



Semnan University



Research Article

Solutions To Improve The Performance Of Cuckoo Optimization Algorithm

Hossein Nahid Titkanloo ^{a,*}, Mohadese Nadershahi ^a, Mohammad Golmakani ^b

^a Department of Industrial Engineering, Payame Noor University, Tehran, Iran

^b Ph.d student, Payame Noor University, Tehran, Iran

PAPER INFO

Paper history:

Received: 2022-03-13

Revised: 2022-05-12

Accepted: 2022-10-19

Keywords:

Cuckoo algorithm;
Metaheuristic algorithms;
Optimization;
benchmark functions.

ABSTRACT

This paper presents strategies to enhance the performance of the Cuckoo Search Algorithm. These strategies include modifications and new methods for determining the number of eggs each cuckoo lays, setting the egg-laying radius, and defining the step size of migration. The proposed strategies were evaluated using the Sphere and Rastrigin benchmark functions, and the results indicate that these methods outperform the original Cuckoo Search Algorithm. The first proposal, which focuses on determining the number of eggs each cuckoo lays, showed significant improvement in solving the Sphere and Rastrigin functions. Additionally, the second set of proposals, concerning the egg-laying radius and migration step size, also led to considerable improvements in solving the benchmark functions. Finally, the simultaneous application of both proposed modifications to the basic Cuckoo Search Algorithm demonstrated substantial enhancements in optimizing the benchmark functions. Therefore, the proposed modifications can serve as reliable strategies, alongside other improvements, for solving optimization problems using the Cuckoo Search Algorithm.

DOI: <https://doi.org/>

© 2024 Published by Semnan University Press.

This is an open access article under the CC-BY 4.0 license. (<https://creativecommons.org/licenses/by/4.0/>)

* Corresponding author.

E-mail address: Hossein_Nahid@pnu.ac.ir

How to cite this article:

راهکارهایی جهت بهبود عملکرد الگوریتم بهینه سازی فاخته

حسین ناهید تیتکانلو^{۱*}، محدثه نادرشاهی^۱، محمد گلمکانی^۲

اطلاعات مقاله	چکیده
دریافت مقاله: ۱۳۹۵/۰۳/۲۰	این مقاله به ارائه راهکارهایی جهت بهبود عملکرد الگوریتم فاخته می‌پردازد. این راهکارها شامل اصلاحات و روش‌های جدیدی در تعیین تعداد تخم‌های هر فاخته، تعیین شعاع تخم‌گذاری، و تعیین گام مهاجرت می‌باشد. راهکارهای پیشنهادی با استفاده از توابع محک اسفیر و راستریجین ارزیابی شده‌اند و نتایج نشان می‌دهند که این روش‌ها در مقایسه با نسخه اولیه الگوریتم فاخته، عملکرد بهتری دارند. پیشنهاد اول در تعیین تعداد تخم‌های هر فاخته، در حل توابع محک اسفیر و راستریجین، عملکرد بسیار خوبی داشته است. همچنین، پیشنهادات اصلاحی دوم در مورد تعیین شعاع تخم‌گذاری و گام مهاجرت نیز به بهبود قابل قبولی در حل توابع محک مورد بررسی منجر شده‌اند. در نهایت، اعمال همزمان دو پیشنهاد اصلاحی در الگوریتم پایه فاخته، بهبودهای قابل توجهی در بهینه‌سازی توابع محک مورد بررسی نشان داده است. لذا اصلاحات پیشنهادی می‌توانند به عنوان راهکارهای قابل اتکا در کنار سایر اصلاحات اعمال شده در الگوریتم فاخته، برای حل مسائل بهینه‌سازی مورد استفاده قرار گیرند.
بازنگری مقاله: ۱۳۹۵/۰۶/۲۰	
پذیرش مقاله: ۱۳۹۵/۰۷/۱۵	
واژگان کلیدی: الگوریتم فاخته، الگوریتم‌های فراابتکاری، بهینه‌سازی، توابع محک.	

DOI: <https://doi.org/>

© 2024 Published by Semnan University Press.

This is an open access article under the CC-BY 4.0 license. (<https://creativecommons.org/licenses/by/4.0/>)

۱- مقدمه

اصلی این روش‌ها، کارایی ناکافی در مسائل بهینه‌سازی پیچیده و بزرگ است، به طوری که زمان اجرای آن‌ها به صورت نمایی با افزایش ابعاد مسئله افزایش می‌یابد. در نتیجه، استفاده از الگوریتم‌های دقیق برای مسائل بزرگ و پیچیده معمولاً عملی نیست.

در مقابل، الگوریتم‌های تقریبی که به عنوان الگوریتم‌های فراابتکاری نیز شناخته می‌شوند، طراحی شده‌اند تا بتوانند جواب‌های خوبی که به جواب بهینه نزدیک هستند را در زمان کوتاه‌تری بیابند. این الگوریتم‌ها به ویژه برای مسائل بهینه‌سازی پیچیده که یافتن جواب بهینه دقیق در آن‌ها بسیار زمان‌بر است، مناسب هستند. الگوریتم‌های تقریبی با

بهینه‌سازی به معنای دستیابی به بهترین نتیجه ممکن برای یک مسئله با توجه به شرایط موجود است. تاکنون، روش‌های متعددی برای حل مسائل بهینه‌سازی توسعه یافته‌اند. این روش‌ها و الگوریتم‌ها به دو دسته الگوریتم‌های دقیق^۳ و الگوریتم‌های تقریبی^۴ تقسیم بندی میشوند. الگوریتم‌های دقیق به گونه‌ای طراحی شده‌اند که قادر به یافتن جواب بهینه برای مسائل مختلف به صورت دقیق و قطعی هستند. با این حال، این الگوریتم‌ها برای تمامی مسائل بهینه‌سازی قابل استفاده نیستند. یکی از مشکلات

* پست الکترونیک نویسنده مسئول: Hossein_Nahid@pnu.ac.ir

^۱ استادیار، گروه مهندسی صنایع، دانشگاه پیام نور، تهران، ایران

^۲ دانشجوی دکتری مهندسی صنایع، دانشگاه پیام نور، تهران، ایران

^۳ Exact

^۴ Approximate

شعاع تخم‌گذاری که در بخش ۲-۱ تشریح شده است، نشان‌دهنده محدوده مجازی است که هر فاخته می‌تواند برای تخم‌گذاری انتخاب کند و در واقع دامنه مجاز تعیین متغیرهای پاسخ است. تعیین شعاع تخم‌گذاری در الگوریتم فاخته به تعداد تخم‌های هر فاخته مرتبط است و رابطه‌ای با کیفیت پاسخ به دست آمده ندارد. لذا به دنبال آن هستیم تا با ایجاد ارتباط بین شیوه تخم‌گذاری و میزان شایستگی هر فاخته الگوریتم را بهبود دهیم.

• تغییر ضریب گام مهاجرت و زاویه حرکت فاخته‌ها در

طول اجرای الگوریتم

در الگوریتم پایه فاخته، ضریب و زاویه حرکت به صورت تصادفی در دامنه‌های تعریف شده تعیین می‌شود. در این پژوهش می‌خواهیم این موارد را به کیفیت پاسخ‌ها مرتبط نماییم.

۲- مبانی تئوری و بررسی کارهای انجام شده

۲-۱- الگوریتم فاخته‌ها

الگوریتم بهینه‌سازی فاخته به‌عنوان یکی از الگوریتم‌های کارآمد در حوزه بهینه‌سازی، برای حل مسائل بهینه‌سازی غیرخطی و مسائل بهینه‌سازی پیوسته توسعه یافته است. این الگوریتم، که به‌عنوان یکی از جدیدترین و قوی‌ترین الگوریتم‌های فراابتکاری مبتنی بر جمعیت شناخته می‌شود، در سال ۲۰۱۱ توسط رامین رجبیون معرفی شد. الگوریتم بهینه‌سازی فاخته با الهام از رفتارهای طبیعی و استراتژی‌های تخم‌گذاری فاخته‌ها، توانایی ارائه راه‌حل‌های کارآمد و مؤثر برای مسائل پیچیده بهینه‌سازی را دارد [۲]. مطالعات متعددی که پیرامون این الگوریتم انجام شده‌اند، حاکی از سرعت بالای همگرایی آن در یافتن نقطه بهینه سراسری، به‌ویژه در مسائل پیوسته با ابعاد بالا، هستند [۳]. [۴]. افزون بر این، این الگوریتم به دلیل انعطاف‌پذیری و کارایی بالا در حل مسائل پیچیده، کاربردهای گسترده‌ای در حوزه‌های مختلف داشته است [۵-۱۲]. در ادامه مراحل این الگوریتم بر اساس مقاله رجبیون [۲] ارائه می‌گردد. برای توصیف بهتر فلوچارت الگوریتم فاخته در شکل (۱) آمده است.

استفاده از تکنیک‌های هوشمند و به کارگیری روش‌های ابتکاری، قادر به تولید جواب‌هایی هستند که اگرچه دقیقاً بهینه نیستند، اما کیفیت آن‌ها به اندازه‌ای است که در بسیاری از کاربردها قابل قبول و مفید باشد [۱].

