مجله تحقیق در عملیات در کاربردهای آن،سال هجدهم،شماره دوم (پیاپی ۶۹)، تابستان ۰۰، ص ص ۱۲۴-۱۰۷
شاپا۷۲۸۶-۲۲۵۱

روش ترکیبی تاگوچی و تحلیل پوششی داده‌ها برای تعیین پارامترها و عملگرهای الگوریتم‌های فراابتکاری- الگوریتم ژنتیک برای حل مساله جریان کارگاهی جایگشتی دوباره واردشونده

مائده فصیحی[۱]، سید اسماعیل نجفی[۲*]، رضا توکلی مقدم[۳]، مصطفی حاجی آقائی کشتلی[۴]

۱- دانشجوی دکتری، گروه مهندسی صنایع، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران

۲- استادیار، گروه مهندسی صنایع، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران

۳- استاد، دانشکده مهندسی صنایع، پردیس دانشکده‌های فنی، دانشگاه تهران، تهران، ایران

۴- استادیار، گروه مهندسی صنایع، دانشگاه علم و فناوری مازندران، بهشهر، ایران

رسید مقاله: ۱۸ آذر ۱۳۹۸

پذیرش مقاله:۲۴ دی ۱۳۹۹

چکیده

کارایی الگوریتم‌های فراابتکاری ارتباط مستقیمی با تنظیم پارامترهای آن دارد، به طوری که انتخاب نادرست پارامترهای الگوریتمی کاملاً کارا، باعث ناکارآمدی آن می‌گردد. در این تحقیق ترکیب روش طراحی آزمایشات به روش تاگوچی و روش تحلیل پوششی داده‌ها جهت بالا بردن کارایی الگوریتم ژنتیک برای حل مساله جریان کارگاهی جایگشتی دوباره واردشونده به‌کار گرفته می‌شود. سناریوهای مختلفی جهت انتخاب اپراتورهای الگوریتم ژنتیک برای واحدهای تحت ارزیابی شکل می‌گیرند. در ابتدا با استفاده از روش تاگوچی برای هر واحد، پارامترهای بهینه با هدف کمینه‌سازی تابع هدف (حداکثر دیرکرد کارها) مشخص شده، سپس واحدهای کارا جهت تعیین بهترین عملگرهای الگوریتم با توجه به بهینه تابع هدف در کم‌ترین زمان ممکن، تعیین شده و رتبه‌بندی می‌گردند. این تحقیق می‌تواند به عنوان روشی برای تنظیم پارامترهای دیگر الگوریتم‌های تکاملی و فراابتکاری با هدف اجتناب از معایب مربوط به روش‌های آزمایش و خطا به‌کار گرفته شود.

کلمات کلیدی: زمان‌بندی، جریان کارگاهی جایگشتی دوباره واردشونده، پارامترها و عملگرهای الگوریتم ژنتیک، تحلیل پوششی داده‌ها، طراحی آزمایشات تاگوچی، مدل رتبه‌بندی اندرسن-پترسن.

۱ مقدمه

زمان‌بندی تولید عبارت است از تعیین توالی زمانی و تخصیص سفارشات مشتریان به منابع موجود تولید (اعم از پرسنل، ماشین‌ها، ابزار و غیره) به منظور انجام مجموعه‌ای از عملیات‌های مربوطه. معمولاً زمان‌بندی با توجه به

* عهده‌دار مکاتبات

آدرس الکترونیکی: najafi1515@yahoo.com

[Downloaded from jorlu.ir on 2026-11-28]

[DOR: 20.1001.1.22517286.2021.18.2.2.7]

[DOI: 10.52547/jamlu.18.2.107]

اهدافی نظیر: کمینه‌سازی زمان کار در جریان ساخت و موجودی کار در جریان ساخت، دستیابی به موعدهای تعهدشده، بیشینه‌سازی خروجی و بهره‌برداری بیشتر از مراکز کاری انجام می‌شود. بسیاری از مسایل زمان‌بندی که تمام مفروضات دنیای واقعی را درنظر می‌گیرند، از نظر محاسباتی بسیار پیچیده شده، به طوری که در کلاس مسایل NP-Hard قرار می‌گیرند، که باعث می‌شود یافتن حل بهینه برای موارد با اندازه بزرگ‌تر در یک زمان معقول، غیرممکن گردد. در این‌گونه موارد، یکی از راه‌های یافتن حل قابل قبول استفاده از الگوریتم‌های فراابتکاری مانند الگوریتم ژنتیک (GA)، شبیه‌سازی تبرید (SA)، جستجوی ممنوع (TS) و نظیر آن می‌باشد. از آنجا که در استفاده از الگوریتم‌های فراابتکاری، تنظیم پارامترهای ورودی آنها همیشه یک مشکل چالش برانگیز در بهبود کیفیت حل‌های ناشی از آنها می‌باشد، در تحقیقات اخیر تنظیم پارامترهای این الگوریتم‌ها با روش‌هایی مانند آزمایش و خطا، طراحی آزمایشات، تاگوچی و غیره مورد توجه قرار گرفته است.

به منظور کالیبره کردن الگوریتم‌ها، روش‌های مختلفی برای تحقیق تجربی طراحی آماری وجود دارد. با این حال، رویکرد غالب مورد استفاده و روش جامع، یک آزمایش فاکتوریل کامل است [۱-۳]. این رویکرد همیشه کارآمد نیست زیرا زمانی‌که تعداد عامل‌ها به میزان قابل توجهی افزایش یابند، بررسی دشوار می‌گردد. برای کاهش تعداد آزمایش‌های مورد نیاز، آزمایش‌های فاکتوریل کسری[1] (FFEs) توسعه داده شد [۴]. FFE تنها بخشی از کل ترکیبات ممکن را برای تخمین اثر اصلی عوامل و برخی از تعاملات آنها را در نظر می‌گیرد. تاگوچی [۵] یک خانواده از ماتریس‌های FFE را ایجاد کرد که در نهایت تعداد آزمایشات را کاهش داد، با این حال اطلاعات کافی ارایه می‌دهند. در روش تاگوچی، آرایه‌های متعامد برای مطالعه تعداد زیادی متغیر تصمیم با تعداد کمی از آزمایش‌ها مورد استفاده قرار می‌گیرند. چنگ و چانگ [۶] یک الگوریتم ژنتیک برای بهینه‌سازی مساله زمان‌بندی جریان کارگاهی و با استفاده از طراحی آزمایشی تاگوچی جهت پیش‌بینی یک برنامه ترکیبی برای پارامترهای بهینه الگوریتم پیشنهادی ارایه کردند. آنها هفت فاکتور از الگوریتم پیشنهادی، شامل حل اولیه، روش انتخاب، روش تقاطع، نرخ جهش، تعداد جمعیت، نرخ تقاطع، و روش جهش را آزمایش و بهینه‌سازی کردند. غلامی و همکاران [۷] با درنظر گرفتن فرض توالی وابسته به زمان راه اندازی و شکست‌های تصادفی ماشین‌ها، یک الگوریتم ژنتیک برای حل مساله زمان‌بندی جریان کارگاهی را توسعه دادند. آنها برای تنظیم پارامترهای الگوریتم، از روش تاگوچی استفاده می‌کردند. نادری و همکاران [۸] همچنین برای تنظیم پارامترهای الگوریتم SA که برای حل زمان‌بندی جریان کارگاهی تلفیقی توسعه داده شده از روش تاگوچی با فرض توالی وابسته به زمان آماده‌سازی استفاده کردند. آنها فاکتورهایی نظیر حل اولیه، الگوی سرد شدن، زمان‌بندی سرد شدن، درجه حرارت اولیه، تعداد جستجوهای همسایگی در هر درجه حرارت، دماهای اولیه و توقف همچنین ساختار جستجو همسایگی به عنوان فاکتور قابل کنترل SA و با انتخاب سطوح مختلف برای هر یک از این فاکتورها، مقادیر بهینه را با استفاده از روش تاگوچی پیدا کردند. زندیه و رومانی [۹] برای تعیین مقادیر پارامترهای الگوریتم‌های فراابتکاری جغرافیای زیستی و ژنتیک جهت حل مساله زمان‌بندی و قبول سفارش در

[1] Fractional Factorial Experiments (FFEs)

[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]

محیط تک ماشینه از روش تاگوچی استفاده کردند. آزاده و همکاران [۱۰] برای دستیابی به بهترین حل مساله مسیریابی موجودی با حمل و نقل تک محصول فاسد شدنی، پارامترهای الگوریتم پیشنهادی را با استفاده از روش تاگوچی تعیین کردند.

