



Semnan University

Journal of Modeling in Engineering

Journal homepage: <https://modelling.semnan.ac.ir/>

ISSN: 2783-2538



Research Article

Solutions to Improve the Performance of Cuckoo Optimization Algorithm

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

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

PAPER INFO

Paper history:

Received: 2024-06-01

Revised: 2024-12-21

Accepted: 2025-01-01

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/10.22075/jme.2025.34272.2676>

© 2025 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:

H. Nahid Titkanloo , M. Nadershahi and M. Golmakani, "Solutions To Improve The Performance Of Cuckoo Optimization Algorithm," Journal of Modeling in Engineering, 23 82 (2025): 51-64, doi: 10.22075/jme.2025.34272.2676

راهکارهایی جهت بهبود عملکرد الگوریتم بهینه‌سازی فاخته

حسین ناهیدتیکانلو^{۱*}، محدثه نادرشاهی^۱، محمد گلمکانی^۱

اطلاعات مقاله	چکیده
دریافت مقاله: ۱۴۰۳/۰۳/۱۲ بازنگری مقاله: ۱۴۰۳/۱۰/۰۱ پذیرش مقاله: ۱۴۰۳/۱۰/۱۲	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.

واژگان کلیدی:

الگوریتم فاخته،
الگوریتم‌های فراابتکاری،
بهینه‌سازی،
توابع محک.

DOI: <https://doi.org/10.22075/jme.2025.34272.2676>

© 2025 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

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^۲ و الگوریتمهای تقریبی^۳ تقسیم‌بندی 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 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 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].

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

۱. گروه مهندسی صنایع، دانشگاه پیام نور، تهران، ایران

² Exact

³ Approximate

استناد به این مقاله:

حسین ناهیدتیکانلو، محدثه نادرشاهی و محمد گلمکانی، "راهکارهایی جهت بهبود عملکرد الگوریتم بهینه‌سازی فاخته"، مدل سازی در مهندسی، ۸۲ (۱۴۰۴): ۶۴-۵۱
doi: 10.22075/jme.2025.34272.2676

برای تخم گذاری انتخاب کند و در واقع دامنه مجاز تعیین متغیرهای پاسخ است. تعیین شعاع تخم گذاری در الگوریتم فاخته به تعداد تخم‌های هر فاخته مرتبط است و رابطه ای با کیفیت پاسخ به دست آمده ندارد. لذا به دنبال آن هستیم تا با ایجاد ارتباط بین شیوه تخم گذاری 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]. 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.

الگوریتم‌های تقریبی نیز به سه دسته الگوریتم‌های ابتکاری^۴ و فراابتکاری^۵ و فوق ابتکاری^۶ 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 بهینه سازی 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].

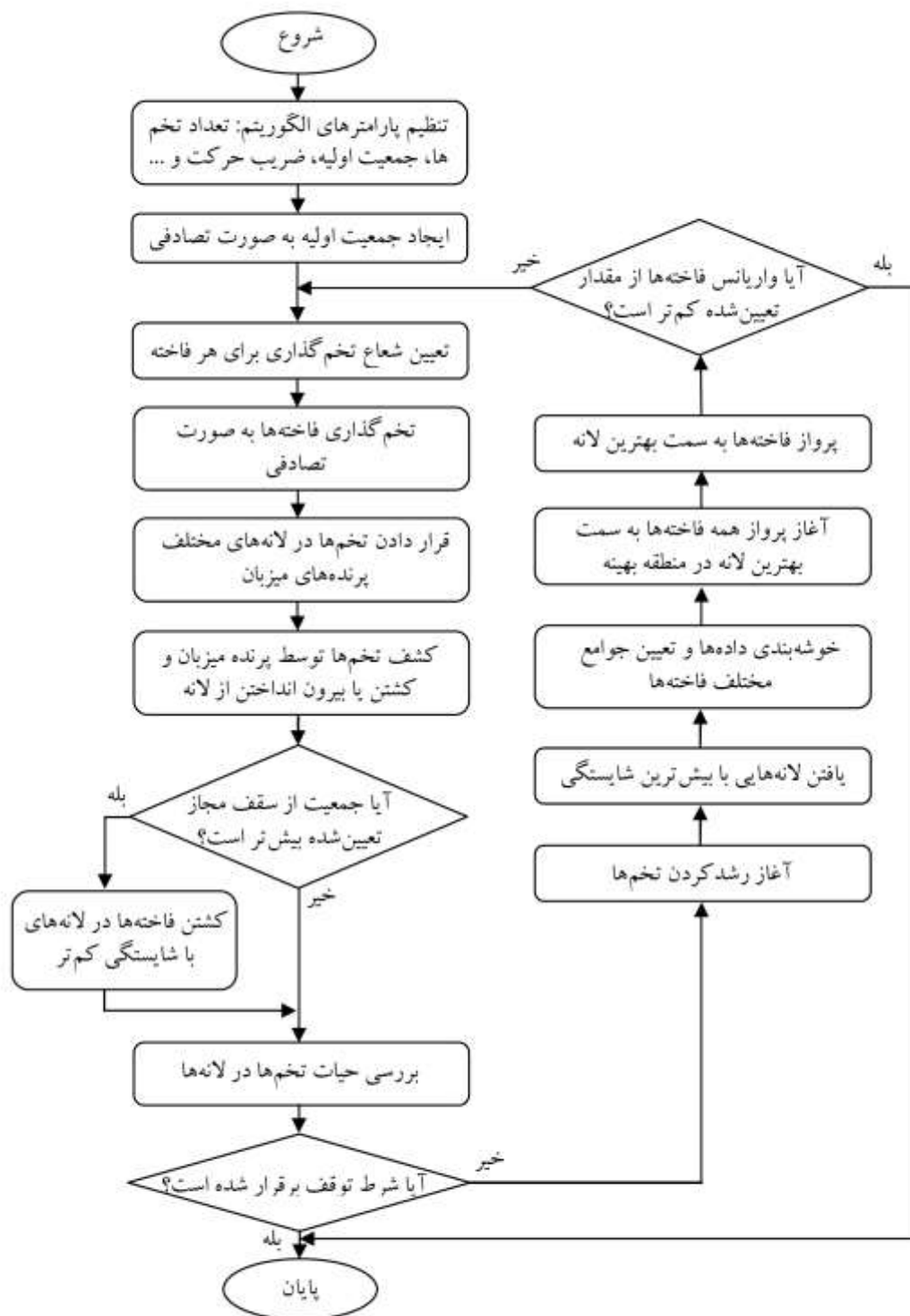
الگوریتم فاخته با وجود مزیت‌های نسبی، در مواردی مانند تغییر شعاع همگرایی به صورت صحیح و پرداختن به یک جستجوی محلی قوی بالاخص در مسائل با ابعاد بزرگ، به جمعیت بزرگی از فاخته‌ها نیاز دارد. همچنین عملکرد پرواز یا مهاجرت به سمت نقطه بهینه، در مسائل با ابعاد بالا، مبهم بوده و نیاز به اصلاح و بازتعریف دارد [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 تخم گذاری بر اساس میزان شایستگی هر فاخته، اصلاح شعاع همگرایی، بازتعریف عملکرد پرواز به سمت نقطه بهینه، تغییراتی است که در این این پژوهش مورد بررسی قرار می‌گیرد و جزئیات آن به شرح زیر می‌باشد:

