مهندسی صنایع و مدیریت شریف، (تابستان ۱۳۹۳)
دوره‌ی ۱-۳۰، شماره‌ی ۱/۱، ص. ۵۹-۶۷

Original Article

ارائه‌ی یک رویکرد جست‌وجوی شناور و مبتنی بر روش پارتو به‌منظور حل مسئله‌ی تولید کارگاهی انعطاف‌پذیر چندهدفه

سهیل عزیزی بروجردی (دانشجوی کارشناسی ارشد)
پرویز فتاحی * (دانشیار)
گروه مهندسی صنایع، دانشگاه بوعلی‌سینا همدان
مهدی فتح‌الله (استادیار)
گروه مهندسی صنایع، دانشگاه آزاد اسلامی واحد کرج

در این نوشتار مسئله‌ی زمان‌بندی تولید کارگاهی انعطاف‌پذیر در حالت چندهدفه با سه معیار «دوره زمانی ساخت»، «مجموع بارکاری» و «بیشینه بارکاری ماشین‌ها» مورد بررسی قرار می‌گیرد. بدین‌منظور ابتدا مسئله بررسی و مدل ریاضی آن ارائه شده، و سپس با توجه به ماهیت بسیار پیچیده‌ی مسئله از یک نگرش یک‌پارچه برای حل آن استفاده شده است. نگرش پیشنهادی با استفاده از یک رویه جست‌وجوی شناور و چندین الگوریتم ابتکاری جست‌وجوی محلی، مسئله را از حالت بررسی سلسله‌مراتبی به حالت یک‌پارچه تبدیل می‌کند و سپس با استفاده از یک الگوریتم مبتنی بر روش‌های پارتو و NSGAII [1] مجموعه جواب پارتو را به دست می‌آورد. به‌منظور تحلیل کارایی و اثربخشی الگوریتم پیشنهادی، تجزیه و تحلیل‌های عددی در ابعاد مختلف انجام و با نتایج تحقیق‌های قبلی مقایسه شده است. نتایج به دست آمده کارایی الگوریتم پیشنهادی را هم در کاهش تغییرپذیری جست‌وجو و هم در بهبود جواب‌های نهایی نشان می‌دهد.

واژگان کلیدی: تولید کارگاهی انعطاف‌پذیر، تغییرپذیری، الگوریتم‌های ابتکاری چندهدفه، رویکرد جست‌وجوی شناور.

soheil.azizi1@gmail.com
fattahi@basu.ac.ir
fath@iust.ac.ir

۱. مقدمه

«تولید کارگاهی» یکی از سیستم‌های رایج تولید در تمامی کشورهای دنیاست. به‌همین دلیل تلاش‌های گسترده‌یی برای افزایش بهره‌وری این سیستم‌ها صورت گرفته است. افزایش شدت رقابت بین مؤسسات تولیدی در راستای کاهش قیمت‌ها، تحویل به‌موقع و افزایش رضایت مشتریان موجب شده است که استفاده‌ی بهینه از منابع تولیدی و افزایش بهره‌وری واحد از اهمیت بسیار بالایی برخوردار باشد. یکی از عوامل بسیار مهم در افزایش بهره‌وری و رضایت مشتریان وجود یک سیستم برنامه‌ریزی قوی در واحد تولیدی است.

زمان‌بندی تولید، مسئله‌یی است که بر تحویل به‌موقع کالاها و نیز بر استفاده‌ی بهینه از سرمایه‌های سازمان اثرگذار است. در هر مسئله‌ی زمان‌بندی، زمان‌های شروع و پایان فعالیت‌ها، دستگاه‌ها و تجهیزات مورد نیاز و... تعیین می‌شود. با توجه به تأثیر زمان‌بندی بر اهداف سازمان‌های تولیدی، ارائه‌ی مدل‌های زمان‌بندی از دیرباز مد نظر بوده است. با این که محققین زیادی در زمینه‌ی ارائه‌ی مدل‌های زمان‌بندی فعالیت کرده‌اند، هنوز مدل‌هایی که کاملاً جواب‌گوی نیاز سازمان‌ها باشد به‌صورت جامع ارائه نشده است. «تولید کارگاهی» یکی از شاخه‌های مسائل زمان‌بندی و نیز یکی از مهم‌ترین مسائل بهینه‌سازی ترکیبی است. «تولید کارگاهی کلاسیک» شامل زمان‌بندی مجموعه‌یی از کارها روی مجموعه‌یی از ماشین‌هاست، با این محدودیت که هر کار دارای یک سری عملیات است که باید به‌صورت متوالی انجام پذیرد. مسئله‌ی تولید کارگاهی مسئله‌یی بسیار سخت است[1] و روش‌های مختلف هیوریستیک و متاهیوریستیک برای حل آن بسط داده شده است.

۱.۱. مرور ادبیات

در سال‌های اخیر الگوریتم‌های متاهیوریستیک زیادی مانند جست‌وجوی ممنوع، بهینه‌سازی ذرات، بهینه‌سازی مورچگان و الگوریتم ژنتیک به‌منظور حل مسئله‌ی تولید کارگاهی انعطاف‌پذیر تک‌هدفه ــ شامل دو رویکرد سلسله‌مراتبی و رویکرد یک‌پارچه ــ در مطالعات مختلف مورد بررسی قرار گرفته است. در رویکرد سلسله‌مراتبی، مسئله به دو زیرمسئله‌ی توالی و تخصیص تبدیل، و به‌طور مستقل از هم بررسی می‌شوند. برخی این رویکرد را برای مسئله بسط دادند، اما آن را در حالت یک‌پارچه تخصیص و

* نویسنده مسئول
تاریخ: دریافت ۱۳۹۰/۱۰/۱۰، اصلاحیه ۱۳۹۱/۲/۳، پذیرش ۱۳۹۱/۳/۲۳.

توالی همزمان و وابسته به هم بررسی کرده‌اند.[۳و۲] عده‌یی نیز الگوریتم جست‌وجوی ممنوع را برای حل به روش یک‌پارچه توسعه دادند.[۵و۴]

علاوه بر رویکردهای مورد استفاده برای حل مسئله، نحوه‌ی رمزگذاری راه‌حل (ساختارکدینگ یک جواب) نیز نقش مهمی در توسعه‌ی ابزارهای حل مسئله دارد. در این ارتباط می‌توان به ساختارهایی اشاره کرد که توسط محققین برای حل مسئله‌ی تولید کارگاهی انعطاف‌پذیر ارائه شده است. نمایش موازی کارها،[۶] کوروموزومی به‌صورت جدول $J * M$ ارائه می‌کرد که هر ردیف یک کار را نمایش می‌دهد و هر عضو یک جفت مرتب ماشین ـ زمان شروع را برای هر یک از عملیات آن کار نشان می‌داد. نمایش غیرمستقیم براساس یک جفت کوروموزوم A و B ارائه شد؛[۷] A رشته‌ی ماشین‌های تخصیص داده شده و B رشته‌ی توالی عملیات است. در مطالعات دیگر، روش رمزگذاری دیگری ارائه شد[۸] که هر عضو آن نشان‌گر عملیات صورت گرفته است و شامل شماره‌ی ماشین تخصیص‌یافته، زمان شروع و زمان پایان عملیات می‌شود.

