل چندهدفه غیرخطی عدد صحیح برای حل مسئله سیستم‌های تولید سلولی پیشنهاد می‌شود. در مدل پیشنهادی این مقاله، از دو تابع هدف حداقل‌سازی هزینه و حداقل‌سازی واریانس هزینه استفاده می‌شود. در یک سیستم تولید سلولی، تعدادی سلول و در هر سلول، تعدادی جایگاه وجود دارد که در هر جایگاه حداکثر یک ماشین قرار می‌گیرد. طی چند دوره مختلف، چند نوع محصول خواهیم داشت که هر محصول دارای تعدادی مراحل تولید است. تقاضای هر محصول نیز در دسترس است. محصولات به‌صورت دسته‌ای جابه‌جا می‌شوند و برخی از عملیات‌ها با بیش از یک ماشین قابل پردازش هستند.

هدف از ارائه مدل پیشنهادی، انتخاب برنامه تولیدی با کمترین مجموع هزینه‌ها است. از آنجا که هزینه‌های سیستم به‌صورت اعداد فازی مثلثی تعریف می‌شوند، برای حل مدل باید عملیات فازی‌زدایی صورت گیرد که در این مقاله، برای انجام فازی‌زدایی از روش میانگین (رابطه ۵) استفاده می‌شود. با توجه به عدم قطعیت در هزینه‌ها، تابع هدف حداقل‌سازی واریانس هزینه سعی دارد تخصیص‌هایی را انتخاب کند که کمترین میزان پراکندگی را داشته باشند.

**نتایج مورد انتظار.** طی فرآیند طراحی سیستم تولید سلولی، به‌دنبال دستیابی به نتایج زیر هستیم:

- تخصیص بهترین روش تولید برای پردازش مراحل در هر دوره؛
- طراحی چیدمان مناسب در هر دوره.

**پارامترهای ورودی.** پارامترهای ورودی مدل پیشنهادی که در آغاز دوره برنامه‌ریزی باید مشخص شوند، عبارت‌اند از:

- تعداد دوره برنامه‌ریزی؛
- تعداد سلول و تعداد جایگاه در هر سلول؛
- تعداد محصولات و تقاضا برای هر نوع محصول در هر دوره؛
- تعداد مراحل تولید هر محصول؛
- قابلیت ماشین‌ها برای انجام عملیات‌ها؛
- هزینه‌های فازی.

**مفروضات.** مفروضات مدل پیشنهادی به شرح زیر هستند:

- تعداد سلول و تعداد جایگاه در سلول‌ها مشخص و برابر است؛
- هزینه نصب و راه‌اندازی ناچیز است؛

- زمان در دسترس ماشین نامحدود است؛
- راندمان ماشین ۱ و ضایعات صفر است؛
- هزینه‌های انبارداری ناچیز است؛

**محدودیت‌ها.** محدودیت‌های مدل پیشنهادی به شرح زیر هستند:

- در هر دوره و در هر جایگاه، حداکثر یک ماشین جای می‌گیرد؛
- در هر دوره، هر ماشین حتماً در یک جایگاه قرار می‌گیرد؛
- پردازش هر مرحله از یک محصول فقط روی یک ماشین انجام می‌شود؛
- تمام مراحل پردازش یک محصول در هر دوره انجام می‌شود؛
- تعداد ماشین‌ها کوچک‌تر یا مساوی تعداد کل جایگاه‌ها است.

### نمادها

**اندیس‌ها.** اندیس‌های مورد استفاده در مدل‌سازی مدل پیشنهادی این مقاله عبارت‌اند از:

- $P$ : تعداد دوره‌ها
- $K$ : تعداد محصولات
- $C$ : تعداد سلول‌های موجود
- $M$ : تعداد جایگاه‌ها در هر سلول
- $N$ : تعداد ماشین‌های موجود
- $L_k$ : تعداد مراحل تولید محصول  $k$ ام
- $p$ : شمارنده هر دوره  $p = 1, \dots, P$
- $k$ : شمارنده هر محصول  $k = 1, \dots, K$
- $l$ : شمارنده مراحل تولید هر محصول  $l = 1, \dots, L_k$
- $u, w$ : شمارنده‌های سلول  $u, w = 1, \dots, C$
- $x, y$ : شمارنده‌های جایگاه  $x, y = 1, \dots, M$
- $i, j$ : شمارنده‌های ماشین  $i, j = 1, \dots, N$

**پارامترها.** پارامترهای ورودی مدل پیشنهادی که در آغاز افق برنامه‌ریزی باید مشخص شوند، عبارت‌اند از:

$B_k$ : اندازه دسته محصول  $k$ ام



- $d_{xy}$  فاصله جایگاه  $x$  از جایگاه  $y$  در داخل یک سلول واحد
- $d_{uw}$  فاصله سلول  $u$  از سلول  $w$
- $D_{pk}$  تقاضای محصول  $k$ ام در دوره  $p$ ام
- $S_{kl}$  مجموعه ماشین‌هایی که می‌توانند در مرحله  $l$ ام، محصول  $k$ ام را پردازش کنند
- $\tilde{c}_{kli}$  هزینه فازی پردازش مرحله  $l$ ام از محصول  $k$ ام روی ماشین  $i$ ام
- $\tilde{A}_i$  هزینه فازی جابه‌جایی ماشین  $i$ ام
- $\tilde{C}^{in}$  هزینه فازی جابه‌جایی دسته‌های محصول داخل سلول
- $\tilde{C}^{out}$  هزینه فازی جابه‌جایی دسته‌های محصول خارج سلول
- $Var(\tilde{c}_{kli})$  واریانس هزینه فازی پردازش عملیات مرحله  $l$ ام محصول  $k$ ام روی ماشین  $i$ ام که با توجه به تابع عضویت تعریف‌شده در رابطه ۱۰ به صورت رابطه ۱۱ قابل محاسبه است [۷]:

$$\mu_{\tilde{c}_{kli}}(c) = \begin{cases} 0 & \text{for } c < c_{kli}^1 \\ \frac{c - c_{kli}^1}{c_{kli}^2 - c_{kli}^1} & \text{for } c_{kli}^1 \leq c < c_{kli}^2 \\ \frac{c_{kli}^3 - c}{c_{kli}^3 - c_{kli}^2} & \text{for } c_{kli}^2 \leq c \leq c_{kli}^3 \\ 0 & \text{for } c > c_{kli}^3 \end{cases} \quad \text{رابطه (۱۰)}$$

$$Var(\tilde{c}_{kli}) = \frac{(c_{kli}^1 + c_{kli}^3)^2}{24} \quad \text{رابطه (۱۱)}$$

**متغیرهای تصمیم.** متغیرهای تصمیم به کاررفته در مدل پیشنهادی عبارت‌اند از:

- $e_{puxi}$ : اگر در دوره  $p$ ام، ماشین  $i$ ام در جایگاه  $x$  سلول  $u$  قرار گیرد، آنگاه این متغیر برابر ۱ بوده و در غیر این صورت برابر صفر است.
- $o_{pkli}$ : اگر در دوره  $p$ ام، عملیات مرحله  $l$ ام کالای  $k$ ام به وسیله ماشین  $i$  پردازش شود، آنگاه این متغیر برابر ۱ بوده و در غیر این صورت برابر صفر است.

