دوره ۱۳، شماره بهار و تابستان ۱۳۹۹

مجله فناوری اطلاعات در طراحی مهندسی
Information Technology in Engineering Design
http://ited.sinaweb.net

یک الگوریتم جستجوی کلاغ بهبود یافته مبتنی بر یادگیری معکوس برای حل مسائل بهینه سازی

عبدالرضا حاتم لو *[۱] محمد علیزاده چناقلو [۲]

(۱) گروه مهندسی کامپیوتر، واحد خوی، دانشگاه آزاد اسلامی، خوی، ایران

(۲) گروه مهندسی کامپیوتر، واحد خوی، دانشگاه آزاد اسلامی، خوی، ایران

(تاریخ دریافت : ۱۳۹۹/۱/۱۲ تاریخ پذیرش : ۱۳۹۹/۱۲/۲۶)

چکیده

الگوریتم جستجوی کلاغ یکی از الگوریتم‌های فرا ابتکاری مبتنی بر رفتار کلاغ‌ها در طبیعت بوده که برای حل مسائل بهینه‌سازی ارائه شده است. این الگوریتم مبتنی بر جمعیت است و با ایده این‌که کلاغ‌ها مواد غذایی اضافی خود را پنهان کرده و در صورت نیاز آن را دوباره پیدا و مصرف می‌کنند، طراحی و پیاده‌سازی شده است. تمام الگوریتم‌های فرا ابتکاری ازجمله الگوریتم جستجوی کلاغ ممکن است دارای مشکلاتی مانند همگرایی زودرس و همگرایی دیررس و گیر افتادن در بهینه محلی باشند که برای حل این مشکلات از ترکیب روش‌های دیگر بهره می‌برند. برای این منظور در این مقاله یک الگوریتم جستجوی کلاغ بهبودیافته مبتنی بر یادگیری معکوس ارائه‌شده است. در روش پیشنهادی از روش مبتنی بر معکوس برای افزایش سرعت همگرایی و دقت الگوریتم جستجوی کلاغ در دو مرحله استفاده‌شده است. عملکرد الگوریتم پیشنهادی با استفاده از ۲۰ تابع محک با سایر الگوریتم‌ها مقایسه گردیده است. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی در مقایسه با الگوریتم جستجوی کلاغ استاندارد و سایر الگوریتم‌های فرا ابتکاری پایه از لحاظ معیارهای بهترین جواب، بدترین جواب، میانگین و انحراف معیار جوابها بهتر عمل کرده است.

واژگان کلیدی: الگوریتم جستجوی کلاغ بهبودیافته، یادگیری معکوس، مسائل بهینه‌سازی.

* **عهده‌دار مکاتبات:**

عبدالرضا حاتم لو

نشانی: گروه مهندسی کامپیوتر، واحد خوی، دانشگاه آزاد اسلامی، خوی، ایران

پست الکترونیک: hatamlou@iaukhoy.ac.ir **تلفن : ۰۹۱۴۴۶۲۷۸۹۳**

۱- مقدمه

به دلیل افزایش پیچیدگی مسائل بهینه‌سازی پیوسته و ناتوانی روش‌های ریاضی برای ارائه راه‌حل بهینه، الگوریتم‌های فرا ابتکاری راهکار مناسبی برای مسائل بهینه‌سازی پیوسته هستند [۱-۳]. روش‌های ریاضی در حل بسیاری از مسائل علمی و مهندسی مورد استفاده قرارگرفته‌اند و دامنه بسیار وسیعی از مسائل مختلف را پوشش می‌دهند اما روش‌های ریاضی با وجود کارایی دقیق، هنوز هم در حل بسیاری از مسائل بهینه‌سازی با مشکلات زیادی مواجه هستند [۴]. روش های فرا ابتکاری در مقایسه با تکنیک های بهینه سازی سنتی قابلیت های بهتری برای پرهیز از بهینه محلی دارند. دلیل این موضوع ماهیت تصادفیِ روش های فرا ابتکاری است که به آنها اجازه می‌دهد از راه حل های بهینه بیرون بیایند و کل فضای جستجو را بگردند. فضای جستجوی مسائل واقعی معمولا ناشناخته و بسیار پیچیده است و بهینه های محلی زیادی در آن وجود دارد، به طوری که روش های فرا ابتکاری برای بهینه سازی این مسائل واقعی چالش انگیز گزینه های خوبی هستند [۵].

در مقاله [۳] یک روش فرا ابتکاری جدید به نام الگوریتم جستجوی کلاغ معرفی شده است که این روش مبتنی بر رفتار هوشمندانه کلاغ‌ها بوده و در راستای حل مسائل بهینه‌سازی توسعه داده شده است. الگوریتم جستجوی کلاغ یک روش مبتنی بر جمعیت است و ایده اصلی شکل‌گیری این روش از این نظریه که کلاغ‌ها مواد غذایی اضافی خود را در مخفیگاه‌هایی پنهان کرده و هنگامی که به آن مواد غذایی نیاز دارند، آن‌ها را مجدداً یافته و مصرف می‌کنند، الهام گرفته شده است. نتایج حاصل از الگوریتم جستجوی کلاغ با نتایج بدست آمده از الگوریتم‌های گوناگون مقایسه شده است که نتایج این شبیه‌سازی نشان می‌دهد که الگوریتم جستجوی کلاغ در مقایسه با سایر الگوریتم‌ها نتایج قابل قبولی از خود ارائه داده است.تمام الگوریتم های فرا ابتکاری از جمله الگوریتم جستجوی کلاغ ممکن است دارای مشکلاتی مانند همگرایی زودرس و همگرایی دیررس و گیر افتادن در دام محلی هستند. روش یادگیری مبتنی بر معکوس که ابتدا توسط تیزهوش و همکاران معرفی شده است [۶]. یادگیری مبتنی بر معکوس یک مفهوم موثر برای ارتقای روشهای بهینه سازی مختلف است. ایده اصلی یادگیری مبتنی بر معکوس، بررسی همزمان فضای مخالف متناظر به عنوان مجموعه دوم از راه حل های کاندید برای دستیابی به راه حل بهتری نسبت به راه حل فعلی است. این روش ثابت کرده است که راه حل ایجاد شده در فضای مخالف، فرصتی برای نزدیک شدن به راه حل مطلوب جهانی نسبت به راه حل انتخاب شده به صورت تصادفی تولید شده است. چندین پژوهش ثابت شده است که روش یادگیری مبتنی معکوس باعث افزایش تنوع جمعیت، بهبود سرعت همگرایی، بهبود دقت و سرعت و بهبود کارایی شده است [۷-۹]. بنابراین در این مقاله یک یک الگوریتم جستجوی کلاغ بهبود یافته مبتنی یادگیری معکوس برای حل مسائل بهینه سازی ارائه خواهد شد.

