مجله تحقیق در عملیات در کاربردهای آن، سال چهاردهم، شماره اول (پیاپی ۵۲)، بهار ۹۶، ص ص ۷۶-۶۳
شاپا ۷۲۸۶-۲۲۵۱

مقایسه و تحلیلی بر استفاده از الگوریتم‌های فراابتکاری برای حل مسایل زمان‌بندی تولید کارگاهی

پروانه سموئی*[1]، پرویز فتاحی[2]
۱- استادیار، دانشگاه بوعلی سینا، دانشکده مهندسی، گروه مهندسی صنایع، همدان، ایران
۲- دانشیار، دانشگاه الزهراء، گروه مهندسی صنایع، تهران، ایران

رسید مقاله: ۱۳ شهریور ۱۳۹۵
پذیرش مقاله: ۲۳ بهمن ۱۳۹۵

چکیده
یکی از مسایل مهم در حوزه‌های تحقیقاتی و کاربردی مدیریت تولید، زمان‌بندی صحیح و مناسب عملیات کارهای مختلف است. از این رو، روش‌ها و رویکردهای مختلفی برای حل این دسته از مسایل، چه در سیستم‌های تولیدی کارگاهی و چه غیرکارگاهی به وجود آمده است. یکی از این رویکردهای حل، با توجه به NP-Hard مسایل زمان‌بندی تولید کارگاهی، استفاده از الگوریتم‌های فراابتکاری نظیر جستجوی ممنوع، انجماد تدریجی، الگوریتم ژنتیک و بهینه‌سازی توده ذرات می‌باشند که در مقالات مختلف مورد بحث و بررسی قرار گرفته‌اند؛ لذا در این مقاله سعی شده است، ضمن حل این دسته از مسایل، به کمک این الگوریتم‌ها، به تحلیل و مقایسه آنها پرداخته شود. بدین منظور مسایلی با ابعاد مختلف در نظر گرفته شده و تحلیل‌های زمانی، کمی و پارامتری در مورد آنها صورت پذیرفته است.

کلمات کلیدی: زمان‌بندی تولید کارگاهی، جستجوی ممنوع (TS)، انجماد تدریجی(SA)، الگوریتم ژنتیک(GA)، بهینه‌سازی توده ذرات(PSO).

۱ مقدمه
یکی از مهمترین مسایلی که بسیاری از کارخانجات با آن برخورد می‌کنند، مساله زمان‌بندی سیستم کارگاهی است. این نوع زمان‌بندی از اولویت زیادی در حوزه‌های مدیریت تولید، مهندسی و مسایل بهینه سازی ترکیبیاتی برخوردار است؛ چرا که این مساله جز مسایل NP-Hard بوده [۱] و ارایه و یا توسعه تکنیک‌هایی جهت حل آن می‌تواند بسیار پراهمیت باشد؛ لذا بسیاری از محققین سعی کرده‌اند تا با ارایه روش‌هایی جهت زمان‌بندی بهتر، به شکل مناسب‌تری به سفارشات مشتریان پاسخگو باشند و ضمن تامین بهتر مواد و قطعات مورد نیاز از اتلاف زمان، سرمایه و نیروی انسانی نیز جلوگیری نمایند. دسته‌ای از این روش‌ها، استفاده از الگوریتم‌های فرا ابتکاری نظیر

* عهده دار مکاتبات
آدرس الکترونیکی: p.samouei@basu.ac.ir

[Downloaded from jorlu.ir on 2026-11-29]
[DOR: 20.1001.1.22517286.2017.14.1.1.6]

جستجوی ممنوع، انجماد تدریجی، ژنتیک و بهینه‌سازی توده ذرات می‌باشند، که نتایج نسبتاً خوبی را به همراه داشته‌اند. به منظور اشاره‌ی مختصری بر این تحقیقات می‌توان به ون لارهوون[۲]، آیدین و فوگارتی [۳]، علمی و همکاران [۴] در زمینه کاربرد انجماد تدریجی، ژنگ و همکاران [۵]، دل آمیکو و تروبیان [۶]، بروکر و نیر [۷] در جستجوی ممنوع، دیویس[۸]، چنگ و همکاران[۹]، ژنگ و وو [۱۰]، ژو و همکاران [۱۱] به کاربرد الگوریتم ژنتیک و شا و سو [۱۲] و زیا و و [۱۳] به استفاده از بهینه‌سازی توده ذرات در زمان‌بندی تولید کارگاهی اقدام نموده‌اند. در سال‌های اخیر نیز منسیا و همکاران [۱۴] یک الگوریتم ممتیک برای مسایل زمان‌بندی تولید کارگاهی با در نظر گرفتن اپراتورها ارایه کردند که از ساختارهای همسایگی در جستجوهای محلی خود استفاده می‌نمود. گائو و همکاران[۱۵] نیز یک الگوریتم بر مبنای بهینه‌سازی توده ذرات و جستجوی همسایگی متغیر برای مساله زمان‌بندی تولید کارگاهی ارایه کردند. اسدزاده [۱۶] نیز یک الگوریتم ژنتیک با جستجوی محلی برای مساله زمان‌بندی تولید کارگاهی ارایه نمود. همچنین او در سال ۲۰۱۶ نیز یک الگوریتم کلونی زنبور عسل مصنوعی موازی برای زمان‌بندی تولید کارگاهی اراده کرد و در آن از استراتژی مهاجرت پویا کمک گرفت[۱۷]. علاوه بر این، اسکاریا و همکاران [۱۸] یک روش موثر برای زمان‌بندی تولید کارگاهی بر مبنای کلونی زنبور عسل ارایه کردند و دو شاخص حداقل کردن زمان تکمیل و کل تاخیرات را در نظر گرفتند. همچنین میرشکاریان و سورماز [۱۹] تعداد ۳۸۰ جنبه برای مسایل زمان‌بندی تولید کارگاهی توسعه دادند و برای ارزیابی آن‌ها از یادگیری ماشین و روش‌های آماری استفاده نمودند و برای ارزیابی همبستگی این عوامل از بهینه‌سازی زمان تکمیل کمک گرفتند. به علاوه، کوپفال و بایرورس [۲۰] مطالعه‌ای روی شش روش جستجوی محلی همسایگی برای مساله زمان‌بندی تولید کارگاهی با هدف حداقل‌سازی تاخیرات موزون انجام دادند. اکرم و همکاران [۲۱] نیز یک الگوریتم انجماد تدریجی سریع برای مسایل زمان‌بندی تولید کارگاهی ارایه نمودند؛ بهشتی نیا و حسنی بیدگلی [۲۲] نیز ترکیبی جدید از الگوریتم های فراابتکاری ژنتیک و شبیه‌سازی تبرید برای حل مساله زمان‌بندی کارگاهی انعطاف پذیر ارایه کردند. اما هیچ یک از این مقالات به مقایسه تحلیلی در نحوه عملکرد این الگوریتم‌ها در ابعاد مختلف نپرداخته‌اند؛ لذا در این مقاله سعی می‌شود ضمن استفاده از ۴ نوع الگوریتم بیان شده، به مقایسه و تحلیل عملکرد آن‌ها پرداخته شود. با استفاده از تحلیل‌ها و مقایسات انجام شده در این مقاله، خوانندگان محترمی که به دنبال یافتن الگوریتم مناسبی برای زمان‌بندی تولید کارگاهی هستند، راحت‌تر می‌توانند از میان الگوریتم‌های مطرح شده انتخابی داشته باشند. علاوه بر این چنانچه بخواهند یک الگوریتم هیبریدی نیز ارایه دهند، ساده‌تر می‌توانند به نقاط قوت و ضعف این الگوریتم‌ها برای حل مسایل زمان‌بندی تولید کارگاهی پی ببرند.

