مجله تحقیق در عملیات در کاربردهای آن، سال دوازدهم، شماره دوم (پیاپی ٤٥)، تابستان ٩٤، ص ص ١٢٤-١٠٩
شاپا ٧٢٨٦-٢٢٥١

الگوریتم فراابتکاری برای اندازه‌گیری کارایی مسایل ابعاد بزرگ در تحلیل پوششی داده‌ها

علی محمودی راد [*1]، مسعود صانعی [2]، صابر ملاعلیزاده زواردهی [3]

۱- دانشجوی دکترای، دانشگاه آزاد اسلامی واحد تهران مرکزی، گروه ریاضی، تهران، ایران

۲- دانشیار، دانشگاه آزاد اسلامی واحد تهران مرکزی، گروه ریاضی، تهران، ایران

۳- استادیار، دانشگاه آزاد اسلامی واحد مسجد سلیمان، گروه مهندسی صنایع، مسجد سلیمان، ایران

رسید مقاله: ۸ بهمن ۱۳۹۳

پذیرش مقاله: ۵ تیر ۱۳۹۴

چکیده

تحلیل پوششی داده‌ها، یکی از موفق‌ترین روش‌ها در ارزیابی اندازه‌گیری کارایی نسبی واحدهای تصمیم‌گیرنده است که از تکنیک برنامه‌ریزی خطی استفاده می‌کند. در مدل‌های تحلیل پوششی داده‌ها، برای اندازه‌گیری کارایی نسبی واحدهای تصمیم‌گیرنده، در حالی که واحدهای تصمیم‌گیرنده و ورودی و خروجی بسیار زیاد هستند حتی با استفاده از کامپیوترهای با سرعت بالا، به محاسبات و زمان پردازش بسیار زیادی نیاز است. در این مقاله، برای رفع این مشکل، الگوریتم فراابتکاری برای ارزیابی کارایی نسبی در ابعاد بسیار بزرگ را پیشنهاد می‌کنیم. چون در ارزیابی مسایل با ابعاد بسیار بزرگ با استفاده از الگوریتم فراابتکاری به زمان پردازش و حافظه کم‌تری نیاز است، لذا ابزار بسیار مناسبی برای ارزیابی کارایی واحدهای تصمیم‌گیرنده می‌باشد. همچنین از آنجایی که عملگرها نقش بسیار مهمی در همگرایی و کیفیت حل الگوریتم‌ها دارند، کلیه عملگرها و پارامترها به منظور بهبود عملکردشان، با روش طراحی آزمایش‌های تاگوچی تنظیم می‌شوند.

کلمات کلیدی: تحلیل پوششی داده‌ها، کارایی، الگوریتم فراابتکاری، مسایل ابعاد بزرگ.

۱ مقدمه

تحلیل پوششی داده‌ها (DEA)، که اولین بار توسط چارنز ، کوپر و رودز [۱] تحت عنوان مدل CCR معرفی شد، تکنیکی غیر پارامتری برای محاسبه کارایی نسبی مجموعه ای از واحدهای تصمیم‌گیرنده (DMUs) با ورودی و خروجی مشترک است که با استفاده از برنامه‌ریزی ریاضی انجام می‌گیرد. عبارت نسبی به این دلیل است که کارایی حاصل نتیجه مقایسه واحدها با یکدیگر است. در سال‌های اخیر در اغلب کشورهای جهان برای ارزیابی عملکرد نهادها و دیگر فعالیت‌های رایج در زمینه‌های مختلف، کاربردهای متفاوتی از DEA دیده شده است.

* عهده‌دار مکاتبات

آدرس الکترونیکی: alimahmoodirad@yahoo.com

علت مقبولیت گسترده‌تر این روش نسبت به سایر روش‌ها، امکان بررسی روابط پیچیده و اغلب نامعلوم بین چندین ورودی و چندین خروجی است که در این فعالیت ها وجود دارد.

با توسعه‌ی روش تحلیل پوششی داده‌ها، این روش در ارزیابی کارایی سازمان‌ها و صنایع گوناگون از جمله بانک‌ها، نیروگاه‌ها، دانشگاه‌ها و... استفاده می‌شود و به عنوان یک ابزار مدیریتی قوی در اندازه‌گیری کارایی، سودمندی قابل توجهی به‌دست آورده و به‌طور گسترده از آن استفاده شده است[۲-۶]. مدل‌های تحلیل پوششی داده‌ها، علاوه بر تعیین میزان کارایی نسبی، نقاط ضعف واحدهای تحت ارزیابی را در متغیرهای مختلف تعیین کرده، و با معرفی الگوهای کارا برای واحدهای ناکارا، به بهبود بهره‌وری و افزایش کارایی سازمان‌ها کمک می‌کنند. الگوهای کارا واحدهایی هستند که نسبت به یک واحد ناکارا، با بهره‌گیری از میزان ورودی مشابه، می‌توانند خروجی بیشتری را تولید کنند یا با استفاده از میزان ورودی کم‌تر، حداقل به همان اندازه خروجی داشته باشند.

در سال‌های اخیر به دلیل رشد و توسعه تکنولوژی و همچنین ارتباط بین علوم مختلف، روش‌های الهام گرفته از طبیعت، برای حل مسایل بهینه‌سازی توسعه یافته‌اند که تحت عنوان روش‌های فراابتکاری در علوم استفاده می‌شوند. رویکردهای فراابتکاری، امروزه کاربرد بسیاری در شاخه‌های مختلف علم بهینه‌سازی پیدا کرده‌اند. مبنای این رویکردها عمدتاً بر اساس نظم یا قواعد موجود در ارگانیسم‌های طبیعی یا برگرفته از دیگر شاخه‌های علوم است. رویکردهای فوق بر خلاف روش‌های دقیق بهینه‌سازی، بدنبال نقاط تا حد ممکن نزدیک به بهینه سراسری می‌باشند به‌طوری‌که نظر تصمیم‌گیرنده را تا سطح قابل قبولی برآورده سازد. به عبارت دیگر، روش‌های فراابتکاری روش‌هایی هستند که جواب‌های نزدیک به بهینه را با یک هزینه محاسباتی قابل قبول جستجو می‌کنند ولی تضمینی برای رسیدن به جواب بهینه نمی‌دهند. به روش‌های فراابتکاری اصطلاحاً روش‌های غیر دقیق نیز گفته می‌شود چرا که مکانیزم‌های تصادفی در ایجاد ساختار آن‌ها نقش مهمی را ایفا می‌کنند.

اخیراً برخی ارگان‌های بزرگ برای ارزیابی DMUs با داده‌های بسیار بزرگ، از مدل‌های DEA استفاده کرده‌اند [۴و۷]. در کاربردهایی از این نوع، به دلیل ابعاد بزرگ مساله مورد بررسی و تعداد زیاد DMU به محاسبات خیلی زیاد نیاز است، به طوری‌که سریع‌ترین کامپیوتر، برای رسیدن به جواب (نمره‌کارایی) به زمان خیلی زیاد نیاز دارند تا مدل‌های ایجاد شده برای هر DMU را حل کنند [۷]. لذا روشی که بتواند در زمان معقولی کارایی هر واحد را در اختیار تصمیم‌گیرنده قرار دهد، می‌تواند بسیار مفید و ارزنده باشد.