- تغییر شعاع تخم‌گذاری^۷

شعاع تخم‌گذاری که در بخش ۱-۲ تشریح شده است، نشان‌دهنده محدوده مجازی است که هر فاخته می‌تواند

⁶ Hyper Heuristic
⁷ Egg Laying Radius

⁴ Heuristic
⁵ Meta-Heuristic

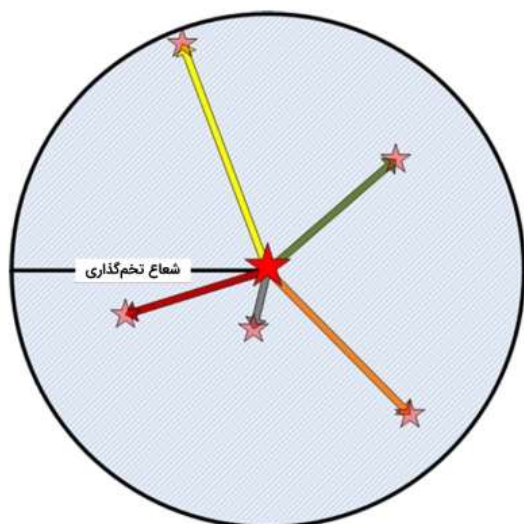


شکل ۱- فلوچارت الگوریتم بهینه سازی فاخته [۲]

رشد می یابند و به فاخته های بالغ تبدیل می شوند. اما تخم هایی که شباهت کمتری به تخم های درون لانه دارند، شناسایی شده و توسط پرند میزبان بیرون انداخته می شوند. این فرآیند تا زمانی ادامه پیدا می کند که تخم های رشد یافته ای که مناسب بودن لانه های آن منطقه را نشان

• مرحله (۱): شکل گیری جمعیت اولیه
الگوریتم با جمعیت اولیه ای از فاخته ها آغاز می شود. این فاخته ها شروع به تخم گذاری می کنند و تخم های خود را در لانه پرندگان میزبان قرار می دهند. برخی از این تخم ها که شباهت بیشتری به تخم های پرند میزبان دارند، فرصت

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_{Low} و var_{Hi} حد بالا و حد پایین متغیرهای تصمیم می‌باشند. در شکل (۲) مفهوم شعاع تخم‌گذاری به صورت شماتیک نشان داده شده است.