روابط ۱۸-۱۲، مدل برنامه‌ریزی فازی چندهدفه پیشنهادی این مقاله را در خصوص مسئله سیستم‌های تولید سلولی ارائه می‌کند:

$$\text{Min } \tilde{Z}_1 = \quad \text{رابطه (۱۲)}$$

$$\sum_{p=1}^P \sum_{u=1}^C \sum_{w=1}^C \sum_{x=1}^M \sum_{y=1}^M \sum_{k=1}^K \sum_{l=2}^{L_k} \sum_{i \in S_{k(l)}} \sum_{j \in S_{kl}} e_{puxi} e_{pwyj} o_{pk(l-1)i} o_{pklj} \left[ \frac{D_{pk}}{B_k} \right] (d_{xy} \tilde{C}^{in} + d_{uw} \tilde{C}^{out}) \quad ۱۲-۱$$

$$+ \sum_{p=1}^P \sum_{k=1}^K \sum_{l=1}^{L_k} \sum_{i \in S_{kl}} o_{pkli} D_{pk} \tilde{c}_{kli} \quad ۱۲-۲$$

$$+ \sum_{p=1}^{P-1} \sum_{u=1}^C \sum_{w=1}^C \sum_{x=1}^M \sum_{y=1}^M \sum_{i=1}^N e_{puxi} e_{(p+1)wyi} (d_{xy} + d_{uw}) \tilde{A}_i \quad ۱۲-۳$$

$$\text{Min } Z_2 = \sum_{p=1}^P \sum_{k=1}^K \sum_{l=1}^{L_k} \sum_{i \in S_{kl}} o_{pkli} \text{Var}(\tilde{c}_{kli}) \quad \text{رابطه (۱۳)}$$

به طوری که:

$$\sum_{i=1}^N e_{puxi} \leq 1, \forall 1 \leq u \leq C \quad 1 \leq x \leq M \quad 1 \leq p \leq P \quad \text{رابطه (۱۴)}$$

$$\sum_{i=1}^N \sum_{x=1}^M e_{puxi} = 1, \forall 1 \leq u \leq C \quad 1 \leq p \leq P \quad \text{رابطه (۱۵)}$$

$$\sum_{i \in S_{kl}} o_{pkli} = 1, \forall 1 \leq k \leq K \quad 1 \leq l \leq L_k \quad 1 \leq p \leq P \quad \text{رابطه (۱۶)}$$

$$e_{puxi} \in \{0,1\}, \forall 1 \leq p \leq P \quad 1 \leq u \leq C \quad 1 \leq x \leq M \quad 1 \leq i \leq N \quad \text{رابطه (۱۷)}$$

$$o_{pkli} \in \{0,1\}, \forall 1 \leq p \leq P \quad 1 \leq k \leq K \quad 1 \leq l \leq L_k \quad 1 \leq i \leq N. \quad \text{رابطه (۱۸)}$$

در مدل پیشنهادی بالا:

- رابطه ۱۲ نشان‌دهنده تابع هدف اول است که در پی حداقل‌سازی هزینه کل سیستم است که در رابطه ۱۲-۱ هزینه جابه‌جایی محصول برای پردازش مراحل در صورتی که عملیات مرحله  $(l-1)$  ام توسط ماشین  $\lambda$  ام که در جایگاه  $x$  سلول  $u$  قرار گرفته و عملیات مرحله  $\lambda$  ام توسط ماشین  $\lambda$  ام که در جایگاه  $y$  سلول  $w$  قرار گرفته است، انجام شود. در رابطه ۱۲-۲، هزینه پردازش مراحل مختلف و در رابطه ۱۲-۳، هزینه جابه‌جایی ماشین‌آلات طی دو دوره متوالی از جایگاه  $x$  سلول  $w$  به جایگاه  $y$  سلول  $u$  فرموله شده است.

- رابطه ۱۳ نشانگر تابع هدف دوم مدل است که با توجه به رابطه ۱۱ به دنبال حداقل‌سازی مجموع واریانس هزینه‌های فازی در پردازش محصولات طی یک افق برنامه‌ریزی است.

- محدودیت ذکر شده در رابطه ۱۴ تضمین می‌کند که در هر دوره و در هر جایگاه، حداکثر یک

ماشین جای می‌گیرد.

– محدودیت ذکر شده در رابطه ۱۵ تضمین می‌کند که در هر دوره، هر ماشین حتماً در یک جایگاه قرار می‌گیرد.

– محدودیت ذکر شده در رابطه ۱۶ تضمین می‌کند که در هر دوره، مرحله  $l$ ام محصول  $k$ ام روی یک ماشین انجام می‌شود و نهایتاً تمام مراحل انجام خواهند شد.

– روابط ۱۷ و ۱۸ متغیرهای تصمیم مدل بالا را در قالب متغیرهای صفر و یک مقید می‌سازند. در مدل بالا، با توجه به اینکه پارامترهای هزینه از نوع اعداد فازی مثلثی هستند، به منظور حل مدل پیشنهادی، اعداد فازی مثلثی با روش میانگین فازی زدایی شده‌اند؛ بنابراین، دلیل افزودن تابع هدف دوم به مدل، حداقل سازی مجموع واریانس‌های مربوط به هزینه فازی تخصیص‌ها است. در این صورت، با توجه به اینکه عملیات برخی مراحل می‌توانند توسط چندین ماشین مختلف انجام شوند، اگر هزینه فازی زدایی شده برای پردازش یک مرحله توسط چند ماشین یکسان باشد، تخصیصی صورت می‌پذیرد که دارای پراکندگی کمتری است. با توجه به پیچیدگی در حل این مسئله، آن را جزو مسائل NP-complete در نظر می‌گیرند [۱۷].

در بخش بعدی، مدل پیشنهادی توسط یک الگوریتم پیشنهادی که از ترکیب الگوریتم‌های NSGA-II و شبیه‌سازی تبرید تدریجی به دست آمده است، حل می‌شود و برای نشان دادن کارایی الگوریتم پیشنهادی، نتایج به دست آمده با نتایج الگوریتم اولیه NSGA-II براساس چندین شاخص مقایسه می‌شوند.

### الگوریتم پیشنهادی

**الگوریتم NSGA-II.** الگوریتم اولیه NSGA-II [۱۲]، یکی از کارآمدترین و مشهورترین الگوریتم‌های ابتکاری چندهدفه است. این الگوریتم از تکنیک مرتب‌سازی نامغلوب و یک فاصله ازدحامی برای دسته‌بندی و انتخاب گروه‌ها استفاده می‌کند. در این الگوریتم، ابتدا نسل جدید جواب‌ها تولید و سپس جواب‌های نسل جاری و نسل جدید در یک مجموعه با یکدیگر ترکیب می‌شوند [۱۲]. در پایان، بهترین پاسخ‌ها برحسب تنوع و نامغلوب بودن، به عنوان راه‌حل نهایی مؤثر انتخاب می‌شوند. شکل‌های ۱-۳ رویه اجرای الگوریتم اولیه NSGA-II را نشان می‌دهند.

$t = 0$  قرار دهید و جمعیت تصادفی  $P_0$  را به اندازه  $N$  به‌عنوان جمعیت فعلی تولید کنید. توابع هدف را ارزیابی کنید و آن‌ها را براساس ارتباط غالب و مغلوب توضیح داده‌شده در الگوریتم ارائه‌شده در شکل ۲ رتبه‌بندی نمایید.

$t = t + 1$  قرار دهید و براساس نرخ تقاطع، تعداد جواب مورد نیاز را به‌عنوان والدین برای تولید فرزند، با معیار مسابقات دوبه‌دو انتخاب و جمعیت جدید را ایجاد کنید.

براساس نرخ عملگر جهش، جواب‌های مورد نیاز را با استفاده از مسابقات دوبه‌دو، از میان کروموزوم‌ها انتخاب و جمعیت جدید را ایجاد کنید.

جمعیت جدید ایجادشده را براساس توابع هدف ارزیابی کنید.

جمعیت جدید و جمعیت فعلی را با هم ترکیب و پس از ارزیابی توابع هدف، آن‌ها را براساس ارتباط غالب و مغلوب رتبه‌بندی کنید.