محققین براساس الگوریتم ژنتیک چندهدفه، کوروموزمی مبتنی بر ماشین‌های تخصیص‌یافته معرفی کردند.[۹] همچنین روشی مطرح کردند که در آن از دو ماتریس استفاده می‌شد:[۱۰] ماتریس اول به معرفی تخصیص و ماتریس دوم به معرفی اولویت عملیات‌های اختصاص داده شده به یک ماشین می‌پرداخت. گائو[۱۱] از یک کروموزم رشته‌یی برای تخصیص ماشین‌ها استفاده نمود که در آن از تکرار جایگشتی براساس زمان‌بندی عملیات‌های اختصاص یافته استفاده می‌کرد. در تحقیقی دیگر، ساختاری دوبخشی برای تخصیص و توالی ارائه شد،[۱۲] در بخش تخصیص هر عملیات به‌تعداد ماشین‌هایی که می‌تواند آن را انجام دهد دارای بردار صفر و ۱ است؛ ۱ به‌معنی انجام این عملیات توسط آن ماشین است. در ادامه نیز، ساختاری دوبخشی شامل مجموعه استراتژی‌هایی برای تعیین تخصیص و استراتژی‌هایی برای تعیین توالی براساس قواعد توزیع[۲] ارائه شد.[۱۳]

توسعه‌ی اهداف مسئله‌ی تولید کارگاهی انعطاف‌پذیر و بررسی حالت چندهدفه در سال‌های اخیر مورد توجه محققین قرارگرفته است. این امر به دو صورت وزن‌دهی تجمعی و مجموعه جواب پارتو بررسی شده است. در حالت اول با استفاده از وزن‌دهی به اهداف، تولید کارگاهی انعطاف‌پذیر به‌یک مسئله‌ی تک‌هدفه تبدیل شده و با استفاده از الگوریتم‌های متاهیوریستیک تک‌هدفه حل شده است. مجموعه جواب پارتو با استفاده از الگوریتم‌های متاهیوریستیک چندهدفه، جواب‌هایی غیرمغلوب را به تصمیم‌گیرنده ارائه می‌دهد. این روش‌ها به‌علت پیچیده‌تر شدن فضای جست‌وجوی مسئله، کم‌تر مورد توجه بوده‌اند. در یکی از مطالعات[۸] مسئله‌ی چندهدفه‌ی تولید کارگاهی انعطاف‌پذیر براساس رویکرد بهینه‌سازی محلی و پارتو در روش‌های فازی بررسی شده است. بدین‌منظور سه هدف طول دوره‌ی ساخت، مقدارکل حجم کاری، و بیشینه حجم کاری را در مدل وارد کرده و برای حل آن یک ساختار رمزگذاری متناسب ارائه دادند. محققین با ارائه‌ی یک الگوریتم جست‌وجوی ممنوعه‌ی چندهدفه، از بعد گسترده‌تری مسئله‌ی تولید کارگاهی انعطاف‌پذیر را بررسی[۱۴] و فرض چندفرایندی را علاوه بر ماشین‌ها در مسئله وارد کردند. همچنین یک روش سلسله‌مراتبی برای حل ابداع،[۱۵] گائو[۱۱] از الگوریتم بهینه‌سازی گروهی ذرات (PSO)[۳] و از الگوریتم انجماد تدریجی (SA)[۴] برای زمان‌بندی عملیات استفاده کردند. در مطالعه‌یی دیگر نیز،[۱۶] با استخراج قوانین توزیع از معیارهای غالب رایج در سایر مسائل زمان‌بندی، و با استفاده از الگوریتم ژنتیک نسبت به حل مسئله اقدام شد.

رویکرد سلسله‌مراتبی با کوچک‌کردن فضای جواب از پیچیدگی مسئله می‌کاهد. رویکرد یک‌پارچه هر دو زیرمسئله را با هم بررسی کرده و به‌طور تصادفی در دو فضا به جست‌وجو می‌پردازد. در حالت سلسله‌مراتبی پس از یافتن تخصیص، مسئله‌ی تولید کارگاهی حل می‌شود. بنابراین همواره برخی جواب‌های خوب از بین می‌روند زیرا مسئله‌ی تولید کارگاهی برای تخصیص مشخص‌شده حل می‌شود. درصورتی که ممکن است تغییر تخصیص جواب‌های بهتری ارائه دهد. در حالت یک‌پارچه چون هیچ روند خاصی جز تغییر تصادفی بین فضای تخصیص و توالی تعیین نمی‌شود، این حالت جست‌وجو منجر به افزایش تغییرپذیری در فضای جواب می‌شود. بنابراین طراحی روندی که علاوه بر کاهش پیچیدگی مسئله (رویکرد سلسله‌مراتبی)، جست‌وجوی جامع‌تری را در فضای جواب مسئله انجام دهد، ضرورت می‌یابد.

چنان که بیان شد، در مطالعات انجام‌شده مسئله به‌صورت سلسله‌مراتبی یا به‌صورت یک‌پارچه حل شده است. در این مقاله با ترکیب یک الگوریتم متاهیوریستیک و الگوریتم‌های جست‌وجوی محلی ابتکاری رویکردی جدید برای حل مسئله ارائه شده است که علاوه بر برخورداری از مزایای هر دو رویکرد سلسله‌مراتبی و یک‌پارچه، از تغییرپذیری مسئله هنگام جست‌وجوی جواب نیز کاسته است. ساختار این نوشتار چنان است که در بخش ۲ فرضیات، محدودیت‌ها و اهداف مسئله‌ی تولید کارگاهی انعطاف‌پذیر چندهدفه توصیف شده، و در بخش ۳ رویکرد جست‌وجوی شناور و الگوریتم NSGAII برای حل مسئله‌ی تولید کارگاهی انعطاف‌پذیر چندهدفه معرفی شده است. در بخش ۴ نیز الگوریتم پیشنهادی ارائه شده است. نتایج محاسباتی، تحلیل و بحث نیز در بخش ۵ آورده شده است. در نهایت، در بخش ۶ بحث و نتیجه‌گیری و همچنین پیشنهادات برای تحقیقات آتی ارائه شده است.

۲. معرفی مسئله

در مسئله‌ی تولید کارگاهی انعطاف‌پذیر، زمان‌بندی اجرای n کار روی mماشین انجام می‌شود. هرکار دارای یک توالی از عملیات $O_{j,h}, h = ۱, ... h_j$ است که $O_{j,h}$ و h_j به‌ترتیب نشان‌دهنده‌ی hامین عملیات از کار j و تعداد عملیات مورد نیاز برای اجرای j است. مجموعه ماشین‌ها عبارت‌اند از $M = \{m_۱, m_۲, ..., m_m\}$. اندیس j نشان‌دهنده‌ی کار، اندیس h نشان‌دهنده‌ی عملیات و اندیس i نشان‌دهنده‌ی ماشین است. عملیات $O_{j,h}$ را می‌توان روی ماشین $M_{j,h} \subset M$ با زمان پردازش $P_{i,j,h}$ اجرا کرد. مجموعه‌ی $M_{j,h}$ براساس $a_{i,j,h}$ تعریف می‌شود و اندیس k مجموعه عملیات تخصیص‌یافته به هر ماشین را مشخص می‌کند. طبق این توضیحات پارامترهای مدل عبارت‌اند از: n تعداد کارها؛ m تعداد ماشین؛ $a_{i,j,h}$ عملیات‌های تخصیص یافته به ماشین که چنین تعریف می‌شود:

$$a_{i,j,h} = \begin{cases} ۱ & \text{if } O_{j,h} \text{ is done on machine } i \\ ۰ & \text{elsewhere} \end{cases}$$

$P_{i,j,h}$ زمان انجام عملیات؛ $O_{j,h}$ در صورتی که روی ماشین i پردازش شود. متغیرهای تصمیم مسئله نیز عبارت‌اند از: $C_{\max}$ طول دوره ساخت؛ w_i حجم کاری ماشین i.

$$y_{i,j,h} = \begin{cases} ۱ & \text{if machine } i \text{ is selected for } O_{j,h} \\ ۰ & \text{elsewhere} \end{cases}$$

$$x_{i,j,h,k} = \begin{cases} ۱ & \text{if } O_{j,h} \text{ is done on machine } i \text{ in priority } k \\ ۰ & \text{elsewhere} \end{cases}$$

$t_{j,h}$ زمان شروع عملیات $O_{j,h}$؛ $T\,m_{i,k}$ زمان شروع به کار ماشین i در اولویت k؛ K_i تعداد عملیات تخصیص یافته به ماشین i؛ $P\,s_{j,h}$ زمان انجام عملیات $O_{j,h}$ بعد از انتخاب ماشین جهت پردازش آن.