در ادامه مقاله ابتدا در بخش ۲ به بررسی الگوریتم‌ها فرا ابتکاری بهبود یافته پرداخته شده است. در بخش ۳، مفاهیم اولیه الگوریتم کلاغ و روش پیشنهادی تشریح می‌شود. در بخش ۴، کارایی و عملکرد الگوریتم پیشنهادی با استفاده توابع محک استاندارد مورد تست و برسی قرارگرفته و با دیگر الگوریتم‌ها مقایسه شده است. در بخش پایانی مقاله نتیجه‌گیری کلی و کارهای آینده ارائه می‌شود.

۲- کار های قبلی

یک الگوریتم تکاملی تفاضلی مبتنی بر معکوس برای زمانبندی جریان وقفه بر مبنای اندازه گیری تنوع در سال ۲۰۱۳ ارائه شده است [۱۰]. در این مقاله برای بهبود ویژگی بهینه سراسری جهانی الگوریتم تکاملی تفاضلی، یک رویکرد الگوریتم تکاملی تفاضلی مبتنی بر اندازه گیری تنوع جمعیت برای تنظیم نرخ متقابل و همچنین برای بهبود نرخ همگرایی الگوریتم تکاملی تفاضلی، الگوریتم تکاملی تفاضلی مبتنی بر معکوس که یادگیری مبتنی بر معکوس را برای شروع و پرش تولید برای بهینه سازی راه حل جهانی به کار می‌گیرد، پیشنهاد شده است. در نهایت شبیه سازی‌ها و مقایسه‌ها براساس معیارهای زمانبندی جریان وقفه انجام شد که نشان می‌دهد که الگوریتم پیشنهادی هم موثر و کارآمد است.

در سال ۲۰۱۶ الگوریتم گروه میگوها مبتنی بر معکوس برای حل مسئله پخش بار اقتصادی بار ارائه شده است [۱۱]. در این مقاله، الگوریتم گروه میگوها با یادگیری مبتنی بر معکوس برای بهبود سرعت همگرا و دقت الگوریتم گروه میگوها پایه ترکیب شده است. روش پیشنهادی برای رسیدن به نتایج مطلوب در هنگام کار با محدودیت های عملیاتی در مسئله پخش بار اقتصادی و بارگذاری نقطه شیر از این بهبود استفاده کرده است. اثربخشی روش پیشنهادی با انجام آزمونهای عددی بر روی پنج سیستم مختلف استاندارد بررسی و تایید شده است. مقایسه نتایج عددی با روشهای به دست آمده دیگر، اعتبار و استحکام الگوریتم پیشنهادی را نسبت به سایر روشهای موجود تأیید می‌کند.

در سال ۲۰۱۷ یک الگوریتم بهبود یافته سینوس و کوسینوس مبتنی بر روش مبتنی بر معکوس برای حل مسائل بهینه سازی ارائه شده است [۷]. در واقع این مقاله نسخه بهبود یافته الگوریتم سینوس و کوسینوس را ارائه می‌دهد که از یادگیری مبتنی بر معکوس به عنوان مکانیسم برای جستجوی بهتر فضای جستجو و تولید راه حل‌های دقیق تر ارائه استفاده می‌کند. الگوریتم بهبود یافته پیشنهادی بر چندین توابع محک ریاضی در حالت‌های مختلف و همچنین بر روی چندین مسئله بهینه سازی واقعی آزمایش شد. نتایج آزمایش نشان می‌دهد که روش پیشنهادی در پیدا کردن راه حل‌های بهینه در فضای جستجوی پیچیده بهتر از سایر الگوریتم‌های مقایسه ای عمل می‌کند. همچنین در تحقیق دیگری یک الگوریتم شیر مورچه مبتنی معکوس را ارائه شده است [۱۲]. در این مقاله برای افزایش اکتشاف الگوریتم شیر مورچه از دو روش استفاده شده است که در روش اول از توزیع لاپلاس به جای توزیع یکنواخت بهره برده شده است و در روش دوم از روش یادگیری مبتنی بر معکوس برای جستجوی فضای مخالف بهره برده شده است. در این مقاله از از ۲۷ تابع محک ریاضی، از جمله طیف گسترده ای از ویژگی‌های مختلف و ابعاد مختلف برای تایید نتایج استفاده شده است. نتایج نشان می - دهد که الگوریتم پیشنهادی توانسته است بهتر مسائل را حل کند.

در سال ۲۰۱۸ یک الگوریتم بهبود یافته جستجوی هارمونی مبتنی بر روش مبتنی بر معکوس برای حل مسائل بهینه سازی ارائه شده است [۹]. در این مقاله برخی از بهبود را بر اساس نظریه یادگیری مبتنی بر معکوس برای سرعت بخشیدن به نرخ همگرایی الگوریتم‌های فرا ابتکاری معرفی کرده است. البته طرح پیشنهادی با استفاده از یک روش پیشنهادی برای برآورد مقیاس قطعی در هنگام محاسبه یک راه حل کاندید استفاده شده است. در این مقاله، در نهایت روش یادگیری مبتنی بر معکوس در چارچوب الگوریتم جستجوی هارمونی تعبیه شده است. یک مجموعه جامع از توابع محک برای ارزیابی روش پیشنهادی استفاده شد. نتایج حاصل از این آزمایشات بسیار امیدوار کننده بود.

در تحقیق دیگر بهبود الگوریتم بهینه سازی ملخ با استفاده از یادگیری مبتنی بر معکوس ارائه شده است [۱۳]. در این مقاله یک نسخه بهبود یافته از الگوریتم بهینه سازی ملخ را بر

اساس یادگیری مبتنی بر معکوس به نام OBLGOA برای حل توالی بهینه سازی محک و مشکلات مهندسی ارائه می-دهد. الگوریتم پیشنهاد شده OBLGOA شامل دو مرحله است: مرحله اول، جمعیت اولیه را تولید می‌کند و در مقابل آن با استفاده از یادگیری مبتنی بر معکوس؛ و مرحله دوم از یادگیری مبتنی بر معکوس به عنوان یک مرحله اضافی برای به روز رسانی جمعیت الگوریتم بهینه سازی ملخ در هر تکرار استفاده می‌کند. با این حال، یادگیری مبتنی بر معکوس تنها به نیمی از راه حل‌ها برای کاهش پیچیدگی زمان اعمال می‌شود. برای بررسی عملکرد OBLGOA پیشنهاد شده، شش مجموعه سری آزمایشی انجام شد و شامل بیست و سه توابع محک و چهار مشکل مهندسی استفاده شد. آزمایشات نشان داد که نتایج الگوریتم پیشنهاد شده برتر از ۱۰ الگوریتم شناخته شده در این دامنه بود. در نهایت، نتایج به دست آمده ثابت می‌کند که الگوریتم OBLGOA می‌تواند نتایج رقابتی را برای مشکلات مهندسی بهینه سازی در مقایسه با الگوریتم-های پیشرفته تر ارائه دهد.

