فصلنامه مدیریت راهبردی در سیستم های صنعتی
(مدیریت صنعتی سابق)
دانشگاه آزاد اسلامی واحد سنندج
سال شانزدهم/ شماره ۵۸ / زمستان ۱۴۰۰

مقایسه و رتبه‌بندی الگوریتم‌های فراابتکاری با استفاده از روش‌های تصمیم‌گیری گروهی

حجت‌اله رجبی مشتاقی
دانشجوی دکتری گروه مدیریت صنعتی، دانشکده مدیریت و اقتصاد، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران
عباس طلوعی اشلقی (نویسنده مسؤول)
استاد گروه مدیریت صنعتی، دانشکده مدیریت و اقتصاد، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران
Email: toloie2020@gmail.com
محمدرضا معتدل
استادیار گروه مدیریت صنعتی، دانشکده مدیریت ، واحد تهران مرکزی، دانشگاه آزاد اسلامی، تهران، ایران

تاریخ دریافت: ۱۴۰۰/۰۱/۰۵ * تاریخ پذیرش ۱۴۰۰/۰۷/۲۴

چکیده

در سال‌های اخیر، شاهد ظهور و گسترش الگوریتم‌های فراابتکاری و استفاده از آنها جهت حل مسائل پیچیده، غیرخطی و NP-hard بوده‌ایم. هدف از انجام این تحقیق رتبه‌بندی الگوریتم‌های فراابتکاری با استفاده از روش‌های تصمیم‌گیری گروهی بوده است. در این راستا، پنج الگوریتم شامل: GA، PSO، ABC، SFLA و ICA انتخاب و با بهره‌گیری از ۱۵ تابع تست استاندارد و هم‌چنین با در نظر گرفتن دو شاخص «میانگین تابع هدف» و «میانگین زمان محاسباتی» مقایسه‌ها انجام شد. در ادامه الگوریتم ها بوسیله سه تکنیک تصمیم‌گیری گروهی شامل:«کوک وسیفرد»، «کندرست» و «دادسون» رتبه‌بندی گردیدند. علاوه بر این، در این پژوهش برای خروج از گره حاصل از یکسان شدن رتبه برخی از گزینه‌ها در روش-های کندرست و دادسون راه حل‌هایی پیشنهاد و سپس الگوریتم‌های تحت بررسی، با روش‌های پیشنهادی نیز رتبه‌بندی شدند. در نهایت رتبه‌بندی کلی با استفاده از یک مدل تخصیص انجام شد، که نتایج آن به صورت زیر است: رتبه یکم PSO، رتبه دوم ICA، رتبه سوم GA، رتبه چهارم ABC و رتبه پنجم SFLA .

کلمات کلیدی: الگوریتم‌های ازدحامی و تکاملی، الگوریتم‌های فراابتکاری، رتبه‌بندی الگوریتم‌های فراابتکاری، روش‌های تصمیم‌گیری گروهی.

۱-مقدمه

در دنیای واقعی مسائل زیادی از قبیل حمل و نقل، جایابی انبارها یا نقاط فروش محصولات، طراحی شبکه های ارتباطی، زمانبندی و ... همگی از مسایل پیچیده و ترکیبی به شمار می‌روند. این مسائل در اندازه‌های عملی بزرگ‌اند و نمی‌توان جواب بهینه آنها را در زمان قابل قبولی به دست آورد با توجه به نیاز به حل این مسائل، چاره‌ای جزء بسنده نمودن به جواب‌های زیر بهینه نبوده است، با این شرط که این راه‌حل‌ها دارای کیفیت قابل پذیرش باشند و در مدت زمان معقولی حاصل شوند. بدین منظور الگوریتم‌های «هیوریستیک[1]» ارائه شدند که می توانند تا اندازه‌ای مناسب عملکرده و گاهی تا حد بهینگی پیش روند. با این حال، یکی از مشکلات حل نشدهٔ آنها قرار گرفتن در بهینه‌های محلی است. برای رفع این مشکل، الگوریتم‌های «متاهیوریستیک (فراابتکاری)[2]» مطرح گردیدند (Alam Tabriz et al., 2013) این الگوریتم‌ها روشی برای حل مسایل بهینه‌سازی‌اندکه با ایجاد جواب (های) تصادفی و براساس عملگرهای خود، با تغییر جواب (های) تصادفی ایجاد شده به سمت بهینگی حرکت می‌کنند (Ghahramani Nahr, 2019).

قابلیت استفاده از الگوریتم‌های متاهیوریستیک (فراابتکاری) برای حل مسائل پیچیده و در حال تغییر دنیای واقعی موجب شد تا به‌کارگیری این الگوریتم‌ها و طراحی آنها به شکلی پویا دنبال شود. همچنین طبق نظریه NFL[3] منطقاً اثبات شده است هیچ الگوریتم فراابتکاری برای حل تمام مسایل بهینه‌سازی جواب درستی ارائه نمی‌دهد (Wolpert and Macready, 1997). به همین دلیل است پیوسته شاهد ارائه الگوریتم‌های جدیدی هستیم. بطوریکه از سال ۲۰۰۰ میلادی تاکنون الگوریتم‌های فراابتکاری زیادی ارائه شده اند. در ارائه الگوریتم‌های جدید عملکرد آن با الگوریتم‌های فراابتکاری دیگر (به عنوان الگوریتم‌های بنچمارک) و با استفاده از توابع تست[4] مورد مقایسه قرار می‌گیرد. از این جهت، استفاده از روش‌های مناسب برای این مقایسات و ارزیابی‌ها ضروریست. در مطالعات قبلی از قبیل: ارائه الگوریتم «کلونی زنبور مصنوعی» (Karaboga and Basturk, 2007). «الگوریتم الهام گرفته شده از خفاش» (Yang, 2010). «بهینه‌ساز گرگ خاکستری» (Mirjalili et al., 2014). «الگوریتم بهینه‌سازی شیر» (Yazdani and Jolai , 2016). «الگوریتم بهینه‌سازی وال» (Mirjalili and Lewis, 2016). «الگوریتم سنجاقک» (Mirjalili, 2016). «جستجوی فراکتال استوکستیک» (Salimi, 2015).«بهینه‌سازی مهاجرت حیوانات» (Li et al., 2014). « الگوریتم گوزن قرمز» (Fathollahi-Fard et al., 2020). « بهینه‌ساز بازی دارت » (Dehghani et al.,2020). صرفاً جواب‌های به دست آمده از هر الگوریتم مورد بررسی قرار گرفته‌اند، اما به شاخص زمان اجرای الگوریتم (سرعت) توجه نشده است. همچنین تحلیل و مقایسه الگوریتم‌ها صرفاً به صورت آماری بوده و از روش‌های رتبه‌بندی پژوهش عملیاتی استفاده نشده است.