در ادامه، با توجه به موارد ذکر شده در فوق، و نیز ضمن تشریح محدودیت‌های به کار رفته، مدل برنامه‌ریزی خطی مسئله ارائه شده است: تابع هدف f_1 طول دوره‌ی ساخت، تابع هدف f_2 بیشینه بارکاری، و تابع هدف f_3 حجم کلی بارکاری را تعیین می‌کند. محدودیت ۱ برابر طول دوره‌ی ساخت است؛ محدودیت ۲ زمان پردازش عملیات $O_{j,h}$ روی ماشین انتخاب شده را مشخص می‌کند؛ و محدودیت ۳ همان محدودیت پیش‌نیاز عملیات است. محدودیت‌های ۴، ۵ و ۶ موجب می‌شود که هر عملیاتی زمانی انجام شود که ماشین مورد نظر برای اجرای آن بیکار بوده و عملیات پیش‌نیازی آن انجام شده باشد. محدودیت ۷ نیز ماشین‌هایی را که قابلیت انجام هر عملیات را دارند مشخص می‌کند. محدودیت ۸ عملیات‌ها را به ماشین تخصیص داده و توالی عملیات‌های تخصیص‌یافته را روی هر ماشین مشخص می‌کند. محدودیت‌های ۹ و ۱۰ نیز موجب می‌شوند که هر عملیات فقط در یک اولویت و فقط روی یک ماشین پردازش شود.

$$Min\,(f_1 = C_{\max}, f_2 = \max_{1\le i\le m} w_i\; f_3 = \sum_{i=1}^{m} w_i)$$

S.t.

$$c_{\max} \ge t_{j,h_j} + P\,s_{j,h_j} \quad for \quad j = 1, \dots, n; \qquad (1)$$

$$\sum_i y_{i,j,h} \,.\, P_{i,j,h} = P\,s_{j,h} \;\; for\, j = 1, \dots, n;\; h = 1, \dots, h_j; \qquad (2)$$

$$t_{j,h} + P\,s_{j,h} \le t_{j,h+1} \; for\; j = 1, \dots, n; h = 1, \dots, h_{j-1}; \qquad (3)$$

$$T\,m_{j,k} + P\,s_{j,h} \,.\, x_{i,j,h,k} \le T\,m_{j,k+1} \; for\; i = 1, \dots, m;$$
$$j = 1, \dots, n; h = 1, \dots, h_j; k = 1, \dots, k_j - 1; \qquad (4)$$

$$T\,m_{j,k} \le t_{j,h} + (1 - x_{i,j,h,k}).L \; for\; i =$$
$$1, \dots, m;\; j = 1, \dots, n; h = 1, \dots, h_j; k = 1, \dots, k_i; \qquad (5)$$

$$T\,m_{i,k} + (1 - x_{i,j,h,k}).L \ge t_{j,h} \; for\; i = 1, \dots, m;$$
$$j = 1, \dots, n; h = 1, \dots, h_j; k = 1, \dots, k_i; \qquad (6)$$

$$y_{i,j,h} \le a_{i,j,h} \; for\; i = 1, \dots, m;\; j = 1, \dots, n;\; h = 1, \dots, h_j; \qquad (7)$$

$$\sum_j \sum_h x_{i,j,h,k} = 1 \; for\; i = 1, \dots, m; \; ; k = 1, \dots, k_i; \qquad (8)$$

$$\sum_i y_{i,j,k} = 1 \; for\; j = 1, \dots, n; h = 1, \dots, h_j; \qquad (9)$$

$$\sum_k x_{i,j,h,k} = y_{i,j,h} \; for\; i = 1, \dots, m;\; j = 1, \dots, n;$$
$$h = 1, \dots, h_j; \qquad (10)$$

$$t_{j,h} \ge 0 \; for\; j = 1, \dots, n; h = 1, \dots, h_j; \qquad (11)$$

$$P\,s_{j,h} \ge 0 \; for\; j = 1, \dots, n; h = 1, \dots, h_j; \qquad (12)$$

$$T\,m_{i,k} \ge 0 \; for\; i = 1, \dots, m; k = 1, \dots, k_i; \qquad (13)$$

$$x_{i,j,h,k} \in \{0, 1\} \; for\; i = 1, \dots, m;\; j = 1, \dots, n;$$
$$h = 1, \dots h_j; k = 1, \dots, k_i; \qquad (14)$$

$$y_{i,j,h} \in \{0, 1\} \; for\; i = 1, \dots, m;\; j = 1, \dots, n;\; h = 1, \dots, h_j; \qquad (15)$$

۳. الگوریتم NSGAII

NSGAII یک الگوریتم نخبه‌گرای چندهدفه است که در آن یک حافظه به جواب‌های پارتو اختصاص می‌یابد و در هر بار تولید نسل این حافظه به‌روز می‌شود. بنابراین الگوریتم هر بار بهترین جواب‌ها را از ابتدای تولید نسل با خود همراه دارد و با جواب‌های جدیدتر مقایسه می‌کند. در حال حاضر این الگوریتم یکی از کارآمدترین روش‌ها برای حل مسائل چندهدفه است. NSGAII همچنین در چند مطالعه‌ی انجام‌شده درخصوص تولید کارگاهی انعطاف‌پذیر مورد توجه و استفاده محققین بوده است.[۱۷و۱۸] در بهینه‌سازی چندهدفه، $\hat{x}$ یک راه حل غیرمسلط نامیده می‌شود، اگر نتوان ارزش‌های موجود از تمامی اهداف را به‌طور همزمان توسط هیچ راه حل علمی دیگری بهبود بخشید. به‌عبارت دیگر، $\hat{x}$ یک راه حل مؤثر خواهد بود چنانچه هیچ راه حل علمی دیگر نظیر $x\varepsilon s$ وجود نداشته باشد که به‌ازای تمامی توابع هدف f_i : $f(x) \le f_i(\hat{x})$ و حداقل یک تابع هدف وجود داشته باشد که: $f(x) \le f_i(\hat{x})$. به مجموعه‌ی این جواب‌های غیرمسلط «مجموعه جواب‌های پارتو» نیز می‌گویند. در این روش ابتدا جمعیت اولیه، و سپس فرزندان تولید می‌شوند و از ترکیب آنها جمعیت جدید برای رتبه‌بندی ایجاد می‌شود. اگر جمعیت اولیه (Pt) دارای N عضو، و جمعیت فرزندان (Qt) نیز دارای N عضو باشد، بنابراین جمعیت ترکیبی حاصل از آنها (Rt) دارای $2N$ عضو است. پس از تولید و ترکیب جمعیت، یک مرتب‌سازی براساس نامغلوب بودن اعضای (Rt) انجام، و تمامی این اعضا رتبه‌بندی می‌شوند. در هر مرحله‌ی تولید نسل، جواب‌های نامغلوب برای قرارگیری در مجموعه‌ی والدین (Pt) با یکدیگر رقابت می‌کنند. روال تخصیص الگوریتم مطابق الگوریتم‌هایی است که در ادامه تشریح شده‌اند.