ساختار مقاله پیش رو به شرح زیر است:

بخش دوم به معرفی زمان‌بندی تولید کارگاهی و مدل ریاضی آن اختصاص دارد. در بخش سوم نیز توضیحات مختصری در زمینه الگوریتم‌های به کارگرفته شده داده می‌شود. بخش چهارم به مقایسه و تجزیه و تحلیل استفاده از این الگوریتم‌ها در زمان‌بندی تولید کارگاهی می‌پردازد. نهایتاً در فصل پنجم جمع‌بندی و نتیجه‌گیری این مقاله ارایه شده است.

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.1.1.6]

۲ زمان‌بندی تولید کارگاهی

مساله زمان‌بندی تولید کارگاهی شامل زمان‌بندی n کار J_1، J_2، ...، J_n روی m ماشین M_1، M_2، ...، M_m می‌باشد که هر کار نیز دارای h_j عملیات می‌باشد که باید به ترتیب انجام گیرند. علاوه‌براین هر ماشین در هر لحظه از زمان تنها قادر است یک عملیات را پردازش کند. به عبارت دیگر امکان پردازش همزمان چند عملیات روی یک ماشین امکان‌پذیر نیست. هدف از زمان‌بندی در این سیستم تولیدی تعیین توالی عملیات‌ها برای هر ماشین به گونه‌ای است که تابعی از قبل تعریف شده را بهینه نمود. مدل ریاضی این مساله در زیر آمده است:

$$Min \quad C_{\max}$$

s.t.

$$C_{\max} \geq t_{jh_j} + p_{j,h_j} \qquad j=1,...,n \qquad (1)$$

$$t_{jh_j} + p_{j,h} \leq t_{j,h+1} \qquad j=1,...,n; \quad h=1,...,h_j-1 \qquad (2)$$

$$Tm_{i,k} + p_{j,h}.x_{i,j,h,k} \leq Tm_{i,k+1} \qquad i=1,...,m; \quad j=1,...,n; \quad h=1,...,h_j; \quad k=1,...,k_i-1 \qquad (3)$$

$$Tm_{i,k} \leq t_{j,h} + (1-x_{i,j,h,k})L \qquad i=1,...,m; \quad j=1,...,n; \quad h=1,...,h_j; \quad k=1,...,k_i \qquad (4)$$

$$Tm_{i,k} + (1-x_{i,j,h,k})L \geq t_{j,h} \qquad i=1,...,m; \quad j=1,...,n; \quad h=1,...,h_j; \quad k=1,...,k_i \qquad (5)$$

$$\sum_i \sum_j x_{i,j,h,k} = 1 \qquad i=1,...,m; \quad k=1,...,k_i \qquad (6)$$

$$\sum_j x_{i,j,h,k} = a_{i,j,h} \qquad i=1,...,m; \quad j=1,...,n; \quad h=1,...,h_j \qquad (7)$$

$$t_{j,h} \geq o \qquad j=1,...,n; \quad h=1,...,h_j \qquad (8)$$

در مدل ریاضی فوق، $C_{\max}$ ماکزیمم زمان تکمیل کلیه کارها، $t_{j,h}$ زمان شروع پردازش عملیات $O_{j,h}$، $p_{j,h}$ زمان پردازش عملیات $O_{j,h}$، $x_{i,j,h,k}$ می‌تواند صفر و یا یک باشد. زمانی که عملیات $O_{j,h}$ روی ماشین i در اولویت k انجام شود، مقدار ۱ و در بقیه موارد مقدار صفر را خواهد پذیرفت. علاوه‌براین $Tm_{i,k}$ نشاندهنده‌ی زمان شروع کارها روی ما شین i در اولویت k می‌با شد. همچنین L یک عدد خیلی بزرگ و $a_{i,j,h}$ این قابلیت را ن شان می‌دهد که عملیات $O_{j,h}$ بتواند روی ماشین i انجام شود یا خیر. این مقدار نیز می‌تواند صفر و یا یک را پذیرا باشد.

در این مدل، محدودیت(۱) مقدار تابع هدف را معرفی می‌کند.محدودیت(۲) روابط پیش‌نیازی را در بر می‌گیرد. این محدودیت زمان شروع هر عملیات را به اتمام عملیات قبلی از همان کار محدود می‌کند. محدودیت(۳) این الزام را به وجود می‌آورد که در صورتی یک ماشین می‌تواند به کار در نوبت $k+1$ بپردازد که کار آن ماشین در نوبت k به اتمام رسیده باشد. محدودیت (۴) و (۵) این الزام را به وجود می‌آورد که در صورتی یک عملیات می‌تواند شروع شود که هم محدودیت پیش نیازی و هم محدودیت ماشین را رعایت کرده باشد. محدودیت (۶) موجب می‌گردد که به هر ماشین در هر نوبت، تنها یک عملیات از یک کار اختصاص یابد و محدودیت (۷) موجب می‌گردد که برای اجرای یک عملیات از ماشین اختصاص داده شده به آن عملیات استفاده

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.1.1.6]

می‌شود.

۳ الگوریتم‌های به کارگرفته شده

این قسمت به معرفی مختصری از الگوریتم‌های جستجوی ممنوع، انجماد تدریجی، ژنتیک و بهینه‌سازی توده ذرات اختصاص دارد.

۳-۱ جستجوی ممنوع

ساختار کلی این الگوریتم به گونه‌ای است که ابتدا از یک جواب اولیه شروع و سپس بهترین جواب همسایه را از میان همسایه‌های جواب فعلی انتخاب می‌کند. در صورتی که این جواب در فهرست ممنوعه قرار نداشته باشد، الگوریتم به جواب همسایه حرکت می‌کند، در غیراین‌صورت، الگوریتم معیاری به نام معیار تنفس را چک خواهد کرد. بر اساس معیار تنفس اگر جواب همسایه از بهترین جواب یافت شده تا کنون بهتر باشد، الگوریتم به آن حرکت خواهد کرد؛ حتی اگر آن جواب در فهرست ممنوعه باشد. پس از حرکت الگوریتم به جواب همسایه، فهرست ممنوعه به روزرسانی می‌شود. به این معنا که حرکت قبل که به وسیله‌ی آن به جواب همسایه حرکت کردیم در فهرست ممنوعه قرار داده می‌شود تا از بازگشت مجدد الگوریتم به آن جواب و ایجاد سیکل جلوگیری شود. در واقع فهرست ممنوعه ابزاری در الگوریتم جستجوی ممنوعه‌است که توسط آن از قرار گرفتن الگوریتم در بهینه‌ی محلی جلوگیری می‌شود. پس از قرار دادن حرکت قبلی در فهرست ممنوعه، تعدادی از حرکت‌هایی که قبلا در فهرست ممنوعه قرار گرفته بودند از فهرست خارج می‌شوند. مدت زمانی که حرکت‌ها در فهرست ممنوعه قرار می‌گیرند توسط یک پارامتر که زمان ممنوعه نام دارد تعیین می‌شود. حرکت از جواب فعلی به جواب همسایه نیز تا جایی ادامه می‌یابد که شرط خاتمه دیده شود.