معیار انتخاب جمعیت: ابتدا براساس ارتباط غالب و در صورتی که اجزاء دارای رتبه یکسانی هستند، براساس فاصله ازدحام که در الگوریتم ارائه‌شده در شکل ۳ توضیح داده شده است، تعداد  $N$  جواب منتخب را مشخص و جایگزین جمعیت فعلی کنید. در صورت ارضا شدن معیار توقف، اجرای الگوریتم را متوقف و در غیر این صورت، گام ۳ را اجرا کنید.

شکل ۱. گام‌های اجرای الگوریتم NSGA-II

۱. تعریف شمارنده  $i$  به‌عنوان تعداد گروه و دسته با مقدار اولیه  $i = 1$

۲. تعریف مجموعه دسته اول  $F_i = \emptyset$

۳. برای هر عضو دلخواه  $p$  از جمعیت، مراحل زیر را انجام دهید:

۱-۳. تعریف مجموعه اعضای از جمعیت که توسط  $p$  مغلوب می‌شوند  $S_p = \emptyset$

۲-۳. تعداد اعضای از جمعیت که بر عضو  $p$  غلبه می‌کنند  $n_p = 0$

۳-۳. برای هر عضو  $q$ ، از جمعیت غیر از  $p$ ، مراحل زیر را انجام دهید:

۱-۳-۳. اگر  $p$  بر  $q$  غلبه می‌کند، آنگاه  $\{q\}$   $S_p = S_p + \{q\}$

۲-۳-۳. اگر  $q$  بر  $p$  غلبه می‌کند، آنگاه  $n_p = n_p + 1$

۴-۳. اگر  $n_p = 0$  باشد، آنگاه:

۱-۳-۴. عضو  $p$  دارای رتبه  $i$  است:  $rank_p = i$

۲-۳-۴. عضو  $p$  از دسته  $i$  است:  $F_i = F_i + \{p\}$

۴. تعریف مجموعه موقتی  $Q = \emptyset$

۵. به‌ازای هر عضو دلخواه  $p$  از اعضای مجموعه  $F_i$  عضو دلخواه  $q$  از مجموعه  $S_p$  را انتخاب کنید:

۱-۵. کاهش یک‌واحدی تعداد اعضای غلبه‌شونده بر  $(n_q = n_q - 1)$

۵. اگر  $n_q = 0$  باشد (دیگر هیچ عضوی به‌جز  $p$  را مغلوب نمی‌کند) آنگاه:  $Q = Q \cup \{q\}$

۶. شماره دسته یا گروه را یک واحد افزایش دهید:  $i = i + 1$

مجموعه  $Q$  را به‌عنوان دسته و گروه بعدی در سیستم رتبه‌بندی لحاظ کنید:  $F_i = Q$

به گام ۴ بازگردید.

شکل ۲. گام‌های رتبه‌بندی نامغلوب

۱. مقدار  $C.D_1$  و  $C.D_n$  (به ترتیب مقدار فاصله ازدحامی پاسخ‌های اول و  $m$ ) را برابر بی‌نهایت قرار دهید.
۲. برای تمام بردارهای نامغلوب در یک مرز پارتو، مقدار توابع هدف را محاسبه و به صورت صعودی مرتب کنید.
۳. برای تمام عناصر مرتب‌شده از  $i=2$  تا  $i=n-1$  در یک مرز پارتو، مقدار  $d_{ij}$  را به ازای تمامی اهداف  $(j=1, \dots, K)$  با توجه به رابطه ۱۹ محاسبه کنید:

$$d_{ij} = \frac{|f_j^{i-1} - f_j^{i+1}|}{|f_j^1 - f_j^n|} \quad \text{رابطه ۱۹}$$

در رابطه بالا  $f_j^i$  مقدار تابع هدف  $f_j$  برای پاسخ  $i$ ام است.

۴. مقدار فاصله ازدحام را برای تمام عناصر با توجه به رابطه ۲۰ محاسبه کنید:

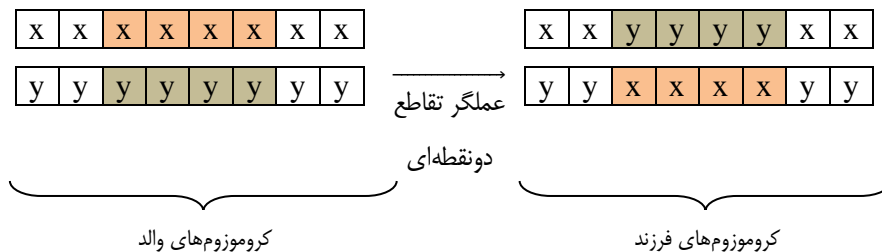
$$C.D_i = \sum_{j=1}^k d_{ij} \quad \text{رابطه ۲۰}$$

۵. برای انتخاب پاسخ‌های مناسب براساس معیار فاصله ازدحام در یک مرز پارتو، ابتدا  $C.D_i$ ها را به صورت نزولی مرتب و سپس تعداد پاسخ مورد نیاز را از ابتدا انتخاب کنید.

شکل ۳.۳. کام‌های محاسبه فاصله جمعیت

**نحوه نمایش جواب مسئله تولید سلولی در الگوریتم NSGA-II.** در این بخش، نحوه نمایش جواب مسئله تولید سلولی در الگوریتم NSGA-II که توسط کمالی دولت‌آبادی و همکاران (۲۰۰۹) ارائه شده است، تشریح می‌شود؛ اما قبل از ارائه نحوه نمایش جواب، عملگرهای تقاطع و جهش معرفی می‌شوند.

**عملگر تقاطع.** ابتدا کروموزوم‌های والد از طریق مسابقات دوبه‌دو انتخاب می‌شوند و از دو نقطه که به صورت تصادفی انتخاب می‌شوند، عملیات برش انجام می‌شود و فرزندهای جدید تولید می‌شوند. شکل ۴ نمایش کلی عملگر تقاطع مورد استفاده در این تحقیق را نشان می‌دهد.



شکل ۴. نمایش کروموزوم‌های تولیدشده براساس عملگر تقاطع دونقطه‌ای

**عملگر جهش.** برای تولید کروموزوم‌های جدید براساس عملگر جهش، ابتدا با استفاده از

مسابقات دوبه‌دو یک کروموزوم انتخاب می‌شود و سپس دو جهش، یکی در بخش ژن‌های اولیه  $(x_1)$  و دیگری در بخش ژن‌های ثانویه  $(x_2)$ ، صورت می‌گیرد؛ بدین صورت که ابتدا یک ژن از مجموعه ژن‌های اولیه به صورت تصادفی انتخاب و مقدار داخل آن با عددی تصادفی از توزیع یکنواخت ۱ تا  $(C+1)$  تغییر می‌کند. سپس، یک ژن نیز از مجموعه ژن‌های ثانویه به صورت تصادفی انتخاب و مقدار داخل آن با عددی تصادفی از توزیع یکنواخت ۱ تا  $(N+1)$  جایگزین می‌شود.

هر پاسخ به‌عنوان یک کروموزوم دارای  $P \times N$  ( $P$  تعداد دوره و  $N$  تعداد ماشین‌ها) ژن اولیه به‌منظور تعیین جایگاه ماشین‌ها در هر دوره است و  $F$  ژن ثانویه به‌منظور تخصیص ماشین‌ها برای انجام عملیات دارد. شماره سلول ماشین در هر دوره از رابطه ۲۱ و شماره جایگاه از رابطه ۲۲ محاسبه می‌شود که  $M$  تعداد جایگاه موجود در سلول است.

$$\text{رابطه (۲۱)} \quad \hat{u}x_1 = \text{شماره سلول}$$

$$\text{رابطه (۲۲)} \quad \hat{u}M = (x_1 - \hat{u}x_1) \quad \text{شماره جایگاه ماشین}$$

به طوری که  $[ ]$  نماد تابع جزء صحیح و  $[ ]$  نماد تابع سقف هستند. ژن‌ها توسط مقادیر تصادفی پر می‌شوند؛ به عبارت دیگر، برای ژن‌های اولیه  $x_1$ ، اعداد تصادفی با توزیع یکنواخت از ۱ تا  $(C+1)$  و برای ژن‌های ثانویه  $x_2$ ، اعداد تصادفی با توزیع یکنواخت از ۱ تا  $(N+1)$  تولید می‌شوند. تعداد ژن‌های ثانویه ( $F$ ) نیز از رابطه ۲۳ محاسبه می‌شود:

$$\text{رابطه (۲۳)} \quad F = P \times \sum_{k=1}^K L_k$$

در رابطه ۲۳،  $P$  تعداد دوره‌ها،  $K$  تعداد کالاها و  $L_k$  تعداد مراحل پردازش کالای  $k$ ام است. ماشین تخصیص داده‌شده از جزء صحیح هر ژن قرارگرفته در ژن‌های ثانویه به دست می‌آید که به ترتیب تخصیص داده می‌شود.