۱.۳. الگوریتم NSGAII پیشنهادی

گام. جمعیت اولیه (Pt) تعداد N تولید تصادفی داشته است. از عملگرهای انتخاب معمول در الگوریتم ژنتیک برای تولید فرزندان استفاده می‌شود و جمعیت فرزندان (Pt) تولید می‌شود.

گام ۱. جمعیت فرزندان و والدین را با یکدیگر ترکیب کرده و (Rt)ساخته می‌شود: $R_t = P_t \cup Q_t$.

گام ۲. با قرار دادن $i = 1,\ P_t + 1 = \phi$، تا زمانی که $|P_{t+1}| + |F_i| < N$ عملیات زیر را تکرار کنید:

$$\{P_{t+1} = P_{t+1} \cup F_i, i = i + 1\}$$

گام ۳. عملگر مرتب‌سازی ازدحام را برای جواب‌های رتبه‌بندی شده در مجموعه‌های (F_i) اجرا، و برای $(N - |P_{t+1}|)$ جواب باقی‌مانده مجموعه‌ی (P_{i+1}) را تکمیل کنید.

گام ۴. با استفاده از الگوریتم انتخاب مسابقه‌یی ازدحام و عملگرهای ترکیب و جهش، جمعیت فرزندان (Q_{t+1}) را از (P_{t+1}) ایجاد کنید.

۲.۳. عملگر مسابقه‌یی ازدحام

گام ۱. قرار دهید $L = |F|$ یا L برابر اندازه مجموعه F، برای هر جواب i از مجموعه F قرار می‌دهیم $d_i = 0$.

گام ۲. برای هر تابع هدف $i = 1, 2, 3$ مجموعه f_i را به ترتیب نزولی و برحسب ارزش

آنها مرتب می‌کنیم. درواقع بردار اندیس مرتب شده I^i که در آن $I^i = Sort(f_i >)$ تولید می‌شود.

گام ۳. از $m = ۱$ تا $m = M$ یک مقدار بزرگ برای حدود جواب‌ها اختصاص داده و یا $d_{I_۱^m} = d_{I_۲^m} = \infty$ و برای تمام جواب‌های دیگر، $j = ۲, ۳, ... (l-۱)$ قرار دهید:

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{(I_{j+۱}^m)} - f_m^{(I_{j-۱}^m)}}{f_m^{\max} - f_m^{\min}}$$

۳.۳. رویه‌ی مرتب‌سازی نامغلوب

گام ۱. برای هر $p, q \in P$ (P جمعیت جاری) اگر q, p را تحت تسلط قرار می‌دهد، قرار دهید: $S_p = S_q + \{q\}$

اگر q, p را تحت تسلط قرار می‌دهد، قرار دهید: $n_p = n_q + ۱$

اگر $n_p = ()$ قرار دهید: $F_۱ = F_۱ \cup \{p\}$.

گام ۲. قرار دهید $i = ۱$، تا زمانی که $F_i = \phi$ قرار دهید $H = \phi$ برای هر $p \in F_i$ و هر $q \in s_p$ قرار دهید: $n_q = n_q - ۱$

اگر $n_p = ()$ قرار دهید: $H = H \cup \{q\}$

$F_۱ = H \quad i = i + ۱$.[۱۹]

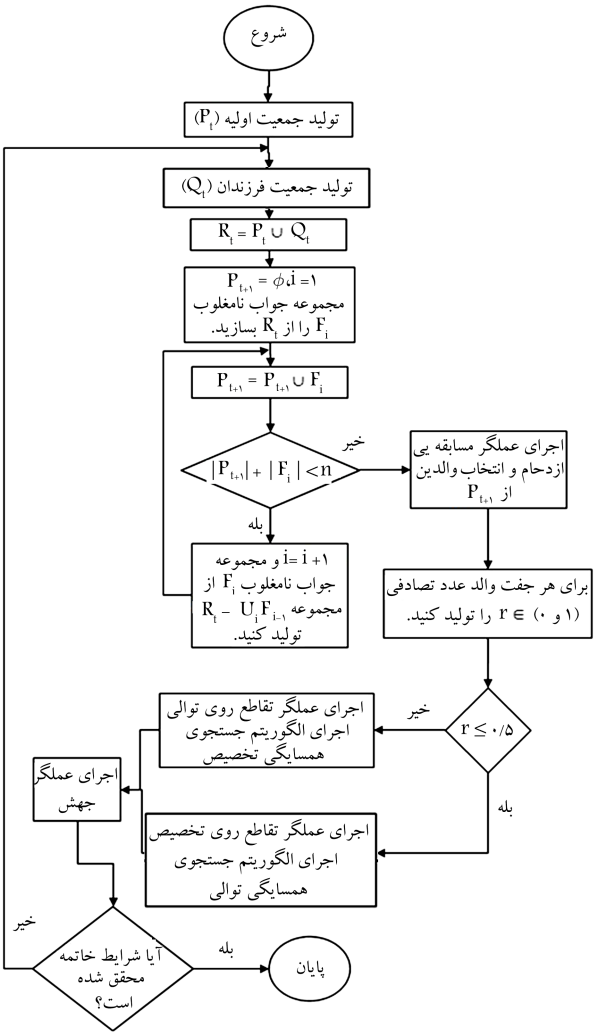
۴. نگرش مورد استفاده برای حل مسئله

چنان که پیش‌تر اشاره شد، با توجه به دوبخشی بودن فضای جست‌وجوی جواب مسئله، تا به‌حال رویکردهای سلسله‌مراتبی و یک‌پارچه برای حل مسئله مورد استفاده قرارگرفته‌اند. رویکرد جست‌وجوی شناور به‌کمک الگوریتم‌های ابتکاری جست‌وجوی محلی، ابتدا مسئله را به دو زیرمسئله‌ی تخصیص و توالی تبدیل می‌کند. سپس جست‌وجو در فضای مسئله‌ی تخصیص انجام می‌گیرد، اما پس از رسیدن به نقطه‌ی قابل قبول با استفاده از الگوریتم ابتکاری، جست‌وجو در فضای مسئله‌ی توالی برای رسیدن به نقطه‌ی بهینه انجام می‌پذیرد. در حالت توالی نیز این روند برای تخصیص تکرار می‌شود. انطباق این رویکرد با الگوریتم‌های مختلف متفاوت است. در این نوشتار برای حل مسئله از ترکیب الگوریتم NSGAII و جست‌وجوی شناور استفاده می‌شود. همچنین برای کارآمدتر شدن الگوریتم پیشنهادی از الگوریتم‌های جست‌وجوی همسایگی مبتنی بر مسیر بحرانی استفاده شده است. در شکل ۱ ساختار الگوریتم پیشنهادی نمایش داده شده است.

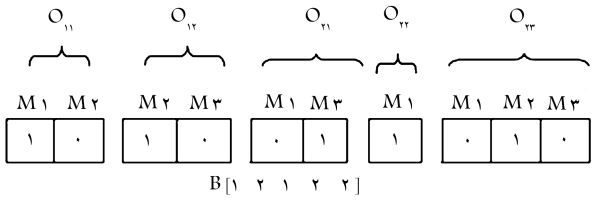
چنان که مشاهده می‌شود، مراحل اصلی الگوریتم کاملاً مشابه الگوریتم NSGAII است، با این تفاوت که در مرحله‌ی تقاطع برای هر جفت والد با تولید یک عدد تصادفی بین ۰ و ۱ معین می‌شود که تقاطع روی قسمت توالی والد انجام شود یا روی تخصیص. در ادامه نیز براساس تقاطع انجام شده، الگوریتم جست‌وجوی همسایگی روی کورومزم جدید اجرا می‌شود.