شکل ۲- مفهوم شعاع تخم‌گذاری [۲]

پس از پایان فرآیند تخم‌گذاری، تخم‌هایی که شباهت کمتری به تخم‌های پرند میزبان دارند، توسط پرند میزبان شناسایی شده و دور انداخته می‌شوند. بنابراین، پس از هر فرایند تخم‌گذاری $\rho\%$ 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 فاخته‌ای، سود میانگین^{۱۰} هر خوشه محاسبه می‌شود. بهترین زیستگاه موجود در گروه فاخته‌ای که بیشترین سود میانگین را دارد، به عنوان زیستگاه هدف و مقصد مهاجرتی

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"^۸ 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 فرایند 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:

$$habitate = [x_1, x_2, x_3, \dots, x_{N_{var}}] \quad (۱)$$

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:

$$Profit = f_b(x_1, x_2, x_3, \dots, x_{N_{var}}) \quad (۲)$$

• مرحله (۲): تخم‌گذاری فاخته‌ها

شعاع تخم‌گذاری نشان‌دهنده دامنه مجاز مقادیری است که هر فاخته می‌تواند برای متغیرها انتخاب کند. در یک مسئله بهینه‌سازی، شعاع تخم‌گذاری برای هر فاخته متناسب با تعداد کلی تخم‌ها، حد پایین و حد بالای متغیرها محاسبه می‌شود. محدوده تخم‌گذاری با ELR^9 نشان داده می‌شود و از رابطه زیر به دست می‌آید:

$$ELR = \alpha * \frac{n}{N} * (var_{Hi} - var_{Low}) \quad (۳)$$

¹⁰ Mean Profit

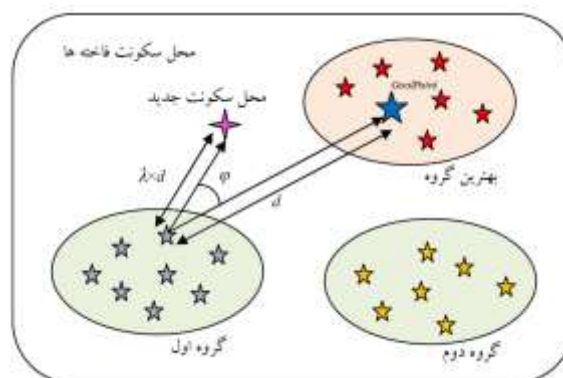
⁸ Habitat

⁹ Egg Laying Radius

دیگر گروه‌های فاخته‌ای انتخاب می‌شود.

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

هر فاخته فقط λ درصد مسیر را با درجه انحراف ϕ رادیان 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.

$$\begin{aligned} \lambda &\sim U(0,1) \\ \phi &\sim U(-\omega, \omega), (\omega = \frac{\pi}{6}) \end{aligned} \quad (۴)$$

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.

$$\begin{aligned} \text{habitate}_{new} &= \text{habitate}_{old} + \lambda \\ &\quad * \cos \phi \\ &\quad \otimes (\text{habitate}_{best} \\ &\quad - \text{habitate}_{old}) \end{aligned} \quad (۵)$$

• 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 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، فرایند اجرای الگوریتم بهینه‌سازی فاخته به پایان می‌رسد.

شبه کد الگوریتم بهینه‌سازی فاخته به شرح زیر می‌باشد:

• در ابتدا، جمعیت فاخته‌ها از طریق مقداردهی 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 پرند 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، فرایند اجرای الگوریتم بهینه‌سازی فاخته پایان می‌یابد؛ در غیر این صورت، الگوریتم به گام دو منتقل شده و الگوریتم دوباره اجرا می‌شود.

۲-۲- مطالعات انجام شده مرتبط با بهبود الگوریتم

بهینه‌سازی 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 تخم گذاری فاخته‌ها اصلاح شده است و یا در منبع [۱۵] نسخه تطبیقی از الگوریتم فاخته با اعمال اصلاحاتی در استراتژی مهاجرت و شعاع تخم گذاری 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 ^{۱۱}	۲۰۱۴	[۲۱]
الگوریتم ترکیبی فاخته- گرانشی	۲۰۱۴	[۲۲]
الگوریتم جستجوی فاخته گسسته	۲۰۱۵	[۲۳]
اعمال اصلاحاتی در استراتژی مهاجرت و شعاع تخم گذاری الگوریتم فاخته	۲۰۲۰	[۱۵]

۳- پیشنهادهایی جهت بهبود الگوریتم بهینه سازی فاخته

۳-۱- ایجاد ارتباط بین شایستگی و تعداد تخم گذاری هر فاخته

1

Genetic Cuckoo Optimization Algorithm

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.

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

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_{Ci} : 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 بهینه‌سازی 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، شعاع تخم‌گذاری بزرگ‌تر و دامنه جستجوی بیش‌تری اختصاص داده می‌شود.

۳-۲- ایجاد ارتباط بین گام مهاجرت و شایستگی

در الگوریتم فاخته، مهاجرت فاخته‌ها تحت تأثیر دو پارامتر φ و λ انجام می‌شود. در شکل پایه الگوریتم، مقدار λ به صورت تصادفی با توزیع یکنواخت تعیین می‌گردد. این روش می‌تواند منجر به مهاجرت‌های غیربهینه شود، زیرا کیفیت

تفاوت اساسی در پیچیدگی و ویژگی‌های آن‌ها 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 می‌شود. پس از اعمال هر کدام از تغییرات یاد شده، الگوریتم حداقل ۳۰ بار بر روی هر کدام از توابع محک اجرا می‌گردد و معنادار بودن تغییرات حاصل شده در قالب معیارهای دقت حل و زمان همگرایی الگوریتم بررسی می‌گردد.

