

یک روش تلفیقی جدید جهت تخصیص افزونگی در سیستم‌های تولیدی با استفاده از NSGA-II و MOPSO اصلاح شده

علی محتشمی*

(تاریخ دریافت: ۹۲/۶/۲۳ - تاریخ پذیرش: ۹۳/۱/۱۷)

چکیده

این مقاله به معرفی یک مدل ریاضی چندهدفه جهت تخصیص افزونگی در سیستم‌های تولیدی می‌پردازد. در بسیاری از خطوط تولید و مونتاژ در صنعت، توابع توزیع ورودی قطعاً، مدت زمان‌های پردازش، مدت زمان تابا زمانی ماشین‌ها و مدت زمانهای تعمیر از توابع توزیع عمومی تبعیت میکنند. روش پیشنهادی این مقاله با استفاده از رویکرد تلفیقی شبیه‌سازی کامپیوتری و متدولوژی سطح پاسخ، قابلیت در نظرگیری پارامترهای زمانی مبتنی بر توابع توزیع عمومی در خطوط تولید را داراست. در مدل ریاضی این مقاله سه هدف حداکثر کردن نرخ تولید، حداقل کردن هزینه‌ها و حداکثر کردن کیفیت محصولات در نظر گرفته شده است. جهت حل مدل ریاضی چندهدفه پیشنهادی، از دو الگوریتم فراابتکاری تکاملی الگوریتم ژنتیک با مرتب‌سازی نامغلوب و بهینه‌سازی ازدحام ذرات چندهدفه استفاده شده است. نتایج محاسباتی مؤثر بودن دو الگوریتم فوق در تولید جواب‌های نامغلوب برای مسئله تخصیص افزونگی در سیستم‌های تولیدی نامطمئن را نشان می‌دهد. به علاوه، نتایج حاصل از مقایسه این دو الگوریتم نشان‌دهنده کیفیت بالاتر جواب‌های الگوریتم ژنتیک با مرتب‌سازی نامغلوب در این مسئله است.

کلمات کلیدی: خط تولید، متدولوژی سطح پاسخ، شبیه‌سازی، الگوریتم ژنتیک با مرتب‌سازی نامغلوب، بهینه‌سازی ازدحام ذرات چندهدفه

مقدمه

یک افزونگی جزئی است که به یک زیرسیستم اضافه می‌شود تا مسیرهای حرکت را افزایش داده و در نتیجه منجر به افزایش قابلیت اطمینان سیستم، خروجی سیستم یا ظرفیت سیستم شود (کو و همکاران، ۲۰۰۱). طراحی بهینه سیستم‌هایی که از اجزاء مختلفی تشکیل شده‌اند عبارت‌است از تعیین ساختار و پیکربندی سیستم به نحوی که بالاترین قابلیت اطمینان، خروجی یا ظرفیت را با کمترین هزینه به همراه داشته باشد. در ادبیات موضوع مسائل بهینه‌سازی قابلیت اطمینان، سه نوع مسئله مختلف مبتنی بر نوع متغیر تصمیم‌گیری مسئله وجود دارد که عبارتند از تخصیص قابلیت اطمینان، تخصیص افزونگی، و تخصیص قابلیت اطمینان-افزونگی (راوی و ردی، ۲۰۰۰؛ کو و پراساد، ۲۰۰۰؛ کو و وان، ۲۰۰۷).

مدل کلی مسئله بهینه‌سازی قابلیت اطمینان را می‌توان به صورت زیر فرموله کرد:

$$\text{Maximize } f(r, x) \quad (1)$$

subject to

$$g_i(r, x) \leq b_i \text{ for } i = 1, 2, \dots, m \quad (2)$$

$$l_j \leq x_j \leq u_j, \quad x_j \text{ is a positive integer number for } j = 1, 2, \dots, n \quad (3)$$

$$r_j \in (0, 1) \text{ for } j = 1, 2, \dots, n \quad (4)$$

به طوری که n و m به ترتیب نشان‌دهنده تعداد اجزاء و تعداد محدودیت‌ها است؛ r_j نشان‌دهنده قابلیت اطمینان جزء j ام؛ x_j تعداد افزونگی‌های یکسان؛ g_i محدودیت i ام؛ و b_i

1- Redundancy

2- Kuo

3- Ravi

4- Reddy

5- Kuo

6- Prasad

7- Kuo

8- Wan

حداکثر تعداد مجاز منبع نام است. اگر قابلیت اطمینان، z برای تمام z ها تنها متغیرهای تصمیم باشند، مسئله به عنوان یک مسئله تخصیص قابلیت اطمینان نامیده می شود. اگر تعداد افزونگی ها، x_j برای تمام z ها تنها متغیرهای مسئله باشند، مسئله تبدیل به یک مسئله تخصیص افزونگی (RAP) می شود؛ و اگر متغیرهای تصمیم مسئله شامل هر دوی قابلیت اطمینان اجزاء و تعداد افزونگی ها باشد، مسئله به عنوان مسئله تخصیص قابلیت اطمینان - افزونگی نامیده می شود (یه ، ۲۰۰۹). از آنجایی که یک اصل بسیار مشهور وجود دارد مبنی بر اینکه افزونگی اجزاء مؤثر تر از افزونگی سیستم است، هدف RAP محدود به تعیین تعداد اجزاء افزونگی است. در بسیاری از مطالعات به مسئله RAP توجه بسیار زیادی شده است چراکه بهبود قابلیت اطمینان اجزاء در عمل بسیار دشوار است (یون و کیم ، ۲۰۰۴؛ یون و همکاران، ۲۰۰۷؛ گرشوین و اسکور ، ۲۰۰۰؛ دالری و گرشوین، ۱۹۹۲؛ ابدول کادر ، ۲۰۰۶؛ بورمن ، ۱۹۹۵). باین وجود، هر چند که اضافه کردن اجزاء به سیستم می تواند منجر به بهبود قابلیت اطمینان یا ظرفیت سیستم شود اما هزینه های سیستم را نیز افزایش می دهد. بنابراین، اهداف معمول در مسائل RAP ، دستیابی به قابلیت اطمینان بالا یا توان عملیاتی بالا با حداقل هزینه است.

مسئله RAP تاکنون مورد توجه بسیاری از محققین بوده است (جونگ و کیم ، ۲۰۰۰؛ ناهاس و همکارانش، ۲۰۰۷؛ دالری و فرین ، ۱۹۹۸؛ کومار و همکارانش، ۲۰۰۹؛

1- Redundancy Allocation Problem (RAP)

2- Yeh

3- Yun

4- Kim

5- Gershwin

6- Schor

7- Dallery

8- Abdul - Kader

9- Burman

10- Jeong

11- Kim

12- Nahas

13- Dallery

14- Frein

15- Kumar

اوزینب و همکارانش، ۲۰۰۸؛ ذوالفقاری و همکارانش ۲۰۰۸؛ گارگ و همکارانش، ۲۰۱۳؛ ژاو و همکارانش، ۲۰۱۳؛ هه و همکارانش، ۲۰۱۳).

در این مقاله سعی شده برای مسئله *RAP* در سیستم‌های تولیدی روشی ارائه شود تا محدودیت‌ها و نواقص مدل‌های قبلی تا حدی برطرف شود و از در نظرگیری مفروضات ساده‌کننده که مدل‌ها را از واقعیت سیستم‌های تولیدی دور می‌کند اجتناب شده و مدل جامع‌تری نسبت به تحقیقات پیشین ارائه شود. در این مقاله یک مدل ریاضی چندهدفه با توابع هدف حداکثرسازی نرخ تولید، حداقل‌سازی هزینه‌ها و حداکثرسازی کیفیت محصولات ارائه می‌شود. با توجه به اینکه با اضافه شدن ماشین‌های جدید و با تکنولوژی جدید و پیشرفته‌تر، تولید قطعات نامنطبق در ایستگاه‌های کاری کاهش می‌یابد، بنابراین با کاهش قطعات نامنطبق، محصولات نامنطبق نیز کاهش می‌یابد. بنابراین در این مقاله برخلاف تحقیقات پیشین، علاوه بر توابع هدف حداکثرسازی نرخ تولید و حداقل‌سازی هزینه‌ها، تابع هدف حداکثرسازی کیفیت نیز در نظر گرفته می‌شود. از طرف دیگر در تحقیقات پیشین، همچنان که برخی از آن‌ها ذکر شد، زمان‌های پردازش، زمان بین‌خرابی‌ها و زمان‌های تعمیر به صورت قطعی یا یک تابع توزیع آماری مشخص در نظر گرفته شده است و ظرفیت تولید بر اساس یک تابع توزیع خاص با ارائه یک روش ابتکاری محاسبه شده است، در حالی که اگر توابع توزیع آماری تغییر یابد، روش‌های ابتکاری مطرح شده دیگر قادر به تخمین ظرفیت تولید برای ترکیب‌های مختلف ماشین‌ها در ایستگاه‌ها نیست. روش پیشنهادی مقاله حاضر از آنجایی که جهت تحلیل خط تولید مبتنی بر شبیه‌سازی کامپیوتری و طراحی آزمایش‌ها است، قابلیت در نظرگیری هر تابع توزیعی برای زمان‌های پردازش، زمان‌های بین‌خرابی‌ها و زمان‌های تعمیر را دارا بوده و همچنین از آنجایی که جهت تحلیل خط تولید از شبیه‌سازی استفاده می‌کند، توانایی بررسی سیستم‌های بسیار پیچیده تولیدی را نیز دارا است.

