



## مقایسه و رتبه بندی الگوریتم های فراابتکاری با استفاده از روش های تصمیم گیری 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

محمد رضا 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 سال های اخیر، شاهد ظهور و گسترش الگوریتم های فراابتکاری و استفاده از آنها جهت حل مسائل پیچیده، غیرخطی و NP-hard بوده ایم. هدف از انجام این تحقیق رتبه بندی الگوریتم های فراابتکاری با استفاده از روش های تصمیم گیری گروهی بوده است. در این راستا، پنج الگوریتم شامل: GA، PSO، ABC، SFLA و ICA انتخاب و با بهره گیری از ۱۵ تابع تست استاندارد و همچنین با در نظر گرفتن دو شاخص «میانگین تابع هدف» و «میانگین زمان محاسباتی» مقایسه ها انجام شد. در ادامه الگوریتم ها بوسیله سه تکنیک تصمیم گیری گروهی شامل: «کوک و سیفرد»، «کندرست» و «دادسون» رتبه بندی گردیدند. علاوه بر این، در این پژوهش برای خروج از گره حاصل از یکسان شدن رتبه برخی از گزینه ها در روش های کندرست و دادسون راه حل هایی پیشنهاد و سپس الگوریتم های تحت بررسی، با روش های پیشنهادی نیز رتبه بندی شدند. در نهایت رتبه بندی 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:** الگوریتم های ازدحامی و تکاملی، الگوریتم های فراابتکاری، رتبه بندی الگوریتم های فراابتکاری، روش های تصمیم گیری گروهی.

## ۱- مقدمه

در دنیای واقعی مسائل زیادی از قبیل حمل و نقل، جابجایی انبارها یا نقاط فروش محصولات، طراحی شبکه‌های 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»<sup>۱</sup> ارائه شدند که می‌توانند تا اندازه‌ای مناسب عملکردده و گاهی تا حد بهینگی پیش روند. با این حال، یکی از مشکلات حل نشده آنها قرار گرفتن در بهینه‌های محلی است. برای رفع این مشکل، الگوریتم‌های «متاهیوریستیک (فراابتکاری)»<sup>۲</sup> مطرح گردیدند (Alam Tabriz et al., 2013) این الگوریتم‌ها روشی برای حل مسایل بهینه‌سازی اند که 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<sup>۳</sup> 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 شده‌اند. در ارائه الگوریتم‌های جدید عملکرد آن با الگوریتم‌های فراابتکاری دیگر (به عنوان الگوریتم‌های بنچمارک) و با استفاده از توابع تست<sup>۴</sup> 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»<sup>۵</sup> (ABC)، «الگوریتم قورباغه جهنده»<sup>۶</sup> (SFLA) و «الگوریتم رقابت استعماری»<sup>۷</sup> (ICA)؛ بوسیله ۱۵ تابع تست استاندارد مورد ارزیابی قرار گرفته و سپس با استفاده از سه تکنیک تصمیم‌گیری گروهی شامل: کوک و سیفرد<sup>۸</sup>، کندرست<sup>۹</sup> و دادسون<sup>۱۰</sup> 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.

