



Semnan University

Journal of Modeling in Engineering

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

ISSN: 2783-2538



Research Article

Proposing an Improved Version of the Bat Algorithm

Davar Giveki ^{a,*}, Javad Ebrahimi ^b, Maryam Sarshar ^c

^a Department of Computer Engineering, Technical and Engineering Faculty of Malayer University, Malayer, Iran

^b Department of Electrical Engineering, Institute of Higher Education Afarinesh Alam Gostar Borujard, Borujard, Iran

^c Department of Computer Engineering, Institute of Higher Education Afarinesh Alam Gostar Borujard, Borujard, Iran

PAPER INFO

Paper history:

Received: 2024-02-03

Revised: 2024-04-28

Accepted: 2024-06-05

Keywords:

Optimization;
Swarm intelligence;
Meta-Heuristic algorithm;
Local optimum;
Bat algorithm.

ABSTRACT

The bat algorithm is an example of meta-heuristic algorithms from the collective swarm intelligence, which is based on the echolocation behavior of bats. This algorithm preserves the diversity of the solution by using a frequency tuning method that can quickly and efficiently shift from exploration to exploitation. Therefore, when a fast and accurate solution is needed, this algorithm becomes an efficient optimizer for any application. Although the bat algorithm has many practical benefits, it also has some disadvantages. One of these disadvantages that reduces its efficiency is being trapped in the local optimum. To solve the mentioned problem in this research, the position and speed of the initial population is updated in three ways with different formulas, this makes the final answer of the problem not trapped in the local optimum and diversity occurs in the population. In this article, the performance of the improved bat algorithm on 11 sample objective functions has been investigated and compared with other similar algorithms, and finally the results show the superiority and accuracy of this algorithm compared to similar samples.

DOI: <https://doi.org/10.22075/jme.2024.33011.2608>

© 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: Davood.giveki@gmail.com

How to cite this article:

Giveki, D., Ebrahimi, J., & Sarshar, M. (2024). Proposing an improved version of the bat algorithm. *Journal of Modeling in Engineering*, 22(79), 267-279. doi: 10.22075/jme.2024.33011.2608

معرفی نسخه بهبودیافته الگوریتم خفاش

داور گیوکی^{۱*}، جواد ابراهیمی^۲، مریم سرشار^۳

اطلاعات مقاله	چکیده
دریافت مقاله: ۱۴۰۲/۱۱/۱۴	الگوریتم خفاش، نمونه‌ای از الگوریتم‌های فراابتکاری از خانواده هوش جمعی است که براساس رفتار پژواکیایی خفاش بنا شده است. این الگوریتم تنوع راه حل را با استفاده از روش تنظیم فرکانس حفظ می‌کند که می‌تواند به سرعت و به صورت کارآمد از مرحله اکتشاف به بهره‌برداری تغییر مکان دهد. بنابراین، هنگامی که به یک راه حل سریع و دقیق نیاز باشد، این الگوریتم به یک بهینه‌ساز کارآمد برای هر برنامه کاربردی تبدیل می‌شود. الگوریتم خفاش با وجود فواید زیاد و کاربردی، دارای معایبی نیز است. یکی از این معایب که باعث کاسته شدن کارایی آن می‌شود، به دام افتادن در بهینه محلی است. برای حل مشکل مذکور در این پژوهش موقعیت و سرعت جمعیت اولیه را به سه روش با فرمول‌های مختلف بروز کرده، این امر باعث می‌شود تا پاسخ نهایی مسئله در بهینه محلی به دام نیفتد و تنوع در جمعیت رخ دهد. در این مقاله عملکرد الگوریتم خفاش بهبودیافته روی ۱۱ تابع هدف نمونه بررسی و با سایر الگوریتم‌های مشابه مقایسه شده است، که نهایتاً نتایج حاصل شده نشان از برتری و دقت این الگوریتم نسبت به نمونه‌های مشابه دارد.
بازنگری مقاله: ۱۴۰۳/۰۲/۰۹	
پذیرش مقاله: ۱۴۰۳/۰۳/۱۶	
واژگان کلیدی:	
بهینه‌سازی، هوش جمعی، الگوریتم فراابتکاری، بهینه محلی، الگوریتم خفاش.	
DOI: https://doi.org/10.22075/jme.2024.33011.2608	
© 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

در دهه‌های اخیر، پژوهش‌ها درباره روش‌های فراابتکاری به طرز سریعی گسترش یافته است و تعداد الگوریتم‌های الهام گرفته از طبیعت برای حل مسائل بهینه‌سازی چالش برانگیز به طرز چشمگیری افزایش یافته است به صورتی که بیش از ۴۰ الگوریتم الهام گرفته از طبیعت در سال‌های اخیر ابداع شده‌اند [۱]. این الگوریتم‌ها براساس تقلید از پدیده‌های زیستی یا فیزیکی بنا شده‌اند و مزیت آن‌ها در ساده بودن پیاده‌سازی‌شان است. علاوه بر این، نیازی به اطلاعات ریاضی ندارند و می‌توانند به خوبی بهینه سراسری را در مسائل مختلف بیابند. همچنین، می‌توانند در مسائل متنوعی که در

رشته‌های مختلف وجود دارند، به کار گرفته شوند. به عبارت دیگر، می‌توان گفت که الگوریتم‌های فراابتکاری در طراحی‌های مهندسی و برای حل مسائل مختلف به شدت محبوب شده‌اند [۲، ۳]. در یک نگاه کلی روش‌های فراابتکاری به سه گروه تقسیم می‌شوند که عبارتند از: روش‌های مبتنی بر فیزیک، مبتنی بر جمعیت و مبتنی بر تکامل [۴]. اولین گروه روش‌های مبتنی بر فیزیک هستند که قوانین فیزیکی جهان را تقلید می‌کنند. از جمله روش‌های موفق گروه اول می‌توان به تبرید شبیه‌سازی شده [۵]، سیاه‌چاله [۶] و مانند آن اشاره نمود. همچنین، روش‌های مبتنی بر تکامل که از قوانین تکامل طبیعی الهام گرفته‌اند عبارتند از: الگوریتم‌های

* پست الکترونیک نویسنده مسئول: Davood.giveki@gmail.com

۱. گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی دانشگاه ملایر، ملایر، ایران

۲. گروه مهندسی برق، موسسه آموزش عالی آفرینش علم گستر بروجرد، بروجرد، ایران

۳. گروه مهندسی کامپیوتر، موسسه آموزش عالی آفرینش علم گستر بروجرد، بروجرد، ایران

استناد به این مقاله:

[۱۹] از ترکیب الگوریتم خفاش و روش k-means برای ارائه یک روش خوشه بندی جدید استفاده کردند. ونگ و همکاران او همچنین از ترکیب جست و جوی هارمونی و الگوریتم خفاش برای بهبود الگوریتم خفاش استفاده کردند [۲۰]. فیستر و همکاران در روشی جدید با ترکیب الگوریتم خفاش و تفاضل تکاملی، الگوریتم خفاش استاندارد را بهبود دادند [۲۱]. در پژوهشی دیگر چن و همکاران از ترکیب الگوریتم خفاش و ماشین بولترمن برای بهبود کارایی الگوریتم خفاش استفاده کردند [۲۲]. همچنین برخی دانشمندان در پژوهشی دیگر برای بهبود الگوریتم خفاش از یک تابع بلندی صدای جدید استفاده کرده‌اند [۲۳]. در [۲۴] از یک عملگر جهش خاص برای بهبود الگوریتم خفاش استفاده شده است، همچنین در پژوهشی دیگر، لی و همکاران با ترکیب الگوریتم بهینه‌سازی کرم میوه و الگوریتم خفاش، نسخه‌ای بهبود یافته از الگوریتم خفاش را ارائه کرده‌اند [۲۵]. در آخر هم زو و همکاران در مقاله خود استفاده از یک الگوریتم غیرخطی را برای به روز کردن وزن اینرسی پارامترها در الگوریتم خفاش پیشنهاد داده‌اند [۲۶]. چنانچه مرور پیشینه نشان می‌دهد الگوریتم‌های فراابتکاری و به خصوص الگوریتم خفاش در حل بسیاری از مسائل مهندسی کارساز بوده است. با این وجود این الگوریتم همچنان مشکلاتی دارد که موجب به دام افتادن آن در بهینه محلی می‌شود. به عنوان مثال، تنوع جمعیت اولیه در این الگوریتم خفاش و سایر فراابتکاری‌ها بسیار ضعیف است و در مراحل پایانی جستجو برخی گند عمل کرده و از قدرت جستجوی محلی پایینی برخوردار هستند. همچنین ثابت بودن شیوه حرکت ذرات یکی از مسائلی است که در اکثر الگوریتم‌ها وجود دارد و خود می‌تواند باعث کاهش کیفیت پاسخ‌های بهینه شود.

