

توسعه دو الگوریتم چند هدفه برای حل مسئله چند هدفه زمانبندی کارگاهی منعطف با در نظر گرفتن توان مصرفی ماهانه

سید حبیب اله رحمتی*
مصطفی زندیه**

چکیده

در این مقاله، به منظور واقعی تر ساختن مسئله زمانبندی کار کارگاهی منعطف (FJSP) و نزدیک تر نمودن آن با مسائل دنیای واقعی، یک عامل عملیاتی به مدل کلاسیک این مسئله افزوده می‌شود. این عامل که بهینه سازی میزان توان الکتریکی مصرفی در طول یک ماه می‌باشد، کلیدی ترین عامل در محاسبات برق مصرفی شرکت‌های صنعتی به شمار می‌آید. دیدن این عامل در مدل سازی، با توجه به شرایط جدید کشور بعد از حذف یارانه‌ها اهمیت پر رنگتری نیز پیدا کرده است. در کنار این هدف، ۲ هدف مرسوم نیز که زمان تکمیل کارها و بار کاری ماشین بحرانی می‌باشند نیز در نظر گرفته شده اند. به منظور حل مدل چند هدفه، دو الگوریتم به نام‌های الگوریتم چند هدفه مبتنی بر جغرافیای زیستی (MOBBO^۲) و الگوریتم چند هدفه جستجوی هارمونی (MOHS^۳) را ارائه خواهیم کرد. این دو الگوریتم برای اولین بار است که به محیط گسسته مسائل زمانبندی معرفی می‌شوند. در پایان به وسیله توسعه چند مسئله معروف از مدل مربوطه، عملکرد الگوریتم‌های ارائه شده را به صورت آماری مقایسه خواهیم نمود. کلید واژگان: کار کارگاهی منعطف، توان مصرفی الکتریکی، الگوریتم بهینه‌سازی، جستجوی هارمونی

* دانشجوی دکترا، مهندسی صنایع، دانشکده مهندسی صنایع، دانشگاه صنعتی امیرکبیر، تهران، ایران (نویسنده مسئول)
sd_rahmati@atu.ac.ir

** عضو هیأت علمی دانشکده مدیریت و حسابداری، دانشگاه شهید بهشتی، تهران، ایران

1- Flexible job shop scheduling problem (FJSP)
2- Multi-objective biogeography based optimization
3- Multi-objective harmony search

تاریخ پذیرش: ۹۱/۱۰/۸

تاریخ دریافت: ۹۱/۲/۳۰

مقدمه

مسئله کار کارگاهی منعطف (FJSP) به عنوان شاخه ای از مسائل برنامه ریزی تولید بوده و مدلی توسعه یافته از مدل کار کارگاهی (JSP^۱) می باشد [۴]. در محیط JSP، هر عمل میتواند توسط یک ماشین و با توالی از پیش تعیین شده ای انجام شوند. درحالی که در FJSP فرض اول برداشته شده و برای هر عمل مجموعه ای از ماشین های قابل^۲، ارائه می شوند. در اثر این فرض جدید، در FJSP علاوه بر تعیین توالی کارها با مسئله دیگری نیز مواجه هستیم و آن تخصیص ماشینی قابل برای هر عمل است. در نتیجه می توان گفت پیچیدگی FJSP به مراتب از پیچیدگی JSP بیشتر است. به همین منظور به دلیل آنکه مسئله JSP به کلاس مسائل NP-Hard تعلق دارد، قابل اثبات است که FJSP نیز به این دسته از پیچیدگی مسائل تعلق داشته باشد [۳]. با توجه به مطالب گفته شده، در FJSP با دو زیر مسئله مواجه هستیم: تعیین توالی کارها و تخصیص ماشین به هر عمل (از مجموعه ماشین های قابل برای انجام آن عمل).

در ادبیات تحقیق، اکثر مقالات به ارائه روش های حل برای مسئله مذکور پرداخته اند. در این میان، برخی از مطالعات، این دو زیر مسئله را به صورت جداگانه بهینه سازی کرده اند. این رویکرد به رویکرد سلسله مراتبی معروف است و در آن یک مسئله پیچیده به دو زیر مسئله ساده تر تقسیم می شود. بردندی مارت [۳] در رویکرد سلسله مراتبی خود زیر مسئله تعیین توالی را بوسیله قوانین توزیع و زیر مسئله تخصیص را به وسیله الگوریتم جستجوی ممنوع (TS) حل کرد. بارتز و چمبر [۱] زیر مسائل خود را بوسیله الگوریتم TS حل کردند. ایکسیا و وو [۲۶] از الگوریتم تبرید شیبه سازی شده (SA) برای زیر مسئله توالی و از الگوریتم بهینه سازی دسته ای ذرات (PSO) برای تخصیص استفاده کردند. از سوی دیگر، اکثر محققان، این دو زیر مسئله را با هم در نظر گرفتند. این رویکرد که به رویکرد یکپارچه معروف است، معمولاً از رویکرد سلسله مراتبی سخت تر است ولی معمولاً جواب های بهتری را ارائه می کند. هیورینک و همکاران [۱۳] در رویکرد یکپارچه شان از الگوریتم TS استفاده نمودند. چن و همکاران [۵] الگوریتم ژنتیکی ارائه کردند که در آن از یک کروموزوم ۲ رشته ای

1- Job shop scheduling problem
2- Capable

استفاده شده بود. یکی از رشته‌های این کروموزوم برای نمایش تخصیص و دیگری برای نمایش توالی به کار گرفته شده بود. ماسترولی لی و گامبرلا [۱۸] از یک الگوریتم TS با دو ساختار همسایگی استفاده کردند. در ادامه کاسم و همکاران [۱۴] مسئله FJSP را در دو حالت تک هدفه و چند هدفه بررسی کردند. حالت چند هدفه FJSP همچنین توسط گائو و همکاران [۱۰] نیز بررسی شد. یزدانی و همکاران [۲۷] الگوریتم جستجوی متغیر همسایگی (VNS^۱) را ارائه کردند. اخیراً نیز رحمتی و زندیه [۱۹] الگوریتم مبتنی بر جغرافیای زیستی (BBO) را برای این مسئله، در حالت تک هدفه، ارائه نمودند.

مقاله‌ای هم به توسعه مدل این مسئله پرداخته اند. ترابی و همکاران [۲۴] در یک افق محدود، یک مدل عدد صحیح آمیخته^۲ برای مسئله FJSP با در نظر گیری زمانبندی چرخه میزان بهره اقتصادی توسعه دادند. فتاحی و همکاران [۸] به منظور حل موثر تر FJSP در سایزهای کوچک یک مدل جدید و به منظور حل آن در سایزهای بزرگ دو الگوریتم هیورستیک ارائه کردند. آنها همچنین ۶ ساختار جستجوی ترکیبی نیز ارائه نمودند. فتاحی و همکاران [۷] همپوشانی عملها را نیز برای FJSP در نظر گرفتند. وانگ و وو [۲۵] نگهداری و تعمیرات پیشگیرانه را در FJSP در نظر گرفتند. فروتوس و همکاران [۹] تابع هدف هزینه را برای مدل مسئله در نظر گرفته و آن را به وسیله الگوریتم ژنتیک مرتب سازی غیر مغلوب (NSGAII^۳) ترکیب شده با الگوریتم SA حل کردند.