در خصوص به کارگیری و ترکیب روش تحلیل پوششی داده‌ها با روش‌های هوش مصنوعی، تلاش‌های انجام شده بسیار محدود و اندک است که نشان از سرنخ‌های جدید در گسترش این موضوع می‌دهد. در ادامه به برخی از کارهای انجام گرفته در این زمینه اشاره شده است.

امروزنژاد و شل [۷] به منظور ارزیابی کارایی نسبی DMUs با مجموعه داده‌های با مقیاس بزرگ از شبکه عصبی استفاده کردند. آن‌ها برای اندازه‌گیری کارایی، پنج دسته داده‌های تصادفی بزرگ را انتخاب نمودند و نتایج به‌دست آمده از شبکه عصبی را با مدل‌های DEA معمولی مقایسه کردند. تحلیل آن‌ها نشان داد که خطای

تخمین برای مجموعه داده‌های بسیار بزرگ کاهش می‌یابد. همچنین از شبکه عصبی برای تعیین نمرات کارایی نسبی DMUs در ابعاد بزرگ استفاده شده است [٨ -١١].

لکزایی و همکاران [١٢] برای تخمین کارایی DMUs در مجموعه داده‌های بسیار بزرگ، الگوریتم الکترومغناطیس را پیشنهاد کردند. با توجه به نقش پارامترها و عملگرها در همگرایی و کیفیت الگوریتم، روش طراحی آزمایشات تاگوچی را بکار گرفتند. رحیمیان و همکاران [١٣] الگوریتم تفاضل تکاملی (DE) را برای ارزیابی کارایی DMUs با داده‌های ابعاد بزرگ، توسعه دادند و به منظور بهبود اجرای الگوریتم از روش طراحی آزمایشات تاگوچی استفاده کردند.

یودهایاکومار و همکاران [١٤] مساله DEA را در یک بانک با ورودی‌ها و خروجی‌های تصادفی در نظر گرفتند. در الگوریتم ژنتیک (GA) ارایه شده تنها از یک عملگر تقاطع تک نقطه‌ای و یک عملگر جهش آشفته برای حل مدل استفاده شده است. آن‌ها از تابع هدف تصادفی و محدودیت‌های شانسی استفاده کردند و شدنی بودن و محدودیت‌های شانسی با تکنیک شبیه سازی مورد بررسی قرار گرفت.

آزاده و همکاران [١٥] یک الگوریتم ترکیبی GA-DEA برای ارزیابی ورودی‌های بحرانی از دو دیدگاه کارایی و هزینه در یک مرکز انتقال برق ارایه نمودند. آن‌ها از یک معیار خاص و مدل‌های ابرکارایی تخصیص هزینه‌ها، برای تحلیل حساسیت و تعیین ورودی‌های بحرانی بر اساس کارایی و هزینه استفاده نمودند.

طلوع و همکاران [١١] به منظور ارزیابی کارایی و بهره‌وری از مجموعه داده‌های بسیار بزرگ با مقادیر منفی، یک الگوریتم ترکیبی با استفاده از شبکه عصبی، پیشنهاد کردند.

در این مقاله، به منظور به‌دست آوردن کارایی با مجموعه داده‌های بسیار بزرگ، الگوریتم ژنتیک و شبیه سازی تبرید (SA) پیشنهاد می‌گردد. بدلیل عدم نیاز به حل مساله برنامه‌ریزی خطی برای هر DMU و زمان پردازش بسیار کمتر نسبت به مدل های کلاسیک DEA، الگوریتم ژنتیک ابزار بسیار مناسبی برای ارزیابی کارایی هر واحد تصمیم گیرنده با مسایل با ابعاد بسیار بزرگ می‌باشد.

ساختار مقاله به این شرح است که در ادامه بیان می‌گردد. در بخش دوم مدل DEA و مفاهیم مورد استفاده در این مقاله بیان می‌شود. الگوریتم ژنتیک و شبیه سازی تبرید همراه با جزییات آن‌ها به ترتیب در بخش سوم و چهارم مورد بررسی قرار می‌گیرد. در بخش پنجم روش طراحی آزمایشات تاگوچی مورد استفاده را تشریح نموده و در بخش ششم نتیجه‌گیری و پیشنهادات ارایه می‌گردد.

٢ تحلیل پوششی داده‌ها

کارایی یک واحد عبارت از مقایسه‌ی ورودی‌ها و خروجی‌های آن با یکدیگر است. در ساده‌ترین حالت که واحدی یک ورودی را مصرف کرده و یک خروجی می‌دهد، کارایی به صورت خارج قسمت خروجی بر ورودی تعریف می‌شود. ولی اغلب، به خاطر پیچیدگی واحدهای تصمیم‌گیری و این‌که واحدهای یک سازمان اهداف متعددی را دنبال می‌کنند، در نظر گرفتن چندین ورودی و چندین خروجی اجتناب‌ناپذیر است. در چنین وضعیتی، کارایی را می‌توان به صورت حاصل تقسیم ترکیبی وزنی از خروجی‌ها بر ترکیبی وزنی از ورودی‌ها

تعریف کرد. به عبارت دیگر، برای هر کدام از خروجی‌ها و ورودی‌ها، وزنی به عنوان ارزش و قیمت آن‌ها در نظر گرفته می‌شود و ارزش کل خروجی‌ها بر قیمت کل ورودی‌ها تقسیم می‌شود. مشکل اساسی، تعیین وزن‌ها (ارزش خروجی‌ها یا قیمت ورودی‌ها) به منظور ترکیب آن‌ها می‌باشد. تحلیل پوششی داده‌ها روشی است که برای یافتن این وزن‌ها طراحی شده است. مدل‌های مبتنی بر تحلیل پوششی داده‌ها مدل‌های برنامه‌ریزی ریاضی هستند که برای اندازه‌گیری کارایی تکنیکی واحدهای تصمیم‌گیری طراحی شده‌اند [۱].

فرض کنید $\{DMU_j \quad j=1,2,...,n\}$، n واحد تصمیم‌گیرنده متجانس باشند که با به‌کار بردن بردار ورودی $x_j=(x_{1j},...,x_{mj})^T>0$ $(j=1,2,...,n)$، بردار خروجی y_j $(j=1,2,...,n)$ را تولید می‌کند که $y_j=(y_{1j},...,y_{sj})^T>0$ و $j=1,2,...,n$. مدل تامسون و همکاران [۱۶] تحت عنوان مدل TDT برای ارزیابی نمره‌ی کارایی نسبی DMU_o به صورت زیر است:

$$\max_{(u,v)} \frac{\frac{u^T y_o}{v^T x_o}}{\max_{1\le j\le n}\{\frac{u^T y_j}{v^T x_j}\}} \qquad (1)$$

$$s.t.$$
$$u\ge 0,\ v\ge 0.$$

که در آن $v\in\mathbb{R}^{m\times 1}$ و $s\in\mathbb{R}^{s\times 1}$ به‌ترتیب بردارهای وزن ورودی و خروجی هستند.