لذا با در نظر گرفتن مشکلات موجود، در این پژوهش سعی بر آن است تا با ارائه روشی جدید نسخه استاندارد الگوریتم خفاش را بهبود دهیم. به صورتی که فرایند اکتشاف و بهره‌برداری الگوریتم پیشنهادی در مقایسه با نسخه اولیه از کارایی بهتری برخوردار شود. الگوریتم پیشنهادی با ایجاد تنوع و گوناگونی و با بروزرسانی موقعیت و سرعت خفاش‌ها با سه روش مختلف در یک زمان، باعث می‌شود که الگوریتم به بهینه سراسری نزدیک شده و در بهینه محلی به دام نیفتد. در این راستا و برای رفع مشکل اکتشاف بهینه سراسری، از سه روش حرکتی: الگوریتم خفاش استاندارد،

ژنتیک [۷]، تکامل راهبردی [۸]، بهینه‌ساز مبتنی بر جغرافیای زیستی [۹] و غیره هستند. دسته سوم روش‌های مبتنی بر جمعیت هستند که سعی در تقلید رفتار جمعی یک سیستم زیستی به عنوان گروه‌های حیوانی دارند. از روش‌های موفق این دسته می‌توان به بهینه‌سازی ازدحام ذرات [۱۰]، بهینه‌سازی کلونی مورچه [۱۱]، الگوریتم بهینه‌سازی گرگ خاکستری [۱۲]، الگوریتم بهینه‌سازی نهنگ [۳۲]، الگوریتم خفاش [۱۳] و نظایر آن هستند. در ادامه به بررسی ویژگی‌های برخی از این مقالات می‌پردازیم. الگوریتم خفاش توسط آقای شین-شی یانگ در سال ۲۰۱۰ معرفی شد که الگوریتم رفتار پژواک ریزخفاش‌ها را شبیه‌سازی می‌کند و در عین سادگی انعطاف‌پذیری بالایی دارد. بنابراین به راحتی می‌تواند برای حل مسائل پیچیده و مختلف استفاده شود [۱۳، ۱۴]. پژوهش‌ها نشان می‌دهند، الگوریتم خفاش اولین روش مبتنی بر تنظیم فرکانس و توابع پژواکی است که تنظیم فرکانس و پژواک در آن نقش عملگر جهش را ایفا می‌کنند. این قابلیت می‌تواند دوگان‌هایی در الگوریتم‌های مشابه مانند: جستجوی هارمونی [۱۵]، بهینه‌سازی ازدحام ذرات و تبرید شبیه‌سازی شده داشته باشد. به عبارت دیگر، الگوریتم خفاش مزیت‌هایی مشابه با فراابتکاری‌های مبتنی بر هوش جمعی دارد. علاوه بر این، نرخ انتشار پالس و تغییرات در بلندی صدا از قابلیت‌های منحصر به فرد این الگوریتم هستند به صورتی که هرچه جستجو به بهینه سراسری نزدیک‌تر شود، قدرت بهره‌برداری الگوریتم نیز افزایش می‌یابد [۱۴]. این قابلیت باعث می‌شود که الگوریتم خفاش روی ناحیه‌ای که بهینه سراسری در آن وجود دارد، تمرکز کند، که این قابلیت یک مزیت شاخص برای الگوریتم خفاش نسبت به سایر الگوریتم‌های فراابتکاری است. به طور کلی، بسیاری از الگوریتم‌های فراابتکاری از پارامترهای ثابت از پیش تعیین شده، استفاده می‌کنند. در حالی که، الگوریتم خفاش با کنترل پارامترهای نرخ انتشار پالس و تغییرات در بلندی صدا، فرآیند بهینه‌سازی را انجام می‌دهد [۱۴]. این الگوریتم به طور خودکار از اکتشاف به بهره‌برداری تغییر وضعیت می‌دهد که این امر در نهایت همگرایی الگوریتم به بهینه سراسری را تضمین می‌کند [۱۶، ۱۵].

ناکامورا و همکاران [۱۷] و میرجلیلی و همکاران [۱۸] نسخه‌های گسسته‌ای از الگوریتم خفاش را ارائه دادند که برای حل مسائل دارای انتخاب ویژگی خاص و طبقه‌بندی استفاده می‌شوند. در پژوهشی دیگر، کوماراسامی و واهی

$$V_i^t = V_i^{t-1} + (X_i^{t-1} - X^*) \cdot f_i \quad (4)$$

$$X_i^t = X_i^{t-1} + V_i^t \quad (5)$$

اگر عدد تصادفی $\text{rand} \in [0,1]$ بزرگتر از r_i بود، آنگاه یک جواب محلی جدید، اطراف بهترین جواب کل جمعیت (X_{new}) مطابق با رابطه (۶) تولید می‌کند.

$$X_{\text{new}} = X^* + \sigma \cdot \text{randN}(0,1)_{1 \times \text{nvar}} \cdot A^t \quad (6)$$

$$A^t = \frac{\sum_{i=1}^P A_i^t}{P}$$

در رابطه (۶)، σ به عنوان انحراف معیار عملکرد خفاش در نظر گرفته می‌شود که مقدار آن برابر با $0/1$ است. همچنین $\text{randN}(0,1)$ عددی تصادفی با توزیع نرمال در بازه بین صفر تا یک است که ابعاد آن به اندازه متغیرهای مسئله، مورد حل (nvar) است. همچنین A^t میانگین بلندی صدای همه خفاش‌ها در تکرار t ام است. اگر عدد تصادفی rand ($\text{rand} \in [0,1]$) کوچکتر از A_i و همچنین $f(x_{\text{new}_i}) < f(x_{\text{old}_i})$ بود، آنگاه مقادیر بلندی صدا (A) و نرخ انتشار (r) را مطابق روابط (۷) و (۸) به روز کن و مقدار $f(x_{\text{old}_i}) = f(x_{\text{new}_i})$ را قرار بده.

$$A_i^{t+1} = \alpha A_i^t \quad (7)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (8)$$

در روابط (۷) و (۸)، α فاکتور کاهش A و γ فاکتور کاهش r هستند که به ترتیب برابر با $0/9$ و $0/1$ در نظر گرفته می‌شوند. همچنین t شماره تکرار و i شماره جمعیت هستند. اگر $f(x_{\text{new}_i}) < f(x^*)$ آنگاه:

$$x^* = x_{\text{new}_i}$$

$$f(x^*) = f(x_{\text{new}_i})$$

۳- الگوریتم خفاش بهبودیافته

همانطور که قبلاً گفته شد الگوریتم خفاش یک الگوریتم قدرتمند است، که در بهره‌برداری بسیار خوب عمل می‌کند اما در اکتشاف کمی ضعف دارد و به راحتی در بهینه محلی به دام می‌افتد. به همین منظور و برای رفع این مشکل در الگوریتم خفاش و برای ایجاد تنوع، جمعیت اولیه به سه قسمت مساوی تقسیم می‌شود و در هر قسمت از یک سوم جمعیت با بهره‌گیری از یک روش جدید، و با اعمال سه روش متفاوت عملکرد الگوریتم بهبود داده شده است. بر روی هر ثلث از جمعیت اولیه، با اجرای عملیات مختلف و با ایجاد اصلاحاتی در بروزرسانی موقعیت و سرعت خفاش‌ها سعی

الگوریتم خفاش با تغییر فاکتور وزن اینرسی و الگوریتم خفاش با پرواز لوی، برای به‌روزرسانی موقعیت و سرعت خفاش‌ها استفاده شده است که نتایج به دست آمده حاکی از کارایی و بهینگی نسخه بهبودیافته الگوریتم نسبت به نسخه استاندارد آن است.