الگوریتم‌های تقریبی نیز به سه دسته الگوریتم‌های ابتکاری^۵ و فراابتکاری^۶ و فوق‌ابتکاری^۷ بخش بندی می‌شوند. الگوریتم‌های ابتکاری به طور کلی به دنبال یافتن جواب‌های خوب با استفاده از روش‌های ساده و مستقیم هستند. این الگوریتم‌ها اغلب دارای دو نقطه ضعف اصلی هستند: اول، امکان توقف در نقاط بهینه محلی که باعث می‌شود نتوانند به بهینه سراسری دست یابند؛ دوم، همگرایی زودرس به این نقاط که ممکن است مانع از کشف سایر نواحی مهم فضای جستجو شود. برای حل این نقص، الگوریتم‌های فرا ابتکاری ارائه شده اند [۱]. در دهه‌های اخیر تعداد زیادی از این نوع الگوریتم‌ها توسعه یافته است. الگوریتم بهینه‌سازی فاخته یکی از این الگوریتم‌ها می‌باشد که به واسطه عملکرد مناسب خود در سالیان اخیر مورد توجه قرار گرفته است.

این الگوریتم، بر اساس شیوه زندگی بهینه و ویژگی‌های جالب نظیر تخم‌گذاری و تولید مثل خانواده‌ای از پرندگان به نام فاخته توسعه یافته است [۲].

الگوریتم فاخته با وجود مزیت‌های نسبی، در مواردی مانند تغییر شعاع همگرایی به صورت صحیح و پرداختن به یک جستجوی محلی قوی بالاخص در مسائل با ابعاد بزرگ، به جمعیت بزرگی از فاخته‌ها نیاز دارد. همچنین عملکرد پرواز یا مهاجرت به سمت نقطه بهینه، در مسائل با ابعاد بالا، مبهم بوده و نیاز به اصلاح و بازتعریف دارد [۳]. در این پژوهش اعتقاد بر این است که با در نظر گرفتن بدنه اصلی الگوریتم استاندارد فاخته، می‌توان با تغییر برخی پارامترهای آن، اصلاحاتی در جهت ارتقای دقت و سرعت همگرایی این الگوریتم ایجاد نمود. بهبود شیوه تخم‌گذاری بر اساس میزان شایستگی هر فاخته، اصلاح شعاع همگرایی، بازتعریف عملکرد پرواز به سمت نقطه بهینه، تغییراتی است که در این پژوهش مورد بررسی قرار می‌گیرد و جزئیات آن به شرح زیر می‌باشد:

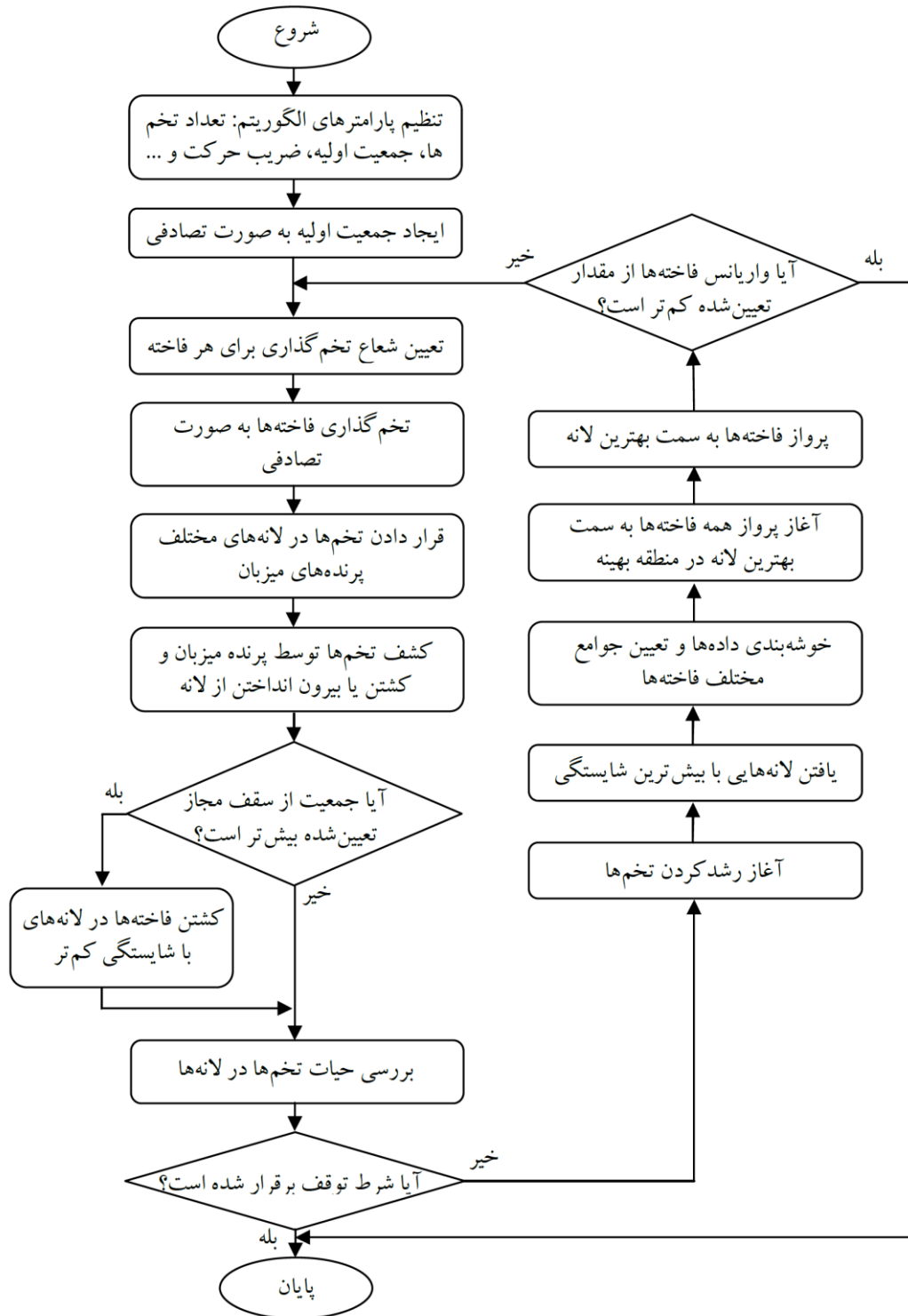
• تغییر شعاع تخم‌گذاری^۸

⁵ Heuristic

⁶ Meta-Heuristic

⁷ Hyper Heuristic

⁸ Egg Laying Radius



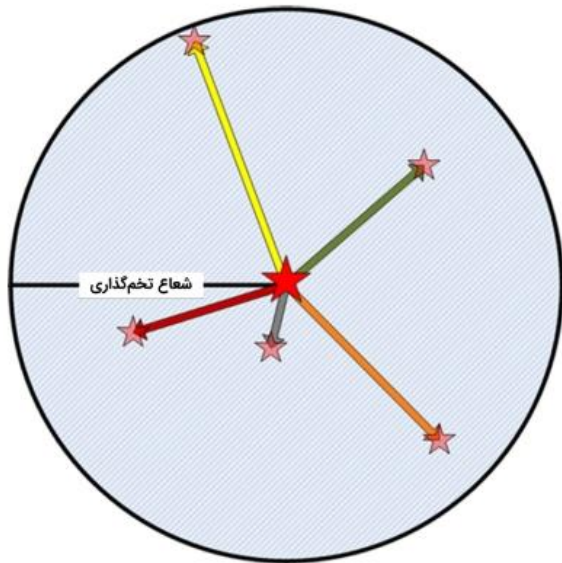
شکل ۱- فلوجارت الگوریتم بهینه سازی فاخته [۲]

شباهت بیشتری به تخم‌های پرندگانی میزبان دارند، فرصت رشد می‌یابند و به فاخته‌های بالغ تبدیل می‌شوند. اما تخم‌هایی که شباهت کمتری به تخم‌های درون لانه دارند، شناسایی شده و توسط پرندگانی میزبان بیرون انداخته

• مرحله (۱): شکل گیری جمعیت اولیه
 الگوریتم با جمعیت اولیه‌ای از فاخته‌ها آغاز می‌شود. این فاخته‌ها شروع به تخم‌گذاری می‌کنند و تخم‌های خود را در لانه پرندگان میزبان قرار می‌دهند. برخی از این تخم‌ها که

$$ELR = \alpha * \frac{n}{N} * (var_{Hi} - var_{Low}) \quad (3)$$

α در این رابطه یک عدد مثبت است که حداکثر نسبت محدوده مجاز تغییر متغیرها را تعیین می‌کند، n تعداد تخم‌های فاخته، N تعداد کل تخم‌ها و var_{Hi} و var_{Low} حد بالا و حد پایین متغیرهای تصمیم می‌باشند. در شکل (۲) مفهوم شعاع تخم گذاری به صورت شماتیک نشان داده شده است.



شکل ۲- مفهوم شعاع تخم گذاری [۲]

پس از پایان فرآیند تخم گذاری، تخم‌هایی که شباهت کمتری به تخم‌های پرنده میزبان دارند، توسط پرنده میزبان شناسایی شده و دور انداخته می‌شوند. بنابراین، پس از هر فرآیند تخم گذاری $\rho\%$ از تخم‌های فاخته از بین می‌روند.

