Journal of **N**ovel **R**esearches on **S**mart **P**ower **S**ystems - Vol. 14 - No. 3 – Autumn 2025
P-ISSN: 2322-2468, E-ISSN: 2981-0272, https://jeps.dezful.iau.ir/

Evaluation, Analysis, and Performance Comparison of the Gold Search Algorithm with Several Metaheuristic Algorithms

Ali Akbar Neghabi[*] [1], *Assistance Professor*, Babak Lotfi [2], *Ph.D. Student*, Narges Rahmani [3], *Ph.D. Student*

1- Department of Computer Engineering and Information Technology, Sab.C., Islamic Azad University, Sabzevar, Iran
Aa_neghabi@iau.ac.ir
2- Department of Computer Engineering and Information Technology, Sab.C., Islamic Azad University, Sabzevar, Iran
Babak.lotfi@iau.ac.ir
3- Department of Computer Engineering and Information Technology, Sab.C., Islamic Azad University, Sabzevar, Iran
Narges.rahmani5554@iau.ac.ir

Abstract:

In the present study, the Gold Search Algorithm is compared with six other metaheuristic algorithms, namely Harris Hawks Optimization, Adaptive Opposition-Based Slime Mold Algorithm, Spider Wasp Optimizer, Dandelion Optimizer, Marine Predators Algorithm, and Equilibrium Optimizer. This comparison is conducted by solving ten standard benchmark functions, F1 to F10, across dimensions of 10, 30, and 50, with 100 independent runs. Performance is evaluated using three criteria: accuracy (mean of the best solution), stability (standard deviation), and speed (execution time). The evaluation results and rankings indicate that Harris Hawks Optimization and Adaptive Opposition-Based Slime Mold Algorithm outperform the others in most functions and dimensions in terms of accuracy and stability. The Gold Search Algorithm performs well, especially on functions F1 and F8, in finding the best solutions, and ranks third overall. Regarding speed, the Spider Wasp Optimizer is the fastest, while the Dandelion and Marine Predators algorithms are the slowest. As the dimensionality increases, the efficiency of most algorithms decreases; however, the top-performing algorithms manage to maintain their stability. This study demonstrates that the choice of an appropriate algorithm depends on the nature of the function and the priority of criteria, and no single algorithm is completely superior across all conditions and criteria. Finally, it is suggested that future research employ algorithm hybridization and fine-tuning of parameters to improve performance on more complex problems and real-world applications.

Keywords: Gold Search Algorithm, Adaptive Opposition-Based Slime Mould Algorithm, Harris Hawks Algorithm, Dandelion Algorithm, Metaheuristic Optimization, Performance Comparison.

Received: 29 Sep. 2025
Revised: 11 Nov. 2025
Accepted: 22 Nov. 2025

*** Corresponding Author:** Dr. Ali Akbar Neghabi

Citation: A. Neghabi, B. Lotfi, N. Rahmani, “Evaluation, Analysis, and Performance Comparison of the Gold Search Algorithm with Several Metaheuristic Algorithms”, Journal of Novel Researches on Smart Power Systems, vol. 14, no. 3, pp. 55-80, December 2025 (in Persian).

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]

[DOI: jeps.2025.3.548]

ارزیابی، بررسی و مقایسه عملکرد الگوریتم جستجوی طلا با برخی الگوریتم‌های فراابتکاری

علی اکبر نقابی*[۱]، استادیار، بابک لطفی[۲]، دانشجوی دکتری، نرگس رحمانی[۳]، دانشجوی دکتری

۱- گروه مهندسی کامپیوتر و فناوری اطلاعات، واحد سبزوار، دانشگاه آزاد اسلامی، سبزوار، ایران
Aa_neghabi@iau.ac.ir
۲- گروه مهندسی کامپیوتر و فناوری اطلاعات، واحد سبزوار، دانشگاه آزاد اسلامی، سبزوار، ایران
Babak.lotfi@iau.ac.ir
۳- گروه مهندسی کامپیوتر و فناوری اطلاعات، واحد سبزوار، دانشگاه آزاد اسلامی، سبزوار، ایران
Narges.rahmani5554@iau.ac.ir

چکیده

در پژوهش حاضر، الگوریتم جستجوی طلا با شش الگوریتم فراابتکاری دیگر شامل شاهین هریس، کپک مخاطی با تقابل تطبیقی، زنبور عنکبوتی، قاصدک، شکارچیان دریایی و بهینه‌ساز تعادل مقایسه شده است. این مقایسه با حل ده تابع محک استاندارد، از F1 تا F10 در ابعاد ۱۰، ۳۰ و ۵۰ و در ۱۰۰ تکرار مستقل انجام شده است. برای سنجش عملکرد، سه معیار دقت (میانگین بهترین جواب)، پایداری (انحراف معیار) و سرعت (زمان اجرا) در نظر گرفته شده‌اند. نتایج ارزیابی‌ها و رتبه‌بندی‌ها نشان می‌دهد که الگوریتم‌های شاهین هریس و کپک مخاطی با تقابل تطبیقی در بیشتر توابع و ابعاد، از نظر دقت و پایداری برتری دارند. الگوریتم جستجوی طلا به‌ویژه در توابع F1 و F8 در یافتن بهترین جواب‌ها عملکرد خوبی داشته و در رتبه‌بندی نهایی سوم شده است. از نظر سرعت، الگوریتم زنبور عنکبوتی سریع‌ترین و الگوریتم‌های قاصدک و شکارچیان دریایی کندترین هستند. با افزایش ابعاد، کارایی بیشتر الگوریتم‌ها کاهش یافته، اما الگوریتم‌های برتر توانسته‌اند پایداری خود را حفظ کنند. این مطالعه نشان می‌دهد که انتخاب الگوریتم مناسب به ماهیت تابع و اولویت معیارها بستگی داشته و هیچ الگوریتمی در همه شرایط و برای همه معیارها کاملاً برتر نیست. در پایان، پیشنهاد می‌شود در پژوهش‌های آتی از ترکیب الگوریتم‌ها و تنظیم دقیق پارامترها برای بهبود عملکرد در مسائل پیچیده‌تر و کاربردهای واقعی استفاده شود.

واژه‌های کلیدی: الگوریتم جستجوی طلا، الگوریتم کپک مخاطی با تقابل تطبیقی، الگوریتم شاهین هریس، الگوریتم قاصدک، بهینه‌سازی فراابتکاری، مقایسه عملکرد.

تاریخ دریافت: ۱۴۰۴/۷/۱۰
تاریخ بازنگری: ۱۴۰۴/۸/۲۰
تاریخ پذیرش: ۱۴۰۴/۹/۱

* **نام نویسنده مسئول:** دکتر علی اکبر نقابی
نشانی نویسنده مسئول: گروه مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه آزاد اسلامی واحد سبزوار، سبزوار، ایران

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

۱- مقدمه

الگوریتم‌های فراابتکاری، به‌عنوان دسته‌ای از الگوریتم‌های هوشمند، نقش مهمی در حل مسائل بهینه‌سازی ایفا می‌کنند [۱]. این الگوریتم‌ها، با الهام از فرآیندهای طبیعی [۲] و با بهره‌گیری از سازوکارهای تصادفی و ابتکاری [۳] ، قادرند در فضاهای جستجوی پیچیده و دارای بهینه‌های محلی متعدد، به کاوش بپردازند و راه‌حل‌های تقریبی مناسبی را در زمان معقول ارائه دهند[۴]. تنوع الگوریتم‌های فراابتکاری در سال‌های اخیر رشد چشمگیری داشته و الگوریتم‌های جدیدی با الهام از پدیده‌های گوناگون، از رفتار حیوانات و حشرات [۵] گرفته تا قوانین فیزیک و شیمی [۶] معرفی شده‌اند. یکی از الگوریتم‌های فراابتکاری که اخیراً توجه زیادی را به خود جلب کرده است، الگوریتم بهینه‌ساز جستجوی طلا است. این الگوریتم فراابتکاری در سال ۲۰۲۳ ارائه شده است و از وقایع تاریخی تب طلا و رفتار جستجوگران طلا در آن دوران الهام گرفته است. الگوریتم جستجوی طلا از سه رفتار کلیدی جستجوگران طلا پیروی می‌کند که عبارتند از: مهاجرت، همکاری و استخراج طلا. از مزایای این الگوریتم می‌توان به قابلیت حل مسائل پیچیده، تعادل مناسب بین اکتشاف و بهره‌برداری، و توانایی گریز از بهینه‌های محلی اشاره کرد. با توجه به اهمیت الگوریتم‌های فراابتکاری و کاربرد وسیع آن‌ها در علوم مختلف، ارزیابی عملکرد این الگوریتم‌ها ضروری است. الگوریتم جستجوی طلا اولین بار در پژوهش کامران زلفی در سال ۲۰۲۳ معرفی شده است. در آن پژوهش عملکرد الگوریتم جستجوی طلا[۱] با ۱۲ الگوریتم فراابتکاری شناخته‌شده دیگر از جمله الگوریتم ژنتیک[۲]، بهینه‌سازی ازدحام ذرات[۳]، الگوریتم تکامل تفاضلی[۴]، الگوریتم جستجوی گرانشی[۵]، بهینه‌سازی گرگ خاکستری بهبودیافته[۶]، الگوریتم ازدحام سالپ [۷]، الگوریتم کرم شب‌تاب[۸]، الگوریتم سینوسی کسینوسی[۹]، الگوریتم چرخه آب[۱۰]، الگوریتم بهینه‌سازی نهنگ[۱۱]، الگوریتم کومودو[۱۲] و الگوریتم کپک مخاطی[۱۳] مورد مقایسه قرار گرفته است. [۷]. هدف از پژوهش حاضر، مقایسه عملکرد الگوریتم بهینه‌ساز جستجوی طلا با ۶ الگوریتم دیگر شامل الگوریتم‌های بهینه‌سازی شاهین هریس[۱۴][۸] الگوریتم بهینه‌ساز قاصدک[۱۵][۹]، الگوریتم کپک مخاطی با تقابل تطبیقی[۱۶][۱۰]، الگوریتم بهینه‌ساز تعادل[۱۷][۱۱]، الگوریتم شکارچیان دریایی[۱۸] [۱۲] و الگوریتم زنبور عنکبوتی[۱۹] می‌باشد[۱۳]. این الگوریتم‌ها هر یک با سازوکاری منحصربه‌فرد، به دنبال یافتن راه‌حل‌های بهینه برای مسائل مختلف هستند. برای بررسی عملکرد الگوریتم‌های انتخابی، ۱۰ تابع استاندارد در نظر گرفته شده است. همچنین، از سه معیار بهترین پاسخ[۲۰]، انحراف معیار[۲۱]، زمان اجرا[۲۲] استفاده می‌شود. با مقایسه عملکرد الگوریتم جستجوی طلا با این الگوریتم‌ها، نتایج این مقایسه می‌تواند به درک بهتر نقاط قوت و ضعف هر الگوریتم و انتخاب مناسب‌ترین الگوریتم برای حل مسائل مختلف کمک کند[۱۴].

۲. پیشینه پژوهش

در این بخش، تحقیقات پیشین مرتبط با موضوع مورد بررسی قرار گرفته است. با توجه به گستردگی الگوریتم‌های فراابتکاری و کاربردهای متنوع آن‌ها، تحقیقات زیادی در زمینه مقایسه و ارزیابی عملکرد این الگوریتم‌ها انجام شده است. الگوریتم‌های فراابتکاری طی دو دهه اخیر به‌دلیل توانایی بالا در حل مسائل پیچیده، غیرخطی، نامقید و چندبعدی، به یکی از مهم‌ترین حوزه‌های پژوهشی در بهینه‌سازی تبدیل شده‌اند[۱۵]. گرچه روش‌های کلاسیک مبتنی بر گرادیان در برخی مسائل کارآمد هستند، روش‌های سنتی در مسائل واقعی با مینیمم‌های محلی زیاد، محدودیت غیرخطی و فضای جستجوی بزرگ، کارایی خود را از دست می‌دهند. در این شرایط، الگوریتم‌های فراابتکاری (الهام‌گرفته از طبیعت، زیست‌شناسی، فیزیک و دینامیک) جایگزین قدرتمندی هستند[۱۶]. در این پژوهش، تحقیقات مرتبط با الگوریتم‌های جستجوی طلا، شاهین هریس، قاصدک، کپک مخاطی با تقابل تطبیقی، بهینه‌ساز تعادل، شکارچیان دریایی و زنبور عنکبوتی بررسی شده است. الگوریتم‌های فراابتکاری به دو دسته پایه و توسعه‌یافته تقسیم می‌شوند؛ الگوریتم ژنتیک و ازدحام ذرات از پرکاربردترین‌ها هستند. در یک مطالعه، پنج الگوریتم (کلونی زنبور عسل، جهش قورباغه، رقابت استعماری و...) با ۱۵ تابع تست و معیارهای میانگین تابع هدف و زمان محاسباتی مقایسه شده‌اند. با این حال، در این پژوهش الگوریتم‌های جدیدتر مانند الگوریتم جستجوی طلا و الگوریتم بهینه‌سازی شاهین هریس مورد ارزیابی قرار نگرفتند [۱۷]. در پژوهشی با عنوان «مقایسه عملکرد الگوریتم‌های فراابتکاری با منحنی سنجه رسوب» که در سال ۲۰۲۳ منتشر شد، عملکرد الگوریتم‌هایی مانند بهینه‌سازی ازدحام ذرات الگوریتم ژنتیک، بهینه‌سازی کلونی مورچه‌ها و بهینه‌سازی گرگ خاکستری مورد بررسی قرار گرفت. نتایج این پژوهش نشان داد که الگوریتم گرگ خاکستری عملکرد بهتری

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548] Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