هاردز و همکاران [۱۱] به منظور حل مساله تخصیص عناصر در مونتاژ تخته مدار چاپی از الگوریتم ژنتیک استفاده کردند. سپس، به ترتیب برای حل مساله تعیین توالی، شامل ۱۰ عنصر تعبیه‌شده در یک ماشین، آنها طرح‌های آزمایشی را جهت تعیین بهترین نوع عملگر تقاطع، نوع عملگر تولید مجدد، نرخ تقاطع و نرخ تولید مجدد به کار بردند.

روش دیگر برای تنظیم پارامترهای الگوریتم‌های فراابتکاری، روش سطح پاسخ (RSM) است. RSM ترکیبی از روش‌های ریاضی و آماری است که مناسب برای مدل‌سازی و تجزیه و تحلیل مسایلی است که در آن متغیر پاسخ توسط چندین متغیر ورودی تحت تاثیر قرار می‌گیرد، که هدف آن بهینه‌سازی پاسخ‌هاست. در میان تحقیقات اخیر تنظیم پارامترهای الگوریتم‌های فراابتکاری، وانگ و وو [۱۲] از فرآیند شش مرحله‌ای بر اساس RSM استفاده کرده که در آن برای شناسایی و بهینه‌سازی پارامترهای الگوریتم SA با درنظرگرفتن محدودیت‌های زمان محاسباتی توسعه داده شد.

روش تحلیل پوششی داده‌ها[1] (DEA)، مجموعه‌ای به نام امکان تولید را ساخته و با درنظر گرفتن تخمینی از تابع تولید به عنوان مرز آن، سطوح کارایی را برای هر یک از ورودی‌ها و خروجی‌ها مشخص می‌نماید [۱۳]. در ارزیابی کارایی نسبی واحدهای تصمیم‌گیرنده به کمک مدل‌های DEA، معمولا بیش از یک واحد تصمیم‌گیرنده کارا شناسایی می‌گردد و رتبه‌بندی این واحدهای تصمیم‌گیرنده کارا از اهمیتی خاص برخوردار است [۱۴]. قابلیت رتبه‌بندی واحدها در این روش می‌تواند در تنظیم پارامترهای الگوریتم‌های فراابتکاری به کار گرفته شود.

این مقاله روی تنظیم پارامترها و عملگرهای الگوریتم ژنتیک برای حل یک مساله جریان کارگاهی جایگشتی دوباره‌شونده[2] (RPFS) m-ماشینه با هدف کمینه‌سازی حداکثر دیرکرد کارها (Tmax) متمرکز می‌شود که توسط فصیحی و همکاران [۱۵] مورد مطالعه قرار گفت. جهت بالا بردن کارایی الگوریتم ژنتیک برای حل مساله RPFS، روش ترکیبی تاگوچی و تحلیل پوششی داده‌ها به‌کار گرفته می‌شود. قابل ذکر است که به‌کارگیری روش تحلیل پوششی داده‌ها در کنار روش تاگوچی امکان در نظر گرفتن عامل زمان را علاوه بر تابع هدف برای تعیین بهترین عملگر و پارامترهای الگوریتم حل فراهم می‌نماید، از آنجایی‌که تاگوچی روشی چند پاسخی نیست.

این مقاله بدین صورت سازمان‌دهی شده است: مساله مورد مطالعه در بخش ۲ شرح داده می‌شود. در بخش ۳ الگوریتم ژنتیک برای حل مساله زمان‌بندی جریان کارگاهی جایگشتی دوباره وارد شونده (RPFS) شرح داده می‌شود. پارامترها و عملگرهای الگوریتم ژنتیک با استفاده از طراحی آزمایشات تاگوچی و تحلیل پوششی داده‌ها برای مساله مورد نظر در بخش ۴ تعیین می‌گردد. نتیجه‌گیری در آخرین بخش آورده می‌شود.

[1] Data Envelopment Analysis (DEA)
[2] Reentrant Permutation Flow Shop (RPFS)

[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]

۲ شرح مساله و مفروضات

کارگاه دوباره واردشونده یک نوع جدید کارگاه ساخت است. ملاقات ماشین‌های مشخصی بیش از یک‌بار یک کار، از خصوصیت اساسی یک کارگاه دوباره وارد شونده است [۱۶]. جریان کارگاهی دوباره واردشونده به این معنی است که n کار روی m ماشین در یک توالی معین پردازش می‌شوند و توالی هر کار روی ماشین‌ها به صورت $M_1, M_2, ..., M_m, M_1, M_2, ..., M_m, ..., and\ M_1, M_2, ..., M_m$ می‌باشد. هر کار می‌تواند به چندین سطح که روی M_1 شروع و در M_m پایان می‌پذیرد تجزیه شود. RFS در بسیاری از سیستم‌های تولیدی، به خصوص در صنایع با تکنولوژی بالا وجود دارد. مثلا در ساخت نیمه رساناها، روند پردازش به طور زیادی دوباره واردشونده است [۱۷] و [۱۸].

اخیراً سیستم‌های ساخت با خطوط دوباره وارد شونده در جامعه علمی بسیار مورد توجه قرار گرفته‌اند. گریوز و همکاران [۱۹] برای نخستین بار مساله دوباره وارد شونده را در سال ۱۹۸۳ مورد مطالعه قرار دادند که در صنعت الکترونیک به کار گرفته می‌شود. با توجه به اینکه کاهش زمان دیرکرد نسبت به موعد تحویل کالا از دید مشتری مهم بوده و به نوعی به تولید بهنگام (JIT) اشاره دارد، زمان‌بندی RPFS چند ماشینه در محیط قطعی با هدف کمینه‌سازی حداکثر دیرکرد کارها مورد توجه می‌باشد. از آنجایی‌که مساله زمان‌بندی جریان کارگاهی دو ماشینه با هدف کمینه‌سازی دیرکرد به عنوان یک NP-hard شناخته شده است [۲۰]، مساله مورد بحث در این تحقیق به طور قطع NP-hard است. بنابراین الگوریتم فرااِبتکاری برای حل آن به کار گرفته می‌شود.

برای مسایل زمان‌بندی RPFS، مفروضات در نظر گرفته شده بدین صورت می‌باشد: هر کاری ممکن است ماشین‌های مشخصی را بیش از یک بار ملاقات کند. ترتیب ماشین برای هر n کار و ترتیب کار برای هر m ماشین در هر سطح یکسان است. هر دو عملیات متوالی یک کار باید روی ماشین‌های متفاوت پردازش شوند. زمان‌های پردازش مستقل از توالی و قطعی می‌باشند. همه اطلاعات شناخته و ثابت شده‌اند. همه‌ی کارها برای پردازش در زمان صفر که ماشین‌ها بیکارند آماده‌اند. بریدگی به هیچ عنوان جایز نیست. وقتی که یک عملیات شروع می‌شود، باید قبل از عملیات دیگری که می‌تواند روی آن ماشین شروع شود کامل شود. ماشین‌ها هرگز خراب نمی‌شوند و در تمام مدت زمان‌بندی در دسترس‌اند. محدودیت‌های تکنولوژیکی شناخته شده‌اند. از هر نوع ماشین فقط یکی وجود دارد. برای انجام کارها، فضای انتظار نامحدود وجود دارد.