ادامه این پژوهش به شکل زیر سازماندهی شده است. در بخش ۲ الگوریتم خفاش استاندارد معرفی گردیده است. بخش ۳ به روش پیشنهادی برای بهبود الگوریتم خفاش استاندارد می‌پردازد. بخش ۴ به نتایج شبیه‌سازی‌ها و مقایسه‌ها می‌پردازد. در نهایت نتیجه‌گیری‌ها و پیشنهادهای آتی در بخش ۵ بیان می‌شوند.

۲- الگوریتم خفاش استاندارد

الگوریتم خفاش مبتنی بر ویژگی پژواک‌یابی ریزخفاش‌ها عمل می‌کند. تمامی خفاش‌ها با به کارگیری پژواک‌یابی می‌توانند مسافت را تخمین بزنند و تفاوت بین طعمه و موانع ثابت را تشخیص دهند. خفاش‌ها به صورت تصادفی با سرعت V_i در موقعیت X_i و با فرکانس f_{min} با طول موج متغیر ثابت λ و بلندی صدای A_{min} در جستجوی شکار هستند. آن‌ها می‌توانند به طور اتوماتیک طول موج پالس‌های منتشر شده از خود را تنظیم کرده و نرخ انتشار پالس خود یعنی $\text{er}(0,1)$ را طبق نزدیکی طعمه خود مطابقت دهند. اگرچه بلندی صدا می‌تواند به طرق مختلف تغییر کند؛ اما فرض می‌شود که بلندی از یک مقدار بزرگ A_0 مثبت تا یک مقدار مینیمم ثابت یعنی A_{min} تغییر کند. همچنین می‌توان از این تقریب استفاده کرد که به طور کلی فرکانس f در یک بازه $[f_{\text{min}}, f_{\text{max}}]$ است که مربوط به یک طیف طول موج $[\lambda_{\text{min}}, \lambda_{\text{max}}]$ می‌باشد (Yang, 2010).

$$f_i = f_{\text{min}} + (f_{\text{max}} - f_{\text{min}})\beta \quad (1)$$

$$r_i^1 = r_0 \cdot \text{ones}(P, 1) \quad (2)$$

$$A_i^1 = A_0 \cdot \text{ones}(P, 1) \quad (3)$$

در رابطه ۱، $\beta \in [0, 1]$ یک بردار تصادفی است که دارای توزیع یکنواخت است. همچنین r_0 و A_0 به ترتیب نرخ انتشار و بلندی صدای اولیه هستند که مقادیر آن‌ها برابر $0/5$ و $0/9$ در نظر گرفته شده است. همچنین f_{min} و f_{max} به ترتیب 0 و 2 در نظر گرفته شده‌اند. سرعت و موقعیت خفاش‌ها را مطابق با روابط (۴) و (۵) محاسبه و به‌روز می‌کنیم.

همچنین در رابطه (۱۲)، $\text{diff}X_i^t$ اندازه فاصله هر خفاش تا بهترین خفاش در هر تکرار است؛ که از این رابطه برای به دست آوردن مقدار فرکانس با استفاده از رابطه (۱۳) استفاده می‌شود [۱۳].

۳-۳- مرحله سوم (پرواز لوی)

در معادله به‌روزرسانی سرعت الگوریتم خفاش استاندارد قبل از سرعت در تکرار قبل V_i^{t-1} ضرب ثابت یک قرار دارد که این ضرب ثابت یک برای جستجوی سراسری مناسب نیست و انعطاف‌پذیری خفاش را کاهش می‌دهد و باعث گیر افتادن در بهینه محلی می‌شود. الگوریتم بهینه‌سازی ازدحام ذرات، جستجوی سراسری و محلی را با تنظیم وزن اینرسی متعادل می‌کند. برای ثلث سوم کل جمعیت P سرعت و موقعیت خفاش‌ها را مطابق با این روش و با استفاده از روابط (۱۴) تا (۱۹) محاسبه و به‌روز می‌کنیم:

$$V_i^t = \omega_{\text{iter}} V_i^{t-1} + (X_i^{t-1} - X^*) \cdot f_i \quad (14)$$

$$\omega_{\text{iter}} = \omega_{\text{max}} - \text{atan} \frac{4t}{\text{Max}_{\text{iter}}} (\omega_{\text{max}} - \omega_{\text{min}}) \quad (15)$$

در رابطه (۱۴) سرعت با استفاده از ضرب جدید محاسبه شده است. رابطه (۱۵)، همان ضرب وزنی کاهش‌ی است؛ ω_{min} و ω_{max} به ترتیب حداقل و حداکثر ضرب وزنی کاهش‌ی هستند که مقادیر $0/42$ و $0/9$ را به خود اختصاص داده‌اند. Max_{iter} نیز نشان‌دهنده حداکثر تعداد تکرارها است. با توجه به رابطه (۱۵) این بار ضرب وزنی کاهش‌ی به صورت مثلثاتی حرکت خود را دنبال می‌کند [۲۸].

سال ۱۹۲۶، پل لوی، ریاضیدان فرانسوی، پروازهای لوی را پیشنهاد کرد. پدیده Lévy Flights در طبیعت بسیار رایج است. فعالیت‌های جستجوی موجوداتی مانند زنبورها، شغال‌ها، میمون‌ها و رفتارهای شکار انسان، همگی با مدل حرکت تصادفی پروازهای Lévy مطابقت دارد. به عنوان مثال، برخی از گیاهخواران به طور تصادفی در یک منطقه خاص حرکت می‌کنند تا منبعی از تغذیه را پیدا کنند، اما اگر نتوانند آن را پیدا کنند، به سرعت به منطقه دیگری می‌روند و سپس راه رفته قبلی را از سر می‌گیرند. این می‌تواند به طور موثر از اتلاف وقت در مکانی با منابع ناکافی جلوگیری کند. در طول فرآیند تکراری، تنوع جمعیت کاهش می‌یابد و توانایی جستجوی سراسری تضعیف می‌شود. در نتیجه الگوریتم به راحتی در بهینه محلی قرار می‌گیرد و دچار همگرایی زودرس می‌شود. برای حل این مشکل از روش

شده تا عملکرد الگوریتم بهبود داده شود.

۳-۱- مرحله اول

در ثلث اول جمعیت، به منظور به‌روزرسانی سرعت و موقعیت خفاش از روش الگوریتم خفاش استاندارد استفاده شده است. یعنی به‌روزرسانی موقعیت و سرعت خفاش عیناً مشابه الگوریتم خفاش استاندارد صورت گرفته شده است [۱۴].