-
- 1- Ozineb
 - 2- Zoulfaghari
 - 3- Garg
 - 4- Zhao
 - 5- He

این مقاله به صورت زیر سازماندهی شده است. بخش دوم به ارائه مدل ریاضی پیشنهادی می پردازد. در بخش سوم روش پیشنهادی حل مسئله ارائه می شود. در بخش چهارم مثال عددی جهت نشان دادن عملکرد مدل ریاضی پیشنهادی و روش پیشنهادی حل ارائه می شود و در نهایت بخش پنجم به نتیجه گیری کلی مقاله اختصاص دارد.

مدل ریاضی پیشنهادی

همچنان که پیشتر ذکر شد، این مقاله درصدد است، یک مدل ریاضی چندهدفه پیشنهادی جهت تخصیص افزونگی در خطوط تولید و مونتاژ نامطمئن را ارائه کند. در واقع مدل ریاضی این مسئله در پی یافتن تعداد بهینه (یا نزدیک بهینه) ماشین های موازی در هر ایستگاه است به نحوی که اهداف مسئله، شامل حداکثرسازی نرخ تولید، حداقل سازی هزینه ها و حداکثرسازی کیفیت، در حد مطلوبی قرار گیرند و در ادامه این مدل ریاضی پیشنهادی تشریح می شود.

تابع هدف اول مدل ریاضی

از آنجایی که هدف اول حداکثر کردن نرخ تولید بوده و نرخ تولید تابعی از تعداد و نوع ماشین های مورد استفاده در ایستگاه های کاری است، داریم:

$$\text{Maximize } Z_1 = f(x_{ij}) \quad (5)$$

رابطه (۵) یک برآوردگر نرخ تولید است که تابعی از تعداد و نوع ماشین های نوع Z است که باید در ایستگاه i وجود داشته باشد (x_{ij}) . ماشین نوع Z می تواند از جمله ماشین های موجود در ایستگاه باشد یا از جمله ماشین های جدیدی که می تواند به ایستگاه اضافه شود. از آنجایی که زمان های پردازش، زمان بین خرابی ها و زمان های تعمیر ماشین ها به صورت تابع توزیع عمومی در نظر گرفته شده اند که در ادامه به ارائه روش پیشنهادی تخمین نرخ تولید برای ترکیبات مختلف ماشین ها می پردازد. منظور از ترکیبات مختلف، پیکربندی های

مختلف خط تولید است که نشان‌دهنده تعداد و نوع ماشین‌های موازی در هر ایستگاه است. که در ادامه مقاله به جای واژه پیکربندی از واژه ترکیب استفاده می‌شود.

روش پیشنهادی تخمین نرخ تولید

* شبیه‌سازی خط تولید

شبیه‌سازی گسسته - پیشامد یک راه مؤثر تخمین عملکرد تقریباً تمامی سیستم‌ها است به شرط اینکه داده‌های ورودی از صحت مناسبی برخوردار باشند (لاوویه و همکاران، ۲۰۰۹). در واقع شبیه‌سازی کامپیوتری قابلیت ارزیابی سیستم‌های بسیار پیچیده با در نظر گیری هر تابع توزیع آماری و روابط پیچیده مؤلفه‌های سیستم را دارا است. بنابراین می‌توان از این رویکرد برای ارزیابی خط تولید و تخمین نرخ تولید برای ترکیب‌های مختلف بهره برد.

* تحلیل متا مدل

یک مدل شبیه‌سازی بیانی از سیستم واقعی است، در حالی که واژه متا مدل در اینجا به معنی یک تخمین ریاضی مدل شبیه‌سازی است (کلیجنن و سرجنت، ۲۰۰۰؛ کلیجنن، ۱۹۸۷). تعداد زیادی از توابع ریاضی برای توسعه متامدل‌ها استفاده شده است. نوعی از متامدل‌ها که استفاده زیادی در مطالعات شبیه‌سازی داشته است، مدل‌های چند جمله‌ای رگرسیونی است (مونتگومری، ۲۰۰۱). در این مطالعه، طرح آزمایش کسری دو سطحی (برای غربال کردن عامل‌ها) و طرح سه سطحی *Box-Behnken* (جهت برآورد متامدل) به ترتیب جهت ساخت متامدل چند جمله‌ای بر اساس مدل شبیه‌سازی کامپیوتری استفاده می‌شود. یک متامدل درجه دوم کامل که قابلیت در نظر گیری اثرات اصلی، متعامد و درجه دو را دارا است به صورت معادله (۶) نوشته می‌شود.

-
- 1- Lavoie
 - 2- Kleijnen
 - 3- Sargent
 - 4- Montgomery

$$y = \beta_0 + \sum_{i=1}^n \beta_i x_i + \sum_{i=1}^n \beta_{ii} x_i^2 + \sum_{i < j} \beta_{ij} x_i x_j + e \quad (6)$$

تابع هدف دوم مدل ریاضی

متامدل به دست آمده توسط روش پیشنهادی بخش ۱, ۱, ۲, نرخ تولید را برای ترکیبات مختلف تخمین می‌زند. هر چقدر تعداد ماشین‌های موازی در ایستگاه‌های کاری بیشتر باشد، می‌توان انتظار نرخ تولیدی بالاتر داشت، اما به قیمت افزایش هزینه‌های سیستم. بنابراین مسئله طراحی بهینه خط تولید می‌تواند به این صورت مطرح شود: پیدا کردن ترکیبی از ماشین‌ها که نرخ تولید را حداکثر کرده و هزینه‌ها را حداقل کند، بنابراین داریم:

(۷)

$$\begin{aligned} \text{Minimize } Z_2 = & \sum_{i=1}^m \sum_{j=1}^n c_{ij} (x_{ij} - l_{ij}) + \sum_{i=1}^m \sum_{j=1}^n a_{ij} (x_{ij} - l_{ij}) + \sum_{i=1}^m k_i y_i \\ & + \sum_{i=1}^m \sum_{j=1}^n h_{ij} x_{ij} + \sum_{i=1}^m \sum_{j=1}^n o_{ij} x_{ij} \end{aligned}$$

رابطه (۷) تابع هدف دوم مدل است که هزینه‌های کل را حداقل می‌کند. تابع هدف دوم

مدل از پنج بخش تشکیل شده است. بخش اول، $\sum_{i=1}^m \sum_{j=1}^n c_{ij} (x_{ij} - l_{ij})$ هزینه خرید ماشین‌آلات جدید را محاسبه می‌کند به طوری که x_{ij} نشان‌دهنده تعداد ماشین نوع j است که باید در ایستگاه i وجود داشته باشد، c_{ij} نشان‌دهنده قیمت خرید یک واحد x_{ij} است، l_{ij} نشان‌دهنده تعداد ماشین موجود نوع j در ایستگاه i است. از آنجایی که x_{ij} تعداد ماشین نوع j که باید در ایستگاه i وجود داشته باشد (شامل ماشین‌های قدیمی و جدید) و l_{ij} تعداد ماشین موجود (قدیمی) نوع j در ایستگاه i تعریف شده است، بنابراین تعداد ماشین‌های جدید در ایستگاه i توسط $(x_{ij} - l_{ij})$ محاسبه می‌شود که با ضرب این عبارت در c_{ij} ، هزینه خرید محاسبه می‌شود.