الگوریتم BBO مانند الگوریتم‌های ژنتیک (GA) و PSO، یک الگوریتم نشات گرفته شده از طبیعت است. در این الگوریتم، ایده مهاجرت گونه‌های زیستی که قسمتی از علم جغرافیای زیستی است برای حل مسائل بهینه سازی استفاده می‌شود. الگوریتم BBO برای اولین بار توسط سیمون [۲۲] ارائه شد. سیمون در اولین مقاله خود در این زمینه، تعاریف و مراحل اجرایی الگوریتم BBO را شرح داد. این الگوریتم همچنین عملکرد خوبی را روی مسائل دنیای واقعی مثل ارزیابی و تشخیص منابع آب زیر زمینی [۱۶] و حتی مسائل اقتصادی مثل مسئله توزیع بار اقتصادی [۲]، از خود نشان داده است. در این مقاله نسخه چند هدفه ای از این الگوریتم ارائه خواهد شد. الگوریتم

1- Variable neighborhood search

2- mixed integer nonlinear program (MINLP)

3- Non dominated sorting algorithm (NSGAII)

دیگر مورد استفاده در این مقاله، الگوریتم جستجوی هارمونی^۱ می‌باشد. الگوریتم جستجوی هارمونی (HS) یک الگوریتم تکاملی مبتنی بر عملکرد سازهای موسیقی^۲ است که اولین بار توسط گیم^۳ و همکاران [۱۱] معرفی شد. این الگوریتم با مفهومی ساده و تعداد پارامترهای کم و آسان بر مبنای علم آمار و ریاضیات است [۱۷]. این الگوریتم از آلات موسیقی الهام گرفته و از بداهه‌سرایی سراینده‌گان تقلید می‌نماید. تابع هدف HS تخمین و اندازه‌گیری هارمونی و اثری است که آن موسیقی در فرد می‌گذارد تا حالت مناسبی از هارمونی در شخص ایجاد شود. باید به این نکته اشاره شود که HS فرآیند بداهه‌سرایی کیفی را به فرآیند بهینه‌سازی کمی تبدیل می‌کند. یکی از پیشرفت‌های مهم در مدل‌های برنامه‌ریزی تولید در نظرگیری عوامل عملیاتی است. از این عوامل، می‌توان به میزان توان مصرفی ماشین‌ها اشاره نمود. این عامل از آن جهت اهمیت دارد که میزان هزینه برق مصرفی بر اساس میزان توان مصرفی ماشین‌ها محاسبه می‌شود. در نتیجه بهینه‌سازی توان مصرفی، بهینه‌سازی هزینه مصرفی را نیز به دنبال خواهد داشت. در نظرگیری این عامل با توجه به شرایط جدید اقتصادی کشور بعد از حذف یارانه‌ها اهمیت پر رنگ تری نیز پیدا می‌کند. زیرا هزینه‌های بالای انرژی مصرفی، می‌تواند به راحتی وضعیت واحد صنعتی را به زیر نقطه سر به سر بکشانند. به همین منظور، بهتر است در مسئله بهینه‌سازی مورد مطالعه، عمل‌ها به ماشین‌هایی تخصیص یابند که آنها را با جریان مصرفی کمتری انجام می‌دهند تا در نهایت توان مصرفی کمتری نیز در طول ماه مصرف شود. این مهم، یکی دیگر از توسعه‌هایی است که این مقاله برای اولین بار به ادبیات این مسئله معرفی می‌کند. با توجه به مطالب گفته شده، در این مقاله ابتدا مدل کلاسیک مسئله FJSP را با در نظرگیری عامل میزان توان مصرفی توسعه خواهیم داد و سپس دو الگوریتم چند هدفه شامل MOBSO و MOHS را متناسب با ساختار مسئله مورد مطالعه ایجاد خواهیم نمود. در پایان به منظور ارزیابی الگوریتم‌های توسعه داده شده، ابتدا نمونه مسائل ارائه شده توسط برندی مارت [۳] را برای مدل مربوطه توسعه داده و سپس عملکرد الگوریتم‌ها را با یکدیگر مقایسه خواهیم کرد. همچنین، به منظور افزایش دقت

1- Harmony search
2- Music-inspired
3- Geem

مقیاسات، از یک نوع آزمون آماری غیر پارامتریک به نام من ویتنی^۱ [۱۲] نیز استفاده خواهیم نمود.

سایر بخش‌های این تحقیق به صورت مقابل دسته بندی شده اند. در بخش بعدی حالت کلاسیک مدل FJSP و مدل توسعه داده شده برای آن را بررسی خواهد نمود. در بخش ۳، ابتدا مفاهیم اولیه الگوریتم‌های چند هدفه ارائه خواهد شد و سپس به توضیح الگوریتم‌های چند هدفه پیشنهادی می‌پردازیم. در بخش ۴، ابتدا چند مثال از ادبیات FJSP توسعه داده شده و سپس عملکرد الگوریتم توسعه داده شده، به صورت آماری، مورد ارزیابی قرار خواهد گرفت. در پایان، نتایج نهایی و پیشنهاد مطالعات آتی در بخش ۵ ارائه خواهند شد.

مسئله زمانبندی کار کارگاهی منعطف (FJSP)

این مسئله، مدل زمانبندی است که در آن n کار $(J_i, i \in \{1, 2, \dots, n\})$ باید روی m ماشین $(M_k, k \in \{1, 2, \dots, m\})$ انجام شود. برای هر کار، یک یا چند عمل $(O_{jik}, j \in \{1, 2, \dots, n_i\})$ را می‌توان در نظر گرفت (در این مجموعه n_i نشان دهنده تعداد کل عمل‌های کار i است). برای هر عمل (O_{jik}) ، مجموعه ای از ماشین‌های قابل^۲ (ماشین‌هایی که توانایی انجام آن عمل خاص را دارند) در نظر گرفته می‌شود. بدین معنی که هر عمل (O_{jik}) از کار i ام (J_i) ، توسط یک ماشین (k) از مجموعه مخصوص به خود (M_{ij}) انجام می‌شود. همچنین برای عمل j از کار i ام، P_{jik} و t_{jik} به ترتیب نشان دهنده زمان انجام و زمان شروع عمل O_{jik} روی ماشین k می‌باشند. در نتیجه FJSP به منظور حداقل کردن تابع هدف (و یا توابع هدف) دو مسئله اصلی را دنبال می‌کند، تخصیص ماشین‌ها به عمل‌ها و تعیین توالی عمل‌های تخصیص یافته به هر ماشین [۱۹]. همچنین کمیت‌های زیر نیز در مدل سازی مفهوم جدید استفاده شده اند:

P : توان مصرفی

V : ولتاژ مصرفی

I_{jik} : نشان دهنده جریان مصرفی عمل j از کار i روی ماشین k

1- Mann-Whitney

2- Set of capable machines

x_{jik} : متغیر تخصیص در مدل کلاسیک

I : کل جریان مصرفی برنامه زمان بندی

D : تعداد روزهای کاری در طول ماه، H نشان دهنده تعداد ساعت کاری در طول روز (عدد ۶۰ نیز برای تبدیل ساعت به دقیقه در روابط استفاده خواهد شد)

C_{max} : زمان تکمیل کارها

با توجه به موارد گفته شده مدل این مسئله به صورت زیر قابل ارائه است [۹]:

$$\text{Min } C_{\max} = \max_{n, n_i} (t_{jik} + p_{jik}) \quad (1)$$

$$\text{Min } \text{CWL} = \left\{ \sum_{i=1}^n \sum_{j=1}^{n_i} p_{ijk} x_{ijk} \right\} k : 1, 2, 3, \dots, m \quad (2)$$

$$\text{Min } \text{TCP} = \left\{ \sqrt{3} * V * (\sum_{i \in J} \sum_{j \in n_i} \sum_{k \in M} (x_{jik} I_{jik})) * \sin \varphi * \{D * H * 60 * (1 / C_{\max})\} \right\} \quad (3)$$