۴-۱- الگوریتم پیشنهادی اول

در الگوریتم پیشنهادی اول، فرآیند تخصیص تعداد تخم‌ها برای هر فاخته با رویکردی هوشمندانه و مبتنی بر شایستگی هر فاخته صورت می‌گیرد. بر اساس توضیحات ارائه شده در بخش ۳-۱، شایستگی هر فاخته که به کیفیت زیستگاه آن اشاره دارد، معیار اصلی در تعیین تعداد تخم‌هاست. این روش، برخلاف الگوریتم پایه که تعداد تخم‌ها را به صورت تصادفی یا بدون توجه به شایستگی زیستگاه تخصیص می‌دهد، منجر به بهره‌برداری مؤثرتر از فاخته‌های برتر می‌شود.

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	$MinNymberOfEggs = 3$
حداکثر تخم‌های هر فاخته ۸ عدد	$MaxNymberOfEggs = 8$
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	$MaxNymberOfCuckoos = 80$
ضریب شعاع تخم گذاری ^{۱۴}	$RadiusCoeff(\alpha) = 5$

^{۱۴} این پارامتر در الگوریتم فاخته پایه اعمال شده است.

زیستگاه‌ها در فرآیند تعیین طول گام مهاجرت لحاظ نمی‌گردد. در روش پیشنهادی این پارامتر به صورت غیر تصادفی و بر اساس کیفیت زیستگاه‌ها مقداری می‌شود. بر این اساس فاخته‌های برتر با طول گام کوچکتر و فاخته‌های بدتر با طول گام بیشتری به مهاجرت اقدام کنند. لذا رابطه (۷) برای تعیین λ مرتبط با هر فاخته استفاده می‌شود:

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

۳-۳- اصلاح نحوه مهاجرت فاخته‌ها

پارامتر دیگری که در تعیین نحوه مهاجرت فاخته‌ها اثرگذار 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، عملاً خود به تغییر جهت مهاجرت و آن هم تصادفی و غیرقابل پیش‌بینی منتهی می‌شود. 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، الگوریتم‌های مورد بررسی شامل نسخه پایه الگوریتم فاخته و نسخه‌های پیشنهادی اصلاح شده، با استفاده از زبان برنامه‌نویسی پایتون پیاده‌سازی شده‌اند. برای مقایسه و ارزیابی عملکرد الگوریتم‌ها، از توابع محک معروف شامل تابع راستریجین^{۱۲} و اسفیر^{۱۳} استفاده می‌شود.

تابع راستریجین یک تابع غیرخطی چندجمله‌ای است که به دلیل داشتن تعداد زیادی مینیمم محلی، به عنوان یکی از چالش‌برانگیزترین توابع برای الگوریتم‌های بهینه‌سازی شناخته می‌شود. این ویژگی باعث می‌شود که توانایی الگوریتم در یافتن مینیمم سراسری در حضور مینیمم‌های 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 می‌شود، زیرا با ارائه یک فضای جستجوی ساده، امکان تمرکز بر عملکرد ذاتی الگوریتم فراهم می‌شود [۱۵]. انتخاب این دو تابع به دلیل