۳-۲ انجماد تدریجی

الگوریتم انجماد تدریجی برای اولین بار توسط کرک پاتریک و همکارانش ابداع شد. این الگوریتم در سال ۱۹۸۳ با الهام از فرآیند انجماد تدریجی فلزات ذوب شده و بررسی رفتار آن‌ها در زمان سرد شدن تدریجی و همچنین رسیدن به کم‌ترین سطح انرژی به دست آمده است. در این الگوریتم در صورتی که S_i و E_i به ترتیب حالت ماده و انرژی آن در وضعیت i و S_j و E_j حالت ماده و انرژی آن در وضعیت jباشد، اگر اختلاف بین انرژی حالات i و jکوچک‌تر یا مساوی صفر باشد این حالت پذیرفته می‌شود، در غیر این صورت حالت جدید j با یک احتمال پذیرفته خواهد شد. این الگوریتم، از یک پارامتر دما برای کنترل الگوریتم در طول بهبود جواب و همچنین از الگوی تابع بولتزمن که نوعی توزیع احتمال است، استفاده می‌کند.

۳-۳ ژنتیک

این الگوریتم برای اولین بار توسط جان هلند معرفی شده است و با استفاده از این ایده اقدام به حل مسایل می‌کند

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.1.1.6]

که در طبیعت از ترکیب کروموزوم‌های بهتر، نسل‌های بهتری پدید می‌آیند و در این بین، گاهی اوقات جهش‌هایی نیز در کروموزوم‌ها روی می‌دهد که ممکن است باعث بهتر شدن نسل بعدی شوند.

۴-۳ بهینه‌سازی توده ذرات

الگوریتم PSO که در آن، از رفتار زندگی اجتماعی پرندگان و ماهی‌ها الگوبرداری شده است و از رفتار هر یک از اعضا که ذره نامیده می شود، استفاده می‌کند. این روش با یک گروه از جواب‌های تصادفی (ذرات) شروع به کار می‌کند و به تمام آن‌ها به صورت تصادفی مکان و سرعتی اختصاص می‌دهد. برای یافتن بهترین جواب، ذرات را باید بر فراز فضای جواب به حرکت درآورد و هرگاه یکی از آن‌ها جواب مناسبی یافت، یافته‌های خود را در اختیار سایر ذرات قرار می‌دهد تا آن‌ها نیز به سمت بهترین جواب حرکت کنند. در مراحل مختلف این الگوریتم، سرعت و مکان ذرات به گونه‌ای تغییر می‌کنند که در هر لحظه، هر ذره مکان خود را در فضای جستجو، با توجه به بهترین مکانی که تاکنون در آن قرار داشته و بهترین مکانی که کل همسایگانش به دست آورده‌اند، تنظیم می‌کند.

٤ مقایسه الگوریتم‌ها در زمان‌بندی تولید کارگاهی

قبل از اینکه به مقایسه الگوریتم‌ها پرداخته شود قابل ذکر است که در مساله JSP نوع کدینگ مساله می‌تواند بسیار حائز اهمیت باشد. از میان نحوه‌های کدینگ مساله، از کدینگ Gen & Cheng استفاده شده است که اولاً ساده و قابل فهم بوده و ثانیاً همواره در فضای شدنی مساله قرار می‌گیرد. علاوه‌براین برای تمامی پارامترهای موجود در الگوریتم‌ها تنظیم پارامتر صورت پذیرفته است. به گونه‌ای که بر اساس آزمایشات اولیه، برای هر یک از پارامترهای مساله در دو سطح، مقادیری انتخاب شدند و پس از بررسی اثرات جداگانه و متقابل به کمک نرم‌افزار Minitab مقادیر پارامترها به دست آمدند. این کار برای هر یک از مسایل به طور جداگانه انجام گرفته است. در این بخش، نحوه به کارگیری الگوریتم و همچنین پاسخ‌های به دست آمده و سایر تحلیل‌های مربوط به آن‌ها را در ابعاد مختلف می‌توان به دست آورد.

٤-١ جستجوی ممنوع

در الگوریتم به کار رفته جواب اولیه به صورت کاملاً تصادفی ایجاد می‌شود. علاوه‌براین، برای ایجاد همسایگی و پیدا کردن یک توالی جدید از بردار توالی موجود، به جابه‌جایی دو عملیات مختلف برای دو کار متفاوت اقدام شده است. همچنین برای اجرای الگوریتم از طول لیست ممنوع و تعداد تکرار ثابت استفاده گردید. با تنظیم پارامترهایی که صورت پذیرفت مشخص شد مساله با ۳ کار و ۳ ماشین به پارامترهای انتخابی حساس نیستند؛ اما برای سایر مسایلی که در جدول زیر بدان اشاره شده است، انتخاب پارامترها می‌تواند مهم باشد. به گونه‌ای که از میان پارامترهای مورد نظر ماکزیمم تکرار برابر ۲۰۰ و دوره تنفس ۱۰ بهترین نتایج را حاصل می‌نمود؛ لذا با این پارامترها مسایل حل شدند. از سوی دیگر به دلیل آنکه نتایج قابل اطمینان‌تری حاصل شود، ۵ بار الگوریتم مورد

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.1.1.6]

نظر اجرا شد.

٤-٢ انجماد تدریجی

برای حل این مسایل به کمک الگوریتم SA نیز جواب اولیه به صورت تصادفی ایجاد شدند. به‌علاوه برای ایجاد یک توالی جدید از بردار توالی موجود، دو عدد تصادفی که نشاندهنده‌ی موقعیت عناصر در بردار توالی می‌باشند، ایجاد شـــده و اعداد موجود در موقعیت‌ها با هم جابه‌جا می‌شـــوند. برای تعیین پارامترهای دمای ابتدایی، دمای انتهایی، نرخ سردسازی هندسی و طول زنجیره مارکوف پس از آزمایش‌های اولیه، هر یک از این پارامترها در دو سطح لحاظ شدند. با توجه به آزمایش‌های انجام شده، مشخص شد که مساله ۳×۳ به تغییر پارامترها به دلیل ابعاد کوچکش حساس نیست. علاوه‌براین برای سایر مسایل بهترین پارامترها، مطابق جدول زیر به دست آمد:

جدول ۱. بهترین پارامترهای به دست آمده برای حل به کمک SA

طول زنجیره مارکوف	نرخ تبرید	دمای انتهایی	دمای ابتدایی
۲۰	۰/۹۵	۱	۵۰۰

٤-٣ الگوریتم ژنتیک

برای جواب اولیه که به صـــورت جمعیتی از کروموزوم‌ها هســـتند، هر کروموزوم به صـــورت توالی‌های کاملاً تصادفی از عملیات کارها تعریف شدند. برای ایجاد توالی‌های جدید نیز با توجه به احتمال باقی ماندن کروموزوم در نسل بعد، تعدادی که بهترین پاسخ‌ها را به دست می‌آورند نگه داشته شدند. همچنین با توجه به احتمال جهش نیز تعداد لازم از جمعیت به صورت کاملاً تصادفی انتخاب شده و سپس برای هر کروموزوم، دو موقعیت از توالی آن به تصادف انتخاب شده و جایگاه شان عوض شدند. برای ایجاد نسل جدید نیز با توجه به احتمال ترکیب، دو کروموزوم به تصـــادف انتخاب شـــده و به کمک روشppx لقاح در آن صـــورت پذیرفت(خوانندگان محترم می‌توانند برای اطلاع از این روش به مرجع [۲۳] مراجعه کنند) و از آنجا که در اثر هر لقاح دو فرزند ایجاد می‌شـــدند، مجموعه فرزندان، بهترین کروموزوم‌ها و کروموزوم‌های جهش یافته، جمعیت جدید الگوریتم ژنتیک را ایجاد نمودند.