Subject to:

$$t_{jik} - t_{sih} \geq p_{sih}; \text{ if } O_{sih} \text{ precedes } O_{jik} \quad \forall i \in J, \forall \{h, k\} \in M \quad (4)$$

$$t_{jik} - t_{sgk} \geq p_{sgk}; \text{ if } O_{sgk} \text{ precedes } O_{jik} \quad \forall \{i, g\} \in J, \forall k \in M \quad (5)$$

$$\sum_k x_{jik} = 1; \quad \forall i \in J, \forall j \in n_i \quad (6)$$

Where:

$$t_{jik} = \max \left\{ t_{(j-1)ih} + p_{(j-1)ih}, t_{sgk} + p_{sgk}, 0 \right\}, t_{jik} \geq 0, \quad \forall i \in J, \forall j \in n_i, \forall k \in M \quad (7)$$

$$\begin{cases} x_{jik} = 0 & \forall i \in J, j \in n_i, k \notin M_{ij} \\ x_{jik} \in \{0, 1\} & \forall i \in J, j \in n_i, k \in M_{ij} \end{cases} \quad (8)$$

$$\forall \{s, j\} \in n_i, \forall \{k, h\} \in M, \forall \{g, i\} \in J$$

در این مدل، رابطه ۱ زمان تکمیل کارها را محاسبه می کند و رابطه ۲، مدل بار کاری ماشین بحرانی را نشان می دهد. این دو هدف از اهداف مرسوم و مورد استفاده در ادبیات FJSP هستند [۱۹]. رابطه ۳ مفهوم جدید مورد نظر مقاله را مدل میکند. این رابطه، در یک نگاه کلی به دنبال حداقل سازی توان مصرفی مورد استفاده واحد صنعتی در طول یک ماه است. این رابطه از دو قسمت اصلی تشکیل شده است که در روابط ۹ و ۱۰ از یکدیگر جدا شده اند. رابطه ۹ نشان دهنده توان مصرفی در طول مدت زمان یک برنامه زمانبندی و رابطه ۱۰ نشان دهنده تعداد برنامه های اجرا شده در ماه می باشد.

$$P = \sqrt{3} * V * (\sum_{i \in J} \sum_{j \in n_i} \sum_{k \in M} (x_{jik} I_{jik})) * \sin \varphi \quad (9)$$

$$D * H * 60 * (1 / C_{\max}) \quad (10)$$

پایه محاسباتی برای رابطه ۹، رابطه ۱۱ می‌باشد [۲۰]. در رابطه ۱۱، I که نشان دهنده کل جریان مصرفی برنامه زمان بندی می‌باشد از رابطه ۱۲ بدست می‌آید.

$$P = \sqrt{3} * V * I * \sin \varphi \quad (11)$$

$$I = \sum_{i \in J} \sum_{j \in n_i} \sum_{k \in M} (x_{jik} I_{jik}) \quad (12)$$

بدین ترتیب از ضرب دو رابطه ۹ و ۱۰، کل توان مصرفی در طول یک ماه بدست می‌آید. حال با شرح مثالی این مفهوم جدید را بیشتر توضیح می‌دهیم. فرض کنیم، در برنامه ای، زمان تکمیل برابر ۱ ساعت ($C_{\max} = 60'$) و توان مصرفی در طول این مدت برابر ۱۰۰ کیلو وات ساعت ($(\sqrt{3} * 360 * (\sum_{i \in J} \sum_{j \in n_i} \sum_{k \in M} (x_{jik} I_{jik})) * \sin(90) = 100)$) باشد. حال با توجه مقادیر در نظر گرفته شده برای تعداد روز در ماه (D) و تعداد ساعات روز کاری (H)، کل توان مصرفی در طول ماه، برابر ۲۰۰۰۰ کیلو وات خواهد بود ($TCP = 100 * 25 * 8 * 60 * (1 / 60)$).

سایر روابط ارائه شده در مدل، فضای موجه مدل بهینه سازی را مدل سازی می‌کنند. رابطه ۴، تضمین می‌کند ۲ عمل متوالی از یک کار مشخص، به طور هم زمان پردازش نشوند. رابطه ۵، بیان می‌کند دو عمل متوالی روی یک ماشین، نباید هم پوشانی زمانی داشته باشند. رابطه ۶ تخصیص کارها به ماشین‌ها را بر عهده دارد. رابطه ۷، زمان شروع هر عمل را برای رسیدن به یک برنامه فعال محاسبه می‌کند و رابطه ۸، شرایط تخصیص پذیری عمل‌ها به ماشین‌ها را بررسی می‌نماید.

روش حل

در این قسمت، الگوریتم BBO چند هدفه (MOBBO) و الگوریتم هارمونی چند هدفه (MOHS) را برای حل مدل توسعه خواهیم داد.

مفاهیم الگوریتم BBO

این الگوریتم، الگوریتمی جدید و نشأت گرفته از طبیعت است که از اصول علم جغرافیای زیستی برای جستجو در فضای جواب استفاده می‌کند. به طور کلی، علم جغرافیای زیستی به عنوان زیر شاخه ای از علم بیولوژی، به مطالعه رفتار گونه‌های زیستی مختلف در طول زمان و مکان می‌پردازد [۲۲]. سایمون اصول این شاخه علمی را

برای حل مسائل بهینه سازی توسعه داد [۲۲]. الگوریتم او همانند الگوریتم‌های GA و PSO یک الگوریتم جمعیتی است. بدین معنی که در آن جمعیتی از جوابها (عضوهای منفرد^۱) برای جستجوی بهترین جوابها استفاده می‌شود.

در الگوریتم ژنتیک، هر کروموزوم به عنوان عضوی منفرد شناخته می‌شد و میزان برازندگی^۲ خود را داشت. به طور مشابه در BBO، هر منطقه زیستی^۳ به عنوان یک عضو منفرد شناخته می‌شود و دارای " شاخص میزان مطلوبیت زندگی^۴ (HSI) " مخصوص به خود است. در این الگوریتم نیز مانند الگوریتم ژنتیک (که جواب با برازندگی بیشتر بهتر بود)، جواب (منطقه زیستی) با HSI بالاتر نشان دهنده جواب خوب می‌باشد. در BBO، خواص از منطقه‌هایی (جواب‌هایی) با HSI بالاتر به منطقه‌هایی با HSI پایینتر مهاجرت می‌کنند. به عبارت دیگر، منطقه‌های با HSI پایین با گرفتن خواص از منطقه‌های با HSI بالا، خود را شبیه به آنها می‌کنند. در این الگوی مهاجرت دو نوع عملگر مهاجرت وجود دارد، مهاجرت خروجی^۵ و مهاجرت ورودی^۶. مهاجرت خروجی برای آن جوابی مطرح می‌شود که HSI بالایی دارد و خواص خود را به اشتراک می‌گذارد و به طور مشابه مهاجرت ورودی برای جوابی مطرح می‌شود که HSI پایینی دارد و خواص را می‌پذیرد. قابل ذکر است هر کدام از این دو نوع مهاجرت نرخ خاص خود را دارد که با نام‌های نرخ ورود (λ_i) و نرخ خروج (μ_j) شناخته می‌شوند. حال، الگوریتم BBO با استفاده از این دو عملگر در اپراتور تقاطع و همچنین با استفاده از اپراتور جهش، به دنبال آن جواب (جوابهایی) است که HSI را حداکثر نمایند.

شروع الگوریتم^۷

عملکرد الگوریتم‌های فرا ابتکاری به طور قابل ملاحظه‌ای به جمعیت اولیه آنها بستگی دارد. برای این مهم، در این مقاله، از روش آقای وانگ استفاده شده است [۲۵]. ابتدا جمعیت توالیها به صورت تصادفی تولید می‌شود و سپس برای هر عمل از هر

1- Individuals
2- Fitness
3- Habitat
4- Habitat suitability index (HSI)
5- Emigration
6- Immigration
7- Initialization

کروموزوم در جمعیت دو ماشین انتخاب می‌شود (از مجموعه ماشین‌های توانا برای انجام عمل انتخابی). سپس اگر یک عدد تصادفی ($\text{Rand} \in [0,1]$) تولید شده، از ۰,۸ کمتر بود ماشینی که زمان پردازش کمتری دارد انتخاب می‌شود، در غیر این صورت ماشین با زمان پردازش بیشتر انتخاب می‌شود.

نمایش جوابها و نحوه رمز گشایی آنها^۱

یک نحوه نمایش معروف برای FJSP یک مجموعه دو برداری است که در آن از یک بردار برای نمایش توالی عملیات و از بردار دیگر برای نمایش ماشین تخصیص یافته شده به هر عمل استفاده می‌شود [۱۹، ۲۵]. برای کد زدایی نیز از روش برنامه زمان بندی فعال استفاده شده است [۱۹].