برای مثال، اگر در یک مسئله ۲ دوره‌ای با ۳ ماشین، ۲ سلول و در هر سلول ۲ جایگاه در نظر گرفته شود و تعداد کالاها برابر ۲ باشد، به طوری که کالای شماره یک دارای ۲ مرحله پردازش و کالای شماره دو دارای ۱ مرحله پردازش باشد، آنگاه تعداد ژن‌های اولیه برابر ۶ و تعداد ژن‌های ثانویه برابر ۶ خواهد بود؛ بنابراین، اگر جوابی مطابق شکل ۵ نمایش داده شود، آنگاه داریم:

۱/۲	۲/۴	۱/۸	۲/۹	۱/۳	۲/۱	۱/۳	۲/۱	۳/۷	۲/۳	۳/۸	۱/۵
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

شکل ۵. نحوه نمایش جواب

در شکل ۵، از سمت چپ ۶ عدد اول به عنوان ژن‌های اولیه و ۶ عدد دوم به عنوان ژن‌های ثانویه هستند که برای مثال، ژن‌های اول و هفتم به ترتیب به صورت زیر تفسیر می‌شوند:

۱/۲: در دوره اول، ماشین اول در سلول ۱ و در جایگاه ۱ قرار می‌گیرد.

$$x_1 = 1/2 = 1$$

$$M = 2 = \text{تعداد جایگاه موجود در سلول}$$

$$x_1 = 1/2 = 1 \text{ شماره جایگاه ماشین}$$

ژن اول در مجموعه ژن‌های ثانویه مربوط به مرحله اول کالای شماره یک در دوره اول و ژن دوم مربوط به مرحله دوم از کالای شماره یک در دوره اول است و بقیه ژن‌ها به همین ترتیب نشان‌دهنده مراحل هستند. حال از اعداد داخل ژن‌ها باید ماشین تخصیص داده شده را استخراج کرد که برای این منظور از جزء صحیح اعداد داخل ژن استفاده می‌شود؛ بنابراین، در مورد تفسیر ژن هفتم می‌توان گفت:

۱/۳: در دوره اول، ماشین ۱ برای تولید مرحله ۱ کالای ۱ تخصیص می‌یابد.

$$x_7 = 1/3 = 1 \text{ شماره ماشین}$$

لازم به یادآوری است که به منظور موجه بودن جواب پس از اجرای عملگر تقاطع، برای جواب‌های غیرموجه جریمه‌ای در نظر گرفته شده است که به تابع هدف هزینه اضافه می‌شود؛ بنابراین، در تکرارهای بعدی، جواب‌های غیرموجه به دلیل بالا بردن هزینه حذف می‌شوند و جواب‌های موجه باقی خواهند ماند.

**شبیه‌سازی تبرید<sup>۱</sup>.** الگوریتم شبیه‌سازی تبرید [۲۲]، یک الگوریتم فراابتکاری مبتنی بر روابط ریاضی و احتمالی برای یافتن نقاط بهینه است؛ به عبارت دیگر، این الگوریتم سعی دارد که تخمین مناسبی از جواب بهینه در فضای کلی جواب بیابد. نام این الگوریتم از فرآیند تبرید در مهندسی مواد گرفته شده است که شامل فرآیند گرم کردن و سپس سرد کردن کنترل شده است تا اندازه کریستال‌ها در ماده افزایش یابد. برای حل یک مسئله بهینه‌سازی، الگوریتم شبیه‌سازی تبرید ابتدا از یک جواب اولیه شروع و سپس در یک حلقه تکرار به سمت جواب‌های همسایه حرکت می‌کند. اگر جواب همسایه بهتر از جواب فعلی باشد، الگوریتم آن را به عنوان جواب فعلی قرار می‌دهد (به سویی آن حرکت می‌کند) و در غیر این صورت، آن جواب را با احتمال  $\exp(-\Delta/T)$  به عنوان جواب فعلی می‌پذیرد [۲۲]. در رابطه اخیر،  $\Delta$  تفاوت میان تابع هدف جواب فعلی و

1. Simulated Annealing (SA)

جواب همسایه است و  $T$  پارامتر دما است که مقدار آن از رابطه ۲۴ محاسبه می‌شود:

$$T_{in} = \frac{f_{min} - f_{max}}{\ln P_c} \quad \text{رابطه ۲۴}$$

در رابطه ۲۴،  $f_{min}$  حد پایین و  $f_{max}$  حد بالا در اولین اجرا است و  $P_c$  مقداری بسیار نزدیک به یک است که در این مقاله، مقدار  $0.95$  برای آن در نظر گرفته شده است [۵]. در این الگوریتم و در گام‌های اولیه، دما بسیار بالا تعریف می‌شود تا احتمال بیشتری برای پذیرش جواب‌های بدتر وجود داشته باشد. به این صورت، امکان فرارگیری در بهینه محلی کاهش می‌یابد. در هر دما، الگوریتم اجرا می‌شود و سپس دما با استفاده از ضریب آلفا به آرامی کاهش داده می‌شود. با کاهش تدریجی دما، در گام‌های پایانی الگوریتم، احتمال کمتری برای پذیرش جواب‌های بدتر وجود خواهد داشت؛ بنابراین، الگوریتم به سمت بهترین جواب همگرا می‌شود [۲۲].

**الگوریتم ترکیبی پیشنهادی (NSGA-II و شبیه‌سازی تبرید).** همان‌طور که در بخش ۴-۱ نیز مطرح شد، در الگوریتم NSGA-II، پس از ایجاد جمعیت جدید از جمعیت جاری با کمک عملگرهای الگوریتم ژنتیک و پس از مرتب‌سازی براساس جواب غالب و فاصله ازدحامی، به تعداد جمعیت جاری از جمعیت جدید حفظ و برای اجرای بعدی الگوریتم استفاده می‌شود. در ترکیب الگوریتم ژنتیک و شبیه‌سازی تبرید، از بین جمعیت جاری مرحله قبل و جمعیت جدید تولیدشده توسط عملگرهای تقاطع و جهش، جمعیت مورد نیاز برای اجرای بعدی با استفاده از شبیه‌سازی تبرید و براساس مقایسه نظیربه‌نظیر انتخاب می‌شود؛ لذا باید حداقل به اندازه جمعیت جاری مرحله قبل، جمعیت جدید برای مرحله بعدی تولید شود. هدف از این تغییر، جلوگیری از قرار گرفتن جواب در بهینه محلی است. گفتنی است که در این روش، اهداف حداقل‌سازی مهم هستند. تعداد جواب‌های جدید تولیدشده باید حداقل به اندازه تعداد جمعیت جاری باشد تا مقایسه نظیربه‌نظیر با جمعیت جاری صورت گیرد. آنگاه پس از مقایسه نظیربه‌نظیر و مرتب‌سازی براساس جواب غالب و فاصله ازدحامی، به اندازه تعداد جمعیت جاری از جمعیت جدید تولیدشده را انتخاب می‌کنیم.