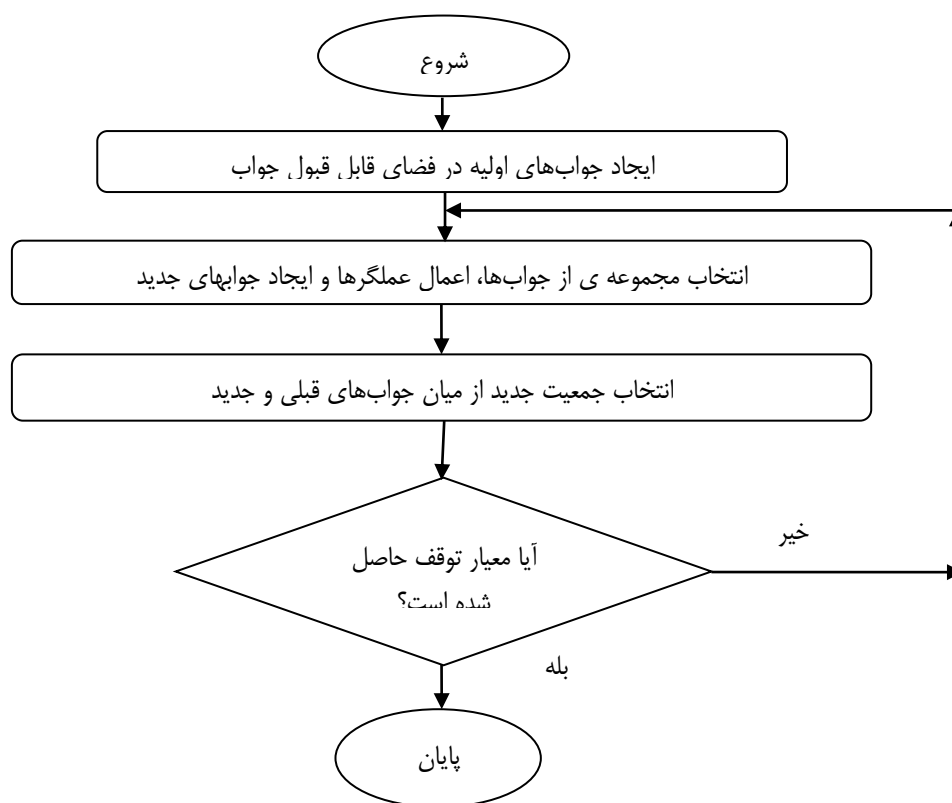
سوالات پژوهشی که در این تحقیق مطرح بوده و به آنها پاسخ داده شده است، عبارتند از: ۱- رتبه‌بندی الگوریتم‌های فراابتکاری به چه صورت می‌باشد؟ ۲- رتبه‌بندی الگوریتم‌های فراابتکاری با روش‌های مختلف تصمیم‌گیری به چه صورت می‌باشد؟ ۳- چگونه می‌توان روش‌های کندرست و دادسون را در صورت یکسان بودن رتبه برخی از گزینه‌ها توسعه داد؟ بنابراین نوآوری‌های 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 گزینه‌ها 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<sup>۱۱</sup> برای حل مسایل پیچیده، غیرخطی و NP-hard<sup>۱۲</sup>، گسترش چشمگیری یافته و امروزه کاربردهای وسیعی در شاخه‌های مختلف علوم دارد. به صورت کلی نحوه کار الگوریتم‌های متاهوریستیک 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<sup>۱۳</sup> و یا مبتنی بر جمعیت<sup>۱۴</sup>) در فضای جواب شروع می‌شود. به جواب‌های اولیه ایجاد شده (در الگوریتم‌های مبتنی بر جمعیت) جمعیت، کلونی، گروه و ... و همچنین به هر یک از جواب‌ها 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 و PSO 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 (Asgarpour, 2014).

- ۱- تعیین رتبه هر گزینه توسط هر یک از تصمیم‌گیرندگان (در این تحقیق توابع تست به عنوان تصمیم‌گیرنده و الگوریتم‌ها به عنوان گزینه‌ها در نظر گرفته شده‌اند).
- ۲- استفاده از یک تابع متریک یا فاصله برای کمینه کردن عدم توافق در بین رتبه‌بندی‌ها. در نتیجه رتبه نهایی از کمینه‌سازی 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 ( $i = 1, 2, \dots, m$ ) می‌بایست مجموع قدر مطلق زیر کمینه گردد:

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

بنابراین برای عدم توافق گروهی خواهیم داشت:

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

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

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

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

لذا مدل زیر با توجه به ماتریس جایگشت  $H_{m \times m}$ ، به منظور دسترسی به توافق گروهی از رتبه بندی‌ها آرایه می‌شود:

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

s.t:

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

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

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

(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 ماکسی‌مکس 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، تک‌نمایی یا چندنمایی 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 می‌گیرند (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