استراتژی انتخاب^۲

این قدم یکی از مهمترین قدمهای متمایز کننده BBO با سایر الگوریتم‌های مشابه است. در BBO دو نوع استراتژی انتخاب وجود دارد، یکی برای عملگرهای مهاجرت (ورودی و خروجی) و دیگری برای عملگر جهش. تفاوت مهم دیگر، که در بالا هم ذکر شد آن بود که در BBO (بر خلاف GA) اعضاء جمعیت از بین نمی‌روند بلکه در طول تکرارهای مختلف اصلاح می‌شوند. به همین منظور هدف استراتژی انتخاب در این الگوریتم آن است که تعیین کند، آیا منطقه ای خاص اصلاح شود یا نه؟ و بعد اگر قرار شد اصلاح شود خواص جدید خود را از کدام جواب دریافت کند. البته خود واژه اصلاح به دو استراتژی مهاجرت و جهش تقسیم می‌شود که باید در هر تکرار درمورد اجرای هر یک از این دو استراتژی، روی هر جواب تصمیم گیری شود. جزئیات این دو نوع استراتژی در دو زیر بخش زیر ارائه خواهد شد.

استراتژی انتخاب برای عملگرهای مهاجرت

در این قسمت با ۲ تصمیم مواجه هستیم. اول اینکه آیا منطقه ای خاص H_i می‌خواهد تغییر کند یا نه؟ برای تعیین این تصمیم عدد تصادفی تولید شده ($\text{Rand} \in [0,1]$) با میزان ورودی (λ_i) مقایسه می‌شود. تصمیم دوم آنست که تعیین شود

1- Solution representation and decoding

2- Selection strategy

منطقه مستعد تغییر H_i از کدام منطقه (H_j) می خواهد خاصیت پذیرد. برای این منظور از چرخه رولت روی میزانهای خروجی (μ_j) استفاده می شود.

استراتژی انتخاب برای جهش

برای تعیین این استراتژی عددی تصادفی تولید شده ($\text{Rand} \in [0,1]$) را با نرخ جهش (m_i) مقایسه می کنیم. خروجی این قسمت تعیین می کند که آیا منطقه مورد نظر جهش یابد یا نه. لازم به ذکر است مفهوم و نحوه محاسبه مقادیر نرخهای λ_i ، μ_j و m_i در بخش های آتی توضیح داده خواهد شد.

اپراتور مهاجرت^۱

مهاجرت اپراتوری است که برای اصلاح جواب به وسیله سایر جواب ها استفاده می شود. ایده اصلی این اپراتور همان مهاجرت در علم جغرافیای زیستی است که نشان دهنده حرکت گونه ها و خواص زیستی در بین بوم های مختلف زیستی است. در این فرآیند، هر جواب H_i ، با توجه به نرخ ورودش (λ_i) برای دریافت خواص (و گونه) انتخاب شده و در همین راستا جواب H_j ، با توجه به نرخ خروجش (μ_j) برای به اشتراک گذاشتن خواص (و گونه) انتخاب می شود.

نرخ های ورود و خروج بعد از محاسبه HSI قابل محاسبه خواهند بود. روابط ۱۳ و ۱۴ نحوه محاسبه هر یک از این نرخ ها را نشان می دهند. در این روابط، k_i نشان دهنده رتبه هر جواب بعد از مرتب سازی بر اساس HSI ها می باشد. بدیهی است که جواب بهتر رتبه بهتر و در نتیجه k_i بیشتری نیز دارد. بعد از محاسبه این دو نرخ استراتژی های انتخاب مهاجرت همان طور که در زیر بخش قبلی توضیح داده شدند، قابل اجرا هستند.

$$\lambda_i = I(1 - \frac{k_i}{n}) \quad (13)$$

$$\mu_i = E(\frac{k_i}{n}) \quad (14)$$

متغیر دیگری که در مفاهیم مربوط به مهاجرت در علم جغرافیای زیستی وجود دارد، احتمال وجود S گونه زیستی در یک منطقه است. این متغیر از معادله ای شبیه به معادله ارائه شده در رابطه ۱۵ پیروی می کند. علت آن هم این است که برای مدل کردن حرکت گونه ها از زمان t تا زمان $t + \Delta t$ یکی از سه حالت زیر اتفاق می افتد

1- Migration operator

(طبق قواعد مارکوف و فرآیندهای تصادفی) تا در نهایت S گونه در زمان $t + \Delta t$ در منطقه باشد:

در زمان t ، S گونه داریم و این مقدار در طول بازه $t + \Delta t$ تغییر نمی کند. در زمان t ، $S-1$ گونه داریم و در طول بازه $t + \Delta t$ یک مهاجرت ورودی اتفاق می افتد. در زمان t ، $S+1$ گونه داریم و در طول بازه $t + \Delta t$ یک مهاجرت خروجی اتفاق می افتد.

$$P_s(t + \Delta t) = P_s(t)(1 - \lambda_s \Delta t - \mu_s \Delta t) + P_{s-1} \lambda_{s-1} \Delta t + P_{s+1} \mu_{s+1} \Delta t \quad (15)$$

بعد از حل معادله ارائه شده در رابطه ۷ در یک حالت پایدار، P_i به صورت رابطه ۱۶ بدست می آید [۲۲]. در این رابطه v_i تابعی از اندازه جمعیت بوده و توسط رابطه ۱۷ به دست می آید. i' در رابطه ۱۷ عددی است با شرط $\text{ceil}((n+1)/2)$.

$$P_i = \frac{v_i}{\sum_{i=1}^n v_i} \quad (16)$$

$$v_i = \begin{cases} \frac{n!}{(n+1-i)!(i-1)!} & i = 1, 2, 3, \dots, i' \\ v_n + 2 - i & i = i' + 1, \dots, n + 1 \end{cases} \quad (17)$$

همان طور که گفته شد، در BBO ابتدا باید تعیین شود که در هر تکرار آیا لازم است جوابی خاص اصلاح شود یا نه؟ و بعد در صورت نیاز به اصلاح، باید دید جواب مربوطه خاصیت را از کدام جواب بگیرد. شرح این دو تصمیم در بخش استراتژی های انتخاب ارائه شده اند. برای این منظور بعد از محاسبه نرخهای λ_i ، μ_j و انجام اپراتور انتخاب (همان طور که در بخش قبل توضیح داده شد)، اپراتور مهاجرت انجام میشود. البته به دلیل آنکه در نوع نمایش جوابهای استفاده شده در این مقاله، ۲ نوع بردار وجود دارد (بردار توالی و بردار تخصیص)، برای هر کدام، از یک اپراتور تقاطع خاص استفاده شده است. برای بردار تخصیص از تقاطع نگه دارنده چند نقطه ای^۱ (MPX) و برای بردار توالی از تقاطع بهبود دهنده پیشینیزی عملیات^۲ (IPOX) استفاده شده است [۱۹].

1- Multipoint preservative crossover (MPX)

2- Improved precedence operation crossover (IPOX)

اپراتور جهش

این اپراتور همانند الگوریتم ژنتیک برای ایجاد تنوع گرایی در الگوریتم BBO به آن افزوده شده است. لازم به یاد آوری است که در BBO، در هر بار تکرار، برای هر جواب با توجه به نرخ m_i و روشی که در استراتژی انتخاب جهش توضیح داده شد، باید در مورد جهش یافتن تصمیم گرفته شود. اگر لازم شد تا جوابی جهش یابد، از روش استفاده شده توسط [۱۹] استفاده می‌کنیم.