در مقایسه با سایر الگوریتم‌ها داشت[۱۸]. با این حال، در این پژوهش، الگوریتم‌های جدیدتری مانند الگوریتم بهینه‌ساز قاصدک و الگوریتم زنبور عنکبوتی مورد ارزیابی قرار نگرفتند. در مقاله‌ای با عنوان «مقایسه الگوریتم‌های فراابتکاری نوین برای بهینه‌سازی چندهدفه سیستم‌های منابع آب» که در سال ۲۰۲۳ منتشر شد، عملکرد الگوریتم‌های بهینه‌سازی ازدحام ذرات چندهدفه[۲۳]، بهینه‌سازی ملخ چندهدفه[۲۴] و[۲۵] مورد بررسی قرار گرفت. نتایج نشان داد که الگوریتم‌های الگوریتم شیر مورچه چندهدفه و بهینه‌سازی ملخ چندهدفه در مقایسه با بهینه‌سازی ازدحام ذرات چندهدفه کارایی بالاتری دارند[۱۹]. با این حال، در این پژوهش، الگوریتم‌های جدیدتری مانند الگوریتم کپک مخاطی با تقابل تطبیقی و الگوریتم بهینه‌ساز تعادل مورد ارزیابی قرار نگرفتند[۲۰]. در سال ۲۰۲۱، پژوهشی با عنوان «اندازه‌گیری عملکرد الگوریتم‌های بهینه‌سازی فراابتکاری» ارائه کردند. در این تحقیق، استراتژی‌های نوینی برای ارزیابی دقیق عملکرد الگوریتم‌های فراابتکاری معرفی شد که شامل معیاری جدید برای امتیازدهی به الگوریتم‌ها با در نظر گرفتن مقدار نهایی بهینه‌سازی و سرعت همگرایی بود. آن‌ها عملکرد الگوریتم‌هایی مانند بهینه‌سازی ازدحام ذرات، منطق فازی، الگوریتم ژنتیک و تبرید شبیه‌سازی‌شده را بررسی کردند و نتایج نشان داد که الگوریتم ازدحام ذرات بهترین عملکرد را دارد. با این حال، الگوریتم‌های جدیدتری مانند الگوریتم شکارچیان دریایی و الگوریتم بهینه‌ساز شاهین هریس در این مطالعه مورد ارزیابی قرار نگرفتند[۲۱]. در سال ۲۰۲۰، پژوهشی با عنوان "بررسی برخی از الگوریتم‌های فراابتکاری مبتنی بر جمعیت برای آموزش شبکه عصبی چندلایه" ارائه کردند. در این پژوهش، ۱۵ الگوریتم فراابتکاری مبتنی بر جمعیت، از جمله جستجوی گرانشی، خفاش و جستجوی فاخته مورد بررسی قرار گرفتند. نتایج نشان داد که ازدحام ذرات بهترین عملکرد را در آموزش شبکه‌های عصبی داشت. با این حال، در این پژوهش الگوریتم‌های جدیدتر مانند الگوریتم الگوریتم بهینه‌ساز تطبیقی و الگوریتم بهینه‌ساز جستجوی طلا مورد ارزیابی قرار نگرفتند[۲۲].ایزوگو و همکاران در سال ۲۰۱۹، پژوهشی با عنوان "مطالعه مقایسه‌ای الگوریتم‌های بهینه‌سازی فراابتکاری برای مسئله کوله‌پشتی" ارائه کردند. در این پژوهش، عملکرد الگوریتم‌هایی مانند کلونی مورچگان، جستجوی حریصانه و یک الگوریتم ترکیبی شامل جستجوی حریصانه و کلونی مورچگان مورد بررسی قرار گرفت. نتایج نشان داد که الگوریتم ترکیبی بهترین عملکرد را در حل مسئله کوله‌پشتی داشت[۲۳]. در مقاله‌ای با عنوان، مقایسه الگوریتم‌های فراابتکاری نوین در بهره‌برداری از سدهای چندمنظوره که در سال ۲۰۲۳ منتشر شد، عملکرد الگوریتم‌های فراابتکاری شامل چرخه آب ، گرگ خاکستری و ترکیب این دو الگوریتم در حالت‌های تک‌هدفه و چندهدفه بررسی شد. هدف اصلی این پژوهش، بهینه‌سازی مدل بهره‌برداری از سدهای لتیان و دز بود. در حالت تک‌هدفه، کارآمدی الگوریتم‌ها با استفاده از ۸ تابع محک استاندارد ارزیابی شد و سپس در حل مسئله بهره‌برداری بهینه از مخزن لتیان به کار گرفته شدند. همچنین، عملکرد الگوریتم‌ها با شاخص‌های عملکرد مخزن مقایسه شد. نتایج نشان داد که الگوریتم ترکیبی در مقایسه با دو الگوریتم دیگر، عملکرد بهتری دارد. به‌طوری که در مدل بهره‌برداری سد لتیان، الگوریتم ترکیبی با مقدار تابع هدف ۰,۰۱۱، تأمین ۹۸,۵۷ درصد از نیازهای پایین‌دست، میزان کل کمبودهای ۱,۴۳ درصدی و شاخص پایداری ۹۷ درصدی، عملکرد مناسب‌تری نسبت به نهنگ آبی و گرگ خاکستری از خود نشان داد. علاوه بر این، منحنی همگرایی و رهاسازی این الگوریتم نیز نتایج بهتری را نشان داد[۲۴]. با وجود نتایج خوب در برخی الگوریتم‌ها، از الگوریتم جدیدی مانند شکارچیان دریایی استفاده نشده است. بیشتر پژوهش‌های پیشین روی الگوریتم‌های قدیمی متمرکز بوده و الگوریتم‌های جدیدتر (جستجوی طلا، شاهین هریس، قاصدک، کپک مخاطی با تقابل تطبیقی، بهینه‌ساز تعادل، شکارچیان دریایی و زنبور عنکبوتی) کمتر بررسی شده‌اند. با وجود جذابیت این روش‌ها، بررسی ادبیات پژوهشی نشان می‌دهد که:

- تعداد مطالعات مقایسه‌ای که مجموعه‌ای از این الگوریتم‌های نوظهور را هم‌زمان ارزیابی کرده باشند بسیار محدود است .بخش بزرگی از مطالعات تنها به بررسی یک الگوریتم جدید روی چند تابع محک یا یک مسئله کاربردی خاص بسنده کرده‌اند. ارزیابی آماری قوی مانند آزمون‌های ویلکاکسون، فریدمن در اغلب مطالعات غایب است. چارچوب‌های مقایسه‌ای جامع که الگوریتم‌های نسل جدید را هم‌سطح با یکدیگر تحلیل کنند، تقریباً وجود ندارند.

این وضعیت نشان می‌دهد که علی‌رغم توسعه الگوریتم‌های نوین[۲۵]، هنوز تحقیقات کافی برای مقایسه سیستماتیک آن‌ها انجام نشده و اعتبار تجربی بسیاری از این روش‌ها نیازمند بررسی بیشتر است.

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

۳- مرور ادبیات پژوهش

در این بخش، مبانی نظری و پیشینه پژوهش مرور می‌شود. به دلیل پیچیدگی مسائل بهینه‌سازی در مهندسی و علوم داده، روش‌های دقیق اغلب کارایی ندارند و الگوریتم‌های فراابتکاری به‌عنوان راهکارهایی انعطاف‌پذیر و مستقل از مسئله مطرح شده‌اند. هدف، معرفی الگوریتم‌های جدید الهام‌گرفته از طبیعت، فیزیک و رفتار جمعی (مانند جستجوی طلا، شاهین هریس، شکارچیان دریایی، کپک مخاطی تطبیقی، قاصدک، زنبور عنکبوتی و بهینه‌ساز تعادل) است که هرکدام رویکرد متفاوتی برای تعادل بین اکتشاف و بهره‌برداری دارند. این مرور، پایه‌ای برای طراحی و مقایسه الگوریتم پیشنهادی پژوهش فراهم می‌کند.

۳-۱- الگوریتم‌های فراابتکاری

الگوریتم‌های تقریبی به دو دسته اصلی الگوریتم‌های ابتکاری و الگوریتم‌های فراابتکاری تقسیم می‌شوند. [۲۶]. الگوریتم‌های ابتکاری با وجود سادگی و سرعت اجرای بالا، با دو چالش عمده مواجه هستند: اولاً، احتمال گیر افتادن در نقاط بهینه محلی و ثانیاً، همگرایی زودرس به این نقاط. این محدودیت‌ها باعث شده است که الگوریتم‌های فراابتکاری به‌عنوان راه‌حلی کارآمد برای غلبه بر این مشکلات و بهبود عملکرد الگوریتم‌های ابتکاری مطرح شوند [۲]. الگوریتم‌های فراابتکاری، به‌عنوان زیرمجموعه‌ای از الگوریتم‌های بهینه‌سازی تقریبی، با ارائه مکانیزم‌هایی برای فرار از نقاط بهینه محلی و جست‌وجوی گسترده‌تر در فضای جواب، قابلیت کاربرد در مسائل پیچیده و متنوع را دارند[۲۷] در دهه‌های اخیر انواع مختلفی از این نوع الگوریتم‌ها توسعه‌یافته است. واژه "فراابتکاری" را اولین بار "گلوور" به کار برده است [۲۸]. الگوریتم‌های فراابتکاری به‌دلیل توانایی‌هایشان در حل مسائل بهینه‌سازی سخت و پیچیده، در دهه‌های اخیر توسعه چشمگیری داشته‌اند. این الگوریتم‌ها قادرند در مدت‌زمانی معقول، راه‌حل‌های تقریباً بهینه را برای مسائل با مقیاس بزرگ و پیچیدگی بالا ارائه دهند [۴]. مسائل بهینه‌سازی از جنبه‌های مختلفی همچون تعداد اهداف مانند مسائل تک‌هدفه یا چندهدفه، نوع متغیرهای تصمیم مانند متغیرهای پیوسته یا گسسته، و شرایط محدودیت‌ها مانند مسائل مقید یا نامقید قابل دسته‌بندی هستند[۲۹]. به دلیل پیچیدگی ذاتی این مسائل، استفاده از الگوریتم‌های دقیق برای حل آن‌ها، به‌ویژه در مسائل با مقیاس بزرگ، اغلب با چالش‌هایی همچون زمان محاسباتی بالا و ناتوانی در یافتن پاسخ دقیق در بازه زمانی قابل قبول مواجه است، همان‌گونه که در منابع مختلف از جمله منابع [۳۰, ۳۱] اشاره شده است. الگوریتم‌های فراابتکاری بر اساس منبع الهام به سه دسته تقسیم می‌شوند: مبتنی بر رفتار جمعی (شاهین هریس، قاصدک)، مبتنی بر فرایندهای فیزیکی (بهینه‌ساز تعادل) و الهام‌گرفته از فرایندهای زیستی (کپک مخاطی، زنبور عنکبوتی). هر دسته راهبرد متفاوتی برای تعادل بین اکتشاف و بهره‌برداری دارد. در این پژوهش، مجموعه‌ای از الگوریتم‌های نسبتاً جدید و متنوع برای مقایسه عملکرد در مسائل بهینه‌سازی پیوسته انتخاب شده‌اند. معیارهای انتخاب: تازگی، سادگی نسبی، تنوع در نوع الهام و قابلیت پیاده‌سازی عمومی.

۳-۲- مروری بر الگوریتم‌های استفاده شده

در این بخش، مروری مختصر بر الگوریتم‌های فراابتکاری مورد استفاده در این پژوهش ارائه می‌شود. این الگوریتم‌ها عبارت اند از: الگوریتم جستجوی طلا، الگوریتم شاهین هریس، الگوریتم بهینه‌ساز قاصدک، کپک مخاطی با تقابل تطبیقی، الگوریتم بهینه‌ساز تعادل، الگوریتم شکارچیان دریایی، الگوریتم بهینه‌ساز زنبور عنکبوتی. در ادامه، به بررسی هر یک از این الگوریتم‌ها، مکانیزم عملکرد و ویژگی‌های اصلی آن‌ها خواهیم پرداخت. این الگوریتم‌ها از جدیدترین و کارآمدترین الگوریتم‌های فراابتکاری هستند که در سال‌های اخیر معرفی شده‌اند و هر کدام با الهام از پدیده‌های طبیعی و رفتارهای موجودات مختلف، راهکارهای نوینی برای حل مسائل بهینه‌سازی ارائه می‌دهند. این الگوریتم‌ها بر اساس منبع الهام به سه دسته تقسیم می‌شوند: مبتنی بر رفتار جمعی (شاهین هریس، شکارچیان دریایی، قاصدک)، برگرفته از فرآیندهای زیستی (زنبور عنکبوتی، کپک مخاطی با تقابل تطبیقی) و مبتنی بر مدل‌های فیزیکی (بهینه‌ساز تعادل). هدف از این انتخاب، پوشش طیف گسترده‌ای از راهبردهای جستجو است. جدول ۱ خلاصه ویژگی‌ها، مزایا و چالش‌های هر الگوریتم را نشان می‌دهد.

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

جدول ۱: مرور تطبیقی الگوریتم‌های فراابتکاری مورد استفاده در پژوهش
Table 1. A comparative overview of the metaheuristic algorithms employed in this research

نام الگوریتم	سال معرفی	منبع الهام / پدیده طبیعی	دسته‌بندی	مزیت اصلی	چالش / محدودیت
جستجوی طلا	2023	رفتار جستجوگران طلا در دوران تب طلا	Swarm-based	تعادل مناسب بین اکتشاف و بهره‌برداری، سادگی پارامترها	جدید بودن و کمبود مطالعات کاربردی گسترده
شکارچیان دریایی	2020	رفتار شکارچیان دریایی (کوسه، تن، شمشیرماهی)	Swarm-based	ترکیب حرکت براونی و لوی، کشف مؤثر نواحی جدید	وابستگی به پارامترها و نوسان در همگرایی
زنبور عنکبوتی	2023	رفتار زنبورهای عنکبوتی ماده در شکار و جفت‌گیری	Bio-inspired	چهار استراتژی متفاوت برای جستجو، تنوع بالا	پیچیدگی محاسباتی نسبتاً بیشتر
شاهین هریس	2019	رفتار شکار گروهی شاهین‌های هریس	Swarm-based	ساختار ساده، اکتشاف قوی در مراحل اولیه	امکان همگرایی زودرس در ابعاد بالا
قاصدک	2022	پراکنش دانه‌های قاصدک توسط باد	Swarm-based	سه فاز واضح (صعود، نزول، فرود) با تنوع حرکتی	تنظیم حساس پارامترهای α و k
کپک مخاطی با تقابل تطبیقی	2021	رفتار کپک لجن و یادگیری معکوس	Bio-inspired	ترکیب یادگیری معکوس و تطبیقی برای تعادل جستجو	پیچیدگی محاسباتی نسبتاً بالا
بهینه‌ساز تعادل	2019	فرایندهای تعادل فیزیکی و دینامیکی	Physics-based	مدل‌سازی ریاضی شفاف، همگرایی تدریجی پایدار	حساسیت به پارامترهای نرخ تعادل

این تنوع، بستر مناسبی برای مقایسه جامع عملکرد الگوریتم‌ها در مسائل بهینه‌سازی پیوسته فراهم می‌کند. در بخش بعد، نحوه پیاده‌سازی، تنظیم پارامترها و فرآیند ارزیابی تشریح خواهد شد.

۳-۲-۱-الگوریتم بهینه‌ساز جستجوی طلا

الگوریتم جستجوی طلا (۲۰۲۳، توسط کامران زلفی) یک الگوریتم فراابتکاری الهام‌گرفته از رفتار جستجوگران طلا در دوران تب طلا است. این الگوریتم مفاهیم مهاجرت، استخراج و همکاری را شبیه‌سازی می‌کند. در این الگوریتم، جستجوگران طلا به عنوان جمعیت اولیه در نظر گرفته می‌شوند و هر یک موقعیتی متناظر با یک راه‌حل بالقوه دارد. الگوریتم از سه مفهوم اصلی برای به‌روزرسانی موقعیت جستجوگران استفاده می‌کند:

- مهاجرت: جستجوگران به سمت بهترین موقعیت طلا که همان بهترین راه‌حل فعلی است، حرکت می‌کنند.
- استخراج طلا: جستجوگران در اطراف موقعیت فعلی خود به دنبال طلا می‌گردند.
- همکاری: جستجوگران با یکدیگر همکاری می‌کنند تا مناطق جدیدی را برای جستجوی طلا کشف کنند.