۳- روش پیشنهادی

در این بخش به توضیح کامل در مورد روش پیشنهادی و نحوه استفاده از روش مبتنی معکوس در الگوریتم جستجوی کلاغ خواهیم پرداخت. ما این بخش را به چند زیر بخش جداگانه تقسیم کردیم که به تشریح الگوریتم جستوجوی کلاغ، روش مبتنی بر معکوس، نحوه بهبود الگوریتم جستجوی کلاغ به کمک روش مبتنی و نهایت یک سری توابع محک برای ارزیابی روش پیشنهادی خواهیم پرداخت.

۳-۱ الگوریتم های جستجوی کلاغ

کلاغ‌ها (گونه کلاغ یا کلاغیان) به عنوان باهوش‌ترین پرندگان در نظر گفته می‌شوند. آن‌ها نسبت به جثه خود، بزرگ‌ترین مغز را در میان جانوران دارند. با توجه به نسبت مغز-به-بدن، مغز آن‌ها تنها اندکی از مغز انسان کوچکتر است. مدارک و شواهد فراوانی مبنی بر هوشمندی کلاغ‌ها وجود داردعلاوه بر این، آن‌ها می‌توانند از ابزار استفاده کرده، به روش‌های پیشرفته با یکدیگر ارتباط برقرار کرده و مواد غذایی پنهان‌شده خود را پس از ماه‌ها بازیابند [۳, ۱۴, ۱۵].

- کلاغ‌ها به شکل گروهی زندگی می‌کنند.
- کلاغ‌ها موقعیت مکانی مخفیگاه‌های مواد غذایی خود را به خاطر می‌سپارند.
- کلاغ‌ها به منظور سرقت، یکدیگر را تعقیب می‌کنند.
- کلاغ‌ها از ذخایر خود، در مقابل احتمال به سرقت رفتن آن‌ها، محافظت می‌کنند.

فرض می‌کنیم که یک محیط d بُعدی شامل تعدادی کلاغ وجود دارد. تعداد کلاغ‌ها (اندازه دسته) برابر N و موقعیت مکانی کلاغ شماره i در زمان (تکرار) $iter$ در فضای جستجو به وسیله بردار $x^{i,iter}$ به ازای $(i = 1,2,\dots,N; iter = 1,2,\dots,iter_{max})$ نمایش داده می‌شود که $x^{i,iter} = [x_1^{i,iter}, x_2^{i,iter}, \dots, x_d^{i,iter}]$ و $iter_{max}$ تعداد کل تکرارها است. هر کلاغ دارای حافظه‌ای است که با استفاده از آن، موقعیت مکانی مخفیگاه خود را به خاطر می‌سپارد. در تکرار $iter$، موقعیت مکانی مخفیگاه کلاغ i با $m^{i,iter}$ نمایش داده می‌شود. این بهترین مکانی است که کلاغ i تا کنون کشف کرده است. در واقع، موقعیت مکانی بهترین تجربه هر کلاغ در حافظه آن ذخیره می‌گردد. کلاغ‌ها در محیط اطراف خود حرکت کرده و به جستجوی منابع غذایی (مخفیگاه‌ها) بهتر می‌پردازند.

فرض کنید که در تکرار $iter$، کلاغ j بخواهد مخفیگاه خود، یعنی $m^{j,iter}$ را بازرسی نماید. در این تکرار، کلاغ i تصمیم می‌گیرد که به منظور دستیابی به مخفیگاه کلاغ j، آن را تعقیب نماید. در این حالت، دو وضعیت ممکن است پیش آید:

وضعیت ۱: کلاغ j نمی‌داند که کلاغ i در حال تعقیب آن است. در نتیجه، کلاغ i مخفیگاه کلاغ j را می‌یابد. در این حالت، موقعیت مکانی جدید کلاغ i به صورت زیر بدست می‌آید [۳]:

$$x^{i,iter+1} = \begin{cases} x^{i,iter} + r_i \times fl^{i,iter}(m^{j,iter} - x^{i,iter}) & r_j \geq AP_{j,Iter} \\ a\ random\ location & otherwise \end{cases} \quad (۲)$$

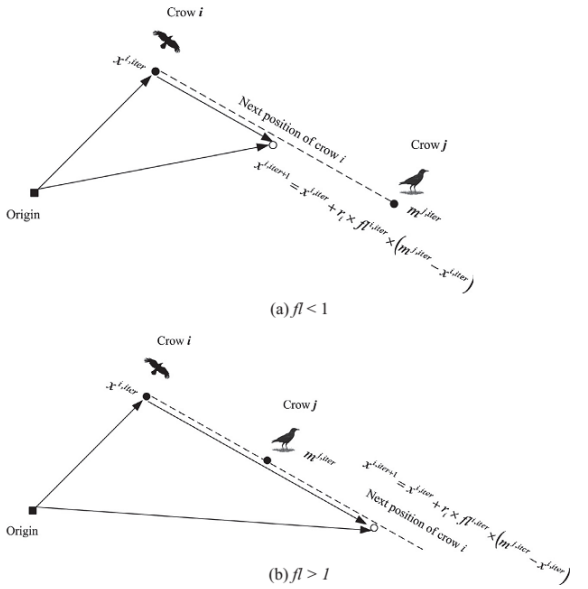
که r_j عددی تصادفی با توزیع یکنواخت بین اعداد صفر و ۱ است و $AP^{i,iter}$ نشان‌دهنده احتمال آگاه شدن کلاغ j در تکرار $iter$ است. الگوریتم‌های فرا ابتکاریباید تعادل مناسبی میان تنوع‌بخشی و گسترده‌سازی برقرار سازند. در الگوریتم جستجوی کلاغ، دو مشخصه تنوع‌بخشی و گسترده‌سازی، اساساً به وسیله مؤلفه‌ای به نام احتمال آگاهی (AP) کنترل و تنظیم می‌شوند. با کاهش مقدار احتمال آگاهی، الگوریتم جستجوی کلاغ به سمت انجام جستجوی محلی در مکانی که در آن پاسخی شایسته یافت شده، سوق داده می‌شود. در نتیجه، استفاده از مقادیر کوچک برای AP، میزان گسترده‌سازی را افزایش می‌دهد. از سوی دیگر، با افزایش احتمال آگاهی، احتمال جستجو کردن مکان‌های مجاور پاسخ‌های فعلی کاهش یافته و الگوریتم جستجوی کلاغ به سمت کاوش فضای جستجو به صورت سراسری (ایجاد آرایش تصادفی) هدایت می‌گردد. بنابراین، استفاده از مقادیر بزرگ برای AP، میزان تنوع‌بخشی را افزایش می‌دهد [۳].

$$x^{i,iter+1} = x^{i,iter} + r_i \times fl^{i,iter}(m^{j,iter} - x^{i,iter}) \quad (۱)$$