۲-۱ تولید مسایل نمونه

تعداد کارها (n)، تعداد ماشین‌ها (m) و تعداد سطوح (L)، اندازه مساله ($n \times m \times L$) اندازه مساله RPFS را تعیین می‌نمایند. در این تحقیق مسایل آزمایشگاهی به دو دسته طبقه‌بندی می‌شوند: مسایل کوچک، مسایل بزرگ. انواع مسایل کوچک شامل ۳×۳×۳، ۴×۴×۴، ۵×۴×۳، ۵×۵×۴، ۶×۸×۵، ۷×۸×۴، ۸×۸×۴، ۹×۷×۴، ۹×۹×۳ و ۱۰×۶×۳ می‌شوند. مسایل بزرگ شامل ۱۱×۱۷×۵، ۱۲×۲۰×۶، ۱۳×۱۹×۷، ۱۴×۱۸×۹، ۱۵×۱۷×۶، ۱۶×۱۶×۷، ۱۷×۱۵×۸، ۱۸×۱۶×۶، ۱۹×۱۲×۱۰ و ۲۰×۱۵×۸ می‌شوند. زمان پردازش هر عملیات برای هر کدام از این مسایل یک عدد تصادفی تولید شده در بازه‌ی [۱,۱۰۰] است، از آنجایی که زمان‌های پردازش اغلب مسایل

[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]

بنچ مارک در این دامنه ایجاد می‌شوند [۲۱]. الگوریتم ژنتیک در زبان برنامه نویسی MATLAB R2016a تحت سیستم عامل مایکروسافت ویندوز ۱۰ و با CPU ۲/۳۰ گیگا هرتز و رم ۸ گیگا بایت کد و اجرا شده است.

۲-۲ تولید موعدهای تحویل کالا

تولید موعدهای تحویل کار i ام (d_i) بدین صورت می‌باشد: یک توالی تصادفی شامل n کار تولید می‌گردد. موعدهای تحویل به طور یکنواخت در بازه $\left[P(1-T-R/2);P(1-T+R/2)\right]$ توزیع می‌شود. T فاکتور دیرکرد و R حدود موعد تحویل نامیده می‌شود. P یک حد پایین برای زمان تکمیل کارهاست [۲۲] که به صورت معادله (۱) برای مساله مورد بررسی در این مقاله، به دست می‌آید:

$$\max\left\{\max_{1\le j\le m}\left\{\sum_{k=1}^{nL}t_{kj}+\min_{k}\sum_{v=1}^{j-1}t_{kv}+\min_{k}\sum_{v=j+1}^{m}t_{kv}\right\},\ \max_{1\le k\le nL}\sum_{j=1}^{m}t_{kj}\right\} \qquad (1)$$

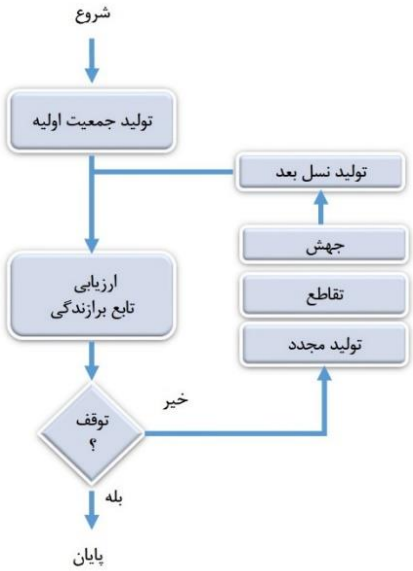
چهار سناریو برای موعدهای تحویل با ترکیبات مختلف برای T={۰/۲ , ۰/۴} و R={۰/۶ , ۱/۲}، وجود دارد. برای هر ترکیب از تعداد کارها (n)، تعداد ماشین‌ها (m) تعداد سطح‌ها (L) و ماتریس زمان‌های پردازش، ما ۴ مثال طبق سناریوهای تعریف شده در بالا ایجاد می‌کنیم. نتایج ارایه شده حاصل میانگین برای این ۴ مثال می‌باشد.

۳ الگوریتم ژنتیک

الگوریتم ژنتیک (GA) یکی از روش‌های جستجوی فراابتکاری در کلاس الگوریتم‌های بهینه‌سازی تصادفی است که اولین بارتوسط هلند [۲۳] معرفی گردید. این الگوریتم برای بهینه‌سازی مسایل پیچیده با فضای جستجوی ناشناخته مناسب است. GA یک فضای مساله را با یک جمعیت از کروموزوم‌ها که هر کدام به شکل یک حل در فضای مساله رمزگشایی می‌شود، جستجو می‌کند و کروموزوم‌ها را برای تحقیق بعدی با توجه به کارایی آنها انتخاب می‌نماید. شکل ۱ چارچوب الگوریتم ژنتیک را نشان می‌دهد.

پارامترها در GA شامل اندازه جمعیت، تعداد نسل‌ها، احتمال تقاطع، احتمال جهش، احتمال تولید مجدد می‌باشد. جمعیت اولیه به طور تصادفی تولید می‌گردد. با تابع برازندگی (ارزیابی)، هر کروموزوم رمزگشایی شده و به آن یک معیار و مقدار برازندگی نسبت داده می‌شود و در روند انتخاب، کروموزوم‌های متناسب‌تر احتمال بیشتری برای انتخاب شدن برای نسل بعدی دارند. تابع برازندگی با $F_i=(f_{\max}-f_i)^{\alpha}$ مشخص شده است، بدین صورت که ابتدا $T_{\max}$ برای همه‌ی کروموزوم‌های جمعیت محاسبه می‌شود و بزرگ‌ترین $T_{\max}$ در میان همه کروموزوم‌ها در جمعیت متداول پیدا شده و به عنوان $f_{\max}$ مشخص می‌گردد. تفاوت بین هر دیرکرد (f_i) و $f_{\max}$ با توان ۱/۰۰۵، مقدار برازندگی آن کروموزوم مشخص است. مقیاس پایین توان (α) توسط گیلیز [۲۴] پیشنهاد داده شد.

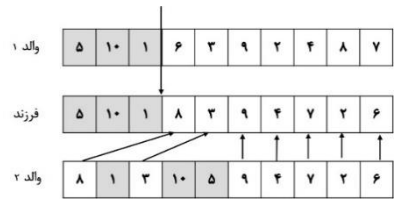
[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]



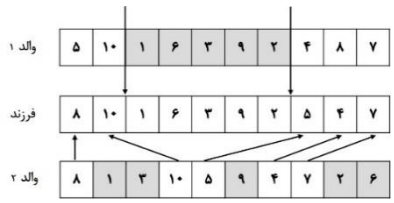
شکل ۱. چارچوب الگوریتم ژنتیک

انتخاب، فرآیندی برای انتخاب فرزند از والدین برای نسل بعدی است. در این تحقیق فرآیند انتخاب والدین از طریق فرآیند انتخاب چرخه رولت به کار گرفته می‌شود.

تقاطع که عملگر اصلی GA می‌باشد، عملیات تولید یک رشته ی جدید (فرزند) از دو رشته والد است. تقاطع تک نقطه ای[1] و دو نقطه‌ای[2] در شکل‌های ۲ و ۳ نشان داده شده است.