افزودن ماشین‌های موازی جدید نوع i به ایستگاه i ، هزینه نصب و راه‌اندازی (a_{ij}) و همچنین هزینه‌های ثابت (k_i) به سیستم تحمیل می‌کند. هر چقدر تعداد ماشین‌های موازی جدید در ایستگاه i بیشتر باشد، هزینه‌های نصب و راه‌اندازی بیشتر خواهد بود. از طرف دیگر،

افزودن ماشین‌های موازی به ایستگاه i ، هزینه‌های ثابتی را در بردارد که مرتبط با تعداد ماشین‌های اضافه شده نیست (مانند هزینه توقف تولید و غیره)، به عبارت دیگر در اثر اضافه شدن ماشین‌های موازی جدید به ایستگاه i ، این ایستگاه هزینه ثابتی را متحمل می‌شود، خواه تعداد ماشین‌های موازی جدید یک باشد خواه بیش از یک.

از آنجایی که اضافه کردن ماشین‌های جدید به ایستگاه، هزینه‌های نصب و راه‌اندازی را به سیستم تحمیل می‌کند، بخش دوم، $\sum_{i=1}^m \sum_{j=1}^n a_{ij} (x_{ij} - l_{ij})$ ، هزینه کل نصب و راه‌اندازی ماشین‌های جدید را محاسبه می‌کند. از آنجایی که x_{ij} تعداد ماشین نوع j که باید در ایستگاه i وجود داشته باشد (شامل ماشین‌های قدیمی و جدید) و l_{ij} تعداد ماشین موجود (قدیمی) نوع j در ایستگاه i تعریف شده است، بنابراین تعداد ماشین‌های جدید در ایستگاه i توسط $(x_{ij} - l_{ij})$ محاسبه شده که با ضرب این عبارت در a_{ij} ، هزینه نصب و راه‌اندازی ماشین‌ها محاسبه می‌شود.

در بخش سوم رابطه (۳)، $\sum_{i=1}^m k_i y_i$ هزینه ثابت افزودن ماشین‌های جدید محاسبه شده است. این هزینه توسط متغیر y_i (که یک متغیر ۰ و ۱ است) لحاظ شده است به طوری که این متغیر (y_i) برابر با ۱ است اگر تعداد ماشین‌های جدید در ایستگاه i بزرگتر یا مساوی ۱ باشد و برابر صفر است اگر تعداد ماشین‌های جدید در ایستگاه i برابر صفر باشد.

بخش چهارم، $\sum_{i=1}^m \sum_{j=1}^n h_{ij} x_{ij}$ هزینه‌های نیروی انسانی لازم برای کار با ماشین‌آلات را محاسبه می‌کند به طوری که h_{ij} نشان‌دهنده هزینه اپراتوری کار با ماشین نوع j در ایستگاه i در واحد زمان است. از آنجایی که هر دوی ماشین‌های قدیمی و جدید نیازمند اپراتور هستند، از این رو با ضرب h_{ij} در x_{ij} هزینه‌های اپراتوری برای ماشین‌های قدیمی و جدید محاسبه می‌شود.

بخش پنجم، $\sum_{i=1}^m \sum_{j=1}^n o_{ij} x_{ij}$ هزینه‌های عملیاتی و نگهداری و تعمیر را برای ماشین‌آلات محاسبه می‌کند به طوری که o_{ij} نشان‌دهنده هزینه عملیاتی و نگهداری و تعمیر ماشین i در ایستگاه j در واحد زمانی است. از آنجایی که هزینه‌های عملیاتی و نگهداری

و تعمیر برای هر دوی ماشین‌های قدیمی و جدید باید لحاظ شود، از این رو با ضرب O_{ij} در x_{ij} هزینه‌های عملیاتی و نگهداری و تعمیر برای ماشین‌های قدیمی و جدید محاسبه می‌شود.

تابع هدف سوم مدل ریاضی

رابطه (۸) یک متامدل برآوردگر است که نرخ تولید محصولات نامنطبق را حداقل می‌کند و این امر منجر به حداکثر شدن کیفیت محصولات نهایی می‌شود.

$$\text{Minimize } Z_3 = g(x_{ij}) \quad (8)$$

با توجه به اینکه ترکیب ماشین‌آلات در خط تولید بر روی نرخ تولید محصولات نامنطبق اثر می‌گذارد، می‌توان نرخ تولید محصولات نامنطبق را تابعی از تعداد و نوع ماشین‌های نوع i که باید در ایستگاه i وجود داشته باشد در نظر گرفت.

محدودیت‌های مدل ریاضی

محدودیت (۹) تضمین می‌کند که فضای اشغالی ناشی از کل ماشین‌آلات خط تولید از فضای موجود S فراتر نرود، به طوری که S_{ij} نشان‌دهنده فضای مورد نیاز ماشین نوع i در ایستگاه i است.

$$\sum_{i=1}^m \sum_{j=1}^n s_{ij} x_{ij} \leq S \quad (9)$$

محدودیت (۱۰) تضمین می‌کند که هزینه خرید ماشین‌آلات جدید از بودجه خرید ماشین‌آلات جدید P فراتر نرود.

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} (x_{ij} - l_{ij}) \leq P \quad (10)$$

محدودیت (۱۱) تضمین می‌کند که هزینه نیروی انسانی لازم برای ماشین‌آلات (قدیمی و جدید) از بودجه در نظر گرفته شده H برای نیروی انسانی فراتر نرود.

$$\sum_{i=1}^m \sum_{j=1}^n h_{ij} x_{ij} \leq H \quad (11)$$

محدودیت (۱۲) تضمین می‌کند که هزینه‌های عملیاتی و نگهداری تعمیر ماشین‌ها از بودجه در نظر گرفته شده O فراتر نرود.

$$\sum_{i=1}^m \sum_{j=1}^n o_{ij} x_{ij} \leq O \quad (12)$$

محدودیت (۱۳) تضمین می‌کند که کل هزینه‌های سیستم که شامل هزینه خرید ماشین‌آلات جدید، هزینه نصب و راه‌اندازی ماشین‌آلات، هزینه‌های اپراتوری، و هزینه‌های عملیاتی و نگهداری و تعمیر ماشین‌آلات می‌شود، از بودجه کل تعیین شده C فراتر نرود.

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} (x_{ij} - l_{ij}) + \sum_{i=1}^m \sum_{j=1}^n a_{ij} (x_{ij} - l_{ij}) + \sum_{i=1}^m \sum_{j=1}^n h_{ij} x_{ij} + \sum_{i=1}^m \sum_{j=1}^n o_{ij} x_{ij} \leq C$$

در مواقعی که تصمیم‌گیرنده در نظر داشته باشد که ترکیبات به دست آمده منجر به نرخ تولیدی بالاتر از میزان تعیین شده باشد، محدودیت (۱۴) تضمین می‌کند که نرخ تولید حداقل به اندازه l_f باشد.

$$f(x_{ij}) \geq l_f \quad (14)$$

محدودیت (۱۵) تضمین می‌کند که نرخ تولید محصولات نامنطبق ترکیب به دست آمده حداکثر به میزان u_g است.

$$g(x_{ij}) \leq u_g \quad (15)$$

محدودیت (۱۶) تضمین می‌کند که اگر تعداد ماشین‌آلات خریداری شده در ایستگاه i بزرگتر یا مساوی یک باشد، هزینه‌های ثابت لحاظ شود به طوری که y_i یک متغیر ۰-۱ و M یک عدد بسیار بزرگ در نظر گرفته شده است.

$$\sum_{j=1}^n (x_{ij} - l_{ij}) \leq M y_i \quad \forall i = 1, 2, \dots, m \quad (16)$$

محدودیت (۱۷) تضمین می‌کند که تعداد ماشین نوع z در ایستگاه i حداقل به اندازه l_{ij} و حداکثر به اندازه u_{ij} باشد.

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall i = 1, 2, \dots, m, \quad \forall j = 1, 2, \dots, n \quad (17)$$

محدودیت (۱۸) تضمین می‌کند که تعداد ماشین‌های نوع j در ایستگاه i بزرگتر یا مساوی صفر و عدد صحیح باشد.

$$x_{ij} \geq 0 \& Integer \quad \forall i = 1, 2, \dots, m, \quad \forall j = 1, 2, \dots, n \quad (18)$$

روش پیشنهادی حل مسئله

مدل ارائه شده در بخش ۲، یک مسئله چندهدفه است که تابع هدف اول و سوم و همچنین محدودیت‌های (۱۴) و (۱۵) غیرخطی هستند، همچنین برخی متغیرهای مدل در تابع هدف و محدودیت‌ها به صورت ۰-۱ هستند. بنابراین، الگوریتم‌های تکاملی برای این مسئله جهت به دست آوردن جواب‌های بهینه/نزدیک بهینه می‌تواند مناسب باشد.