• مرحله (۳): مهاجرت فاخته‌ها

جامعه فاخته‌ای که بیشترین سود، به معنای دسترسی به بیشترین منابع غذایی و بهترین شرایط محیطی برای زندگی را دارد، به عنوان مقصد مهاجرت فاخته‌ها در دیگر جوامع انتخاب می‌شود. لذا جمعیت فاخته‌ای موجود در محیط، با استفاده الگوریتم خوشه‌بندی K-means به گروه‌های مختلف تقسیم و طبقه‌بندی می‌شوند. این عمل به فاخته‌ها کمک می‌کند تا بهترین مقصد مهاجرت را برای تامین نیازهایشان انتخاب کنند [۱۳]. در شبیه‌سازی‌های انجام شده، مناسب‌ترین مقدار پارامتر k برای الگوریتم خوشه بندی، بین ۳ تا ۵ خوشه است. پس از گروه‌بندی جمعیت

می‌شوند. این فرآیند تا زمانی ادامه پیدا می‌کند که تخم‌های رشد یافته‌ای که مناسب بودن لانه‌های آن منطقه را نشان می‌دهند، باقی بمانند.

تعداد تخم‌های زنده در هر منطقه نشان‌دهنده شایستگی و سود بیشتر در آن منطقه است. هرچه تعداد تخم‌های زنده بیشتر باشد، سود بیشتری در آن ناحیه به دست می‌آید. این الگوریتم به دنبال یافتن بهترین محل برای زندگی و بقا تخم‌های فاخته است. بنابراین، مسیریابی بهینه تخم‌ها به منظور افزایش فرصت بقا و بهبود شرایط زندگی آن‌ها، هدف اصلی این الگوریتم است.

در الگوریتم بهینه‌سازی فاخته، مفهوم "زیستگاه"^۹ به یک آرایه از مقادیر متغیرهای مسئله اشاره دارد که به عنوان محل سکونت یا لانه برای فاخته‌ها عمل می‌کند. هر زیستگاه به عنوان یک نمونه یا جواب کاندید برای مسئله بهینه‌سازی مورد نظر تلقی می‌شود. زیستگاه‌ها در طول فرآیند تکاملی به مرور به جواب بهینه سراسری همگرا می‌شوند. برای حل یک مسئله توسط الگوریتم بهینه‌سازی فاخته، باید هر راه حل (که آن را محل سکونت یا لانه می‌نامیم) به صورت برداری از متغیرها شکل بگیرد.

در یک فضای N بعدی، محل سکونت هر فاخته مادر یا فاخته میزبان، توسط یک بردار با N بعد نمایش داده می‌شود که به صورت زیر تعریف می‌شود:

$$habitate = [x_1, x_2, x_3, \dots, x_{N_{var}}] \quad (1)$$

همانطور که ذکر شد، مقادیر متغیرها در الگوریتم بهینه‌سازی فاخته به صورت پیوسته هستند. مقدار سود یا شایستگی هر زیستگاه، با استفاده از مقدار تابع برازندگی در آن نقطه به صورت زیر محاسبه می‌شود:

$$Profit = f_b(x_1, x_2, x_3, \dots, x_{N_{var}}) \quad (2)$$

• مرحله (۲): تخم گذاری فاخته‌ها

شعاع تخم گذاری نشان‌دهنده دامنه مجاز مقادیری است که هر فاخته می‌تواند برای متغیرها انتخاب کند. در یک مسئله بهینه‌سازی، شعاع تخم گذاری برای هر فاخته متناسب با تعداد کلی تخم‌ها، حد پایین و حد بالای متغیرها محاسبه می‌شود. محدوده تخم گذاری با ELR^{10} نشان داده می‌شود و از رابطه زیر به دست می‌آید:

¹⁰ Egg Laying Radius

⁹ Habitat

می‌شود. این پارامتر مسئولیت کنترل و محدود کردن تعداد فاخته‌های موجود در محیط را دارد.

• مرحله (۶): همگرایی به جواب بهینه

پس از انجام تعدادی تکرار در فرایند الگوریتم بهینه‌سازی فاخته، همه اعضای جمعیت فاخته به بهترین زیستگاه موجود در محیط همگرا می‌شوند. وقتی بیش از ۹۵ درصد اعضای جمعیت فاخته به یک زیستگاه همگرا شوند، فرآیند اجرای الگوریتم بهینه‌سازی فاخته به پایان می‌رسد.

شبه کد الگوریتم بهینه‌سازی فاخته به شرح زیر می‌باشد:

• در ابتدا، جمعیت فاخته‌ها از طریق مقاردهی تصادفی به متغیرهای تابع هدف مسأله بهینه‌سازی مورد نظر، تشکیل می‌شود.

• در مرحله بعد، تعدادی تخم به هر یک از فاخته‌ها اختصاص داده می‌شود.

• برای هر فاخته، شعاع تخم‌گذاری تعیین می‌شود

• هر فاخته در محدوده‌ای که بر اساس شعاع تخم‌گذاری مشخص شده است، تخم‌گذاری می‌کند.

• تخم‌های فاخته‌ای که توسط پرنده میزبان شناسایی شوند (جواب‌های ضعیف) از بین می‌روند.

• تخم‌های فاخته باقی مانده، سر از تخم بیرون می‌آورند و رشد می‌کنند (بالغ می‌شوند).

• زیستگاه هر کدام از فاخته‌های بالغ ارزیابی می‌شود (میزان سود مشخص می‌شود).

• با توجه به پارامتر N_{max} جمعیت فاخته‌ای موجود در محیط محدود می‌شود و فاخته‌هایی که در بدترین زیستگاه‌ها زندگی می‌کنند، از جمعیت حذف می‌شوند.

• جمعیت فاخته‌ها بر اساس میزان سود خوشه‌بندی می‌شوند.

• پس از انجام خوشه‌بندی بر روی جمعیت فاخته، میانگین سود هر گروه محاسبه شده و بهترین زیستگاه به عنوان مقصد مهاجرت گروه‌های دیگر انتخاب می‌شود.

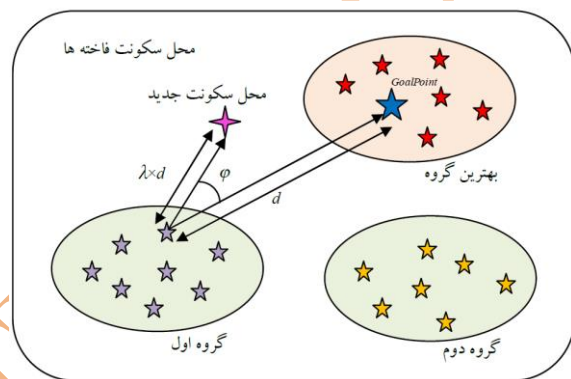
• جمعیت فاخته‌ای به سمت زیستگاه بهینه انتخاب شده مهاجرت می‌کنند.

• در صورتی که کلیه فاخته‌ها به نقطه بهینه سراسری مهاجرت نمایند یا بیش از ۹۵ درصد از جمعیت فاخته‌ها به یک زیستگاه خاص همگرا گردند، فرآیند اجرای الگوریتم بهینه‌سازی فاخته پایان می‌یابد؛ در غیر این

فاخته‌ای، سود میانگین^{۱۱} هر خوشه محاسبه می‌شود. بهترین زیستگاه موجود در گروه فاخته‌ای که بیشترین سود میانگین را دارد، به عنوان زیستگاه هدف و مقصد مهاجرتی دیگر گروه‌های فاخته‌ای انتخاب می‌شود.

• مرحله (۴): پرواز به سوی بهترین محل سکونت

هر فاخته فقط λ درصد مسیر را با درجه انحراف ϕ رادین به سمت زیستگاه هدف پرواز می‌کند (شکل ۳)، که این پارامترها به فاخته‌ها کمک می‌کنند تا مناطق بیشتری را برای یافتن زیستگاه بهینه با بیشترین منابع غذایی و بهترین شرایط زندگی جستجو کنند [۲].



شکل ۳- نمایش شماتیک پرواز فاخته‌ها به سمت بهترین محل سکونت در فضای دو بعدی [۲]

برای هر فاخته، مقادیر λ و ϕ به صورت زیر تعریف می‌شوند.

$$\lambda \sim U(0,1)$$

$$\phi \sim U(-\omega, \omega), (\omega = \frac{\pi}{6}) \quad (4)$$

در این رابطه، پارامتر λ یک متغیر تصادفی است که اعدادی بین ۰ و ۱ را شامل می‌شود و از توزیع یکنواخت پیروی می‌کند. در این رابطه، پارامتر ω نیز نشان‌دهنده میزان درجه انحراف از مکان زیستگاه هدف است. برای هم‌گرایی جمعیت فاخته‌ها به بهینه سراسری، مقدار $\pi/6$ برای پارامتر ω به عنوان یک مقدار مناسب در نظر گرفته می‌شود [۲]. با این توصیف، رابطه حرکت در فضای پیوسته به صورت زیر تعریف می‌شود.

$$habitate_{new} = habitate_{old} + \lambda * \cos\phi \otimes (habitate_{best} - habitate_{old}) \quad (5)$$

• مرحله (۵): تعادل جمعیت فاخته‌ها در محیط

با توجه به اینکه حفظ تعادل در جمعیت فاخته‌ها بسیار اهمیت دارد، یک پارامتر کنترلی به نام N_{max} تعریف

¹¹ Mean Profit

۳- پیشنهادهایی جهت بهبود الگوریتم بهینه سازی فاخته

۳-۱- ایجاد ارتباط بین شایستگی و تعداد تخمگذاری هر فاخته

در الگوریتم فاخته، در مرحله اولیه، یک حد پایین و بالا برای تعداد تخم‌های هر فاخته تعیین می‌شود. سپس با استفاده از اعداد تصادفی با توزیع یکنواخت که در این بازه تولید می‌گردند، تعداد تخم‌های هر فاخته مشخص می‌شود. این فرآیند ممکن است منجر به تخصیص تعداد تخم کمتر به فاخته‌های با عملکرد بهتر و تعداد تخم بیشتر به فاخته‌های ضعیف‌تر شود. چنین وضعیتی می‌تواند میانگین شایستگی جمعیت فاخته‌ها را کاهش داده و فرآیند رسیدن به جواب بهینه را کند نماید.