در ارائه الگوریتم‌های: «حرکت یونها» (Javidy et al., 2015). «توپ طلایی» (Osaba et al., 2014). «حرکت اسپرم» (Raouf and Hezam, 2017). « الگوریتم بهینه‌سازی ازدحام سوسک » (Wang and Yang, 2018). نیز هر چند به شاخص زمان توجه شده است، با این حال در آنها نیز برای تحلیل و مقایسه الگوریتم‌ها، صرفاً از تحلیل‌های آماری استفاده شده و از تکنیک‌های تصمیم‌گیری استفاده نشده است. در ارائه الگوریتم‌های «مهاجرت حیوانات»، «جستجوی فراکتال استوکستیک»، «بهینه‌سازی حسابی» (Abualigah et al., 2021). نیز هر چند به رتبه اولیه الگوریتم‌های مورد بررسی توجه شده اما برای رتبه‌بندی از مدل خاصی استفاده نشده است، بلکه صرفاً براساس رتبه هر الگوریتم در هر تابع تست به صورت عددی و آماری مورد بررسی قرار گرفته است. لذا در این تحقیق، با هدف رتبه‌بندی الگوریتم‌های فراابتکاری با استفاده از روش‌های تصمیم‌گیری گروهی، جهت پوشش شکاف‌های پژوهشی مذکور، ضمن توجه به دو شاخص میانگین جواب‌ها (شاخص دقت در یافتن جواب بهینه) و زمان اجرای الگوریتم (سرعت)؛ پنج الگوریتم فراابتکاری شامل: «الگوریتم ژنتیک (GA)» ، «ازدحام ذرات (PSO)»،

1. Heuristic
2. Meta-heuristic
3. No Free Lunch
4. Test Functions

«کلونی زنبور مصنوعی[۵] (ABC)» ، «الگوریتم قورباغه جهنده[۶] (SFLA)» و « الگوریتم رقابت استعماری[۷] (ICA)»؛ بوسیله ۱۵ تابع تست استاندارد مورد ارزیابی قرار گرفته و سپس با استفاده از سه تکنیک تصمیم‌گیری گروهی شامل: کوک وسیفرد[۸]، کندرست[۹] و دادسون[۱۰] رتبه‌بندی گردیدند.

با توجه به اینکه در رتبه‌بندی با روش‌های کندرست و دادسون در برخی مواقع، بعضی از گزینه‌ها رتبه یکسان دارند، لذا در این مقاله، راه‌حلی جهت توسعه این روش‌ها در مواقع یکسان شدن رتبه گزینه‌ها پیشنهاد شده است و سپس رتبه‌بندی الگوریتم‌ها با این روش‌ها نیز انجام شده است.

سوالات پژوهشی که در این تحقیق مطرح بوده و به آنها پاسخ داده شده است، عبارتند از: ۱- رتبه‌بندی الگوریتم‌های فراابتکاری به چه صورت می باشد؟ ۲- رتبه‌بندی الگوریتم‌های فراابتکاری با روش‌های مختلف تصمیم‌گیری به چه صورت می باشد؟ ۳- چگونه می‌توان روش‌های کندرست و دادسون را در صورت یکسان بودن رتبه برخی از گزینه‌ها توسعه داد؟

بنابراین نوآوری های این تحقیق عبارتند از:

۱- مقایسه الگوریتم‌ها با در نظر گرفتن هم‌زمان دو شاخص میانگین جواب‌ها و زمان اجرای الگوریتم (دقت و سرعت)

۲- استفاده از چند روش تصمیم‌گیری گروهی برای رتبه‌بندی الگوریتم‌ها و سپس رتبه بندی کلی با استفاده از یک مساله تخصیص

۳- توسعه روش‌های کندرست و دادسون در مواقعی که برخی از گزینه ها رتبه یکسان دارند.

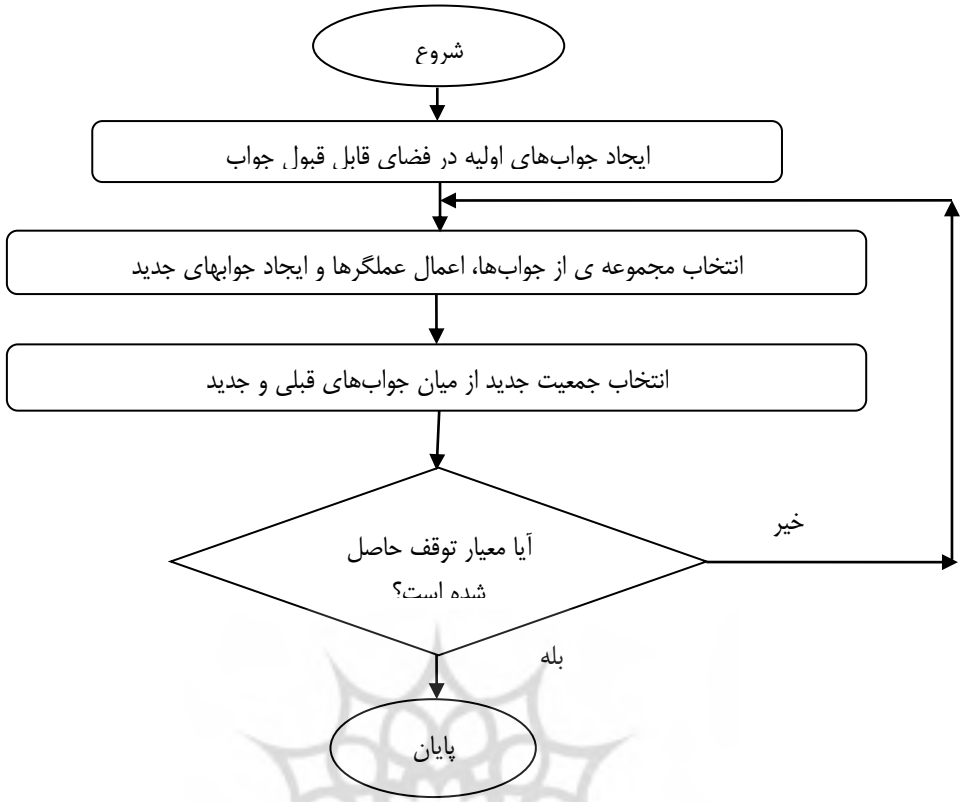
۲- روش‌شناسی تحقیق

الگوریتم‌های فراابتکاری زیر شاخه‌ای از هوش محاسباتی یا محاسبات نرم‌اند و هوش محاسباتی نیز زیر شاخه‌ای هوش مصنوعی است (Mohammad Pour Zarandi, 2013). در سال‌های اخیر استفاده از الگوریتم‌های فراابتکاری[۱۱] برای حل مسایل پیچیده، غیرخطی و NP-hard[۱۲]، گسترش چشمگیری یافته و امروزه کاربردهای وسیعی در شاخه‌های مختلف علوم دارد.