الگوریتم مبتنی بر جغرافیای زیستی چند هدفه (MOBBO)

همان طور که گفته شد، در این مقاله برای توسعه یک نسخه چند هدفه از این الگوریتم از روش مبتنی بر پارتو استفاده شده و با استفاده از اپراتورهای استفاده شده در الگوریتم NSGAI نسخه ای چند هدفه از الگوریتم BBO را ارائه خواهد شد. برای این منظور در روند تکاملی که در زیر بخش قبلی توضیح داده شد، به جای آنکه الگوریتم ژنتیک و اپراتورهای آن برای ایجاد Q_r از P_r استفاده شوند، الگوریتم BBO و اپراتورهایش (انتخاب، مهاجرت و جهش) استفاده خواهند شد. در حقیقت قلب جستجوی الگوریتم NSGAI را که الگوریتم ژنتیک می‌باشد، با الگوریتم BBO تعویض خواهیم کرد. در شکل ۱ شبه کد الگوریتم توسعه داده شده ارائه شده است.

الگوریتم جستجو هارمونی چند هدفه (MOHS)

هنگامی که یک سراینده یک آلت موسیقی را می‌نوازد با سه حالت روبرو است: (۱) نواختن بر اساس حافظه‌اش؛ (۲) نواختن بر اساس تنظیم کوک و (۳) نواختن تصادفی. این سه حالت در واقع سه عملگر کیفی مد نظر در HS می‌باشد که به ترتیب عملگر حافظه هارمونی^۱ (HM)، تنظیم کوک، و تصادفی نواختن است. در حقیقت، فرآیند بداهه-سرای HS ترکیب این سه عملگر می‌باشد. قدمهای اصلی پیاده‌سازی HS در زیر بخش‌های ذیل ارائه می‌شود.

1- Harmony memory (HM)

پارامترهای مسئله را تنظیم کنید
 تولید جمعیت اولیه
 ارزیابی بوم های زیستی و محاسبه HSI برای هر یک از اعضاء جمعیت
 مرتب سازی بوم ها بر اساس HSI
 محاسبه m_j, P_j, μ_j و λ_j بر اساس مرتبه های بدست آمده بعد از مرتب سازی
 حلقه بیرونی: تا رسیدن به شماره تکرار Num. iteration قدم های زیر انجام دهید (for $i=1$: Num. iteration)
 $P_t = \text{population}$
 حلقه داخلی: تا رسیدن به شماره تکرار های Pop. Size قدم های زیر انجام دهید (for $j=1$: Pop. Size)
 عددی تصادفی تولید کنید $Rand \in [0,1]$
 اگر $Rand \leq \lambda_j$
 با استفاده از روش باینری تورنمنت یک بوم را انتخاب کنید: $H_i(SIV)$
 اپراتور مهاجرت را انجام دهید $H_j(SIV) \leftarrow H_i(SIV)$
 اگر $Rand \leq \lambda_j$ بر قرار نیست
 بوم بدون تغییر بماند
 پایان شرط
 عددی تصادفی تولید کنید $Rand \in [0,1]$
 اگر $Rand \leq m_j$
 اپراتور جهش را انجام دهید
 پایان شرط
 پایان حلقه داخلی
 $Q_t = \text{new population}$
 $R_t = P_t \cup Q_t$
 اپراتور مرتب سازی نا مغلوب را روی R_t انجام دهید
 اپراتور فاصله ازدحامی را روی R_t انجام داده آنرا مرتب نمایید
 جمعیت نسل بعد P_{t+1} را تشکیل دهید
 بر اساس مرتبه های بدست آمده بعد از مرتب سازی m_j, P_j, μ_j و λ_j بروز رسانی شوند
 پایان حلقه بیرونی



شکل ۱. شبه کد الگوریتم MOBBO

بردار هارمونی^۱ و فرآیند رمزگشایی

طریقه تعریف ساختار بردار هارمونی از جمله تأثیرگذارترین بخش ها برای افزایش کارایی الگوریتم های بهینه سازی ترکیباتی می باشد. گفتنی است، بردار هارمونی دقیقاً همانند جوابهای BBO تولید و رمزگشایی می شود.

1- Harmony Vector

فرآیند بداهه‌سرایي^۱

به منظور اجرای فرآیند بداهه‌سرایي در تکرارهای مختلف الگوریتم، یک عدد تصادفی تولید شده و سپس یک یا دو عملگر HS جهت اجرای فرآیند بداهه‌سرایي مورد استفاده قرار می‌گیرند. تشریح عملگرهای HS در سه بخش زیر تشریح شده و روابط (۱۸)، (۱۹) و (۲۰) احتمال استفاده شدن هر یک از عملگرها را نشان می‌دهد.

نرخ مراجعه به HM: استفاده از این عملگر در الگوریتم جستجوی هارمونی مشابه بحث نخبه‌گرایی در الگوریتم ژنتیک است. در واقع این عملگر تضمین می‌کند که بهترین هارمونی‌ها در طی فرآیند بهینه‌سازی از حافظه پاک نخواهند شد. این عملگر با نرخ به نام نرخ ملاحظه حافظه هارمونی^۲ (HMCR) کنترل می‌شود. مقادیر کوچک این نرخ باعث کندی همگرایی الگوریتم خواهد شد. علت نیز وجود تعداد کمی هارمونی برگزیده در فرآیند بداهه‌سرایي است. از طرفی دیگر، مقادیر بالای این نرخ باعث این می‌شود که تنها هارمونی‌های HM انتخاب شده و مورد استفاده قرار گیرند لذا الگوریتم در بهینه‌های محلی مستقر خواهد شد. بدین منظور گیم^۳ و همکاران [۱۱] پیشنهاد دادند که مقدار این نرخ در بازه [0.75 – 0.95] تنظیم شود.

عملگر تنظیم کوک: در دنیای موسیقی، تنظیم کوک به معنای تغییر ساختار فرکانس‌ها می‌باشد که معادل تولید جواب‌های همسایه در فرآیند بهینه‌سازی است. در واقع فضایی از جواب که توسط عملگرهای دیگر یافت نمی‌شود، کشف و مورد جستجو قرار می‌گیرد. این عملگر از نرخ به نام نرخ تنظیم کوک^۴ (r_{pa}) بهره‌جسته تا تنظیمات را کنترل نماید. کارکرد این عملگر مشابه عملگر جهش در الگوریتم ژنتیک است. بنابراین مقادیر بالای عملگر تنظیم کوک نوع‌گرایی^۵ را در الگوریتم افزایش خواهد داد. افزایش این نرخ باعث این می‌شود که الگوریتم مانند یک روش جستجوی تصادفی عمل نماید. بدین منظور گیم و همکاران [۱۱] پیشنهاد دادند که نرخ r_{pa} در بازه [0.1 – 0.5] تنظیم شود. همان‌طور که در بالا اشاره شد احتمال بهره‌جستن از این عملگر در رابطه (۱۹) نشانه داده شده است. به منظور تنظیم کوک در این قسمت، از

1- Improvising Process

2- Harmony memory considering rate

3- Geem

4- Pitch-adjusting rate

5- Diversification

اپراتور جهش الگوریتم BBO استفاده می‌شود.

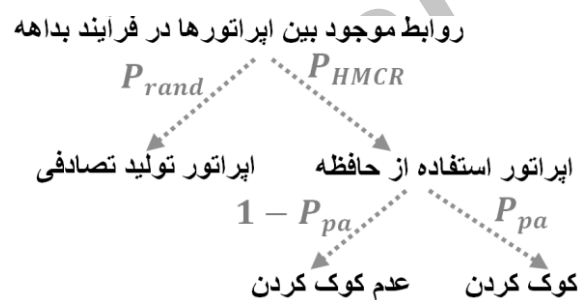
عملگر تصادفی سرایی^۱: همانطور که از اسم این عملگر مشخص است، جواب‌هایی را به صورت تصادفی تولید کرده و وارد جمعیت می‌کند. احتمال تصادفی سرایی نیز از رابطه (۲۰) قابل محاسبه است.

$$P_{HMCR} = HMCR \quad (18)$$

$$P_{pa} = HMCR * r_{pa} \quad (19)$$

$$P_{rand} = 1 - HMCR \quad (20)$$

در انتها رابطه بین مقادیر احتمالی مختلف در HS و در واقع مدلسازی تمامی حالاتی که در تکرارهای مختلف الگوریتم رخ می‌دهد، در شکل ۲ به وضوح قابل مشاهده است.