الگوریتم‌های تکاملی چندهدفه

الگوریتم‌های تکاملی زیرمجموعه‌ای از محاسبات تکاملی هستند که قابلیت دست‌یابی به جواب‌های بهینه/نزدیک بهینه را در مسائل خطی/غیرخطی، پیوسته/گسسته، محدب/غیر محدب با استفاده از غالباً تئوری‌های بیولوژیکی دارا هستند (دب، ۲۰۰۱؛ تان و همکاران، ۲۰۰۱). در این مقاله، دو نوع از الگوریتم‌های فراابتکاری چندهدفه الگوریتم ژنتیک با مرتب سازی نامغلوب و بهینه‌سازی ازدحام ذرات چندهدفه به عنوان ابزارهای بهینه‌سازی جهت حل مسئله *RAP* ارائه شده است.

الگوریتم ژنتیک با مرتب‌سازی نامغلوب

Deb الگوریتم ژنتیک با مرتب‌سازی نامغلوب (*NSGA-II*) را ارائه کرد (دب، ۲۰۰۲). *Deb* و *Agrawal* یک عملگر تقاطع به نام عملگر *SBX* جهت ترکیب دو کروموزوم

1- Deb

2- Tan

3- Non-Dominated Sorting Genetic Algorithm - NSGA-II

4- Multiple Objective Particle Swarm Optimization – MOPSO

(کروموزوم‌های والد) و ایجاد دو کروموزوم جدید (کروموزوم‌های فرزند) ارائه کردند که این روش در این مقاله به‌عنوان عمل گر تقاطع استفاده شده است (دب و اگراوال، ۱۹۹۵). این عمل گر تا حدی شبیه به تقاطع تک نقطه‌ای است. تابع توزیع احتمال در این روش به‌صورت زیر است:

$$P(\beta_i) = \begin{cases} 0.5(\eta_c + 1)\beta_i^{\eta_c}, & \text{if } \beta_i \leq 1 \\ 0.5(\eta_c + 1)\frac{1}{\beta_i^{\eta_c+2}}, & \text{otherwise} \end{cases} \quad (19)$$

$$\beta_i = \begin{cases} \left(2u_i\right)^{\frac{1}{\eta_c+1}}, & \text{if } u_i \leq 0.5 \\ \left(\frac{1}{2(1-u_i)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases} \quad (20)$$

که در آن: $P(\beta_i)$ = احتمال تقاطع، β_i = تفاوت بین توابع هدف والدها و فرزندان، η_c = یک عدد ثابت که نشان‌دهنده تفاوت بین توابع هدف والد و فرزندان است (مقادیر بالاتر η_c احتمال ساختن جواب‌های نزدیک جواب‌های والد را بیشتر می‌کند)، u_i = یک عدد تصادفی بین $[0, 1]$. تفاوت ذکر شده بین والدین و فرزندان توسط رابطه (۲۱) محاسبه می‌شود و مقادیر فرزندان توسط رابطه (۲۲) محاسبه می‌شود.

$$\beta_i = \left| \frac{x_1^{child} - x_2^{child}}{x_1^{parent} - x_2^{parent}} \right| \quad (21)$$

$$\begin{aligned} x_1^{child} &= 0.5 \left[(1 + \beta_i)x_1^{parent} + (1 - \beta_i)x_2^{parent} \right] \\ x_2^{child} &= 0.5 \left[(1 - \beta_i)x_1^{parent} + (1 + \beta_i)x_2^{parent} \right] \end{aligned} \quad (22)$$

به طوری که x_1^{child} و x_2^{child} = ارزش کروموزوم‌های فرزند اول و دوم و همچنین x_1^{parent} و x_2^{parent} = ارزش کروموزوم‌های والد اول و دوم است. عملگر دیگر GA جهش است. یک عملگر چند جمله‌ای جهش که توسط Deb و $Goyal$ ارائه شده است (دب و گوپال، ۱۹۹۶)، در این مقاله مورد استفاده قرار گرفته است که به صورت زیر است:

$$\delta_i = \begin{cases} (2r_i)^{1/(\eta_m+1)} - 1, & \text{if } r_i < 0.5 \\ 1 - [2(1-r_i)^{1/(\eta_m+1)}], & \text{if } r_i \geq 0.5 \end{cases} \quad (23)$$

به طوری که δ_i = ارزش جهش، r_i = یک عدد تصادفی بین $[0, 1]$ و η_m = ثابت توزیع جهش است. پارامتر δ به صورت رابطه (۲۴) به ارزش ژن‌های والد اضافه می‌شود.

$$x^{child} = x^{parent} + \delta \quad (24)$$

الگوریتم بهینه‌سازی ازدحام ذرات چندهدفه

الگوریتم PSO یک روش بهینه‌سازی مبتنی بر الگوی مهاجرت پرندگان است. در دنیای واقعی، حرکت پرندگان و جستجوی غذا یک سیستم منظم را نشان می‌دهد که در آن هر پرنده موقعیت خود را در بعد زمان بهبود می‌بخشد. بنابراین، در یک مسئله بهینه‌سازی، موقعیت ذره i ام (x_i) می‌تواند توسط یک بردار D بعدی نشان داده شود:

$$x_i = [x_{i1}, x_{i2}, \dots, x_{iD}] \text{ for } i = 1, \dots, N \quad (25)$$

به طوری که D = تعداد متغیرهای تصمیم‌گیری و N = اندازه دسته. به علاوه، بهترین پرنده (با کمترین فاصله از غذا) $Gbest$ نامیده شده و بهترین موقعیتی که یک پرنده در جهت پیدا کردن غذا پیدا کرده، $Pbest$ نامیده می‌شود. گام‌های الگوریتم PSO به صورت زیر است:

در گام اول، جواب‌های تصادفی (x_i) با تابع توزیع نرمال متغیرهای تصمیم‌گیری تولید می‌شود. در مرحله دوم، مقدار تابع هدف برای هر ذره محاسبه شده و در حافظه الگوریتم باقی می‌ماند. در مرحله بعد $Pbest$ و $Gbest$ با توجه به بهترین موقعیت ذرات و دسته که

تاکنون یافت شده است تخصیص داده می‌شود. در مرحله چهارم، شتاب هر ذره با استفاده از رابطه زیر محاسبه می‌شود:

(۲۶)

$$v_{id}^{t+1} = \alpha \left(w^t v_{id}^t + c_1 r_1^t (Pbest_{id} - x_{id}^t) + c_2 r_2^t (Gbest_d - x_{id}^t) \right)$$

for $i = 1, 2, \dots, N$ $d = 1, 2, \dots, D$

$$w^t = w_{max} - \frac{(w_{max} - w_{min}) \times t}{iter_{max}} \quad (۲۷)$$

که در آن v_{id}^{t+1} = شتاب ذره i ام برای بعد d ام در تکرار $t+1$ ام؛ α = فاکتور انقباض که یک ضریب ثابت از پیش تعیین شده است و شتاب ذرات در فضای متغیرهای تصمیم را کنترل می‌کند؛ w^t = پارامتر وزن اینرسی در تکرار t ام که توسط رابطه (۲۷) محاسبه می‌شود. این پارامتر از بیشترین مقدار (w_{max}) در تکرار اول شروع کرده و تا کمترین مقدار در تکرار آخر ($iter_{max}$) پیش می‌رود. در واقع، در ابتدای فرآیند، اثر شتاب بیش از تکرارهای بعدی است. $c1$ = پارامتر فردی و $c2$ = پارامتر جمعی بوده که سهم $Pbest$ و $Gbest$ را مشخص می‌کند و r_1^t و r_2^t اعداد تصادفی بین $[0, 1]$ در تکرار t ام است. بنابراین، هر ذره در فضای تصمیم با بردار شتاب حرکت می‌کند. $Pbest_{id}$ عبارت است از بهترین موقعیت ذره i ام برای بعد d ام که تاکنون یافت شده و $Gbest_d$ عبارت است از بهترین موقعیت دسته برای بعد d ام که تاکنون یافت شده است. در مرحله بعد، شتاب‌های به‌دست آمده توسط حدود پایین (v_{min}) و بالای (v_{max}) شتاب کنترل می‌شوند:

$$v_{min} \leq v_{id}^{t+1} \leq v_{max} \quad (۲۸)$$

در نهایت موقعیت ذره توسط رابطه (۲۹) محاسبه می‌شود:

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (۲۹)$$

موقعیت جدید ذره‌ها به‌عنوان ورودی‌های جدید مدل شبیه‌سازی حرکت ذره‌ها در تکرار دوم استفاده شده و تابع هدف مجدداً محاسبه می‌شود. این فرآیند تا ماکزیمم تعداد تکرارها ادامه می‌یابد.