شکل ۲. تقاطع تک نقطه‌ای

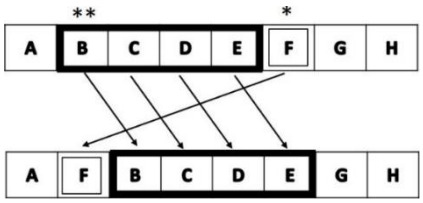


شکل ۳. تقاطع دونقطه‌ای

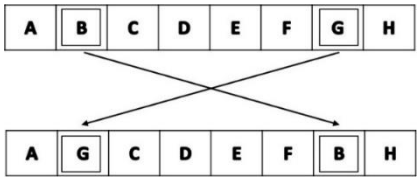
جهش عملگر کاربردی دیگر GA است که برای انتقال از یک حل متداول به حل همسایه‌اش در یک الگوریتم جستجوی محلی به کار گرفته می‌شود. به‌کارگیری این عملگر از بهینگی زودرس و افتادن در بهینه محلی جلوگیری می‌کند. دو عملگر جهش الحاقی[3] و تعویض جفتی (تصادفی)[4] مورد بررسی در این تحقیق، در ادامه شرح داده شده است. جهش الحاقی، الحاق یک کار تصادفی به یک موقعیت تصادفی است که در شکل (۴) نشان داده شده است. در دیگر عملگر جهش، تعویض تصادفی به‌کار گرفته شده است. بدین صورت که کارهای دو موقعیت تصادفی با یکدیگر تعویض می‌گردند، مانند مثال شکل ۵.

[1] One-Point Crossover
[2] Two-Point Crossover
[3] Insertion (shift) Mutation
[4] Swap Mutation

[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]



شکل ٤. جهش الحاقی



شکل ٥. جهش تعویض تصادفی

عملگر تولید مجدد با توجه به احتمال تخصیص‌یافته به آن، شماری از بهترین حل‌ها را به نسل بعدی انتقال می‌دهد. اگر احتمال تقاطع را با P_c و احتمال جهش را با P_m نشان دهیم، احتمال تولید مجدد به صورت $1 - Pc - Pm$ محاسبه می‌گردد.

شرط‌های مورد استفاده‌ی معمول برای توقف الگوریتم ژنتیک عبارتند از تعداد نسل، یک هدف مشخص و یا همگنی جمعیت. در این تحقیق تعداد نسل[1] به عنوان معیار توقف در نظر گرفته شده است.

٤ تعیین پارامترها و عملگرهای الگوریتم ژنتیک

برای اجرای الگوریتم ژنتیک با هدف یافتن بهترین حل‌ها، بهترین پارامترهای ورودی و عملگرها را برای آن جستجو می‌کنیم. بدین منظور روش تاگوچی را به‌کار می‌گیریم. در این بخش، روش تاگوچی برای تنظیم پارامترهای الگوریتم ژنتیک به طور کامل شرح داده شده و نتایج آن برای چهار سناریو حالت‌های مختلف به‌کارگیری عملگرهای الگوریتم ژنتیک (چهار واحد تصمیم‌گیری) آورده شده که در جدول ۱ نشان داده شده است. در ادامه با استفاده از روش تحلیل پوششی داده‌ها واحدهای کارا تعیین و رتبه‌بندی می‌گردند.

جدول ۱. حالت‌های مختلف به‌کارگیری عملگرهای الگوریتم ژنتیک

DMUs	Operators	
DMU۱	One-Point Crossover	Swap Mutation
DMU۲	One-Point Crossover	Shift Mutation
DMU۳	Two-Point Crossover	Swap Mutation
DMU۴	Two-Point Crossover	Shift Mutation

٤-۱ طراحی آزمایشات تاگوچی

معروف‌ترین روش بهینه‌سازی پارامترها روش طراحی آزمایشات و رویه سطح پاسخ می‌باشد. وقتی تعداد پارامترها زیاد باشد آزمایشات بسیاری باید انجام گیرد تا به جواب مطلوب برسیم و به علت نیاز به آزمایشات زیاد، از روش‌هایی مانند تاگوچی یا روش سعی و خطا استفاده می‌شود. روش تاگوچی یک رویکرد سیستماتیک، ساده و در عین حال کارا برای بهینه‌سازی پارامترهای هر الگوریتم، با استفاده از یک مجموعه

[1] Max_gen

[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]

محدودی از آزمایشات، ارایه می‌کند که نتایج حاصل از آزمایشات بر روی تمامی ناحیه آزمایشی که توسط فاکتورها و سطوح تنظیم آنها ساخته می‌شود، معتبر می‌باشند. دو ابزار اصلی روش تاگوچی به‌منظور تعیین و آزمون سیستماتیک سطوح مختلف هر فاکتور، عبارتند از: ۱) طراحی آزمایشات خصوصاً آرایه‌های متعامد (OAs)[1] و ۲) نسبت S/N[2]. ماتریس OA یک ماتریس از اعداد مرتب‌شده در ردیف‌ها و ستون‌ها می‌باشد. هر ردیف، سطح فاکتورها را در هر اجرا و هر ستون یک سطح مشخصی از یک فاکتور را که می‌تواند در هر اجرا تغییر داده شود، نشان می‌دهد. نسبت S/N نیز یک شاخص کیفیت بوده و هدف آزمایشات تاگوچی یافتن بهترین سطح از هر فاکتور می‌باشد، به‌طوری که نسبت S/N مربوط به آن فاکتور در آن سطح بیشینه شود. روش تاگوچی می‌تواند به سه مرحله مختلف تقسیم شود: مرحله طرح‌ریزی، مرحله اجرا و مرحله تجزیه و تحلیل. این مراحل در زیر بخش‌های زیر توضیح داده خواهند شد:

٤-١-١ مرحله طرح‌ریزی

در اولین مرحله، فاکتورهای متنوع و سطوح آنها تعیین می‌شوند. در این تحقیق، این مرحله شامل یافتن پارامترهای الگوریتم پیشنهادی برای حل مساله RPFS می‌باشد. پارامترها و سطوح آنها با توجه به ادبیات موضوع و نتایج تجربی حاصل از تحلیل همگرایی نمودارهای حل در جدول ۲ آورده شده‌اند.

جدول ۲. پارامترهای مؤثر بر الگوریتم ژنتیک پیشنهادی

Factor	Symbol	Level	Type	
Crossover Rate (P_c)	A	۳	A(۱)__۰/۶	
			A(۲)__۰/۷	
			A(۳)__۰/۸	
Mutation Rate (P_m)	B	۳	B(۱)__۰/۰۵	
			B(۲)__۰/۰۷	
			B(۳)__۰/۰۹	
			Small-sized Problems	Large-sized Problems
Population Size (*Pop_size*)	C	۳	C(۱)__۵۰	C(۱)__۱۰
			C(۲)__۱۰۰	C(۲) __۲۰۰
			C(۳)__۲۰۰	C(۳)__۳۰۰
Stopping Condition (*Max_gen*)	D	۳	D(۱)__۱۰	D(۱)__۳۰
			D(۲)__۲۰	D(۲)__۴۰
			D(۳)__۳۰	D(۳)__۵۰

[1] Orthogonal Arrays (OAs)
[2] Signal-to-Noise (S/N)

[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]

اگر از روش سنتی طرح فاکتوریل کامل استفاده کنیم، نیاز به اجرای ۸۱ = ($۳^۴$) آزمایش داشتیم تا بتوان این چهار فاکتور را تنظیم نماییم. با مراجعه به جدول استاندارد آرایه‌های متعامد تاگوچی و یا نرم‌افزار Minitab17، آرایه متعامد ($L_۹(۳^۴)$ به عنوان مناسب‌ترین طرح آرایه متعامد انتخاب می‌شود.آرایه متعامد این طرح در جدول ۳ نشان داده شده است. بنابراین با استفاده از روش تاگوچی، تنها ۹ آزمایش انجام می‌شود که نشان‌دهنده کارایی و به‌صرفه بودن آن می‌باشد.