در روش پیشنهادی (رابطه ۶)، فرآیند تخصیص تخم‌ها با رویکردی هوشمندانه‌تر صورت می‌گیرد. در این روش، نسبت سود هر فاخته (پاسخ مورد نظر) به بهترین پاسخ موجود، به‌عنوان معیاری برای تعیین تعداد تخم هر فاخته به کار گرفته می‌شود. به این ترتیب، به جای تخصیص تصادفی تعداد تخم‌ها، میزان سودمندی یا مزیت زیستگاه هر فاخته به‌طور مستقیم بر تعداد تخم‌های آن تأثیرگذار است.

$$Egg_i = Min_Eggs + \left[(Max_Eggs - Min_Eggs) \times \frac{P_{C_i}}{P_{Max}} \right] \quad (6)$$

که در آن:

- Egg_i : تعداد تخم‌های فاخته i ام
- Min_Eggs : حداقل تعداد تخم‌های مجاز هر فاخته
- Max_Eggs : حداکثر تعداد تخم‌های مجاز هر فاخته
- P_{C_i} : مقدار سود فاخته i (مقدار تابع هدف از جنس سود جواب i)
- P_{Max} : ماکزیمم سود محاسبه شده در بین جمعیت فعلی فاخته‌ها

از طرف دیگر با توجه به رابطه تعیین شعاع تخم‌گذاری در الگوریتم بهینه سازی فاخته، که با تعداد تخم‌های هر فاخته نسبت مستقیم دارد، معیار کیفیت زیستگاه فاخته در تعیین شعاع تخم‌گذاری نیز اعمال می‌شود و به فاخته‌های برتر،

صورت، الگوریتم به گام دو منتقل شده و الگوریتم دوباره اجرا می‌شود.

۲-۲- مطالعات انجام شده مرتبط با بهبود الگوریتم بهینه سازی فاخته

باوجود نقاط قوت متعدد، الگوریتم فاخته محدودیت‌ها و نقاط ضعفی نیز دارد که در مطالعات متعدد مورد بررسی قرار گرفته است. از آنجا که در شکل پایه الگوریتم بهینه سازی فاخته، پارامترها به صورت ثابت در نظر گرفته می‌شوند، اصلاح و تعیین این پارامترها به شکل بهینه به انگیزه ای جهت انجام پژوهش‌های جدید و بهبود این الگوریتم تبدیل شده است. به عنوان مثال در منبع [۱۴] نحوه تعیین شعاع تخم‌گذاری فاخته‌ها اصلاح شده است و یا در منبع [۱۵] نسخه تطبیقی از الگوریتم فاخته با اعمال اصلاحاتی در استراتژی مهاجرت و شعاع تخم‌گذاری ارائه شده است. برخی از مهم‌ترین مطالعات انجام شده در این زمینه در جدول (۱) ذکر شده است.

جدول ۱- مطالعات انجام شده در حوزه بهبود الگوریتم فاخته

اصلاحات انجام شده	سال	منبع
اصلاح الگوریتم برای افزایش سرعت همگرایی	۲۰۱۱	[۳]
ارائه الگوریتم بهینه سازی فاخته مبتنی بر آشوب	۲۰۱۸ ۲۰۲۱	[۱۶، ۴]
توسعه الگوریتم بهینه سازی فاخته گسسته	۲۰۱۴	[۱۷]
افزودن رفتار کوانتومی به الگوریتم فاخته	۱۳۹۲	[۱۸، ۱۶]
ارائه راه حل برای رفع مشکل رکود الگوریتم ناشی از شعاع همگرایی	۲۰۱۲	[۱۴]
اصلاح الگوریتم برای حل مسائل بهینه سازی گسسته و باینری	۲۰۱۷	[۱۹]
استفاده از الگوریتم فاخته برای آموزش شبکه‌های عصبی	۲۰۱۳	[۲۰]
ترکیب الگوریتم ژنتیک و الگوریتم فاخته ^{۱۲}	۲۰۱۴	[۲۱]
الگوریتم ترکیبی فاخته- گرانشی	۲۰۱۴	[۲۲]
الگوریتم جستجوی فاخته گسسته	۲۰۱۵	[۲۳]
اعمال اصلاحاتی در استراتژی مهاجرت و شعاع تخم‌گذاری الگوریتم فاخته	۲۰۲۰	[۱۵]

شعاع تخم گذاری بزرگ تر و دامنه جستجوی بیش تری اختصاص داده می شود.

۲-۳- ایجاد ارتباط بین گام مهاجرت و شایستگی

در الگوریتم فاخته، مهاجرت فاخته‌ها تحت تاثیر دو پارامتر φ و λ انجام می شود. در شکل پایه الگوریتم، مقدار λ به صورت تصادفی با توزیع یکنواخت تعیین می گردد. این روش می تواند منجر به مهاجرت های غیر بهینه شود، زیرا کیفیت زیستگاه ها در فرآیند تعیین طول گام مهاجرت لحاظ نمی گردد. در روش پیشنهادی این پارامتر به صورت غیر تصادفی و بر اساس کیفیت زیستگاه ها مقادری می شود. بر این اساس فاخته های برتر با طول گام کوچکتر و فاخته های بدتر با طول گام بیشتری به مهاجرت اقدام کنند. لذا رابطه (۷) برای تعیین λ مرتبط با هر فاخته استفاده می شود:

$$\lambda = 1 - \frac{Cuckoo's_Profit}{Max\ Profit} \quad (7)$$

۳-۳- اصلاح نحوه مهاجرت فاخته ها

پارامتر دیگری که در تعیین نحوه مهاجرت فاخته ها اثر گذار است، پارامتر φ یا انحراف جهت مهاجرت است. از آن جا که وجود پارامتر گام مهاجرت (λ) و اعمال آن بر روی تمام متغیرهای تصمیم، عملا خود به تغییر جهت مهاجرت و آن هم تصادفی و غیر قابل پیش بینی منتهی می شود. از این رو پارامتر φ در روش پیشنهادی حذف شده است.

۴- بررسی اصلاحات پیشنهادی در الگوریتم بهینه سازی فاخته

در این پژوهش، با هدف بررسی عملکرد الگوریتم فاخته قبل و بعد از اعمال تغییرات پیشنهادی، الگوریتم های مورد بررسی شامل نسخه پایه الگوریتم فاخته و نسخه های پیشنهادی اصلاح شده، با استفاده از زبان برنامه نویسی پایتون پیاده سازی شده اند. برای مقایسه و ارزیابی عملکرد الگوریتم ها، از توابع محک معروف شامل تابع راستریجین^{۱۳} و اسفیر^{۱۴} استفاده می شود.

تابع راستریجین یک تابع غیرخطی چند جمله ای است که به دلیل داشتن تعداد زیادی مینیمم محلی، به عنوان یکی از چالش برانگیزترین توابع برای الگوریتم های بهینه سازی

شناخته می شود. این ویژگی باعث می شود که توانایی الگوریتم در یافتن مینیمم سراسری در حضور مینیمم های محلی متعدد ارزیابی شود. تابع اسفیر یک تابع ساده و محدب است که مینیمم آن به وضوح مشخص و یکتاست. این تابع اغلب برای ارزیابی دقت و سرعت همگرایی الگوریتم در شرایط بدون پیچیدگی استفاده می شود، زیرا با ارائه یک فضای جستجوی ساده، امکان تمرکز بر عملکرد ذاتی الگوریتم فراهم می شود [۱۵].

انتخاب این دو تابع به دلیل تفاوت اساسی در پیچیدگی و ویژگی های آن ها صورت گرفته است، به طوری که عملکرد الگوریتم در مواجهه با فضاهای جستجوی ساده و پیچیده قابل ارزیابی باشد.

در ادامه ابتدا تاثیر اعمال هر کدام از اصلاحات پیشنهادی به صورت جداگانه و سپس تاثیر اعمال همزمان تغییرات بر نتایج حاصل از الگوریتم بررسی می شود. پس از اعمال هر کدام از تغییرات یاد شده، الگوریتم حداقل ۳۰ بار بر روی هر کدام از توابع محک اجرا می گردد و معنادار بودن تغییرات حاصل شده در قالب معیارهای دقت حل و زمان همگرایی الگوریتم بررسی می گردد.