۳-۲- مرحله دوم (تغییر فاکتور وزن اینرسی)

در ثلث دوم جمعیت، از روش تغییر فاکتور وزن اینرسی استفاده شده است. با ادامه روند کار الگوریتم خفاش، الگوریتم به تدریج قابلیت بهره‌برداری را از دست می‌دهد. برای جلوگیری از این مشکل، قابلیت بهره‌برداری با BA با درج ضرب وزنی اینرسی کاهش‌ی خطی بهبود می‌دهیم. سرعت و موقعیت خفاش‌ها مطابق با روابط (۹) تا (۱۳) محاسبه و به‌روز می‌کنیم:

$$V_i^t = \omega_{\text{iter}} V_i^{t-1} + (X_i^{t-1} - X^*) \cdot f_i \quad (9)$$

$$\omega_{\text{iter}} = \frac{\text{Max}_{\text{iter}} - t}{\text{Max}_{\text{iter}}} (\omega_{\text{max}} - \omega_{\text{min}}) + \omega_{\text{min}} \quad (10)$$

$$X_i^t = X_i^{t-1} + V_i^t \quad (11)$$

برای به‌روزرسانی سرعت خفاش از ضرب وزنی کاهش‌ی استفاده شده است که در رابطه (۹)، ω_{iter} ضرب وزنی کاهش‌ی برای به‌روزرسانی سرعت است. مقدار ω_{iter} از رابطه (۱۰) حاصل می‌گردد؛ در جایگاه iter مقدار تعداد تکرار فعلی است، Max_{iter} حداکثر تعداد تکرار است؛ در اینجا ω_{min} و ω_{max} مقادیر حداقل و حداکثر ضرب وزنی هستند که به ترتیب برابر با $0/2$ و $0/9$ هستند. در رابطه (۱۱) موقعیت خفاش به‌روز می‌شود.

$$\text{diff}X_i^t = \sqrt{|(X_i^{t-1} - X^*)^2|} \quad (12)$$

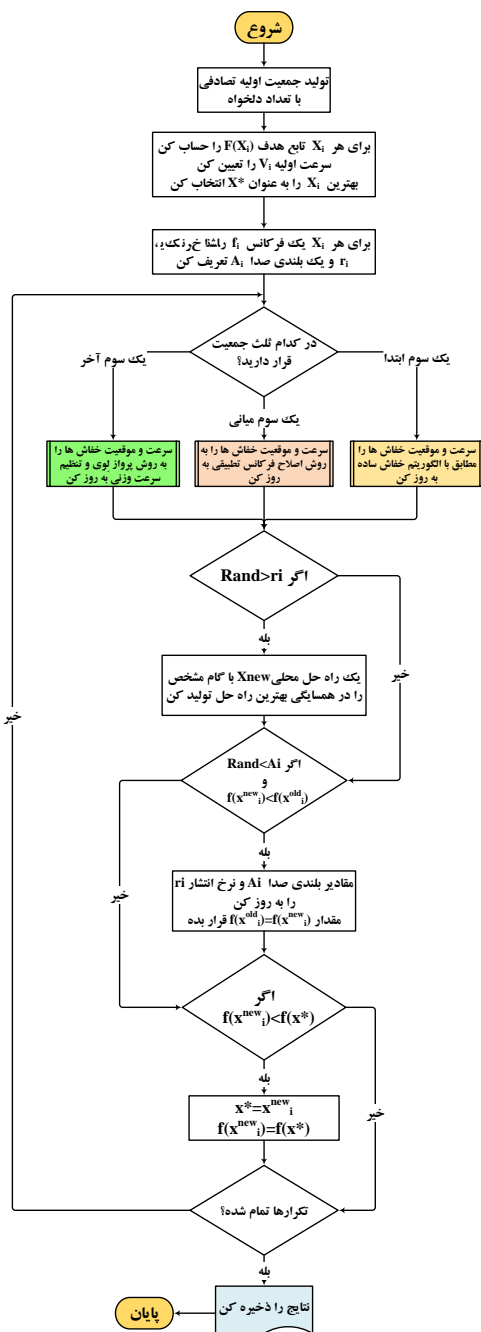
$$f_i = f_{\text{min}} + (f_{\text{max}} - f_{\text{min}}) \frac{\sqrt{|\text{min}(\text{diff}X_i^t) - \text{Avg}(\text{diff}X_i^t)|}}{\text{max}(\text{diff}X_i^t) - \text{min}(\text{diff}X_i^t)} \quad (13)$$

ابتدا فواصل بین راه‌حل هر خفاش و بهترین راه‌حل سراسری برای همه ابعاد ارزیابی می‌شود سپس نزدیک‌ترین و دورترین ابعاد راه‌حل خفاش به ترتیب f_{min} و f_{max} اختصاص داده می‌شود و در نهایت فرکانس ابعاد دیگر در محدوده f_{min} و f_{max} با توجه به فواصل آنها تغییر می‌کند.

vi. Iteration= Iteration+1
b. End
6- End

۴-۳- فلوجارت الگوریتم خفاش بهبودیافته

فلوجارت الگوریتم خفاش بهبودیافته در شکل (۱) نشان داده شده است.



شکل ۱- فلوجارت الگوریتم خفاش بهبودیافته

۴- نتایج شبیه سازی ها و مقایسه ها

الگوریتم خفاش بهبودیافته پیشنهادی با استفاده از متلب R2017b پیاده سازی شده است. در این قسمت به یک مقایسه منصفانه بین نتایج به دست آمده از الگوریتم خفاش

جستجوی پرواز لوی به خفاش اضافه شده و موقعیت خفاش به صورت زیر به روز شده است:

$$X_i^t = \text{Lévy} \cdot X_i^{t-1} + V_i^t \cdot C_i(t) \quad (16)$$

$$\text{Lévy} = 0.01 \frac{\Gamma_1}{|\Gamma_2|^{\frac{1}{\varphi}}} \quad (17)$$

$$\left(\frac{\Gamma(1 + \varphi) \cdot \sin \frac{\pi\varphi}{2}}{\Gamma\left(\frac{1+\varphi}{2}\right) \cdot \varphi \cdot 2^{\frac{\varphi-1}{2}}} \right)^{\frac{1}{\varphi}}$$

$$\Gamma(1 + \varphi) = \int_0^{\infty} x^{\varphi} \cdot e^{-x} dx \quad (18)$$

$$C_i(t) = \theta^P \cdot e^{\left(\frac{-t}{\text{Maxitr}}\right)} \quad (19)$$

در رابطه (۱۷)، Γ_1 و Γ_2 دو عدد تصادفی با توزیع نرمال در بازه بین ۰ تا ۱ هستند. همچنین مقدار φ برابر با ۱/۵ در نظر گرفته شده است. تعریف تابع گاما (Γ) مطابق با رابطه (۱۸) است. رابطه (۱۹)، ضرب تنظیم سرعت است که به شکل نمایی تغییر می کند. θ عددی بین ۰ تا ۱ است و در اینجا مقدار ۰/۰۱ در نظر گرفته شده است و هم تعداد کل جمعیت اولیه است [۲۷]. شبه کد الگوریتم خفاش سه فازی به صورت زیر نشان داده شده است.

شبه کد الگوریتم خفاش سه فازی

- 1- Determine the limits of optimization variables l_b, u_b, D .
- 2- Define the adjustable parameters of the bat algorithm including γ, α, f_{min} and f_{max} .
- 3- Determine the position of each bat based on the values of the pulse emission rate r , the loudness of the sound A and the number of iterations.
- 4- Calculate the value of the objective function for each bat and put the current optimal solution as x^* .
- 5- While Iteration \neq Maximum
 - a. For $i = 1$: Pop
 - i. Update the frequency, speed and position values of the bat based on the equations (1), (4) and (5).
 - ii. Divide the initial population into three equal parts
 1. For the first group, update the speed and position of the bat based on the standard bat with equations (4) and (5).
 2. For the second group, update the speed and position of the bat based on the method of changing the inertial weight factor and using equations (9) and (11).
 3. For the third group, update the velocity and position of the bat based on Levy's bat flight and with relations (14) and (16).
 - iii. If $\text{rand} > r_i$ then
 1. Create a new solution around the best solution
 2. Generate a stochastic solution with random flight based on equation (6).
 - iv. If $\text{rand} < A_i$ & $f(x_i(t+1)) < f(x^*)$ then
 1. Accept the new solution and update the control parameters r_i and A_i based on equations (7) and (8).
 - v. Update the global optimal solution and its corresponding objective function value

الگوریتم‌های کلونی زنبور عسل، ازدحام ذرات، الگوریتم خفاش بهبودیافته با بهینه‌ساز گرگ خاکستری و الگوریتم خفاش بهبودیافته با استراتژی ترکیبی براساس نتایج به دست آمده در تحقیقات قبل مورد مقایسه با این الگوریتم‌ها مورد مقایسه قرار داده شده است. پارامترهای مورد استفاده در هر کدام از الگوریتم‌ها در جدول ۱ بیان شده و قرار داده شده است.