اگر قرار دهیم $\max_{1\le j\le n}\left\{\frac{u^T y_j}{v^T x_j}\right\}=\frac{1}{t}$، آن گاه مدل (۱) به مدل برنامه‌ریزی کسری زیر تبدیل می‌شود:

$$\max \quad \frac{u^T y_o}{v^T x_o}$$
$$s.t. \qquad j=1,2,...,n \qquad (2)$$
$$u^T y_j - v^T x_j \le 1$$
$$u\ge 0,\quad v\ge 0.$$

مدل (۲) به مدل کسری CCR [۱] معروف است. برای تبدیل مدل (۲) به یک مدل برنامه‌ریزی خطی، با استفاده از تبدیلات چارنز و کوپر [۱۷] به مدل زیر خواهیم رسید.

$$\max \quad \theta=u^T y_o$$
$$s.t.$$
$$v^T x_o=1, \qquad (3)$$
$$u^T y_j - v^T x_j\le 0, \qquad j=1,2,...,n,$$
$$u\ge 0,\quad v\ge 0.$$

مدل (۳) همان مدل CCR برای به‌دست آوردن نمره کارایی DMU_o است که آن را فرم پوششی مدل CCR نیز می نامند [۱].

تعریف ۱ [**۱۸**]. اگر مقدار بهینه مدل (۳)، یعنی θ^*، برابر یک باشد، و حداقل یک جواب بهینه (u^*,v^*) با $u^*>0$ و $v^*>0$ وجود داشته باشد، گوییم DMU_o کارای CCR (کارای قوی) است در غیر این صورت گوییم DMU_o ناکارای CCR است.

در مدل‌های DEA کلاسیک برای به‌دست آوردن نمره کارایی DMUها از مدل (۳) استفاده می‌شود و کارا بودن یا نبودن یک واحد را با استفاده از تعریف ۱ به‌دست می‌آید. اما اگر تعداد DMUها بسیار زیاد باشند، به محاسبات و زمان بسیار زیادی نیاز است. همچنین در ارزیابی سازمان‌های بزرگ (در حدود میلیون) با استفاده از تحلیل پوششی داده‌ها، حتی با استفاده از کامپیوترهای با سرعت بالا، به محاسبات و زمان بسیار زیادی نیاز است، چون برای محاسبه کارایی هر واحد تصمیم‌گیرنده در هر بار باید یک مدل برنامه‌ریزی خطی حل شود [۷]. به عبارت دیگر، مدل‌های تحلیل پوششی داده‌های معمولی، قادر به به‌دست آوردن نمره کارایی این تعداد از واحدهای تصمیم‌گیرنده نیستند. برای رفع این مشکل (یعنی به‌دست آوردن نمره کارایی تعداد بسیار زیادی از واحدهای تصمیم‌گیرنده) الگوریتم‌های فراابتکاری پیشنهاد شده که جزییات آن در ادامه تشریح می‌شود.

۳ الگوریتم ژنتیک

از سال ۱۹۶۰ الگوبرداری از پدیده‌های طبیعی برای استفاده در الگوریتم‌های قوی جهت ارایه حل مسایل پیچیده‌ی بهینه‌سازی مورد توجه قرار گرفت که تکنیک‌های محاسبات تکاملی نام گرفتند. یکی از این تکنیک‌ها، الگوریتم ژنتیک است که توسط جان هالند در سال ۱۹۷۵ پیشنهاد شده است [۱۹]. این روش که از روش‌های فراابتکاری الهام گرفته از طبیعت می‌باشد، در واقع یکی از روش‌های جستجوی تصادفی است که بر اساس انتخاب طبیعی و نسل‌شناسی طبیعی کار می‌کند. این الگوریتم دارای تفاوت‌های اساسی با دیگر روش‌های متداول بهینه‌سازی است که گلدبرگ[۲۰] این تفاوت‌ها را به صورت زیر بیان کرده است.

۱. الگوریتم ژنتیک با مجموعه‌ای از جواب‌های کدگذاری شده کار می‌کند نه با خود آن‌ها.
۲. الگوریتم ژنتیک در یک جمعیت از جواب‌ها و یا مجموعه‌ای از آن‌ها شروع به جستجو می‌کند نه با یک جواب و در واقع با کل جامعه کار می‌کند نه با یک فرد جامعه.
۳. الگوریتم ژنتیک از اطلاعات تابع برازش استفاده می‌کند نه از مشتق‌ها یا علوم کمکی دیگر.
۴. الگوریتم ژنتیک از قواعد مبتنی بر احتمال استفاده می‌کند نه از قواعد قطعی.

الگوریتم ژنتیک، الگوریتمی است که با استفاده از الگوهای عملیاتی داروینی در مورد تکثیر بقای احسن و بر اساس فرآیند طبیعی ژنتیک مجموعه‌ای از اشیاء منفرد ریاضی را با میزان تطبیقی خاص به جمعیتی جدید تبدیل می‌کند. این الگوریتم به دلیل ساده و ارایه جوابی با کیفیت به عنوان تکنیکی برای بهینه‌سازی مسایل مهندسی به سرعت توسعه یافت [۲۱و۲۲].

برای راه‌اندازی یک الگوریتم ژنتیک، در ابتدا یک جمعیت اولیه به صورت تصادفی ایجاد شده و به الگوریتم داده می‌شود و در ادامه کار، جمعیت هر نسل جدید با توجه به نسل قبل تولید می‌شود. این کار با انجام سه عمل، تولید مجدد، تقاطع و جهش بر روی کروموزم‌های نسل قبل صورت می‌گیرد. در GA، فرآیند حرکت

به‌گونه‌ای است که جمعیت جدید را به سمت تولید نسل بهتر و حذف کروموزوم های ضعیف‌تر سوق داده می-شود.

در پیوست ۱ مراحل اجرای الگوریتم ژنتیک ارایه شده در این مقاله، نشان داده شده است. با توجه به این الگوریتم‌ها، مراحل اجرای الگوریتم ژنتیک در بخش‌های زیر توضیح داده شده است.

۳-۱ نمایش جواب

یکی از مهم‌ترین تصمیمات در طراحی یک روش فراابتکاری، نحوه نمایش جواب است. نمایش جواب برای تبدیل به کدکردن باید ساده باشد تا زمان پردازش الگوریتم را کاهش دهد. در مساله DEA اندازه جمعیت برابر با تعداد کروموزوم‌ها است به‌طوری که وزن‌های ورودی‌ها و خروجی‌ها، ژن‌های هر کروموزوم را تشکیل می‌دهند. طول کروموزوم‌ها برابر با تعداد ورودی‌ها (m) به علاوه تعداد خروجی‌ها (s)، یعنی $m+s$ می‌باشد و نسل اولیه با این وزن‌های بین صفر و یک عددی مثبت به‌طور تصادفی انتخاب می‌شود.

۳-۲ مکانیسم انتخاب

با توجه به ماکزیمم‌سازی تابع هدف کارایی نسبی در مساله DEA در مدل (۱)، جواب‌های بهتر در برازندگی بالاتری قرار داشته، لذا مقدار برازندگی به عنوان تابع هدف در نظر گرفته می‌شود. با استفاده از مکانیسم انتخاب چرخ رولت، جوابی با مقدار برازندگی بالاتر (یعنی کارایی بهتر) شانس بیشتری برای انتخاب در نسل بعدی دارد.