تنظیم و طراحی مناسب اجزا برای هر الگوریتم از اهمیت بالایی برخوردار است. در الگوریتم پیشنهادی نیز اجزائی نظیر نحوه‌ی رمزگذاری، نحوه‌ی تولید جمعیت اولیه، و نحوه‌ی عملکرد تقاطع و جهش، از اهمیت بالائی برخوردارند. در ادامه به ساختار این اجزاء می‌پردازیم.

در رمزگذاری از ساختار هو[۱۲] استفاده شده است. این ساختار دارای دو بخش تخصیص و توالی است. در بخش تخصیص، هر عملیات به تعداد ماشین‌هایی که



شکل ۱. ساختار الگوریتم پیشنهادی.



شکل ۲. ساختار رمزگذاری ارائه شده توسط هو.

می‌توانند آن عملیات را انجام دهند دارای بردار صفر و ۱ است؛ ۱ به‌معنی انجام این عملیات توسط آن ماشین است. در بخش توالی، هرکار به تعداد عملیات آن دارای تکرار جایگشتی است. در شکل ۲ نمونه‌یی از این ساختار آورده شده است.

به‌منظور تولید جمعیت اولیه از دو روش تصادفی و همچنین الگوریتم ابتکاری هو[۱۲] استفاده می‌شود. در این روش با استفاده از قواعد توزیع برای هر جایگشت تصادفی کار، یک راه‌حل امکان‌پذیر ایجاد می‌شود.

۱.۴. الگوریتم تقاطع و جهش

به‌منظور تطبیق رویکرد جست‌وجوی شناور با الگوریتم NSGAII، از روش‌های ابتکاری در تقاطع استفاده می‌شود (شکل ۱). در این روش دو الگوریتم جست‌وجوی

همسایگی ابتکاری (الگوریتم‌های جست‌وجوی همسایگی توالی و تخصیص) برای بهبود تقاطع براساس جست‌وجوی مسیر بحرانی طراحی شده است. جست‌وجوی مسیر بحرانی رویه‌یی است که چگونگی طراحی جست‌وجوی همسایگی مؤثر را پیرامون راه‌حل ایجاد شده تعیین می‌کند. مسیر بحرانی اولین‌بار در مسئله‌ی تولید کارگاهی[۲۰] مطرح شد. مسیر بحرانی روی مسئله از عملیات با زمان اتمام برابر با دوره‌ی ساخت آغاز و هر بار روی عملیاتی که زمان اتمام آنها برابر با زمان شروع عملیات موجود در مسیر بحرانی است ادامه می‌یابد تا به ابتدا برسد. مراحل اجرایی این الگوریتم عبارت‌اند از: عدد تصادفی $r \in (۰, ۱)$ را ایجاد کنید.

- اگر $r \prec ۰/۵$ عملگر تقاطع را روی قسمت تخصیص کنید.
- الگوریتم جست‌وجوی همسایگی توالی را اجرا کنید. اگر راه‌حل بهبود یافت آنگاه جایگزین راه‌حل اول شود، در غیر این‌صورت راه‌حل اولیه ارائه شود.
- اگر $r \succ ۰/۵$ عملگر تقاطع را روی قسمت توالی اجرا کنید.
- الگوریتم جست‌وجوی همسایگی تخصیص اجرا شود. اگر راه‌حل بهبود یافت آنگاه جایگزین راه‌حل اول شود، در غیر این‌صورت راه‌حل اولیه ارائه شود.

۲.۴. الگوریتم جست‌وجوی همسایگی توالی

گام ۱. مسیر بحرانی را برای جواب به دست آمده تعیین کنید و عملیات را به‌ترتیب بحرانی عمومی، بحرانی و غیر بحرانی مرتب کنید. عملیات بحرانی عمومی عملیاتی است که روی چند مسیر بحرانی قرار دارد.

گام ۲. فهرستی از ماشین‌های قابل انجام برای تمامی عملیات تشکیل دهید، و ماشین تخصیص‌یافته برای هر یک از عملیات مشخص شود.

گام ۳. برای عملیات مسیر بحرانی موجود در فهرست، ماشینی را که زمان انجام آن نسبت به قبلی کم‌ترین است انتخاب (در صورتی که چند انتخاب بود ماشین با کم‌ترین حجم کاری انتخاب شود) و تعویض شود. در صورتی که چنین ماشینی نبود، عملیات بحرانی بعدی بررسی شود.

گام ۴. اگر تمام فهرست بررسی شد و تغییری وجود نداشت، توقف کنید.

۳.۴. الگوریتم جست‌وجوی همسایگی تخصیص

گام ۱. مسیر(های) بحرانی برای راه‌حل مورد نظر را بیابید.

گام ۲. ماشین‌ها را برحسب دوره‌ی ساخت هر یک به‌طور نزولی مرتب کنید.

گام ۳. اولین ماشین در فهرست را انتخاب و دو عملیات بحرانی غیر متعلق به یک کار را روی آن جابه‌جا کنید. در صورت غیر ممکن بودن، ماشین بعدی بررسی شود.

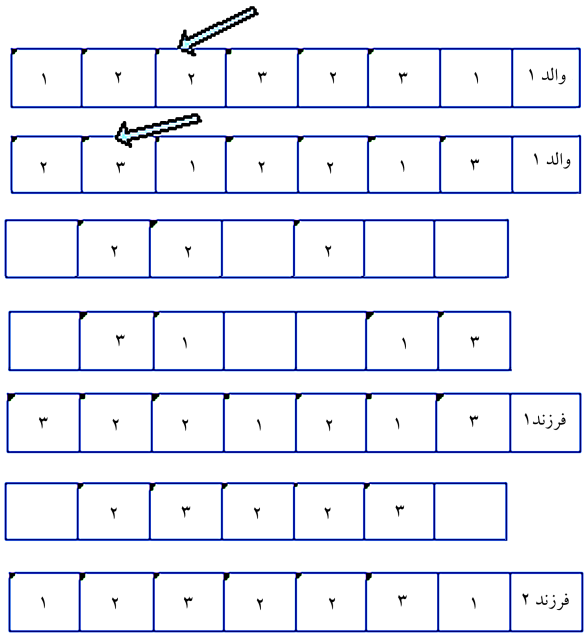
گام ۴. پایان.

به‌منظور تقاطع روی زیرکد تخصیص از تقاطع دونقطه‌یی استفاده می‌شود. تقاطع روی زیرکد توالی مطابق گام‌های زیر است:

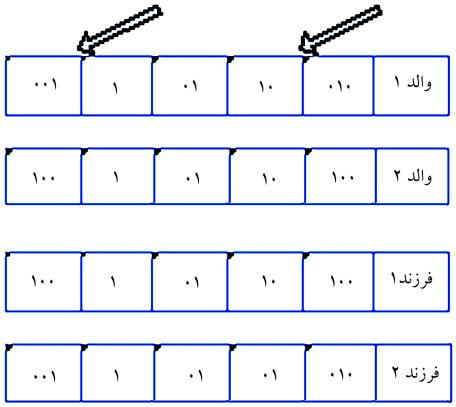
۱. یک کار انتخاب و کلیه عملیات‌های مربوط به کار انتخاب شده از والد ۲ حذف شود.

۲. عملیات باقی‌مانده از والد ۲ در جاهای خالی شده چنان قرار داده شود که ترتیب‌شان به‌هم نخورد.

۳. گام ۱ و ۲ برای والد ۱ تکرار شود.



شکل ۳. مثالی از تقاطع روی زیرکد توالی.



شکل ۴. مثالی از تقاطع روی زیرکد تخصیص.

در شکل‌های ۳ و ۴ عملگرهای تقاطع تخصیص و توالی روی یک مثال نمایش داده شده است.