به صورت کلی نحوه کار الگوریتم‌های متاهیوریستیک برای به دست آوردن جواب بهینه، تا اندازه‌ای شبیه هم است. در اغلب این الگوریتم‌ها، جستجو با ایجاد یک یا تعدادی جواب تصادفی (با توجه نوع الگوریتم از لحاظ مبتنی بودن بر یک جواب[۱۳] و یا مبتنی بر جمعیت[۱۴]) در فضای جواب شروع می شود. به جواب‌های اولیه ایجاد شده (در الگوریتم‌های مبتنی بر جمعیت) جمعیت، کلونی، گروه و ... و همچنین به هر یک از جواب ها به تنهایی، کروموزوم، ذره، مورچه و ... گفته می‌شود. سپس با استفاده از عملگرها جواب یا جواب‌های جدید تولید می‌شود. در ادامه با شیوه‌های مختلف، جواب‌هایی از میان جواب‌های قبلی و جدید انتخاب می‌شود و این عمل آنقدر تکرار می‌شود تا معیار توقف حاصل گردد (Sharifzadeh and Amjady, 2014).

شکل ۱ فرایند فوق را نشان می‌دهد (Sharifzadeh and Amjady, 2014).

5. Artificial Bee Colony Algorithm (ABC)
6. Shuffled Frog Leaping Algorithm (SFJA)
7. Imperialist Competitive Algorithm (ICA)
8. Cook & Seiford
9. Condorcet
10. Dodgson
11. Meta-Heuristic Algorithms
12. Nondeterministic Polynomial hard Problem
13. single-solution
14. population-based



شکل شماره (۱): فرایند کلی الگوریتم‌های متاهیوریستیک برای یافتن جواب بهینه

در این مقاله ۱۵ تابع تست استاندارد به شرح جدول ۱ برای ارزیابی پنج الگوریتم فراابتکاری شامل: GA، PSO، ABC، SFLA و ICA انتخاب شده‌اند. این توابع به عنوان توابع تست استاندارد در ارزیابی و مقایسه الگوریتم‌ها که معمولاً در ارائه الگوریتم‌های جدید انجام می‌شود، مورد بهره‌گیری قرار می‌گیرند. همچنین الگوریتم‌های: GA وPSO به عنوان الگوریتم‌های بنچمارک برای مقایسه الگوریتم‌ها مورد استفاده قرار می‌گیرند. بنابراین، در این مقاله این دو الگوریتم به همراه سه الگوریتم دیگر که از الگوریتم‌های شناخته شده و معروف می‌باشند مورد مقایسه قرار گرفته‌اند در ادامه پس از حل هر یک از توابع بوسیله الگوریتم‌های منتخب (با نرم افزار متلب / MATLAB)، مقایسه اولیه الگوریتم‌ها در دو شاخص میانگین تابع هدف (دقت) و میانگین زمان محاسباتی (سرعت) صورت گرفته و رتبه‌بندی اولیه بر این اساس انجام شده است.

در این مقاله برای رتبه‌بندی نهایی الگوریتم‌ها از سه روش تصمیم‌گیری گروهی شامل: کوک و سیفرد، کندرست و دادسون استفاده شده است. مراحل این روش‌ها به صورت زیر است:

الف) روش کوک و سیفرد

این روش برای تعیین رتبه‌بندی نهایی از m گزینه توسط k تصمیم‌گیرنده پایه‌ریزی شده و مراحل آن به شرح زیر است (Asgharpour, 2014).

۱- تعیین رتبه هر گزینه توسط هر یک از تصمیم‌گیرندگان (در این تحقیق توابع تست به عنوان تصمیم گیرنده و الگوریتم-ها به عنوان گزینه‌ها در نظر گرفته شده‌اند).

۲- استفاده از یک تابع متریک یا فاصله برای کمینه کردن عدم توافق در بین رتبه‌بندی ها. در نتیجه رتبه نهایی از کمینه-سازی برای مجموع قدر مطلق فواصل به دست خواهد آمد.

۳- فرض می‌شود $a_{p,i}$ نشان دهنده رتبه واگذار شده به گزینه i ام (از m گزینه) توسط گیرنده p ام (از k تصمیم‌گیرنده) بوده و a_i بیانگر رتبه توافقی برای آن گزینه باشد. لذا به منظور دسترسی به a_i بهینه (i = 1, 2 ... ,m) می بایست مجموع قدر مطلق زیر کمینه گردد:

$$d_p = \sum_{i=1}^{m} \left| a_{pi} - a_i \right| \quad , \; p = 1 , \dots , k$$

بنابراین برای عدم توافق گروهی خواهیم داشت:

$$d = \sum_{p=1}^{k} d_p = \sum_{p=1}^{k} \sum_{i=1}^{m} \left| a_{p,i} - a_i \right|$$

فرض کنیم $a_i = r$ ، $(r = 1 , or\ 2, \dots, or\ m)$ آنگاه:

$d = \sum_{i=1}^{m} d_{i,r}$ به ازای همه $r = 1 , 2 , \dots , m$

بطوریکه $d_{i,r} = \sum_{p=1}^{k} \left| a_{p,i} - r \right|$

لذا مدل زیر با توجه به ماتریس جایگشت H_{m*m} ، به منظور دسترسی به توافق گروهی از رتبه بندی ها ارایه می شود:

$$\min: \sum_{i=1}^{m} \sum_{r=1}^{m} d_{i,r} \cdot h_{i,r}$$

s.t:

$$\sum_{i=1}^{m} h_{i,r} = 1 \quad ; \quad r = 1 , \dots , m$$

$$\sum_{r=1}^{m} h_{i,r} = 1 \quad ; \quad i = 1 , \dots , m$$

$$h_{i,r} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix}$$

ب)روش کندرست

در این روش اولویت‌بندی براساس ماکسی‌مین می‌باشد. بدین صورت که ماتریس مقایسات زوجی تشکیل و سپس مینیمم هر ردیف مشخص و در نهایت اولویت‌بندی، با استفاده از ماکزیمم موجود از بین مینیمم‌ها خواهد بود (Asgharpour, 2014).

با توجه به امکان ایجاد گره و یکسان شدن رتبه برخی از گزینه‌ها در این روش، برای حل این مساله، در این مقاله پیشنهاد شده است، گزینه‌های مشمول گره، براساس ماکسی‌ماکس رتبه‌بندی گردند.

ج) روش دادسون

اولویت‌بندی در این متد، براساس کمترین تغییر لازم گزینه‌ها برای رسیدن به وضعیت برابر با سایر گزینه‌ها می‌باشد. بدین صورت که ماتریس مقایسات زوجی تشکیل و سپس تعداد تغییرات لازم هر گزینه برای رسیدن به وضعیت برابر با سایر گزینه‌ها محاسبه می‌شود. گزینه‌ای که کمترین تغییر را لازم داشته باشد در اولویت بالاتری قرار دارد (Asgharpour, 2014).