مهاجرت جستجوگران پس از کشف یک معدن طلا، جستجوگران به سمت آن حرکت می‌کنند. موقعیت بهترین معدن طلا به عنوان بهترین راه‌حل فعلی در نظر گرفته می‌شود. فرمول به‌روزرسانی موقعیت جستجوگران در فرمول ۱ و ۲ ارائه شده است:

$$D_1 = C_1.\times X*(t) - X_i(t) \quad (۱)$$

$$X_{newi}(t+1) = X_i(t) + A_1 \times D_1 \quad (۲)$$

در روابط ۱ و ۲، X_i بیانگر موقعیت بهترین معدن طلا، X بیانگر موقعیت جستجوگر است. همچنین $A_1.i$ و C_1 ضرایب برداری هستند که به صورت تصادفی محاسبه می‌شوند. استخراج طلا : هر جستجوگر در اطراف موقعیت فعلی خود به دنبال طلا می‌گردد. این فرآیند با استفاده از روابط ۳ و ۴مدل‌سازی می‌شود.

$$D_2 = X_i(t) - X_r(t) \quad (۳)$$

$$X_{newi}(t+1) = X_r(t) + A_2 \times D_2 \quad (۴)$$

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

در روابط ۳ و ۴ X_r بیانگر موقعیت یک جستجوگر تصادفی و A_2 نیز بیانگر ضریب برداری است که به صورت تصادفی محاسبه می‌شود. همکاری بین جستجوگران: جستجوگران با یکدیگر همکاری می‌کنند تا مناطق جدیدی را برای جستجوی طلا کشف کنند. این فرآیند با استفاده از روابط ۵ و ۶ مدل‌سازی می‌شود:

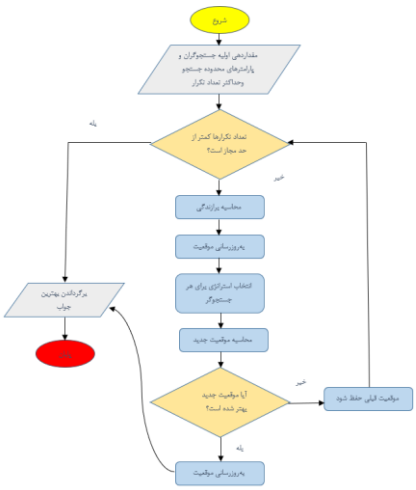
$$D_3 = X_{g2}(t) - X_{g1}(t) \quad (۵)$$

$$X_{newi}(t+1) = X_i(t) + r_1 \times D_3 \quad (۶)$$

در روابط ۵ و ۶ X_{g1} و X_{g2} بیانگر موقعیت دو جستجوگر تصادفی هستند همچنین r_1 یک بردار تصادفی با توزیع یکنواخت در بازه [۰ .۱] است. به‌روزرسانی موقعیت جستجوگران: پس از محاسبه موقعیت جدید، جستجوگران موقعیت خود را به‌روزرسانی می‌کنند. اگر موقعیت جدید بهتر از موقعیت قبلی باشد، جستجوگر به موقعیت جدید حرکت می‌کند. این فرآیند با استفاده از فرمول ۷ مدل‌سازی می‌شود:

$$X_i(t+1) = \{X_{newi}(t+1)\ if\ f(X_{newi}(t+1)) < f(X_i(t)) \quad (۷)$$

الگوریتم جستجوی طلا با استفاده از پارامترهای A_1 و A_2، تعادل بین اکتشاف و بهره‌برداری را تنظیم می‌کند. مقادیر کوچک‌تر برای این پارامترها تمایل الگوریتم به بهره‌برداری را افزایش می‌دهد، در حالی که مقادیر بزرگ‌تر تمایل به اکتشاف را تقویت می‌کند. الگوریتم بهینه‌سازی جستجوی طلا یک الگوریتم فراابتکاری جدید و الهام‌گرفته از رفتار جستجوگران طلا در دوران تب طلا است. این الگوریتم با ترکیب مفاهیم مهاجرت، استخراج طلا و همکاری، توانایی خوبی در یافتن راه‌حل‌های بهینه برای مسائل پیچیده دارد. همچنین، با تنظیم پارامترهای مختلف، تعادل مناسبی بین اکتشاف و بهره‌برداری ایجاد می‌کند. استراتژی همکاری و مکانیزم رها کردن معدن و بازگشت به خانه باعث افزایش تنوع می‌شوند، در حالی که استراتژی‌های مهاجرت و کاوش باعث تشدید جستجو در مناطق پرامید می‌شوند.



شکل ۱: فلوچارت الگوریتم جستجوی طلا

Figure 1. Flowchart of the Gold Search algorithm

۳-۲-۲- الگوریتم بهینه‌سازی شکارچیان دریایی

یک الگوریتم فراابتکاری است که در سال ۲۰۲۰ توسط افشین فرامرزی و همکارانش ارائه شد. این الگوریتم از رفتار شکارچیان دریایی مانند کوسه‌ها، ماهی‌های تن و شمشیرماهی‌ها در طبیعت الهام گرفته است. شکارچیان دریایی از استراتژی‌های خاصی مانند حرکات براونی[۲۶] و لِوی[۲۷] برای یافتن طعمه استفاده می‌کنند. این حرکات بر اساس نظریه جستجوی بهینه طراحی شده‌اند و به شکارچیان کمک می‌کنند تا در محیط‌های مختلف با تراکم طعمه‌های کم یا زیاد، بهینه عمل کنند. در این الگوریتم، فرض بر این است که تعدادی شکارچی و طعمه در یک فضای چندبعدی وجود دارند. موقعیت هر شکارچی و طعمه در این فضا با یک

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

بردار مشخص می‌شود. شکارچیان برای یافتن طعمه تلاش می‌کنند و بهترین موقعیتی که تاکنون به دست آورده‌اند در حافظه‌ای ذخیره می‌شود. الگوریتم از سه مرحله اصلی تشکیل شده است که هر یک نشان‌دهنده سناریوهای مختلف جستجوی شکارچیان است:

جستجوی سراسری: در این مرحله، شکارچیان از حرکت براونی استفاده می‌کنند. حرکت براونی شامل گام‌های کوتاه و تصادفی است که به شکارچیان کمک می‌کند تا فضای جستجو را به‌طور گسترده بررسی کنند. این مرحله برای کشف مناطق جدید در فضای جستجو طراحی شده است. این فرآیند با استفاده از روابطه ۸ و ۹ مدل‌سازی می‌شود.

$$Stepsize_i = R_B \otimes (Elite_i - R_B \otimes Prey_i) \quad (۸)$$

$$Prey_i = Prey_i + P.R \otimes stepsize_i \quad (۹)$$

در روابط ۸ و ۹ R_B یک بردار تصادفی با توزیع نرمال با حرکت براونی است. علامت $\otimes$ بیانگر ضرب عنصر به عنصراست. همچنین $P = 0.5$ یک ثابت است و R یک بردار تصادفی با توزیع یکنواخت در بازه [۰ .۱] است.

ترکیب جستجوی سراسری و محلی: در این مرحله، الگوریتم به دو بخش تقسیم می‌شود:

نیمی از جمعیت از حرکت براونی برای جستجوی سراسری استفاده می‌کنند. و نیمی دیگر از حرکت لِوی برای جستجوی محلی بهره می‌برند. این فرآیند با استفاده از روابط ۱۰ الی ۱۳ مدل‌سازی می‌شود.

- برای نیم اول جمعیت (حرکت لِوی):

$$stepsize_i = R_L \otimes (Elite_i - R_L \otimes Prey_i) \quad (۱۰)$$

$$Prey_i = Prey_i + P.R \otimes stepsize_i \quad (۱۱)$$

- برای نیم دوم جمعیت (حرکت براونی):

$$stepsize_i = R_B \otimes (R_B \otimes Elite_i - Prey_i) \quad (۱۲)$$

$$Prey_i = Prey_i + P.R \otimes stepsize_i \quad (۱۳)$$

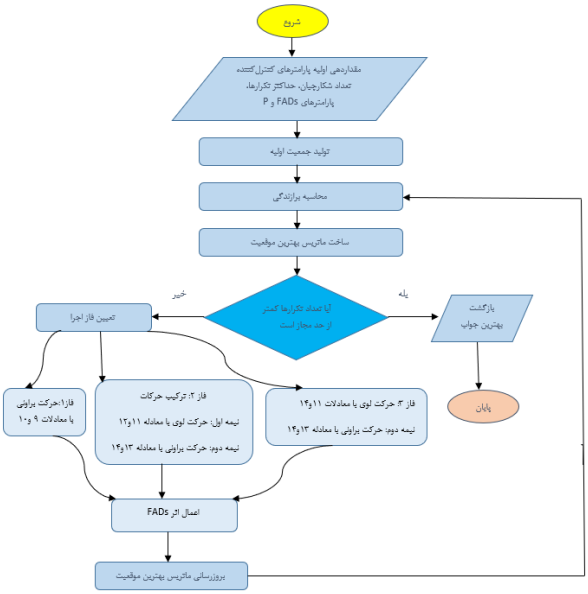
در فرمول ۱۳ R_L یک بردار تصادفی با توزیع لِوی است و در فرمول ۱۴ CF که یک پارامتر تطبیقی است که به مرور زمان کاهش می‌یابد. این فرآیند با استفاده از فرمول ۱۴ مدل‌سازی می‌شود:

$$CF = (1 - IterMaxIter)^{(2\times Iter/MaxIter)} \quad (۱۴)$$

در این الگوریتم، هم شکارچی و هم طعمه به‌عنوان عوامل جستجو عمل می‌کنند. موقعیت شکارچی (که نشان‌دهنده بهترین راه‌حل فعلی است) به‌طور مداوم با توجه به موقعیت طعمه و حرکت‌های تصادفی به‌روزرسانی می‌شود. شکارچیان از اطلاعات به‌دست‌آمده از موقعیت طعمه‌ها استفاده کرده و به سمت بهترین موقعیت حرکت می‌کنند. الگوریتم شکارچیان دریایی از مکانیزم‌هایی مانند تشکیل گرداب[۲۸] و دستگاه‌های تجمع ماهی[۲۹] برای جلوگیری از گیر افتادن در بهینه‌های محلی استفاده می‌کند. این مکانیزم باعث می‌شود که شکارچیان در صورت نیاز، حرکات تصادفی بزرگ‌تری انجام دهند و از نواحی بهینه محلی فرار کنند. فرمول به‌روزرسانی موقعیت با در نظر گرفتن دستگاه‌های تجمع ماهی با استفاده از فرمول ۱۵ صورت می‌گیرد:

$$\begin{aligned} &if\ r \le FADs \quad Prey_i + CF \times [X_{min} + R \times (X_{max_}X_{min}) \otimes U)] \\ &if\ r > FADs \quad Prey_i + [FADs \times (1_r) + r] \times Prey_{r1_}Prey_{r2}) \end{aligned} \quad (۱۵)$$

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025



شکل ۲: فلوچارت الگوریتم شکارچیان دریایی
Figure 2. Flowchart illustrating the Marine Predators Algorithm

۳-۲-۳- الگوریتم بهینه ساز زنبور عنکبوتی

الگوریتم بهینه‌سازی زنبور عنکبوتی که در سال ۲۰۲۳، توسط عبدالباسط و همکاران معرفی شد، یک الگوریتم فراابتکاری است که از رفتار شکار، لانه‌سازی و جفت‌گیری زنبورهای عنکبوتی ماده الهام گرفته شده است. در این الگوریتم، جمعیت اولیه شامل زنبورهای ماده است که هر کدام موقعیتی متناظر با یک راه‌حل بالقوه در فضای جستجو دارند. الگوریتم از چهار استراتژی اصلی برای به‌روزرسانی موقعیت زنبورها استفاده می‌کند.

- جستجو: زنبورهای عنکبوتی به دنبال عنکبوت‌ها در فضای جستجو می‌گردند. این فرآیند با استفاده از فرمول ۱۶ مدل‌سازی می‌شود.

$$SW^{t+1}{}_i = SW^t{}_i + \mu \times \left(SW^t{}_a - SW^t{}_b \right) \quad (۱۶)$$

- استراتژی تعقیب و فرار: زنبورهای عنکبوتی عنکبوت‌هایی را که از تار خود رها شده‌اند، تعقیب می‌کنند. این فرآیند با استفاده از فرمول ۱۷ مدل‌سازی می‌شود:

$$SW^{t+1}{}_i = SW^* + C \times |2 \times r_5 \times SW^t{}_a \times SW^t{}_i| \quad (۱۷)$$

- استراتژی لانه‌سازی: زنبورهای عنکبوتی عنکبوت‌های فلج شده را به لانه‌های از پیش آماده شده می‌برند. این فرآیند با استفاده از فرمول ۱۸ مدل‌سازی می‌شود.

$$SWt + 1i = SW * + cos(2\pi l) \times (SW * - SWti) \quad (۱۸)$$

- استراتژی جفت‌گیری : زنبورهای عنکبوتی با استفاده از عملگرهای ترکیبی، تخم‌گذاری می‌کنند. این فرآیند با استفاده از فرمول ۱۹ مدل‌سازی می‌شود.

$$SW^{t+1}{}_i = SW^* + cos(2\Pi l) \times (SW^* \times SW^t{}_i) \quad (۱۹)$$

در فرمول ۱۹، $SW^t{}_m$ بیانگر زنبور عنکبوتی نر و CR ۳۰ نرخ ترکیب است.

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

پس از محاسبه موقعیت جدید، زنبورهای عنکبوتی موقعیت خود را به‌روزرسانی می‌کنند. اگر موقعیت جدید بهتر از موقعیت قبلی باشد، زنبور عنکبوتی به موقعیت جدید حرکت می‌کند. الگوریتم بهینه ساز زنبور عنکبوتی با استفاده از پارامترهای μ_1 و C تعادل بین اکتشاف و بهره‌برداری را تنظیم می‌کند. مقادیر کوچک‌تر برای این پارامترها تمایل الگوریتم به بهره‌برداری را افزایش می‌دهد، در حالی که مقادیر بزرگ‌تر تمایل به اکتشاف را تقویت می‌کند.