برای پارامترهای احتمال باقی ماندن کروموزوم در تکرار بعد، احتمال جهش، احتمال ترکیب، تعداد جمعیت و تعداد تکرار آزمایش نیز پس از آزمایش‌های اولیه ۲ ســـطح انتخاب شـــدند. جدول زیر نتایج حاصـــل از این انتخاب‌ها را نشان می‌دهد.

جدول۲. بهترین پارامترهای به دست آمده برای حل به کمک GA

تعداد جمعیت	احتمال crossover	احتمال باقی ماندن کروموزوم	تعداد جمعیت	ماکزیمم تکرار
۵۰	۰/۶	۰/۲	۵۰	۵۰۰

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.1.1.6]

٤-٤ بهینه‌سازی توده ذرات

برای حل به کمک این الگوریتم، ابتدا ذراتی با سرعت و موقعیتی تصادفی در فضای حل ایجاد شد. در این مساله هر ذره یک توالی کامل از عملیات کارها منظور گردید و دامنه سرعت و موقعیت ذرات نیز از محدوده منفی مجموع تعداد کل عملیات تا مثبت تعداد کل عملیات لحاظ شد. از سوی دیگر به دلیل آنکه این الگوریتم برای فضای پیوسته ابداع شده است، برای تبدیل موقعیت ذرات در فضای پیوسته PSO به فضای گسسته، از رابطه نظیر به نظیر موقعیت تصادفی جدید ذره و توالی پیشین آن و همچنین مرتب کردن توالی‌ها به شکل صعودی بر مبنای اعداد موقعیت‌ها استفاده شد.

در این الگوریتم، ضرایب یادگیری فردی و اجتماعی و همچنین ضریب اینرسی، در طی اجرای الگوریتم، ثابت لحاظ شده‌اند. مقادیر انتخابی در جدول زیر آمده است.

جدول۳. بهترین پارامترهای به دست آمده برای حل به کمک PSO

ضریب اینرسی	تعداد جمعیت	ضریب یادگیری فردی	ضریب یادگیری اجتماعی	تعداد تکرار
۱/۲	۲۰	۱/۵	۲/۵	۵۰۰

٤-٥ پاسخ‌های به دست آمده از الگوریتم‌ها

پاسخ‌های به دست آمده از این ۴ الگوریتم در ۵ تکرار مورد بررسی قرار گرفته شده است. جدول ۴ مقادیر جواب بهینه، بهترین پاسخ از ۵ تکرار، میانگین و همچنین انحراف معیار آن‌ها را نشان می‌دهد.

همان‌طور که دیده می‌شود، در ۸ مساله حل شده به کمک الگوریتم جستجوی ممنوع، ۴ تا از مسایل توانسته‌اند به جواب بهینه دست یابند. علاوه‌براین به جز ۱ حالت انحراف معیارها با افزایش بعد مساله به شکل صعودی افزایش می‌یابند. همچنین واضح است که انحراف معیار مساله ۳×۳ برابر صفر شده است. بدین معنا که در تمامی اجراهای الگوریتم، به جواب بهینه رسیده‌اند. چنین چیزی می‌تواند به دلیل ابعاد کوچک مساله باشد.

الگوریتم SA نیز در ۴ مساله‌ای که روش جستجوی ممنوع در ۵ بار اجرا توانسته بود به جواب بهینه دست یابد، به جواب بهینه رسیده است، با این تفاوت که در ۳ مورد انحراف معیار برابر صفر شده است. بدین معنا که در ۳ دسته از این مسایل دقیقاً هر اجرا به جواب بهینه رسیده است. در حالی که در جستجوی ممنوع تنها در مساله ۳×۳ تمام تکرارها به جواب بهینه دست یافته‌اند. که البته از یک منظر می‌تواند قدرت الگوریتم SA را نسبت به TS در حل این دسته از موارد نشان دهد.

الگوریتم ژنتیک نیز در ۴ مساله توانسته است به حل بهینه دست یابد؛ اما تنها در ۲ دسته از این مسایل است که در هر ۵ بار اجرا به جواب بهینه رسیده است.

جدول٤. پاسخ‌های به دست آمده از الگوریتم‌ها

Name	Dimension (n×m)	Optimal	Cmax(TS)			Cmax(SA)			Cmax(GA)			Cmax(PSO)		
			Best	Avg	stDv	Best	Avg	stDv	Best	Avg	stDv	Best	Avg	stDv
	۳×۳	۲۲	۲۲	۲۲	۰	۲۲	۲۲	۰	۲۲	۲۲	۰	۲۲	۲۲	۰
MT۰۶	۶×۶	۵۵	۵۵	۵۸/۶	۱/۴۳	۵۵	۵۷	۱/۴۱	۵۵	۵۸/۲۵	۱/۴۶	۵۵	۵۷/۲۳	۱/۱۰۲

[Downloaded from jorlu.ir on 2026-11-29]

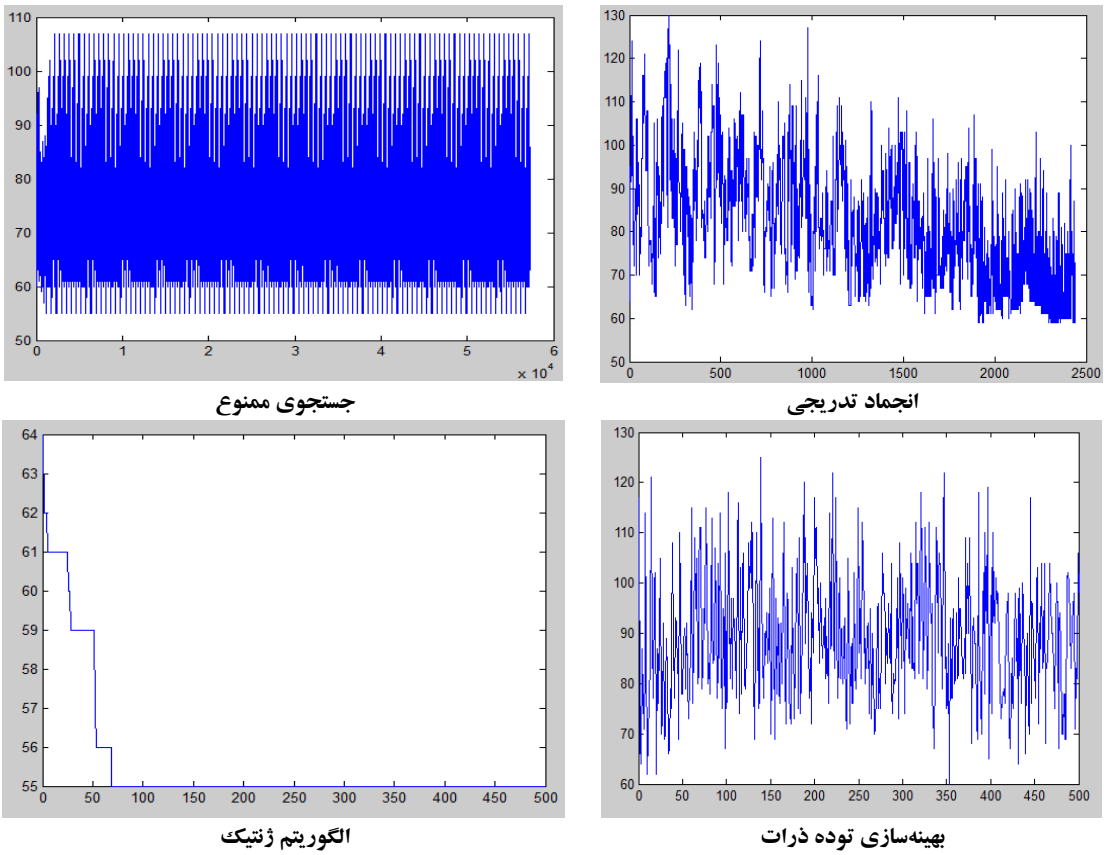
[DOR: 20.1001.1.22517286.2017.14.1.1.6]