با توجه به امکان ایجاد گره و یکسان شدن رتبه برخی از گزینه‌ها در این روش، برای حل این مساله، در این مقاله پیشنهاد شده است، گزینه‌های مشمول گره، براساس داشتن بیشترین ترجیحات رتبه‌بندی گردند. بدین معنی که گزینه‌ها چقدر ترجیح بیشتر از مقداری که لازم دارند تا به وضعیت برابر با سایر گزینه‌ها برسند، دارند.

د)توابع تست (Test Functions)

جدول ۱ توابع تست مورد استفاده در این مقاله را به همراه مشخصات آنها از قبیل: نام تابع، دامنه، تعداد متغیرهای تابع، تک نمایی یا چند نمایی بودن تابع، فرمول و مقدار بهینه سراسری هر تابع، نشان می‌دهد. این توابع از تست‌های استاندارد است که معمولاً در ارزیابی الگوریتم‌ها مورد استفاده قرار می گیرند (Molga and Smutnicki, 2005)، (Kaveh and Talatahari, 2010)، (Mirjalili et al., 2014)، (Tabari and Arshad, 2017)، (Abualigah et al., 2021).

جدول شماره (۱): توابع تست استاندارد به همراه مشخصات آنها

تابع	دامنه	بعد	ویژگی	فرمول تابع		بهینه سراسری
F1 (Sphere)	[-100 , 100]	30	تک نمایی	$F(x) = \sum_{i=1}^{n} X_i^2$	$F(x) =$	$F(x) = 0$
F2 (Sum of different power)	[-1 , 1]	30	تک نمایی	$F(x) = \sum_{i=1}^{n} \lvert X_i \rvert^{i+1}$	$F(x) =$	$F(x) = 0$
F3 (Drop wave)	[-5.12 , 5.12]	2	چند نمایی	$F(x) = \frac{1+\cos(12\sqrt{X_1^2 + X_2^2})}{0.5(X_1^2 + X_2^2) + 2}$	$F(x) =$	$F(x) = -1$
F4 weighted sphere) (model	[-5.12 , 5.12]	30	تک نمایی	$F(x) = \sum_{i=1}^{n} i \cdot X_i^2$	$F(x) =$	$F(x) = 0$
F5 (Griewangk)	[-600 , 600]	30	چند نمایی	$F(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	$F(x) =$	$F(x) = 0$
F6 Moved axis) parallel hyper- (ellipsoid	[-5.12 , 5.12]	30	تک نمایی	$F(x) = \sum_{i=1}^{n} 5 \cdot i \cdot X_i^2$	$F(x) =$	$F(x) = 0$
F7 (Rastrigin)	[-5.12 , 5.12]	30	چند نمایی	$F(x) = 10 \cdot n + \sum_{i=1}^{n} (x_i^2 + 10\cos(2 \cdot \pi \cdot x_i))$	$F(x) =$	$F(x) = 0$
F8 (Rosenbrock)	[-2.048 , 2.048]	30	تک نمایی	$F(x) = \sum_{i=1}^{n} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$	$F(x) =$	$F(x) = 0$
F9 Becker and) (Lago	[-10 , 10]	2	چند نمایی	$F(x) = (\lvert x_1 \rvert - 5)^2 + (\lvert x_2 \rvert - 5)^2$	$F(x) =$	$F(x) = 0$
F10 (Booth)	[-10 , 10]	2	چند نمایی	$F(x) = (X_1 + 2X_2 - 7)^2 + (2X_1 + X_2 - 5)^2$	$F(x) =$	$F(x) = 0$
F11 (Quartic)	[-1.28 , 1.28]	30	تک نمایی	$F(x) = \sum_{i=1}^{n} (ix_i)^4 + Random(0,1)$	$F(x) =$	$F(x) = 0$
F12 (Step)	[-100 , 100]	30	تک نمایی	$F(x) = \sum_{i=1}^{n} (\lvert X_i + 0.5 \rvert)^2$	$F(x) =$	$F(x) = 0$
F13 (Beale)	[-4.5 , 4.5]	2	تک نمایی	$F(x) = (1.5 - X_1 + X_1X_2)^2 + (2.25 - X_1 + X_1X_2^2)^2 + (2.625 - X_1 + X_1X_2^3)^2$	$F(x) =$	$F(x) = 0$
F14 (Colville)	[-10 , 10]	4	تک نمایی	$F(x) = 100(X_1^2 + X_2)^2 + (X_1 - 1)^2 + (X_3 - 1)^2 + 90(X_3^2 + X_4)^2 + 10.1(X_2 - 1)^2 + X_4 - 1)^2 + 19.8(X_2 - 1)(X_4 - 1)$	$F(x) =$	$F(x) = 0$

F15 (Matyas)	[-10 , 10]	2	تک نمایی	$F(x) = 0.26(X_1^2 + X_2^2) - 0.48X_1X_2$	$F(x) = 0$

۳-نتایج و بحث

در این بخش پس از تنظیم پارامترها توابع تست توسط الگوریتم‌های منتخب حل و نتایج تحلیل شده و رتبه‌بندی اولیه انجام می‌گیرد. سپس الگوریتم‌ها با استفاده از تکنیک‌های تصمیم‌گیری گروهی شامل: کوک و سیفرد، کندرست و دادسون رتبه‌بندی نهایی می‌گردند. در این مقاله، جهت تنظیم پارامترها از روش مبتنی بر طرح‌های عاملی کامل استفاده شده است. بدین صورت که شرط خاتمه الگوریتم (معیار توقف)، تعداد تکرار معین قرار داده شد و سپس هر ترکیب از پارامترها ۷ بار اجرا گردید. در نهایت براساس میانگین پاسخ‌ها و ضریب تغییرات، ترکیب بهینه پارامترها انتخاب شدند (Eshghi and Karimi-Nasab, 2016).
در این بخش توابع تست (که در جدول ۱ نشان داده شد)، بوسیله الگوریتم‌های GA، PSO، ABC، SFLA، ICA که همگی از الگوریتم‌های مبتنی بر جمعیت می‌باشند، با تعداد جمعیت و تکرارهای برابر، بوسیله نرم افزار متلب (MATLAB) با ۳۰ بار اجرا (Run) حل شد. که نتایج آن در دو شاخص میانگین جواب‌ها (دقت) و زمان اجرای الگوریتم (سرعت) و رتبه‌بندی اولیه، در جدول ۲ نشان داده می شود.