جدول ۳. ماتریس OA به‌کار رفته در طرح تاگوچی انتخابی

Trial	Control factor level			
	A	B	C	D
۱	A(۱)	B(۱)	C(۱)	D(۱)
۲	A(۱)	B(۲)	C(۲)	D(۲)
۳	A(۱)	B(۳)	C(۳)	D(۳)
۴	A(۲)	B(۱)	C(۲)	D(۳)
۵	A(۲)	B(۲)	C(۳)	D(۱)
۶	A(۲)	B(۳)	C(۱)	D(۲)
۷	A(۳)	B(۱)	C(۳)	D(۲)
۸	A(۳)	B(۲)	C(۱)	D(۳)
۹	A(۳)	B(۳)	C(۲)	D(۱)

٤-۱-۲ مرحله اجرا

در روش طراحی تاگوچی با دو نوع فاکتور مواجه می‌شویم: فاکتورهای قابل کنترل و فاکتورهای غیرقابل کنترل. تمامی فاکتورهای ذکر شده در مرحله قبل، جزء فاکتورهای قابل کنترل می‌باشند، چرا که می‌توانیم سطوح آنها را انتخاب و تنظیم کنیم. برای استفاده از روش تاگوچی باید ابتدا نسبت S/N را برای مساله پیدا کنیم به گونه‌ای که کم‌ترین نسبت بهترین باشد. اما از آنجا که با یک مساله کمینه‌سازی سرو کار داریم، از نسبت S/N در معادله (۲) زیر استفاده می‌کنیم (K تعداد اجرای هر آزمایش و Y پاسخ فرآیند می‌باشد.):

$$S/N_{smaller\ is\ better} = -۱۰\ log\left(\frac{۱}{K}\sum_{i=۱}^{k} Y_i^۲\right) \qquad (۲)$$

در روش تاگوچی، فقط یک مقدار را به عنوان خروجی (Y) در هر بار اجرای الگوریتم در نظر گرفته و مقدار S/N را بر اساس همین مقدار برای هر بار اجرای الگوریتم محاسبه می‌کنیم. مقدار T_{max} به عنوان خروجی در نظر گرفته می‌شود.

[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]

٤-١-٣ مرحله تجزیه و تحلیل

زمانی که کلیه نسبت‌های S/N و میانگین پاسخ‌ها به ازای هر یک از آزمایشات محاسبه شدند، روش تاگوچی از یک رویکرد نموداری برای تجزیه و تحلیل داده‌ها استفاده می‌کند. در این رویکرد، نمودارهای متوسط نسبت‌های S/N و نیز متوسط میانگین پاسخ‌ها برای هر فاکتور و به ازای سطوح مختلف آنها، ترسیم می‌شوند. سپس سطوح بهینه هر فاکتور جایی است که ۱) نمودار S/N بیشینه و ۲) نمودار میانگین پاسخ‌ها کمینه شود (این موضوع وابسته به نوع هدف می‌باشد).

همان‌گونه که اشاره شد، در این تحقیق مسایل با اندازه‌های کوچک و بزرگ برای کارایی الگوریتم ژنتیک به‌کار برده می‌شود. هر کدام از این مسایل ۴ مرتبه توسط الگوریتم پیشنهادی جهت جلوگیری از اثرات جواب‌های تصادفی اجرا می‌شود و میانگین مقدار T_{max} در نظر گرفته می‌شود. در ادامه نتیجه تجزیه و تحلیل به روش تاگوچی به صورت نمودار نسبت‌های S/N و میانگین پاسخ‌های به‌دست آمده در هر اندازه از مسایل مربوطه آورده شده‌اند.

٤-١-٣-١ تجزیه و تحلیل مسایل با اندازه کوچک

در شکل ۶ نمودارهای نسبت S/N و نیز میانگین‌های پاسخ برای مسایل با اندازه کوچک نشان داده شده‌اند. همان‌طور که در این شکل فوق مشاهده می‌شود، نمودارهای نسبت S/N و میانگین‌های پاسخ، در تمام موارد یکدیگر را تایید می‌کنند. سطوح بهینه پارامترها و اپراتورهای الگوریتم در مسایل کوچک در جدول ۴ آورده شده است:

جدول ٤. سطح بهینه فاکتورهای الگوریتم ژنتیک به ازای مسایل کوچک

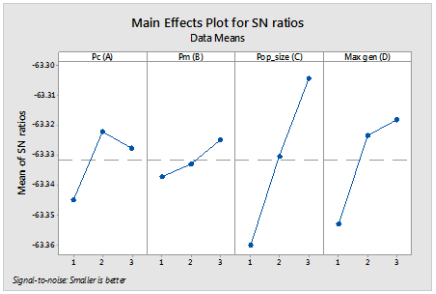
Factor	Symbol	Optimum level			
		DMU۱	DMU۲	DMU۳	DMU۴
Crossover Rate (P_c)	A	A(۲) —۰/۷	A(۲) —۰/۷	A(۲) —۰/۷	A(۲) —۰/۸
Mutation Rate (P_m)	B	B(۳) —۰/۰۹	B(۳) —۰/۰۷	B(۳) —۰/۰۷	B(۳) —۰/۰۷
Population Size (*Pop_size*)	C	C(۳) —۲۰۰	C(۳) —۲۰۰	C(۳) —۲۰۰	C(۳) —۲۰۰
Stopping Condition (*Max_gen*)	D	D(۳) —۳۰	D(۲) —۲۰	D(۲) —۲۰	D(۳) —۳۰

٤-١-٣-٢ تجزیه و تحلیل مسایل با اندازه بزرگ

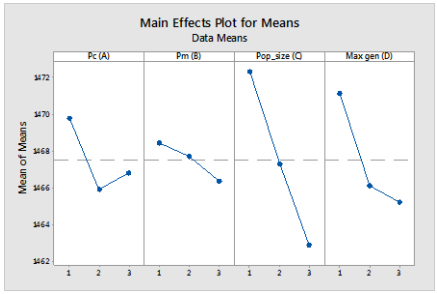
در شکل ۷ نمودارهای نسبت S/N و نیز میانگین‌های پاسخ برای مسایل با اندازه بزرگ نشان داده شده‌اند. با توجه این شکل، نمودارهای نسبت S/N و میانگین‌های پاسخ، در تمام موارد یکدیگر را تایید می‌کنند. سطوح بهینه پارامترها و اپراتورهای الگوریتم در مسایل بزرگ در جدول ۵ نشان داده شده است.

[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]