بهبودیافته با الگوریتم‌های ژنتیک، خفاش استاندارد، کلونی زنبور عسل مصنوعی [۳۱]، ازدحام ذرات [۳۱]، الگوریتم خفاش بهبودیافته با بهینه‌ساز گرگ خاکستری [۳۰] و الگوریتم خفاش بهبودیافته با استراتژی ترکیبی [۲۹] پرداخته شده است. در این راستا علاوه بر الگوریتم خفاش بهبودیافته، الگوریتم ژنتیک و خفاش استاندارد نیز به صورت کامل اجرا شده و نتایج آن‌ها گزارش شده است.

جدول ۱- پیکربندی پارامترها برای هر الگوریتم بهینه‌سازی

الگوریتم	پارامتر	مقدار
TPBA	w_{min}	0.2
	w_{max}	0.9
	r_1, r_2	rand[0,1]
	ϕ	1.5
	θ	0.01
GA	Mutation Rate	0.001
	Elitism rate	0.2
	Crossover rate	0.95
BA	Minimum Frequency	0
	Maximum Frequency	2
	Velocity	0
	Loudness	0.9
	Rate of Pulse Emission	0.5
PSO	Maximum weight	0.9
	Minimum weight	0.4
ABC	r	Rand[-1,1]
	Φ	Rand[0,1]
HSIBA	Fixed frequency	$f_{min} = -2, f_{max} = 2$
	Fixed flight weight	$w_{min} = 0.2, w_{max} = 0.8$
	Initial pulse frequency	$r_i^0 = 0, i \in \{1, 2, \dots, I\}$
	Initial pulse loudness	$A_i^0 = 2, i = \{1, 2, \dots, I\}$
IBA	A	0.9
	R	0.5
	Population Size	50

از حداقل‌های محلی سروکار داشته باشیم. الگوریتم خفاش بهبودیافته در راستای ارتقای الگوریتم خفاش استاندارد با ایجاد تغییراتی به بهبود عملکرد الگوریتم خفاش پرداخته شده است. در این مرحله نتایج حاصل شده از این الگوریتم مورد بررسی قرار گرفته شده است. الگوریتم خفاش بهبودیافته بر روی ۱۱ تابع هدف با دو سناریو مورد بررسی قرار داده شده است: در یک سناریو تعداد جمعیت اولیه ثابت و تعداد تکرارها تغییر داده شده است. در سناریوی دوم تعداد تکرارها ثابت و جمعیت اولیه متغیر در نظر گرفته شده است. در سناریوی اول با تغییر در تعداد تکرارها، شش حالت ایجاد شده که هر حالت به صورت جداگانه مورد بررسی قرار گرفته شده است. در حالت اول تعداد تکرارها ۱۰۰، در

در این مقاله جهت اعتبارسنجی الگوریتم خفاش بهبودیافته از ۱۱ تابع هدف پیوسته استفاده شده است. این توابع با تمام قیدها و دامنه آن‌ها در جدول ۱ نشان داده شده است. به طور معمول، این توابع را می‌توان به سه دسته: (۱) توابع تک وجهی، (۲) توابع چندوجهی و (۳) توابع تست چندوجهی با بعد ثابت؛ تقسیم کرد. در این جدول، D بعد تابع را نشان می‌دهد. در بهینه‌سازی، چون ما با حداقل‌های محلی زیادی روبرو هستیم توابع چندوجهی با ابعاد بالا بسیار مسائل سختی هستند. برای این نوع مسائل، چون الگوریتم‌ها در فرار از بهینه محلی ضعیف هستند و تعیین یک بهینه تقریباً سراسری را به چالش کشیده‌اند، نتایج نهایی مهم‌تر می‌باشد (مانند توابع شماره ۷ و ۸ و ۹). همچون توابع پیشین، توابع شماره ۱۰ و ۱۱ جز توابع کم‌بعدی چندوجهی قرار دارند؛ چون ابعاد این توابع کم است، باید با تعداد کمی

بررسی قرار گرفته شده است. در این سناریو نیز حالت اول تعداد جمعیت اولیه ۲۵، حالت دوم ۵۰، حالت سوم ۷۵، حالت چهارم ۱۰۰، حالت پنجم ۱۲۵ مورد بررسی قرار داده شده است.

حالت دوم ۲۰۰، حالت سوم ۳۰۰، حالت چهارم ۴۰۰، حالت پنجم ۵۰۰، حالت ششم ۶۰۰ در نظر گرفته شده است. در سناریوی دوم و با تغییر جمعیت اولیه، پنج حالت ایجاد شده و الگوریتم با جدول‌ها و شکل‌های حاصل شده مورد

جدول ۲- توابع هدف و قیود

فرمول	تابع معیار	بعد تابع	رده	حالت مطلوب
$F01 = \sum_{i=1}^D x_i^2$	Sphere	30	[-100,100]	0
$F02 = \sum_{i=1}^D x_i + \prod_{i=1}^n x_i $	Schwefel 2.22	30	[-10,10]	0
$F03 = x^2 + y^2 + 25(\sin x)^2 + (\sin y)^2$	Sine function	30	[-100,100]	0
$F04 = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	Generalized Rosenbrock	30	[-30,30]	0
$F05 = \sum_{i=1}^D ([x_i + 0.5])^2$	STEP	30	[-100,100]	0
$F06 = \sum_{i=1}^D ix_i^4 + \text{random}[0,1]$	Quartic	30	[-1.28,1.28]	0
$F07 = \sum_{i=1}^D -x_i \sin(\sqrt{ x_i })$	Generalized Schwefel	30	[-500,500]	-418.9829*n
$F08 = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	Rastrigin	30	[-5.12,5.12]	0
$F09 = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i\right) + 20 + e$	Ackley	30	[-32,32]	8.8818e-16
$F10 = 4x_1^4 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	Six-Hump Camel-Back	2	[-5,5]	-1.0316285
$F11 = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	Goldstein Price	2	[-5,5]	3

می‌یابد، این مقایسه‌ها توانایی الگوریتم‌ها برای غلبه بر به دام افتادن در بهینه محلی و برتری الگوریتم خفاش بهبودیافته پیشنهادی نسبت به سایر الگوریتم‌ها را نشان می‌دهد. در بخش دوم و در جدول ۲ به مقایسه بر مبنای تغییر در جمعیت اولیه پرداخته شده است. با توجه به نتایج حاصل شده و مقایسه این نتایج با سایر الگوریتم‌های ذکر شده، نشان‌دهنده قدرتمندتر و کارآمدتر بودن الگوریتم خفاش بهبودیافته پیشنهادی نسبت به سایر الگوریتم‌های فراابتکاری مذکور می‌باشد.

۴-۱- مقایسه توابع بر مبنای تغییر در تعداد تکرار
در این بخش الگوریتم خفاش بهبودیافته پیشنهادی با سایر الگوریتم‌های فراابتکاری مورد مقایسه قرار گرفته است. ملاک مقایسه‌ها براساس مقدار کمینه حاصل شده برای تابع تناسب و میانگین برای هر تابع می‌باشد. در بخش اول و در جدول ۳ به مقایسه نتایج مشترک این پژوهش با سایر الگوریتم‌ها از لحاظ تغییر در تعداد تکرارها پرداخته شده است. تا زمانی که بعد فضای جستجو افزایش می‌یابد، مقدار توابع هدف در حداقل محلی به صورت تصاعدی افزایش