جدول شماره (۲): نتایج اجرای الگوریتم‌ها در حل توابع تست

توابع			الگوریتم				
			ABC	PSO	SFLA	ICA	GA
F1 (Sphere)	n=30	میانگین جواب	۱/۵۸۷E+۰۰	۸/۲۸۹E-۰۳	۸/۰۸۷E-۰۷	۱/۶۲۱E-۰۲	۳/۵۹۰E+۰۰
		رتبه	۴	۲	۱	۳	۵
		میانگین زمان اجرا	۸/۴۳	۴/۱۱۳	۱۴/۳۰۴	۳/۶۳۰	۷/۸۹۱
		رتبه	۴	۲	۵	۱	۳
F2 (Sum of different power)	n=30	میانگین جواب	۱/۹۳۱E-۰۷	۸/۶۰۶E-۲۱	۵/۲۰۶E-۱۹	۴/۰۰E-۲۷	۳/۲۸۰E-۰۹
		رتبه	۵	۲	۳	۱	۴
		میانگین زمان اجرا	۷/۰۴۶	۴/۸۸۱	۱۹/۷۲۸	۴/۲۵۱	۵/۵۵۸
		رتبه	۴	۲	۵	۱	۳
F3 (Drop wave)	n=2	میانگین جواب	-۹/۹۹۹E -۰۱	-۱	-۱	-۹/۹۵۸E -۰۱	-۱
		رتبه	۲	۱	۱	۳	۱
		میانگین زمان اجرا	۴/۲۳۸	۲/۹۸۰	۵/۰۰۹	۲/۵۰۲	۴/۳۱۰
		رتبه	۳	۲	۵	۱	۴
F4 (weighted sphere model)	n=30	میانگین جواب	۸/۵۳۶E-۰۲	۲/۴۵۳E-۰۴	۴/۰۹۰E-۰۸	۵/۷۵۵E-۰۴	۱/۳۵۷E-۰۱
		رتبه	۴	۲	۱	۳	۵
		میانگین زمان اجرا	۵/۷۲۲	۴/۶۴۳	۱۷/۸۲۷	۳/۷۴۲	۷/۰۹۷
		رتبه	۳	۲	۵	۱	۴
F5 (Griewank)	n=30	میانگین جواب	۱/۰۰۹E+۰۰	۴/۱۰۳E-۰۲	۹/۲۸۲E-۰۳	۵/۱۱۰E-۰۲	۱/۱۰۳E+۰۰
		رتبه	۴	۲	۱	۳	۵

		میانگین زمان اجرا	۵/۴۹۹	۵/۱۰۱	۷/۲۷۴	۴/۰۵۹	۵/۲۶۶
		رتبه	۴	۲	۵	۱	۳
F6 (Moved axis parallel hyper-ellipsoid)	n=30	میانگین جواب	۳/۴۴۵E-۰۱	۱/۳۲۲E-۰۳	۳/۳۵۰E-۰۷	۳/۴۷۳E-۰۳	۶/۳۵۶E-۰۱
		رتبه	۴	۲	۱	۳	۵
		میانگین زمان اجرا	۸/۳۸۸	۴/۷۷۸	۱۸/۹۸۵	۳/۷۷۹	۷/۴۱۳
		رتبه	۴	۲	۵	۱	۳
F7 (Rastrigin)	n=30	میانگین جواب	۲/۱۳۵E+۰۲	۳/۷۸۵E+۰۱	۵/۹۱۳E+۰۱	۳/۶۲۳E+۰۱	۵/۴۸۶E+۰۰
		رتبه	۵	۳	۴	۲	۱
		میانگین زمان اجرا	۵/۷۰۷	۴/۹۳۹	۱۸/۷۲۰	۳/۸۲۴	۷/۳۶۴
		رتبه	۳	۲	۵	۱	۴
F8 (Rosenbrock)	n=30	میانگین جواب	۱/۴۸۳E-۰۷	۱/۰۰۶E-۱۱	۲/۵۷۳E-۰۶	۱/۶۷۷E-۰۳	۸/۸۱۲E-۰۳
		رتبه	۲	۱	۳	۴	۵
		میانگین زمان اجرا	۷/۷۵۳	۳/۷۲۶	۷/۸۸۰	۳/۴۷۴	۴/۵۲۵
		رتبه	۴	۲	۵	۱	۳
F9 (Becker and Lago)	n=2	میانگین جواب	۱/۸۰۲E-۰۸	۵/۳۵۶E-۱۷	۰	۲/۵۵۹E-۲۳	۰
		رتبه	۴	۳	۱	۲	۱
		میانگین زمان اجرا	۵/۸۰۹	۲/۹۴۶	۵/۳۶۴	۲/۵۷۸	۳/۵۷۷
		رتبه	۵	۲	۴	۱	۳
F10 (Booth)	n=2	میانگین جواب	۲/۶۲۰E-۱۳	۴/۰۴۵E-۱۷	۰	۱/۱۰۶E-۱۴	۰
		رتبه	۴	۲	۱	۳	۱
		میانگین زمان اجرا	۵/۹۱۵	۳/۰۹۲	۵/۱۰۴	۲/۴۸۲	۳/۵۰۶
		رتبه	۵	۲	۴	۱	۳
F11 (Quartic)	n=30	میانگین جواب	۷/۸۳۵E-۰۲	۱/۸۴۲E-۰۲	۲/۹۴۹E-۰۲	۱/۱۸۳E-۰۱	۵/۳۰۶E-۰۴
		رتبه	۴	۲	۳	۵	۱
		میانگین زمان اجرا	۶/۱۴۹	۴/۷۰۲	۸/۹۸۱	۳/۷۶۱	۵/۰۸۵
		رتبه	۴	۲	۵	۱	۳
F12 (Step)	n=30	میانگین جواب	۱/۶۲۳E+۰۰	۶/۸۳۵E-۰۳	۱/۵۳۳E-۰۶	۲/۳۲۰E-۰۲	۳/۴۰۰E+۰۰
		رتبه	۴	۲	۱	۳	۵
		میانگین زمان اجرا	۷/۸۳۵	۴/۳۸۱	۱۶/۲۱۰	۳/۵۱۷	۷/۰۴۵
		رتبه	۴	۲	۵	۱	۳
F13 (Beale)	n=2	میانگین جواب	۶/۰۹۹E-۰۸	۱/۹۶۵E-۱۶	۰	۲/۰۸۳E-۱۱	۳/۳۱۹E-۰۵
		رتبه	۴	۲	۱	۳	۵
		میانگین زمان اجرا	۵/۸۳۲	۳/۰۰۶	۷/۰۲۶	۳/۸۴۹	۳/۹۲۷
		رتبه	۴	۱	۵	۲	۳
F14 (Colville (	n=2	میانگین جواب	۱/۷۵۵E-۰۱	۲/۸۳۴E-۰۲	۳/۴۷۹E-۰۴	۴/۵۲۸E-۰۲	۲/۰۴۰E-۰۱
		رتبه	۴	۲	۱	۳	۵

		میانگین زمان اجرا	۵/۱۰۴	۳/۸۱۴	۹/۷۲۶	۵/۰۱۴	۴/۹۲۰
		رتبه	۴	۱	۵	۳	۲
F15 (Matyas)	n=2	میانگین جواب	۱/۶۷۹E-۱۱	۶/۲۹۰E-۱۸	۵/۲۵۲E-۵۴	۹/۱۰۳E-۱۴	۸/۴۰۹E-۱۶
		رتبه	۵	۲	۱	۴	۳
		میانگین زمان اجرا	۴/۲۲۰	۲/۹۲۸	۷/۴۹۱	۳/۶۱۳	۳/۷۳۸
		رتبه	۴	۱	۵	۲	۳

در این بخش الگوریتم‌ها با استفاده از روش‌های تصمیم‌گیری گروهی رتبه‌بندی نهایی می‌گردند (برای حل مدل روش کوک و سیفرد، از نرم افزار لینگو / LINGO استفاده شده است). ابتدا رتبه الگوریتم‌ها در توابع تست محاسبه شده که در جدول ۳ نشان داده می شود.