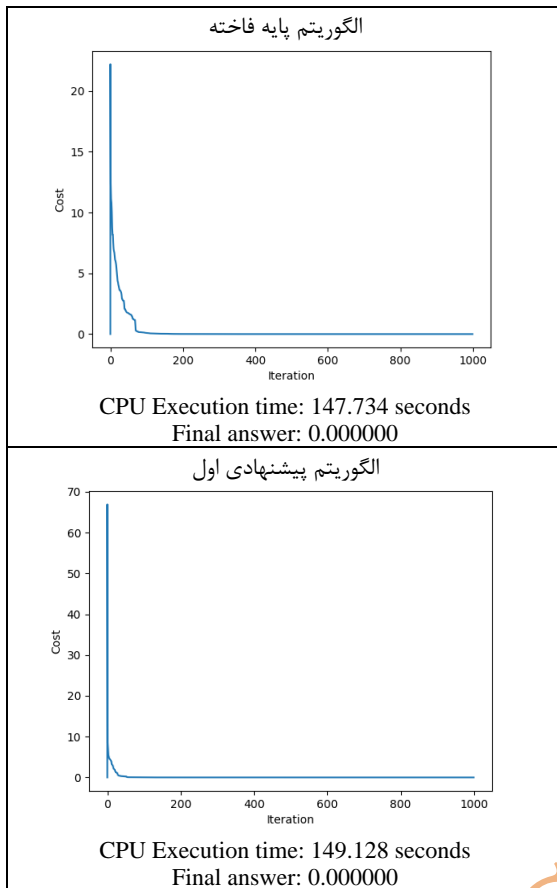
۱-۴- الگوریتم پیشنهادی اول

در الگوریتم پیشنهادی اول، فرآیند تخصیص تعداد تخم ها برای هر فاخته با رویکردی هوشمندانه و مبتنی بر شایستگی هر فاخته صورت می گیرد. بر اساس توضیحات ارائه شده در بخش ۳-۱، شایستگی هر فاخته که به کیفیت زیستگاه آن اشاره دارد، معیار اصلی در تعیین تعداد تخم هاست. این روش، برخلاف الگوریتم پایه که تعداد تخم ها را به صورت تصادفی یا بدون توجه به شایستگی زیستگاه تخصیص می دهد، منجر به بهره برداری مؤثرتر از فاخته های برتر می شود.

به منظور اعمال این الگوریتم بر روی تابع محک اسفیر، ابتدا شرایط اولیه مسئله به دقت تعریف شده و تنظیمات پارامترهای کلیدی الگوریتم مشخص گردید. این تنظیمات به طور کامل در جدول (۲) ارائه شده است.

¹³ Rastrigin

¹⁴ Sphere



شکل ۴- مقایسه نتیجه اجرای الگوریتم پیشنهادی اول و الگوریتم پایه بر روی تابع محک اسفیر

جدول ۳- نتایج حاصل از سی بار اجرای الگوریتم پیشنهادی اول و الگوریتم پایه بر روی تابع محک اسفیر

شماره بار اجرا	زمان اجرا		خروجی نهایی	
	الگوریتم پیشنهادی پایه	الگوریتم پیشنهادی پایه	الگوریتم پیشنهادی پایه	الگوریتم پیشنهادی پایه
۱	۷۸.۶۰	۹۵.۶۱	۰.۰۰۰۲۱	۰.۰۰۰۶۹
۲	۹۱.۶۱	۵۵.۵۹	۰.۰۰۰۱۷	۰.۰۰۰۰۷
۳	۲۱.۶۰	۲۴.۶۰	۰.۰۰۰۲۴	۰.۰۰۰۰۷
۴	۲۴.۶۱	۶۵.۶۱	۰.۰۰۰۱۷	۰.۰۰۰۰۷
۵	۷۹.۶۰	۰۹.۶۱	۰.۰۰۰۲۵	۰.۰۰۰۰۷
۶	۰۴.۶۲	۹۱.۵۹	۰.۰۰۰۰۲	۰.۰۰۰۰۷
۷	۳۹.۶۰	۴۲.۶۰	۰.۰۰۰۲۳	۰.۰۰۰۴۵
۸	۷۷.۶۲	۵۲.۶۱	۰.۰۰۰۰۲	۰.۰۰۰۰۷
۹	۴۹.۶۰	۷۴.۵۹	۰.۰۰۰۱۸	۰.۰۰۰۱۱۹
۱۰	۲۰.۶۲	۷۴.۵۹	۰.۰۰۰۲۱	۰.۰۰۰۰۸
۱۱	۹۹.۶۱	۵۴.۵۹	۰.۰۰۰۰۲	۰.۰۰۰۱۰۱
۱۲	۱۴.۶۲	۰۵.۶۰	۰.۰۰۰۰۲	۰.۰۰۰۰۲۱

جدول ۲- پارامترهای مساله و الگوریتم

عنوان	مقدار
تعداد متغیرها ۳۰ عدد	$N_{var} = 30$
حد پایین متغیرها	$Var_{Low} = -10$
حد بالای متغیرها	$Var_{Hi} = +10$
تعداد فاخته اولیه ۵۰ عدد	$NumCuckoos = 50$
حداقل تخم‌های هر فاخته ۳ عدد	$MinNumberOfEggs = 3$
حداکثر تخم‌های هر فاخته ۸ عدد	$MaxNumberOfEggs = 8$
تعداد تکرارهای الگوریتم ۱۰۰۰ بار	$MaxIter = 1000$
تعداد خوشه‌ها ۵ عدد	$K - ClusterNum = 5$
ضریب حرکت به سمت جواب بهینه ۰.۵	$MotionCoeff(F) = 0.5$
حداکثر تعداد فاخته‌ها در هر تکرار عدد	$MaxNumberOfCuckoos = 80$
ضریب شعاع تخم گذاری ^{۱۵}	$RadiusCoeff(\alpha) = 5$

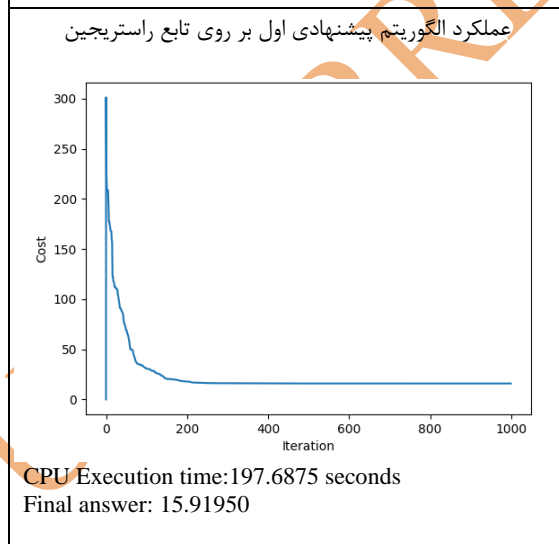
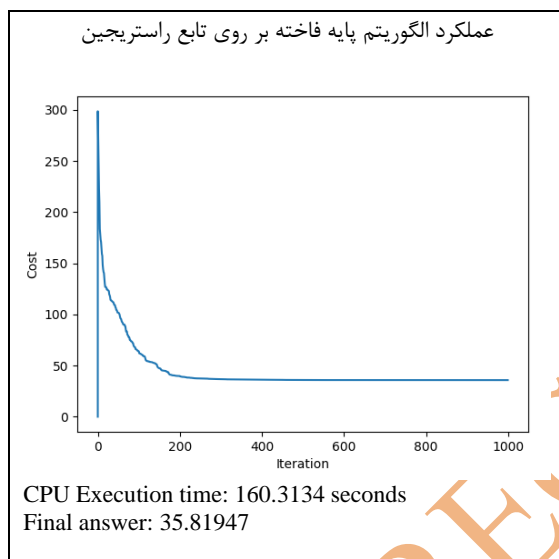
نتیجه اجرای الگوریتم پایه و الگوریتم اصلاح شده بر روی تابع محک اسفیر در شکل (۴) نمایش داده شده است. همان‌طور که در این شکل مشاهده می‌شود، الگوریتم اصلاح شده به دلیل بهبودهای اعمال شده، سرعت همگرایی بهتری در مقایسه با الگوریتم پایه از خود نشان می‌دهد. این در حالی است که بار محاسباتی اضافی ناشی از اصلاحات، از نظر زمانی ناچیز می‌باشد.

با توجه به همگرایی هر دو الگوریتم به مقدار صفر در تکرارهای بالا، تعداد تکرارهای الگوریتم از ۱۰۰۰ به ۴۰۰ کاهش یافت تا بازدهی بیشتری در اجرای الگوریتم‌ها حاصل شود. نتایج به دست آمده از سی بار اجرای هر دو الگوریتم در جدول (۳) ارائه شده است. همان‌طور که ملاحظه می‌شود در تمامی تکرارها، الگوریتم پیشنهادی عملکرد بهتری نسبت به الگوریتم پایه دارد. در مرحله بعد معنادار بودن این بهبودها از طریق آزمون آماری میانگین مورد بررسی قرار می‌گیرد.

^{۱۵} این پارامتر در الگوریتم فاخته پایه اعمال شده است.