که r_i یک عدد تصادفی با توزیع یکنواخت بین دو عدد صفر و ۱ بوده و $fl^{j,iter}$ نشان‌دهنده طول پرواز کلاغ i در تکرار $iter$ است. شکل (۱) نمود تصویری این وضعیت و همچنین تأثیر fl بر توانایی جستجو را نمایش می‌دهد. مقادیر کوچک fl منجر به جستجوهای محلی (در مجاورت $x^{i,iter}$) و مقادیر بزرگ نیز منتج به جستجوهای سراسری (دور از $x^{i,iter}$) خواهد شد. همانطور که در شکل ۱ (a) مشاهده می‌گردد، در صورتی که مقدار fl کوچکتر از ۱ انتخاب گردد، مکان بعدی کلاغ i بر روی خط‌چین میان $x^{i,iter}$ و $m^{i,iter}$ قرار می‌گیرد. همچنین، با توجه به شکل ۱ (b)، اگر مقدار fl بیشتر از ۱ انتخاب گردد، مکان بعدی کلاغ i بر روی خط‌چینی که ممکن است از $m^{i,iter}$ عبور کند، قرار می‌گیرد.

وضعیت ۲: کلاغ j می‌داند که کلاغ i در حال تعقیب آن است. در نتیجه، کلاغ j به منظور پیشگیری از مورد سرقت واقع شدن ذخایر خود، کلاغ i را، با رفتن به مکانی دیگر از فضای جستجو، فریب می‌دهد [۳].

به طور کلی، رابطه‌های ۱ و ۲ را می‌توان به صورت زیر بیان نمود:



شکل ۱: فلوچارت وضعیت ۱ در الگوریتم جستجوی کلاغ (a) $fl < 1$ و (b) $fl > 1$. کلاغ i می‌تواند در تمامی نقاط خط‌چین قرار گیرد[۳].

- **پیاده‌سازی الگوریتم جستجوی کلاغ در مسئله بهینه‌سازی**

در این بخش، فرآیند گام‌به‌گام پیاده‌سازی الگوریتم جستجوی کلاغ شرح داده می‌شود.

گام ۱- مقداردهی اولیه مسئله و مؤلفه‌های قابل‌تنظیم: مسئله بهینه‌سازی، متغیرهای تصمیم‌گیری و قیود در این مرحله تعریف می‌گردند. سپس مؤلفه‌های قابل‌تنظیم الگوریتم جستجوی کلاغ (شامل اندازه دسته (N)، تعداد کل تکرارها ($iter_{max}$)، طول پرواز (fl) و احتمال آگاهی (AP)) مقداردهی می‌شوند.

گام ۲- مقداردهی اولیه موقعیت مکانی و حافظه کلاغ‌ها: تعداد N کلاغ، به عنوان اعضای دسته، به صورت تصادفی در یک فضای جستجوی d بُعدی قرار می‌گیرند. هر کلاغ نشان‌دهنده یک پاسخ شدنی از مسئله و d نیز نماینده تعداد متغیرهای تصمیم‌گیری است.

$$\boldsymbol{Position\ Crow} = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & \cdots & X_{1,d} \\ X_{2,1} & X_{2,2} & \cdots & \cdots & X_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{n,1} & X_{n,1} & & & X_{n,d} \end{bmatrix} \quad (۳)$$

مقدار اولیه حافظه هر کلاغ تعیین می‌گردد. با توجه به اینکه در اولین تکرار، کلاغ‌ها هیچ تجربه‌ای ندارند، فرض می‌شود که آن‌ها مواد غذایی خود را در موقعیت‌های مکانی اولیه خود مخفی کرده‌اند.

$$\mathbf{Memory\ Crow} = \begin{bmatrix} \mathbf{M_{1,1}} & \mathbf{M_{1,2}} & \cdots & \cdots & \mathbf{M_{1,d}} \\ \mathbf{M_{2,1}} & \mathbf{M_{2,2}} & \cdots & \cdots & \mathbf{M_{2,d}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{M_{n,1}} & \mathbf{M_{n,1}} & & & \mathbf{M_{n,d}} \end{bmatrix} \quad (٤)$$

گام ۳- ارزیابی مقادیر تابع هدف (میزان شایستگی): کیفیت موقعیت مکانی هر کلاغ، با قرار دادن مقادیر متغیرهای تصمیم‌گیری در تابع هدف محاسبه می‌گردد.

گام ۴- تولید موقعیت‌های مکانی جدید: کلاغ‌ها موقعیت‌های مکانی جدیدی را به صورت زیر در فضای جستجو تولید می‌کنند: فرض کنید کلاغ i بخواهد موقعیتی جدید ایجاد کند. بدین منظور، این کلاغ به صورت تصادفی یکی از کلاغ‌های موجود در دسته را انتخاب می‌کند (به عنوان مثال کلاغ j) و آن را تعقیب کرده تا مکان مخفیگاه آن کلاغ را (m^j) بیابد. موقعیت جدید کلاغ i به وسیله رابطه (۲) محاسبه می‌شود. این فرآیند برای تمامی کلاغ‌ها اجرا می‌شود.

گام ۵- بررسی قابلیت اجرای موقعیت‌های مکانی جدید: قابلیت اجرای موقعیت‌های مکانی جدید هر کلاغ بررسی می‌شود. در صورتی که موقعیت جدید هر کلاغ شدنی و قابل‌اجرا باشد، آن کلاغ موقعیت مکانی خود را به‌روزرسانی می‌کند؛ در غیر این صورت، در موقعیت مکانی فعلی خود باقی مانده و هیچ حرکتی به منظور ایجاد مکان جدید انجام نمی‌دهد

گام ۶- ارزیابی تابع هدف (شایستگی) موقعیت‌های مکانی جدید: مقدار تابع هدف برای موقعیت جدید تمامی کلاغ‌ها محاسبه می‌گردد.

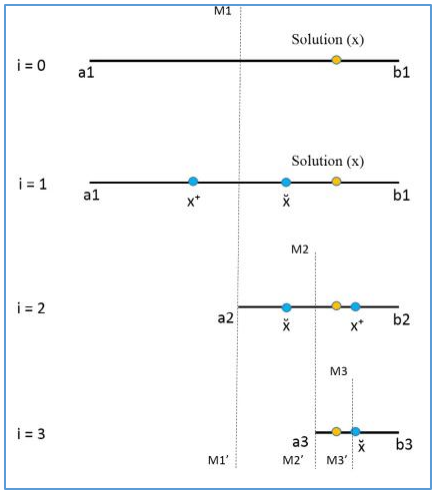
گام ۷- به‌روزرسانی حافظه: کلاغ‌ها حافظه خود را به صورت زیر به‌روزرسانی می‌کنند: $f(.)$ نشان‌دهنده مقدار تابع هدف است.

$$\boldsymbol{m^{i,iter}} = \begin{cases} \boldsymbol{x^{i,iter}} & \boldsymbol{Ff(x^{i,iter}) > f(m^{i,iter})} \\ \boldsymbol{m^{i,iter}} & \boldsymbol{otherwise} \end{cases} \quad (۵)$$

گام ۸: بررسی معیار خاتمه: گام‌های ۴ الی ۷، تا زمان رسیدن به $iter_{max}$ تکرار می‌شوند. هنگامی که معیار خاتمه فراهم گردد، بهترین موقعیت مکانی ذخیره‌شده در حافظه‌ها، که بر حسب مقدار تابع هدف تعیین گردیده، به عنوان پاسخ مسئله بهینه‌سازی انتخاب می‌گردد.