بهبینه سراسری	فرمول تابع	ویژگی	بعد	دامنه	تابع
$F(x) = \sum_{i=1}^n x_i^2$	$F(x) = \sum_{i=1}^n x_i^2$	تک نمایی	30	[-100, 100]	F1 (Sphere)
$F(x) = \sum_{i=1}^n  x_i ^{i+1}$	$F(x) = \sum_{i=1}^n  x_i ^{i+1}$	تک نمایی	30	[-1, 1]	F2 (Sum of different power)
$F(x) = \frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5(x_1^2 + x_2^2) + 2}$	$F(x) = \frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5(x_1^2 + x_2^2) + 2}$	چند نمایی	2	[-5.12, 5.12]	F3 (Drop wave)
$F(x) = \sum_{i=1}^n i \cdot x_i^2$	$F(x) = \sum_{i=1}^n i \cdot x_i^2$	تک نمایی	30	[-5.12, 5.12]	F4 (weighted sphere ) (model)
$F(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$F(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	چند نمایی	30	[-600, 600]	F5 (Griewangk)
$F(x) = \sum_{i=1}^n 5 \cdot i \cdot x_i^2$	$F(x) = \sum_{i=1}^n 5 \cdot i \cdot x_i^2$	تک نمایی	30	[-5.12, 5.12]	F6 (Moved axis ) parallel hyper- (ellipsoid)
$F(x) = 10 \cdot n + \sum_{i=1}^n (x_i^2 + 10 \cos(2 \cdot \pi \cdot x_i))$	$F(x) = 10 \cdot n + \sum_{i=1}^n (x_i^2 + 10 \cos(2 \cdot \pi \cdot x_i))$	چند نمایی	30	[-5.12, 5.12]	F7 (Rastrigin)
$F(x) = \sum_{i=1}^n 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$	$F(x) = \sum_{i=1}^n 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$	تک نمایی	30	[-2.048, 2.048]	F8 (Rosenbrock)
$F(x) = ( x_1  - 5)^2 + ( x_2  - 5)^2$	$F(x) = ( x_1  - 5)^2 + ( x_2  - 5)^2$	چند نمایی	2	[-10, 10]	F9 (Becker and ) (Lago)
$F(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	$F(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	چند نمایی	2	[-10, 10]	F10 (Booth)
$F(x) = \sum_{i=1}^n (ix_i)^4 + Random(0,1)$	$F(x) = \sum_{i=1}^n (ix_i)^4 + Random(0,1)$	تک نمایی	30	[-1.28, 1.28]	F11 (Quartic)
$F(x) = \sum_{i=1}^n ( x_i + 0.5 )^2$	$F(x) = \sum_{i=1}^n ( x_i + 0.5 )^2$	تک نمایی	30	[-100, 100]	F12 (Step)
$F(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	$F(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	تک نمایی	2	[-4.5, 4.5]	F13 (Beale)
$F(x) = 100(x_1^2 + x_2^2) + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 + x_4^2) + 10.1(x_2 - 1)^2 + x_4 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1)$	$F(x) = 100(x_1^2 + x_2^2) + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 + x_4^2) + 10.1(x_2 - 1)^2 + x_4 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1)$	تک نمایی	4	[-10, 10]	F14 (Colville)

$F(x) = \frac{F(x)}{0.26(x_1^2 + x_2^2) - 0.48x_1x_2}$	تک نمایی	2	[-10, 10]	F15 (Matyas)
--	-------------	---	-----------	-----------------