^{۱۲} Rastrigin

^{۱۳} Sphere

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

ردیف	زمان اجرا		خروجی نهایی	
	الگوریتم پایه	الگوریتم پیشنهادی	الگوریتم پایه	الگوریتم پیشنهادی
۱	۹۵.۶۱	۷۸.۶۰	۰.۰۰۰۶۹	۰.۰۰۰۲۱
۲	۵۵.۵۹	۹۱.۶۱	۰.۰۰۰۰۷	۰.۰۰۰۱۷
۳	۲۴.۶۰	۲۱.۶۰	۰.۰۰۰۰۷	۰.۰۰۰۲۴
۴	۶۵.۶۱	۲۴.۶۱	۰.۰۰۰۰۷	۰.۰۰۰۱۷
۵	۰۹.۶۱	۷۹.۶۰	۰.۰۰۰۰۷	۰.۰۰۰۲۵
۶	۹۱.۵۹	۰۴.۶۲	۰.۰۰۰۰۷	۰.۰۰۰۰۲
۷	۴۲.۶۰	۳۹.۶۰	۰.۰۰۰۰۴۵	۰.۰۰۰۲۳
۸	۵۲.۶۱	۷۷.۶۲	۰.۰۰۰۰۷	۰.۰۰۰۰۲
۹	۷۴.۵۹	۴۹.۶۰	۰.۰۰۰۱۱۹	۰.۰۰۰۱۸
۱۰	۷۴.۵۹	۲۰.۶۲	۰.۰۰۰۰۸	۰.۰۰۰۲۱
۱۱	۵۴.۵۹	۹۹.۶۱	۰.۰۰۰۱۰۱	۰.۰۰۰۰۲
۱۲	۰۵.۶۰	۱۴.۶۲	۰.۰۰۰۰۲۱	۰.۰۰۰۰۲
۱۳	۲۲.۶۰	۶۸.۶۱	۰.۰۰۰۰۹۶	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	۰.۰۰۰۱۹
۱۷	۸۰.۶۰	۸۲.۶۰	۰.۰۰۰۱۱۶	۰.۰۰۰۲۳
۱۸	۲۴.۶۰	۰۹.۶۲	۰.۰۰۰۰۷۳	۰.۰۰۰۰۲
۱۹	۱۴.۵۹	۰۱.۶۱	۰.۰۰۰۰۷	۰.۰۰۰۰۲
۲۰	۳۸.۶۱	۷۷.۶۰	۰.۰۰۰۰۷	۰.۰۰۰۰۲
۲۱	۵۱.۶۱	۹۷.۶۱	۰.۰۰۰۰۴۶	۰.۰۰۰۰۲
۲۲	۰۸.۶۱	۷۰.۶۱	۰.۰۰۰۰۷	۰.۰۰۰۲۵
۲۳	۲۲.۵۹	۱۸.۶۲	۰.۰۰۰۰۷۹	۰.۰۰۰۰۲
۲۴	۷۱.۵۹	۱۲.۶۲	۰.۰۰۰۰۷۲	۰.۰۰۰۲۱
۲۵	۶۴.۵۹	۷۹.۶۲	۰.۰۰۰۰۴۸	۰.۰۰۰۰۲
۲۶	۴۷.۶۰	۹۵.۶۰	۰.۰۰۰۱۰۴	۰.۰۰۰۲۱
۲۷	۶۴.۶۰	۴۸.۶۳	۰.۰۰۰۰۷	۰.۰۰۰۲۲
۲۸	۷۵.۶۰	۳۱.۶۲	۰.۰۰۰۰۷	۰.۰۰۰۱۸
۲۹	۹۸.۶۰	۳۸.۶۱	۰.۰۰۰۰۷	۰.۰۰۰۱۹
۳۰	۵۵.۶۰	۸۶.۶۱	۰.۰۰۰۰۴۵	۰.۰۰۰۲۲
میانگین	۴۰.۶۰	۶۸.۶۱	۰.۰۰۰۰۷۲	۰.۰۰۰۲۱

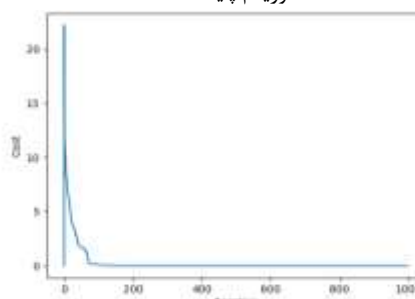
نتایج آزمون آماری در جدول ۴ نشان می‌دهد که بهبود پاسخ‌ها در الگوریتم اصلاح‌شده نسبت به الگوریتم پایه از نظر آماری کاملاً معنادار است. همچنین، هرچند افزایش زمان محاسباتی در الگوریتم اصلاح‌شده در مقایسه با الگوریتم پایه از نظر آماری معنادار می‌باشد، این افزایش در مقایسه با بهبود کارایی و دقت الگوریتم ناچیز تلقی می‌شود. با این حال، به دلیل محدودیت فضای مقاله، جزئیات مربوط

به 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 به‌طور کامل در جدول ۲ ارائه شده است.

نتیجه اجرای الگوریتم پایه و الگوریتم اصلاح‌شده بر روی تابع محک اسفیر در شکل (۴) نمایش داده شده است. همان‌طور که در این شکل مشاهده می‌شود، الگوریتم اصلاح‌شده به دلیل بهبودهای اعمال‌شده، 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 به‌دست‌آمده 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.

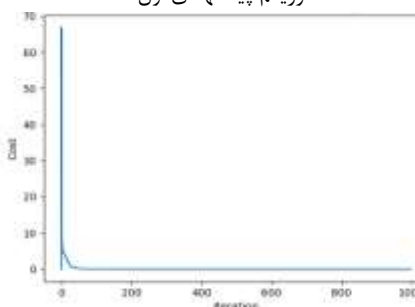
الگوریتم پایه فاخته



CPU Execution time: 147.734 seconds

Final answer: 0.000000

الگوریتم پیشنهادی اول

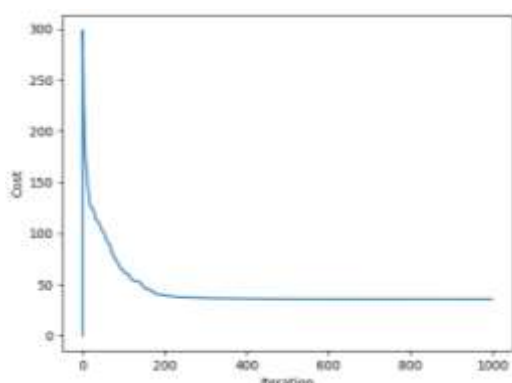


CPU Execution time: 149.128 seconds

Final answer: 0.000000

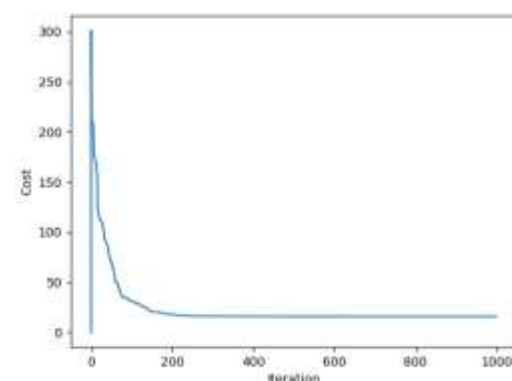
شکل ۴- مقایسه نتیجه اجرای الگوریتم پیشنهادی اول و الگوریتم پایه بر روی تابع محک اسفیر