• اعمال الگوریتم اصلاح شده بر روی تابع راستریجین

مشابه با حالت قبل شرایط اولیه مساله و پارامترهای اجرای الگوریتم مطابق جدول (۲) تعریف گردید. نتایج حاصل از اجرای الگوریتم بر روی تابع راستریجین در ادامه آمده است. همانطور که در شکل (۵) مشخص است، الگوریتم اصلاح شده پیشنهادی از نظر همگرایی و جواب بهینه، عملکرد بسیار بهتری نسبت به الگوریتم پایه در تابع محک راستریجین از خود نشان می‌دهد. از نظر بارمحاسباتی و زمان اجرا، الگوریتم پیشنهادی در ۱۰۰۰ تکرار حدود ۳۷ ثانیه بیشتر زمان می‌برد.



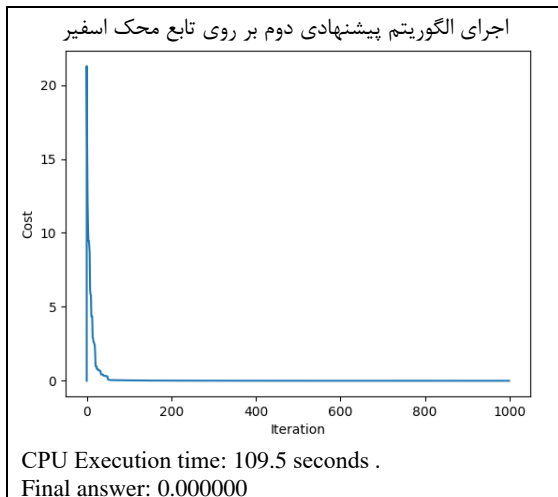
شکل ۵- مقایسه نتیجه اجرای الگوریتم پیشنهادی اول و الگوریتم پایه بر روی تابع محک راستریجین

شماره آماره	زمان اجرا		خروجی نهایی	
	الگوریتم پایه	الگوریتم پیشنهادی	الگوریتم پایه	الگوریتم پیشنهادی
۱۳	۲۲.۶۰	۶۸.۶۱	۰.۰۰۰۹۶	۰.۰۰۰۲۱
۱۴	۲۱.۶۰	۸۳.۶۱	۰.۰۰۰۰۵	۰.۰۰۰۰۲
۱۵	۴۲.۶۰	۹۹.۶۱	۰.۰۰۰۰۷	۰.۰۰۰۰۲
۱۶	۷۱.۵۹	۴۰.۶۲	۰.۰۰۰۰۷۴	۰.۰۰۰۰۱۹
۱۷	۸۰.۶۰	۸۲.۶۰	۰.۰۰۰۱۱۶	۰.۰۰۰۰۲۳
۱۸	۲۴.۶۰	۰۹.۶۲	۰.۰۰۰۰۷۳	۰.۰۰۰۰۰۲
۱۹	۱۴.۵۹	۰۱.۶۱	۰.۰۰۰۰۰۷	۰.۰۰۰۰۰۲
۲۰	۳۸.۶۱	۷۷.۶۰	۰.۰۰۰۰۰۷	۰.۰۰۰۰۰۲
۲۱	۵۱.۶۱	۹۷.۶۱	۰.۰۰۰۰۰۴۶	۰.۰۰۰۰۰۰۲
۲۲	۰۸.۶۱	۷۰.۶۱	۰.۰۰۰۰۰۰۷	۰.۰۰۰۰۰۰۲۵
۲۳	۲۲.۵۹	۱۸.۶۲	۰.۰۰۰۰۰۰۷۹	۰.۰۰۰۰۰۰۰۲
۲۴	۷۱.۵۹	۱۲.۶۲	۰.۰۰۰۰۰۰۷۲	۰.۰۰۰۰۰۰۰۲۱
۲۵	۶۴.۵۹	۷۹.۶۲	۰.۰۰۰۰۰۰۴۸	۰.۰۰۰۰۰۰۰۰۲
۲۶	۴۷.۶۰	۹۵.۶۰	۰.۰۰۰۰۰۱۰۴	۰.۰۰۰۰۰۰۰۰۲۱
۲۷	۶۴.۶۰	۴۸.۶۳	۰.۰۰۰۰۰۰۰۷	۰.۰۰۰۰۰۰۰۰۲۲
۲۸	۷۵.۶۰	۳۱.۶۲	۰.۰۰۰۰۰۰۰۷	۰.۰۰۰۰۰۰۰۰۱۸
۲۹	۹۸.۶۰	۳۸.۶۱	۰.۰۰۰۰۰۰۰۷	۰.۰۰۰۰۰۰۰۰۱۹
۳۰	۵۵.۶۰	۸۶.۶۱	۰.۰۰۰۰۰۰۰۴۵	۰.۰۰۰۰۰۰۰۰۲۲
میانگین	۴۰.۶۰	۶۸.۶۱	۰.۰۰۰۰۰۰۰۷۲	۰.۰۰۰۰۰۰۰۰۲۱

نتایج آزمون آماری در جدول (۴) نشان می‌دهد که بهبود پاسخ‌ها در الگوریتم اصلاح شده نسبت به الگوریتم پایه از نظر آماری کاملاً معنادار است. همچنین، هرچند افزایش زمان محاسباتی در الگوریتم اصلاح شده در مقایسه با الگوریتم پایه از نظر آماری معنادار می‌باشد، این افزایش در مقایسه با بهبود کارایی و دقت الگوریتم ناچیز تلقی می‌شود. با این حال، به دلیل محدودیت فضای مقاله، جزئیات مربوط به نتایج آزمون معنادار بودن تفاوت زمان اجرای الگوریتم‌ها ارائه نشده است. این تحلیل نشان‌دهنده اثربخشی اصلاحات پیشنهادی در بهبود سرعت همگرایی و دقت الگوریتم با حفظ زمان محاسباتی در محدوده قابل قبول است.

جدول ۴- آزمون معنادار بودن بهبود حاصل از الگوریتم

p-Value	مقدار آماره	میانگین پاسخ‌ها	نتایج الگوریتم پایه
۰	۱۱.۱۷۴۵۶۲	۰.۰۰۰۰۶۴۶	نتایج الگوریتم پیشنهادی اول
		۰.۰۰۰۰۰۲	



شکل ۶- اجرای الگوریتم پیشنهادی دوم بر روی تابع اسفیر

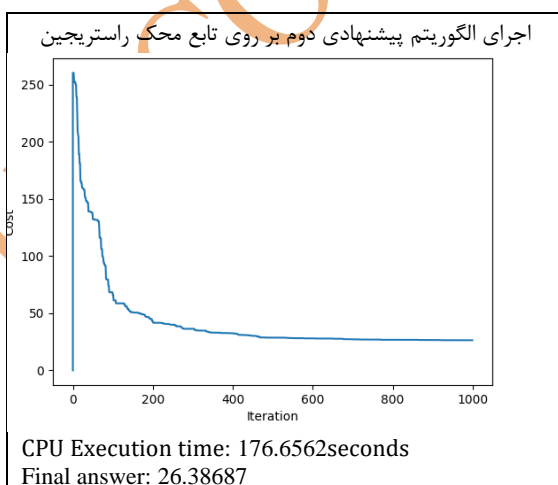
با توجه به همگرایی سریع هر دو الگوریتم به مینیم مطلق تابع اسفیر، جهت مقایسه عملکرد و انجام آزمون معنادار بودن تفاوت عملکرد، تعداد دفعات تکرار الگوریتم در هر بار اجرا به ۲۰۰ بار کاهش داده شد و محاسبات آماری بر مبنای ۳۰ بار اجرای برنامه انجام گرفت.

جدول ۵- آزمون معنادار بودن بهبود حاصل از الگوریتم

پیشنهادی دوم بر روی تابع محک اسفیر

p-Value	مقدار آماره	میانگین پاسخها	
.	۳۹.۰۱۱۶۲	۰.۰۰۰۵۳۸	نتایج الگوریتم پایه
		۰.۰۰۰۰	نتایج الگوریتم پیشنهادی دوم

همانطور که در جدول (۶) نشان داده شده است، بهبود حاصل شده از نظر آماری کاملاً معنادار می‌باشد.



شکل ۷- اجرای الگوریتم پیشنهادی دوم بر تابع راستریچین

در جدول (۵) آزمون معنادار بودن بهبود حاصل از الگوریتم پیشنهادی اول در مقایسه با الگوریتم پایه بر اساس نتایج به دست آمده بر روی تابع راستریچین نشان داده شده است. همانطور که نتایج ارائه شده در جداول فوق نشان می‌دهد الگوریتم پیشنهادی بر روی هر دو تابع محک مورد بررسی عملکرد بسیار مناسبی نسبت به الگوریتم پایه فاخته ارائه می‌دهد.