شکل ۳: فلوچارت الگوریتم زنبور عنکبوتی

Figure 3. Flowchart of the Spider Bee Algorithm

۳-۲-۴- الگوریتم بهینه‌سازی شاهین هریس

الگوریتم بهینه‌سازی شاهین هریس یک الگوریتم فراابتکاری جدید است که در سال ۲۰۱۹ توسط علی اصغر حیدری و همکاران ارائه شد. این الگوریتم از رفتار شکار و تعقیب شاهین‌های هریس در طبیعت الهام گرفته است. شاهین‌های هریس از استراتژی‌های خاصی برای شکار طعمه‌های خود استفاده می‌کنند، هر شاهین موقعیتی در فضای جستجو دارد که نشان‌دهنده یک راه‌حل بالقوه برای مسئله است. الگوریتم بهینه‌سازی شاهین هریس از دو فاز اصلی برای به‌روزرسانی موقعیت شاهین‌ها استفاده می‌کند:

- فاز اکتشاف: شاهین‌ها به دنبال طعمه در فضای جستجو می‌گردند.
- فاز بهره‌برداری: شاهین‌ها با استفاده از استراتژی‌های مختلف، طعمه را شکار می‌کنند.

فاز اکتشاف: در این فاز، شاهین‌ها به دنبال طعمه در فضای جستجو می‌گردند. این فرآیند با استفاده از فرمول‌های ۲۰ و ۲۱ مدل‌سازی می‌شود.

$$X_{(t+1)} = X_{rand}(t) - r_1 \left| X_{rand}(t) - 2r_2 X_{(t)} \right| \qquad if q \geq 0.5 \qquad (۲۰)$$

$$X_{rabbit}(t) - X_m(t) - r_3(LB + r_4(UB - LB)| \qquad q < 0.5))if\ X_{(t)}5 \qquad (۲۱)$$

در فرمول ۲۰ $X_{(t+1)}$ بیانگر موقعیت جدید شاهین در تکرار t+1، $X_{rand}(t)$ بیانگر موقعیت تصادفی یک شاهین از جمعیت است. در فرمول ۲۱ $X_{rabbit}(t)$ نشان‌دهنده موقعیت طعمه،$X_m(t)$میانگین موقعیت شاهین‌ها، r_1, r_2, r_3, r_4, q : اعداد تصادفی در بازه [۰ .۱] است.LB و UB حد پایین و بالای متغیرها را نشان می‌دهد.

فاز بهره‌برداری: در این فاز، شاهین‌ها با استفاده از استراتژی‌های مختلف، طعمه را شکار می‌کنند. این استراتژی‌ها شامل محاصره نرم، محاصره سخت، حمله سریع و حمله پیشرونده هستند. این فرآیند با استفاده از فرمول‌های ۲۲ تا ۲۴ مدل‌سازی می‌شود.

- محاصره نرم[۳۱]:

$$X_{(t+1)} = \Delta X(t) - E \mid JX_{rabbit}(t) - X(t) \mid \qquad (۲۲)$$

- محاصره سخت[۳۲]:

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]

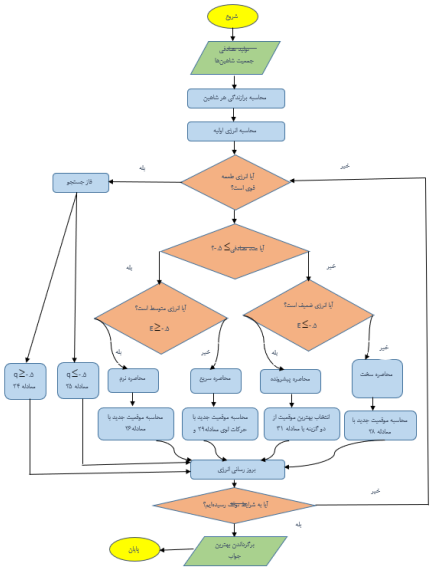
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

$$X(t+1) = X_{rabbit}(t) - E \mid \Delta X(t) \mid \quad (۲۳)$$

- حمله سریع[۳۳]:

$$Y = X_{rabbit}(t) - E \mid JX_{rabbit}(t) - X(t) \mid \quad (۲۴)$$

الگوریتم بهینه‌سازی شاهین هریس با استفاده از پارامتر انرژی فرار طعمه[۳۴] ، تعادل بین اکتشاف و بهره‌برداری را تنظیم می‌کند. مقادیر بزرگ‌تر برای پارامتر انرژی فرار طعمه تمایل الگوریتم به اکتشاف را افزایش می‌دهد، در حالی که مقادیر کوچک‌تر تمایل به بهره‌برداری را تقویت می‌کند. الگوریتم بهینه‌سازی شاهین هریس یک الگوریتم فراابتکاری جدید و الهام‌گرفته از رفتار شاهین‌های هریس در طبیعت است. این الگوریتم با ترکیب استراتژی‌های اکتشاف و بهره‌برداری، توانایی خوبی در یافتن راه‌حل‌های بهینه برای مسائل پیچیده دارد.



شکل ۴: فلوچارت الگوریتم شاهین هریس

Figure 4. Flowchart of the Harris Hawks Optimization algorithm

۳-۲-۵- الگوریتم بهینه‌سازی قاصدک

الگوریتم بهینه‌سازی قاصدک یک الگوریتم فراابتکاری جدید است که در سال ۲۰۲۲ توسط شیه جی ژائو و همکاران ارائه شد. این الگوریتم از رفتار پراکندگی دانه‌های قاصدک در طبیعت الهام گرفته است. دانه‌های قاصدک با استفاده از باد به مناطق دور دست منتقل می‌شوند و در آنجا رشد می‌کنند.

. الگوریتم بهینه‌سازی قاصدک از سه فاز اصلی برای به‌روزرسانی موقعیت دانه‌ها استفاده می‌کند:

- فاز صعود[۳۵] : دانه‌ها به سمت بالا حرکت می‌کنند و در فضای جستجو پراکنده می‌شوند.
- فاز نزول[۳۶] : دانه‌ها به آرامی به سمت پایین حرکت می‌کنند.
- فاز فرود[۳۷]: دانه‌ها در موقعیت‌های تصادفی فرود می‌آیند و رشد می‌کنند.

فاز صعود در این فاز، دانه‌ها به سمت بالا حرکت می‌کنند و در فضای جستجو پراکنده می‌شوند. این فرآیند با استفاده از فرمول‌های زیر مدل‌سازی می‌شود:

فاز نزول: در این فاز، دانه‌ها به آرامی به سمت پایین حرکت می‌کنند. این فرآیند با استفاده از فرمول ۲۵ مدل‌سازی می‌شود.

$$X_{(t+1)} = X_{(t)} - \alpha \times \beta t \times (Xmean_{(t)} - \alpha \times \beta t \times X_{(t)}) \quad (۲۵)$$

در فرمول ۲۵ β بیانگر حرکت براونی و Xmean بیانگر میانگین موقعیت دانه‌ها در تکرار t است.

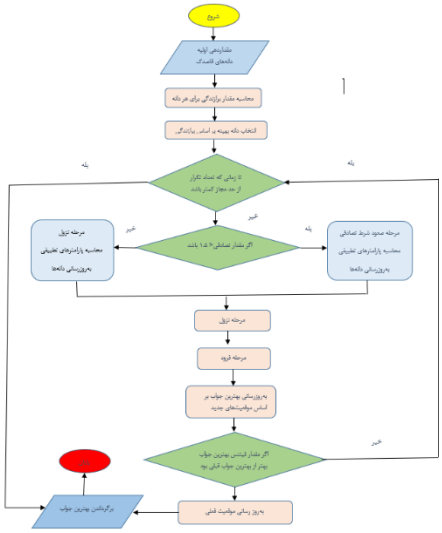
[Downloaded from jeps.dezful.iau.ir on 2026-11-13]

[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

فاز فرود: در این فاز، دانه‌ها در موقعیت‌های تصادفی فرود می‌آیند و رشد می‌کنند. این فرآیند با استفاده از فرمول ۲۶ مدل‌سازی می‌شود.

$$X_{(t+1)} = X_{elite} + levy(\lambda) \times \alpha \times (X_{elite} - X_{(t)} \times \delta) \quad (۲۶)$$

در فرمول ۲۶ Xelite بیانگر موقعیت بهینه دانه در تکرار t، -levy(λ) تابع پرواز لوی[۳۸] برای مدل‌سازی حرکات تصادفی و delta تابع افزایشی خطی را نشان می‌دهد. الگوریتم بهینه‌سازی قاصدک با استفاده از پارامترهای مختلف مانند α[۳۹] و k، تعادل بین جستجوی مناطق جدید و جستجوی دقیق‌تر در مناطق شناخته شده را تنظیم می‌کند. مقادیر بزرگ‌تر برای α تمایل الگوریتم به اکتشاف را افزایش می‌دهد، در حالی که مقادیر کوچک‌تر تمایل به بهره‌برداری را تقویت می‌کند. این الگوریتم با ترکیب استراتژی‌های اکتشاف و بهره‌برداری، توانایی خوبی در یافتن راه‌حل‌های بهینه برای مسائل پیچیده دارد.



شکل ۵: فلوچارت الگوریتم قاصدک

Figure 5. Flowchart of the Dandelion Optimizer

۳-۲-۶- الگوریتم کپک مخاطی با تقابل تطبیقی

الگوریتم کپک مخاطی با تقابل تطبیقی یک الگوریتم فرااابتکاری جدید است که در سال ۲۰۲۱ توسط مانوج کومار نایک و همکاران ارائه شد. این الگوریتم از رفتار کپک لجن در طبیعت الهام گرفته است. این الگوریتم با شبیه‌سازی این فرآیند، به دنبال یافتن راه‌حل‌های بهینه برای مسائل پیچیده است. الگوریتم کپک مخاطی با تقابل تطبیقی از دو فاز اصلی برای به‌روزرسانی موقعیت کپک‌ها استفاده می‌کند:

فاز اکتشاف :در این فاز، کپک‌ها به دنبال غذا در فضای جستجو می‌گردند. این فرآیند با استفاده از فرمول ۲۷ مدل‌سازی می‌شود.

$$X_{LB}(t) + V_b(W \cdot X_{LB}(t) - X_B(t)) \quad (۲۷)$$
$$r_1 \geq \delta \; and \; r_2 < p_i$$

در فرمول۲۷ X(t+1) بیانگر موقعیت جدید کپک در تکرار t+1، X_{LB} (t): بهترین موقعیت محلی در تکرار t، X_B (t): موقعیت تصادفی یک کپک از جمعیت. ، Vb و Vc: سرعت‌های تصادفی،r_1 و r_2 : اعداد تصادفی در بازه [۰. ۱]. ، δ : پارامتر کنترل کننده اکتشاف و p_i : آستانه تصمیم‌گیری برای به‌روزرسانی موقعیت است.

فاز بهره‌برداری: در این فاز، کپک‌ها با استفاده از استراتژی‌های مختلف، غذا را پیدا می‌کنند. این فرآیند با استفاده از یادگیری معکوس و استراتژی تطبیقی مدل‌سازی می‌شود:

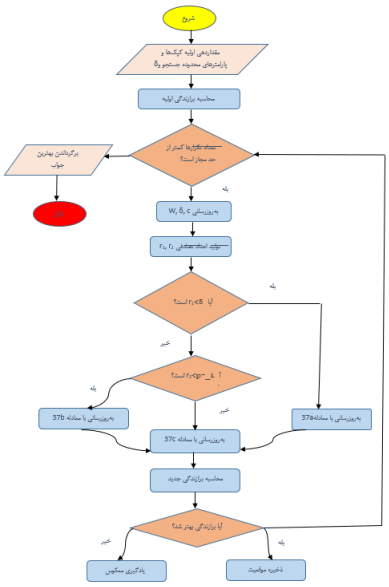
- یادگیری معکوس[۴۰] : در این مرحله، موقعیت معکوس کپک‌ها محاسبه می‌شود و با موقعیت فعلی مقایسه می‌شود تا بهترین موقعیت انتخاب شود. این فرایند با فرمول ۲۸ مدل‌سازی می‌شود:

$$XO_i{}^j(t) = min(X_i(t)) + max(X_i(t)) - Xn_i{}^j(t) \quad (۲۸)$$

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]

[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

الگوریتم کپک مخاطی با تقابل تطبیقی با استفاده از پارامترهای مختلف مانند δ[41] و p_i ، تعادل بین اکتشاف و بهره‌برداری را تنظیم می‌کند. مقادیر بزرگ‌تر برای δ تمایل الگوریتم به اکتشاف را افزایش می‌دهد، در حالی که مقادیر کوچک‌تر تمایل به بهره‌برداری را تقویت می‌کند.



شکل ۶: فلوچارت الگوریتم کپک مخاطی با تقابل تطبیقی

Figure 6. Flowchart of the Adaptive Opposition-Based Slime Mould Algorithm

۳-۲-۷-الگوریتم بهینه‌ساز تعادل

الگوریتم بهینه‌ساز تعادل یک الگوریتم فراابتکاری جدید است که در سال ۲۰۱۹ توسط فرامرزی و همکاران معرفی شده است. این الگوریتم از مدل‌های تعادل فیزیکی و دینامیکی الهام گرفته شده است. الگوریتم بهینه‌ساز تعادل با شبیه‌سازی این فرآیند، به دنبال یافتن راه‌حل‌های بهینه برای مسائل پیچیده است. الگوریتم بهینه‌ساز تعادل از دو فاز اصلی برای به‌روزرسانی موقعیت ذرات استفاده می‌کند:

- فاز اکتشاف: ذرات به دنبال مناطق جدید در فضای جستجو می‌گردند.
- فاز بهره‌برداری: ذرات با استفاده از استراتژی‌های مختلف، جستجوی دقیق‌تری در مناطق شناخته شده انجام می‌دهند.

فاز اکتشاف: در این فاز، ذرات به دنبال مناطق جدید در فضای جستجو می‌گردند. این فرآیند با استفاده از فرمول ۲۹ مدل‌سازی می‌شود:

$$X(t+1) = X_{eq} + \left(X(t) - X_{eq}\right) \times F + G/\lambda V \times (1-F) \quad (۲۹)$$

تابع نمایی که تعادل بین اکتشاف و بهره‌برداری را کنترل می‌کند. G[42]: نرخ تولید که به بهبود جستجو کمک می‌کند،λ : نرخ چرخش[43] که سرعت حرکت ذرات را کنترل می‌کند،V[44] : حجم کنترل که به عنوان یک پارامتر ثابت در نظر گرفته می‌شود.
فاز بهره‌برداری: در این فاز، ذرات با استفاده از استراتژی‌های مختلف، جستجوی دقیق‌تری در مناطق شناخته شده انجام می‌دهند.

۴. روش تحقیق

این پژوهش، الگوریتم جستجوی طلا را با شش الگوریتم فراابتکاری دیگر (شاهین هریس، قاصدک، کپک مخاطی با تقابل تطبیقی، بهینه‌ساز تعادل، شکارچیان دریایی و زنبور عنکبوتی) مقایسه کرده است. الگوریتم‌ها روی توابع محک F1 تا F10 در ابعاد ۱۰،

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

۳۰ و ۵۰ و با ۱۰۰ تکرار اجرا شده‌اند. عملکرد آنها بر اساس بهترین مقدار تابع هدف، انحراف معیار و زمان اجرا ارزیابی شده است. شبیه‌سازی‌ها در MATLAB R2022a روی سیستم مجهز به پردازنده Intel Core i7-11700F ، ۱۶ گیگابایت رم و SSD NVMe انجام شده‌اند. هر تابع محک ۳۰ بار به طور مستقل اجرا شده و نتایج با میانگین، انحراف معیار، رتبه‌بندی و تحلیل آماری گزارش شده است.