### 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 می‌شود.

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

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

۵/۲۶۶	۴/۰۵۹	۷/۲۷۴	۵/۱۰۱	۵/۴۹۹	میانگین زمان اجرا		
۳	۱	۵	۲	۴	رتبه		
۶/۳۵۶E-۰۱	۳/۴۷۳E-۰۳	۳/۳۵۰E-۰۷	۱/۳۲۲E-۰۳	۳/۴۴۵E-۰۱	میانگین جواب	n=30	F6 (Moved axis parallel hyper- ellipsoid)
۵	۳	۱	۲	۴	رتبه		
۷/۴۱۳	۳/۷۷۹	۱/۹۸۵	۴/۷۷۸	۸/۳۸۸	میانگین زمان اجرا		
۳	۱	۵	۲	۴	رتبه		
۵/۴۸۶E+	۳/۶۲۳E+۰۱	۵/۹۱۳E+۰۱	۳/۷۸۵E+۰۱	۲/۱۳۵E+۰۲	میانگین جواب	n=30	F7 (Rastrigin)
۰	۲	۴	۳	۵	رتبه		
۱	۲	۴	۳	۵	رتبه		
۷/۳۶۴	۳/۸۲۴	۱/۷۲۰	۴/۹۳۹	۵/۷۰۷	میانگین زمان اجرا		
۴	۱	۵	۲	۳	رتبه		
۸/۱۱۲E-۰۳	۱/۶۷۷E-۰۳	۲/۵۷۳E-۰۶	۱/۰۰۶E-۱۱	۱/۴۸۳E-۰۷	میانگین جواب	n=30	F8 (Rosenbrock)
۵	۴	۳	۱	۲	رتبه		
۴/۵۲۵	۳/۴۷۴	۷/۸۸۰	۳/۷۲۶	۷/۷۵۳	میانگین زمان اجرا		
۳	۱	۵	۲	۴	رتبه		
۰	۲/۵۵۹E-۲۳	۰	۵/۳۵۶E-۱۷	۱/۸۰۲E-۰۸	میانگین جواب	n=2	F9 (Becker and Lago)
۱	۲	۱	۳	۴	رتبه		
۳/۵۷۷	۲/۵۷۸	۵/۳۶۴	۲/۹۴۶	۵/۸۰۹	میانگین زمان اجرا		
۳	۱	۴	۲	۵	رتبه		
۰	۱/۱۰۶E-۱۴	۰	۴/۰۴۵E-۱۷	۲/۶۲۰E-۱۳	میانگین جواب	n=2	F10 (Booth)
۱	۳	۱	۲	۴	رتبه		
۳/۵۰۶	۲/۴۸۲	۵/۱۰۴	۳/۰۹۲	۵/۹۱۵	میانگین زمان اجرا		
۳	۱	۴	۲	۵	رتبه		
۵/۳۰۶E-۰۴	۱/۱۸۳E-۰۱	۲/۹۴۹E-۰۲	۱/۸۴۲E-۰۲	۷/۸۳۵E-۰۲	میانگین جواب	n=30	F11 (Quartic)
۱	۵	۳	۲	۴	رتبه		
۵/۰۸۵	۳/۷۶۱	۸/۹۸۱	۴/۷۰۲	۶/۱۴۹	میانگین زمان اجرا		
۳	۱	۵	۲	۴	رتبه		
۳/۴۰۰E+	۲/۳۲۰E-۰۲	۱/۵۳۳E-۰۶	۶/۸۳۵E-۰۳	۱/۶۲۳E+۰۰	میانگین جواب	n=30	F12 (Step)
۰	۳	۱	۲	۴	رتبه		
۵	۳	۱	۲	۴	رتبه		
۷/۰۴۵	۳/۵۱۷	۱۶/۲۱۰	۴/۳۸۱	۷/۸۳۵	میانگین زمان اجرا		
۳	۱	۵	۲	۴	رتبه		
۳/۳۱۹E-۰۵	۲/۰۸۳E-۱۱	۰	۱/۹۶۵E-۱۶	۶/۰۹۹E-۰۸	میانگین جواب	n=2	F13 (Beale)
۵	۳	۱	۲	۴	رتبه		
۳/۹۲۷	۳/۸۴۹	۷/۰۲۶	۳/۰۰۶	۵/۸۳۲	میانگین زمان اجرا		
۳	۲	۵	۱	۴	رتبه		
۲/۰۴۰E-۰۱	۴/۵۲۸E-۰۲	۳/۴۷۹E-۰۴	۲/۸۳۴E-۰۲	۱/۷۵۵E-۰۱	میانگین جواب	n=2	F14 (Colville)
۵	۳	۱	۲	۴	رتبه		



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

n=2      F15 (Matyas)

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 می‌شود.

جدول شماره (۳): رتبه اولیه الگوریتم‌ها

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

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

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

$$d_{ABC,1} = 0 \cdot |1-1| + 2 \cdot |2-1| + 3 \cdot |3-1| + 20 \cdot |4-1| + 5 \cdot |5-1| = 88$$

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

۵۱      ۳۷      ۳۱      ۴۹      ۶۹      GA

---

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

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

s.t:

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

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

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

$$\min = 88 \cdot h_{11} + 58 \cdot h_{12} + 32 \cdot h_{13} + 12 \cdot h_{14} + 32 \cdot h_{15} + 27 \cdot h_{21} + 7 \cdot h_{22} + 33 \cdot h_{23} + 63 \cdot h_{24} + 93 \cdot h_{25} + 67 \cdot h_{31} + 59 \cdot h_{32} + 51 \cdot h_{33} + 49 \cdot h_{34} + 53 \cdot h_{35} + 34 \cdot h_{41} + 30 \cdot h_{42} + 34 \cdot h_{43} + 58 \cdot h_{44} + 86 \cdot h_{45} + 69 \cdot h_{51} + 49 \cdot h_{52} + 31 \cdot h_{53} + 37 \cdot h_{54} + 51 \cdot h_{55};$$