شکل ۲. روابط احتمالاتی موجود بین اپراتورها در الگوریتم جستجوی هارمونی

الگوریتم جستجوی هارمونی چند هدفه (MOHS)

در زیر بخش‌های قبلی کلیات HS شرح داده شد. حال با ادغام اپراتورهای چند هدفه ای که برای BBO نیز به کار بردیم، نسخه چند هدفه این الگوریتم به صورت شبه کد ارائه شده در

1- Randomization operator

پارامترهای مسئله را تنظیم کنید

مجموعه هارمونی ها یا حافظه هارمونی را تولید کنید
مجموعه هارمونی ها یا حافظه هارمونی را ارزیابی کنید

اپراتور مرتب سازی نا مغلوب را روی مجموعه هارمونی انجام دهید (Rank)
میزان فاصله ازدحامی اعضا را محاسبه کنید (CD)
جمعیت را بر اساس CD و Rank مرتب سازی نمایید

حلقه بیرونی: تا رسیدن به شماره تکرار Num. iteration قدم های زیر انجام دهید (for $i=1$: Num. iteration)
 $P_t = \text{population}$

حلقه داخلی: تا رسیدن به شماره تکرار های Pop. Size قدم های زیر انجام دهید (for $j=1$: Pop. Size)

عددی تصادفی تولید کنید $Rand \in [0, 1]$

اگر $Rand < HMCR$

هارمونی تصافی انتخاب کنید H :

عددی تصادفی تولید کنید $Rand \in [0, 1]$

حال اگر $Rand < P_{pa}$

هارمونی H را کوک نمایید:

وگرنه

هارمونی بدون تغییر بماند

پایان شرط

وگرنه

هارمونی جدیدی را بداهمه نوازی نمایید H :

پایان شرط

مجموعه جواب هارمونی یا حافظه هارمونی را بروز کن

پایان حلقه داخلی

$Q_t = \text{new population}$

$R_t = P_t \cup Q_t$

اتور مرتب سازی نا مغلوب را روی R_t انجام دهید

اپراتور فاصله ازدحامی را روی R_t انجام داده آنرا مرتب نمایید

جمعیت نسل بعد P_{t+1} را تشکیل دهید

پایان حلقه بیرونی

شکل ۳. شبه کد الگوریتم MOHS

شکل ۳ پیشنهاد می شود. گفتنی است، این الگوریتم توسط سیاسوبرامانی و سوآراپ^۱ [۲۳] نیز چند هدفه شده است. ولی آنها صرفاً مفهوم غلبه را وارد کرده اند و

دیگر بحث تنوع گرایی و اپراتور فاصله ازدحام را در نظر نگرفته اند. لازم به یاد آوری است، در اینجا نیز همانند MOBBO، تفاوت عمده با الگوریتم NSGAI، در قلب جستجو می باشد. بدین معنی که به جای استفاده از الگوریتم ژنتیک، الگوریتم هارمونی برای تولید نسل جدید استفاده شده است. یعنی، بعد از ایجاد همسایگی در نسل ها، دیگر رویکرد چند هدفه برخورد با جمعیت در هر سه الگوریتم NSGAI، MOBBO و MOHS یکسان خواهد بود.

نتایج محاسباتی

به منظور ارزیابی مدل توسعه داده شده و بررسی عملکرد الگوریتم معرفی شده، ابتدا چند مسئله از مسائل ارائه شده توسط برندی مارت [۳] را توسعه میدهم. برای این منظور در مواردی که ماشینی می تواند عملی خاص را انجام دهد برای آن (علاوه بر زمان پردازش) میزان جریان مصرفی را نیز در نظر میگیریم. سپس تعدادی شاخص چند هدفه تعریف نموده و با توجه به آنها الگوریتم های معرفی شده را با یکدیگر مقایسه نموده و نتایج را ارائه خواهیم کرد. لازم به ذکر است، در هر دو الگوریتم تعداد جمعیت ۱۵۰ در نظر گرفته شده است. همچنین در الگوریتم MOBBO تعداد حلقه های اجرای برنامه ۱۵۰ عدد در نظر گرفته شده است که برابر با تعداد حلقه های بیرونی الگوریتم MOHS است. در الگوریتم MOHS تعداد دورهای حلقه داخلی ۲۰ دور، درصد ورود به حافظه (P_{HMCR}) ۸۰ درصد و در کوک کردن (P_{pa}) ۴۰ درصد در نظر گرفته شده اند. گفتنی است V ، ρ ، D و H به ترتیب ۳۶۰ ولت، ۹۰ درجه، ۲۵ روز و ۸ ساعت در نظر گرفته شده اند و اعداد مربوط به جریان هم در به صورت تصادفی در بازه [10-100] آمپر تولید شده اند.

شاخص های ارزیابی عملکرد الگوریتم های چند هدفه

در این زیر بخش به توضیح خلاصه ای از شاخص های استفاده شده اشاره می شود. برای توضیحات بیشتر شاخص ها می توان به مراجع مربوطه مراجعه نمود.

- فاصله گذاری^۱: این روش توسط اسکات^۲ [۲۱] ارائه شده و فاصله نسبی جواب

1- Spacing
2- Scott

- های متوالی را محاسبه می کند. این شاخص هر چه کمتر باشد بهتر است.
- معیار پوشش مجموعه^۱: این معیار، نسبت جواب هایی از مجموعه جواب B را که به صورت توسط جواب هایی از مجموعه جواب های A غلبه می شوند، محاسبه می کند [۲۸]. این شاخص هر چه بیشتر باشد بهتر است.
- فاصله از جواب ایده آل^۲: این معیار، میانگین فاصله جواب های پارتو از مبدا مختصات را محاسبه می نماید. هر چه این معیار کمتر باشد، کارایی الگوریتم بیشتر خواهد بود [۱۵].
- بیشترین گسترش^۳: این معیار طول قطر مکعب فضایی که توسط مقادیر انتهایی اهداف برای مجموعه جواب های نا مغلوب بکار می رود را اندازه گیری می کند [۲۸]. این شاخص هر چه بیشتر باشد بهتر است.
- معیار تعداد جواب های پارتو (NOS^۴): هر چه تعداد جواب های پارتو یک روش بیشتر باشد، آن روش مطلوب تر است.

خروجی شاخص ها

در این قسمت، خروجی الگوریتم ها، روی شاخص های معرفی شده، ارائه خواهد شد. برای این منظور ابتدا در جدول (۱) اعداد بدست آمده از شاخص ها ارائه شده و سپس با توجه به این اعداد آزمایش هایی آماری انجام می شود. برای این منظور، از آزمون من ویتنی که آزمونی غیر پارامتریک شناخته می شود، استفاده می کنیم [۱۲]. نتایج این آزمون در جدول (۲) ارائه شده اند. نتایج بدست آمده از جدول (۲) در قالب نمودار آماری جعبه ای (Box-plot) نیز ارائه خواهد شد، که در شکل های (۵) و (۶) نمایش داده شده اند. شکل (۵) برای شاخص هایی ارائه شده است که در جدول (۲) تفاوت معنی داری در دو الگوریتم دارند و شکل (۶) نیز برای شاخص هایی است که در جدول (۲) تفاوت معنی داری را نشان نمی دهند. دو شکل (۴) و (۷) نیز برای تحلیل دقیق تر شاخص ها ارائه شده اند. در ادامه بخش، با توجه به این جداول و شکل ها، عملکرد شاخص ها را نسبت به هر کدام از الگوریتم ها بررسی خواهیم کرد.