۴-۱- تحلیل نتایج اولیه

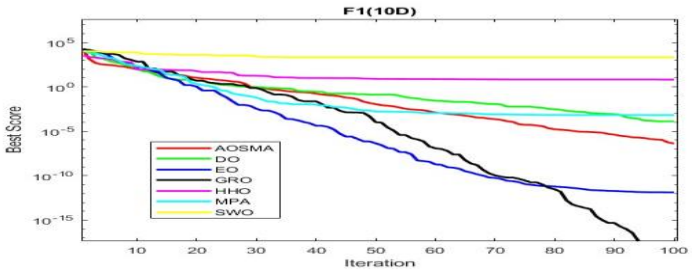
در این بخش، نتایج اولیه اجرای الگوریتم‌ها روی توابع تست ارائه شده است. هر الگوریتم در برخی توابع بهتر عمل کرده و تفاوت در دقت و سرعت دیده می‌شود. برای مقایسه عادلانه، تعداد تکرار و اندازه جمعیت برای همه الگوریتم‌ها یکسان و سایر پارامترها مطابق مقالات اصلی تنظیم شده‌اند. در الگوریتم جستجوی طلا، پارامترهای A1 ، A2و GP به ترتیب ۲,۰، ۱,۵ و ۰,۵ قرار داده شده‌اند. نتایج نهایی در بخش ۵ بحث خواهد شد.

جدول ۲: تنظیمات پارامترهای الگوریتم‌های استفاده‌شده در این پژوهش

Table 2. Parameter settings of the algorithms employed in this study

الگوریتم	اندازه جمعیت	تعداد تکرار	پارامترهای ویژه	منبع تنظیم
(Gold Search) الگوریتم جستجوی طلا	30	100	GP ،A2 ،A1	پارامترهای کنترل مهاجرت و استخراج؛ GP برای تنظیم وزن
Harris Hawks Optimization (HHO) الگوریتم بهینه‌سازی شاهین هریس	30	100	LF ،J ،E	Eانرژی فرار؛ J قدرت پرش؛ LF پرواز لوی
Equilibrium Optimizer (EO) بهینه‌ساز تعادل	30	100	V ،λ ،G ،F	پارامترهای دینامیکی الگوریتم بهینه‌ساز تعادل
Marine Predators Algorithm (MPA) الگوریتم شکارچیان دریایی	30	100	پیش‌فرض	سه فازRandom Walk. ،Lévy ، Brownian
Dandelion Optimizer (DO) بهینه‌سازی قاصدک	30	100	پیش‌فرض	از پارامترهای استاندارد مقاله مرجع
Adaptive Opposition Slime Mould Algorithm (AOSMA) الگوریتم تطبیقی کپک لجن با تقابل	30	100	پیش‌فرض	پارامترهای سازگار و کنترلی کپک مخاطی با تقابل تطبیقی
Spider Wasp Optimizer (SWO) الگوریتم بهینه‌سازی زنبور عنکبوتی	30	100	نرخ ارتباطی و جستجوی محلی	مقادیر الگوریتم زنبور عنکبوتی

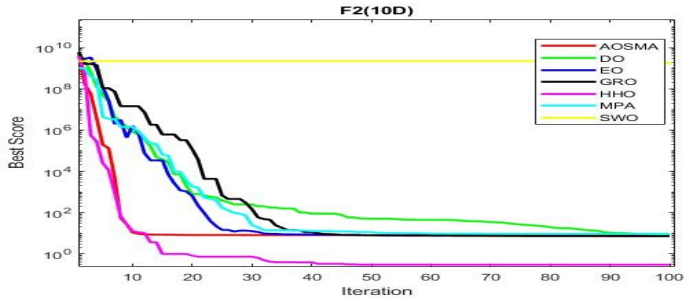
شکل ۷، منحنی همگرایی الگوریتم‌های مختلف برای حل تابع F1 در بعد ۱۰ و طی ۱۰۰ تکرار را نشان می‌دهد. نتایج به‌وضوح بیانگر آن هستند که الگوریتم جستجوی طلا در مقایسه با سایر الگوریتم‌ها، بالاترین سرعت همگرایی را داشته و با دقت بسیار بالا به پاسخ بهینه این تابع رسیده است. در مقابل، الگوریتم‌هایی مانند زنبور عنکبوتی ، شاهین حریص و بهینه‌ساز تعادل نتوانسته‌اند همگرایی مناسبی به سمت مقدار بهینه داشته باشند و عملاً از دستیابی به پاسخ مطلوب بازمانده‌اند



شکل ۷: منحنی همگرایی توسط تابع F1 در بعد۱۰

Figure 7. Convergence curves of the algorithms on function F1 with 10 dimensions

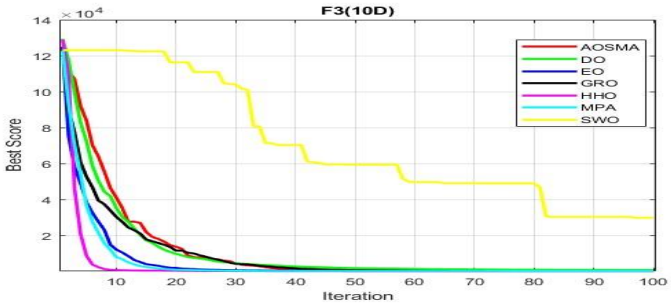
[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025



شکل ۸: منحنی همگرایی توسط تابع F2 در بعد۱۰

Figure 8. Convergence curves of the algorithms on function F2 with 10 dimensions

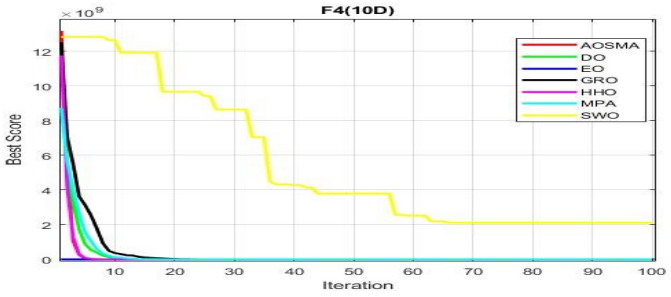
شکل ۸، منحنی همگرایی الگوریتم‌ها را برای تابع F2 در فضای ۱۰ بعدی و ۱۰۰ تکرار نشان می‌دهد. الگوریتم شاهین حریص در تکرارهای اولیه سرعت بالایی داشته و به سطح قابل قبولی از پاسخ بهینه رسیده است. در مقابل، الگوریتم جستجوی طلا عملکرد ضعیفی داشته و با نوسانات زیاد نتوانسته همگرا شود. الگوریتم زنبور عنکبوتی نیز کاهش معناداری در مقدار تابع نداشته و بدون پیشرفت باقی مانده است.



شکل ۹: منحنی همگرایی توسط تابع F3 در بعد۱۰

Figure 9. Convergence curves of the algorithms on function F3 with 10 dimensions

شکل ۹، منحنی همگرایی الگوریتم‌ها را برای تابع F3 در فضای ۱۰ بعدی و ۱۰۰ تکرار نشان می‌دهد. اکثر الگوریتم‌ها به خوبی به پاسخ بهینه همگرا شده‌اند. الگوریتم شاهین حریص با سرعت بیشتری نسبت به سایرین همگرا شده و در همان تکرارهای ابتدایی به مقدار مطلوب رسیده است. در مقابل، الگوریتم زنبور عنکبوتی عملکرد موفقی نداشته و تا پایان تکرارها به همگرایی مناسب نرسیده است؛ منحنی آن کاهش پله‌ای و ناکامل را نشان می‌دهد.

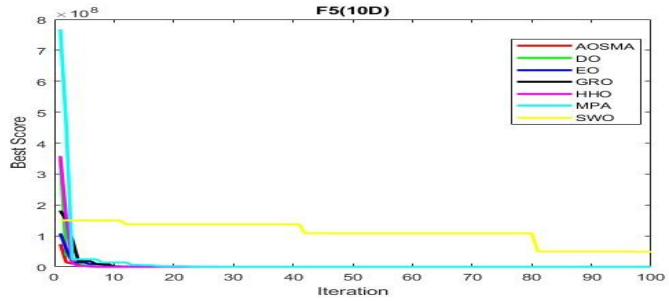


شکل ۱۰: منحنی همگرایی توسط تابع F4 در بعد۱۰

Figure 10. Convergence curves of the algorithms on function F4 with 10 dimensions

شکل ۱۰، منحنی همگرایی الگوریتم‌ها را برای تابع F4 در فضای ۱۰ بعدی و ۱۰۰ تکرار نشان می‌دهد. اکثر الگوریتم‌ها عملکرد مشابه و همگرایی سریعی داشته‌اند، اما الگوریتم کپک مخاطی با تقابل تطبیقی و شاهین حریص همگرایی دقیق‌تر و سریع‌تری داشتند. در مقابل، الگوریتم زنبور عنکبوتی روندی ناپایدار و پله‌ای نشان داده و از نظر سرعت و دقت ضعیف‌تر عمل کرده است.

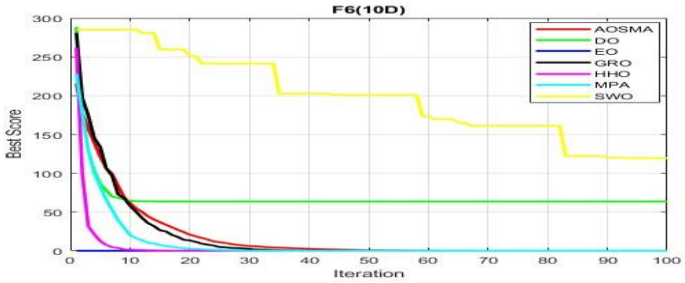
[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025



شکل ۱۱: منحنی همگرایی توسط تابع F5 در بعد ۱۰

Figure 11. Convergence curves of the algorithms on function F5 with 10 dimensions

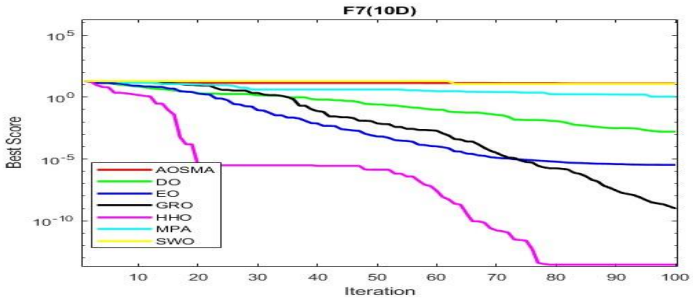
شکل ۱۱، منحنی همگرایی الگوریتم‌ها برای تابع F5 در فضای ۱۰ بعدی و ۱۰۰ تکرار را نشان می‌دهد. بیشتر الگوریتم‌ها رفتار مشابهی داشته و در تکرارهای ابتدایی به سرعت به مقادیر پایین همگرا شده‌اند. الگوریتم کپک مخاطی با تقابل تطبیقی عملکرد بهتری داشته و همگرایی دقیق‌تر و پایدارتری ارائه کرده است.



شکل ۱۲: منحنی همگرایی توسط تابع F6 در بعد ۱۰

Figure 12. Convergence curves of the algorithms on function F6 with 10 dimensions

شکل ۱۲، منحنی همگرایی الگوریتم‌ها را برای تابع F6 در فضای ۱۰ بعدی و ۱۰۰ تکرار نشان می‌دهد. بیشتر الگوریتم‌ها سرعت همگرایی مناسبی داشته‌اند. الگوریتم شاهین حریص بیشترین سرعت را داشته و در تکرارهای اولیه با شیب تند به سمت مقدار بهینه حرکت کرده است. در مقابل، الگوریتم‌های قاصدک و به‌ویژه زنبور عنکبوتی عملکرد ضعیف‌تری داشته‌اند؛ زنبور عنکبوتی با کاهش پله‌ای و مقدار هدف بالاتر، همگرایی کند و ناکامل نشان داده است.

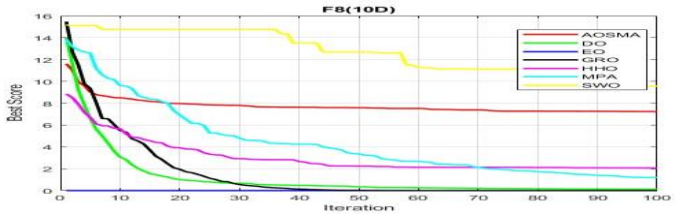


شکل ۱۳: منحنی همگرایی توسط تابع F7 در بعد ۱۰

Figure 13. Convergence curves of the algorithms on function F7 with 10 dimensions

شکل ۱۳، منحنی همگرایی الگوریتم‌ها را برای تابع F7 در فضای ۱۰ بعدی و ۱۰۰ تکرار نشان می‌دهد. الگوریتم شاهین حریص با سرعت بسیار بالا بهترین عملکرد را داشته است. الگوریتم جستجوی طلا در رتبه دوم، با سرعت و دقت مناسب به پاسخ بهینه نزدیک شده است. در مقابل، الگوریتم‌های کپک مخاطی با تقابل تطبیقی، زنبور عنکبوتی و بهینه‌ساز تعادل همگرایی مناسبی نداشته‌اند؛ منحنی‌هایشان تقریباً بدون تغییر یا با کاهش بسیار کند بوده است.

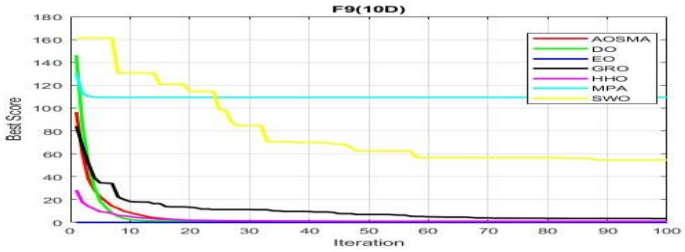
[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025



شکل ۱۴: منحنی همگرایی توسط تابع F8 در بعد ۱۰

Figure 14. Convergence curves of the algorithms on function F8 with 10 dimensions

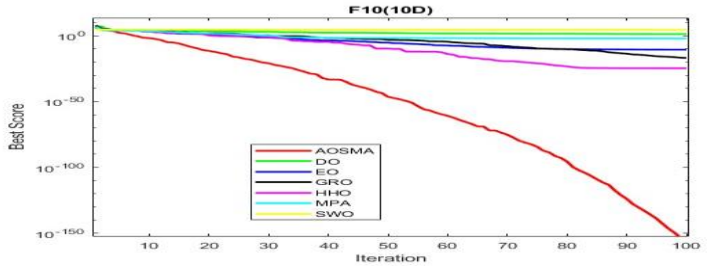
شکل ۱۴، منحنی همگرایی الگوریتم‌ها برای تابع F8 در فضای ۱۰ بعدی و ۱۰۰ تکرار را نشان می‌دهد. الگوریتم جستجوی طلا در تکرارهای اولیه با سرعت بالا مقدار تابع را کاهش داده و به سطح قابل قبولی رسیده است. در رتبه دوم، الگوریتم بهینه‌ساز تعادل نیز سرعت همگرایی بالایی داشته است. در مقابل، الگوریتم‌های شکارچیان دریایی، شاهین حریص و کپک مخاطی با تقابل تطبیقی همگرایی کندتری داشته‌اند و زنبور عنکبوتی با کاهش پله‌ای و محدود، عملکرد ضعیف‌تری نشان داده است.