برای جهش از ساختاری انعطاف‌پذیر و وابسته به تعداد تکرار الگوریتم استفاده می‌شود. استفاده از تابع احتمال وابسته به تعداد دفعات تولید نسل می‌تواند هم تنوع را افزایش دهد و هم از تغییرپذیری، به‌خصوص در مراحل نهایی اجرای الگوریتم، بکاهد. تابع احتمال جهش عبارت است از: $p = \frac{۱}{nn}$ (nn شماره‌ی نسل جاری است)؛ بدین‌ترتیب افزایش شماره‌ی نسل موجب کاهش احتمال جهش می‌شود.

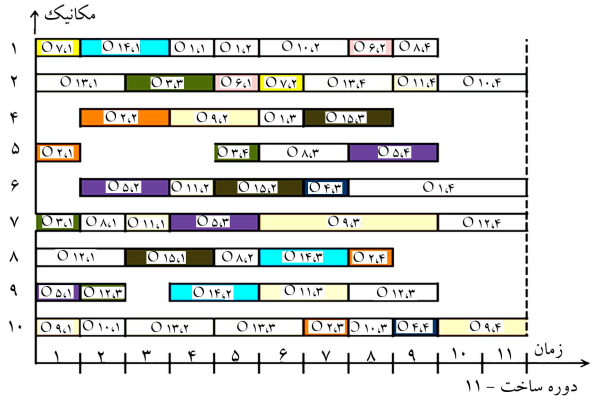
در عملگر جهش تخصیص، دو عملیات تصادفی انتخاب می‌شود و ماشین‌های دیگری از فهرست قابل انجام به آنها تخصیص می‌یابد. گام‌های زیر عملگر جهش توالی را نشان می‌دهد: دو عدد تصادفی $r_۱$ و $r_۲$ بین $۲ \leq r_۱ \leq r_۲ \leq m - ۱$ (m تعداد کل عملیات) تولید شود. عملیات مکان $r_۱$ و $r_۲$ با هم جابه‌جا شود. در صورتی که در موقعیت‌های $r_۱$ و $r_۲$ کار مشابه بود، دو عدد تصادفی دیگر انتخاب شود.

۵. نتایج محاسباتی

در این قسمت، نتایج محاسباتی حاصل از اجرای الگوریتم آورده شده است. برای بررسی عملکرد الگوریتم از دو دسته مسئله استفاده شده است: ۱. مسائل نمونه‌ی کایسم؛[۸] ۲. مسائل تولیدشده به‌طور تصادفی. از مسائل دسته‌ی اول برای مقایسه‌ی عملکرد الگوریتم با سایر روش‌ها استفاده شده است، و دسته‌ی دوم مسائل به‌منظور مقایسه‌ی عملکرد الگوریتم با رویکردهای سلسله‌مراتبی و یک‌پارچه در حل به روش الگوریتم NSGAII بررسی شده است. الگوریتم با استفاده از نرم‌افزار MATLAB کدنویسی شده و روی رایانه‌یی با مشخصات پردازنده ۲٫۲Ghz و حافظه‌ی ۴GB اجرا شده است. شرایط اولیه‌ی اجرای الگوریتم را پس از چند اجرای آزمایشی، به‌صورت: تعداد اعضای جمعیت برابر ۱۰۰، روش ایجاد جمعیت اولیه تصادفی (۴۰٪) و نیز روش CDR (۶۰٪)،[۱۲] و تعداد تولید نسل برابر ۵۰۰ تعیین شده است.

برای بررسی نتایج محاسباتی ابتدا الگوریتم روی مسئله‌یی با ۸ کار و ۸ ماشین[۸] اجرا شده است. نتایج محاسباتی تفصیلی حاصل از این اجرا در جدول ۱ خلاصه شده است. منظور از میانگین اعضای رتبه ۱، متوسط تعداد عضوهای پارتو در هر شماره نسل طی ۲۰ بار اجرای متوالی الگوریتم است. چنان‌که از این نتایج برمی‌آید الگوریتم به‌طور متوسط در طی ۲۰ اجرای انجام‌شده در نسل ۴۵۰ به همگرایی می‌رسد. جواب‌های پارتو این نسل تا انتهای نسل ثبات دارد که علت آن کاهش تغییرپذیری الگوریتم است. از طرفی واریانس جواب‌ها بین راه‌حل‌های پارتو تقریباً دارای مقدار کاهشی است که علت آن تنوع الگوریتم در ابتدای اجراست. مشاهده می‌شود که به‌تدریج با افزایش شماره‌ی نسل، تعداد جواب‌هایی که از رتبه‌ی ۱ حذف می‌شوند کاهش می‌یابد که این نشان از کاهش تغییرپذیری با افزایش شماره‌ی نسل است.

همچنین مجموعه جواب پارتو به دست آمده از اجرای الگوریتم برای مسائل نمونه‌ی کایسم[۸] با نتایج به دست آمده از الگوریتم‌هایی که اخیراً ارائه شده، برای حل مسائلی همچون Al+CGA،[۸] PSO+SA،[۱۵] PSO+TS[۹] و P-DABC[۲۱] مقایسه شده و نتایج آن در جدول ۲ ارائه شده است. سه مسئله‌ی کایسم با ابعاد ۸ * ۸، ۱۰ * ۱۰ و ۱۵ * ۱۰ در این جدول بررسی شده است. داده‌های این مسائل براساس اجراهای برگرفته از مراجع برای سه هدف دوره ساخت (MS)، مقدار کلی بارکاری (TW) و بیشینه بارکاری (W) آورده شده است. مشاهده‌می‌شود که الگوریتم NSGAII با رویکرد جست‌وجوی شناور (الگوریتم پیشنهادی) در مقایسه با سایر روش‌ها نتایج نسبتاً بهتری را ارائه داده است. برای مسئله‌ی ۸ * ۸ تابع هدف بیشینه حجم کاری بهبود یافته، و نسبت به روش‌های دیگر به مقدار کم‌تری رسیده است. برای مسئله‌ی ۱۰ * ۱۰ همانطورکه مشخص است جواب‌های پارتو ارائه شده بر برخی از جواب‌های ارائه شده در مطالعات پیشین منطبق است. همچنین در مسئله‌ی ۱۵ * ۱۰ علاوه بر بهبود تابع هدف بیشینه حجم کاری، جواب پارتوی دوم (۱۱ و ۹۱ و ۱۱) ارائه شده بر کلیه‌ی جواب‌های پارتو دیگر روش‌ها منطبق است. نمودار اجرایی این راه‌حل در شکل ۵ نمایش داده شده است. علاوه بر مسائل نمونه‌ی کایسم با طراحی مسائل تصادفی M۰۱ الی M۰۶ با ابعاد مختلف، به سه روش رویکرد جست‌وجوی شناور، سلسله‌مراتبی و یک‌پارچه الگوریتم NSGAII اجرا، و نتایج با هم مقایسه شده است.



شکل ۵. نمودار اجرایی راه حل دوم مسئله‌ی ۱۵ * ۱۰.

جدول ۱. نتایج تفصیلی حاصل از اجراهای مختلف الگوریتم براساس شماره نسل در مسئله‌ی ۸*۸.