در شکل‌های ۶ و ۷، رویه اجرای الگوریتم ترکیبی پیشنهادی (NSGA-II+SA) ارائه شده است:



قرار دهید  $t = 0$  و جمعیت تصادفی  $P_0$  را به اندازه  $N$ ، به عنوان جمعیت فعلی تولید کنید. توابع هدف را ارزیابی کنید و براساس ارتباط غالب و مغلوب توضیح داده شده در الگوریتم ارائه شده در شکل ۴، آن‌ها را رتبه‌بندی کنید.

قرار دهید  $t = t + 1$  و براساس نرخ تقاطع، تعداد جواب مورد نیاز را به عنوان والدین برای تولید فرزند، با معیار مسابقات دوبه‌دو انتخاب و جمعیت جدید را ایجاد کنید.

براساس نرخ جهش، تعداد جواب مورد نیاز برای جهش را با معیار مسابقات دوبه‌دو انتخاب و جمعیت جدید را ایجاد کنید. جمعیت جدید ایجاد شده را براساس توابع هدف ارزیابی کنید. توجه کنید که مجموع تعداد جواب‌های ایجاد شده حداقل برابر جمعیت اولیه باشد.

جمعیت جدید ایجاد شده را براساس ارتباط غالب و فاصله ازدحام مرتب کنید.

براساس الگوریتم ارائه شده در شکل ۶، جمعیت اولیه برای اجرای بعدی الگوریتم را انتخاب کنید.

در صورت ارضا شدن معیار توقف، اجرای الگوریتم را متوقف و در غیر این صورت، گام ۳ را اجرا کنید.

شکل ۶. گام‌های اجرای الگوریتم پیشنهادی NSGA-II+SA

برای تمام اعضای مجموعه پاسخ‌های ایجاد شده، گام‌های زیر تکرار شود:

۱. نامین عضو از جمعیت اولیه مرتب شده ( $firstpop(i)$ ) و جمعیت جدید مرتب شده ( $newpop(i)$ ) انتخاب شود.
  - ۱-۱. اگر  $newpop(i)$  بر  $firstpop(i)$  غالب بود، به عنوان نامین عضو جمعیت اولیه انتخاب شود.
  - ۱-۲. اگر  $firstpop(i)$  بر  $newpop(i)$  غالب بود، به عنوان نامین عضو جمعیت اولیه انتخاب شود.
  - ۱-۳. اگر هیچ‌یک از این دو عضو بر هم غالب نبودند، آنگاه پس از بدون واحد کردن یا هم واحد کردن اهداف، داریم:
 

رابطه ۲۵ مجموع توابع هدف به ازای  $firstpop(i)$  - مجموع توابع هدف به ازای  $newpop(i)$   $\Delta = newpop(i)$
  - ۱-۳-۱. اگر مقدار  $\Delta$  کوچک‌تر یا مساوی صفر بود، آنگاه  $newpop(i)$  به عنوان نامین عضو جمعیت اولیه انتخاب شود.
  - ۱-۳-۲. در غیر این صورت، براساس شبیه‌سازی تبرید مقدار  $p$  و یک عدد تصادفی مطابق روابط ۲۶ و ۲۷ محاسبه شود:
 

رابطه ۲۶  $p = \exp\left(\frac{-\Delta}{T}\right)$

رابطه ۲۷  $r = rand[0,1]$
- اگر  $r$  کوچک‌تر از  $p$  بود، آنگاه  $newpop(i)$  به عنوان نامین عضو جمعیت اولیه و در غیر این صورت،  $firstpop(i)$  به عنوان نامین عضو جمعیت اولیه انتخاب شود.

شکل ۷. انتخاب جمعیت برای اجرای بعدی الگوریتم پیشنهادی NSGA-II+SA

#### ۴. تحلیل داده‌ها و یافته‌های پژوهش

**تولید مسائل نمونه.** برای مقایسه کارایی دو الگوریتم، از روش پیشنهادی رنجبر بورانی و همکاران (۲۰۱۰) استفاده شده است که در آن، تعدادی مسئله نمونه حل و برای هر مسئله بخشی از بهترین جواب‌ها ذخیره می‌شوند تا مبنای مقایسه دو الگوریتم قرار گیرند. کلیه مسائل توسط نرم‌افزار متلب نسخه ۱۰ کدنویسی و اجرا شده است. این آزمایشات روی یک کامپیوتر لپ‌تاپ با ۲/۶۶ گیگاهرتز، ۴ گیگابایت حافظه و سیستم عامل ویندوز ۷ اجرا شده است. مطابق با جدول ۱، اطلاعات مربوط به پارامترهای ورودی مربوط به ۱۱ مسئله نمونه به صورت تصادفی ایجاد شده است. مسائل نمونه به صورت تصادفی با استفاده از توزیع یکنواخت، تولید و سپس

توسط هر دو الگوریتم NSGA-II و NSGA-II+SA حل شده‌اند [۳۳]. هر مسئله توسط کدی به صورت  $P - C - M - N - K$  نمایش داده می‌شود که  $P$  تعداد دوره،  $C$  تعداد سلول،  $M$  تعداد جایگاه در هر سلول،  $N$  تعداد ماشین و  $K$  تعداد تنوع کالا برای تولید در هر دوره هستند. تقاضای هر نوع کالا، تعداد مراحل و ماشین‌آلاتی که قابلیت پردازش هر مرحله را دارند، به صورت تصادفی ایجاد می‌شوند.

جدول ۱. داده‌های ورودی مسئله

پارامتر	توزیع پارامتر
تقاضا	$Unif(10,55)$
کلیه هزینه‌های فازی مثلثی	$Unif(1,15)$
اندازه دسته	$Unif(2,15)$
فاصله بین سلول و بین جایگاه	$Unif(2,8)$

**تنظیم پارامترها.** برای اجرای الگوریتم‌های NSGA-II و NSGA-II+SA، تعدادی از پارامترهای مهم و اساسی براساس طراحی آزمایشات تاگوچی تعیین می‌شوند. از آنجا که کیفیت و تعداد جواب‌های مؤثر ایجادشده در هر اجرا بسیار مهم است، برای تعیین پارامترها، تعداد جواب‌های نامغلوب ایجادشده به عنوان نتیجه مطلوب در نظر گرفته می‌شوند. این مقدار از مقایسه جواب‌های ایجادشده در تمام آزمایشات به دست می‌آید. بدین منظور، مسئله‌ای را برای ۴ دوره، ۴ سلول، ۴ جایگاه در هر سلول، ۶ ماشین و ۵ کالا که دیگر پارامترها به صورت تصادفی تولید شدند، در نظر گرفته شد که بهترین ترکیب برای پارامترهای دو الگوریتم براساس طرح آرایه متعامد  $L_{16}$  مطابق با جدول ۲ است:

جدول ۲. پارامترهای تنظیم‌شده الگوریتم‌های NSGA-II و NSGA-II+SA

الگوریتم	نرخ جهش ( $P_m$ )	نرخ تقاطع ( $P_r$ )	جمعیت اولیه ( $NP$ )	تعداد تکرار ( $it$ )	آلفا ( $\alpha$ )
NSGA-II	۰/۱۵	۰/۹	۵۰	۲۵۰	---
NSGA-II+SA	۰/۱۵	۰/۸۵	۵۰	۲۷۵	۰/۹

علاقه‌مندان برای مطالعه بیشتر در خصوص تنظیم پارامترهای الگوریتم‌های فراابتکاری توسط طراحی آزمایشات تاگوچی، می‌توانند به مطالعات کومار و همکاران (۲۰۱۵) و ژانگ و همکاران (۲۰۱۵) مراجعه نمایند.