Name	Dimension (n×m)	Optimal	Cmax(TS)			Cmax(SA)			Cmax(GA)			Cmax(PSO)		
			Best	Avg	stDv	Best	Avg	stDv	Best	Avg	stDv	Best	Avg	stDv
LA۰۱	۱۰×۵	۶۶۶	۶۸۹	۷۰۶/۲	۱۴/۰۷	۶۷۲	۶۹۵/۶	۱۲/۴۶	۶۷۵	۶۹۳/۲	۹/۴۹	۶۹۱	۷۱۶/۶	۱۱/۶۹
LA۰۶	۱۵×۵	۹۲۶	۹۲۶	۹۳۲/۳	۶/۰۶	۹۲۶	۹۲۶	۰	۹۲۶	۹۲۶	۰	۹۴۶	۹۶۴/۲	۱۱/۰۸
LA۱۱	۲۰×۵	۱۲۲۲	۱۲۲۲	۱۲۳۵/۲	۱۴/۸۹	۱۲۲۲	۱۲۲۲	۰	۱۲۲۲	۱۲۲۳/۲	۲/۴	۱۲۷۸	۱۲۸۳/۴	۳/۸۷
LA۲۱	۱۵×۱۰	۱۰۴۶	۱۲۱۷	۱۲۷۵/۲	۴۶/۹۶	۱۲۰۳	۱۲۳۴/۸	۳۸/۸۴	۱۱۶۴	۱۱۹۰	۳۴/۹۶	۱۴۰۳	۱۴۳۲/۲	۲۴/۶۴
LA۲۶	۲۰×۱۰	۱۲۱۸	۱۴۶۰	۱۵۰۶/۲	۳۶/۸۴	۱۳۹۱	۱۴۳۶/۴	۲۶/۳۳	۱۳۵۶	۱۴۰۰	۲۸/۲۳	۱۵۹۹	۱۶۴۱/۶	۲۸/۰۴
LA۳۱	۳۰×۱۰	۱۷۸۴	۱۸۸۱	۱۹۶۷/۲	۵۶/۷۵	۱۸۹۶	۱۹۶۳/۸	۴۲/۴۲	۱۸۴۰	۱۸۶۷/۴	۲۴/۵۶	۲۱۵۵	۲۲۰۵	۲۹/۷۹

از جدول۴ وا ضح ا ست، در ۵ اجرایی که الگوریتم PSO نیز دا شته ا ست، ۲ م ساله توانسته‌اند به حل بهینه د ست یابند. در اولین مساله نیز در هر ۵ تکرار به جواب بهینه رسیده است.

٤-٦ تمام جستجوها

شکل‌های زیر تمام جستجوهای یک عنصر را در تکرارهای مختلف حل مساله MT06 نشان می‌دهد:



شکل ۱. تمام جستجوهای موجود در روند اجرای الگوریتم

همان‌طور که پیش از این گفته شـد، در جسـتجوی ممنوع به تمام جابه‌جایی‌های دو عملیات مختلف برای دو کار متفاوت اقدام شده است؛ لذا عملیات کاوش در همسایگی‌ها به شدت دیده می‌شود.

از شکل انجماد تدریجی نیز واضح است که با توجه به روند اتخاذ شده، در مراحل ابتدایی کاوش الگوریتم

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.1.1.6]

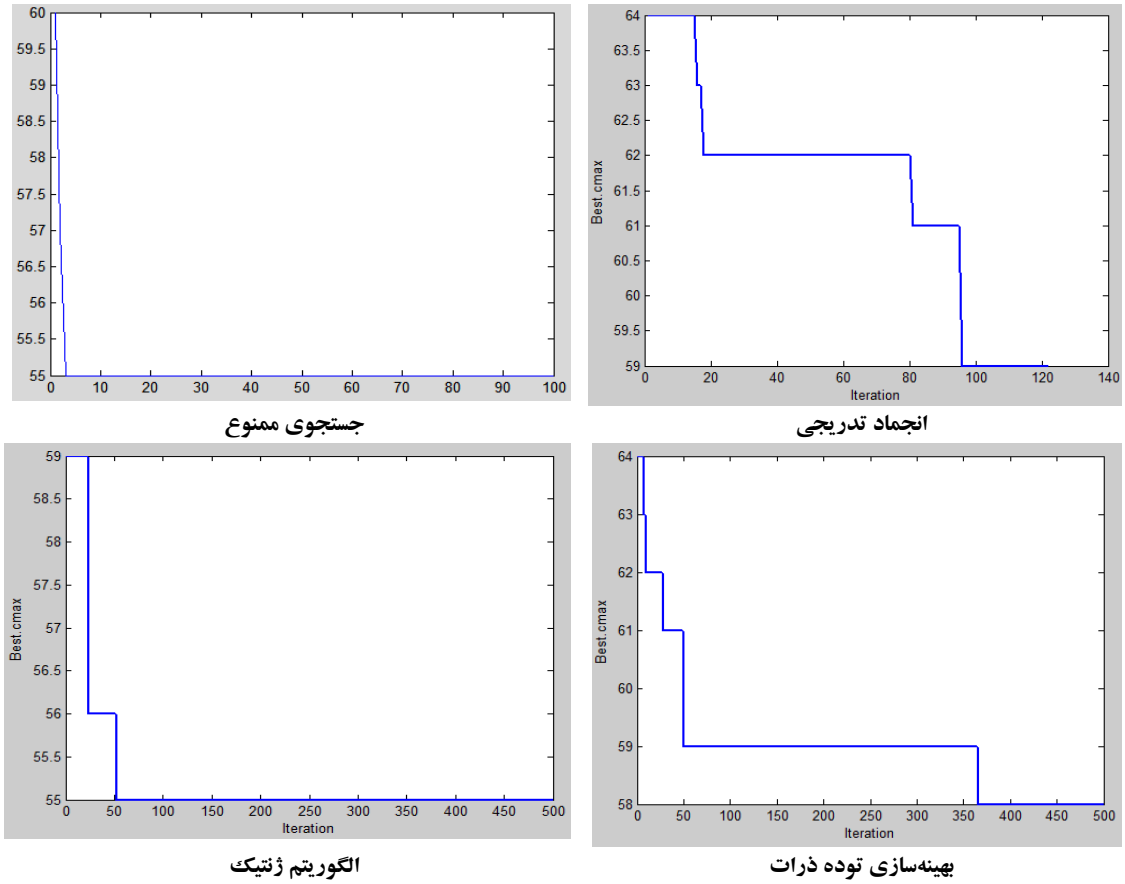
و در مراحل انتهایی تشدید بیشتر است.