جدول ۳- مقایسه توابع بر مبنای تعداد تکرارها

تابع	GA	BA	PSO [32]	ABC [32]	HSIBA [30]	IBA [31]	LMBA [33]	PBA [34]	QABA [35]	MBA	
F ₀₁	Best Fitness	1.080191633	0.002859515	NA	NA	NA	2.71E-01	7.060E-29	5.0978E-19	0.000	0
	Mean	1.081075693	0.005955662	7.8157	3.5678E-5	0	2.71E-01	4.584E-10	2.6329E-18	0.00015	0.017508395
	Iteration	500	500	500	500	50	2000	-	-	30	600
	Rank	7	5	9	6	1	8	3	4	2	1
F ₀₂	Best Fitness	0.47846051	0.35486462	NA	NA	NA	7.90E-03	1.743E-15	1.5384E-09	0.000	4.2494E-172
	Mean	0.4904954	0.496594825	0.9676	7.7417E-06	2.005E-171	2.64E-02	2.406E-5	2.8843E-09	1.301E-309	0.497301219
	Iteration	600	600	950	950	50	2000	-	-	30	600
	Rank	8	9	10	6	5	7	4	3	2	1
F ₀₃	Best Fitness	2.16987E-12	2.07909E-06	NA	1.6214E-04	NA	NA	NA	NA	NA	0
	Mean	2.16987E-12	0.000952735	65.5158	NA	NA	NA	NA	NA	NA	0.000137362
	Iteration	600	200	500	500	NA	NA	NA	NA	NA	600
	Rank	2	3	4	5	-	-	-	-	-	1
F ₀₄	Best Fitness	0.138808651	3.87961E-06	0.241219	0.0607	2.867E-01	2.90E-01	3.176E-1	2.3212E-02	1.816E-6	9.87232E-08
	Mean	0.138808651	0.00055202	NA	NA	NA	3.02E+01	8.131	7.4372E-01	1.026E-4	0.001290841
	Iteration	300	600	8000	8000	50	2000	-	-	30	300
	Rank	4	3	6	10	7	8	9	5	1	2
F ₀₅	Best Fitness	0.463766435	0.003326823	2.6550E-1	4.8813E-04	2.147E-01	3.05E-1	1.046E-6	4.2960E-19	2.982E-9	0.043913321
	Mean	0.477631344	0.006155159	NA	NA	NA	3.22E-1	8.421E-3	2.5237E-18	2.816E-9	0.070776533
	Iteration	600	600	15	15	50	2000	-	-	30	500
	Rank	10	4	7	8	9	6	3	1	2	5
F ₀₆	Best Fitness	0.08012038	0.051199224	0.1118227	0.0621	4.149E-04	7.09E-03	NA	3.5365E-04	NA	4.00883E-07
	Mean	0.101107447	0.115097773	NA	NA	NA	1.71E-01	NA	3.3026E-03	NA	0.029213071
	Iteration	500	500	1500	1500	50	2000	NA	-	NA	300
	Rank	7	5	8	6	3	4	-	2	--	1
F ₀₇	Best Fitness	-6622.341335	-4398.981	-2.7722E+03	-1.2255E+03	NA	NA	NA	NA	NA	-6622
	Mean	-6622.295212	-4326.620213	NA	NA	NA	NA	NA	NA	NA	-5111.84693
	Iteration	500	100	1500	1500	NA	NA	NA	NA	NA	300
	Rank	3	1	4	5	-	-	-	-	-	2
F ₀₈	Best Fitness	0.002468744	0.02168926	0.2548421	0.1979254	NA	2.34E-01	0.000	0.0000E+0	0.000	0
	Mean	0.002468756	0.051412267	NA	NA	NA	2.71E-01	1.047E-5	1.1842E-16	0.000	0.0001836007
	Iteration	600	600	40	40	NA	2000	-	-	30	600
	Rank	5	6	9	8	-	7	3	2	1	4
F ₀₉	Best Fitness	0.272956616	0.40067189	0.74085	0.190133	8.881E-16	4.74E-03	8.882E-16	1.2250E-09	8.882E-16	4.44089E-16
	Mean	0.290366715	0.46082013	NA	NA	NA	9.95E-03	2.817E-9	2.8492E-09	8.882E-16	0.144813252
	Iteration	600	300	60	60	50	2000	-	-	30	300
	Rank	6	8	7	9	2	4	3	2	1	5
F ₁₀	Best Fitness	-1.031627021	-1.031610741	-1.313	-1.0316	NA	-1.03E+00	-1.0316	-1.0316E+00	-1.0316	-1.031605582
	Mean	-1.031627021	-1.031263643	NA	NA	NA	-1.03E+00	-1.0316	-1.0316E+00	-1.0316	-1.031143216
	Iteration	200	400	200	200	NA	2000	-	-	30	300
	Rank	3	2	6	5	-	4	7	7	7	1
F ₁₁	Best Fitness	3	3.000070107	4.2797	3	NA	NA	3.0000	3.0000E+00	3.0000	3.000049729
	Mean	3	3.020137099	NA	NA	NA	NA	11.6056	3.0000E+00	3.0093	3.014742468
	Iteration	600	500	180	180	NA	NA	-	-	30	600
	Rank	1	5	6	7	-	-	10	2	3	4

تکرارها، در این بخش به بررسی نتایج حاصل شده از الگوریتمها براساس تغییر در جمعیت اولیه پرداخته شده

۴-۲- مقایسه توابع بر مبنای تغییر در جمعیت اولیه پس از بررسی نتایج حاصل شده براساس تغییر در تعداد

تکرارها آورده شده است. با توجه به نمودار می‌توان مشاهده نمود که الگوریتم خفاش بهبودیافته به مقدار بهینه تابع هدف نزدیک بوده و به نتایج بهتری دست یافته است. در شکل (۳) مقایسه‌ها از لحاظ تغییر در جمعیت اولیه بین الگوریتم خفاش بهبودیافته با دو الگوریتم فراابتکاری دیگر (الگوریتم خفاش و ژنتیک) نشان داده شده است. همانطور که در نمودارها قابل مشاهده می‌باشد الگوریتم خفاش بهبودیافته بسیار بهتر عمل کرده و به نتایج بهتری دست یافته است.

است. در جدول ۴ با تغییر در جمعیت اولیه الگوریتم‌ها به مقایسه نتایج حاصل شده از آن پرداخته شده است. نتایج حاصل شده حاکی از آن است که الگوریتم خفاش بهبودیافته پیشنهادی نسبت به سایر الگوریتم‌های مورد بررسی به نتایج بهتری دست یافته و از عملکرد بهتری برخوردار می‌باشد که نشان‌دهنده برتری این الگوریتم نسبت به سایر الگوریتم‌ها می‌باشد. در شکل (۲) نمودار میله‌ای جهت مقایسه الگوریتم خفاش بهبودیافته با دو الگوریتم خفاش و ژنتیک با تغییر در تعداد

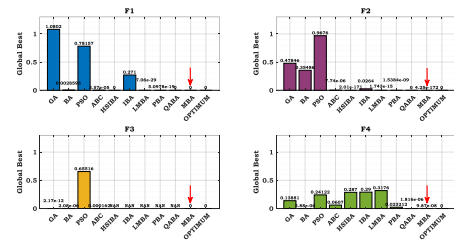
جدول ۴- مقایسه توابع بر مبنای جمعیت اولیه