$$h_{11} + h_{12} + h_{13} + h_{14} + h_{15} = 1;$$

$$h_{21} + h_{22} + h_{23} + h_{24} + h_{25} = 1;$$

$$h_{31} + h_{32} + h_{33} + h_{34} + h_{35} = 1;$$

$$h_{41} + h_{42} + h_{43} + h_{44} + h_{45} = 1;$$

$$h_{51} + h_{52} + h_{53} + h_{54} + h_{55} = 1;$$

$$h_{11} + h_{21} + h_{31} + h_{41} + h_{51} = 1;$$

$$h_{12} + h_{22} + h_{32} + h_{42} + h_{52} = 1;$$

$$h_{13} + h_{23} + h_{33} + h_{43} + h_{53} = 1;$$

$$h_{14} + h_{24} + h_{34} + h_{44} + h_{54} = 1;$$

$$h_{15} + h_{25} + h_{35} + h_{45} + h_{55} = 1;$$

End

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

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

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

تابع  $f_c$  را به ازای جدول ۶ (از مقایسات زوجی) در نظر می گیریم

جدول شماره (۶): مقایسات زوجی  $f_c$

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

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

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

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)

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

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 می‌باشد:

$$\text{ABC: } 15 + 1 + 12 + 4 = 32$$

$$\text{PSO: } \cdot$$

$$SFLA: ۴/۵+۲+۳/۵=۱۰$$

$$ICA: ۰$$

$$GA: ۱۰/۵+۸=۱۸/۵$$

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

$$PSO \sim ICA > SFLA > GA > ABC$$

در این روش نیز الگوریتم های PSO و ICA مشترکاً رتبه اول و SFLA ، GA و ABC به ترتیب در رتبه های بعدی قرار می گیرند.

روش پیشنهادی برای تعیین رتبه الگوریتم های 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 گزینه ها برسند، برای رتبه بندی 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 قرار می گیرد. لذا، رتبه بندی به صورت زیر خواهد بود:

$$PSO > ICA > SFLA > GA > ABC$$

در این قسمت نتایج رتبه بندی های انجام شده با روش های مختلف در جدول ۹ خلاصه می شود. سپس در جدول ۱۰ رتبه هر یک از الگوریتم ها در ۵ روش رتبه بندی، به صورت آماری آورده شده و در نهایت با استفاده از یک مدل تخصیص، رتبه بندی کلی انجام می شود.

جدول شماره (۹): نتایج رتبه بندی ها

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

جدول شماره (۱۰): رتبه هر الگوریتم در پنج روش رتبه بندی 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 رتبه‌بندی 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.

$$\begin{aligned} \text{Max} &= 3 \cdot h_{14} + 2 \cdot h_{15} + 4 \cdot h_{21} + 1 \cdot h_{22} + 2 \cdot h_{32} + 2 \cdot h_{33} + 1 \cdot h_{35} + 3 \cdot h_{41} + 2 \cdot h_{42} + 3 \cdot h_{53} + 2 \cdot h_{54}; \\ h_{11} + h_{12} + h_{13} + h_{14} + h_{15} &= 1; \\ h_{21} + h_{22} + h_{23} + h_{24} + h_{25} &= 1; \\ h_{31} + h_{32} + h_{33} + h_{34} + h_{35} &= 1; \\ h_{41} + h_{42} + h_{43} + h_{44} + h_{45} &= 1; \\ h_{51} + h_{52} + h_{53} + h_{54} + h_{55} &= 1; \\ h_{11} + h_{21} + h_{31} + h_{41} + h_{51} &= 1; \\ h_{12} + h_{22} + h_{32} + h_{42} + h_{52} &= 1; \\ h_{13} + h_{23} + h_{33} + h_{43} + h_{53} &= 1; \\ h_{14} + h_{24} + h_{34} + h_{44} + h_{54} &= 1; \\ h_{15} + h_{25} + h_{35} + h_{45} + h_{55} &= 1; \end{aligned}$$

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 و سیفرد، 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، راه‌حلی پیشنهاد و سپس الگوریتم‌های 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:

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

روش کوک و سیفرد:

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

روش کندرست:

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

روش کندرست با اعمال توسعه پیشنهادی:

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

روش دادسون:

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

روش دادسون با اعمال توسعه پیشنهادی:

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

رتبه‌بندی کلی

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.

#### ۴-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., Hajiaghahi-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.