DMU۱ - Small-sized Problems

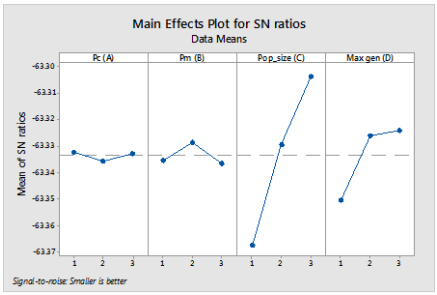


S/N

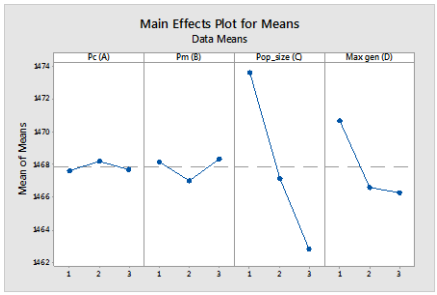


Mean

DMU۲ - Small-sized Problems

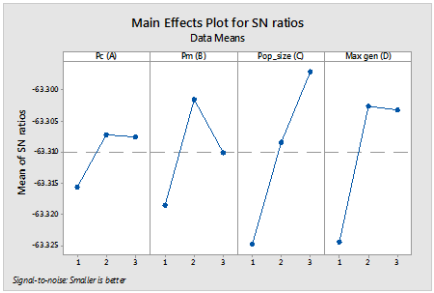


S/N

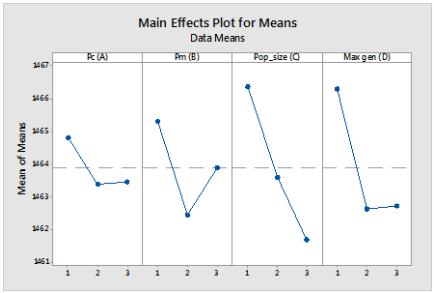


Mean

DMU۳ – Small-sized Problems

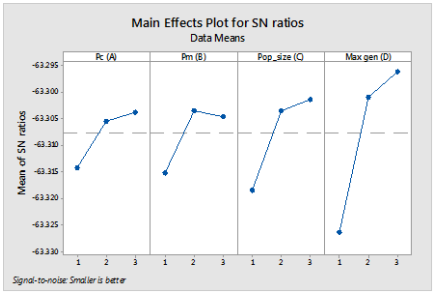


S/N

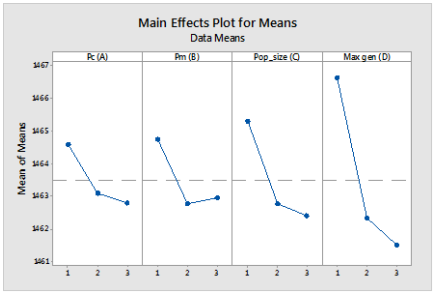


Mean

DMU۴ - Small-sized Problems



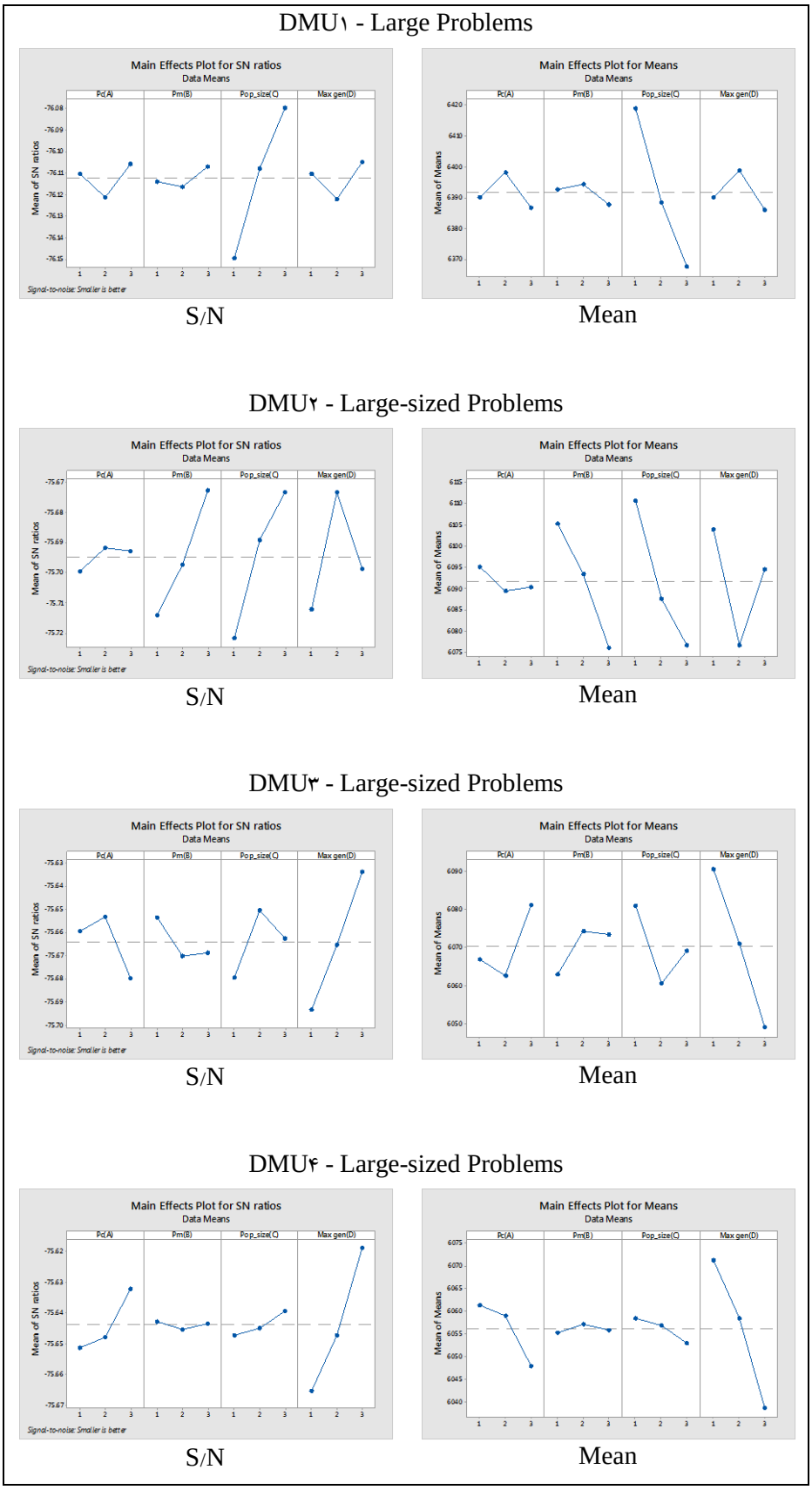
S/N



Mean

شکل ٦. نمودارهای نسبت S/N و میانگین‌های پاسخ در مسایل کوچک

[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]



شکل ۷. نمودارهای نسبت S/N و میانگین‌های پاسخ در مسایل بزرگ

[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]

جدول ۵. سطح بهینه فاکتورهای الگوریتم ژنتیک به ازای مسایل بزرگ

Factor	Symbol	Optimum level			
		DMU۱	DMU۲	DMU۳	DMU۴
Crossover Rate (P_c)	A	A(۳) —۰/۸	A(۲) —۰/۷	A(۲) —۰/۷	A(۳) —۰/۸
Mutation Rate (P_m)	B	B(۳) —۰/۰۹	B(۳) —۰/۰۹	B(۱) —۰/۰۵	B(۱) —۰/۰۵
Population Size (*Pop_size*)	C	C(۳) —۳۰۰	C(۳) —۳۰۰	C(۲) —۲۰۰	C(۳) —۳۰۰
Stopping Condition (*Max_gen*)	D	D(۳) —۵۰	D(۲) —۴۰	D(۳) —۵۰	D(۳) —۵۰

٤-۲ تحلیل پوششی داده‌ها

یکی از رایج‌ترین مدل‌ها برای ارزیابی عملکرد، تحلیل پوششی داده‌ها است که یک روش برنامه‌ریزی ریاضی برای ارزیابی واحدهای تصمیم‌گیری (DMUs) است [۲۵]. این روش بر اساس مجموعه‌ای از مشاهدات تجربی، مرز کارایی را از طریق مقایسه عملکرد نسبی هر یک از واحدها برآورد می‌کند. بنابراین، ورودی و خروجی واحدهای تصمیم‌گیری به گونه‌ای تنظیم می‌گردد که بتواند در حل این مدل برنامه‌ریزی ریاضی استفاده شود. در اینجا هدف پیدا کردن موثرترین ترکیب عملگرهای الگوریتم ژنتیک برای حل مساله است. مقادیر ورودی و خروجی مدل DEA برای مسایل کوچک و بزرگ به ترتیب در جدول‌های ۶ و ۷ آورده شده است.

توجه به نکات کلیدی در مورد ورودی‌ها و خروجی‌های جدول اهمیت دارد. واضح است که DEA برای افزایش کارایی از طریق استفاده از منابع کم (ورودی) و به دست آوردن حداکثر محصولات (خروجی) تلاش می‌کند. بنابراین، اگر بخواهیم از تابع هدف مساله به عنوان خروجی مدل استفاده کنیم، باید برخی تغییرات را اعمال کرد، زیرا از نوع هزینه بوده و به حداقل رساندن آن مطلوب است. همان مساله نیز در مورد زمان اجرا الگوریتم صدق می‌کند. به همین دلیل، مقادیر ذکر شده به صورت معکوس در نظر گرفته می‌شود.