۳-۲روش مبتنی بر معکوس

روش یادگیری مبتنی بر معکوس که برای بهبود هم‌گرایی روش‌های فرا ابتکاری برای یافتن راه‌حل جهانی مسئله بهینه‌سازی استفاده می‌شود [۷]. به طور کلی، روش‌های فرا ابتکاری با تولید جمعیت اولیه (که حاوی راه‌حل‌های تصادفی است) به عنوان تلاش برای یافتن راه‌حل بهینه شروع به کار می‌کنند. این راه‌حل‌های اولیه به طور تصادفی یا بر اساس دانش قبلی از قبیل مشخص کردن جستجوی دامنه یا معیارهای دیگر تولید می‌شوند. با این حال، در صورت فقدان این دانش، این روش‌ها نمی‌توانند به یک راه‌حل بهینه همگرا شوند، زیرا آن‌ها به طور تصادفی در فضای جستجو کار می‌کنند. در شکل ۲ روش مبتنی بر معکوس برای نزدیک شدن به جواب بهینه نشان داده شده است.



شکل ۲: حل یک مساله با استفاده از روش مبتنی بر معکوس با نصف کردن فاصله جستجو بر اساس نزدیک شدن به جواب بهینه [۱۶]

البته برای بهبود الگوریتم پیشنهادی روش مبتنی بر معکوس را می‌توان به دو صورت استفاده کرد.

- برای تولید جمعیت
- برای تولید جمعیت بعدی در هنگام ترکیب دو الگوریتم

در واقع در حالت اول برای تولید جمعیت اولیه استفاده می‌شود و در حالت دوم بیشتر هدف این است که جمعیت در هنگام تکرار نسل‌ها از کیفیت بهتری برخوردار شود. بنابراین در بخش بعدی در روش الگوریتم جستجوی کلاغ مبتنی بر معکوس از هر دو مدل برای پیاده سازی روش پیشنهادی بهره خواهیم برد. در ادامه روش مبتنی بر معکوس را توضیح می‌دهیم.

تعریف نقطه قرینه: اگر $P(x_1, x_2, \ldots, x_n)$ یک راه حل در فضا n بعدی باشد و به شرط x_1, x_2 و $x_n \in R$ and $xi \in [a_i, b_i]\ \forall i \in 1, 2 \ldots n$ باشد فضای مخالف ان به صورت رابطه ۶ تعریف می‌شود [۳۴].

$$x_i^+ = a_i + b_i - x_i.$$
$$P^+(x_1^+, x_2^+, \dots, x_n^+), \qquad (۶)$$

در رابطه ۶، x_i^+ نشان دهنده یک راه حل مبتنی بر معکوس به دست آمده است و a_i کمترین مقدار برای بعد i ام راه حل و b_i کمترین مقدار برای بعد i ام راه حل را نشان می‌دهد. اکنون بهینه سازی بر اساس بهینه سازی مبتنی بر معکوس با استفاده از تعریف نقطه قرینه در رابطه ۶ را تعریف می‌کنیم که بعد از تولید نقطه قرینه از روی راه حل کاندید می‌توان f($\breve{p}$)> =f (p) نشان داد که آنگاه راه حل $\breve{p}$ می‌تواند جایگزین نقطه p شده و در غیر این صورت از نقطه p شروع می‌کنیم.

۳-۳ تولید راه حل مبتنی بر معکوس

در بخش نحوه تولید یک راه حل با استفاده از روش مبتنی بر معکوس کامل تشریح شد و بنابراین ما در ادامه از این روش برای تولید جمعیت مبتنی بر معکوس استفاده خواهیم کرد. در نتیجه می‌توان جمعیت اولیه تصادفی را به صورت رابطه ۷ تولید کرد.

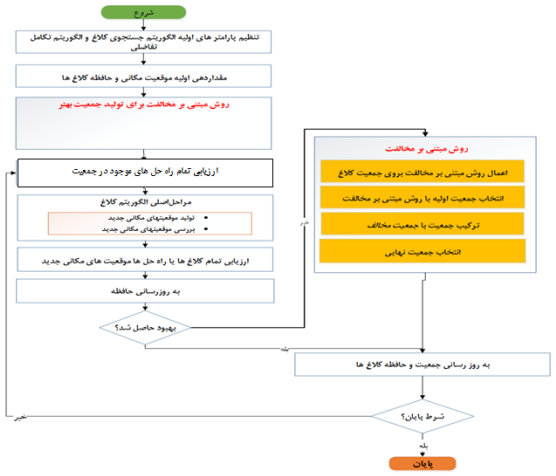
$$M = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \vdots & \dots & \dots & \dots \\ x_1^N & x_2^N & \dots & x_d^N \end{bmatrix} \begin{matrix} f(x^1) \\ f(x^2) \\ \vdots \\ f(x^N) \end{matrix} \qquad (۷)$$

طبق رابطه ۷ ما هر یک از راه حل تصادفی را بین یک محدود خاص تولید کرده و جمعیت تصادفی را در اینجا با M نام گذاری کردیم که اعمال رابطه ۱ می‌توان کل راه حل‌های موجود در جمعیت را به فضای مخالف برد. بنابراین با توجه به رابطه ۲ و جمعیت تصادفی M، جمعیت مبتنی بر معکوس جدید در رابطه ۸ نشان داده شده است.

$$p = \begin{bmatrix} \widetilde{x_1^1} & \widetilde{x_2^1} & \dots & \widetilde{x_{r1}^1} \\ \widetilde{x_1^2} & \widetilde{x_2^2} & \dots & \widetilde{x_{r1}^1} \\ \vdots & \dots & \dots & \dots \\ \widetilde{x_1^N} & \widetilde{x_2^N} & \dots & \widetilde{x_{rd}^N} \end{bmatrix} \begin{matrix} f(\widetilde{x_r^1}) \\ f(\widetilde{x_r^2}) \\ \vdots \\ f(\widetilde{x_r^N}) \end{matrix} \qquad (۸)$$