شاخص‌های اندازه‌گیری به منظور مقایسه نتایج حاصل از دو الگوریتم NSGA-II و

NSGA-II+SA. برای اعتبارسنجی نتایج حاصل از الگوریتم پیشنهادی و همچنین میزان کارایی عملکرد آن، براساس تعدادی از شاخص‌های اندازه‌گیری روش‌های چن هدفه که در ادبیات موضوع بسیار پرکاربرد هستند، مقایسه‌ای میان نتایج به‌دست‌آمده الگوریتم‌های NSGA-II و NSGA-II+SA انجام می‌شود. این شاخص‌های اندازه‌گیری به‌صورت زیر تعریف می‌شوند:

**فاصله نسل‌ها<sup>۱</sup>.** مفهوم فاصله نسل‌ها توسط ون ولدهویزن و لامونت (۱۹۹۸)، به‌عنوان روشی برای برآورد اینکه عناصر مجموعه‌ای از بردارهای نامغلوب چقدر از صفر فاصله دارند، معرفی شد که به‌صورت رابطه ۲۸ محاسبه می‌شود:

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad \text{رابطه (۲۸)}$$

به طوری که  $d_i^2$  فاصله اقلیدسی مقادیر تابع هدف هریک از بردارهای نامغلوب تا نقطه صفر در فضایی به ابعاد تعداد توابع هدف است. مقدار فاصله نسل‌ها بیانگر میانگین مجموع فواصل اقلیدسی است که می‌توان با توجه به آن، دوری یا نزدیکی مرز پارتو را نسبت به مرز پارتو بهینه مقایسه کرد.

**فاصله‌بندی<sup>۲</sup>.** شاخص فاصله‌بندی می‌تواند در تشخیص یکنواختی پراکندگی جواب‌ها در منحنی پارتو استفاده شود [۲۶]. این شاخص اندازه‌گیری را برای محاسبه واریانس بردارهای همسایگی در بردارهای نامغلوب در یک منحنی پارتو به‌صورت رابطه ۲۹ می‌توان تعریف کرد:

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad \text{رابطه (۲۹)}$$

به طوری که:

$$d_i = \min_j \left( \sum_{t=1}^T |f_t^i(\mathbf{x}) - f_t^j(\mathbf{x})| \right), \quad i, j = 1, \dots, n, i \neq j \quad \text{رابطه (۳۰)}$$

در رابطه ۲۹،  $\bar{d}$  مقدار میانگین  $d_i$ ها،  $n$  تعداد بردارهای نامغلوب در منحنی پارتو و  $T$  تعداد توابع هدف هستند. ابتدا  $d_i$ ها توسط رابطه ۳۰ محاسبه می‌شوند که همان مقدار فاصله تا نزدیک‌ترین همسایگی جواب نام است.  $S$  که همان شاخص فاصله‌بندی است، توسط رابطه ۲۹ به‌دست می‌آید. هرچه مقدار  $S$  به صفر نزدیک‌تر باشد، نشان‌دهنده پراکندگی بیشتر و بهتر بودن

1. Generational Distance (GD)  
2. Spacing

جواب‌ها است.

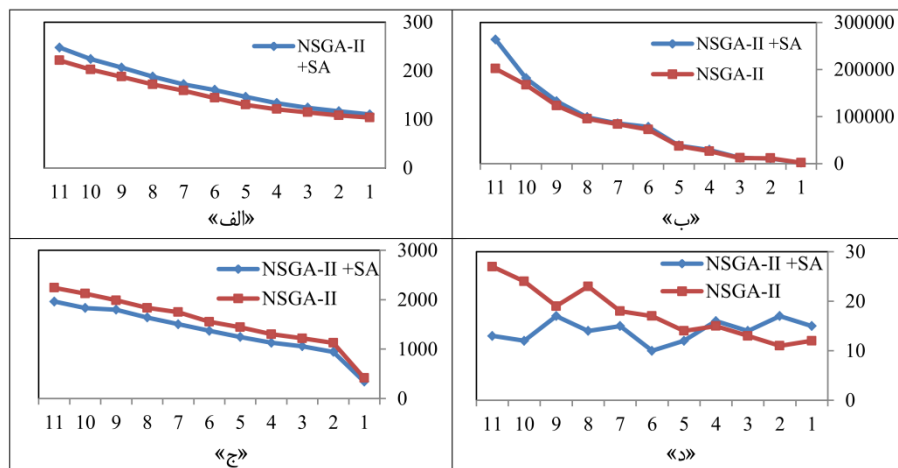
شاخص کیفیت<sup>۱</sup>. شاخص تعداد جواب‌های پارتو به‌تنهایی شاخص مناسبی برای مقایسه دو الگوریتم NSGA-II و NSGA-II+SA نیست؛ زیرا مشخص نیست که چه تعداد از جواب‌های پارتو ارائه‌شده توسط یک الگوریتم کیفیت بالایی دارند. شاخص کیفیت با کنار هم گذاشتن جواب‌های غالب تولیدشده هر دو الگوریتم مذکور و محاسبه نسبت سهم هر روش در تعداد جواب‌های پارتو نهایی قابل محاسبه است [۳۸، ۳۹].

نتایج حاصل از حل مسائل نمونه و اعتبارسنجی. در جدول ۳، نتایج به‌دست‌آمده از ۱۱ مسئله نمونه توسط الگوریتم‌های NSGA-II و NSGA-II+SA برحسب شاخص‌های عملکردی (کیفیت، فاصله‌بندی و فاصله نسل‌ها) و زمان اجرا ارائه شده است.

جدول ۳. عملکرد الگوریتم‌های NSGA-II و NSGA-II+SA در حل مسائل نمونه

شماره مسئله	ابعاد مسئله	زمان CPU		فاصله نسل‌ها		فاصله‌بندی		شاخص کیفیت	
		NSGA-II	NSGA-II+SA	NSGA-II	NSGA-II+SA	NSGA-II	NSGA-II+SA	NSGA-II	NSGA-II+SA
۱	۲-۲-۲-۲-۲	۱۰۳/۹	۱۱۰/۳	۱۹۸۴/۶	۱۹۶۷/۷	۴۱۷	۳۴۲	۱۲	۱۵
۲	۲-۳-۲-۵-۳	۱۰۸/۴	۱۱۶/۸	۱۱۶۶۳	۱۱۰۹۴	۱۱۲۷/۳	۹۴۳/۵	۱۱	۱۷
۳	۲-۳-۴-۶-۳	۱۱۴/۶	۱۲۴/۲	۱۲۱۵۸	۱۲۳۰۱	۱۲۱۸/۴	۱۰۵۶/۹۳	۱۳	۱۴
۴	۳-۳-۴-۶-۳	۱۲۱/۲	۱۳۳/۷	۲۶۳۶۶	۲۹۲۲۰	۱۳۰۱/۴	۱۱۲۹/۹۴	۱۵	۱۶
۵	۳-۳-۴-۶-۴	۱۳۰/۳	۱۴۶/۵	۳۷۲۴۷	۳۸۴۶۵	۱۴۴۳/۵۸	۱۲۴۴/۶۷	۱۴	۱۲
۶	۳-۳-۴-۸-۴	۱۴۴/۲	۱۶۰/۹	۷۲۶۵۸	۷۸۳۳۶	۱۵۵۴/۷	۱۳۷۱/۱۸	۱۷	۱۰
۷	۴-۴-۴-۸-۴	۱۵۹/۴	۱۷۲/۳	۸۳۸۳۷	۸۵۳۷۷	۱۷۴۷/۸۳	۱۵۰۴/۳۴	۱۸	۱۵
۸	۵-۴-۴-۸-۵	۱۷۲/۱	۱۸۸/۱	۹۵۲۳۳	۹۸۷۶۲	۱۸۳۴/۳۸	۱۶۳۹/۲۵	۲۳	۱۴
۹	۵-۴-۴-۹-۵	۱۸۷/۹	۲۰۶/۷	۱۲۳۴۰۸	۱۲۳۶۳۹	۱۹۸۷/۵۴	۱۷۹۷/۷۶	۱۹	۱۷
۱۰	۶-۴-۴-۹-۶	۲۰۲/۷	۲۲۴/۲	۱۶۷۳۵۱	۱۸۱۶۳۷	۲۱۲۴/۴	۱۸۳۲/۴	۲۴	۱۲
۱۱	۶-۵-۴-۹-۶	۲۲۱/۵	۲۴۷/۹	۲۰۱۷۶۴	۲۶۳۸۵۱	۲۲۴۱/۴	۱۹۶۳/۳	۲۷	۱۳