جدول شماره (۳): رتبه اولیه الگوریتم‌ها

رتبه	الگوریتم				
	ABC	PSO	SFLA	ICA	GA
یکم	۰	۵	۱۱	۱۳	۵
دوم	۲	۲۳	۰	۴	۱
سوم	۳	۲	۳	۱۰	۱۲
چهارم	۲۰	۰	۳	۲	۴
پنجم	۵	۰	۱۳	۱	۸
جمع	۳۰	۳۰	۳۰	۳۰	۳۰

اطلاعات جدول ۳ استخراج شده از جدول ۲ بوده و بدین صورت محاسبه شده است که به عنوان نمونه، الگوریتم ABC در ۳۰ تست انجام یافته (۱۵ تابع در ۲ شاخص)، در هیچ یک از تست‌ها رتبه یکم را کسب نکرده است، ۲ بار رتبه دوم را کسب نموده و به همین ترتیب ۳ بار سوم، ۲۰ بار چهارم و ۵ بار پنجم شده است. این محاسبات به همین صورت برای سایر الگوریتم‌ها نیز انجام یافته و در جدول ۳ نشان داده شده است.

در ادامه مطابق با روش کوک و سیفرد ، ابتدا با استفاده از فرمول زیر $d_{i,r}$ را برای تمام i ها و r ها محاسبه می‌کنیم تا ماتریس فواصل به دست آید.

$$d_{i,r} = \sum_{p=1}^{k} \left| a_{p,i} - r \right|$$

به عنوان نمونه $d_{ABC,1}$ به صورت زیر محاسبه می گردد:

$$d_{ABC,1} = ۰|۱-۱|+۲|۲-۱|+۳|۳-۱|+۲۰|۴-۱| +۵|۵-۱|=۸۸$$

به همین صورت همه $d_{i,r}$ را محاسبه کرده و ماتریس فواصل به دست می‌آید که در جدول ۴ نشان داده شده است.

جدول شماره (۴): ماتریس فواصل

الگوریتم	رتبه				
	یکم	دوم	سوم	چهارم	پنجم
ABC	۸۸	۵۸	۳۲	۱۲	۳۲
PSO	۲۷	۷	۳۳	۶۳	۹۳
SFLA	۶۷	۵۹	۵۱	۴۹	۵۳
ICA	۳۴	۳۰	۳۴	۵۸	۸۶

GA	۶۹	۴۹	۳۱	۳۷	۵۱

در ادامه پس از مدل‌سازی مساله با استفاده از ماتریس فواصل مطابق با مدل زیر که در بخش ۳ توضیح داده شد، با حل این مساله، رتبه‌بندی نهایی به دست خواهد آمد.

$$\min: \sum_{i=1}^{m}\sum_{r=1}^{m} d_{i,r} \cdot h_{i,r}$$

s.t:

$$\sum_{i=1}^{m} h_{i,r} = 1 \quad ; \quad r = 1, \ldots, m$$

$$\sum_{r=1}^{m} h_{i,r} = 1 \quad ; \quad i = 1, \ldots, m$$

$$h_{i,r} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix}$$

```
min=88*h11+58*h12+32*h13+12*h14+32*h15+27*h21+7*h22+33*h23+63*h24+93*h25+6
7*h31+59*h32+51*h33+49*h34+53*h35+34*h41+30*h42+34*h43+58*h44+86*h45+69*h5
1+49*h52+31*h53+37*h54+51*h55;
h11+h12+h13+h14+h15=1;
h21+h22+h23+h24+h25=1;
h31+h32+h33+h34+h35=1;
h41+h42+h43+h44+h45=1;
h51+h52+h53+h54+h55=1;
h11+h21+h31+h41+h51=1;
h12+h22+h32+h42+h52=1;
h13+h23+h33+h43+h53=1;
h14+h24+h34+h44+h54=1;
h15+h25+h35+h45+h55=1;
End
```

پس از حل مساله فوق (بوسیله نرم افزار لینگو) رتبه‌بندی نهایی به دست می آید، که جدول ۵ این رتبه‌بندی را نشان می‌دهد.

جدول شماره (۵): رتبه‌بندی نهایی با روش کوک و سیفرد

الگوریتم	ICA	PSO	GA	ABC	SFLA
رتبه	یکم	دوم	سوم	چهارم	پنجم

تابع f_c را به ازای جدول ۶ (از مقایسات زوجی) در نظر می‌گیریم

جدول شماره (۶): مقایسات زوجی f_c

الگوریتم	ABC	PSO	SFLA	ICA	GA	$f_c = min_j$
ABC	–	۰	۱۴	۳	۱۱	۰
PSO	۳۰	–	۱۹	۱۵	۲۵	۱۵
SFLA	۱۶	۱۰	–	۱۳	۱۰	۱۰
ICA	۲۷	۱۵	۱۷	–	۲۳	۱۵
GA	۱۹	۴	۱۷	۷	–	۷

جدول ۶ از مقایسات زوجی الگوریتم‌ها در رتبه‌بندی اولیه، از جدول ۲ به دست آمده است. به عنوان مثال: ABC در هیچ یک از تست‌ها رتبه بالاتری از PSO نداشته است و به همین ترتیب در ۱۴ تست رتبه بالاتری از SFLA، در ۳ تست رتبه بالاتری از ICA و در ۱۱ تست رتبه بالاتری از GA داشته است. سایر مقایسات نیز به همین ترتیب انجام شده است.

اولویت‌بندی، با استفاده از ماکزیمم موجود از بین مینیمم‌ها (f_c) به صورت زیر است:

$$f_c(\text{PSO, ICA}) > f_c(\text{SFLA}) > f_c(\text{GA}) > f_c(\text{ABC}) \xrightarrow{\text{بنابراین}} \text{ICA} \sim \text{PSO} > \text{SFLA} > \text{GA} > \text{ABC}$$

لذا در این روش، الگوریتم‌های PSO و ICA مشترکاً رتبه اول، SFLA دوم، GA سوم و ABC چهارم (رتبه آخر) را کسب کرده‌اند.