۴-۳ الگوریتم جستجوی کلاغ مبتنی بر معکوس

الگوریتم جستجوی کلاغ یکی از الگوریتم‌های فرا ابتکاری مبتنی بر رفتار کلاغ‌ها در طبیعت بوده که برای حل مسائل بهینه‌سازی معرفی‌شده است. این الگوریتم مبتنی بر جمعیت است و با ایده این که کلاغ‌ها مواد غذایی اضافی خود را پنهان کرده و در صورت نیاز آنها را دوباره پیدا و مصرف می‌کنند، طراحی و پیاده سازی شده است. این الگوریتم نیز همانند سایر الگوریتم‌های فرا ابتکاری ممکن است دارای همگرایی زودرس یا دیررس باشد که با اضافه کردن روش مبتنی بر معکوس می‌توان باعث افزایش تنوع جمعیت، بهبود سرعت همگرایی، بهبود دقت و سرعت و بهبود کارایی این الگوریتم شد. ما در ابتدا فلوچارت روش پیشنهادی را در شکل ۳ نشان داده ایم و در ادامه در مورد روش پیشنهادی بیشتر توضیح خواهیم داد.



شکل ۳: فلوچارت روش پیشنهادی

در شکل۳، الگوریتم جستجوی کلاغ مبتنی بر معکوس نشان داده شده است. همان‌طور که مشاهده می‌کنید در روش پیشنهادی ما دو حالت از روش مبتنی معکوس برای بهبود الگوریتم جستجوی کلاغ استفاده کردیم. در حالت اول از روش مبتنی بر معکوس برای تولید بهتر جمعیت اولیه در شروع الگوریتم جستجوی کلاغ و در حالت دوم در حین تکرار الگوریتم جستجوی کلاغ برای

افزایش کارایی الگوریتم جستجوی کلاغ استفاده شده است.

۴- نتایج ارزیابی

برای اینکه الگوریتم‌های استاندارد زنبورعسل مصنوعی و الگوریتم جستجو کلاغ و جستجوی هارمونی، خفاش و تکامل تفاضلی و همینطور الگوریتم پیشنهادی بطور عادلانه مقایسه شوند، تمام پارامترهای الگوریتم‌های فرا ابتکاری باید از لحاظ تعداد تکرار و همینطور تعداد جمعیت با یکدیگر مساوی باشند؛ بنابراین همانطور که در جدول ۱ مشاهده می‌کنید، پارامترهای اولیه برای الگوریتم جستجو کلاغ و دیگر الگوریتم‌ها مقداردهی شده است.

جدول۱: مقادیر پارامترهای الگوریتم پیشنهادی و الگوریتم‌های مورد مقایسه

الگوریتم	پارامترها و مقادیر
جستجوی هارمونی	bw=0.2,HMCR=0.5,PAR=0.3, population size is 50
زنبور عسل مصنوعی	limit = 5D, population size is 50, Nonlooker=50
الگوریتم خفاش	r=0.5;A=0.5;population size is 50
الگوریتم جستجو کلاغ	Ap=0.8;population size is 50
الگوریتم پیشنهادی	Ap=0.8;and population size is 50

الگوریتم پیشنهادی و سایر الگوریتم‌ها با استفاده از ۲۰ تابع محک استاندارد در زمینه بهینه سازی مورد تست و بررسی می‌گیرند. این توابع محک استاندارد به عنوان مسائل بهینه سازی پیوسته به صورت کمینه سازی معرفی شده اند. برای ارزیابی و آزمایش تمام الگوریتم‌های فرا ابتکاری چه به عنوان روش جدید و ترکیبی و بهبود یافته، از این توابع استفاده می‌شود که می‌توانند نرخ همگرایی و میزان قدرت الگوریتم - های فرا ابتکاری در پیدا کردن جواب را نشان دهند. مشخصات توابع محک استفاده شده در جدول ۲ نماش داده شده است.

جدول۲: مشخصات توابع محک استفاده شده [15]

No	Formula	Bounds	D
F01	$\sum_{i=1}^{d} x_i^2$	$[-100.100]^d$	2
F02	$\sum_{i=1}^{d} \lvert x_i \rvert^{i+1}$	$[-100.100]^d$	2
F03	$\sum_{i=1}^{d} \sum_{i=1}^{i} x_i^2$	$[-65.65]^d$	2
F04	$\sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \left(\frac{x_i}{\sqrt{i}}\right)$	$[-600.600]^d$	4
F05	$10d + \sum_{i=1}^{d} x_i^2 - 10\cos(2\pi x_i)$	$[-5.12.5.12]^d$	4

F06	$sin^2(\pi w_1) + \sum_{i=1}^{d+1}(w_i - 1)^2[1 + 10sin^2(\pi w_1 + 1)] + (w_d - 1)^2[1 + 10sin^2(\pi w_d)]$ Where $w_i = 1 + (x_i - 1)/4$	$[-5.12.5.12]^d$	4
F07	$20exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right) - exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i\right) + 20 + \exp(1)$	$[-32.32]^d$	8
F08	$418.9829 - \sum_{i=1}^{d} x_i \sin(\sqrt{\lvert x_i \rvert})$	$[-500.500]^d$	8
F09	$\sum_{i=1}^{d-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)]$	$[-10.10]^d$	8
F10	$\sum_{i=1}^{d} x_i^2 + (\sum_{i=1}^{d} 0.5ix_i)^2 + 0.5ix_i)^4$	$[-5.10]^d$	8
F11	$(x_i - 1)^2 + \sum_{i=2}^{d} 2x_i^2 - x_{i-1})^2$	$[-10.10]^d$	8
F12	$-\sum_{i=1}^{d} \sin(x_i) sin^{20}\left(\frac{ix_i^2}{\pi}\right)$	$[0.\pi]^d$	8
F13	$\sum_{i=2}^{d/4}[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-2})^4 + 10(x_{4i-3} + x_{4i})^4]$	$[-10.10]^d$	10
F14	$\sum_{i=1}^{d} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	$[-10.10]^d$	10
F15	$0.5\sum_{i=1}^{d}(x_i^4 - 16x_i^2 + 5x_i) + 39.16599d$	$[-100.100]^d$	10
F16	$x_i^2 + 10^6\sum_{i=1}^{d} x^2$	$[-10.10]^d$	10
F17	$0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	$[-10.10]^d$	10
F18	$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	$[-4.5.4.5]^d$	20
F19	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5.5]^d$	20
F20	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$	$[-100.100]^d$	20

نتایج بدست آمده توسط الگوریتم‌ها شامل معیارهای بدترین، بهترین، میانگین و انحراف معیار جوابها در ۵۰ اجرای مستقل می‌باشند. نحوه محاسبه هر یک از این معیارها در جدول ۳ آورده شده است.