۳-۳ عملگرهای الگوریتم ژنتیک

۳-۳-۱ تولید مجدد

از آن جایی که والدین بهتر با احتمال بیشتری فرزندان بهتری تولید می‌کنند، بنابراین لازم است که بهترین جواب‌های هر نسل به نسل بعدی منتقل شوند، در نتیجه $pr\%$ از کروموزوم‌هایی با مقدار برازندگی بهتر به نسل بعدی انتقال داده می‌شوند، که این عمل را تولید مجدد گویند.

۳-۳-۲ عملگر تقاطع

هدف اصلی از انجام عملگر تقاطع، جستجو در فضای جواب است و مهم‌ترین عملگر در الگوریتم ژنتیک می‌باشد. این عملگر دو کروموزوم (دو والدین) را از جمعیت نسل قبل می‌گیرد و با نسل بعد معاوضه می‌کند تا فرزندان جدید تولید کند. انواع مختلفی از این عملگر وجود دارد [۲۱ و۲۲] که برخی از آن‌ها عبارتند از: عملگر تک نقطه‌ای، عملگر دونقطه‌ای، عملگر یکنواخت و عملگر حسابی.

۳-۳-۳ عملگر جهش

در الگوریتم ژنتیک بعد از عملگر تقاطع ، عملگر جهش انجام می‌گیرد. عملگر جهش را می‌توان شکل ساده ای از جستجوی محلی در نظرگرفت. هدف اصلی از انجام عملگر جهش، اجتناب از همگرایی بهینه محلی است. عملگرهای جهش فراوانی وجود دارد [۲۱ و۲۲] که برخی از آن‌ها عبارتند از: عملگر جهش تعویض، عملگر جهش تعویض بزرگ، عملگر جهش جابجایی، عملگر جهش آشفته.

٤ ساختار الگوریتم شبیه‌سازی تبرید

الگوریتم شبیه‌سازی تبرید (SA) یک الگوریتم بهینه‌سازی فراابتکاری ساده و اثربخش است که توسط کریک پاتریک و همکارانشان [۲۳] برای حل مسایل بهینه‌سازی پیشنهاد شد. این الگوریتم یک الگوریتم جستجوی محلی یا یک روش فراابتکاری می‌باشد که قادر است خود را از دام بهینه موضعی رها کند. راحتی به کارگیری، همگرایی و استفاده از حرکات خاصی جهت دوری از قرارگیری در دام بهینه موضعی، از جمله خصوصیاتی هستند که باعث شده‌اند تا این روش در دو دهه اخیر مورد توجه قرار گیرد.

در SA ابتدا یک جواب آغازین (S_0) و یک دمای اولیه (T_0) به‌طور تصادفی تولید می‌شود و سپس جواب دیگری در همسایگی این جواب (S') انتخاب می‌شود. قرار می‌دهیم $T = T_0$ و مقدار تغییرات تابع هدف یعنی $\Delta_{s',s_0} = f(S') - f(S_0)$ محاسبه می‌گردد. اگر $\Delta_{s',s_0} < 0$ باشد، جواب S' جایگزین جواب S_0می‌شود، در غیر این صورت، جواب S'، به عنوان جواب جدید با احتمال $p = \exp\left(-\Delta_{s',s_0}/T\right)$ به نسل بعد منتقل می‌شود. روند بیان شده برای هر درجه حرارتی تا تعداد تکرار از پیش تعیین شده ادامه می‌یابد تا زمانی که شرط توقف برقرار شود (رسیدن به دمای نهایی). پس از این که تعداد تکرارها به مقدار معین از پیش تعیین شده رسید، دمای سیستم کاهش می‌یابد و S_0 تقریبی از جواب بهینه است. یک مزیت مهم روش شبیه‌سازی تبرید نسبت به روش‌های دیگر، قابلیت این روش در جلوگیری از گیر افتادن در بهینه موضعی می‌باشد. این الگوریتم از یک جستجوی تصادفی استفاده می‌کند که نه تنها تغییراتی که منجر به بهبود تابع هدف می‌شود را می‌پذیرد، بلکه برخی از تغییراتی را که منجر به عدم بهبود تابع هدف می‌شود را نیز شدنی می‌داند. گام‌های اصلی الگوریتم شبیه‌سازی تبرید در پیوست ۲ نشان داده شده است.

٥ طراحی آزمایشات

٥-١ طراحی مسایل آزمایشی

برای ارزیابی کیفیت جواب روش‌های فراابتکاری توسعه داده شده در این تحقیق، از یک طرح جدید برای تولید داده‌های آزمایشی استفاده می‌شود. داده‌های مورد نیاز شامل تعداد *DMU*ها، ورودی‌ها و خروجی‌ها است. برای اجرای الگوریتم ۳۲ مساله به تصادف تولید می‌شود که در آن هشت مساله برای آزمایش انتخاب شده‌اند. تعداد *DMU*ها اندازه مساله را تعیین می‌کند. برای هر اندازه مساله، چهار نوع زیر مساله، مانند A، B ، C و D در نظر گرفته شده است. در هر اندازه مساله، زیر مسایل در تعداد ورودی‌ها و خروجی‌ها متفاوتند که به ترتیب، زیر

مسایل افزایش می‌یابند. مقادیر ورودی‌ها و خروجی‌ها از توزیع یکنواخت بین بازه [۵۰ و ۱۰] انتخاب می‌شوند. اندازه مسایل و تعداد *DMU*s ، تعداد ورودی‌ها، تعداد خروجی‌ها و محدوده‌ی مقادیر ورودی‌ها و خروجی‌ها در جدول ۱ نشان داده شده است.

با توجه به زمان حل زیاد روش‌های دقیق در سطوح متوسط و بزرگ، حل مساله با نرم افزارهای دقیق امکان‌پذیر نبوده لذا روش فراابتکاری مورد استفاده قرار می‌گیرد. برای اجرای روش فراابتکاری بکار گرفته شده در این پژوهش، از نرم‌افزار MATLAB استفاده شده است. پردازشگر اجرا کننده الگوریتم، یک کامپیوتر با مشخصات دو هسته‌ای 2.8 GHz و 4 GB RAM می‌باشد.

جدول ۱ .تولید مسایل آزمایشی

		نوع مساله (تعداد خروجی و تعداد ورودی)			
اندازه مساله	*DMU*s	A	B	C	D
۱	۵۰	(۴و۴)	(۸و۴)	(۴و۸)	(۸و۸)
۲	۱۰۰	(۵و۵)	(۱۰و۵)	(۵و۱۰)	(۱۰و۱۰)
۳	۱۵۰	(۵و۵)	(۱۰و۵)	(۵و۱۰)	(۱۰و۱۰)
۴	۲۰۰	(۱۰و۱۰)	(۲۰و۱۰)	(۱۰و۲۰)	(۲۰و۲۰)
۵	۴۰۰	(۱۰و۱۰)	(۲۰و۱۰)	(۱۰و۲۰)	(۲۰و۲۰)
۶	۶۰۰	(۱۵و۱۵)	(۳۰و۱۵)	(۱۵و۳۰)	(۳۰و۳۰)
۷	۸۰۰	(۱۵و۱۵)	(۳۰و۱۵)	(۱۵و۳۰)	(۳۰و۳۰)
۸	۱۰۰۰	(۲۰و۲۰)	(۴۰و۲۰)	(۲۰و۴۰)	(۴۰و۴۰)

به منظور نمایش کیفیت جوابِ الگوریتم پیشنهادی، یک راه برای حل مساله استفاده از داده‌های آزمایشی است.