حرکت به سمت نقطه بهینه در PSO تک هدفه ($SOPSO$) متفاوت از PSO چندهدفه ($MOPSO$) است. در الگوریتم $SOPSO$ ، هر ذره تابع هدف خود را دنبال می کند. در الگوریتم $MOPSO$ ، تعداد توابع هدف بیش از یک است. بنابراین، الگوریتم باید با حرکت در جهت هر تابع هدف تطبیق پیدا کند. الگوریتم PSO بردار ارزیابی شده ($VEPSO$) یک الگوریتم چندهدفه است که برای هر تابع هدف از یک دسته استفاده می کند. بنابراین، هر ذره در هر دسته $Pbest$ خود را داشته، اما $Gbest$ هر دسته توسط $Gbest$ دسته های دیگر برای تکرارهای بعدی جایگزین می شود (پاراسوپولوس و وراهاتیس، ۲۰۰۲).

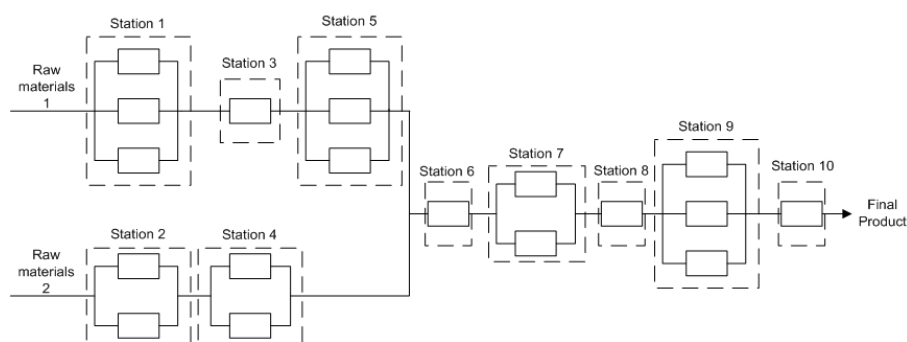
در روش پیشنهادی این مقاله، یک آرشیو پویا برای تبادل ذرات تولید شده در هر تکرار ساخته می شود. بنابراین، ذرات با یکدیگر مقایسه شده و جواب های غیرمغلوب نگهداری شده و بقیه جواب ها در پایان تکرار حذف می شوند. با این مکانیزم، در هر تکرار می توان به اعضای آرشیو خارجی دسترسی داشت و همچنین سائز آرشیو خارجی به صورت پویا تغییر می کند.

مثال عددی

جهت نشان دادن عملکرد مدل ریاضی پیشنهادی و الگوریتم حل پیشنهادی، شکل (۱) را در نظر بگیرید. خط تولید نشان داده شده شامل ۱۰ ایستگاه کاری بوده که هر ایستگاه عملیات متفاوتی را انجام داده و تعداد ماشین متفاوتی دارد به طوری که زمان های پردازش، زمان بین خرابی ها و زمان های تعمیر از توابع توزیع عمومی پیروی می کند.

پژوهشگاه علوم انسانی و مطالعات فرهنگی
پرتال جامع علوم انسانی

-
- 1- Single-objective particle swarm optimization
 - 2- Vector evaluated PSO
 - 3- Parasopoulos
 - 4- Vrahatis



شکل ۱. خط تولید با ۱۰ ایستگاه کاری

با در نظر گیری توضیحات فوق، مسئله مورد نظر تعیین بهینه/نزدیک بهینه نوع و تعداد ماشین های جدیدی است که باید به هر ایستگاه اضافه شود. بنابراین در جدول (۱) مشخصات عملکردی ماشین های جدید که می توان به ایستگاه ها اضافه کرد برای هر ایستگاه به تفکیک نشان داده شده است. همچنان که مشاهده می شود برای ایستگاه های ۱ و ۲ و ۳ و ۵ و ۶ و ۱۰ دو نوع ماشین وجود دارد؛ برای ایستگاه های ۷ و ۹ یک نوع ماشین و برای ایستگاه های ۴ و ۸ سه نوع ماشین وجود دارد.

جدول ۱ - مشخصات عملکردی ماشین های جدید برای ایستگاه های ۱ تا ۱۰

ایستگاه	ماشین های جدید	زمان پردازش	زمان بین خرابی ها	زمان تعمیر
1	۱	$G(18,7,3)$	$W(12000,70,20)$	$G(1100,250,25)$
	۲	$G(19,6,4)$	$W(12500,60,25)$	$G(1250,300,30)$
2	۱	$W(24,8,6)$	$W(12500,220,17)$	$G(1200,220,56)$
	۲	$W(21,7,5)$	$W(10500,230,17)$	$G(1200,250,56)$
3	۱	$W(40,8,3)$	$G(20000,180,35)$	$G(400,20,50)$
	۲	$W(35,7,4)$	$G(17000,230,56)$	$G(430,25,42)$
4	۱	$G(25,9,11)$	$G(19560,230,45)$	$E(2100)$
	۲	$G(20,10,6)$	$G(17500,100,22)$	$G(1100,20,15)$
	۳	$W(23,14,10)$	$E(15000)$	$E(1700)$

ادامه جدول ۱ - مشخصات عملکردی ماشین‌های جدید برای ایستگاه‌های ۱ تا ۱۰

ایستگاه	ماشین های جدید	زمان پردازش	زمان بین خرابی‌ها	زمان تعمیر
5	۱	$G(28,14,9)$	$W(16000,100,55)$	$G(1400,200,70)$
	۲	$G(25,12,8)$	$W(12400,150,33)$	$G(1500,220,60)$
6	۱	$E(40)$	$G(17500,80,40)$	$W(1200,180,50)$
	۲	$Normal(35,3)$	$W(14000,75,30)$	$G(1100,130,60)$
7	۱	$W(45,17,6)$	$G(12500,350,100)$	$Normal(1200,50)$
8	۱	$Normal(150,7)$	$W(8000,250,30)$	$G(1700,200,30)$
	۲	$E(155)$	$W(7900,300,20)$	$W(1900,100,40)$
	۳	$Normal(145,10)$	$G(7400,280,18)$	$W(1500,150,60)$
9	۱	$W(50,20,14)$	$W(8000,400,130)$	$G(1500,120,40)$
10	۱	$G(60,20,12)$	$G(7000,340,150)$	$W(1400,100,30)$
	۲	$W(55,24,16)$	$G(8500,220,100)$	$G(1300,120,40)$

G: Gamma, W: Weibull, E: Exponential

بر طبق رویکرد ارائه شده در بخش ۱، ۲، ۱، به منظور دست یابی به متامدلی که نرخ تولید را برای هر ترکیب تخمین بزنند، در گام اول، شبکه صف شکل (۱) شبیه سازی شده است. در گام بعد، با استفاده از طرح عاملی کسری دوسطحی²⁰⁻¹² و طرح سه سطحی *Box-Bhenken* متامدل زیر جهت تخمین نرخ تولید برآورد شده است.