شماره نسل	میانگین اعضای رتبه ۱	میانگین راه‌حل‌های ردشده از رتبه ۱	دوره ساخت رتبه ۱		بیشینه حجم کاری رتبه ۱		مقدار کل حجم کاری رتبه ۱	
			میانگین	واریانس	میانگین	واریانس	میانگین	واریانس
۵	۱۰	۶	۱۶	۰٫۷	۱۲	۰٫۶	۷۵	۰٫۹
۱۵	۱۰	۶	۱۶	۰٫۷	۱۲	۰٫۶	۷۵	۰٫۹
۴۰	۱۰	۶	۱۶	۰٫۷	۱۲	۰٫۶	۷۵	۰٫۹
۶۰	۷	۶	۱۶	۰٫۷	۱۲	۰٫۶	۷۵	۰٫۹
۱۰۰	۷	۷	۱۵	۰٫۵	۱۲	۰٫۶	۷۴	۰٫۸۵
۱۴۰	۷	۵	۱۵	۰٫۵	۱۲	۰٫۶	۷۴	۰٫۸۵
۱۹۰	۵	۳	۱۵	۰٫۵	۱۱٫۵	۰٫۵	۷۴	۰٫۸۵
۲۳۰	۵	۲	۱۶	۰٫۷	۱۱٫۵	۰٫۵	۷۳	۰٫۸۵
۲۹۰	۵	۲	۱۶	۰٫۷	۱۱٫۵	۰٫۵	۷۳	۰٫۸۵
۳۴۰	۵	۱	۱۵	۰٫۷	۱۲	۰٫۶	۷۳	۰٫۸۵
۴۵۰	۵	۰	۱۴٫۵	۰٫۵	۱۲	۰٫۶	۷۳	۰٫۷۵
۵۵۰	۵	۰	۱۴٫۵	۰٫۵	۱۱٫۵	۰٫۵۵	۷۳	۰٫۷۵
۶۷۰	۵	۰	۱۴٫۵	۰٫۵	۱۱٫۲	۰٫۴	۷۳	۰٫۷۵
۷۵۰	۵	۰	۱۴٫۵	۰٫۵	۱۱٫۲	۰٫۴	۷۳	۰٫۷۵
۸۰۰	۵	۰	۱۴٫۵	۰٫۵	۱۱٫۲	۰٫۴	۷۳	۰٫۷۵

جدول ۲. مقایسه نتایج حاصل از اجرای الگوریتم پیشنهادی با سایر روش‌ها.

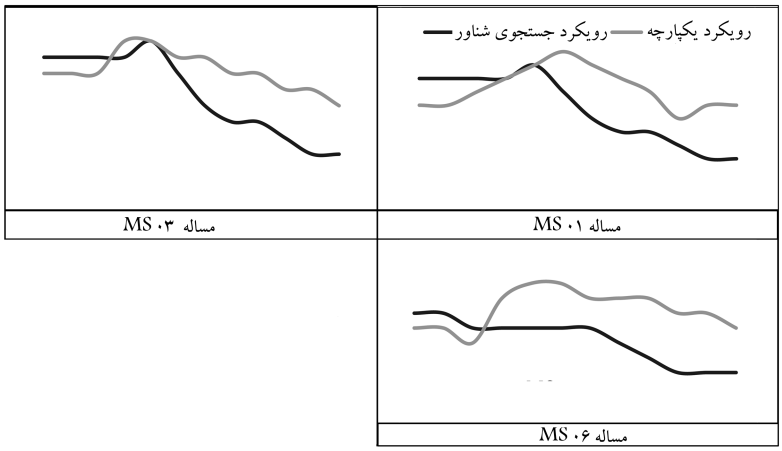
اندازه مسئله	تابع هدف	AL+CGA			PSO+SA		PSO+TS		P-DABC			الگوریتم پیشنهادی		
۸ * ۸	MS	۱۴	۱۵	۱۶	۱۵	۱۶	۱۴	۱۵	۱۴	۱۵	۱۶	۱۴	۱۵	۱۶
	TW	۷۷	۷۹	۷۵	۷۵	۷۳	۷۷	۷۵	۷۷	۷۵	۷۳	۷۷	۷۲	۷۷
	W	۱۲	۱۳	۱۳	۱۲	۱۳	۱۲	۱۲	۱۲	۱۲	۱۳	۱۲	۱۱	۱۱
۱۰ * ۱۰	MS	۷	۷		۷		۷		۸	۷	۸	۷	۸	۸
	TW	۴۳	۴۵		۴۴		۴۳		۴۱	۴۳	۴۲	۴۲	۴۲	۴۱
	W	۵	۵		۶		۶		۷	۵	۵	۶	۵	۷
۱۰ * ۱۵	MS	۱۱	۲۳	۲۴	۱۲		۱۱		۱۲	۱۱		۱۱	۱۱	
	TW	۹۳	۹۵	۹۱	۹۱		۹۳		۹۱	۹۳		۹۳	۹۱	
	W	۱۱	۱۱	۱۱	۱۱		۱۱		۱۱	۱۱		۱۰	۱۱	

جدول ۳. مقایسه‌ی نتایج حاصل از اجرای الگوریتم پیشنهادی با سه رویکرد سلسله‌مراتبی، یک‌پارچه و جست‌وجوی شناور.

شماره مسئله	ابعاد مسئله	رویکرد سلسله‌مراتبی (۱)		رویکرد یک‌پارچه (۲)		رویکرد جست‌وجوی شناور (۳)	
		تعداد جواب پارتو نهایی	تعداد جواب غیر مسلط نسبت به هر سه رویکرد	تعداد جواب پارتو نهایی	تعداد جواب غیر مسلط نسبت به هر سه رویکرد	تعداد جواب پارتو نهایی	تعداد جواب غیر مسلط نسبت به هر سه رویکرد
M۰۱	۴ * ۵	۶	۰	۶	۶ نسبت به (۱) ۵ نسبت به کل	۶	۶ نسبت به کل
M۰۲	۵ * ۶	۵	۰	۶	۶ نسبت به (۱) ۲ نسبت به کل	۶	۶ نسبت به کل
M۰۳	۷ * ۷	۵	۰	۷	۷ نسبت به (۱) ۴ نسبت به کل	۷	۷ نسبت به کل
M۰۴	۱۰ * ۱۰	۶	۰	۶	۶ نسبت به کل	۷	۷ نسبت به کل۲
M۰۵	۱۰ * ۱۲	۵	۰	۶	۶ نسبت به (۱) ۳ نسبت به کل	۷	۷ نسبت به کل
M۰۶	۱۲ * ۱۵	۷	۰	۸	۸ نسبت به (۱) ۳ نسبت به کل	۸	۸ نسبت به کل

برای حالت سلسله‌مراتبی ابتدا زیرمسئله‌ی تخصیص حل، و سپس زیرمسئله‌ی توالی بررسی می‌شود. در این حالت از تقاطع دو نقطه‌یی استفاده شده است. برای حالت یک‌پارچه نیز از همان تقاطع دونقطه‌یی استفاده شده است و تقاطع به‌صورت احتمالی بین تخصیص یا توالی رخ داده است. شرایط اولیه‌ی اجرای الگوریتم برای هر سه حالت یکسان و مانند بخش قبلی است. روش تولید جمعیت اولیه نیز مانند الگوریتم پیشنهادی است. برای مقایسه‌ی این سه رویکرد، معیارهای کیفیت جواب‌های پارتو و میزان تغییرپذیری مسیر رسیدن به جواب‌های نهایی بررسی شده و نتایج این اجراها در جدول ۳ ارائه شده است. چنان که مشاهده می‌شود، رویکرد جست‌وجوی شناور، نسبت به دو رویکرد دیگر، دارای تعداد جواب‌های پارتوی بیشتری است. برای تعیین کیفیت جواب‌های پارتو، کلیه جواب‌های پارتو نهایی تعیین شده توسط هر سه رویکرد برای هر مسئله را به‌صورت دو به دو در یک مجموعه قرار داده و جواب‌های غیرمسلط تعیین می‌شود. همانطورکه در جدول ۳ مشخص است، کلیه جواب‌های پارتو حاصل از اجرای الگوریتم با رویکرد سلسله‌مراتبی بر جواب‌های پارتوی یک‌پارچه و جست‌وجوی شناور منطبق است. همچنین تعدادی از جواب‌های پارتو رویکرد یک‌پارچه نیز بر رویکرد جست‌وجوی شناور منطبق است. این امر نشان‌گر آن است که کیفیت بهتر جواب‌های حاصل از اجرای الگوریتم با رویکرد جست‌وجوی شناور است.