۵-۲ بهینه سازی به روش تاگوچی (تحلیل نسبت S/N)

منظور از بهینه‌سازی به روش تاگوچی، استفاده از تحلیل نسبت سیگنال به نویز (S/N) است. جهت تشخیص و به‌دست آوردن شرایط بهینه از بین اجراهای آزمایشی از تحلیل S/N استفاده می‌شود. بعد از انجام این تحلیل برای هر فاکتور بهترین سطح معرفی می‌گردد. ایده اصلی پنهان در تحلیل S/N این است که در شرایط بهینه، که در مسایل کنترل کیفی و طراحی مطمین بسیار مطلوب است، شرایطی است که در آن حساسیت عملکرد و یا خروجی نسبت به اغتشاش‌ها کم‌ترین مقدار باشد. به عبارت دیگر، می‌توان گفت که در یک آزمایش زمانی شرایط بهینه را داریم که ایجاد تغییرات در خروجی بیشتر بر تغییرات مقادیر پاسخ که تحت کنترل هستند، استوار باشد تا بر مقادیر اغتشاش. لذا تحلیل نسبت S/N مشخص می‌کند که شرایط بهینه مربوط به جایی است که در آنجا، نسبت پاسخ به اغتشاش ماکزیمم باشد. تحلیل نسبت S/N بسته به اینکه نوع مساله از کدام دسته فوق باشد، متفاوت است. در ادامه تحلیل نسبت S/N در سه قسمت بررسی می شود.

قسمت اول مربوط به شرایطی است که در آن هر چه مقدار خروجی کم‌تر باشد بیشتر مورد پسند است. پارامتر تحلیل نسبت S/N را با η نشان می‌دهیم که برای شرایط هر چه کم‌تر بهتر، به صورت رابطه زیر است.

$$\eta_i = -10\log_{10}\left(\frac{1}{n}\sum_{i=1}^{n} y_i^2\right) \quad (۴)$$

که در آن y_i مقدار خروجی به ازای i امین اجرای آزمایش و n تعداد تکرار آزمایش است. از بین مقادیر η، هر کدام که مقدار عددی بزرگ‌تری را نشان دهد، بیانگر شرایط بهینه است.

قسمت دوم مربوط به شرایطی است که در آن مقدار اسمی بهترین شرایط را ایجاد می‌کند. برای به‌دست آوردن پارامتر η_iدر این دسته مسایل از رابطه زیر استفاده می‌کنیم.

$$\eta_i = 10\log_{10}\left(\frac{\mu^2}{\delta^2}\right) \quad (۵)$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} y_i \quad (۶)$$

$$\delta^2 = \frac{1}{n-1}\sum_{i=1}^{n} (y_i - \mu)^2 \quad (۷)$$

که در آن η مقدار میانگین و δ^2 مقدار واریانس است.

قسمت سوم مربوط به مسایلی است که در آن هر چه مقدار خروجی بیشتر باشد، بهتر است. در چنین مسایلی پارامتر η از رابطه زیر به‌دست می‌آید.

$$\eta_i = -10\log_{10}\left(\frac{1}{n}\sum_{i=1}^{n} (\frac{1}{y_i^2})\right) \quad (۸)$$

در کلیه شرایط فوق بزرگ‌ترین مقدار η_i بهترین شرایط را نشان می‌دهد.

زمانی که از طراحی آزمایش کسری استفاده می‌کنیم، به‌دلیل این‌که تعدادی از اجراهای آزمایش از مقدار کل حذف شده‌اند، ممکن است این اشکال وارد باشد که نتوان از بین فاکتورها و سطوح، شرایط بهینه واقعی را استخراج کرد. برای حل این مشکل به جای تمرکز بر تحلیل η_iهای به‌دست آمده برای هر اجرا بر روی η_i مربوط به یک سطح متمرکز می‌شویم. هر سطحی که η بزرگ‌تری به‌دست دهد، مربوط به شرایط بهینه است. η برای یک سطح از میانگین η_iهای اجرایی به‌دست می‌آید که فاکتور مطلوب در سطح مورد نظر بوده است.

۳-۵ تنظیم پارامتر

کیفیت یک الگوریتم فراابتکاری تا حدود زیادی به تعیین مناسب پارامترهای آن بستگی دارد. بنابراین اگر پارامترها به‌طور صحیح تنظیم نشوند، ممکن است که الگوریتم به جواب بهینه همگرا نشود. روش‌های متعددی برای تعیین پارامترها در الگوریتم‌های فراابتکاری وجود دارد. جهت تنظیم پارامترهای الگوریتم‌های توسعه داده شده در این تحقیق، از روش طراحی آزمایش‌های تاگوچی استفاده شده است.

در روش تاگوچی از یک آرایه عمودی برای سازماندهی نتایج آزمایشی استفاده می‌شود. روش کار به این گونه است که یک‌سری از سطوح مختلف پارامترهای مؤثر بر الگوریتم، برمبنای شاخص‌های ورودی که معمولاً از مقدار تابع هدف استفاده می‌شود، مورد بررسی قرار گرفته و بهترین ترکیب پارامترها، بر اساس نتایج حاصل از آزمایش‌های صورت گرفته بر مبنای آرایه عمودی انتخابی، به‌عنوان مقادیر مطلوب جهت تنظیم پارامترها پیشنهاد می‌شود.

در روش تاگوچی نسبت S/N در حکم متغیر نسبت است که تابع هدف در هر اجرا به این نسبت تبدیل می‌شود تا بر طبق آن تصمیم‌گیری شود. در این تحقیق با توجه به نسبت S/N انتخاب شده مناسب ماهیت مسایل این تحقیق، بیشترین نسبت S/N برای هر فاکتوردر هر الگوریتم، به عنوان فاکتور بهینه انتخاب می‌شود.

در الگوریتم ژنتیک ارایه شده، فاکتورهای کنترلی عبارتند از: اندازه جمعیت، درصد تقاطع، احتمال جهش، نوع عملگر تقاطع و نوع عملگر جهش. برای الگوریتم شبیه‌سازی تبرید، درجه حرارت اولیه (T_0)، تعداد جستجوی همسایگی در هر درجه حرارت ($n_{\max}$) و نرخ کاهش درجه حرارت (α) فاکتورهای کنترلی هستند. جدول ۲ پارامترهای مؤثر بر دو الگوریتم پیشنهادی به همراه سطوح و عملگرهای در نظر گرفته شده برای آن‌ها را ارایه می‌دهد.