(۳۰)

$$\begin{aligned}
 \text{Production rate} = & -889.109 + 3.594x_1 + 18.375x_2 + 15.927x_3 + 7.552x_4 + 66.698x_5 + 6.604x_6 - 1.875x_7 + 767.583x_8 - \\
 & 26.083x_9 + 497.625x_{10} - 0.359x_{12} - 2.297x_{22} - 4.141x_{32} - \\
 & 0.729x_{42} - 9.188x_{52} - 0.924x_{62} + 0.188x_{72} - 123.465x_{82} + \\
 & 2.174x_{92} - 36.609x_{102} - 0.271x_3x_4 + 0.719x_3x_5 + 0.375x_3x_6 + \\
 & 0.208x_3x_8 + 3.25x_3x_{10} - 0.083x_4x_5 + 0.354x_4x_{10} + \\
 & 2.531x_5x_{10} + 0.139x_6x_8 + 0.458x_6x_{10} + 202.583x_8x_{10}
 \end{aligned}$$

رابطه (۳۰) نرخ تولید را براساس تعداد ماشین‌هایی که باید در هر ایستگاه قرار گیرند تخمین می‌زند. لازم به توضیح است که در این رابطه، اثراتی آورده شده است که به لحاظ آماری معنی‌دار بوده و مقدار p -value برای آن‌ها کمتر از ۰,۰۵ بوده است. به‌عنوان مثال اثر تعاملی متغیر ۸ و ۱۰ به جهت اینکه به لحاظ آماری معنی‌دار بوده در رابطه (۳۰) قرار گرفته است، اما اثر تعاملی متغیر ۴ و ۷ به جهت اینکه مقدار p -value آن بیشتر از ۰,۰۵ است در رابطه (۳۰) قرار نگرفته است. توضیح دیگر اینکه آزمون تحلیل واریانس متامدل برازش شده (۳۰) مقدار آماره آزمون فیشر را برابر $F=56.19$ ، مقدار p -value برای مدل رگرسیونی را برابر ۰,۰۰۰، مقدار p -value برای اثرات خطی را برابر ۰,۰۴۹، مقدار p -value برای اثرات مربع را برابر ۰,۰۰۰، مقدار p -value برای اثرات تعاملی را برابر ۰,۰۰۰ و مقدار R -square را برابر ۹۷,۲۳ درصد نشان می‌دهد. همچنین آزمون نرمال بودن متغیر پاسخ و باقی مانده‌های مدل رگرسیونی نیز با استفاده از آزمون کالموگروف - اسمیرنوف انجام شده که نتایج حاکی از نرمال بودن این متغیرهاست. بنابر توضیحات فوق می‌توان نتیجه گرفت که متامدل برازش شده از کیفیت لازم جهت تخمین نرخ تولید برخوردار است. به‌علاوه جهت مقایسه عملکرد متامدل برازش شده با مدل شبیه‌سازی، از رویکرد ارائه شده در (دوریکس و پیروال، ۲۰۰۳) استفاده شده است به این صورت که ۱۰ ترکیب مختلف از مقادیر ورودی به صورت تصادفی انتخاب شده و خروجی مدل شبیه‌سازی با خروجی حاصل از متامدل مقایسه شده و میزان میانگین قدرمطلق خطا برای آن محاسبه شده است. با توجه به اینکه در این مثال این شاخص برابر ۲,۴۶ درصد محاسبه شده است و این مقدار کمتر از ۳ درصد است، صحت متامدل تأیید می‌شود.

پس از فرموله کردن تابع هدف اول مدل، توابع هدف دیگر مدل و محدودیت‌ها باید فرموله شوند. براساس مدل ریاضی ارائه شده در بخش ۲، تابع هدف دوم مدل ریاضی که هزینه‌ها را حداقل می‌کند با استفاده از ستون‌های هزینه جدول ۲ فرموله می‌شود.

جدول ۲ - آیت‌های هزینه‌ای و فضای موردنیاز ماشین‌های جدید

ایستگاه	نوع ماشین	قیمت خرید (واحد پولی)	هزینه نصب و راه اندازی (واحد پولی)	هزینه نیروی انسانی (واحد پولی)	هزینه عملیاتی (واحد پولی)	هزینه ثابت راه‌اندازی ماشین‌های جدید (واحد پولی)	فضای موردنیاز (متر مربع)
۱	۲	3500	300	780	1250	80	1.8
۲	۲	6200	650	1690	2254	150	2.5
۳	۱	11000	800	2080	3960	220	3.3
۴	۲	4300	380	988	1548	100	2.1
۵	۱	1200	150	390	432	40	1.1
۶	۱	7000	650	1690	2520	190	2.6
۷	۱	5500	400	1040	1980	120	2.3
۸	۳	40000	2800	4280	7654	820	3.9
۹	۱	6000	550	1430	2160	140	2.4
۱۰	۱	65000	3100	5060	12563	750	4.5

تابع هدف سوم مدل ریاضی پیشنهادی، تابع حداقل‌کننده نرخ تولید محصولات نامنطبق است. براساس رویکرد ارائه‌شده در بخش ۱، ۲، ۱ متامدل زیر جهت برآورد عدم تطابق محصولات برآورد شده‌است:

(۳۱)

$$\begin{aligned}
 \text{Disconformity} = & 0.2699 + 0.0051x_1 + 0.0027x_2 - 0.0069x_3 - \\
 & 0.003x_4 - 0.0046x_5 + 0.0014x_6 - 0.0023x_7 - 0.0226x_8 + 0.0091x_9 - \\
 & 0.0703x_{10} - 0.0005x_{12} - 0.0003x_{22} - 0.0011x_{32} + 0.0003x_{42} - \\
 & 0.001x_{52} - 0.0003x_{62} + 0.0003x_{72} + 0.0023x_{82} - 0.0006x_{92} + \\
 & 0.0087x_{102} - 0.0002x_1x_2 + 0.0003x_1x_3 + 0.0002x_1x_4 + \\
 & 0.0004x_1x_5 - 0.0002x_1x_6 - 0.0002x_1x_9 - 0.0002x_1x_{10} + \\
 & 0.0003x_2x_3 - 0.0003x_2x_4 + 0.0003x_2x_5 + 0.0012x_3x_5 + \\
 & 0.0009x_3x_6 + 0.0003x_3x_7 + 0.001x_3x_8 - 0.0002x_3x_9 + \\
 & 0.0001x_3x_{10} - 0.0004x_4x_5 + 0.0002x_4x_7 + 0.0002x_4x_{10} - \\
 & 0.0006x_5x_7 + 0.0005x_5x_8 + 0.0006x_5x_9 + 0.0009x_5x_{10} + \\
 & 0.0003x_6x_8 - 0.0002x_6x_9 - 0.0003x_8x_9 - 0.0025x_8x_{10}
 \end{aligned}$$

جهت در نظر گیری محدودیت‌های مدل براساس مدل ریاضی ارائه شده در بخش ۲، محدودیت‌های زیر در نظر گرفته شده است: حد پایین ماشین‌ها در ایستگاه‌های ۱ تا ۱۰ به ترتیب ۳ و ۲ و ۲ و ۳ و ۱ و ۲ و ۳ و ۱ و ۳ و ۱ (تعداد ماشین موجود در هر ایستگاه) و حد بالای ماشین‌ها به ترتیب ۷ و ۶ و ۵ و ۸ و ۵ و ۷ و ۸ و ۷ و ۹ و ۵ (با توجه به فضای موجود) در نظر گرفته شده است؛ کل فضای موجود جهت قرار گیری ماشین‌ها ۱۴۰ مترمربع در نظر گرفته شده است؛ بودجه خرید ماشین‌ها ۶۵۰۰۰۰ واحد پولی در نظر گرفته شده است؛ بودجه نیروی انسانی ۱۰۰۰۰۰ واحد پولی در نظر گرفته شده است؛ بودجه هزینه‌های عملیاتی ۱۸۰۰۰۰ واحد پولی در نظر گرفته شده است؛ بودجه کل ۹۰۰۰۰۰ واحد پولی در نظر گرفته شده است؛ حداقل نرخ تولید ۱۰۰۰ واحد در نظر گرفته شده است.