شکل ۱۵: منحنی همگرایی توسط تابع F9 در بعد ۱۰

Figure 15. Convergence curves of the algorithms on function F9 with 10 dimensions

شکل ۱۵، منحنی همگرایی الگوریتم‌ها را برای حل تابع F9 در فضای ۱۰ بعدی و طی ۱۰۰ تکرار نمایش می‌دهد. نتایج نشان می‌دهد که الگوریتم قاصدک در تکرارهای ابتدایی با سرعت بالا مقدار تابع هدف را کاهش داده و به سطح قابل قبولی از پاسخ بهینه دست یافته است. در رتبه دوم، الگوریتم شاهین حریص قرار دارد که دارای سرعت همگرایی بالا بوده و توانسته به سرعت به مقادیر پایین تابع هدف نزدیک شود.



شکل ۱۶: منحنی همگرایی توسط تابع F10 در بعد ۱۰

Figure 16. Convergence curves of the algorithms on function F10 with 10 dimensions

الگوریتم کپک مخاطی با تقابل تطبیقی بیشترین سرعت همگرایی را دارد و در توابع پیچیده مانند F10 و F5 بهترین عملکرد را نشان می‌دهد. شاهین حریص در اکثر توابع (F1, F2, F3, F6, F7) همگرایی سریع و رتبه دوم را دارد. جستجوی طلا در برخی توابع (F2, F8) عملکرد اولیه خوبی دارد ولی در برخی دیگر (F1, F7) به بهینه نمی‌رسد. قاصدک و بهینه‌ساز تعادل عملکرد متوسط، و زنبور عنکبوتی و شکارچیان دریایی عملکرد ضعیفی دارند. نتیجه: کپک مخاطی بهترین از نظر سرعت و دقت، شاهین حریص پایدار، و جستجوی طلا وابسته به مسئله است. عملکرد جستجوی طلا با توابع F1 تا F10 (۱۰ بعدی، ۱۰۰ تکرار) و معیارهای بهترین پاسخ، انحراف معیار و زمان اجرا ارزیابی شده است.

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

جدول ۳: مقایسه عملکرد الگوریتم‌ها با توابع تک وجهی و چندوجهی در بعد۱۰ و تکرار ۱۰۰

Table 3. Performance comparison of the algorithms on unimodal and multimodal test functions with 10 dimensions and 100 iterations

F		AOSMA	DO	EO	GRO	HHO	MPA	SWO
F1	Best Score	۴.۲۶۰۵ e-۷	۱.۳۱۲۸ e-۰۴	۴.۰۴۳۴ e-۱۱	۲.۱۲۳۶ e-۲۰	۶.۵۳۰۶ e+۰۰	۶.۶۸۳۰ e-۰۴	۲.۱۲۷۷ e+۰۳
	SD	۱.۴۴۸۰ e-۰۵	۳.۱۶۵۶ e-۰۵	۷.۱۶۰۳ e-۱۱	۲.۱۷۷۹ e-۱۸	۷.۴۸۴۲ e+۰۰	۳.۶۸۲۰ e-۰۴	۹.۱۰۸۶ e+۰۲
	Run Time(s(	۰.۱۷۳۰	۰.۰۶۲۶	۰.۱۵۴۵	۰.۴۰۶۸	۰.۵۶۵۱	۰.۳۳۰۵	۰.۰۸۷۷
F10	Best Score	۲.۸۹۶۹ e-۱۵۶	۵.۷۰۶۱ e+۰۰	۱.۱۷۴۴ e-۰۹	۱.۳۴۱۷ e-۱۹	۳.۱۱۰۹ e-۲۱	۸.۵۸۶۹ e-۰۲	۴.۱۲۶۴ e+۰۴
	SD	۱.۷۰۱۱ e-۰۱۵۴	۱.۰۸۴۳ e+۰۱	۱.۲۲۴۰ e-۱۹	۸.۱۱۵۹ e-۱۹	۵.۶۰۸۹ e-۰۲۱	۱.۵۶۶۰ e-۰۲	۳.۶۶۱۷ e+۰۷
	Run Time(s)	۰.۳۱۱۷	۰.۰۳۶۸	۰.۰۹۱۵	۰.۴۱۷۰	۰.۰۳۰۰	۰.۲۰۹۱	۰.۱۶۴۴

در این تحقیق، عملکرد هفت الگوریتم بهینه‌سازی روی ده تابع محک استاندارد با بعد ۱۰ ارزیابی شد. الگوریتم شاهین هریس در توابعF2 ، F3، F4، F6، F7و F9 برتر بود، که نشان‌دهنده دقت و پایداری بالای آن است. الگوریتم کپک مخاطی با تقابل تطبیقی در توابع F5 و F10 بهترین عملکرد را داشت. زنبور عنکبوتی کمترین زمان اجرا را در بیشتر توابع داشت، اما دقت و پایداری ضعیف‌تری نسبت به دو الگوریتم برتر نشان داد. جستجوی طلا در توابع F1 و F8 رتبه اول و درF2 ، F3، F4، F7 عملکرد مناسب داشت، اما در برخی موارد همگرا نشد. قاصدک و بهینه‌ساز تعادل عملکرد قابل قبول اما پایین‌تر از الگوریتم‌های برتر داشتند. نتیجه: الگوریتم‌های شاهین هریس و کپک مخاطی با تقابل تطبیقی برای مسائل پیچیده چندبعدی مناسب‌تر هستند و انتخاب الگوریتم به ویژگی‌های تابع هدف بستگی دارد.

جدول ۴: مقایسه عملکرد الگوریتم‌ها با توابع تک وجهی و چندوجهی در بعد۳۰ و تکرار ۱۰۰

Table 4. Performance comparison of the algorithms on unimodal and multimodal test functions with 30 dimensions and 100 iterations

F		AOSMA	DO	EO	GRO	HHO	MPA	SWO
F1	Best Score	۱.۵۴۲۲ e+۰۰	۲.۹۶۲۶ e+۰۰	۵.۹۳۴۱ e-۰۵	۸.۹۲۴۷ e-۱۰	۱.۳۱۱۱ e+۲	۸.۴۶۵۰ e-۰۲	۱.۰۸۲۰ e+۰۴
	SD	۴ .۲۸۶۷ e-۱	۵ .۰۱۱۲ e+۰۰	۳.۴۱۷۹ e-۰۵	۴.۱۴۹۴ e-۹	۳.۳۰۱۵ e+۱	۱.۱۹۷۶ e-۰۱	۱.۹۳۴۸ e+۰۳
	Run Time(s)	۰.۵۷۸۶	۰.۰۸۶۸	۰.۷۲۷۶	۰.۴۶۳۳	۰.۶۹۵۵	۰.۴۷۹۹	۰.۰۸۳۷
F10	Best Score	۳.۱۱۶۵ e-۱۵۲	۴.۷۵۹۶ e+۰۲	۳.۰۵۱۲ e-۰۴	۷.۹۹۳۴ e-۰۸	۱.۰۴۵۰ e-۲۳	۲.۰۹۷۲ e-۰۱	۹.۶۴۲۶ e+۰۴
	SD	۱.۷۵۶۹ e-۱۵۱	۲.۱۹۴۹ e+۰۲	۴.۱۴۹۱ e-۰۴	۵.۴۸۵۳ e-۰۸	۲.۳۶۵۲ e-۱۶	۳.۵۴۵۳ e-۰۱	۸.۲۷۷۴ e+۰۶
	Run Time(s)	۰.۶۳۵۳	۰.۰۹۳۵	۰.۱۹۰۴	۰.۴۲۶۱	۰.۰۳۴۷	۰.۴۲۵۱	۰.۲۷۳۷

جدول ۴ عملکرد هفت الگوریتم را روی توابع F1 تا F10 بر اساس بهترین پاسخ، انحراف معیار و زمان اجرا مقایسه می‌کند.الگوریتم کپک مخاطی با تقابل تطبیقی در اکثر توابع بهترین پاسخ و در F1 تا F4 کمترین انحراف معیار را دارد. شاهین هریس در F3 و F4 بهترین مقدار تابع و در F10 سریع‌ترین زمان اجرا را ثبت کرده است. زنبور عنکبوتی کمترین زمان اجرا را در بیشتر توابع دارد اما دقت و پایداری پایین‌تری دارد. جستجوی طلا عملکرد قابل قبول و در برخی توابع رتبه دوم را کسب کرده است. شکارچیان دریایی، بهینه‌ساز تعادل و قاصدک عملکرد متوسطی دارند. نتیجه: الگوریتم‌های کپک مخاطی با تقابل تطبیقی و شاهین هریس به عنوان دو الگوریتم برتر توصیه می‌شوند.

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

جدول ۵: مقایسه عملکرد الگوریتم‌ها با توابع تک وجهی و چندوجهی در بعد۵۰ و تکرار ۱۰۰

Table 5. Performance comparison of the algorithms on unimodal and multimodal test functions with 50 dimensions and 100 iterations

		AOSMA	DO	EO	GRO	HHO	MPA	SWO
F1	Best Score	۱.۷۳۶۰ e+۰۱	۱.۲۷۴۱ e+۰۲	۳.۲۲۱۵ e-۰۳	۳.۱۰۷۱ e-۰۶	۲.۰۴۰۴ e+۲	۳.۱۵۴۲ e-۰۱	۲.۳۵۳۷ e+۰۴
	SD	۷.۲۴۹۸ e+۰۰	۱.۵۲۹۹ e+۰۲	۳.۸۳۵۲ e-۰۳	۵.۷۰۵۷ e-۰۶	۴.۶۲۱۸ e+۰۱	۳.۲۸۰۳ e-۰۲	۳.۲۸۲۳ e+۰۳
	Run Time(s)	۰.۲۶۵۹	۰.۰۶۶۰	۰.۲۳۱۷	۰.۴۷۳۸	۰.۶۱۵۹	۰.۵۷۴۲	۰.۰۹۸۱
F10	Best Score	۳.۷۸۶۹ e-۱۴۹	۱.۲۲۲۵ e+۰۳	۶.۵۵۰۹ e-۰۳	۲.۸۱۳۶ e-۰۷	۱.۴۴۵۶ e-۲۶	۱.۳۵۳۳ e+۰۰	۳.۵۹۴۷ e+۰۶
	SD	۲.۸۵۵۲ e-۱۴۵	۹.۱۷۸۳ e+۰۲	۱.۷۱۵۷ e-۰۳	۱.۱۶۸۷ e-۰۵	۲.۹۳۰۰ e-۲۰	۶.۶۸۴۷ e-۰۱	۹.۸۴۶۶ e+۰۶
	Run Time(s)	۰.۳۶۶۸	۰.۰۶۶۷	۰.۲۱۲۸	۰.۴۶۶۱	۰.۰۴۹۳	۰.۴۰۰۸	۰.۳۰۹۷

جدول ۵ عملکرد هفت الگوریتم را در سه معیار بهترین مقدار، انحراف معیار و زمان اجرا برای توابع تک‌وجهی و چندوجهی نشان می‌دهد. جستجوی طلا در F1 و F8 بهترین دقت و کمترین انحراف معیار را دارد. شاهین هریس درF2 ، F3، F4وF7 بسیار پایدار و دقیق است. کپک مخاطی با تقابل تطبیقی در توابع پیچیده‌تر برتر است. از نظر زمان اجرا، زنبور عنکبوتی در اکثر توابع سریع‌ترین است؛ قاصدک در F1 و F3 و شاهین هریس در F10 نیز سریع عمل کرده‌اند. هیچ الگوریتمی در همه معیارها برتر نیست. جستجوی طلا دقت بالا، زنبور عنکبوتی سرعت بالا، و کپک مخاطی و شاهین هریس در مسائل چندوجهی دقت و پایداری مناسبی دارند.

۴-۲- نتایج تجربی و تحلیل عملکرد

در این بخش، به تحلیل جامع و مقایسه عملکرد الگوریتم جستجوی طلا با شش الگوریتم فراابتکاری دیگر شامل شاهین هریس، کپک مخاطی با تقابل تطبیقی، زنبور عنکبوتی، قاصدک، شکارچیان دریایی و بهینه‌ساز تعادل میپردازیم. این مقایسه بر اساس نتایج حاصل از ۱۰۰ بار اجرای مستقل هر الگوریتم بر روی ۱۰ تابع تست استاندارد F1 تا F10 در ابعاد ۱۰، ۳۰ و ۵۰ انجام شده است. معیارهای ارزیابی شامل میانگین بهترین جواب یافت شده، انحراف معیار نتایج و میانگین زمان اجرای الگوریتم ها بوده است.

۴-۲-۱-تحلیل بر اساس میانگین بهترین جواب

جداول ۲، ۳ و ۴ میانگین بهترین جواب الگوریتم‌ها را در ابعاد ۱۰، ۳۰ و ۵۰ نشان می‌دهند. الگوریتم‌های شاهین هریس و کپک مخاطی با تقابل تطبیقی در اکثر توابع و همه ابعاد رتبه‌های اول و دوم را دارند. جستجوی طلا در F1 و به‌ویژه F8 عملکرد عالی داشته و در سایر توابع در رتبه‌های میانی قرار گرفته است. شکارچیان دریایی و قاصدک در برخی توابع خوب عمل کرده‌اند، اما زنبور عنکبوتی و بهینه‌ساز تعادل کیفیت پایین‌تری دارند. با افزایش ابعاد، دقت اکثر الگوریتم‌ها کاهش می‌یابد، اما دو الگوریتم برتر برتری نسبی خود را حفظ می‌کنند.

۴-۲-۲- عوامل مؤثر بر عملکرد الگوریتم‌ها

عملکرد الگوریتم‌های فراابتکاری تحت تأثیر عواملی مانند توان اکتشاف، توان بهره‌برداری، تنوع جمعیت، حساسیت به مقداردهی اولیه و هزینه محاسباتی است. در این پژوهش، الگوریتم‌های کپک مخاطی با تقابل تطبیقی و شاهین هریس ترکیب مؤثری از اکتشاف و بهره‌برداری ارائه داده‌اند که منجر به پایداری و دقت بالاتر در توابع تک‌وجهی و چندوجهی شده است. در مقابل، الگوریتم‌های با بهره‌برداری ضعیف یا عملیات سنگین، در برخی توابع چندوجهی در بهینه‌های محلی گرفتار شده یا انحراف معیار بالاتری نشان داده‌اند. این یافته‌ها در ادامه با جداول و نمودارهای کمی تحلیل خواهند شد.