جدول ٦. مقادیر ورودی و خروجی مدل DEA مسایل با اندازه کوچک

DMUs	P_c I۱	P_m I۲	*Pop_size* I۳	*Max_gen* I۴	Avg. T_{max} O۱	Avg. Run Time O۲
DMU۱	۰/۷	۰/۰۹	۲۰۰	۳۰	$\frac{۱}{۱۴۵۱}$	$\frac{۱}{۱۴}$
DMU۲	۰/۶	۰/۰۷	۲۰۰	۳۰	$\frac{۱}{۱۴۵۹}$	$\frac{۱}{۱۳/۵}$
DMU۳	۰/۷	۰/۰۷	۲۰۰	۲۰	$\frac{۱}{۱۴۵۶}$	$\frac{۱}{۹/۸}$
DMU۴	۰/۸	۰/۰۷	۲۰۰	۳۰	$\frac{۱}{۱۴۴۹}$	$\frac{۱}{۱۵}$

[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]

جدول ۷. مقادیر ورودی و خروجی مدل DEA مسایل با اندازه بزرگ

DMUs	P_c I۱	P_m I۲	*Pop_size* I۳	*Max_gen* I۴	Avg. T_{max} O۱	Avg. Run Time O۲
DMU۱	۰/۸	۰/۰۹	۳۰۰	۵۰	۱/۶۰۸۸	۱/۹۱۵
DMU۲	۰/۷	۰/۰۹	۳۰۰	۴۰	۱/۶۰۳۵	۱/۷۳۲
DMU۳	۰/۷	۰/۰۵	۲۰۰	۵۰	۱/۶۰۲۶	۱/۶۲۴
DMU۴	۰/۸	۰/۰۵	۳۰۰	۵۰	۱/۶۰۲۴	۱/۹۰۸

با توجه به توضیحات بالا، مدلی با چهار واحد تصمیم‌گیری، چهار ورودی و دو خروجی داریم. در اینجا به دنبال کاراترین واحد تصمیم‌گیری (DMU) بدون تغییر در مقادیر چهار دسته ورودی و با افزایش مقدارهای خروجی هستیم. با توجه به اطلاعات و مفروضات فوق، مدل (۳) را برای بازده به مقیاس ثابت DEA برای همه DMUها ایجاد کرده و سپس با Lingo حل می‌کنیم.

$$\begin{aligned} &\max \varphi \\ &s.t. \ \sum_{j=1}^{4} \lambda_i x_j \le x_p \ , \\ &\sum_{r=1}^{4} \lambda_j y_j \ge \varphi y_p \ , \\ &\lambda_j \ge 0 \ , \ j = 1,2,\ldots,n. \end{aligned} \qquad (۳)$$

پس از اجرای مدل برای هر یک از چهار واحد تصمیم‌گیری، نتایج ثبت‌شده در جدول‌های ۸ و ۹ به ترتیب برای مسایل با اندازه‌های کوچک و بزرگ به‌دست آمده است.

جدول ۸. مقادیر کارایی و وزن‌ها DMUهای مدل DEA برای مسایل با اندازه کوچک

DMUs	φ	λ_1	λ_2	λ_3	λ_4
DMU۱	۱	۱	۰	۰	۰
DMU۲	۱	۰	۱	۰	۰
DMU۳	۱	۰	۰	۱	۰
DMU۴	۱	۰	۰	۰	۱

جدول ۹. مقادیر کارایی و وزن‌ها DMUهای مدل DEA برای مسایل با اندازه بزرگ

DMUs	φ	λ_1	λ_2	λ_3	λ_4
DMU۱	۱/۱۵	۰	۰/۷۱	۰/۴۳	۰
DMU۲	۱	۰	۱	۰	۰
DMU۳	۱	۰	۰	۱	۰
DMU۴	۱	۰	۰	۰	۱

[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]

اگر φ حاصل از مدل بالا برای هر DMU برابر با ۱ باشد، آن واحد کارآمد است و راه حل بهینه است، اما اگر φ یک عدد بزرگ‌تر از ۱ باشد، می‌توان گفت که حل دیگری در مجموعه امکان تولید وجود دارد که بهتر از حل به‌دست آمده بوده و بنابراین این حل مطلوب نیست. همان‌طور که در جدول ۸ آمده است، همه‌ی واحدهای تصمیم برای مسایل کوچک کارا هستند. اما برای مسایل بزرگ واحد۱، با توجه به مقدار φ به‌دست آمده برای آن واحد، ناکارآمد است (جدول ۹). به عبارت دیگر، عملگرهای تقاطع تک نقطه‌ای و جهش تعویض تصادفی برای مسایل بزرگ منجر به حل ناکارامد در مقایسه با دیگر ترکیب عملگرها می‌گردد.

با این حال، ما فقط به دنبال یک مدل منحصر به فرد هستیم که توسط آن الگوریتم برای مساله مورد نظرمان اجرا شود. بنابراین، باید بین واحدهای کارا انتخاب کنیم. برای این منظور از روش‌های رتبه‌بندی DMUهای موثر استفاده شده است که در ادامه توضیح داده شده است.

اندرسن و پترسن [۲۶] یک روش برای ارزیابی واحدهای کارآمد پیشنهاد دادند که تعیین یک واحد کارآمد را ممکن ساخت. با استفاده از این روش، واحدهای تصمیم‌گیری ناکارا از مجموعه امکان تولید حذف شده و پس از آن DEA درDMUهای باقی مانده اعمال می‌شود. مدل (۴) اندرسن پترسن (AP) برای واحدهای کارآمد به شرح زیر است:

$$
\begin{aligned}
&\max \ \sum_{r=۱}^{۲} u_r y_{rp} \\
&s.t. \\
&\sum_{i=۱}^{۴} v_i x_{ip} = ۱ , \\
&\sum_{r=۱}^{۲} u_r y_{rj} - \sum_{i=۱}^{۴} v_i x_{ij} \le ۰ , \quad j = ۱,۲,۳,۴, \quad j \ne p , \\
&u_r \ge ۰ , \ r = ۱,۲ , \\
&v_i \ge ۰ , \ i = ۱,۲,۳,۴.
\end{aligned}
\qquad (۴)
$$

جدول ۱۰. مقادیر کارایی و وزن‌های DMUهای کارا در مدل اندرسون-پترسن برای مسایل با اندازه کوچک

DMUs	$\theta_{A.P}$	$u_۱$	$u_۲$	$v_۱$	$v_۲$	$v_۳$	$v_۴$
۱	۱/۰۰۳۸	۰/۱۰۲	۱۴۲۲/۹۹۸	۰/۱۰۱	۰	۰	۰/۰۰۵
۲	۱/۱۵۲۳	۰	۱۶۷۰/۸۴۱	۰/۴۰۷	۰	۰	۰
۳	۲/۰۶۶۳	۲۰/۲۵۰	۰	۰	۰	۰	۰/۰۵۰
۴	۱/۰۱۴۸	۰	۱۴۳۸/۰۰	۰	۰/۳۵۰	۰/۰۰۵	۰

[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]

جدول ۱۱. مقادیر کارایی و وزن‌های DMUهای کارا در مدل اندرسون-پترسن برای مسایل بزرگ

DMUs	$\theta_{A.P}$	u_1	u_2	v_1	v_2	v_3	v_4
۲	۰/۲۴۷۹	۷۵۲۴/۵۱۳	۰/۸۲۷	۰	۰	۰	۰/۰۲۵
۳	۱/۸۹۷۵	۰	۱۱۸۴/۰۶۷	۰	۷/۸۳۸	۰/۰۰۳	۰
۴	۰/۰۰۰۳	۶۰۲۶/۰۰۰	۰	۰	۲۰	۰	۰

با توجه به مقادیر $\theta_{A.P}$ در نتایج جدول‌های ۱۰ و ۱۱، برای مسایل با اندازه کوچک به ترتیب DMUهای ۳، ۲، ۴ و ۱ کارا بوده و برای مسایل با اندازه بزرگ به ترتیب DMUهای ۳، ۲و۴ کارا می‌باشند. درنتیجه می‌توان بیان نمود که برای حل مسایل RPFS بهترین ترکیب پارامترها و عملگرا از طریق DMU۳ (عملگرهای تقاطع دو نقطه ای و جهش تعویض تصادفی) منجر به نتیجه بهترین تابع هدف با توجه به زمان محاسباتی می‌گردد. قابل به ذکر است DMU۱ که برای مسایل کوچک دارای رتبه‌ی آخر شد، برای مسایل با اندازه بزرگ ناکارا گردید. شایان ذکر است که موراتا [۲۷] نشان داده بود که تقاطع دونقطه‌ای برای مسایل جریان کارگاهی موثر است. همان‌طور که از نتایج به‌دست آمده برمی‌آید، با بالارفتن اندازه مساله، تعیین پارامترها و عملگرهای الگوریتم حل اهمیت بیشتری پیدا می‌کند. روش پیشنهادی می‌تواند به عنوان روشی برای تنظیم پارامترهای و تعیین عملگرهای الگوریتم‌های فراابتکاری به جای استفاده از روش آزمون و خطا، زمانی که مقادیر بهینه پارامترها مشخص نیست، استفاده شود.