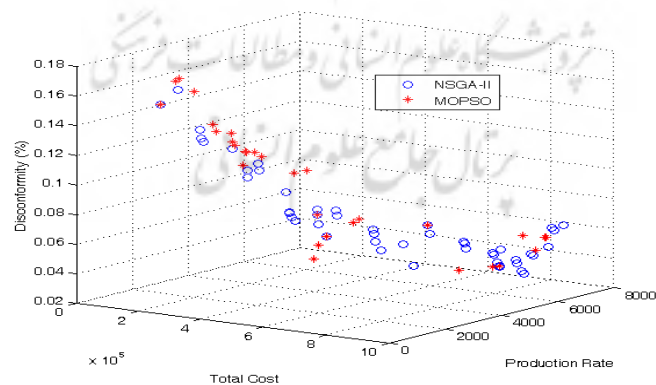
با در نظر گیری توابع هدف و محدودیت‌های مدل، بر طبق روش حل پیشنهادی بخش ۳، مدل ریاضی با استفاده از الگوریتم‌های پیشنهادی *NSGA-II* و *MOPSO* حل شده است. پارامترهای الگوریتم *NSGA-II* و *MOPSO* در این مطالعه به صورت تجربی/آزمایشی تعیین شده است. براین اساس برای الگوریتم *NSGA-II* اندازه جمعیت ۱۰۰ (حد بالای آزمایش شده)، نرخ جهش ۰٫۰۶ و تعداد تکرارها ۴۰۰ به دست آمده است (۴۰۰ حد بالای آزمایش شده بوده است)، به علاوه نرخ تقاطع در هر بار استفاده از طریق رابطه (۱۹) محاسبه شده است. همچنین برای الگوریتم *MOPSO* اندازه جمعیت ذرات برابر ۱۰۰ (حد بالای آزمایش شده)، فاکتور انقباض برابر ۰٫۷، پارامتر فردی برابر ۱٫۵، پارامتر جمعی برابر ۲، w_{max} برابر ۰٫۹، w_{min} برابر ۰٫۱ و تعداد تکرارها برابر ۳۰۰ بدست آمده است. تعداد جواب‌های نهایی غیرمسلط یافت شده توسط *NSGA-II*، ۵۰ عدد و تعداد جواب‌های نهایی غیرمسلط یافت شده توسط *MOPSO*، ۳۱ عدد بوده است. جدول ۳، هشت جواب از کل هشتاد و یک جواب به دست آمده را نشان می‌دهد. هر یک از ستون‌های S_1 تا S_8 نشان دهنده یکی از جواب‌های مسئله است. به عنوان نمونه، ستون S_5 نشان می‌دهد که اگر تعداد ماشین‌های ایستگاه‌های ۱ تا ۱۰ به ترتیب ۳، ۲، ۱، ۲، ۱، ۵، ۲، ۴، ۳ و ۴ باشد، نرخ تولید ۵۰۳۱ بوده که از این تعداد، ۲۵۶ قطعه ($۵۰۳۱ * ۰٫۰۵۱ = ۲۵۶$) با استانداردهای کیفی تطابق ندارد. با توجه به تعداد

ماشین‌های موجود فعلی در هر ایستگاه، جواب فوق عبارت خواهد بود از عدم خرید ماشین جدید برای ایستگاه‌های اول، دوم، سوم و چهارم، خرید دو ماشین جدید نوع ۱ برای ایستگاه پنجم، عدم خرید ماشین جدید برای ایستگاه‌های ششم و هفتم، خرید سه ماشین جدید نوع ۳ برای ایستگاه هشتم، عدم خرید ماشین جدید برای ایستگاه نهم، خرید سه ماشین جدید نوع ۱ برای ایستگاه دهم. شکل ۲ جواب‌های به دست آمده توسط دو الگوریتم *NSGA-II* و *MOPSO* را به صورت توأم در فضای سه بعدی نشان می‌دهد.

جدول ۳- برخی از جواب‌های یافت شده توسط *NSGA-II* و *MOPSO*

متغیر	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>S4</i>	<i>S5</i>	<i>S6</i>	<i>S7</i>	<i>S8</i>
$x1$	3	3	3	3	3	3	3	3
$x2$	3	2	2	2	2	2	2	2
$x3$	4	1	1	1	1	1	1	3
$x4$	4	2	2	2	2	2	3	4
$x5$	4	3	5	5	5	5	5	4
$x6$	4	1	1	1	1	1	1	4
$x7$	2	2	2	2	2	2	3	2
$x8$	7	6	5	4	4	4	3	3
$x9$	3	3	3	3	3	3	3	3
$x10$	5	5	5	5	4	3	3	2
<i>O1</i>	7378	7106	6696	6026	5031	3962	3454	2559
<i>O2</i>	868197	696265	645915	591181	505458	419735	381357	363048
<i>O3</i>	0.0580	0.0393	0.0428	0.0547	0.0518	0.0663	0.0723	0.1169

هدف نام: *Oj*



شکل ۲- نمودار تلفیقی نرخ تولید - هزینه کل - درصد عدم تطابق (*MOPSO* و *NSGA-II*)

جهت مقایسه کیفیت جواب‌های به‌دست‌آمده دو الگوریتم *NSGA-II* و *MOPSO*، سه معیار ارزیابی در نظر گرفته شده است که عبارتند از تعداد جواب‌های پارتوی به‌دست‌آمده، فاصله پارتوی به‌دست‌آمده از پارتوی بهینه و شاخص کیفیت.

تعداد جواب‌های به‌دست‌آمده پارتو (NPS)^۱ - این پارامتر تعداد جواب‌های بهینه پارتوی به‌دست‌آمده را نشان می‌دهد. تعداد جواب‌های بهینه پارتوی به‌دست‌آمده در این مسئله توسط *NSGA-II* برابر با ۵۰ و تعداد جواب‌های بهینه پارتوی به‌دست‌آمده در این مسئله توسط *MOPSO* برابر با ۳۱ است. بنابراین براساس شاخص *NPS*، پارتوی به‌دست‌آمده توسط *NSGA-II* مناسب‌تر از الگوریتم *MOPSO* است.

فاصله نسل (GD)^۲ - این معیار قادر است متوسط فاصله یک سری جواب‌های پارتو را از یک پارتوی بهینه یا ایده‌آل محاسبه نماید (ولدھوزین، ۱۹۹۹):

$$GD = \frac{1}{NS} \sqrt{\sum_{i=1}^{NS} d_i^2} \quad (33)$$

شاخص *GD* برای پارتوی *NSGA-II* برابر با ۰,۱۴۶۸۷ و برای *MOPSO* برابر با ۰,۲۰۵۹۸ است. بنابراین براساس شاخص *GD* پارتوی روش *NSGA-II* مناسب‌تر است.

معیار کیفیت (QM) - این معیار توسط یکپارچه‌سازی جواب‌های بهینه پارتوی دو الگوریتم اندازه‌گیری می‌شود. اگر *P1* و *P2* به ترتیب جواب‌های پارتوی بهینه الگوریتم‌های *A1* و *A2* باشد، جواب‌های پارتوی تلفیقی دو الگوریتم شامل *P* جواب بهینه پارتوی می‌شود که این مقدار از *P1* و *P2* بیشتر بوده و از *P1+P2* کمتر است. شاخص *QM* برای الگوریتم‌های *A1* و *A2* به ترتیب توسط *P1/P* و *P2/P* محاسبه می‌شود. شاخص *QM* در این مثال برای *NSGA-II* برابر ۰,۷۳۱ و برای الگوریتم *MOPSO* برابر ۰,۲۶۸ است. بنابراین، براساس شاخص *QM* نیز جواب‌های پارتوی *NSGA-II* از جواب‌های پارتوی *MOPSO* مناسب‌تر است.

1- The number of pareto solutions (NPS)

2- Genrational Distance (GD)

3- Veldhuizen

4- Quality metric

در کل براساس شاخص‌های NPS ، GD و QM محاسبه شده برای دو الگوریتم، می‌توان نتیجه گرفت که در این مثال کیفیت جواب‌های پارتوی $NSGA-II$ برتر از $MOPSO$ است.

نتیجه‌گیری

مطالعات زیادی در خصوص تخصیص افزونگی انجام شده است اما این مطالعات بیشتر با هدف حداکثر کردن قابلیت اطمینان در محصولات الکترونیکی انجام شده است و کمتر بحث تخصیص افزونگی در خطوط تولید و مونتاژ مورد بررسی قرار گرفته است. همچنین در تحقیقاتی که بر روی تخصیص افزونگی در خطوط تولید صورت پذیرفته است، هیچ‌یک از مدل‌های پیشین قابلیت در نظرگیری توابع توزیع عمومی برای تمامی پارامترهای مبتنی بر زمان مانند مدت زمان پردازش، مدت زمان بین خرابی‌ها و مدت زمان‌های تعمیر را نداشته‌اند. از این رو این تحقیق با ارائه یک مدل ریاضی چندهدفه با در نظرگیری اهداف حداکثرسازی نرخ تولید، حداقل‌سازی هزینه کل و حداکثرسازی کیفیت محصولات به مسئله تخصیص افزونگی در خطوط تولید و مونتاژ نامطمئن پرداخته است. از طرفی از آنجایی که روش پیشنهادی تحقیق در پی در نظرگیری تمامی توابع توزیع برای پارامترهای مبتنی بر زمان است، از رویکرد تلفیقی شبیه‌سازی کامپیوتری و متدولوژی سطح پاسخ جهت برآورد نرخ تولید و کیفیت محصولات بهره می‌برد. جهت حل مدل ریاضی پیشنهادی، از دو الگوریتم فراابتکاری تکاملی الگوریتم ژنتیک با مرتب‌سازی نامغلوب ($NSGA-II$) و بهینه‌سازی ازدحام ذرات چندهدفه ($MOPSO$) استفاده شده است. به علاوه جهت مقایسه و انتخاب جواب‌های پارتوی مناسب‌تر از شاخص‌های فاصله نسل (GD)، تعداد جواب‌های به دست آمده پارتو (NPS) و معیار کیفیت (QM) استفاده شده است. نتایج حل دو الگوریتم نشان می‌دهد که تعداد جواب‌های بهینه پارتو برای $NSGA-II$ برابر ۵۰ و برای الگوریتم $MOPSO$ برابر ۳۱ است؛ همچنین شاخص GD برای $NSGA-II$ برابر ۰,۱۴۶ و برای الگوریتم $MOPSO$ برابر ۰,۲۰۵ می‌باشد؛ به علاوه شاخص QM برای $NSGA-II$ برابر ۰,۳۳۱ و برای الگوریتم $MOPSO$ برابر ۰,۲۶۸ است. نتایج محاسباتی نشان می‌دهد که با توجه به سه شاخص مطرح شده، جواب‌های پارتوی بهینه $NSGA-II$ در این مسئله خاص از کیفیت بالاتری