1. Quality Metric (QM)



شکل ۸. «الف» مربوط به زمان CPU، «ب» مربوط به فاصله نسل‌ها، «ج» مربوط به فاصله‌بندی و «د» مربوط به کیفیت

با توجه به نتایج ارائه‌شده در جدول ۳ که شاخص‌های مربوط به عملکرد دو الگوریتم NSGA-II و NSGA-II+SA در حل ۱۱ مسئله نمونه را نشان می‌دهد و با توجه به نمودار «الف» در شکل ۸، زمان اجرای الگوریتم پیشنهادی بیشتر از زمان اجرا برای الگوریتم NSGA-II است که این به دلیل اضافه شدن چرخه انتخاب براساس شبیه‌سازی تبرید و همچنین تکرار بیشتر اجرای الگوریتم، قابل پیش‌بینی است. نمودار «ب» در شکل ۸، مقادیر شاخص فاصله نسل‌ها را نشان می‌دهد. این نمودار مبین کارایی بهتر الگوریتم NSGA-II+SA برای شاخص فاصله نسل‌ها در مسائل با اندازه کوچک است. در نمودار «ج» شکل ۸، عملکرد شاخص فاصله‌بندی برای دو الگوریتم NSGA-II و NSGA-II+SA، به نمایش درآمده است؛ به‌گونه‌ای که از نظر ایجاد جواب غالب با پراکندگی مناسب، الگوریتم پیشنهادی به مراتب بهتر از الگوریتم NSGA-II عمل می‌کند. درنهایت، شاخص کیفیت که در نمودار «د» شکل ۸ مشخص شده است، مبین کارایی بهتر الگوریتم پیشنهادی برای ایجاد جواب‌های غالب در مسائلی با اندازه کوچک و کارایی بهتر الگوریتم NSGA-II برای ایجاد جواب‌های غالب در مسائلی با اندازه بزرگ است.

گفتنی است که حل مسئله مطرح‌شده در روابط ۱۲-۱۸ توسط نرم‌افزار لینگو و روش شمارش کامل نیز بررسی شد که نتایج آن به شرح زیر است:

– مسئله مطرح‌شده در روابط ۱۲-۱۸ با کوچک‌ترین ابعاد تعداد دوره، تعداد سلول، تعداد جایگاه در هر سلول، تعداد ماشین و تعداد تنوع کالا، همگی برابر ۲، در بسته نرم‌افزاری لینگو کدنویسی شد که نهایتاً پس از اجرای حل مسئله، پیغام «جواب موجه در دسترس نمی‌باشد» ظاهر شد؛ به عبارت دیگر، چنانچه ابعاد مسئله در ساده‌ترین شکل به صورت ۲-۲-۲-۲-۲ تعریف شود،

دسترسی به جواب مسئله از طریق نرم‌افزار لینگو امکان‌پذیر نخواهد بود. - با توجه به متغیرهای مسئله مطرح‌شده در روابط ۱۲-۱۸، چنانچه ابعاد مسئله در ساده‌ترین شکل به صورت تعداد دوره، تعداد سلول، تعداد جایگاه در هر سلول، تعداد ماشین و تعداد تنوع کالا، همگی برابر ۲ تعریف شود، آنگاه در روش شمارش کامل، تعداد جایگشت‌های هر متغیر و مجموع تعداد کل جایگشت‌ها به صورت جدول ۴ است:

جدول ۴. تعداد جایگشت‌های مسئله در ساده‌ترین شکل (همگی اندیس‌ها برابر با ۲)

نام متغیر	تعداد جایگشت
$e_{pxix}$	$2 \times 2 \times 2 \times 2 = 16$
$e_{pwyj}$	$1 \times 2 \times 2 \times 2 = 8$
$O_{pklj}$	$1 \times 2 \times 2 \times 1 = 4$
$O_{pk(l-1)i}$	$1 \times 1 \times 1 \times 1 = 1$
مجموع	$16 \times 8 \times 4 \times 1 = 512$

در جدول ۴، تعداد کل جایگشت‌ها برابر با ۵۱۲ است؛ به عبارت دیگر، تعداد تمامی پاسخ‌های موجه و غیرموجه برای مسئله بالا برابر ۵۱۲ است. چنانچه مدت زمان هر بار ویرایش مسئله، جایگذاری مقدار متغیرها در مسئله و محاسبه مقادیر توابع هدف در کمترین زمان مورد انتظار برابر با ۹۰ ثانیه باشد، آنگاه در خوش‌بینانه‌ترین حالت، مدت زمان بررسی کل حالات در روش شمارش کامل، ۱۲/۸ ساعت طول خواهد کشید:

$$\text{ثانیه } 512 \times 90 = 46080$$

$$\text{ساعت } 46080 \div 3600 = 12/8$$

$$\text{شبه‌روز } 12/8 \div 24 = 0/54$$

نتیجه به دست آمده درخصوص استفاده از روش شمارش کامل نشان می‌دهند که با تعریف ابعاد مسئله در ساده‌ترین شکل ۲-۲-۲-۲، نیازمند ۱۲/۸ ساعت یا به عبارتی ۰/۵۴ شبه‌روز خواهیم بود تا همه حالات بررسی شوند و جواب بهینه حاصل شود. به این ترتیب، با توجه به نتایج به دست آمده، استفاده از الگوریتم پیشنهادی به منظور حل مسائل اندازه کوچک، از نظر کاهش حجم و زمان انجام محاسبات و همچنین امکان دستیابی به جواب بهینه می‌تواند قابل توجیه باشد.

## ۵. نتیجه‌گیری و پیشنهادها

در این مقاله، یک مدل چندهدفه برای طراحی چیدمان سلولی و تخصیص کار به ماشین با در نظر گرفتن هزینه‌های فازی، در زمانی که حجم و تنوع تقاضا پویا فرض می‌شود، پیشنهاد شد. سپس الگوریتم NSGA-II با ترکیب الگوریتم‌های ژنتیک و شبیه‌سازی تبرید برای حل مسئله تشکیل سلول تولیدی، با توجه به اهداف و محدودیت‌های تعریف‌شده پیشنهاد شد. به منظور سنجش کارایی الگوریتم پیشنهادی، جواب‌های به‌دست‌آمده از حل تعدادی از مسائل نمونه که به صورت تصادفی انتخاب شده بودند، از نظر شاخص‌های زمان CPU، فاصله نسل‌ها، فاصله‌بندی و کیفیت، بررسی و با نتایج الگوریتم NSGA-II مقایسه شدند. نتایج حاصل از مقایسات نشان دادند که الگوریتم پیشنهادی در ایجاد جواب‌های غالب در مسائلی با اندازه کوچک، دارای کارایی بهتری نسبت به NSGA-II است. همچنین، به منظور بررسی دشواری و زمانبر بودن حل مسئله در ابعاد کوچک و توجیه کارایی الگوریتم پیشنهادی در حل این‌گونه مسائل، مسئله‌ای در ساده‌ترین شکل مطرح شد که نتایج به‌دست‌آمده از حل آن با نرم‌افزار لینگو و روش شمارش کامل، مؤید عملکرد مطلوب الگوریتم پیشنهادی در کاهش حجم و زمان انجام محاسبات و همچنین امکان دستیابی به جواب بهینه بودند. برای انجام مطالعات آتی در خصوص حل مسائل تشکیل سلول‌های تولیدی، استفاده از سایر الگوریتم‌های فراابتکاری، نظیر بهینه‌سازی ازدحام ذرات چندهدفه را برای دستیابی به جواب‌های بهتر می‌توان پیشنهاد داد. همچنین، با توجه به وجود ریسک‌های موجود در سیستم‌های تولیدی، حداقل‌سازی این ریسک‌ها به‌عنوان یکی از اهداف مسئله چندهدفه ارائه‌شده در این مقاله، پیشنهاد می‌شود.