1- Set coverage metric
2- Mean ideal distance (MID)
3- Maximum spread
4- Number of the Pareto solutions

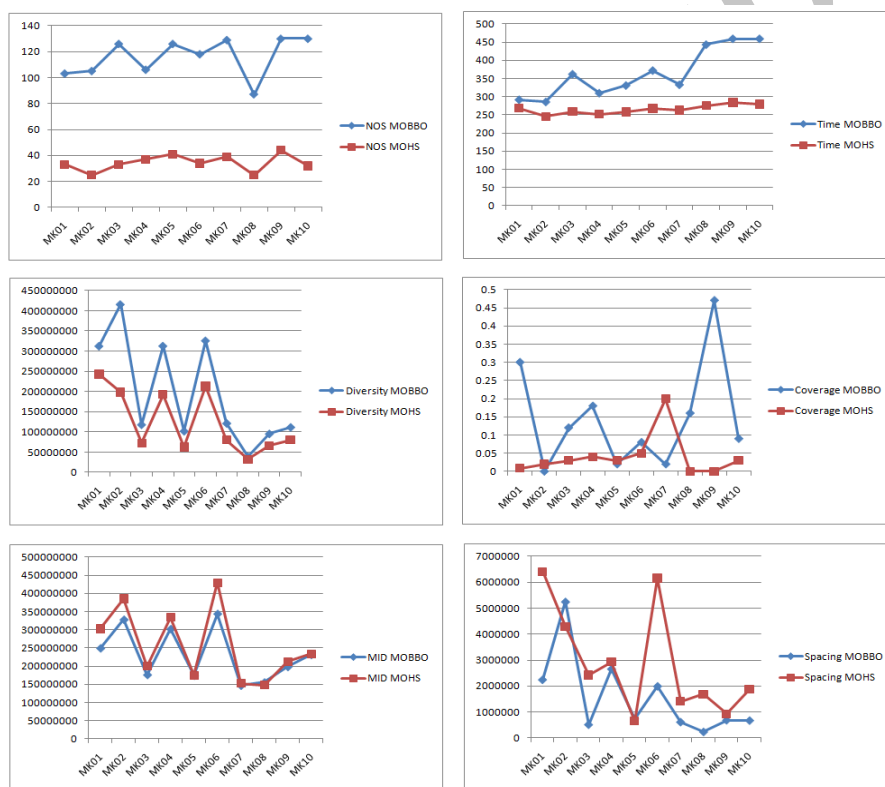
در سطر آخر از جدول (۲) نتایج بدست آمده از هر شاخص، روی کل مجموعه مسائل تولید شده، جمع شده اند. همان طور که با توجه به اعداد پررنگ شده در این سطر مشخص است، در یک نگاه کلی به خروجی ها، الگوریتم MOBBO به جزء در شاخص زمان، در بقیه شاخص ها از الگوریتم MOHS نتیجه بهتری دارد. لازم به یادآوری است که شاخص های NOS، Diversity و Coverage هر بیشتر باشند مطلوب بوده و شاخص های Time، Spacing و MID هر چه کمتر باشند. البته نتایج بدست آمده در این جدول، به صورت گرافیکی در شکل (۴) نمایش داده شده است. در این شکل، برای هر کدام از شاخص ها، روی کل مجموعه مسائل تولیدی، خروجی ها به صورت مجزا رسم شده اند.

Archive of SID

جدول ۱. مقایسه دو الگوریتم MOBBO و MOHS بوسیله شاخص های چند هدفه

شماره مسئله	$n \times m$ سایز مسئله	پیشنهادی MOBBO						پیشنهادی MOHS					
		Time	NOS	Spacing	Diversity	MID	Coverage	Time	NOS	Spacing	Diversity	MID	Coverage
MK02	10*6	286	105	5250569.7	416154592.1	327565968.6	0	245	25	4294079.5	198290170.8	385225974.9	0.02
MK03	15*8	362	126	515668.2	118530400.6	176291807.2	0.12	259	33	2426772.9	72231500.09	200197387.6	0.03
MK04	15*8	310	106	2660756.6	313321736.4	301968165.8	0.18	252	37	2923801.7	192466544.8	334006413.4	0.04
MK05	15*4	331	126	738702.8	102494126.4	174410891.1	0.02	258	41	647850.1	62729338.1	174339893.1	0.03
MK06	10*15	372	118	1994692.9	326367680.8	343595076.4	0.08	267	34	6150522.9	213121509.4	429325049.4	0.05
MK07	20*5	333	129	616473.7	121813510.3	146938155.06	0.02	263	39	1416319.5	79952532.3	152571591.5	0.2
MK08	20*10	443	87	236607.1	39337695.2	155608258.8	0.16	275	25	1686164.8	32332716.7	148187975.2	0
MK09	20*10	459	130	687258.7	95991436.1	198301179.5	0.47	284	44	925317.05	65906728.1	212329980.3	0
MK10	20*15	459	130	675366.8	112232843.7	231260288.9	0.09	279	32	1876090.7	79774286.3	233748008.7	0.03
ارزش کل		3647	1160	15623886	1959199654	2305186862	1.44	2650	343	28747329	1239405814	2572607038	0.409

حال، برای بررسی علمی تر نتایج بدست آمده، برای هر شاخص یک آزمون آماری انجام شده است. نتایج این آزمون‌های آماری در جدول (۲) دیده می‌شود. اولین نتیجه جالب قابل برداشت از این جدول آنست که علارغم آنکه در نگاه کلی به نتایج جدول (۱)، که در آن MOBBO به غیر از زمان در تمام شاخص‌ها بهتر بود، در تحلیل‌های آماری تنها در شاخص NOS برتری آماری معنی دار دارد. نکته جالب دیگر آنکه MOHS چه در نگاه کلی و چه آماری تفاوت معنی دار خود را در مورد زمان حفظ نموده است.

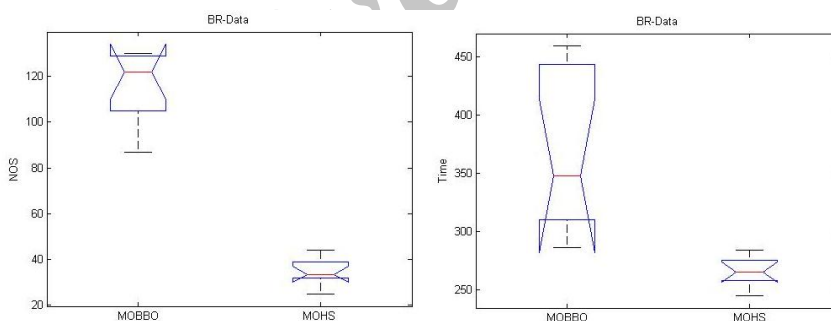


شکل ۴. مقایسه دو الگوریتم MOBBO و MOHS بوسیله مقایسه گرافیکی شاخص‌های چند هدفه در حقیقت، دو الگوریتم تنها در دو شاخص NOS و Time تفاوت معنی دار دارند. این نتیجه در نمودار جعبه ای رسم شده در شکل (۵) نیز قابل بررسی است. قابل ذکر است، همخوانی نتایج جدول (۳) و شکل (۵) با شکل (۴) دلیل دیگری است برای نتایج اعلام شده از شاخص‌ها روی الگوریتم‌ها.

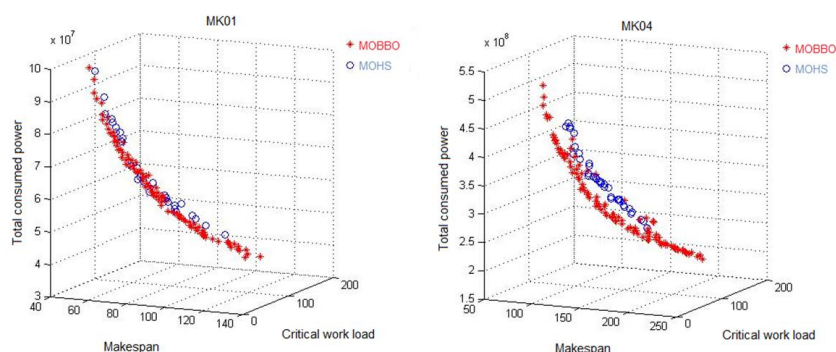
جدول ۲ مقایسه آماری دو الگوریتم MOBBO و MOHS به وسیله شاخص‌های چند هدفه

آزمون من ویتنی		
	P-value	نتایج
Diversity	0.12	عدم رد فرض صفر
Spacing	0.08	عدم رد فرض صفر
MID	0.62	عدم رد فرض صفر
NOS	0.0002	MOBBO برتری دارد بر MOHS
Coverage	0.089	عدم رد فرض صفر
Time	0.0002	MOHS برتری دارد بر MOBBO

همچنین در شکل (۶) نمایی از جوابهای پارتوی بدست آمده از الگوریتم‌ها در ۲ مسئله مختلف نشان داده شده اند. نام مسئله مربوطه بالای هر شکل نشان داده شده است. این شکل نیز دلیلی دیگری است بر اینکه الگوریتم MOBBO در شاخص NOS عملکرد بهتری نسبت به الگوریتم MOHS دارد. همچنین تشابه‌های موجود در خروجی‌های دو الگوریتم نیز از این شکل مشخص می‌شود.