۵ نتیجه‌گیری و پیشنهادها

در این مقاله، به تعیین مقادیر بهینه پارامترها و عملگرهای الگوریتم ژنتیک برای حل مساله جریان کارگاهی جایگشتی دوباره واردشونده پرداخته شد. جهت نیل به این هدف، ابتدا روش تاگوچی برای چهار ترکیب مختلف از عملگرهای الگوریتم ژنتیک با توجه به کمینه کردن تابع هدف (T_{max}) به‌کارگرفته شد و سپس روش تحلیل پوششی داده‌ها، با در نظر گرفتن تابع هدف و زمان حل به عنوان خروجی، DMUهای کارا را مشخص نمود. در روش پیشنهادی هم‌زمان از خاصیت کم کردن تعداد آزمایشات در روش تاگوچی و همچنین رتبه‌بندی سناریوهای مختلف تنظیم الگوریتم حل در تحلیل پوششی داده‌ها با توجه به تابع هدف بهینه در کمینه زمان حل بهره برده شده است.

به طور کلی روش پیشنهادی برای دیگر الگوریتم‌های فراابتکاری، جهت تعیین پارامترها و عملگرها با هدف دستیابی به حل مناسب و قابل قبول در کم‌ترین زمان و با حداقل خطای احتمالی می‌تواند به‌کار گرفته شود. در نهایت می‌توان اطمینان داشت که از خطاهای مربوط به روش آزمون و خطا در تنظیم پارامترهای الگوریتم‌های فراابتکاری جلوگیری می‌نماییم.

[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]

منابع

[1] Montgomery, D. C. (2005). Design and Analysis of Experiments. John Wiley & Sons, 6th Ed, New york.

[2] Ruiz, R., Maroto, C., & Alcaraz, J. (2006). Two new robust genetic algorithms for the flowshop scheduling problem. Omega, 34(5), 461-476.

[3] Ruiz, R., & Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. European Journal of Operational Research, 177(3), 2033-2049.

[4] Cochran WG, Cox GM (1992) Experimental designs, 2nd edn.Wiley, New York.

[5] Ross RJ (1989) Taguchi techniques for quality engineering. McGraw-Hill, New York.

[6] Cheng, B. W., & Chang, C. L. (2007). A study on flowshop scheduling problem combining Taguchi experimental design and genetic algorithm. Expert Systems with Applications, 32(2), 415-421.

[7] Gholami, M., Zandieh, M., & Alem-Tabriz, A. (2009). Scheduling hybrid flow shop with sequence-dependent setup times and machines with random breakdowns" The International Journal of Advanced Manufacturing Technology, 42(1), 189-201.

[8] Naderi, B., Zandieh, M., & Roshanaei, V. (2009). Scheduling hybrid flowshops with sequence dependent setup times to minimize makespan and maximum tardiness. The International Journal of Advanced Manufacturing Technology, 41(11), 1186-1198.

[9] Zandieh, M., & Roumani, M. (2017). A biogeography-based optimization algorithm for order acceptance and scheduling. Journal of Industrial and Production Engineering, 34(4), 312-321.

[10] Azadeh, A., Elahi, S., Farahani, M. H., & Nasirian, B. (2017). A genetic algorithm Taguchi based approach to inventory routing problem of a single perishable product with transshipment. Computers & Industrial Engineering, 104, 124-133.

[11] Hardas, C. S., Doolen, T. L., & Jensen, D. H. (2008). Development of a genetic algorithm for component placement sequence optimization in printed circuit board assembly. Computers & industrial engineering, 55(1), 165-182.

[12] Wang, T. Y., & Wu, K. B. (1999). A parameter set design procedure for the simulated annealing algorithm under the computational time constraint. Computers & operations research, 26(7), 665-678.

[13] Siaby, H., Rodtami Malkhalifeh, M., Hosseinzadeh Lotfi, F., & Behzadi, M.H. (2017). Implementation of indirect impact factor related to inputs and outputs in the performance evaluation of decision making units in DEA, Journal of Operational Research and Its Applications, 13(3), 55-67.

[14] Aghayi, N., Hosseinzadeh Lotfi, F., Gholami, K., & Ghelej Beigi, Z. (2018). Ranking and sensitivity analysis for ranks of DMUs based on the ideal hyperplan, Journal of Operational Research and Its Applications, 15(2), 125-133.

[15] Fasihi, M., Jolai, F., & Tavakkoli-Moghaddam, R. (2011). Minimizing the maximum tardiness on an m-machine reentrant permutation flowshop problem. International Journal of Industrial Engineering and Production Management, 21(4), 179-190.

[16] Wang, M. Y., Sethi, S. P., & van de Velde, S. L. (1997). Minimizing makespan in a class of reentrant shops. Operations Research, 45(5), 702-712.

[17] Pinedo, M. (2002). Scheduling: Theory, Algorithms, and Systems**.** Prentice-Hall, New Jersey.

[18] Uzsoy, R., Lee, C. Y., & Martin-Vega, L. A. (1992)."A review of production planning and scheduling models in the semiconductor industry part I: system characteristics, performance evaluation and production planning. IIE transactions, 24(4), 47-60.

[19] Graves, S. C., Meal, H. C., Stefek, D., & Zeghmi, A. H. (1983). Scheduling of re-entrant flow shops. Journal of Operations Management, 3(4), 197-207

[20] Koulamas, C. (1994). The total tardiness problem: review and extensions. Operations research, 42(6), 1025-1041.

[21] Beasley, J. E. (1990). OR-Library: distributing test problems by electronic mail. Journal of the operational research society, 1069-1072.

[22] Choi, S. W., & Kim, Y. D. (2008). Minimizing makespan on an m-machine re-entrant flowshop." Computers & Operations Research, 35(5), 1684-1696.

[Downloaded from jorlu.ir on 2026-11-28]

[DOR: 20.1001.1.22517286.2021.18.2.2.7]

[DOI: 10.52547/jamlu.18.2.107]

[23] Holland, J. H. (1975). Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. Ann Arbor, MI: University of Michigan Press.
[24] Gillies, A. M. (1985). Machine learning procedures for generating image domain feature detectors. Ann Arbor, MI: University of Michigan Press.
[25] Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. European journal of operational research, 2(6), 429-444.
[26] Andersen, P., & Petersen, N. C. (1993). A procedure for ranking efficient units in data envelopment analysis. Management science, 39(10), 1261-1264.
[27] Murata, T., Ishibuchi, H., & Tanaka, H. (1996). Genetic algorithms for flows shop scheduling problems. Computers & Industrial Engineering, 30(4), 1061– 1071.

[Downloaded from jorlu.ir on 2026-11-28]
[DOR: 20.1001.1.22517286.2021.18.2.2.7]
[DOI: 10.52547/jamlu.18.2.107]