روش پیشنهادی برای تعیین رتبه الگوریتم‌های PSO و ICA که در روش کندرست دارای رتبه یکسان هستند:

در این روش، استفاده از ماکسی‌ماکس ۲ گزینه مشمول گره برای رتبه‌بندی آنها، به شرح جدول ۷ پیشنهاد می‌شود.

جدول شماره (۷): رتبه‌بندی PSO و ICA با روش پیشنهادی

الگوریتم	ABC	PSO	SFLA	ICA	GA	$f_c = \min_j$	$\max_j$
PSO	۳۰	–	۱۹	۱۵	۲۵	۱۵	۳۰
ICA	۲۷	۱۵	۱۷	–	۲۳	۱۵	۲۷

بنابراین، PSO در اولویت بالاتری نسبت به ICA قرار می‌گیرد. لذا، رتبه‌بندی به صورت زیر خواهد بود:

$$\text{PSO} > \text{ICA} > \text{SFLA} > \text{GA} > \text{ABC}$$

۳-۳-۳- رتبه‌بندی با روش دادسون

در این روش، نتایج حاصل از مقایسات زوجی در یک ماتریس به نام ماتریس دادسون ارائه می‌گردد، که در جدول ۸ نشان داده شده است.

جدول شماره (۸): مقایسات زوجی (ماتریس دادسون)

الگوریتم	ABC	PSO	SFLA	ICA	GA
ABC	-	$\frac{0}{30}$	$\frac{14}{16}$	$\frac{3}{27}$	$\frac{11}{19}$
PSO	$\frac{30}{0}$	-	$\frac{19}{10}$	$\frac{15}{15}$	$\frac{25}{4}$
SFLA	$\frac{16}{14}$	$\frac{10}{19}$	-	$\frac{13}{17}$	$\frac{10}{17}$
ICA	$\frac{27}{3}$	$\frac{15}{15}$	$\frac{17}{13}$	-	$\frac{27}{3}$
GA	$\frac{19}{11}$	$\frac{4}{25}$	$\frac{17}{10}$	$\frac{7}{23}$	-

در جدول ۸ به عنوان مثال، ABC در هر ۳۰ تست، از PSO پایین‌تر بوده و لذا برای رسیدن به وضعیت برابر با PSO نیاز به تغییر در ۱۵ تست دارد. سایر اعداد نیز به همین ترتیب محاسبه شده است.

بنابراین، تعداد تغییرات لازم الگوریتم‌ها برای رسیدن به وضعیت برابر با سایر الگوریتم‌ها به صورت زیر می باشد:

ABC: ۱۵+۱+۱۲+۴=۳۲

PSO: ۰

SFLA: ۴/۵+۲+۳/۵=۱۰

ICA: ۰

GA:۱۰ /۵+۸=۱۸/۵

لذا، براساس حداقل تغییر مورد نیاز، اولویت‌بندی به صورت زیر می باشد:

PSO ~ ICA > SFLA > GA > ABC

در این روش نیز الگوریتم‌های PSO و ICA مشترکاً رتبه اول و SFLA ، GA و ABC به ترتیب در رتبه‌های بعدی قرار می‌گیرند.

روش پیشنهادی برای تعیین رتبه الگوریتم‌های PSO و ICA که در روش دادسون دارای رتبه یکسان هستند:

در این روش پیشنهاد شده است، با محاسبه تعداد ترجیحات بیشتر از مقداری که هر گزینه لازم دارند تا به وضعیت برابر با سایر گزینه‌ها برسند، برای رتبه‌بندی استفاده شود. به عبارت دیگر، گزینه مدنظر چقدر از ترجیحات را از دست بدهد همچنان در وضعیت برابر سایر گزینه قرار دارد؟ این محاسبه برای PSO و ICA به صورت زیر است:

PSO: ۱۵+۴/۵+۱۰/۵=۳۰

ICA: ۱۲+۲+۸=۲۲

بنابراین، PSO در اولویت بالاتری نسبت به ICA قرا می‌گیرد. لذا، رتبه‌بندی به صورت زیر خواهد بود:

PSO > ICA > SFLA > GA > ABC

در این قسمت نتایج رتبه‌بندی‌های انجام شده با روش‌های مختلف در جدول ۹ خلاصه می‌شود. سپس در جدول ۱۰ رتبه هر یک از الگوریتم‌ها در ۵ روش رتبه‌بندی، به صورت آماری آورده شده و در نهایت با استفاده از یک مدل تخصیص، رتبه‌بندی کلی انجام می‌شود.

جدول شماره (۹): نتایج رتبه‌بندی ها

روش‌ها	الگوریتم‌ها				
	ABC	PSO	SFLA	ICA	GA
کوک و سیفرد	چهارم	دوم	پنجم	یکم	سوم
کندرست	چهارم	یکم	دوم	یکم	سوم
دادسون	چهارم	یکم	دوم	یکم	سوم
کندرست با روش پیشنهادی	پنجم	یکم	سوم	دوم	چهارم
دادسون با روش پیشنهادی	پنجم	یکم	سوم	دوم	چهارم

جدول شماره (۱۰): رتبه هر الگوریتم در پنج روش رتبه‌بندی به صورت آماری

الگوریتم	رتبه				
	یکم	دوم	سوم	چهارم	پنجم
ABC	۰	۰	۰	۳	۲
PSO	۴	۱	۰	۰	۰
SFLA	۰	۲	۲	۰	۱
ICA	۳	۲	۰	۰	۰
GA	۰	۰	۳	۲	۰

جدول ۱۰ نشان دهنده رتبه هر یک از الگوریتم‌ها در پنج روش رتبه بندی می‌باشد. به عنوان مثال الگوریتم ABC در هیچ یک از روش‌های رتبه‌بندی، رتبه اول، دوم و سوم را به دست نیاورده است و ۳ بار چهارم و ۲ بار هم پنجم شده است. مابقی نیز به همین صورت محاسبه شده است.

با یک مدل تخصیص به صورت زیر، رتبه‌بندی کلی انجام می‌گردد.

```
Max=3*h14+2*h15+4*h21+1*h22+2*h32+2*h33+1*h35+3*h41+2*h42+3*h53+2*h54;
h11+h12+h13+h14+h15=1;
h21+h22+h23+h24+h25=1;
h31+h32+h33+h34+h35=1;
h41+h42+h43+h44+h45=1;
h51+h52+h53+h54+h55=1;
h11+h21+h31+h41+h51=1;
h12+h22+h32+h42+h52=1;
h13+h23+h33+h43+h53=1;
h14+h24+h34+h44+h54=1;
h15+h25+h35+h45+h55=1;
End
```

پس از حل مساله تخصیص فوق (با نرم افزار لینگو) الگوریتم‌ها به صورت کلی رتبه‌بندی شده و نتیجه در جدول ۱۱ نشان داده شده است.