جدول ۳: نحوه محاسبه معیارهای مقایسه

معیار	رابطه محاسبه
بهترین	Best= Min(All Fitness POP)
بدترین	Worst= Max (All Fitness POP)
میانگین	$Mean = \frac{\sum_{i=1}^{Npop} Fitness(i)}{N}$
انحراف معیار	$STD = \sqrt{\frac{1}{N} \sum_{i=1}^{Npop} (Fitness_i - mean(All\ fitness))}$

نتایج الگوریتم جستجو کلاغ بهبود یافته و سایر الگوریتم‌های فرا ابتکاری در جدول ۴ به طور کامل آورده شده است. نتایج این جدول نشان می‌دهد که الگوریتم جستجو کلاغ بهبود یافته ازلحاظ میانگین و انحراف معیار و بهترین و بدترین جواب یافته شده، عملکرد بهتری از سایر الگوریتم‌های فرا ابتکاری دارد. یعنی هم کیفیت جواب‌های یافته شده توسط الگوریتم پیشنهادی بهتر است و هم اینکه با توجه به مقدار کم انحراف معیار جواب‌های یافته شده توسط آن، پایداری بهتری نسبت به سایر الگوریتم‌ها دارد.

جدول ۴: نتایج الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌ها

Function	Eval	ABC	CSA	HS	BAT	Proposed
Function1	Best	-1.00E+00	-1.76E-02	-9.35E-01	-8.04E-61	-1.00E+00
	Worst	-1.53E-06	-2.55E-05	-4.50E-80	0.00E+00	-5.46E-97
	Mean	-9.16E-01	-2.21E-03	-3.56E-01	-4.66E-62	-9.63E-01
	Std	2.49E-01	4.40E-03	3.96E-01	1.39E-61	1.66E-01
Function2	Best	6.42E-10	2.05E-04	2.74E-04	2.78E-10	1.27E-73
	Worst	1.05E-03	1.90E-01	2.51E-02	5.54E-02	1.64E-02
	Mean	4.58E-05	6.13E-03	2.45E-03	1.48E-03	3.46E-04
	Std	2.05E-04	2.12E-02	5.11E-03	5.63E-03	2.32E-03
Function3	Best	1.91E-07	1.52E-02	9.97E-04	1.68E-08	0.00E+00
	Worst	1.10E-01	2.66E+00	1.62E-01	7.81E-02	1.64E-01
	Mean	5.23E-03	6.73E-02	1.02E-02	1.36E-02	2.61E-03
	Std	2.21E-02	2.62E-01	2.84E-02	2.08E-02	1.75E-02
Function4	Best	2.26E-16	2.28E-03	4.04E-05	2.19E-09	4.68E-90
	Worst	1.25E-01	3.84E-01	4.33E-01	4.05E-01	3.76E-01
	Mean	2.58E-03	2.67E-02	1.35E-02	2.12E-02	4.21E-03
	Std	1.49E-02	6.20E-02	5.16E-02	5.87E-02	3.77E-02
Function5	Best	2.69E-11	1.09E+00	1.59E-02	2.24E+01	0.00E+00
	Worst	4.86E+01	6.74E+01	3.77E+01	1.00E+02	3.70E+01
	Mean	6.19E-01	1.46E+01	1.27E+00	3.17E+01	5.50E-01
	Std	4.90E+00	2.14E+01	4.23E+00	1.41E+01	3.98E+00

Function6	Best	5.73E-11	8.36E-03	2.35E-03	5.63E-10	0.00E+00
	Worst	2.68E+00	4.64E+00	1.64E+00	3.83E-01	1.60E+00
	Mean	4.02E-02	4.22E-01	9.24E-02	1.18E-02	2.97E-02
	Std	2.79E-01	7.44E-01	3.15E-01	4.19E-02	1.81E-01
Function7	Best	5.75E-13	3.00E-01	2.66E-01	2.39E+01	0.00E+00
	Worst	3.24E+00	1.97E+02	3.23E+01	2.09E+02	2.54E+02
	Mean	6.23E-02	1.59E+01	2.34E+00	2.57E+01	4.64E+00
	Std	3.48E-01	4.63E+01	6.85E+00	1.85E+01	3.18E+01
Function8	Best	3.99E-07	5.34E-01	1.64E-01	3.60E+00	0.00E+00
	Worst	5.89E+01	6.74E+01	9.37E+01	2.31E+01	7.93E+02
	Mean	7.22E-01	6.16E+00	4.79E+00	5.05E+00	1.37E+01
	Std	5.93E+00	1.74E+01	1.73E+01	4.54E+00	8.96E+01
Function9	Best	-1.86E+02	-1.86E+02	-1.87E+02	-1.87E+02	-1.87E+02
	Worst	-1.06E+02	-1.16E+02	-1.49E+02	-5.65E+01	-1.04E+02
	Mean	-1.84E+02	-1.77E+02	-1.83E+02	-1.79E+02	-1.84E+02
	Std	1.13E+01	2.19E+01	1.09E+01	2.41E+01	1.17E+01
Function10	Best	-1.03E+00	-1.02E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Worst	-8.51E-01	-3.74E-01	2.46E-02	-7.76E-01	-3.11E-01
	Mean	-1.03E+00	-9.27E-01	-1.01E+00	-1.01E+00	-1.02E+00
	Std	2.39E-02	1.55E-01	1.10E-01	5.68E-02	7.21E-02
Function11	Best	2.33E-02	6.36E-05	1.91E-02	1.56E-06	1.12E-40
	Worst	1.23E+00	3.17E-01	4.62E-01	5.43E-03	2.05E-01
	Mean	1.63E-01	1.20E-02	9.66E-02	4.08E-04	2.86E-03
	Std	3.02E-01	4.88E-02	8.83E-02	9.39E-04	2.09E-02
Function12	Best	3.80E+00	1.10E+01	1.69E+01	1.54E+01	1.86E-06
	Worst	2.05E+01	2.01E+01	2.06E+01	1.68E+01	1.98E+01
	Mean	9.92E+00	1.29E+01	1.76E+01	1.61E+01	2.66E+00
	Std	5.07E+00	2.29E+00	1.04E+00	4.28E-01	5.13E+00
Function13	Best	1.03E-12	6.67E+00	5.99E+00	6.72E+01	1.24E-46
	Worst	1.11E+03	1.82E+03	1.70E+03	9.67E+01	2.06E+03
	Mean	2.27E+01	1.46E+02	1.05E+02	7.95E+01	4.01E+01
	Std	1.30E+02	2.50E+02	3.08E+02	7.30E+00	2.61E+02
Function14	Best	8.32E-01	1.38E+00	2.77E+00	2.70E+00	1.59E-11
	Worst	1.07E+01	8.54E+00	8.94E+00	2.75E+00	7.84E+00
	Mean	1.79E+00	2.16E+00	4.04E+00	2.72E+00	7.34E-01
	Std	1.76E+00	1.09E+00	1.46E+00	1.20E-02	1.42E+00
Function15	Best	4.32E-04	5.08E+00	5.85E+00	1.38E+01	4.99E-13
	Worst	2.35E+01	2.44E+01	4.92E+01	2.43E+01	2.58E+00
	Mean	2.28E+00	7.78E+00	9.14E+00	1.52E+01	1.88E-01
	Std	5.23E+00	4.22E+00	6.86E+00	1.22E+00	5.90E-01
Function16	Best	7.26E-06	1.00E+01	1.55E+01	1.19E+02	1.57E-21
	Worst	4.78E+02	3.13E+02	7.00E+02	1.74E+02	5.94E+02
	Mean	2.46E+01	6.28E+01	8.42E+01	1.44E+02	2.18E+01
	Std	7.84E+01	5.21E+01	1.08E+02	1.60E+01	9.27E+01
Function17	Best	1.47E+03	3.14E+03	2.53E+04	2.32E+04	1.32E-08
	Worst	5.21E+04	4.26E+04	6.41E+04	2.37E+04	3.09E+04
	Mean	1.08E+04	7.59E+03	2.99E+04	2.34E+04	1.43E+03
	Std	1.24E+04	5.91E+03	7.86E+03	1.14E+02	4.86E+03