عملکرد الگوریتم پایه فاخته بر روی تابع راستریجین



CPU Execution time: 160.3134 seconds
Final answer: 35.81947

عملکرد الگوریتم پیشنهادی اول بر روی تابع راستریجین



CPU Execution time: 197.6875 seconds
Final answer: 15.91950

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

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

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

پیشنهادی اول بر روی تابع محک اسفیر

p-Value	مقدار آماره	میانگین پاسخ‌ها	
۰	۱۱.۱۷۴۵۶۲	۰.۰۰۰۶۴۶	نتایج الگوریتم پایه
		۰.۰۰۰۰۲	نتایج الگوریتم پیشنهادی اول

• 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.

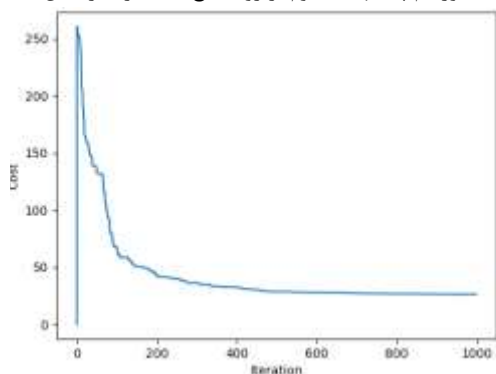
در جدول ۵ آزمون معنادار بودن بهبود حاصل از الگوریتم پیشنهادی اول در مقایسه با الگوریتم پایه بر اساس نتایج به دست آمده بر روی تابع راستریجین نشان داده شده است. همانطور که نتایج ارائه شده در جداول فوق نشان می‌دهد الگوریتم پیشنهادی بر روی هر دو تابع محک مورد بررسی عملکرد بسیار مناسبی نسبت به الگوریتم پایه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

جدول ۵- آزمون معنادار بودن بهبود حاصل از الگوریتم

پیشنهادی اول بر روی تابع محک راستریجین

p-Value	مقدار آماره	میانگین پاسخ‌ها	
۰	۳۳.۰۲۱۸۹	۳۷.۶۸۷۲۹۴	نتایج الگوریتم پایه
		۱۶.۴۴۷۰۷۲	نتایج الگوریتم پیشنهادی اول

اجرای الگوریتم پیشنهادی دوم بر روی تابع محک راستریجین



CPU Execution time: 176.6562seconds

Final answer: 26.38687

شکل ۷- اجرای الگوریتم پیشنهادی دوم بر روی تابع راستریجین

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 بارمحاسباتی 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 ۵- آزمون معنادار بودن بهبود حاصل از الگوریتم

پیشنهادی دوم بر روی تابع محک راستریجین

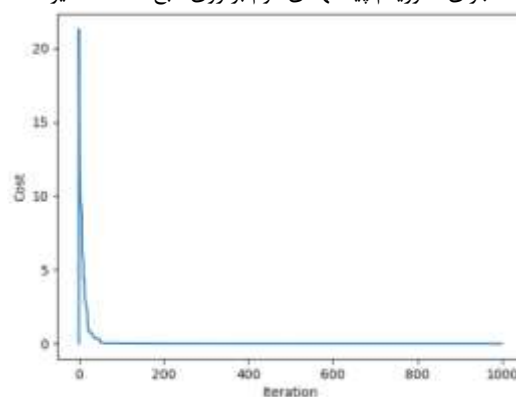
راستریجین	میانگین پاسخ‌ها	مقدار آماره	p-Value
نتایج الگوریتم پایه	۳۷.۶۸۷۲۹۴	۲۹.۷۲۱۲	۰
نتایج الگوریتم پیشنهادی دوم	۲۸.۳۳۹۷۱		

۴-۳- 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

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 مینیمم 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 مینیمم تابع اسفیر عملکرد بهتری نسبت به الگوریتم پیشنهادی اول دارد.

اجرای الگوریتم پیشنهادی دوم بر روی تابع محک اسفیر



CPU Execution time: 109.5 seconds .