۴-۲-۳-تحلیل بر اساس انحراف معیار

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]

[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

جداول ۳، ۴و ۵ انحراف معیار نتایج به دست آمده توسط هر الگوریتم را نشان میدهد. انحراف معیار پایینتر نشاندهنده پایداری بیشتر الگوریتم و تکرارپذیری نتایج آن در اجراهای مختلف است. مشابه نتایج میانگین بهترین جواب، الگوریتم‌های شاهین هریس و کپک مخاطی با تقابل تطبیقی به طور مداوم انحراف معیار کمتری را در اکثر توابع تست و ابعاد ثبت کردهاند که نشان‌دهنده پایداری بالای آنهاست. الگوریتم جستجوی طلا نیز در توابعی که دقت خوبی داشته مانند F1 و F8، پایداری مناسبی نیز از خود نشان داده است. الگوریتمهای رنبور عنکبوتی و بهینه‌ساز تعادل اغلب بالاترین مقادیر انحراف معیار را دارند که نشاندهنده نوسان و عدم پایداری بیشتر در نتایج آنها در تکرارهای مختلف است

۴-۲-۴-تحلیل بر اساس میانگین زمان اجرا

زمان اجرا یک معیار مهم برای ارزیابی کارایی محاسباتی الگوریتم ها، به خصوص در مسائل با ابعاد بالا یا کاربردهای زمانبند است. بر اساس این جدول، الگوریتم زنبور عنکبوتی با کمترین میانگین زمان اجرا، سریع‌ترین الگوریتم در بین الگوریتم‌های مقایسه شده است. الگوریتم‌های کپک مخاطی با تقابل تطبیقی و بهینه‌ساز تعادل نیز از نظر سرعت عملکرد خوبی دارند و در رتبههای بعدی قرار می‌گیرند. الگوریتم جستجوی طلا از نظر زمان اجرا در جایگاه میانی قرار گرفته و سرعت آن قابل قبول است. الگوریتم‌های بهینه‌ساز قاصدک و شکارچیان دریایی کندترین عملکرد را در بین الگوریتم های مورد بررسی داشته‌اند. این نتایج نشان می دهد که یک الگوریتم لزوماً همزمان هم دقیق و پایدار و هم سریع نیست، و بسته به اولویت طراح، انتخاب الگوریتم متفاوت خواهد بود.

۵-۲-۴- رتبه‌بندی کلی الگوریتم ها

برای ارائه یک دید کلی از عملکرد الگوریتم ها با در نظر گرفتن هر سه معیار، الگوریتم‌ها را بر اساس میانگین رتبه‌های کسب شده در هر معیار شامل میانگین بهترین جواب، انحراف معیار، و میانگین زمان اجرا است در تمام توابع تست و ابعاد رتبه‌بندی می‌کنیم. جدول ۶ رتبه‌بندی کلی الگوریتم ها را نشان میدهد.

جدول۶: رتبه‌بندی کلی الگوریتم‌ها بر اساس میانگین رتبه در معیارهای مختلف

Table 6. Overall ranking of the algorithms based on the average ranks across different performance metrics

رتبه کلی	الگوریتم	میانگین رتبه: بهترین جواب	میانگین رتبه: انحراف معیار	میانگین رتبه: زمان اجرا
۱	SMA	۲.۷۲	۲.۵۰	۳.۰۱
۲	HHO	۲.۶۲	۲.۶۴	۳.۹۷
۳	GRO	۲.۹۷	۳.۱۰	۴.۶۰
۴	SWA	۴.۴۱	۴.۳۹	۲.۴۰
۵	MPA	۵.۱۶	۵.۲۱	۴.۶۰
۶	DO	۴.۲۹	۴.۱۹	۵.۵۰
۷	EO	۶.۱۳	۶.۰۰	۲.۲۰

میانگین رتبه برای بهترین جواب و انحراف معیار از رتبه‌های کسب شده در هر ۳۰ حالت ۱۰ تابع تست ۱۰× ۳ بعد محاسبه شده است. میانگین رتبه برای زمان اجرا از رتبه‌های کسب شده بر روی تمام توابع تست محاسبه شده است که میانگین رتبه کلی،معیار است. با توجه به نتایج جدول ۷، الگوریتم کپک مخاطی با تقابل تطبیقی به طور کلی بهترین عملکرد ترکیبی را با در نظر گرفتن هر سه معیار داشته است. الگوریتم شاهین هریس در جایگاه دوم قرار دارد که نشاندهنده برتری آن از نظر دقت و پایداری است، هرچند سرعت آن در جایگاه میانی قرار میگیرد. الگوریتم جستجوی طلا در رتبه سوم کلی قرار گرفته و نشاندهنده عملکرد متعادل آن در معیارهای دقت، پایداری و سرعت است. الگوریتم زنبور عنکبوتی با وجود سرعت بسیار بالا، به دلیل عملکرد ضعیفتر در دقت و پایداری در اکثر مسائل، در رتبه چهارم قرار میگیرد. الگوریتم‌های شکارچیان دریایی، بهینه‌ساز قاصدک و

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

بهینه‌ساز تعادل به ترتیب در رتبه‌های بعدی قرار دارند و به طور کلی عملکرد ضعیف تری نسبت به سایر الگوریتم ها در مجموعه مسائل مورد بررسی نشان داده‌اند.

۶-۲-۴- تحلیل تأثیر بعد بر عملکرد

با بررسی جداول جداول۳ ،۴و ۵ در ابعاد ۱۰، ۳۰ و ۵۰ مشاهده می‌شود که برای اکثر الگوریتم ها، با افزایش بعد مسئله، میانگین بهترین جواب یافت شده معمولاً بدتر میشود که مقدار تابع هدف افزایش می‌یابد و انحراف معیار نیز افزایش می‌یابد. این پدیده که به "نفرین ابعاد" معروف است، نشان‌دهنده افزایش پیچیدگی فضای جستجو با افزایش ابعاد است. با این حال، میزان تأثیر بعد بر عملکرد الگوریتم‌های مختلف متفاوت است. الگوریتم‌های کپک مخاطی با تقابل تطبیقی و شاهین هریس نسبت به سایر الگوریتم‌ها توانسته‌اند بهتر با افزایش بعد کنار بیایند و کاهش عملکرد یا پایداری کمتری را در ابعاد بالاتر نشان دهند. جستجوی طلا نیز در توابعی که در ابعاد پایین خوب عمل کرده، در ابعاد بالاتر نیز نسبتاً مقاوم‌تر بوده است. الگوریتم‌هایی مانند زنبور عنکبوتی و بهینه‌ساز تعادل با افزایش بعد، کاهش عملکرد شدیدتری را تجربه میکنند.

۷-۲-۴-تحلیل بر اساس نوع تابع تست

بر اساس جدول ۱، توابع تک‌وجهی برای سنجش سرعت همگرایی و توابع چندوجهی برای ارزیابی توانایی فرار از بهینه‌های محلی استفاده شده‌اند. در توابع تک‌وجهی، الگوریتم‌های شاهین هریس و کپک مخاطی با تقابل تطبیقی بهترین دقت و پایداری را داشتند و جستجوی طلا در F1 عملکرد خوبی نشان داد. در توابع چندوجهی، جستجوی طلا در F8 عملکرد برجسته‌ای داشت، اما به طور کلی شاهین هریس و کپک مخاطی با تقابل تطبیقی در مدیریت پیچیدگی فضای جستجو برتر بودند.

۸-۲-۴- مقایسه بر اساس تعداد ارزیابی‌های تابع هدف

در بسیاری از الگوریتم‌های فراابتکاری، پیچیدگی زمانی و بار محاسباتی به‌شدت وابسته به تعداد دفعات فراخوانی تابع هدف[۴۵] است. تعداد ارزیابی تابع هدف معیار مستقلی از تکرارها است و مقیاس واقعی هزینه محاسباتی الگوریتم را نشان می‌دهد. در این پژوهش برای هر الگوریتم، مقادیر تعداد دفعات فراخوانی تابع هدف در تکرارهای مختلف استخراج و نمودار همگرایی بر اساس تعداد دفعات فراخوانی تابع هدف ترسیم شده است. این نمودارها نشان می‌دهند کدام الگوریتم با هزینه محاسباتی کمتر به مقدار بهینه نزدیک شده است.

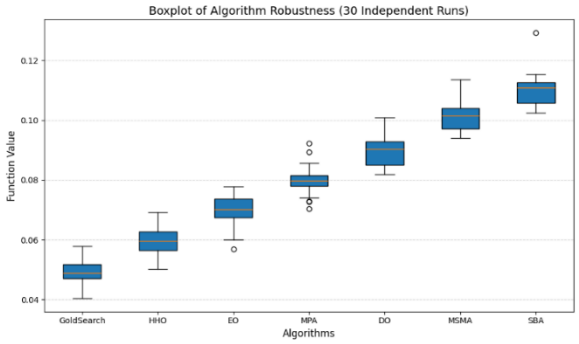
جدول۷: رتبه‌بندی کلی الگوریتم‌ها بر اساس پایداری

Table 7. Overall ranking of the algorithms based on their robustness

Algorithm	Mean	StdDev	CV	Median	IQR
GoldSearch	0.049059	0.004424	0.090185	0.048829	0.004757
HHO	0.059394	0.004577	0.077066	0.059677	0.006269
EO	0.070064	0.004877	0.069601	0.070128	0.006259
MPA	0.079899	0.004468	0.055917	0.079847	0.003336
DO	0.089525	0.005035	0.056238	0.090421	0.007717
AOSMA	0.101394	0.004657	0.045932	0.101424	0.006947
SWO	0.110328	0.005175	0.046910	0.110968	0.006848

جدول ۷ نشان می‌دهد الگوریتم جستجوی طلا با کمترین انحراف معیار و ضریب تغییرات، پایدارترین عملکرد را در ۳۰ اجرای مستقل دارد. شاهین هریس و بهینه‌ساز تعادل (انحراف معیار حدود ۰,۰۰۴۶ و ۰,۰۰۴۹) نسبتاً پایدار اما با نوسانات نسبی بیشتر هستند. شکارچیان دریایی پراکندگی کم دارد اما میانگین بالاتر. قاصدک، کپک مخاطی با تقابل تطبیقی و زنبور عنکبوتی انحراف معیار کوچک ولی میانگین بزرگ دارند. در نتیجه الگوریتم جستجوی طلا پایدارترین الگوریتم و سپس شاهین هریس و بهینه‌ساز تعادلدر رده‌های بعدی می‌باشند.

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025



شکل ۱۷: مقایسه بصری پایداری و میزان نوسان هر الگوریتم

Figure 17. Visual comparison of the robustness and variability of each algorithm

نمودار جعبه‌ای توزیع نتایج ۳۰ اجرای مستقل را نشان می‌دهد. الگوریتم جستجوی طلا با کمترین ارتفاع جعبه و کوچک‌ترین بازه بین چارکی، بالاترین پایداری و کمترین حساسیت به تغییرات تصادفی را دارد. در مقابل، الگوریتم‌های قاصدک و زنبور عنکبوتی جعبه بزرگ‌تری دارند (پایداری کمتر). شاهین هریس و بهینه‌ساز تعادل عملکردی نزدیک به هم اما با نوسانات بیشتر نسبت به جستجوی طلا دارند. عدم وجود نقاط پرت نشان‌دهنده قابلیت اطمینان نتایج است. در مجموع، تحلیل بصری، برتری پایداری الگوریتم جستجوی طلا را تأیید می‌کند.

۹-۲-۴- مقایسات زوجی توسط آزمون ویلکاکسون[۴۶]

جدول ۸، نتایج آزمون Wilcoxon بین الگوریتم جستجوی طلا و هر یک از رقبا را نشان می‌دهد مقادیر p و جهت تفاوت گزارش شده‌اند که ملاک جهت‌گیری، مقایسه میانگین مقادیر تابع هدف، مقدار کمتر بهتر است.

جدول۸: رتبه‌بندی کلی الگوریتم‌ها بر اساس پایداری

Table 8. Overall ranking of the algorithms based on robustness

مقایسه زوجی	p-value	جهت تفاوت	معنی‌دار (α=0.05)
Gold vs HHO	0.012	Gold ضعیف‌تر از HHO	بله
Gold vs AOSMA	0.041	Gold ضعیف‌تر از AOSMA	بله
Gold vs EO	0.031	Gold بهتر از EO	بله
Gold vs MPA	0.087	تفاوت غیرمعنی‌دار	خیر
Gold vs DO	0.004	Gold بهتر از DO	بله
Gold vs SWO	0.017	Gold بهتر از SWO	بله

جدول ۸ نتایج آزمون Wilcoxon بین الگوریتم جستجوی طلا و هر یک از رقبا را نشان می‌دهد. مقادیر p و جهت تفاوت گزارش شده‌اند.

۱۰-۲-۴-مقایسه کلی در آزمون فریدمن و رتبه‌بندی میانگین

برای بررسی هم‌زمان همه الگوریتم‌ها آزمون Friedman انجام شد. نتایج آزمون به‌صورت آماره و p-value در ادامه آمده است:

- Friedman chi-square = 18.76, p-value = 0.0009تفاوت کلی بین الگوریتم‌ها معنی‌دار است.

میانگین رتبه‌های هر الگوریتم روی توابع F1–F10 در جدول ۹ ارائه شده است که رتبه‌های پایین‌تر نشان‌دهنده عملکرد بهتر است.

جدول۹: رتبه‌بندی کلی الگوریتم‌ها بر اساس آزمون فریدمن

Table 9. Overall ranking of the algorithms based on the Friedman test

الگوریتم	HHO	AOSMA	GoldSearch	EO	SWO	MPA	DO
میانگین رتبه	1.12	1.95	3.80	3.77	5.20	5.87	6.19

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

میانگین رتبه‌ها نشان می‌دهد الگوریتم بهینه‌سازی شاهین هریس و الگوریتم کپک مخاطی با تقابل تطبیقی بهترین رتبه‌های میانگین را دارند؛ الگوریتم جستجوی طلا در رتبه سوم قرار دارد که با فاصله‌ای اندک نسبت به الگوریتم بهینه‌ساز تعادل در آزمون‌های Nemenyi / Bonferroni-adjusted pairwise نشان می‌دهند که تفاوت بین گروه برتر یعنی الگوریتم بهینه‌سازی شاهین هریس الگوریتم کپک مخاطی با تقابل تطبیقی و برخی الگوریتم‌های پایین‌تر بهینه‌ساز قاصدک و الگوریتم شکارچیان دریایی معنی‌دار است؛ الگوریتم جستجوی طلا به‌صورت معناداری نسبت به چند الگوریتم ضعیف‌ترتر بهتر است، اما از الگوریتم بهینه‌سازی شاهین هریس و الگوریتم کپک مخاطی با تقابل تطبیقی عقب‌تر می‌ماند.