توابع		GA	BA	PSO	ABC	HSIBA	IBA	MBA
F ₀₁	Best Fitness	0.299721559	0.004393037	0.78157	3.5678E-5	0	2.34E-01	0
	Mean	0.307516644	0.00740416	NA	NA	NA	2.71E-01	0.016990052
	Iteration	125	75	100	100	50	2000	100
	Rank	5	4	6	3	2	7	1
F ₀₂	Best Fitness	0.241643497	0.838411232	0.9676	7.7417E-06	2.005E-171	7.90E-03	1.5045E-192
	Mean	0.24170106	0.842991994	NA	NA	NA	2.64E-02	0.478550357
	Iteration	100	50	100	100	50	2000	100
	Rank	5	6	7	3	2	4	1
F ₀₃	Best Fitness	9.70655E-11	1.04317E-06	0.655158	0.162	NA	NA	0
	Mean	3.60612E-11	0.000340063	NA	NA	NA	NA	0.000117649
	Iteration	25	100	100	100	NA	NA	75
	Rank	2	3	4	5	-	-	1
F ₀₄	Best Fitness	0.065271235	7.86638E-09	0.241219	0.0607	2.867E-01	4.95E-07	4.94762E-07
	Mean	0.065271235	0.00063436	NA	NA	NA	3.02E-07	0.000434898
	Iteration	75	125	100	100	50	2000	75
	Rank	3	1	5	4	6	7	2
F ₀₅	Best Fitness	0.278806545	0.002564626	2.6550E-01	4.8813E-01	2.147E-01	3.05E-1	0.032176009
	Mean	0.303075779	0.00508962	NA	NA	NA	3.22E-1	0.074131302
	Iteration	125	75	100	100	50	2000	125
	Rank	3	1	6	7	4	5	2
F ₀₆	Best Fitness	0.030549602	0.05363214	0.1118227	0.0621	4.149E-04	7.09E-03	5.01441E-05
	Mean	0.036790846	3.443678317	NA	NA	NA	1.71E-01	0.020729725
	Iteration	125	25	100	100	50	2000	125
	Rank	4	6	7	5	2	3	1
F ₀₇	Best Fitness	-6018.908444	-4752.276172	-2.7722E+03	-1.2255E+03	NA	NA	-3955.828405
	Mean	-6018.873283	-4750.553652	NA	NA	NA	NA	-5722.855572
	Iteration	75	75	100	100	NA	NA	25
	Rank	3	2	4	5	-	-	1
F ₀₈	Best Fitness	0.000960591	0.023223761	0.2548421	0.1979254	NA	2.34E-01	0
	Mean	0.000960601	0.044435022	NA	NA	NA	2.71E-01	0.01401398
	Iteration	125	50	100	100	NA	2000	50
	Rank	2	3	6	5	-	4	1
F ₀₉	Best Fitness	0.123406471	0.96315151	0.74085	0.190133	8.881E-16	4.74E-03	4.44089E-16
	Mean	0.123773205	0.00049802	NA	NA	NA	9.95E-03	0.10675032
	Iteration	125	75	100	100	50	2000	75
	Rank	4	6	5	7	2	3	1
F ₁₀	Best Fitness	-1.031627754	-1.031621282	-1.313	-1.0316	NA	-1.03E+00	-1.031619191
	Mean	-1.031628204	-1.031372962	NA	NA	NA	-1.03E+00	-1.031368772
	Iteration	25	75	100	100	NA	2000	100
	Rank	5	2	3	6	-	4	1
F ₁₁	Best Fitness	3	3.000127765	4.2797	3	NA	NA	3.000040154
	Mean	3	3.013634645	NA	NA	NA	NA	3.015433388
	Iteration	125	25	100	100	NA	NA	100
	Rank	1	4	5	2	-	-	3

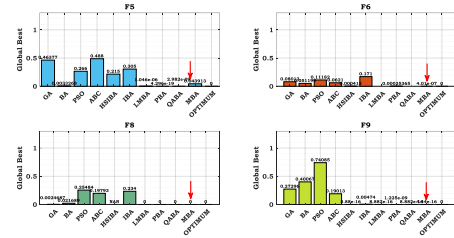
۵- نتیجه‌گیری و پیشنهادهای آتی

در این مقاله الگوریتم جدید خفاش بهبودیافته که روی ۱۱ تابع هدف مورد مطالعه قرار گرفته شده است. در این پژوهش با تغییر در تعداد تکرارها ۱۰۰، ۲۰۰، ۳۰۰، ۴۰۰، ۵۰۰ و ۶۰۰ و همچنین با تغییر در جمعیت اولیه به تعداد ۲۵، ۵۰، ۷۵، ۱۰۰ و ۱۲۵ مورد مطالعه قرار داده شده است و نتایج حاصل شده از آن با الگوریتم‌های ژنتیک، خفاش استاندارد، زنبور عسل، ازدحام ذرات، الگوریتم خفاش بهبودیافته با استراتژی ترکیبی و الگوریتم خفاش بهبودیافته با بهینه‌ساز گرگ خاکستری مورد مقایسه قرار گرفته شده است. با مقایسه این الگوریتم‌ها با یکدیگر می‌توان چنین بیان کرد که الگوریتم خفاش بهبودیافته نسبت به شش الگوریتم دیگر بهتر عمل کرده و به نتایج بهتری دست یافته است. مقدار تابع تناسب حاصل شده برای این الگوریتم نسبت به بقیه الگوریتم‌های مورد مقایسه بسیار کوچکتر بوده و به مقدار بهینه نزدیکتر شده است و حتی در بعضی موارد به خود مقدار بهینه رسیده است که این نتایج حاکی از عملکرد خوب و قوی بودن این الگوریتم نسبت به دیگر الگوریتم‌ها می‌باشد. از لحاظ همگرایی، الگوریتم خفاش بهبودیافته بسیار زودتر از دیگر الگوریتم‌ها همگرا شده و از سرعت همگرایی بالایی برخوردار است. پس می‌توان به این نتیجه رسید که این الگوریتم از عملکرد بسیار بهتری نسبت به شش الگوریتم دیگر دارد.

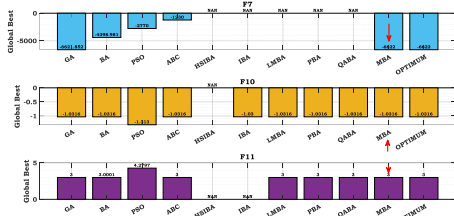
با توجه به مقایسه‌های انجام شده در دو جدول ۳ و ۴ می‌توان به این نتیجه رسید که الگوریتم خفاش بهبودیافته در بین ۲۲ حالت مورد مقایسه، در ۱۶ حالت بین تمام الگوریتم‌های مورد مقایسه دیگر به نتیجه بهتری رسیده است و در رده اول قرار گرفته است مقدار بهترین جواب حاصل شده برای الگوریتم خفاش بهبودیافته نسبت به سایر الگوریتم‌های دیگر بسیار بهتر می‌باشد؛ در چهار مورد و با اختلاف کمی در رده دوم دست قرار گرفته است؛ و تنها در دو مورد در رده سوم واقع شده است. این در حالی است که الگوریتم ژنتیک تنها در دو حالت، الگوریتم خفاش استاندارد در چهار حالت به رده اول رسیده‌اند. الگوریتم ازدحام ذرات، زنبور عسل، خفاش بهبودیافته با استراتژی ترکیبی و خفاش بهبودیافته با بهینه‌ساز گرگ خاکستری در هیچ یک از حالات به رده اول و به بهترین نتیجه دست نیافته‌اند و تنها در مواردی به رده دوم رسیده‌اند. پس می‌توان به این نتیجه رسید که الگوریتم خفاش بهبودیافته نزدیک به ۷۳ درصد



الف

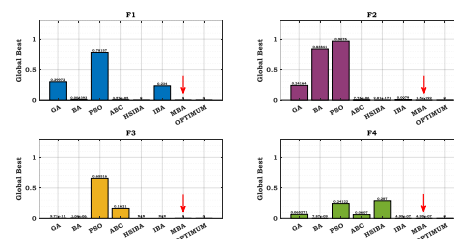


ب

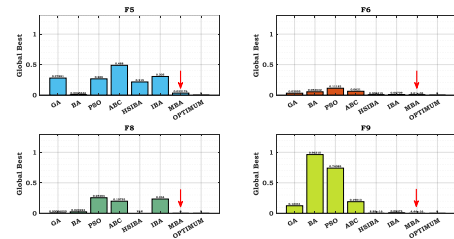


ج

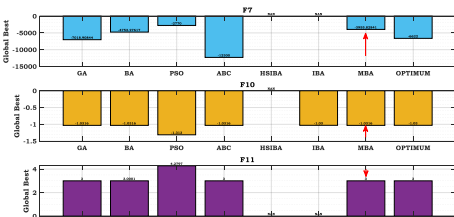
شکل ۲- مقایسه توابع بر مبنای تغییر در تعداد تکرارها



الف



ب



ج

شکل ۳- مقایسه توابع بر مبنای تغییر در جمعیت اولیه

یک سوم جمعیت از عدد کوواریانس استفاده شده است. می‌توان به جای استفاده از عدد تصادفی خطی، جهت بهبود الگوریتم، از عدد لاپلاس نیز استفاده نمود.