Final answer: 0.000000

شکل ۶- اجرای الگوریتم پیشنهادی دوم بر روی تابع اسفیر

با توجه به همگرایی سریع هر دو الگوریتم به مینی مم 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 ۴- آزمون معنادار بودن بهبود حاصل از الگوریتم

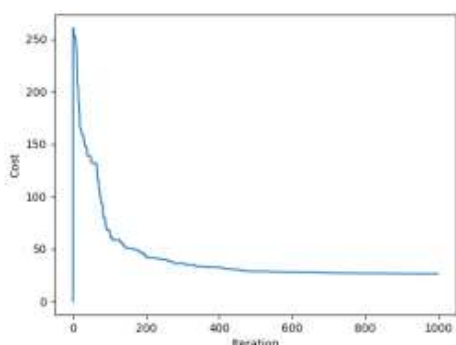
پیشنهادی دوم بر روی تابع محک اسفیر

پیشنهادی دوم	میانگین پاسخ‌ها	مقدار آماره	p-Value
نتایج الگوریتم پایه	۰.۰۰۵۳۸	۳۹.۰۱۱۶۲	۰
نتایج الگوریتم پیشنهادی دوم	۰.۰۰۰۰		

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

• 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 پیشنهادی 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 فاخته بر روی تابع راستریجین نشان می‌دهد به طور میانگین مقدار تابع از ۳۷.۶۸۷۲۹ به ۱۹.۳۶۴۹۵ بهبود یافته است. همچنین میانگین زمان اجرا از حدود ۱۶۱ ثانیه به ۱۷۷ ثانیه افزایش یافته است که نسبت به الگوریتم پیشنهادی اول کمتر می‌باشد. طبق آزمون‌های انجام شده، این تغییرات از نظر آماری معنادار می‌باشند.

اجرای الگوریتم پیشنهادی سوم بر روی تابع محک راستریجین



CPU Execution time: 177.8432seconds

Final answer: 19.1431

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

۵- جمع‌بندی و نتیجه‌گیری

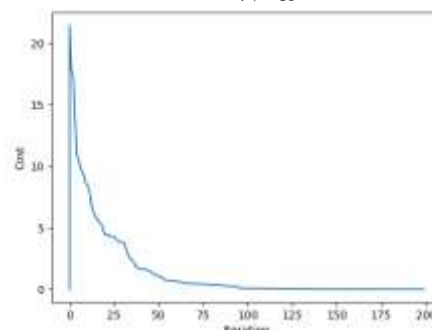
این پژوهش با هدف ارتقای دقت و سرعت همگرایی الگوریتم استاندارد فاخته، مجموعه‌ای از اصلاحات را مورد بررسی قرار داده است. این اصلاحات شامل بهبود روش تخم‌گذاری بر اساس میزان شایستگی هر فاخته، اصلاح شعاع همگرایی، و بازتعریف عملگر پرواز به سمت نقطه بهینه بوده است.

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

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 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.

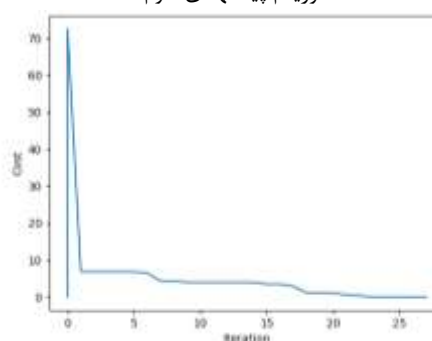
الگوریتم پایه فاخته



CPU Execution time: 29.296 seconds

Final answer: 0.00537

الگوریتم پیشنهادی سوم



CPU Execution time: 04.203 seconds

Final answer: 0.000000

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

الگوریتم‌های فراابتکاری می‌تواند به بهبودهای قابل ملاحظه‌ای در عملکرد آنها منجر شود و آنها را به ابزارهای کارآمدتری برای حل مسائل بهینه‌سازی تبدیل کند.

تقدیر و تشکر

نویسندگان بدین‌وسیله مراتب سپاس صمیمانه خود را از کلیه همکاران و داوران محترمی که با ارائه نظرات ارزشمند خود به بهبود کیفیت این پژوهش کمک کردند، ابراز می‌دارند.

تعارض منافع

نویسندگان اعلام می‌دارند که درخصوص انتشار این مقاله هیچ‌گونه تعارض منافع وجود ندارد.

تأییدیه اخلاقی

نویسندگان تأیید می‌کنند که محتوای این مقاله در هیچ مجله دیگری منتشر نشده و هم‌اکنون نیز در دست بررسی مجله دیگری قرار ندارد.

مشارکت‌های نویسندگان

حسین ناهید تیتکانلو: ایده‌پردازی، طراحی روش‌شناسی، مشارکت در کدنویسی الگوریتم، تحلیل داده‌ها، نظارت کلی بر پروژه، و نگارش نسخه نهایی مقاله. **محدثه نادرشاهی:** تحلیل داده‌ها، اجرای آزمایش‌ها، تهیه و تنظیم جداول و تصاویر، و مشارکت در بهبود نگارش متن. **محمد گلمکانی:** مرور پیشینه پژوهش، کدنویسی الگوریتم، اجرای آزمایش‌ها، و مشارکت در بهبود نگارش متن.

منابع مالی

هیچ‌گونه منبع مالی برای انجام این پژوهش دریافت نشده است.