جدول ۲. فاکتورها و سطوح الگوریتم‌های پیشنهادی

سطوح SA	فاکتورهای SA	سطوح GA	فاکتورهای GA
A(۱) - ۳۰۰	T_0	A(۱) - ۵۰	اندازه جمعیت
A(۲) - ۳۵۰		A(۲) - ۶۰	
A(۳) - ۴۰۰		A(۳) - ۷۰	
B (۱) - ۵۰۰	$n_{\max}$	B (۱) - ۶۰	نرخ تقاطع
B (۲) - ۵۵۰		B (۲) - ۷۰	
B (۳) - ۶۰۰		B (۳) - ۸۰	
		C (۱) - ۰/۰۱	نرخ جهش
C (۱) - ۰/۹۲	α	C (۲) - ۰/۰۲	
C (۲) - ۰/۹۱		C(۳) – ۰/۰۳	
C(۳) – ۰/۹			
		تک نقطه‌ای - D (۱)	نوع تقاطع
		دو نقطه‌ای - D (۲)	
		یکنواخت – D (۳)	
		تعویض – E (۱)	نوع جهش
		تعویض بزرگ – E (۲)	
		آشفته – E (۳)	

سطوح SA	فاکتورهای SA	سطوح GA	فاکتورهای GA
		تغییر مکان - (۴) E	
		وارونگی - (۵) E	

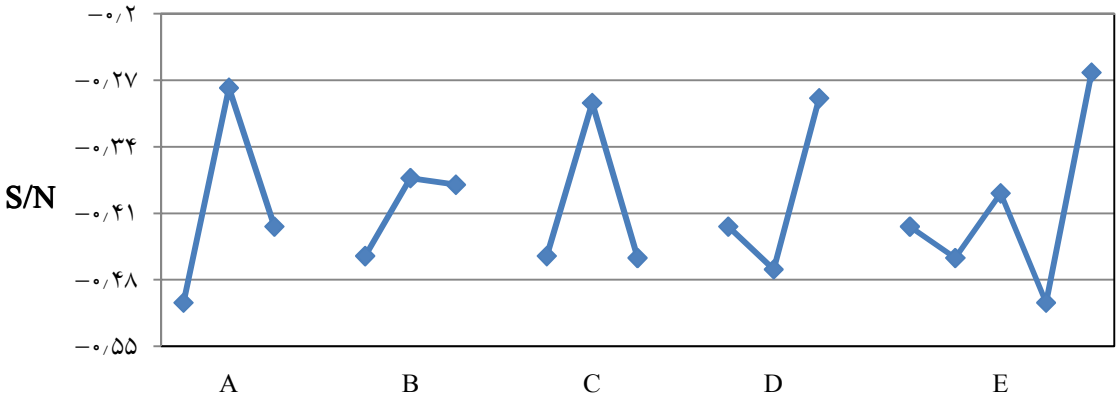
همچنین جهت تحقیق بیشتر و اثبات درستی نتایج اجرای روش تاگوچی، برای نتایج الگوریتم‌های هر مدل، درصد میانگین انحرافات نسبی(RPD) نیز محاسبه شده است.

$$RPD = \frac{\max_{sol} - A\lg_{sol}}{\max_{sol}} \times ۱۰۰ \qquad (۸)$$

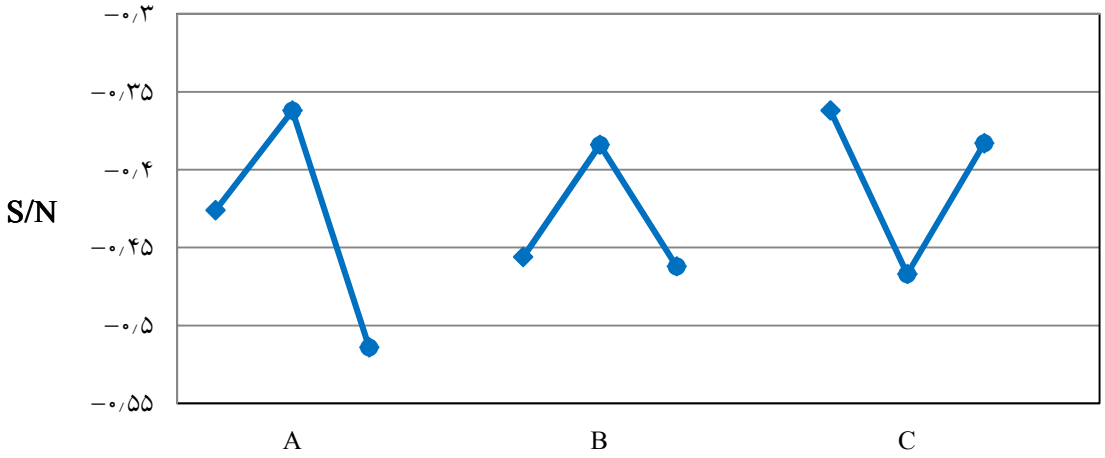
که $A\lg_{sol}$ جواب حاصل از الگوریتم و $\max_{sol}$ ماکزیمم مقدار جواب‌ها است. در این نسبت نیز، بیشترین مقدار RPD برای هر فاکتور در هر الگوریتم به عنوان فاکتور بهینه انتخاب می‌شود.

٥-٤ انتخاب فاکتورهای بهینه الگوریتم‌های حل مساله

پس از به‌دست آوردن نتایج از اجرای آزمایشات در اندازه‌های مختلف، برای تجزیه و تحلیل نتایج الگوریتم‌ها و جهت تعیین پارامترهایی که بهترین جواب را ارایه می‌دهند، نتایج هر آزمایش به نسبت S/N تبدیل می‌شود. هر چقدر مقدار میانگین S/N بیشتر باشد، جواب بهینه‌تر است. نسبت S/N از آزمایشات در هر سطح به طور متوسط، و مقادیر آن‌ها در برابر هر فاکتور کنترلی برای الگوریتم ژنتیک در شکل ۱ و برای الگوریتم شبیه سازی تبرید در شکل ۲ رسم شده است. نتایج به‌دست آمده از آزمایشات با استفاده از نسبت S/N و درصد انحراف نسبی RPD نشان می‌دهد که برای الگوریتم ژنتیک مقادیر بهینه اندازه جمعیت، نرخ تقاطع و نرخ جهش به ترتیب برابر با ۶۰، ۰/۷ و ۰/۰۲ و عملگرهای تقاطع و جهش بهینه به ترتیب عملگر تقاطع یکنواخت و وارونگی می‌باشند. برای الگوریتم شبیه‌سازی تبرید فاکتورهای بهینه درجه حرارت اولیه (T_0)، تعداد جستجوی همسایگی در هر درجه حرارت ($n_{\max}$) و نرخ کاهش درجه حرارت (α) مقادیر بهینه پارامترها به ترتیب برابر ۳۵۰، ۵۵۰ و ۰/۹۲ است.