Function18	Best	2.93E+01	4.46E+01	2.06E+01	2.49E+01	0.00E+00
	Worst	1.04E+02	9.72E+01	1.05E+02	8.21E+01	1.03E+02
	Mean	4.35E+01	6.77E+01	3.79E+01	4.11E+01	7.24E+00
	Std	1.58E+01	1.38E+01	1.91E+01	1.83E+01	1.85E+01
Function19	Best	-7.62E+00	-7.19E+00	-1.27E+01	-1.01E+01	-1.81E+01
	Worst	-5.62E+00	-4.54E+00	-5.13E+00	-6.02E+00	-6.64E+00
	Mean	-7.07E+00	-6.71E+00	-9.38E+00	-9.21E+00	-1.63E+01
	Std	4.73E-01	5.82E-01	1.74E+00	6.73E-01	3.11E+00
Function20	Best	5.69E-02	3.23E+00	2.00E+01	2.08E+01	5.46E-08
	Worst	9.66E+01	9.58E+01	8.19E+01	5.94E+01	6.71E+01
	Mean	8.53E+00	1.36E+01	3.26E+01	2.58E+01	2.92E+00
	Std	1.70E+01	1.56E+01	1.37E+01	4.67E+00	8.80E+00

بطور کلی نتایج آزمایش‌ها نشان می‌دهد که الگوریتم جستجوی کلاغ بهبود یافته عملکرد خیلی بهتری نسبت به سایر الگوریتم‌ها دارد.

۵- نتیجه گیری و کارهای آینده

در این تحقیق یک الگوریتم جستجوی کلاغ بهبود یافته برای حل مسائل بهینه سازی ارائه شد که از روش مبتنی بر معکوس استفاده می‌نماید. برای بهبود عملکرد الگوریتم جستجوی کلاغ، روش مبتنی بر معکوس در شروع الگوریتم جستجوی کلاغ و همچنین در میانه تکرارهای الگوریتم جستجوی کلاغ برای بهبود کیفیت جمعیت جوابها بکار گرفته شد. در نهایت روش پیشنهادی در محیط نرم افزار متلب پیاده سازی شد و با الگوریتم‌های استاندارد زنبورعسل مصنوعی، بهینه‌سازی کلاغ، جستجوی هارمونی و خفاش و با استفاده از ۲۰ تابع محک استاندارد مقایسه شد. نتایج آزمایشات نشان داد که الگوریتم جستجو کلاغ بهبود یافته از لحاظ میانگین و انحراف معیار و بهترین و بدترین جواب، از سایر الگوریتم‌های فرا ابتکاری عملکرد بهتری دارد. در آینده می‌توان الگوریتم پیشنهادی را برای حل مسائل بهینه سازی واقعی از قبیل کوله پشتی، فروشنده دوره گرد و خوشه بندی استفاده کرد.

مراجع

[1] Hatamlou, A., *Black hole: A new heuristic optimization approach for data clustering*. Information sciences, 2013. 222: p. 175-184.

[2] Hatamlou, A., *Heart: a novel optimization algorithm for cluster analysis*. Progress in Artificial Intelligence, 2014. 2: p. 167-173.

[3] Askarzadeh, A., *A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm*. Computers & Structures, 2016. 169: p. 1-12.

[4] Shayanfar, H. and F.S. Gharehchopogh, *Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems*. Applied Soft Computing, 2018. 71: p. 728-746.

[5] Mirjalili, S., S.M. Mirjalili, and A. Lewis, *Grey wolf optimizer*. Advances in engineering software, 2014. 69: p. 46-61.

[6] Tizhoosh, H.R. *Opposition-based learning: a new scheme for machine intelligence*. in *Computational intelligence for modelling, control and automation, 2005 and international conference on intelligent agents, web technologies and internet commerce, international conference on*. 2005. IEEE.

[7] Elaziz, M.A., D. Oliva, and S. Xiong, *An improved opposition-based sine cosine algorithm for global optimization*. Expert Systems with Applications, 2017. 90: p. 484-500.

[8] Sharma, T.K. and M. Pant, *Opposition based learning ingrained shuffled frog-leaping algorithm*. Journal of Computational Science, 2017. 21: p. 307-315.

[9] Sarkhel, R., et al., *An improved Harmony Search Algorithm embedded with a novel piecewise opposition based learning algorithm*. Engineering Applications of Artificial Intelligence, 2018. 67: p. 317-330.

[10] Li, X. and M. Yin, *An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure*. Advances in Engineering Software, 2013. 55: p. 10-31.

[11] Bulbul, S.M.A., et al., *Opposition-based krill herd algorithm applied to economic load dispatch problem*. Ain Shams Engineering Journal, 2018. 9(3): p. 423-440.

[12] Dinkar, S.K. and K. Deep, *Opposition based Laplacian Ant Lion Optimizer*. Journal of Computational Science, 2017. 23: p. 71-90.

[13] Ewees, A.A., M.A. Elaziz, and E.H. Houssein, *Improved Grasshopper Optimization Algorithm using Opposition-based Learning*. Expert Systems with Applications, 2018.

[14] Sayed ,G.I., A.E. Hassanien, and A.T. Azar, *Feature selection via a novel chaotic crow search algorithm*. Neural Computing and Applications, 2019. 31(1): p. 171-188.

[15] Rizk-Allah, R.M., A.E. Hassanien, and S. Bhattacharyya, *Chaotic crow search algorithm for fractional optimization problems*. Applied Soft Computing, 2018. 71: p. 1161-1175.

[16] Saha, S. and V. Mukherjee, *A novel quasi-oppositional chaotic antlion optimizer for global optimization*. Applied Intelligence, 2018. 48(9): p. 2628-2660.