ب. همچنین می‌توان برای یک سوم از جمعیت، در راستای به‌روزرسانی سرعت در الگوریتم از الگوریتم گرگ خاکستری نیز جهت بهبود عملکرد الگوریتم بهره گرفت. ج. به عنوان پیشنهاد دیگر می‌توان جهت به‌روزرسانی سرعت، با ادغام الگوریتم علوفه هرز با الگوریتم خفاش در جهت بهبود عملکرد آن استفاده نمود.

نسبت به الگوریتم‌های دیگر به نتایج بهتری دست یافته است و باعث ارتقا و بهبود الگوریتم خفاش شده است.

۱-۵- پیشنهادات

با توجه به تحقیقات انجام شده در این پژوهش می‌توان موارد زیر را برای ادامه کار پیشنهاد کرد:

الف. در نسخه استاندارد الگوریتم خفاش برای به‌روزرسانی سرعت با اختصاص فرکانس، عدد تصادفی که انتخاب می‌شود به صورت خطی است؛ در این پژوهش برای

مراجع

- [1] I. Fister. "A comprehensive review of bat algorithms and their hybridization." PhD diss, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, 2013.
- [2] H. Rastegar, D. Giveki, and M. Choubin. "EEG signals classification using a new radial basis function neural network and jellyfish meta-heuristic algorithm." *Evolutionary Intelligence* 17, no. 2 (2024): 1197-1208.
- [3] D. Giveki, and M. Karami. "Scene classification using a new radial basis function classifier and integrated SIFT-LBP features." *Pattern Analysis and Applications* 23, no. 3 (2020): 1071-1084.
- [4] S.A. Mirjalili, and A. Lewis. "The whale optimization algorithm." *Advances in Engineering Software* 95 (2016): 51-67.
- [5] S. Kirkpatrick, C. Daniel Gelatt Jr, and M.P. Vecchi. "Optimization by simulated annealing." *Science* 220, no. 4598 (1983): 671-680.
- [6] A. Hatamlou. "Black hole: A new heuristic optimization approach for data clustering." *Information Sciences* 222 (2013): 175-184.
- [7] J.H. Holland. "An introductory analysis with applications to biology, control, and artificial intelligence." *Adaptation in Natural and Artificial Systems*. First Edition, The University of Michigan, USA (1975).
- [8] H.G. Beyer, and H.P. Schwefel. "Evolution strategies—a comprehensive introduction." *Natural Computing* 1 (2002): 3-52.
- [9] D. Simon. "Biogeography-based optimization." *IEEE Transactions on Evolutionary Computation* 12, no. 6 (2008): 702-713.
- [10] J. Kennedy, and R. Eberhart. "Particle swarm optimization." In *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4, pp. 1942-1948. IEEE, 1995.
- [11] M. Dorigo, V. Maniezzo, and A. Colomi. "Ant system: optimization by a colony of cooperating agents." *IEEE Transactions on Systems, Man, and Cybernetics, part b (cybernetics)* 26, no. 1 (1996): 29-41.
- [12] S.A. Mirjalili, S.M. Mirjalili, and A. Lewis. "Grey wolf optimizer." *Advances in Engineering Software* 69 (2014): 46-61.
- [13] S.A. Mirjalili, and A. Lewis. "The whale optimization algorithm." *Advances in Engineering Software* 95 (2016): 51-67.
- [14] X.S. Yang. "A new metaheuristic bat-inspired algorithm." In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, pp. 65-74. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.
- [15] X.S. Yang. "Nature-inspired optimization algorithms: Challenges and open problems." *Journal of Computational Science* 46 (2020): 101104.
- [16] Z.W. Geem, J.H. Kim, and G.V. Loganathan. "A new heuristic optimization algorithm: harmony search." *Simulation* 76, no. 2 (2001): 60-68.
- [17] G.Q. Huang, W.J. Zhao, and Q.Q. Lu. "Bat algorithm with global convergence for solving large-scale optimization problem." *Jisuanji Yingyong Yanjiu* 30, no. 5 (2013): 1323-1328.

- [18] R.Y. Nakamura, L.A. Pereira, K.A. Costa, D. Rodrigues, J.P. Papa, and X.S. Yang. "BBA: a binary bat algorithm for feature selection." In 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images, pp. 291-297. IEEE, 2012.
- [19] S.A. Mirjalili, S.M. Mirjalili, and X.S. Yang. "Binary bat algorithm." *Neural Computing and Applications* 25 (2014): 663-681.
- [20] G. Komarasamy, and A. Wahi. "An optimized K-means clustering technique using bat algorithm." *European Journal of Scientific Research* 84, no. 2 (2012): 263-273.
- [21] G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang. "A bat algorithm with mutation for UCAV path planning." *The Scientific World Journal* 2012, no. 1 (2012): 418946.
- [22] I. Fister Jr, D. Fister, and X.S. Yang. "A hybrid bat algorithm." arXiv preprint arXiv:1303.6310 (2013).
- [23] M.R. Chen, Y.Y. Huang, G.Q. Zeng, K.D. Lu, and L.Q. Yang. "An improved bat algorithm hybridized with extremal optimization and Boltzmann selection." *Expert Systems with Applications* 175 (2021): 114812.
- [24] T. Vu-Huu, S. Pham-Van, Q.H. Pham, and T. Cuong-Le. "An improved bat algorithms for optimization design of truss structures." In *Structures*, vol. 47, pp. 2240-2258. Elsevier, 2023.
- [25] M.R. Ramli, Z. Abal Abas, M.I. Desa, Z. Zainal Abidin, and M. Bader Alazzam. "Enhanced convergence of Bat Algorithm based on dimensional and inertia weight factor." *Journal of King Saud University-Computer and Information Sciences* 31, no. 4 (2019): 452-458.
- [26] K. Li, Y. Han, F. Ge, W. Xu, and L. Liu. "Tracking a dynamic invading target by UAV in oilfield inspection via an improved bat algorithm." *Applied Soft Computing* 90 (2020): 106150.
- [27] F. Xu, S. Zi, J. Wang, and J. Ma. "A computing offloading strategy for UAV based on improved bat algorithm." *Cognitive Robotics* 3 (2023): 265-283.
- [28] Y. Li, X. Li, J. Liu, and X. Ruan. "An improved bat algorithm based on lévy flights and adjustment factors." *Symmetry* 11, no. 7 (2019): 925.
- [29] S. Yilmaz, and E.U. Kucuksille. "Improved bat algorithm (IBA) on continuous optimization problems." *Lecture Notes on Software Engineering* 1, no. 3 (2013): 279.
- [30] Y. Luo, C. Wu, Y. Leng, N. Huang, L. Mao, and J. Tang. "Throughput optimization for NOMA cognitive relay network with RF energy harvesting based on improved bat algorithm." *Mathematics* 10, no. 22 (2022): 4357.
- [31] F. Soleimanian Gharehchopogh. "An Improved Bat Algorithm with Grey Wolf Optimizer for Solving Continuous Optimization Problems." *Journal of Advances in Computer Engineering and Technology* 4, no. 3 (2020): 119.
- [32] H. Salimi. "Stochastic fractal search: a powerful metaheuristic algorithm." *Knowledge-Based Systems* 75 (2015): 1-18.
- [33] Z.J. Li. "Improved bat algorithm based on grouping evolution and hybrid optimization." *Math. Pr. Th* 50 (2020): 141-149.
- [34] T. Vu-Huu, S. Pham-Van, Q.H. Pham, and T. Cuong-Le. "An improved bat algorithms for optimization design of truss structures." In *Structures*, vol. 47, pp. 2240-2258. Elsevier, 2023.
- [35] S. Yu, J. Zhu, and C. Lv. "A quantum annealing bat algorithm for node localization in wireless sensor networks." *Sensors* 23, no. 2 (2023): 782.