شکل ۱. انتخاب فاکتورهای بهینه الگوریتم ژنتیک

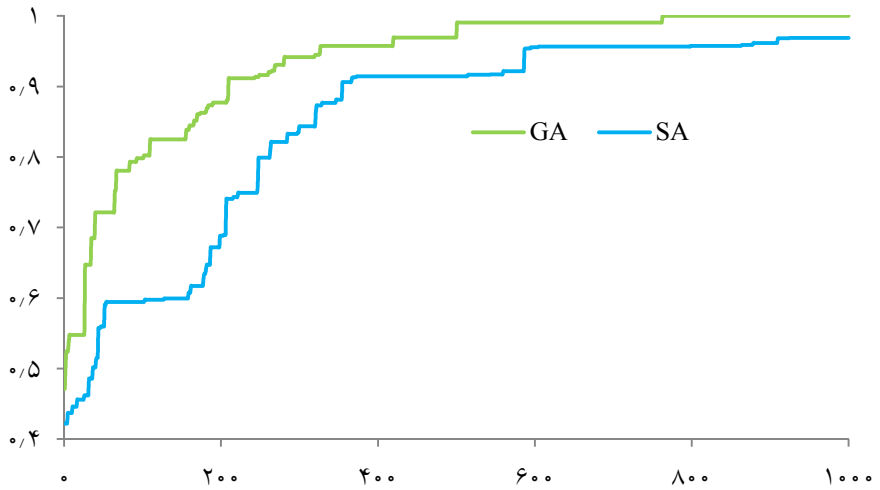


شکل ۲. انتخاب فاکتورهای بهینه الگوریتم شبیه سازی تبرید

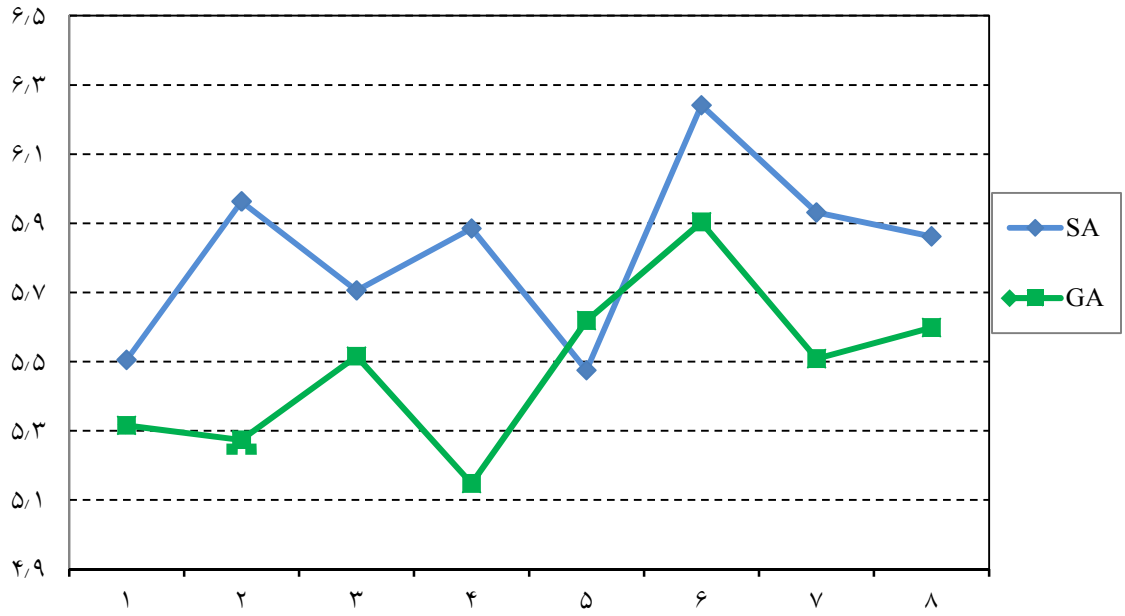
۵-۵ همگرایی الگوریتم‌ها و مقایسه نهایی

پس از تعیین بهترین حالت عملکردی هر الگوریتم با توجه به مقادیر به‌دست آمده، مقایسه بین دو الگوریتم پیشنهادی، انجام خواهد شد. برای منصفانه بودن مقایسه، از معیار زمان یکسان به عنوان شرط توقف در هر دو الگوریتم استفاده می‌شود. برای اطمینان از همگرایی هر الگوریتم به بهترین جواب خود از نمودار همگرایی استفاده شد. نتایج از شکل ۳ نشان می‌دهد که هر دو الگوریتم ژنتیک و شبیه‌سازی تبرید به همگرایی خود رسیدند. لذا افزایش زمان حل تأثیری در بهبود دیگر آن‌ها نخواهد داشت. همان‌گونه که از شکل ۳ مشخص است الگوریتم ژنتیک هم در زمان همگرایی و هم در کیفیت جواب، بهتر از الگوریتم شبیه‌سازی تبرید است.

همان‌طور که از شکل ۴ مشاهده می‌شود، الگوریتم ژنتیک دارای عملکرد بهتری است، زیرا RPD مربوط به این الگوریتم در سطح پایین‌تری قرار دارد. در نتیجه الگوریتم شبیه‌سازی تبرید عملکرد ضعیف‌تری در مقایسه با الگوریتم ژنتیک در این مساله از خود نشان داده است.



شکل ۳. نمودار همگرایی دو الگوریتم ارایه شده در زمان یکسان



شکل ٤. میانگین RPD برای اثر متقابل بین هر الگوریتم و اندازه مساله

٦ نتیجه‌گیری و ارایه پیشنهادات

تحلیل پوششی داده‌ها یکی از شاخه‌های مهم در تحقیق در عملیات است که اخیرا برای ارزیابی واحد های تصمیم گیرنده با مجموعه داده‌های بسیار بزرگ از مدل‌های DEA استفاده شده است. در کاربردهایی از این نوع، به دلیل ابعاد بزرگ مساله مورد بررسی و تعداد زیاد DMU ها، به محاسبات خیلی زیاد نیاز است، لذا روشی که نیاز به حل مساله برای هر DMU که ابعاد و تعداد آن‌ها زیاد می‌باشد نداشته و در زمان معقولی کارایی هر واحد را در اختیار قرار دهد، می‌تواند بسیار حایز اهمیت باشد. برای این منظور، در ابتدا یک طرح تولید داده‌های آزمایشی برای نخستین بار ایجاد گردیده است. با توجه به پیچیدگی حل این مسایل، مدل با روش فراابتکاری الگوریتم ژنتیک و شبیه سازی تبرید حل شده و با بکارگیری تحلیل داده‌ها بر اساس روش‌های ریاضی و آماری و تحلیل آزمایشات عملکرد الگوریتم‌ها و کیفیت جواب‌های به‌دست آمده در عملگرها و پارامترهای متفاوت مورد مقایسه قرار گرفته و بهترین آن‌ها تعیین گردیده است. سپس همگرایی آن‌ها در محور زمان مورد بررسی قرار گرفت که حاکی از همگرایی هر دو الگوریتم به سمت بهترین جواب‌های خود بود. برای بررسی بیشتر عملکرد دو الگوریتم، آن‌ها در اندازه مسایل مختلف با هم مقایسه شدند که الگوریتم ژنتیک نسبت به الگوریتم شبیه‌سازی تبرید برتری داشت.

برخی از موضوعاتی که برای تحقیقات بعدی می‌توان برشمرد، عبارتند از:

١. توسعه دیگر روش‌های فرا ابتکاری مانند الگوریتم بهینه‌سازی گروهی ذرات، الگوریتم ممتیک، الگوریتم زنبور عسل و الگوریتم پرندگان و مقایسه آن‌ها با یکدیگر.
٢. تنظیم پارامترها با دیگر روش‌های طراحی آزمایشات مانند RSM.