در مورد الگوریتم ژنتیک به کار رفته نیز یکی از کروموزوم‌ها انتخاب شده و زمان تکمیل انجام کارهای آن در طی الگوریتم در نظر گرفته شـــد. همان‌طور که دیده می‌شـــود با توجه به اینکه بهترین کروموزوم‌ها برای تکرارهای بعدی انتخاب شده‌اند، روند جستجوها به شکلی است که تمرکز بیشتر بر روی تشدیدها صورت پذیرفته است. به طوری که این کروموزوم بین تکرارهای ۵۰ تا ۱۰۰ توانسته است، به مقدار بهینه مساله که برابر ۵۵ است دست یابد.

در استفاده از الگوریتم PSO نیز باید به خاطر داشت که انتخاب پارامترها بسیار حساس هستند. نمودار مربوط به PSO نیز تمام جستجوهای یکی از ذرات را در طی اجرای این الگوریتم برای مساله MT06 نشان داده است.

٤-٧ روند دستیابی به بهترین جواب

شـــکل‌های زیر روند دســـتیابی به بهترین جواب را در تکرارهای مختلف حل مســـاله MT06 با همان ذرات و کروموزوم‌ها نشان می‌دهد:



شکل ۲. روند دستیابی به بهترین جواب

شکل ۲ نشان می‌دهد که مسیر الگوریتم انجماد تدریجی در روند بهبود تقریباً یک حالت پلکانی منظم را داراست. همچنین این شـکل نشـان می‌دهد که سـایر کروموزوم‌هایی که در الگوریتم ژنتیک مورد اسـتفاده قرار گرفته‌اند،

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.1.1.6]

سریعتر از کروموزوم انتخاب شده که در شکل ۱ روند جستجوهایش نشان داده شد، به مقدار بهینه دست یافته‌اند. علاوه‌براین در شکل مربوط به بهینه‌سازی توده ذرات نمایان است که در تکرارهای ابتدایی بیشترین بهبودها حاصل شده است و طی یک دوره ثبات، دوباره نزولی در مقدار Cmax کاهش یافته است.

٤-٨ مدت زمان اجرا

این الگوریتم‌ها از منظر مدت زمان اجرا نیز مورد بررسی قرار گرفتند. زمان‌های حاصل از حل روی کامپیوتر Pentium(R) 4 CPU 2.8 GHz, 1.75 GB of RAM در جدول زیر نشان داده شده‌اند.

جدول ۵. زمان حل JSP به کمک الگوریتم‌های استفاده شده

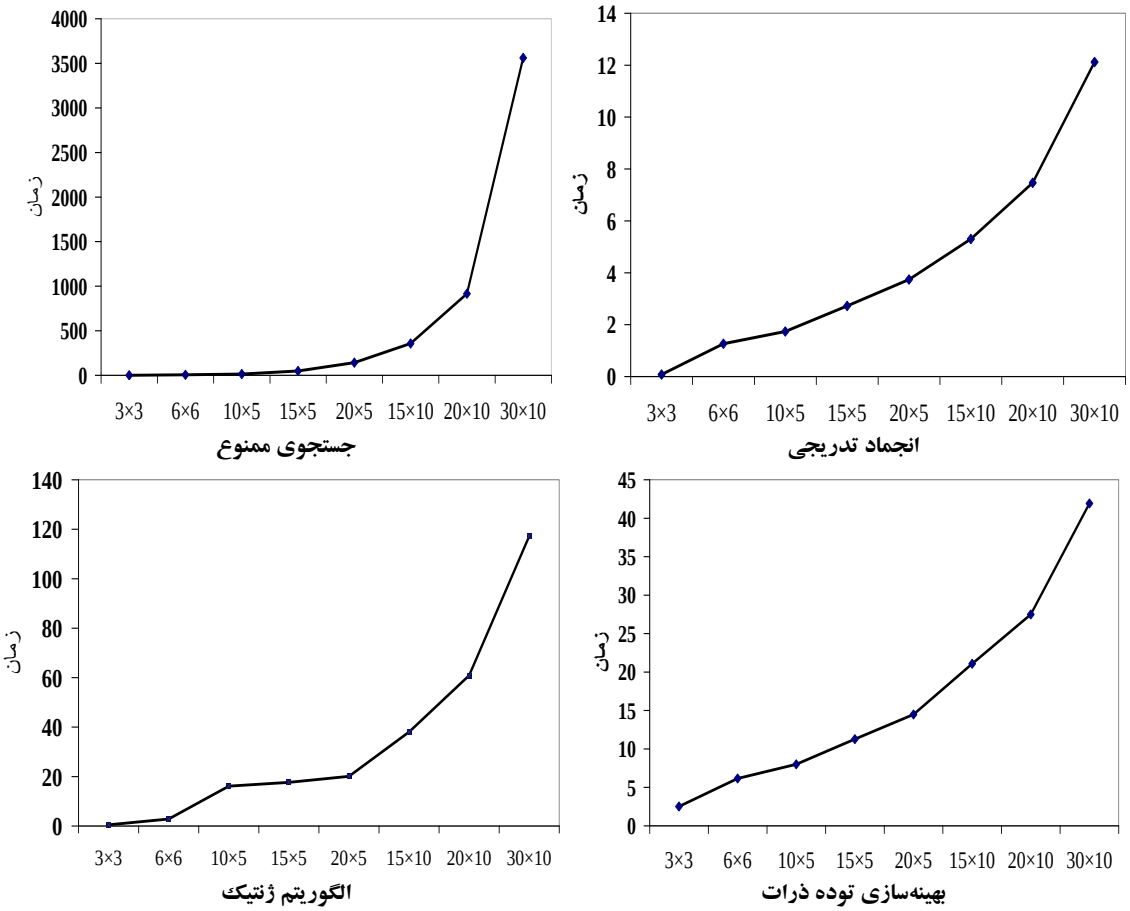
Name	Dimension (n×m)	Elapsed time (sec)			
		TS	SA	GA	PSO
	۳×۳	۰/۲۲۳۳	۰/۰۷۷	۰/۴۹	۲/۵۱۸۱
MT۰۶	۶×۶	۵/۶۹	۱/۲۶۴	۲/۸۱	۶/۱۵۴۶
LA۰۱	۱۰×۵	۱۳/۵۸۶	۱/۷۳۷	۱۶/۱۲	۸/۰۰۵۴
LA۰۶	۱۵×۵	۴۸/۴۸	۲/۷۲۲	۱۷/۶۲	۱۱/۲۷۳
LA۱۱	۲۰×۵	۱۴۱/۴۰۴	۳/۷۳۸	۲۰/۰۸۲	۱۴/۴۸۰
LA۲۱	۱۵×۱۰	۳۵۷/۳۶۲	۵/۲۹۹	۳۸/۰۰	۲۱/۰۶۵
LA۲۶	۲۰×۱۰	۹۱۵/۴۶۲	۷/۴۶۷	۶۰/۸۵	۲۷/۴۸۹
LA۳۱	۳۰×۱۰	۳۵۵۹/۹۸	۱۲/۱۲	۱۱۷/۱۸	۴۱/۹۱۲