جدول ۴: آزمون معنادار بودن بهبود حاصل از الگوریتم

پیشنهادی اول بر روی تابع محک راستریچین

p-Value	مقدار آماره	میانگین پاسخها	
.	۳۳.۰۲۱۸۹	۳۷.۶۸۷۲۹۴	نتایج الگوریتم پایه
		۱۶.۴۴۷۰۷۲	نتایج الگوریتم پیشنهادی اول

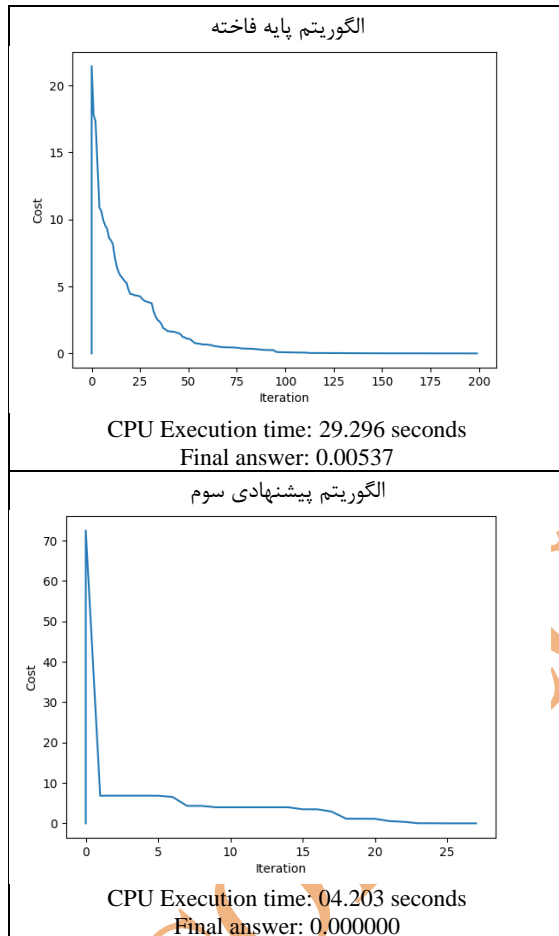
۴-۲- الگوریتم پیشنهادی دوم: اصلاح گام مهاجرت و نحوه مهاجرت فاخته‌ها

همانطور که در بخش قبل اشاره شد، مهاجرت فاخته‌ها تحت تاثیر دو پارامتر φ و λ قرار دارد. در روش پیشنهاد شده پارامتر λ به صورت غیر تصادفی و بر اساس کیفیت زیستگاه‌ها تعیین شده و پارامتر φ یا انحراف جهت مهاجرت که در الگوریتم پایه فاخته به آن اشاره شده است حذف می‌گردد. لذا بر اساس رابطه ۷، فاخته‌های برتر با طول گام کوچکتر و فاخته‌های بدتر با طول گام بیشتری مهاجرت می‌کنند.

در این مرحله، شرایط اولیه مساله و پارامترهای اجرای الگوریتم مطابق جدول (۲) که در بخش قبل ارائه شد تعریف گردید. نتیجه اجرای الگوریتم اصلاح شده در این مرحله بر روی توابع محک مورد بررسی، در ادامه آمده است.

همانطور که در شکل (۶) مشخص است، الگوریتم پیشنهادی دوم سرعت همگرایی بهتری در محاسبه مینیم تابع اسفیر نسبت به نمودار عملکرد الگوریتم پایه که در بخش قبل ارائه شد، دارد. از نظر بارمحاسباتی و زمان اجرا نیز، الگوریتم پیشنهادی ۱۰۰۰ تکرار محاسباتی را حدود ۳۷ ثانیه سریعتر انجام داده است. همچنین نتایج نشان می‌دهد الگوریتم پیشنهادی دوم در پیدا کردن مینیم تابع اسفیر عملکرد بهتری نسبت به الگوریتم پیشنهادی اول دارد.

به الگوریتم پیشنهادی اول و دوم بر روی تابع اسفیر نتایج بهتری ارائه داده است. طبق نتایج به دست آمده خروجی الگوریتم فاخته با اعمال الگوریتم پیشنهادی سوم در ۲۰۰ تکرار، از حدود میانگین ۰.۰۰۵۳۸ به ۰.۰۰۰۰ بهبود یافته است. همچنین در شاخص زمان اجرا میانگین از حدود ۳۰ ثانیه به ۸ ثانیه بهبود حاصل شده است. این بهبودها از نظر آماری کاملاً معنادار می‌باشند.



شکل ۸- مقایسه عملکرد الگوریتم پیشنهادی سوم و الگوریتم پایه فاخته بر تابع محک اسفیر

• اعمال الگوریتم پیشنهادی سوم بر تابع راستریجین نتیجه یک بار اجرای الگوریتم پایه و الگوریتم اصلاح شده بر روی تابع محک راستریجین در شکل (۹) نشان داده شده است. همانطور که در شکل مشخص است، الگوریتم پیشنهادی سوم از نظر همگرایی و جواب بهینه، بر روی تابع راستریجین عملکرد بهتری نسبت به الگوریتم پایه و الگوریتم پیشنهاد شده دوم دارد. اما نسبت به الگوریتم پیشنهادی اول نتایج کمی ضعیف تر است. مقایسه نتایج حاصل از سی بار اجرای الگوریتم پیشنهادی سوم با الگوریتم

در مورد تابع محک راستریجین، الگوریتم اصلاح شده پیشنهادی از نظر همگرایی و جواب بهینه، عملکرد بهتری نسبت به الگوریتم پایه از خود نشان می‌دهد. از نظر بار محاسباتی و زمان اجرا الگوریتم پیشنهادی در ۱۰۰۰ تکرار حدود ۱۶ ثانیه نسبت به الگوریتم پایه زمان بیشتری نیاز دارد.

مطابق نتایج ارائه شده در جداول فوق، عملکرد الگوریتم فاخته با اصلاح پیشنهاد شده در حل تابع محک راستریجین نیز به بهبود نتیجه به طور میانگین از ۳۷.۶۸۷۲۹ به ۲۸.۳۳۹۷۱ منتهی شده است اما نسبت به الگوریتم پیشنهادی اول با میانگین ۱۶.۴۴۷۰۷۲ عملکرد ضعیف تری دارد. همچنین در شاخص زمان اجرا، میانگین از حدود ۱۶۱ ثانیه در الگوریتم پایه به ۱۷۷ ثانیه افزایش یافته است. طبق آزمون‌های انجام شده در جدول (۷)، این تغییرات نیز از نظر آماری معنادار می‌باشند.

جدول ۶- آزمون معنادار بودن بهبود حاصل از الگوریتم

پیشنهادی دوم بر روی تابع محک راستریجین

p-Value	مقدار آماره	میانگین پاسخ‌ها	راستریجین
۰	۲۹.۷۲۱۲	۳۷.۶۸۷۲۹۴	نتایج الگوریتم پایه
		۲۸.۳۳۹۷۱	نتایج الگوریتم پیشنهادی دوم

۳-۴ الگوریتم پیشنهادی سوم: اعمال همزمان اصلاحات پیشنهادی

در این بخش اصلاحات پیشنهادی که در دو بخش قبل بصورت مجزا مورد بررسی قرار گرفت به صورت همزمان در الگوریتم فاخته اعمال می‌گردد و این الگوریتم را الگوریتم پیشنهادی سوم می‌نامیم. نتایج حاصل از انجام این اصلاحات نیز در قالب بهینه سازی توابع محک مورد بررسی قرار می‌گیرد. برای اعمال الگوریتم اصلاح شده در این مرحله، شرایط اولیه مساله و پارامترهای اجرای الگوریتم مطابق جدول (۲) تعریف گردید.

• اعمال الگوریتم پیشنهادی سوم بر روی تابع اسفیر نتیجه یک بار اجرای الگوریتم اصلاح شده سوم بر روی تابع محک اسفیر در شکل (۸) آمده است. به دلیل سرعت زیاد همگرایی در این الگوریتم، اجرای الگوریتم در ۲۵ تکرار به جواب بهینه سراسری رسیده است. این نتیجه نشان می‌دهد الگوریتم پیشنهادی سوم، نسبت

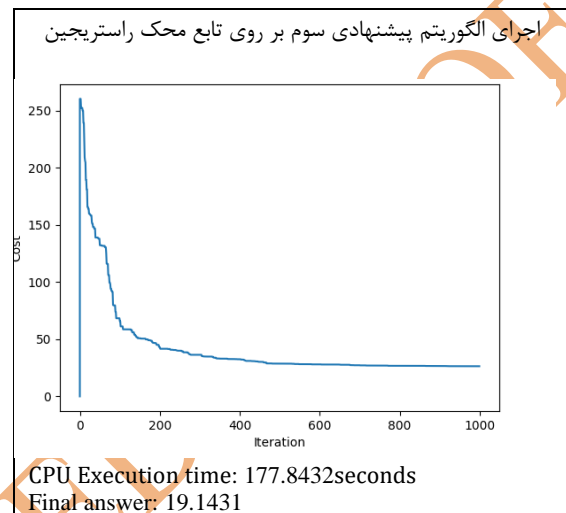
شایستگی بالاتر و بالعکس شود، که این امر می‌تواند سرعت رسیدن به جواب بهینه را کاهش دهد. در الگوریتم پیشنهادی اول، تخصیص تخم‌ها به صورت هوشمندانه‌تر و بر اساس نسبت سود هر فاخته به بهترین جواب انجام می‌شود. نتایج نشان می‌دهد که این روش بهبود قابل توجهی در عملکرد الگوریتم فاخته بر روی هر دو تابع محک اسفیر و راستریجین نسبت به الگوریتم پایه داشته است.