شکل ۵. نمودار جعبه‌ای برای شاخص‌هایی که تفاوت معنی داری دارند



شکل ۶. پارتوهای بدست آمده از الگوریتم‌ها روی ۲ مسئله MK04 و MK01

نتیجه‌گیری و مطالعات آتی

در این مقاله، مساله کار کارگاهی منعطف مورد مطالعه قرار گرفت و مفهومی کاربردی به آن افزوده شد. این مفهوم که بهینه‌سازی توان مصرفی در طول ماه می‌باشد، به همراه دو هدف دیگر که زمان تکمیل کارها و بار کاری ماشین بحرانی می‌باشند، مسئله چند هدفه‌ای را تشکیل دادند. سپس برای حل این مدل، دو الگوریتم جدید چند هدفه، با نام‌های MOBBO و MOHS توسعه داده شدند. نتایج این الگوریتم‌ها روی تعدادی شاخص چند هدفه، به صورت آماری، مورد مطالعه قرار گرفت. بوسیله این آزمایش‌های آماری، دیده شد که الگوریتم MOBBO در شاخص NOS و الگوریتم MOHS در شاخص زمان برتری معنی دارند و در سایر شاخص‌ها، تفاوت معنی‌داری وجود ندارد. مطالعات آتی این تحقیق می‌تواند وارد سازی مفهوم پیشنهادی در سایر مدل‌های زمانبندی و یا حتی سایر حوزه‌های مدل‌سازی‌های مدیریت و مهندسی باشد. در نگاهی محض هم، توسعه مفهوم پیشنهادی به صورت فازی با در نظر گیری جریان‌ها به صورت فازی جذاب می‌باشد. استفاده از الگوریتم‌های توسعه داده شده در سایر حوزه‌ها نیز پیشنهاد می‌شود.

منابع

1. Barnes, J.W., Chambers, J.B., **Flexible job shop scheduling by tabu search**. Graduate program in operations research and industrial engineering. University of Texas, Austin, Technical Report Series, ORP96-09, 1996.
2. Bhattacharya, A., Chattopadhyay, P.K., **Solving complex economic load dispatch problems using Biogeography-based optimization**. *Expert Systems with Applications* 37, 3605–3615, 2010.
3. Brandimarte, P., **Routing and scheduling in a flexible job shop by taboo search**. *Annual Operation Research* 41, 157–183, 1993.
4. Brucker, P., Schlie, R., **Job-shop scheduling with multipurpose machines**. *Computing* 45(4), 369–375, 1990.
5. Chen, J.C., Chen, K.H., Wu, J.J., Chen, C.W., A study of the flexible job shop scheduling problem with parallel machines and reentrant process. *International Journal of Advance Manufacturing Technology* 39(3–4), 344–354, 2008.
6. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T., **A fast elitist non-dominated sorting genetic algorithm for multi-Objective optimization: NSGA-II**. In: *Proceedings of the parallel problem solving from nature VI (PPSN-VI) conference*, 849–858, 2000.
7. Fattahi, Parviz, Jolai, Fariborz, Arkat, Jamal, **Flexible job shop scheduling with overlapping in operations**. *Applied Mathematical Modeling* 33, 3076–3087, 2009.
8. Fattahi, Parviz, Saidi Mehrabad, Mohammad, Jolai, Fariborz, **Mathematical modeling and heuristic approaches to flexible job shop scheduling problems**, *International Journal of Advance Manufacturing Technology*. DOI 10.1007/s10845-007-0026-8, 18:331–342, 2007.
9. Frutos, Mariano, Olivera, Ana Carolina, Tohmé, Fernando, **A memetic algorithm based on a NSGAI scheme for the flexible job-shop scheduling problem**. *Annual Operation Research*, DOI 10.1007/s10479-010-0751-9, 2010.
10. Gao, J., Gen, M., Sun, L.Y., Zhao, X.H., **A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems**. *Computer and Industrial Engineering* 53(1), 149–162, 2007.
11. Geem, Z.W., Kim, J.-H., Loganathan, G.V., **A new heuristic optimization algorithm: harmony search**, *Simulation* 76 (2) 60–68, 2011.
12. Hollander, M., Wolfe, D.A., **Non-parametric Statistical Methods**. John Wiley & Sons, 1973.
13. Hurink, E., Jurisch, B., Thole, M., **Tabu search for the job shop scheduling problem with multi-purpose machine**. *Operations Research Spektrum* 15 (4), 205–215, 1994.
14. Kacem, I., Hammadi, S., Borne, P., **Approach by localization multi-objective evolutionary optimization for flexible job-shops scheduling problems**. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* 32 (1), 1–13, 2002.
15. Karimi, N., Zandieh, M., Karamooz, H.R., **Bi-objective group scheduling in hybrid flexible flow shop: A multi-phase approach**. *Expert Systems with Applications* 37, 4024–4032, 2010.
16. Kundra, H., Kaur, A., Panchal, V., **An integrated approach to biogeography based optimization with case based reasoning for retrieving groundwater possibility**. In: *Proceedings of the Eighth Annual Asian Conference and Exhibition on Geospatial Information, Technology and Applications*, August, Singapore, 2009.
17. Lee, K.S., Geem, Z.W., **A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice**. *Computer Methods in Applied Mechanics and Engineering* 194, 3902–3933, 2005.
18. Mastrolilli, M., Gambardella, L.M., **Effective neighborhood functions for the**

- flexible job shop problem. *Journal of Scheduling* 3(1), 3–20, 2000.**
19. Rahmati, S.H.A., Zandieh, M., **A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem,** *International Journal of Advance Manufacturing Technology*, DOI 10.1007/s00170-011-3437-9, 2011.
 20. Rashid M. H., **Power Electronic.** Academic press, 2001.
 21. Schott, J. R., **Fault tolerant design using single and multicriteria genetic algorithms optimization.** Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 1995.
 22. Simon, D., **Biogeography-based optimization.** *IEEE Transactions on Evolutionary Computation* 12, 702–713, 2008.
 23. Sivasubramani, S., Swarup, K.S., **Environmental/economic dispatch using multi-objective harmony search algorithm,** *Electronic Power System and Research*, doi:10.1016/j.epsr.2011.04.007, 2011.
 24. Torabi, S.A., Karimi, B., Fatemi, Ghomi S.M.T., **The common cycle economic lot scheduling in flexible job shops: The finite horizon case.** *International Journal of Production Economics* 97, 52–65, 2005.
 25. Wang, S., Yu, J., **An effective heuristic for flexible job-shop scheduling problem with maintenance activities,** *Computers & Industrial Engineering*, 59: 436-447, 2010.
 26. Xia, W.J., Wu, Z.M., **An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems.** *Computer and Industrial Engineering* 48(2), 409–425, 2005.
 27. Yazdani M., Amiri M., Zandieh M., **Flexible job-shop scheduling with parallel variable neighborhood search algorithm.** *Expert Systems with Applications* 37, 678–687, 2010.
 28. Zitzler E. **Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications.** Ph.D. Thesis, Dissertation ETH No. 13398, Swiss Federal Institute of Technology (ETH), 1999.

Archive of SID