همان‌طور که از جدول فوق مشخص است، کم‌ترین زمان‌های حل با توجه به پارامترهای استفاده شده، مربوط به الگوریتم SA می‌باشد، به طوری که برای مسایل با ابعاد بالاتر به طور مثال، مساله LA31 اثر این زمان‌ها را به راحتی می‌توان دید. به گونه‌ای که روش جستجوی ممنوع نسبت به انجماد تدریجی بیشتر از ۲۹۳ برابر، زمان نیاز دارد و علی رغم آنکه بهترین جواب به دست آمده از TSدر ۵ بار اجرا اندکی از SA بهتر است؛ اما در مجموع میانگین و انحراف معیار بیشتری نسبت به این الگوریتم داراست. پس چنانچه بخواهیم از نظر میانگین، انحراف معیار و زمان مورد نیاز بسنجیم، الگوریتم SA به طور قطع می‌تواند برای ابعاد بالاتر نسبت به TS بهتر باشد.

شکل‌های زیر زمان حل مسایل مختلف را به کمک الگوریتم‌های مورد نظر نشان می‌دهد. در این شکل‌ها واضح است که زمان اجراهای الگوریتم‌های انجماد تدریجی و بهینه سازی توده ذرات شباهت زیادی به یکدیگر دارند. شکل مربوط به جستجوی ممنوع بیانگر آن است که در ابعاد کم تقریباً یک روال مشاهده می شود؛ اما با افزایش ابعاد به شدت زمان حل زیاد می‌شود.

[Downloaded from jorlu.ir on 2026-11-29]

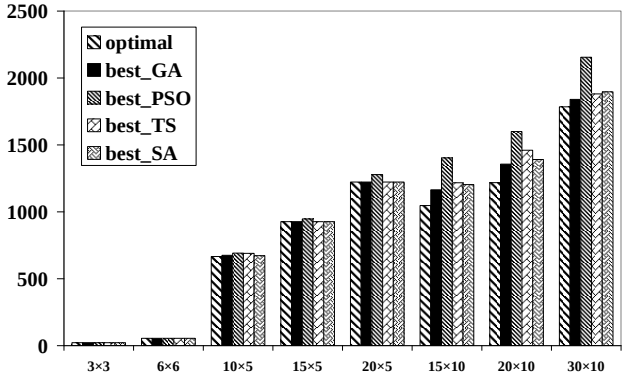
[DOR: 20.1001.1.22517286.2017.14.1.1.6]



شکل ۳. مقایسه زمانی الگوریتم‌ها با توجه به ابعاد مساله

٤-۹ مقایسه بهترین پاسخ‌های به دست آمده در تعداد اجراها یکسان

بهترین پاسخ‌های به دست آمده در ۵ تکرار مختلف الگوریتم در نمودار زیر خلاصه شده است.



شکل ٤. مقایسه بهترین پاسخ‌های به دست آمده

از این شکل واضح است که دو مساله اول که دارای ابعاد کم‌تری می‌باشند، تمام الگوریتم‌ها توانسته‌اند به پاسخ بهینه دست یابند. نکته جالب توجه در این ۵ بار تکرار آنست که بهترین جواب مساله ۱۰×۵ در هیچ یک از الگوریتم‌ها به مقدار بهینه نرسیده است؛ در حالی که در مسایل ۱۵×۵ و ۲۰×۵ که از نظر ابعاد بزرگ‌تر می‌باشند به

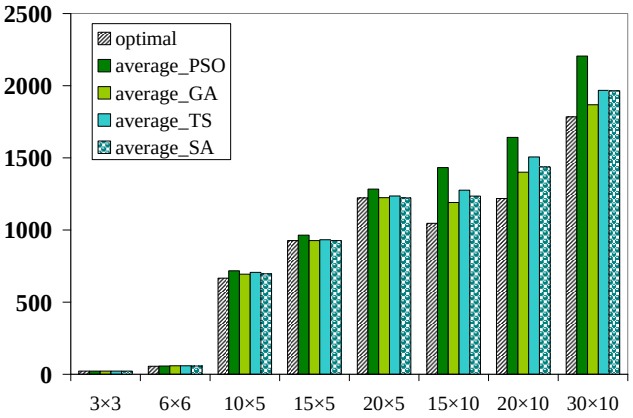
[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.1.1.6]

جز الگوریتم PSO سـایر الگوریتم‌ها توانسـته‌اند به جواب بهینه دسـت یابند. علاوه‌براین از نمودار فوق مشـخص است که PSO با ساختاری که برای آن در نظر گرفته شده است، نسبت به سایر الگوریتم‌ها عملکرد ضعیف‌تری را از خود نشان داده است. در مقابل الگوریتم ژنتیک، انجماد تدریجی و جستجوی ممنوع به ترتیب از منظر بهترین پاسخ‌های به دست آمده در اولویت‌های بهتری قرار گرفته‌اند.

٤-١٠ مقایسه میانگین پاسخ‌های به دست آمده به کمک پارامترهای تعیین شده

چنانچه بخواهیم میانگین پاسـخ‌های به دسـت آمده از الگوریتم‌ها را با هم مقایسـه نماییم واضـح خواهد بود که الگوریتم PSO نسبت به سایر الگوریتم‌ها دارای میانگین بالاتری است و از آنجا که تابع هدف حداقل‌سازی است می‌تواند دلیل دیگری باشد که این الگوریتم نسبت به سایر الگوریتم‌های به کارگرفته شده ضعیف‌تر عمل می‌کند. همچنین از این شـکل مشـخص اسـت که الگوریتم ژنتیک نسـبت به بقیه روش‌های فراابتکاری اسـتفاده شـده بهتر عمل کرده است.



شکل ۵. مقایسه میانگین پاسخ‌های به دست آمده

٤-١١ مقایسه پاسخ‌های به دست آمده در یک زمان مشترک

از آنجا که هر یک از الگوریتم‌ها به زمان متفاوتی برای اجرا نیاز داشـتند، تصـمیم گرفته شـد تا در یک زمان مشترک و در یک تکرار، کیفیت پاسخ‌های به دست آمده را بررسی نماییم. جدول زیر نتایج این بررسی را برای مساله مورد نظر نشان می‌دهد.