علاوه بر جدول ۳، تعداد جواب‌های رد شده از مجموعه پارتو حین اجرای الگوریتم روی سه مسئله‌ی تصادفی انتخابی MS۰۱، MS۰۳ و MS۰۶ با دو رویکرد جست‌وجوی شناور و یک‌پارچه اجرا و مقایسه شده است. همانطورکه در نمودارهای شکل ۶ نمایش داده شده، برای هر سه مسئله که تحت شرایط مشابه اجرا شده‌اند، در ابتدا تعداد جواب‌های رد شده از مجموعه جواب پارتو بیشتر از



شکل ۶. مقایسه نتایج حاصل از اجرای الگوریتم تطبیق با دو رویکرد یکپارچه و جستجوی شناور

حالت جست‌وجوی یک‌پارچه است و این به‌علت استفاده از روش متفاوت تولید راه‌حل و نیز استفاده از تابع احتمال جهش است که مقدار آن در ابتدا بالاست. از طرفی به‌تدریج این تعداد کاهش یافته و مقدار آن کم‌تر از حالت جست‌وجوی یک‌پارچه می‌شود؛ زیرا رویکرد جست‌وجوی شناور مسئله را در فضای واقعی برده و در آنجا به جست‌وجوی نقاط می‌پردازد. این ثبات و عدم پرش‌های غیر تصادفی به فضاهای تخصیص و توالی، منجر به کاهش تعداد جواب‌های رد شده‌ی مجموعه‌ی پارتو می‌شود.

۶. نتیجه‌گیری

در این نوشتار مسئله‌ی تولید کارگاهی انعطاف‌پذیر چندهدفه مورد بررسی قرار گرفت و رویکردی جدید به‌منظور بهینه‌سازی فضای جست‌وجو، با عنوان جست‌وجوی شناور، برای حل مسئله ارائه شد. در این رویکرد علاوه بر جست‌وجوی وسیع‌تر در فضای جواب (برخلاف رویکرد سلسله‌مراتبی) از سختی مسئله و نیز تغییرپذیری مسیر رسیدن به جواب نهایی (برخلاف رویکرد یک‌پارچه) کاسته است. این رویکرد از الگوریتم مبتنی بر جمعیت و روش‌های پارتو NSGAII کمک می‌گیرد. نتایج تجربی حاکی از کارا بودن الگوریتم -- چه در کاهش تغییرپذیری و چه در بهبود جواب -- هستند. مسائل نمونه‌ی کایسم برای بررسی روی روش ارائه شده انتخاب شده و با برخی روش‌های ارائه‌شده توسط محققین مقایسه شد. علاوه بر آن با طراحی مسائل تصادفی، کارکرد این رویکرد با رویکردهای سلسله‌مراتبی و یک‌پارچه مقایسه شد. براساس نتایج به دست آمده مشخص شد که رویکرد جست‌وجوی شناور نسبت به دیگر روش‌ها از کارایی بیشتری برخوردار است. انطباق رویکرد جست‌وجوی شناور با سایر الگوریتم‌های مبتنی بر جمعیت (مانند الگوریتم مورچگان) و نیز توسعه‌ی مجموعه جواب پارتو از جمله کارهایی است که می‌توان برای توسعه‌ی روش ارائه شده دنبال کرد.

پانوشت‌ها

1. non-dominated sorting genetic algorithm II
2. dispatching rules
3. particle swarm optimization
4. simulated annealing

منابع (References)

1. Garey, M.R., Johnson, D.S. and Sethi, R. "The complexity of flow shop and job shop scheduling", *Mathematics of Operations Research*, **1**, pp. 117-129 (1976).
2. Paulli, J. "A hierarchical approach for the FMS scheduling problem", *European Journal of Operational Research*, **86**(1), pp. 32-42 (1995).
3. Saidi-Mehrabad, M. and Fattahi, P. "Flexible job shop scheduling with tabu search algorithms", *International Journal of Advanced Manufacturing Technology*, **32**, pp. 563-570 (2007).
4. Hurink, E., Jurisch, B., and Thole, M. "Tabu search for the job shop scheduling problem with multi-purpose machines", *Operations Research Spectrum*, **15**, pp. 205-215 (1994).
5. Dauzère-Pèrès, S. and Paulli, J. "An integrated approach for modeling andsolving the general multi-processor job-shop scheduling problem using tabusearch", *Annals of Operations Research*, **70**, pp. 281-306 (1997).
6. Mesghouni, K., Hammadi, S. and Borne, P. "Evolution programs for job-shop scheduling", *IEEE International Conference on Systems*, Man, and Cybernetics (SMC '97) (1997).

7. Ihlow, C.J. and Lehman, C. "A genetic algorithm for flexible job-shop scheduling", *Proceeding of the 1999 IEEE International Conference on Robotics & Automation* (1999).

8. Kacem, I., Hammad, S. and Bome, P. "Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems", Systems, Man and Cybernetics, Part C, IEEE Transactions on (2002).

9. Zhang, and Gen, M. "Multistage-based genetic algorithm for flexible job-shop scheduling problem", *Journal of Complexity International*, **11** , pp. 223 (2005).

10. Saidi-Mehrabad, M. and Fattahi, P. "Flexible job shop scheduling with tabu search algorithms", *International Journal of Advanced Manufacturing Technology*, **32**, pp. 563-570 (2007).

11. Gao, J., Gen, M. and Sun, L. "A hybrid of genetic algorithm and bottleneck shifting for flexible job shop scheduling problem", *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation* (2006).

12. Ho, N.B., Tay, J.C. and Lai, E.M.K. "An effective architecture for learning andevolving flexible job-shop schedules", *European Journal of Operational Research*, **179**, pp. 316-333 (2007).

13. Unachak, P. "An adaptive represention for a genetic algorithm in solving flexible job-shop scheduling and rescheduling problem", Michigan State University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philsophy Computer Science (2010).

14. Baykasoglu, A., Ozbakir, L. and Ihsanzonmez, A. "Using multiple objective tabu search and grummers to model and solve multi-objective flexible job shop scheduling problems", *Journal of Intelligent Manufacturing*, **15**, pp. 777-785 (2004).

15. Xia, W. and Wu, Z. "An effective hybrid optimization approach for multi-objective ?exible job-shopscheduling problem", *Computers & Industrial Engineering,* **48**(2), pp. 409-425 (2005).

16. Tay, J. and Ho, N. "Evolving dispatching rules using genetic programming for solving multi-objective flexible job shop problems", *Computers & Industrial Engineering*, **54**, pp. 453-473 (2008) .

17. Morad, I.E., Fatemi Ghomi, S.M.T. and Zandieh, M. "Bi-objective optimization research on integrated fixed time interval preventive maintenance and production for scheduling flexible job-shop problem", Expert Systems with Application **38**, pp.7169-7178 (2010)

18. Frutos, M. Olivera, A. and Tohme, F. "A memetic algorithm based on a NSGA II scheme for the flexible job-shop scheduling problem", *Ann Operation Research*, **181**, pp. 745-765 (2010)

19. Deb, K., *Multi-Objecive Optimization Using Evolutionary Algortihms*, Indian Institute of Technology, Kanpur, India (2001).

20. Adams, J., Balas, E. and Zawack, D. "The shifting bottleneck procedure for job shop scheduling", *Management Science*, **34**(3), pp. 391-401.

21. Li, J., Pan, Q. and Gao, K. "Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems", *International Journal of Manufacturing Technology,* **55**, pp. 1159-1169 (2011).