منابع

[۲] علیرضایی، م، ر.، بافکر شارک، ن.، (۱۳۹۳). بهبود رویکرد تحلیل پوششی داده‌ها برای اندازه‌گیری شاخص توسعه‌ی انسانی مطالعه‌ی موردی بر روی کشورهای منطقه‌ی آسیا و اقیانوس آرام. مجله تحقیق در عملیات و کاربردهای آن، ۱۱ (۴)،۱۳-۱.

[۳] یوسفی،ش.، داریوش محمدی زنجیرانی، د.، عبدالله زاده، ع، ا.، بررسی عملکرد شعب بانک ملت با تکنیک ترکیبی DEA/AHP (مطالعه‌ی موردی: شعب بانک ملت استان بوشهر). مجله تحقیق در عملیات و کاربردهای آن، ۱۱ (۳)، ۱۲۳-۱۰۹.

[1] Charnes, A., Cooper, W.W., Rhodes, E., (1978). Measuring the efficiency of decision making units, European journal of operational research, 2: 429-444.

[4] Toloo, M., Emrouznejad, A., Moreno, P., (2015). A linear relational DEA model to evaluate two-stage processes with shared inputs. Computational and Applied Mathematics, DOI 10.1007/s40314-014-0211-2.

[5] Kim, K.T., Lee, D.J., Park, S. J., Zhang, Y., Sultanov, A., (2015). Measuring the efficiency of the investment for renewable energy in Korea using data envelopment analysis. Renewable and Sustainable Energy Reviews, 47: 694–702.

[6] Costa, A., Markellos, R. N., (1997). Evaluating public transport efficiency with neural network models. Transportation Research Part C: Emerging Technologies, 301–312.

[7] Emrouznejad, A., Shle, E., (2009). A combined neural network and DEA for measuring efficiency of large scale datasets, Computers & Industrial Engineering, 56: 249-254.

[8] Pendharkar, P., Rodger, J., (2003). Technical efficiency-based selection of learning cases to improve forecasting accuracy of neural networks under monotonicity assumption. Decision Support Systems, 36: 117–136.

[9] Wang, S., (2003). Adaptive non-parametric efficiency frontier analysis: A neural network- based model. Computers and Operations Researchm, 30: 279–295.

[10] Samoilenko, S., Osei-Bryson, K. M., (2010). Determining sources of relative inefficiency in heterogeneous samples: Methodology using Cluster Analysis, DEA and Neural Networks. European Journal of Operational Research, 206: 479–487.

[11] Toloo, M., Zandi, A., Emrouznejad, A., (2015). Evaluation efficiency of large-scale data set with negative data: an artificial neural network approach. Journal of Supercomputing, 71: 2397–2411.

[12] Lagzaie, L., Sanei, M., Molla-Alizadeh-Zavardehi, S., Mahmoodirad, A., (2013). The Measuring Efficiency of Large Scale Datasets in DEA with Metaheuristic Algorithm Approach. Journal of Data Envelopment Analysis and Decision Science, Volume 2013: 1-9.

[13] Rahimian, M., Mahmoodirad, A., Molla-Alizadeh-Zavardehi, S., (2013). Differential evolution Algorithm for Measuring Efficiency in DEA. Scientific Journal of Mechanical and Industrial Engineering, 2(3): 57-60.

[14] Udhayakumar, A., Charles, V., Kumar, M., (2012). Stochastic simulation based genetic algorithm for chance constrained data envelopment analysis problems Original Research Article. Omega, 39: 387-397.

[15] Azadeh, A., Ghaderi, S.F., Tarverdian, S., Saberi, M., (2007). Forecasting electrical consumption by integration of neural network, time series and ANOVA. Applied Mathematics and Computation, 186: 1753-1761.

[16] Thompson, R.G., Dharmapala, E. S., Thrall, R. M., (1995). Linked-cone DEA profit ratios and technical efficiency with spplication to Illinois coal mines. International Journal of Production Economics, 39: 99-115.

[17] Charnes, A., Cooper, W. W., (1962). Programming with linear fractional functional. Naval Research Logistics Quarterly, 9:181–186.

[18] Cooper, W. W., Seiford, L. M., Tone, K., (2007). Introduction to Data Envelopment Analysis, Springer.

[19] Holland, J., (1975). Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press.

[20] Goldberg, D.E., (1989). Genetic Algorithms in Search, Optimization & Machine Learning. Addison-Wesley, Reading, MA.

[21] Hajiaghaei-Keshteli, M., (2011). The allocation of customers to potential distribution centers in supply chain networks: GA and AIA approaches. Applied Soft Computing, 11: 2069–2078.

[22] Mahmoodi-Rad, A., Molla-Alizadeh-Zavardehi, S., Dehghan, R., Sanei, M., Niroomand, S., (2014). Genetic and differential evolution algorithms for the allocation of customers to potential distribution centers in a fuzzy environment. International Journal of Advanced Manufacturing Technology, 70: 1939–1954.
[23] Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., (1983). Optimization by simulated annealing. Science, 220: 671–680.

پیوست ۱.

```
Procedure Genetic Algorithm
    Initialization
    Fitness evaluation
   X_best = Update
   Counter  = 0
   While stopping criterion is not met do
        Reproduction (p_r%)
        Crossover (1- p_r%)
        Mutation (All offspring produced by crossover, p_m)
       Fitness evaluation
       X_best = Update
       X_best = Local search
         If X_best is not improved do
          Counter  = Counter  + 1
          If Counter  > no_change do
             Population = restart phase
             Counter  = 0
          End if
       Else
          Counter  = 0
       End if
End while
```

مراحل اجرای الگوریتم ژنتیک

```
Procedure  Local Search
  k = 1
  While  k < n + 1 do
          v = mutated solution
          If  f(v)< f(x)  do
              x = v
              k = n
              If  f(v)< f(x_best)  then
                  x_best = v
              End if
          End if
          k = k + 1
 End while
```

مراحل اجرای جستجوی محلی

پیوست ۲.

۱. شروع

۲. مقدار دهی اولیه:

۲. ۱. یک جواب اولیه انتخاب کنید، (S_0).

۲. ۲. یک دمای اولیه را انتخاب کنید (T_0).

۲. ۳. نرخ کاهش درجه حرارت (α) را انتخاب کنید.

۳. یک جواب جدید به‌طور تصادفی انتخاب نمایید، $S \in N(S_0)$ و قرار دهید: $\Delta = f(S) - f(S_0)$

۴. اگر $\Delta < 0$، آنگاه قرار دهید: $s_0 = s$، در غیر اینصورت یک عدد تصادفی $r \in (0,1)$ انتخاب نمایید.

۵. اگر $r < e^{-\Delta/T}$ آنگاه قرار دهید: $s_0 = s$.

۶. زمانی که تعداد تکرارها در هر دما برابر nt شود، درجه حرارت ($T = T \times \alpha$) را کاهش دهید.

۷. پایان.

گام‌های الگوریتم شبیه‌سازی تبرید