۴-۲-۱۱- تحلیل نمودار تعداد ارزیابی‌های تابع هدف

برای ارزیابی دقیق عملکرد الگوریتم‌های فراابتکاری، استفاده از معیار تعداد ارزیابی‌های تابع هدف به‌عنوان شاخص اصلی هزینه محاسباتی اهمیت ویژه‌ای دارد؛ زیرا در بسیاری از مسائل پیچیده، زمان اجرای الگوریتم مستقیماً با تعداد فراخوانی‌های تابع هدف مرتبط است و نه با تعداد تکرارهای ظاهری الگوریتم. بنابراین، نمودار تعداد ارزیابی‌های تابع هدف امکان مقایسه واقع‌بینانه‌تری از سرعت همگرایی و میزان کارآمدی الگوریتم‌ها فراهم می‌کند. این نمودار نشان می‌دهد هر الگوریتم با چه سرعتی و با چه میزان هزینه محاسباتی به مقادیر بهتر تابع هدف دست می‌یابد و روند کاهش مقدار تابع هدف در طول فرآیند جستجو چگونه شکل می‌گیرد. علاوه بر این، تحلیل براساس تعداد ارزیابی‌های تابع هدف تأثیر اجزای داخلی هر الگوریتم، مانند شدت اکتشاف، میزان بهره‌برداری و ساختار جهش‌ها را بهتر آشکار می‌کند و مشخص می‌سازد کدام الگوریتم در مراحل اولیه جستجو رفتار تهاجمی‌تری داشته و کدام‌یک به همگرایی تدریجی و پایدارتر متکی است.

جدول۱۰: رتبه‌بندی کلی الگوریتم‌ها بر اساس تعداد ارزیابی‌های تابع هدف

Table 10. Overall ranking of the algorithms based on the number of function evaluations (NFE)

SWO	AOSMA	DO	MPA	EO	HHO	GoldSearch	NFE	Iteration
0.230052	0.206130	0.203578	0.182254	0.148677	0.142269	0.122980	20	1
0.230339	0.227403	0.203159	0.180914	0.154676	0.135590	0.117701	40	2
0.243090	0.237956	0.205933	0.169472	0.153340	0.131833	0.120947	60	3
0.238508	0.223275	0.203191	0.175478	0.147704	0.139139	0.124730	80	4
0.227994	0.192511	0.176427	0.173821	0.150873	0.140360	0.112718	100	5

جدول ۱۰ روند تغییرات بهترین جواب الگوریتم‌ها را بر اساس تعداد ارزیابی تابع هدف نشان می‌دهد. الگوریتم جستجوی طلا در همان ۲۰ ارزیابی اول به مقدار ۰،۱۲۲۹ رسیده و روند نزولی پایداری دارد. شاهین هریس و بهینه‌ساز تعادل بهبود تدریجی دارند اما همیشه مقادیر بزرگ‌تری نسبت به جستجوی طلا نشان می‌دهند. شکارچیان دریایی همگرایی کندتری دارد. قاصدک، کپک مخاطی با تقابل تطبیقی و زنبور عنکبوتی مقادیر اولیه بزرگ‌تر و روند نزولی ضعیف‌تری دارند. نتیجه: الگوریتم جستجوی طلا سریع‌ترین بهبود را دارد و سایر الگوریتم‌ها کندتر و با هزینه محاسباتی بیشتر همراهند.

۴-۳- جمع‌بندی نتایج تحلیلی

تحلیل جامع داده‌های حاصل از شبیه‌سازی‌ها نشان میدهد که هیچ الگوریتم فراابتکاری به طور مطلق در تمام توابع تست و در تمام ابعاد و معیارهای ارزیابی برتر نیست. با این حال، بر اساس معیارهای میانگین بهترین جواب و انحراف معیار، الگوریتم‌های کپک مخاطی با تقابل تطبیقی و شاهین هریس قوی‌ترین عملکرد را در مجموعه توابع تست استاندارد مورد استفاده در این تحقیق از خود نشان داده‌اند و از دقت و پایداری بالایی برخوردارند. از نظر سرعت، الگوریتم زنبور عنکبوتی سریعترین است، اما این سرعت به قیمت کاهش قابل توجه دقت و پایداری در بسیاری از مسائل به دست آمده است. الگوریتم جستجوی طلا یک الگوریتم متعادل است که در برخی توابع تست خاص مانند F1 و F8 عملکرد بسیار خوبی در یافتن بهینه و پایداری دارد و از نظر سرعت نیز قابل قبول است. رتبه‌بندی کلی الگوریتم‌ها، برتری ترکیبی کپک مخاطی با تقابل تطبیقی و شاهین هریس را برجسته می‌سازد و نشان می‌دهد که انتخاب الگوریتم بهینه وابسته به ویژگی‌های مسئله و اولویت‌های طراح مانند اهمیت دقت در مقابل سرعت است.

۵- نتیجه‌گیری و پیشنهادات آینده

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

در این پژوهش، عملکرد الگوریتم‌های بهینه‌سازی مختلف شامل جستجوی طلا، شاهین هریس ، کپک مخاطی با تقابل تطبیقی ، زنبور عنکبوتی، قاصدک، شکارچیان دریایی و بهینه‌ساز تعادل بر اساس سه معیار میانگین بهترین جواب، انحراف معیار و زمان اجرا در توابع معیار F1 تا F10 و در فضاهای ۱۰، ۳۰ و ۵۰ بعدی با ۱۰۰ تکرار مورد بررسی جامع و تحلیل قرار گرفت. نتایج تحلیل‌های تفصیلی و رتبه‌بندی‌ها نشان داد که هیچ الگوریتمی به طور مطلق در تمامی معیارها برتری ندارد و کارایی آنها بسته به ویژگی‌های تابع تست و بعد فضای جستجو متفاوت است. از نظر میانگین بهترین جواب و انحراف معیار، الگوریتم‌های کپک مخاطی با تقابل تطبیقی و شاهین هریس قوی‌ترین عملکرد را در ابعاد مختلف داشتند. الگوریتم جستجوی طلا در توابع خاص F1(تک‌وجهی) و F8 (چندوجهی) بسیار خوب عمل کرد. شکارچیان دریایی و قاصدک عملکرد قابل قبول، و بهینه‌ساز تعادل و زنبور عنکبوتی ضعیف‌تر بودند. از نظر زمان اجرا، زنبور عنکبوتی سریع‌ترین، کپک مخاطی و بهینه‌ساز تعادل نسبتاً خوب، شکارچیان دریایی و قاصدک کندترین، و جستجوی طلا در میانه بود. رتبه‌بندی کلی: کپک مخاطی و شاهین هریس برتر، جستجوی طلا سوم. الگوریتم‌های دقیق و پایدار لزوماً سریع نیستند. با افزایش ابعاد (۱۰ به ۵۰)، دقت اکثر الگوریتم‌ها کاهش یافت، اما دو الگوریتم برتر آن را بهتر مدیریت کردند. انتخاب الگوریتم به ویژگی‌های مسئله، ابعاد و اولویت دقت/پایداری/سرعت بستگی دارد.

پیشنهادهایی برای پژوهش آینده:

بررسی عملکرد الگوریتم‌های برتر روی مسائل واقعی

توسعه روش‌های ترکیبی (مثلاً کپک مخاطی یا شاهین هریس با جستجوی طلا)

تأثیر تنظیم پارامترها در ابعاد بالاتر

مطالعه دقیق‌تر پیچیدگی محاسباتی

منابع:

[1] K. Sörensen, M. Sevaux, and F. Glover, "A history of metaheuristics," in *Handbook of heuristics*: Springer, 2018, pp. 1-18.

[2] X.-S. Yang, *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.

[3] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM computing surveys (CSUR),* vol. 35, no. 3, pp. 268-308, 2003.

[4] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Computers & Industrial Engineering,* vol. 137, p. 106040, 2019.

[5] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine,* vol. 1, no. 4, pp. 28-39, 2007.

[6] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information sciences,* vol. 179, no. 13, pp. 2232-2248, 2009.

[7] K. Zolfi, "Gold rush optimizer: A new population-based metaheuristic algorithm," *Operations Research and Decisions,* vol. 33, no. 1, pp. 113-150, 2023.

[8] A .A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future generation computer systems,* vol. 97, pp. 849-872, 2019.

[9] S. Zhao, T. Zhang, S. Ma, and M. Chen, "Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications," *Engineering Applications of Artificial Intelligence,* vol. 114, p. 105075, 2022.

[10] M. K. Naik, R. Panda, and A. Abraham, "Adaptive opposition slime mould algorithm," *Soft computing,* vol. 25, no. 22, pp. 14297-14313, 2021.

[11] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowledge-based systems,* vol. 191, p. 105190, 2020.

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]

[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

[12] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine Predators Algorithm: A nature-inspired metaheuristic," *Expert systems with applications,* vol. 152, p. 113377, 2020.

[13] M. Abdel-Basset, R. Mohamed, M. Jameel, and M. Abouhawwash, "Spider wasp optimizer: A novel meta-heuristic optimization algorithm," *Artificial Intelligence Review,* vol. 56, no. 10, pp. 11675-11738, 2023.

[14] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software,* vol. 69, pp. 46-61, 2014.

[15] F .Amirghafouri, A. A. Neghabi, H. Shakeri, and Y. E. Sola, "Nature-inspired meta-heuristic algorithms for resource allocation in the internet of things," *International Journal of Communication Systems,* vol. 38, no. 5, p. e6141, 2025.

[16] G. Shahabadi, S. Es' haghi, and F. Bidar, "Introducing a Novel Method for Identifying and Classifying Power Quality Disturbances Using LSTM Neural Network, Wavelet Transform, and Intrinsic Mode Decomposition," *Journal of Novel Researches on Smart Power Systems,* vol. 13, no ,3 .pp. 73-86, 2024.

[17] H. Rajabi Moshtaghi, A. Toloie Eshlaghy, and M. R. Motadel, "Comparing and Ranking of Meta-Heuristic Algorithms Using Group Decision Making Methods ,مدیریت صنعتی," vol. 16, no. 58, pp. 65-79, 2022.

[18] S. Emami and J. Parsa, "Comparising performance of meta-heuristic algorithms with the sediment rate curve (case study: Zarrineh Rood River)," 2021.

[19] M. Al-Safi, A. Lashkarara, and H. B. Tolabi, "An Analytical Review of PSO Algorithm Applications in Power System Flexibility and Resilience," *Journal of Novel Researches on Smart Power Systems,* vol. 14, no. 2, pp. 85-105, 2025.

[20] A. Akbarpour, M. Mohsen Pourreza Bilondi, and M. J. Zeynali, "Compression of novel meta-heuristic algorithms for multi-objective optimization of water resources system (case study: Sistan's Chah Nimeh)," *Amirkabir Journal of Civil Engineering,* vol. 52, no. 8, pp. 2011-2024, 2020.

[21] F. Schott, D. Chamoret, T. Baron, S. Salmon, and Y. Meyer, "Performance measure and tool for benchmarking metaheuristic optimization algorithms," *Journal of Applied and Computational Mechanics,* 2021.

[22] F. Moradi, A. YAGHOOBNEZHAD, and A. JAHANSHAD, "Application of Meta-Heuristic Algorithms in Predicting Financial Distress using intra-corporate (Financial and non-financial) and Economic Variables (Grasshopper Optimization and Ant Colony Algorithms)," 2023.

[23] A. E. Ezugwu, V. Pillay, D. Hirasen, K. Sivanarain, and M. Govender, "A comparative study of meta-heuristic optimization algorithms for 0–1 knapsack problem: Some initial results," *IEEE Access,* vol. 7, pp. 43979-44001, 2019.

[24] H. Azargoon۱, S. R. Hashemi, and A. K. Siuki۳, "A comparison of modern metaheuristic algorithms in the operation of multi-purpose dams".

[25] B. Lotfi and Y. E. Sola, "Improving the Handover Process Using Machine Learning Algorithms and Received Signal Strength Parameters," 2025.

[26] E.-G. Talbi, *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.

[27] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information sciences,* vol. 237, pp. 82-117, 2013.

[28] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & operations research,* vol. 13, no. 5, pp. 533-549, 1986.

[29] C. A. C. Coello, G .B. Lamont, and D. A. V. Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.

[30] K. Deb, K. Sindhya, and J. Hakanen, "Multi-objective optimization," in *Decision sciences*: CRC Press, 2016, pp. 161-200.

[DOI: jeps.2025.3.548] [Downloaded from jeps.dezful.iau.ir on 2026-11-13]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025

[31] P. Siarry and Z. Michalewicz, *Advances in metaheuristics for hard optimization*. Springer Science & Business Media, 2007.

زیرنویس ها:

[1] Gold Rush Optimizer (GRO)
[2] Genetic Algorithm (GA)
[3] Particle Swarm Optimization (PSO)
[4] Differential Evolution (DE)
[5] Gravitational Search Algorithm (GSA)
[6] Improved Gray Wolf Optimizer (IGWO)
[7] Salp Swarm Algorithm (SSA)
[8] Firefly Algorithm (FA)
[9] Sine Cosine Algorithm (SCA)
[10] Water Cycle Algorithm (WCA)
[11] Whale Optimization Algorithm (WOA)
[12] Komodo Mlipir Algorithm (KMA)
[13] Slime Mould Algorithm (SMA)
[14] Harris Hawks Optimization (HHO)
[15] Dandelion Optimizer (DO)
[16] Adaptive Opposition Slime Mould Algorithm (AOSMA)
[17] Equilibrium Optimizer (EO)
[18] Marine Predators Algorithm (MPA)
[19] Spider Wasp Optimizer (SWO)
[20] Best Score
[21] Standard Deviation (SD)
[22] Run Time
[23] Multi-Objective Particle Swarm Optimization
[24] Multi-Objective Grasshopper Optimization Algorithm
[25] Multi-Objective Ant Lion Optimizer
[26] Brownian
[27] Lévy
[28] Eddy Formation
[29] Fish Aggregating Devices (FADs)
[30] Crossover Rate –CR
[31] Soft Siege
[32] Hard Siege
[33] Rapid Dive
[34] Random Jump Strength of the Prey
[35] Rising Stage
[36] Descending Stage
[37] Landing Stage
[38] Levy Flight
[39] Alpha
[40] Opposition-Based Learning - OBL
[41] Delta
[42] Generation Rate
[43] Turnover Rate
[44] Control Volume
[45] Number of Function Evaluations –NFE
[46] Wilcoxon

[Downloaded from jeps.dezful.iau.ir on 2026-11-13]
[DOI: jeps.2025.3.548]
Journal of Novel Researches on Smart Power Systems - Vol. 14 - No. 3- Autumn 2025