منجر به تخصیص تعداد تخم کمتر به فاخته‌های با 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

مراجع

- [1] Joshi, Akshata S., Omkar Kulkarni, Ganesh M. Kakandikar, and Vilas M. Nandedkar. "Cuckoo search optimization-a review." *Materials Today: Proceedings* 4, no. 8 (2017): 7262-7269.
- [2] Rajabioun, Ramin. "Cuckoo optimization algorithm." *Applied Soft Computing* 11, no. 8 (2011): 5508-5518.
- [3] Walton, S., O. Hassan, K. Morgan, and M. R. Brown. "Modified cuckoo search: a new gradient free optimisation algorithm." *Chaos, Solitons & Fractals* 44, no. 9 (2011): 710-718.
- [4] Hamidzadeh, Javad. "Feature selection by using chaotic cuckoo optimization algorithm with levy flight, opposition-based learning and disruption operator." *Soft Computing-A Fusion of Foundations, Methodologies & Applications* 25, no. 4 (2021).

- [5] Hakim Pour, Farshad, Siamak Talat Ahary, and Abolfazl Ranjbar. "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, no. 48 (2017): 231-246. (in Persian)
- [6] Hazrati Moghim, Zahra, Ali Kheyroddin, Hossein Naderpour, and Hossein Rahmani. "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, no. 55 (2018): 209-219. (in Persian)
- [7] Armaghani, Saber, and Nima Amjady. "Multi-objective Multi-area Environmental and Economic Dispatch Using Cuckoo Optimization Algorithm." *Journal of Modeling in Engineering* 12, no. 37 (2014): 89-104. (in Persian)
- [8] Yaqoob, Abrar, Navneet Kumar Verma, and Rabia Musheer Aziz. "Optimizing gene selection and cancer classification with hybrid sine cosine and cuckoo search algorithm." *Journal of Medical Systems* 48, no. 1 (2024): 10.
- [9] Zhu, Chang-sheng, Guang-zhao Li, Naranjo Villota Jose Luis, Wen-jing Dong, and Li-jun Wang. "Optimization of RF to alloy elastic modulus prediction based on cuckoo algorithm." *Computational Materials Science* 231 (2024): 112515.
- [10] Huang, Shengxu, Ni Lin, Zhenpo Wang, Zhaosheng Zhang, Shuang Wen, Yue Zhao, and Qian Li. "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 (2024): 234261.
- [11] Shadkam, Elham. "The problem of Resource Leveling in Multi-Project Mode by Cuckoo Optimization Algorithm." *Journal of Civil and Environmental Engineering* 53, no. 110 (2023): 187-197.
- [12] Shehab, Mohammad, Ahamad Tajudin Khader, and Mohammed Azmi Al-Betar. "A survey on applications and variants of the cuckoo search algorithm." *Applied Soft Computing* 61 (2017): 1041-1059.
- [13] Žalik, Krista Rizman. "An efficient k'-means clustering algorithm." *Pattern Recognition Letters* 29, no. 9 (2008): 1385-1391.
- [14] Kahramanli, Humar. "A modified cuckoo optimization algorithm for engineering optimization." *International Journal of Future Computer and Communication* 1, no. 2 (2012): 199.
- [15] Boveiri, Hamid Reza, and Mohamed Elhoseny. "RETRACTED ARTICLE: A-COA: an adaptive cuckoo optimization algorithm for continuous and combinatorial optimization." *Neural computing and applications* 32, no. 3 (2020): 681-705.
- [16] Boushaki, Saida Ishak, Nadjat Kamel, and Omar Bendjehaba. "A new quantum chaotic cuckoo search algorithm for data clustering." *Expert Systems with Applications* 96 (2018): 358-372.
- [17] Ouaraab, Aziz, Belaïd Ahiod, and Xin-She Yang. "Discrete cuckoo search algorithm for the travelling salesman problem." *Neural Computing and Applications* 24, no. 7 (2014): 1659-1669.
- [18] Layeb, Abdesslem. "A novel quantum inspired cuckoo search for knapsack problems." *International Journal of bio-inspired Computation* 3, no. 5 (2011): 297-305.
- [19] Salesi, Sadegh, and Georgina Cosma. "A novel extended binary cuckoo search algorithm for feature selection." In *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)*, pp. 6-12. IEEE, 2017.
- [20] Valian, Ehsan, Saeed Tavakoli, Shahram Mohanna, and Atiyeh Haghi. "Improved cuckoo search for reliability optimization problems." *Computers & Industrial Engineering* 64, no. 1 (2013): 459-468.
- [21] Shishavan, Saeid Talebpour, and Farhad Soleimanian Gharehchopogh. "An improved cuckoo search optimization algorithm with genetic algorithm for community detection in complex networks." *Multimedia Tools and Applications* 81, no. 18 (2022): 25205-25231.
- [22] Naseri, K. "A hybrid cuckoo-gravitation algorithm for cost-optimized QFD decision-making problem." *Journal of Mathematics and Computer Science* 9, no. 4 (2014): 342-351.
- [23] Mahmoudi, Shadi, and Shahriar Lotfi. "Modified cuckoo optimization algorithm (MCOA) to solve graph coloring problem." *Applied Soft Computing* 33 (2015): 48-64.