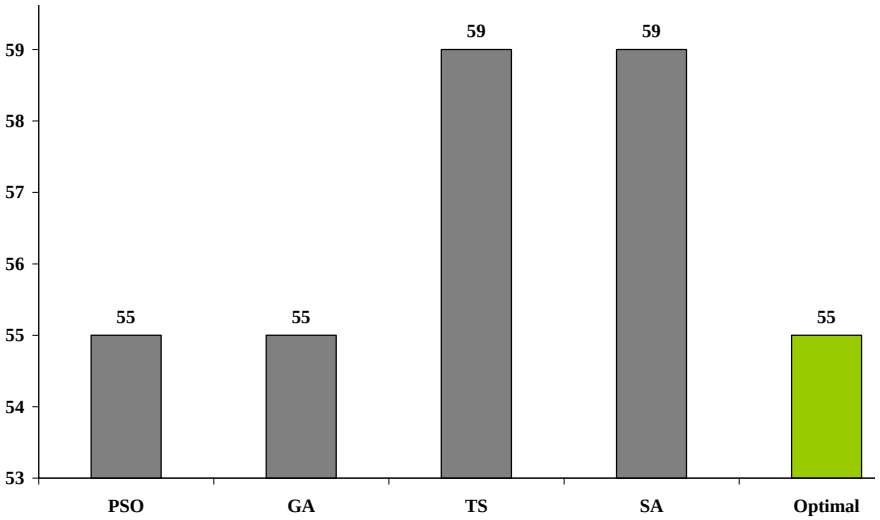
۵ نتیجه‌گیری

در این مقاله سـعی شـد، الگوریتم‌های ژنتیک، جسـتجوی ممنوع، انجماد تدریجی و بهینه‌سـازی توده ذرات برای حل مسـاله زمان‌بندی تولید کارگاهی اسـتفاده شـده و نتایج آن با هم مقایسـه گردد. نتایج نشـان می‌داد روش جستجوی ممنوع نسبت به سایر الگوریتم‌ها نیاز به زمان بیشتری داشت؛ اما جواب بهتری نسبت به بهینه‌سازی توده ذرات ایجاد می‌نمود. علاوه‌براین الگوریتم ژنتیک یکی از بهترین الگوریتم‌هایی بود که برای حل این مسـایل می‌توانست مورد استفاده قرار گیرد؛ چرا که هم از نظر بهترین جواب به دست امده و هم از نظر میانگین پاسخ‌های

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.1.1.6]

به دست آمده در موقعیت بهتری نسبت به سایر الگوریتم‌ها قرار داشت در عین آنکه در زمان معقولی می‌توانست به این پاسخ‌ها دست یابد.



شکل ٦. مقایسه پاسخ‌های به دست آمده در یک زمان مشترک

برای الگوریتم بهینه‌سازی توده ذرات با توجه به اینکه نسبت به جستجوی ممنوع به مراتب به زمان کم‌تری نیاز مند است چنانچه بتوان ساختارهای دیگری به جز آنچه به کار رفت را مورد استفاده قرار داد، شاید بتوان به پاسخ‌های بهتری دست یافت؛ لذا برای تحقیقات آتی می‌توان پیشنهاد داد ساختارهای دیگری از الگوریتم‌ها را به کار گرفت تا بتوان از مزایای آنها به شکل بهتری استفاده نمود.

منابع

[۲۲] بهشتی نیا، م.ع.، حسنی بیدگلی، م.، (۱۳۹۵). ترکیبی جدید از الگوریتم‌های فراابتکاری ژنتیک و شبیه‌سازی تبرید برای حل مساله زمان‌بندی کارگاهی انعطاف پذیر. مجله تحقیق در عملیات در کاربردهای آن، ۱۳(۴)، ۳۷-۲۱.

[۲۳] فتاحی، پ.، (۱۳۸۸). الگوریتم‌های فرا ابتکاری. انتشارات دانشگاه بوعلی سینا، چاپ اول، ۱۳۸۸.

[1] Wang, L., Zheng, D. Z., (2001). An effective hybrid optimization strategy for job-shop scheduling problems. Computers & Operations Research, 28(6), 585–596.

[2] Van Laarhoven, P. J. M., Aarts, E. H. L., Lenstra, J. K., (1992). Job shop scheduling by simulated annealing. Operations Research, 40, 113–125.

[3] Aydin, M. E., Fogarty, T. C., (2004). A simulated annealing algorithm for multi-agents systems: A job shop scheduling application. Journal of intelligent manufacturing, 15(6), 805-814.

[4] Elmi, A., Solimanpur, M., Topaloglu, S., Elmi, A., (2011). A simulated annealing algorithm for the job shop cell scheduling problem with intercellular moves and reentrant parts. Computers & Industrial Engineering, 61(1), 171-178.

[5] Zhang, C. Y., Li, P. G., Guan, Z. L., Rao, Y. Q., (2007). A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem. Computers & Operations Research, 34, 3229–3242.

[6] Dell'Amico, M., Trubian, M., (1993). Applying Tabu search to the job shop scheduling problem. Ann. Ops Res., 40, 231-252.

[7] Brucker, P., Neyer, J., (1998). Tabu Search for the Multi-Mode Job Shop Problem. OR-Spektrum, 20, 21-28.

[8] Davis, L., (1985). Job Shop Scheduling with Genetic Algorithms. Proceedings of the 1st International Conference on Genetic Algorithms, Pittsburgh, PA, 136-140.

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.1.1.6]

[9] Cheng, R., Gen, M., Sujimura, Y., (1996). A tutorial survey of job-shop scheduling problems using genetic algorithms-I. Representation. Computers & Industrial Engineering, 30(4), 983–997.
[10] Zhang, R., Wu, C., (2012). Bottleneck machine identification method based on constraint transformation for job shop scheduling with genetic algorithm. Information Sciences, 188(1), 236-252.
[11] Zhou, H., Feng, Y., Han, L., (2001). The Hybrid Heuristic Genetic Algorithm for Job Shop Scheduling. Computers & Industrial Engineering, 40, 191-200.
[12] Sha, D. Y., Hsu, C. Y., (2006). A hybrid particle swarm optimization for job shop scheduling problem. Computers & Industrial Engineering, 51, 791–808.
[13] Xia, W. J., Wu, Z. M., (2006). A Hybrid Particle Swarm Optimization Approach for the Job-Shop Scheduling Problem. International Journal of Advanced Manufacturing Technology, 29, 360-366.
[14] Mencía, R., Sierra, M. R., Mencía, C., Varela, R., (2015). Memetic algorithms for the job shop scheduling problem with operators. Applied Soft Computing, 34, 94–105.
[15] Gao, L., Li, X., Wen, X., Lu, C., Wen, F., (2015). A hybrid algorithm based on a new neighborhood structure evaluation method for job shop scheduling problem. Computers & Industrial Engineering, 88, 417–429.
[16] Asadzadeh, L., (2015). A local search genetic algorithm for the job shop scheduling problem with intelligent agents. Computers & Industrial Engineering, 85, 376–383.
[17] Asadzadeh, L., (2016). A parallel artificial bee colony algorithm for the job shop scheduling problem with a dynamic migration strategy. Computers & Industrial Engineering, In Press, Corrected Proof.
[18] Scaria, A., George, K., Sebastian, J., (2016). An Artificial Bee Colony Approach for Multi-Objective Job Shop Scheduling. Procedia Technology 25, 1030–1037.(1st Global Colloquium on Recent Advancements and Effectual Researches in Engineering, Science and Technology - RAEREST 2016 on April 22nd & 23rd April 2016).
[19] Mirshekarian, S., Šormaz, D. N., (2016). Correlation of job-shop scheduling problem features with scheduling efficiency. Expert Systems with Applications, 62, 131–147.
[20] Kuhpfahl, J., Bierwirth, C., (2016). A study on local search neighborhoods for the job shop scheduling problem with total weighted tardiness objective. Computers & Operations Research, 66, 44–57.
[21] Akram, K., Kamal, Kh., Zeb, A., (2016). Fast simulated annealing hybridized with quenching for solving job shop scheduling problem. Applied Soft Computing, 49, 510–523.

[Downloaded from jorlu.ir on 2026-11-29]

[DOR: 20.1001.1.22517286.2017.14.1.1.6]