نسبت به الگوریتم *MOPSO* برخوردار می‌شود. لازم به توضیح است که در این مطالعه پارامترهای الگوریتم‌های *NSGA-II* و *MOPSO* به صورت تجربی/آزمایشی تعیین شده است. با وجود آزمایش‌های زیاد انجام شده جهت تعیین پارامترها، جهت مقایسه بهتر دو الگوریتم پیشنهاد می‌شود در مطالعات آتی پارامترهای دو الگوریتم با استفاده از روش‌های طراحی آزمایشات تنظیم شده و سپس دو الگوریتم مورد مقایسه قرار گیرند. از طرف دیگر با توجه به حجم بسیار زیاد محاسبات مسئله (طراحی آزمایشات، شبیه‌سازی هر یک از آزمایش‌ها، ساخت متامدل، ساخت مدل ریاضی بهینه‌سازی، حل مدل با استفاده از الگوریتم‌های فراابتکاری)، مقایسه دو الگوریتم *NSGA-II* و *MOPSO* در این مطالعه فقط بر روی یک مسئله انجام شده و نتایج مقایسه‌ای صرفاً در مورد یک مسئله خاص ارائه شده است، بنابراین در حال حاضر نتایج این تحقیق جهت برتری مطلق یک الگوریتم قابل تعمیم نبوده و جهت بررسی برتری مطلق یکی از الگوریتم‌ها نیاز به حل مسائل بیشتر و استفاده از آزمون‌های آماری مناسب است که پیشنهاد می‌شود این امر در تحقیقات آتی مورد مطالعه قرار گیرد.

پیشنهادها

در تحقیق حاضر سعی شده که ابعاد مناسبی از انعطاف‌پذیری زنجیره‌تأمین جهت مدل‌سازی انتخاب شوند، اما می‌توان ابعاد دیگری مانند فن‌آوری اطلاعات را جهت تکمیل مدل نیز به آن اضافه کرد.

مدل به‌نحوی تنظیم شده که می‌توان در حلقه آخر زنجیره‌تأمین، بخش توزیع را نیز با اضافه کردن یک یا دو محدودیت و یک متغیر صفر و یک تحت پوشش در آورد که در این مطالعه به دلیل عدم دسترسی به اطلاعات سیستم توزیع در محیط عملیاتی از این کار صرف نظر شده است.

در این مقاله با توجه به بررسی‌های صورت گرفته، از مفهوم فروش ازدست‌رفته، برای بیان انعطاف‌پذیری استفاده شده است، که می‌توان به طرق دیگری مانند میزان فروش محصول، ترکیب محصول ارائه شده، سود حاصل از فروش و ... برای ارائه مدل اقدام کرد.

منابع

- Burman, M. H. (1995). *New results in flow line analysis*. MIT, Cambridge, MA.
- Dallery Y., Gershwin S.B. (1992). Manufacturing flow line systems: A review of models and analytical results: Queuing systems theory and applications. *special issue on queuing models of manufacturing systems*. 12 (2), 3-94.
- Dallery, Y., Frein, Y. (1998). An efficient method to determine the optimal configuration of a flexible manufacturing system. *Annals of Operations Research*, 15. 207 – 225.
- Deb K. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions Evolutionary Computations*. 6, 182–97.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. John Wiley and Sons, New York.
- Deb, K., Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex Systems*. 9, 115–148.
- Deb, K., Goyal, M. (1996). A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science Informations*. 26, 30–45.
- Garg, H., Rani, M., Sharma, S. (2013). An efficient two phase approach for solving reliability-redundancy allocation problem using artificial bee colony technique. *Computers & Operations Research*. 40 (12), 2961-2969.
- Gershwin S.B., Schor J.E. (2000). Efficient algorithms for buffer space allocation. *Operations research*. 93 (3), 117 – 144.
- He, P., Wu, K., Xu, J., Wen, J., Jiang, Z. (2013). Multilevel redundancy allocation using two dimensional arrays encoding and hybrid genetic algorithm. *Computers & Industrial Engineering*. 64 (1), 69-83.
- Kleijnen, J. P. C. (1979). Regression meta-models for generalizing simulation results. *IEEE SMC-9*. 2, 93–96.
- Kleijnen, J. P. C., Sargent, R. G. (2000). A methodology for fitting and validating metamodels in simulation. *European Journal of Operations Research*, 120, 14-29.
- Kumar, R., Izui, K., Yoshimura, M., Nishiwaki, Sh. (2009). Multi-objective hierarchical genetic algorithms for multilevel redundancy allocation optimization, *Reliability Engineering and System Safety*. 94 (4), 891-904.

Kuo, W., Prasad, V.R. (2000). An annotated overview of system-reliability optimization. *IEEE transactions on reliability*. 49 (2), 176-187.

Kuo, W., Prasad, V.R., Tillman, F.A., Hwang, C.L. (2001). *Optimal reliability design: fundamentals and applications*. Cambridge University, London.

Kuo, W., Wan, R. (2007). Recent advances in optimal reliability allocation. *IEEE transactions on systems, Man and cybernetics, Part A* 37 (2), 143-156.

Lavoie, P., Kenne, J., Gharbi, A. (2009). Optimization of production control policies in failure-prone homogenous transfer lines. *IEEE Transactions*. 41, 209-222.

Montgomery, D. (2001). *Design and analysis of experiments*. John Wiley and Sons, New York.

Nahas N., Nourelfath, M., Ait-Kadi D. (2007). Coupling ant colony and the degraded ceiling algorithm for the redundancy allocation problem of series-parallel systems. *Reliability Engineering and System Safety*. 92 (2), 211-222.

Ouzineb, M., Nourelfath, M., Gendreau, M. (2008). Tabu search for the redundancy allocation problem of homogenous series-parallel multi-state systems. *Reliability Engineering and System Safety*. 93 (8), 1257-1272.

Parsopoulos, K. E., Vrahatis, M. N. (2002). Particle swarm optimization method in multiobjective problems. *In Proceedings of the 2002 ACM symposium on applied computing*, Madraid, Spain, 603–607.

Ravi, V., Reddy, P.J. (2000). Zimmermann, H. Fuzzy global optimization of complex system reliability. *IEEE transactions on Fuzzy Systems* 8 (3), 241-248.

Tan, K. C., Lee, T. H., Khor, E. F. (2001). Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons. *In Proceeding of IEEE congress on the evolutionary computation*, Korea, Seoul, 979–986.

Veldhuizen, D. (1999). *Multiobjective evolutionary algorithms: Classification, analyses, and new innovations*. Dayton, Ohio: Department of Electrical and Computer Engineering Air Force Institute of Technology.

Yeh, W.C. (2009). A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems. *Expert systems with applications*. 36 (5), 9192-9200.

Yun, W.Y., Kim, J.W. (2004). Multi-level redundancy optimization in series systems. *Computers and Industrial Engineering*. 46 (2), 337-346.

Yun, W.Y., Song, Y.M., Kim, H.G. (2007). Multiple multi-level redundancy allocation in series systems. *Reliability Engineering and System Safety*. 92 (3), 308-313.

Zhao, P., Chan, P., Li, L., Keung Tony Ng, H. (2013). On allocation of redundancies in two-component series systems. *Operations research letters*. 41 (6), 690-693.

Zoufaghari, H., Hamadani, A., Ardakan, M. (2014). Bi-objective redundancy allocation problem for a system with mixed repairable and non-repairable components. *ISA Transactions*. 53 (1), 17-24.