## منابع

1. AramoonBajestani, M., Rabbani, M., A.R. Rahimi-Vahed, & BaharianKhoshkhou, G. (2009). A multi-objective scatter search for a dynamic cell formation problem. *International Journal of Computers & Operations Research*, 36, 777-794.
2. Arikan, F., Güngör, Z. (2009). Modeling of a manufacturing cell design problem with fuzzy multi-objective parametric programming. *Mathematical and Computer Modelling*, 50, 407-420.
3. Askin, R., & Chiu, K. (1990). A Genetic paratitioning procedure for machine assignment and cell formation in group technology. *International Journal of Production Research*, 1555-1572.
4. Askin, R.G., Selim, H.M., & Vakharia, A.J. (1997). A methodology for designing flexible cellular manufacturing systems. *IIE Transactions*, 29, 599-610.
5. Baykasoglyu, A., & Gindy, N.Z. (2001). A simulated annealing algorithm for dynamic layout problem. *Computers and Operations Research*, 28, 1403-1426.
6. Boulif, M., & Atif, K. (2008). A new fuzzy genetic algorithm for the dynamic bi-objective cell formation problem considering passive and active strategies. *International Journal of Approximate Reasoning*, 47, 141-165.
7. Carlsson, C., & Fuller, R. (2001). On possibilistic mean value and variance of fuzzy numbers. *Fuzzy Sets and Systems*, 122, 315-326.
8. Chattopadhyay, M., Sengupta, S., Ghosh, T., Dan, P.K., Mazumdar, S. (2013). Neuro-genetic impact on cell formation methods of Cellular Manufacturing System design: A quantitative review and analysis. *Computers & Industrial Engineering*, 64, 256-272.
9. Chen C.-T., Lin, C.-T., & Huang, S.-F. (2006). A fuzzy approach for supplier evaluation and selection in supply chain management. *International Journal of Production Economics*, 102, 289-301.
10. Cochran, W.G., & Cox, G.M. (1992). *Experimental Designs*. New York: Wiley, 2nd Ed.
11. Das, K., Lashkari, R.S., & Sengupta, S. (2007). Reliability consideration in the design and analysis of cellular manufacturing systems. *International Journal of Production Economics*, 105, 243-262.
12. Deb, K., Samir, A., Amrit, P., & Meyarivan, T. (2002). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-2. *IEEE Transactions on Evolutionary Computation*, 6, 103-112.
13. Dubois, D., & Prade, H. (1987). The mean value of a fuzzy number. *Fuzzy Sets and Systems*, 24, 279-300.
14. Ghotboddini, M., Rabbani, M., & Rahimian, H. (2011). A comprehensive dynamic cell formation design: Benders' decomposition approach. *Expert Systems with Applications*, 38, 2478-248.
15. Hedayat, A.S., Sloane, N.J.A., & Stufken, J. (1999). *Orthogonal Arrays: Theory and Applications*. New York: Springer-Verlag.
16. Qing-dao-er-ji, R., & Wang, Y. (2014). Security based bi-objective flow shop scheduling model and its hybrid genetic algorithm. *Applied Mathematics and Computation*, 243, 637-643.
17. KamaliDolatAbadi, A.H., pasandideh, H.R., & Abdi Khalife, M. (2009). Design and layout of cellular manufacturing system in dynamic



condition. *Journal of Industrial Engineering*, 1-12.

18. Kia, R., Paydar, M.M., Alimardany Jondabeh, M., Javadian, N., & Nejatbakhsh, Y. (2011). A fuzzy linear programming approach to layout design of dynamic cellular manufacturing systems with route selection and cell reconfiguration. *International Journal of Management Science and Engineering Management*, 6, 219-230.

19. Kia, R., Khaksar-Haghani, F., Javadian, N., & Tavakkoli-Moghaddam, R. (2014). Solving a multi-floor layout design model of a dynamic cellular manufacturing system by an efficient genetic algorithm. *Journal of Manufacturing Systems*, 33, 218-232.

20. Kumar, S., Chary, G.H.V.C., & Dastidar, M.G. (2015). Optimization studies on coal-oil agglomeration using Taguchi (L16) experimental design. *Fuel*, 141, 9-16.

21. Kusiak, A. (1987). The generalized group technology concept. *International Journal of Production Research*, 24, 561-569.

22. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., & Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21, 1087-1092.

23. Montgomery, D.C. (2000). *Design and Analysis of Experiments*, New York: Wiley, 5th Ed.

24. Ranjbar-Bourani, M., Tavakkoli-Moghaddam, R., Amoozad-Khalili, H., & Hashemian, S.M. (2010). Applying scatter search algorithm based on TOPSIS to multi-objective cellular manufacturing system design. *International Journal of Industrial Engineering*, 21, 11-21.

25. Ruiz, R., Maroto, C., & Alcaraz, J. (2005). Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics. *European Journal of Operational Research*, 165, 34-54.

26. Schott, J. R. (1995). *Fault tolerant design using single and multicriteria genetic algorithm optimization*. M.S. thesis, Dept. Aeronautics and Astronautics, Massachusetts Inst. Technol., Cambridge.

27. Shafer, S., & Charnes, J. (1994). Cellular versus functional layout under a variety of shop operating conditions. *Decision sciences*, 24, 665-682.

28. Singh, N., & Rajamani, D. (1996). *Cellular Manufacturing System: design, planning and control* Chapman and Hall. New York: Chapman & Hall.

29. Solimanpur, M., Vrat, P., & Shankar, R. (2004). A multiobjective genetic algorithm approach to the design of cellular manufacturing systems. *International Journal of Production Research*, 42, 1419-1441.

30. Taguchi, G. (1990). *Introduction to Quality Engineering*. Tokyo: Asian Productivity Organization.

31. Tavakkoli-Moghaddam, R., & Aryanezhad, M. (2007). A new approach for the cellular manufacturing problem in fuzzy dynamic conditions by a genetic algorithm. *Journal of Intelligent and Fuzzy Systems*, 18, 363-376.

32. Van Veldhuizen, D.A., & Lamont, G.B. (1998). Multiobjective evolutionary algorithm research: A history and analysis. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

33. Venugopal, V., & Narendran, T. (1992). A genetic algorithm approach to the machine Component grouping problem with multiple objectives. *Computer & Industrial Engineering*, 22, 469-480.

34. Xu, H., & Wang, H.P.B. (1989). Part family formation for group technology applications based on fuzzy mathematics. *International Journal of Production Research*, 27, 1637-1651.
35. Yasuda, K., Hu, L., & Yin, Y. (2005). A grouping genetic algorithm for the multiobjective cell formation problem. *International Journal of Production Research*, 43, 829-853.
36. Zadeh, L.A. (1965). Fuzzy sets. *Information and Control*, 8, 338-353.
37. Zhang, F.B., Wang, Z.L., & Yang, M.Y. (2015). Assessing the applicability of the Taguchi design method to an interrill erosion study. *Journal of Hydrology*, 521, 65-73.
38. Zitzler, E., & Thiele, L. (1999). Multi objective evolutionary algorithms: A comparative case study and strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3, 257-271.
39. Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multi objective evolutionary algorithms: Empirical results. *Journal of Evolutionary Computation*, 8, 173-195.