جدول شماره (۱۱): رتبه‌بندی کلی الگوریتم‌ها

الگوریتم	PSO	ICA	GA	ABC	SFLA
رتبه	یکم	دوم	سوم	چهارم	پنجم

در این مقاله، با هدف رتبه‌بندی الگوریتم‌های فراابتکاری با استفاده از روش‌های تصمیم‌گیری گروهی، پنج الگوریتم فراابتکاری شامل: الگوریتم ژنتیک (GA) ، ازدحام ذرات (PSO)، کلونی زنبور مصنوعی (ABC) ، الگوریتم قورباغه جهنده (SFLA) و الگوریتم رقابت استعماری (ICA)؛ برای این منظور انتخاب گردید. سپس مقایسات و ارزیابی‌ها با در نظر گرفتن دو شاخص میانگین تابع هدف و میانگین زمان محاسباتی (سرعت) و بوسیله ۱۵ تابع تست استاندارد صورت پذیرفت. در ادامه الگوریتم‌ها با استفاده از سه تکنیک تصمیم‌گیری گروهی شامل: کوک وسیفرد، کندرست و دادسون رتبه‌بندی گردیدند.

با توجه به اینکه در رتبه‌بندی با روش‌های کندرست و دادسون در برخی مواقع، بعضی از گزینه‌ها رتبه یکسان دارند، لذا در این مقاله برای خروج از گره حاصل از یکسان شدن رتبه برخی از گزینه‌ها در این روش‌ها، راه حل‌هایی پیشنهاد و سپس الگوریتم – های تحت بررسی، با روش‌های پیشنهادی نیز رتبه‌بندی شدند. در نهایت رتبه‌بندی کلی با استفاده از یک مدل تخصیص انجام شده است.

نتایج رتبه‌بندی‌ها به صورت زیر است:

روش کوک و سیفرد: ICA> PSO > GA > ABC > SFLA

روش کندرست: ICA ~ PSO > SFLA > GA > ABC

روش کندرست با اعمال توسعه پیشنهادی: PSO > ICA > SFLA > GA > ABC

روش دادسون: PSO ~ ICA > SFLA > GA > ABC

روش دادسون با اعمال توسعه پیشنهادی: PSO > ICA > SFLA > GA > ABC

رتبه بندی کلی PSO > ICA > GA > ABC > SFLA

در این پژوهش، با اتخاذ رویکردی جدید، الگوریتم‌های فراابتکاری با در نظر گرفتن دو شاخص و با سه روش تصمیم‌گیری گروهی رتبه‌بندی شده‌اند. شیوه رتبه‌بندی پیشنهادی می‌تواند به عنوان روشی بهینه در تحقیقات آینده استفاده شود. هم‌چنین با بهره‌گیری از این روش در پژوهش‌های آتی می‌توان الگوریتم‌های دیگری را نیز مقایسه و رتبه‌بندی کرد.

٤-منابع

1. Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021).The arithmetic optimization algorithm. *Computer methods in applied mechanics and engineering*, 376, 113609.
2. Alam Tabriz, A., Zandieh, M., & Mohammad Rahimi, A. (2013). *Meta-heuristic algorithms in hybrid optimization.* Saffar-Eshraiggi Press. (In Persian).
3. Asgharpour, M. J. (2014). *Group Decision Making and Game Theory in Operation Research*. Tehran, Tehran University press. (In Persian).
4. Dehghani, M., Montazeri, Z., Givi, H., Guerrero, J. M., & Dhiman, G. (2020). Darts game optimizer: a new optimization technique based on darts game. *Int. J. Intell. Eng. Syst*, 13(1), 286-294.
5. Eshghi, K., & Karimi-Nasab, M. (2016).*Analysis of algorithms and Design of metaheuristics.* Tehran, Sharif University Press. (In Persian).
6. Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Tavakkoli-Moghaddam, R. (2020). Red deer algorithm (RDA): a new nature-inspired meta-heuristic. Soft computing, 19(1), 1-29.
7. Ghahramani Nahr, J. (2019). Improve the efficiency and effectiveness of the closed loop supply chain: Wall optimization algorithm and new coding based on priority approach. *Decisions and operations research*, *4*(4), 299-315. (In Persian).
8. Javidy, B., Hatamlou, A., & Mirjalili, S. (2015). Ions motion algorithm for solving optimization problems. *Applied Soft Computing*, 32, 72-79.
9. Karaboga, D., & Basturk. B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization, 39(3), 459–471.
10. Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: charged system Search. Acta Mechanica, 213(3-4), 267-289.
11. Li, X. X., Zhang, J., & Yin, M. (2014). Animal migration optimization: an optimization algorithm inspired by animal migration behavior. *Neural Computing and Applications*, 24(7), 1867–1877.
12. Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, 27(4), 1053-1073.
13. Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95, 51-67.
14. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, 69, 46-61.
15. Mohammad Pour Zarandi, M. E. (2013). *Nonlinear optimization*. Tehran, Tehran University press. (In Persian).
16. Molga, M., & Smutnicki, C. (2005). Test functions for optimization needs. *Test functions for optimization needs*, 101, p. 48.
17. Osaba, E., Diaz, F., & Onieva, E. (2014). Golden ball: a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts. *Applied Intelligence*, 41(1), 145–166.

18. Raouf, O. A., & Hezam, I. M. (2017). Sperm motility algorithm: a novel metaheuristic approach for global optimization. *International Journal of Operational Research (IJOR)*, 28(2), 43-63.
19. Salimi, H. (2015). Stochastic Fractal Search: A powerful metaheuristic algorithm. *Knowledge-Based Systems*, 75, 1-18.
20. Sharifzadeh, H., & Amjady, N. (2014). A Review of metaheuristic algorithms in optimization. Journal of modeling in engineering, 12(38), 27-43. (In Persian). DOI: 10.22075/jme.2017.1677.
21. Tabari, A., & Arshad, A. (2017). A new optimization method: Electro-Search algorithm. Computers and Chemical Engineering, 103, 1–11.
22. Wang, T., & Yang, L. (2018). Beetle swarm optimization algorithm: Theory and application. ArXiv: 1808.00206v2.
23. Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. IEEE transactions on evolutionary computation, 1(1), 67-82.
24. Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. *In Proceedings of the Fourth International Workshop on Nature inspired cooperative strategies for optimization (NICSO 2010), Berlin, Heidelberg*, 65-74.
25. Yazdani, M., & Jolai, F. (2016). Lion Optimization Algorithm (LOA): A nature-inspired meta-heuristic algorithm. *Journal of Computational Design and Engineering*, 3(1), 24-36.

پژوهشگاه علوم انسانی و مطالعات فرهنگی
پرتال جامع علوم انسانی