در الگوریتم پیشنهادی دوم، مهاجرت فاخته‌ها با استفاده از دو پارامتر λ و ϕ بهبود یافته است. در روش پیشنهادی، پارامتر λ به صورت غیرتصادفی و بر اساس کیفیت زیستگاه‌ها تعیین شده و پارامتر ϕ یا انحراف جهت مهاجرت حذف شده است. طبق این رویکرد، فاخته‌های برتر با طول گام کوچکتر و فاخته‌های ضعیف‌تر با طول گام بیشتری مهاجرت می‌کنند. نتایج حاکی از آن است که الگوریتم پیشنهادی دوم سرعت همگرایی بهتری در محاسبه مینیمم تابع اسفیر نسبت به الگوریتم پایه دارد و همچنین در حل تابع محک راستریجین عملکرد بهتری از خود نشان داده است، هرچند نسبت به الگوریتم پیشنهادی اول کمی ضعیف‌تر عمل کرده است.

در نهایت، الگوریتم پیشنهادی سوم که شامل ترکیبی از اصلاحات الگوریتم‌های پیشنهادی اول و دوم است، بررسی شد. این الگوریتم با بهبودهای قابل توجه در سرعت همگرایی و دقت، گزینه مناسبی برای حل مسائل بهینه‌سازی مختلف محسوب می‌شود. نتایج آزمایشات نشان می‌دهد که الگوریتم پیشنهادی سوم در حل توابع محک اسفیر و راستریجین عملکرد بهتری نسبت به الگوریتم پایه داشته و این بهبودها از نظر آماری معنادار بوده‌اند.

این پژوهش تأیید می‌کند که تلاش برای بهبود و توسعه الگوریتم‌های فراابتکاری می‌تواند به بهبودهای قابل ملاحظه‌ای در عملکرد آنها منجر شود و آنها را به ابزارهای کارآمدتری برای حل مسائل بهینه‌سازی تبدیل کند.

فاخته بر روی تابع راستریجین نشان می‌دهد به طور میانگین مقدار تابع از 37.68729 به 19.36495 بهبود یافته است. همچنین میانگین زمان اجرا از حدود 161 ثانیه به 177 ثانیه افزایش یافته است که نسبت به الگوریتم پیشنهادی اول کمتر می‌باشد. طبق آزمون‌های انجام شده، این تغییرات از نظر آماری معنادار می‌باشند.



شکل ۹- اجرای الگوریتم پیشنهادی سوم بر روی تابع محک راستریجین

۵- جمع بندی و نتیجه گیری

این پژوهش با هدف ارتقای دقت و سرعت همگرایی الگوریتم استاندارد فاخته، مجموعه‌ای از اصلاحات را مورد بررسی قرار داده است. این اصلاحات شامل بهبود روش تخم‌گذاری بر اساس میزان شایستگی هر فاخته، اصلاح شعاع همگرایی، و بازتعریف عملگر پرواز به سمت نقطه بهینه بوده است.

در الگوریتم پیشنهادی اول، تعداد تخم‌های هر فاخته بر اساس شایستگی آن تعیین می‌شود. در الگوریتم اصلی فاخته، تعداد تخم‌ها به صورت تصادفی و با توزیع یکنواخت در بازه‌ای مشخص تعیین می‌گردد. این فرایند ممکن است منجر به تخصیص تعداد تخم کمتر به فاخته‌های با

مراجع

- [1] Joshi, A. S., Kulkarni, O., Kakandikar, G. M., & Nandedkar, V. M. (2017). Cuckoo search optimization-a review. *Materials today: proceedings*, 4(8), 7262–7269. DOI:10.1016/j.matpr.2017.07.055
- [2] Rajabioun, R. (2011). Cuckoo optimization algorithm. *Applied soft computing*, 11(8), 5508–5518. DOI: 10.1016/j.asoc.2011.05.008

- [3] Walton, S., Hassan, O., Morgan, K., & Brown, M. R. (2011). Modified cuckoo search: a new gradient free optimisation algorithm. *Chaos, solitons & fractals*, 44(9), 710–718. DOI:10.1016/j.chaos.2011.06.004
- [4] Hamidzadeh, J., & others. (2021). Feature selection by using chaotic cuckoo optimization algorithm with levy flight, opposition-based learning and disruption operator. *Soft computing*, 25(4), 2911–2933. DOI:10.1007/s00500-020-05349-x
- [5] Hakim pour, F., Talat Ahary, S., & Ranjbar, A. (2017). The Assessment and Comparison of a Genetic Algorithm, Simulated Annealing and Cuckoo Optimization Algorithm for Optimization of the Facility Location under Competitive Conditions (Case Study: Banks). *Journal of modeling in engineering*, 15(48), 231–246(in Persian). DOI: 10.22075/jme.2017.2447
- [6] hazrati moghim, zahra, kheyroddin, ali, Naderpour, H., & Rahmani, H. (2018). Comparing the optimization of prestressed concrete bridge deck according to the AASHTO LRFD and AASHTO standard procedures using Cuckoo algorithm. *Journal of modeling in engineering*, 16(55), 209–219(in Persian). DOI: 10.22075/jme.2018.12832.1259
- [7] Armaghani, S., & Amjady, N. (2014). Multi-objective Multi-area Environmental and Economic Dispatch Using Cuckoo Optimization Algorithm. *Journal of modeling in engineering*, 12(37), 89–104(in Persian). DOI: 10.22075/jme.2017.1674
- [8] Yaqoob, A., Verma, N. K., & Aziz, R. M. (2024). Optimizing gene selection and cancer classification with hybrid sine cosine and cuckoo search algorithm. *Journal of medical systems*, 48(1), 10. DOI: 10.1007/s10916-023-02031-1
- [9] Zhu, C., Li, G., Luis, N. V. J., Dong, W., & Wang, L. (2024). Optimization of RF to alloy elastic modulus prediction based on cuckoo algorithm. *Computational materials science*, 231, 112515. DOI:10.1016/j.commatsci.2023.112515
- [10] Huang, S., Lin, N., Wang, Z., Zhang, Z., Wen, S., Zhao, Y., & Li, Q. (2024). A novel data-driven method for online parameter identification of an electrochemical model based on cuckoo search and particle swarm optimization algorithm. *Journal of power sources*, 601, 234261. DOI:10.1016/j.jpowsour.2024.234261
- [11] Shadkam, E. (2023). The problem of Resource Leveling in Multi-Project Mode by Cuckoo Optimization Algorithm. *Journal of civil and environmental engineering*, 53.1(110), 187–197(in persian). DOI: 10.22034/jcee.2021.44761.2015
- [12] Shehab, M., Khader, A. T., & Al-Betar, M. A. (2017). A survey on applications and variants of the cuckoo search algorithm. *Applied soft computing*, 61, 1041–1059. DOI: DOI:10.1016/j.asoc.2017.02.034
- [13] Żalik, K. R. (2008). An efficient k'-means clustering algorithm. *Pattern recognition letters*, 29(9), 1385–1391. DOI:10.1016/j.patrec.2008.02.014
- [14] Kahramanli, H. (2012). A modified cuckoo optimization algorithm for engineering optimization. *International journal of future computer and communication*, 1(2), 199. <http://dx.doi.org/10.7763/IJFCC.2012.V1.52>
- [15] Boveiri, H. R., & Elhoseny, M. (2020). A-COA: an adaptive cuckoo optimization algorithm for continuous and combinatorial optimization. *Neural computing and applications*, 32(3), 681–705. DOI: 10.1007/s00521-018-3928-9
- [16] Boushaki, S. I., Kamel, N., & Bendjeghaba, O. (2018). A new quantum chaotic cuckoo search algorithm for data clustering. *Expert systems with applications*, 96, 358–372. DOI:10.1016/j.eswa.2017.12.001
- [17] Ouaraab, A., Ahiod, B., & Yang, X.-S. (2014). Discrete cuckoo search algorithm for the travelling salesman problem. *Neural computing and applications*, 24, 1659–1669. DOI:10.1007/s00521-013-1402-2
- [18] Layeb, A. (2011). A novel quantum inspired cuckoo search for knapsack problems. *International journal of bio-inspired computation*, 3(5), 297–305. <http://dx.doi.org/10.1504/IJBIC.2011.042260>
- [19] Salesi, S., & Cosma, G. (2017). A novel extended binary cuckoo search algorithm for feature selection [presentation]. 2017 2nd international conference on knowledge engineering and applications, ickea 2017 (Vol. 2017-Janua, pp. 6–12). DOI: 10.1109/ICKEA.2017.8169893
- [20] Valian, E., Tavakoli, S., Mohanna, S., & Haghi, A. (2013). Improved cuckoo search for reliability optimization problems. *Computers & industrial engineering*, 64(1), 459–468. DOI:10.1016/j.cie.2012.07.011
- [21] Shishavan, S. T., & Gharehchopogh, F. S. (2022). An improved cuckoo search optimization algorithm with genetic algorithm for community detection in complex networks. *Multimedia tools and applications*, 81(18), 25205–25231. DOI:10.1007/s11042-022-12409-x

- [22] Naseri, K. (2014). A hybrid cuckoo-gravitation algorithm for cost-optimized QFD decision-making problem. *J math comput sci*, 9, 342–351. [DOI:10.22436/jmcs.09.04.10](https://doi.org/10.22436/jmcs.09.04.10)
- [23] Mahmoudi, S., & Lotfi, S. (2015). Modified cuckoo optimization algorithm (MCOA) to solve graph coloring problem. *Applied soft computing*, 33, 48–64. [DOI:10.1016/j.asoc.2015.04.020](https://doi.org/10.1016/j.asoc.2015.04.020)

UNCORRECTED PROOF