



مروری بر الگوریتم‌های ابتکاری و فراابتکاری

عاطفه محمدی

گروه مهندسی کامپیوتر، واحد اهواز، دانشگاه آزاد اسلامی، اهواز، ایران
Mohammadi@gmail.com

خلاصه: مسائل زیادی در علوم مختلف، از قبیل کامپیوتر، ریاضیات، مهندسی و مدیریت، از نوع مسائل بهینه‌سازی هستند. در این نوع مسائل، ما از میان مجموعه‌ای از جواب‌های ممکن، می‌خواهیم بهترین جواب، که تابع هدف را، بیشینه و یا کمینه کند، پیدا کنیم. خیلی از مسائل بهینه‌سازی که در دنیای واقعی با آن مواجه هستیم، دارای ابعاد بزرگ و پیچیدگی هستند. در این مواقع، رسیدن به جواب بهینه قطعی، بسیار زمانبر است. الگوریتم‌های فراابتکاری یا متاهوریستیک، الگوریتم‌هایی هستند که با استفاده از ابتکار، فضای جستجو را کوچک‌تر می‌سازند. گرچه همیشه در این روش، تضمینی برای پیدا کردن جواب قطعی وجود ندارد، اما این الگوریتم‌ها با درصد قابل قبولی می‌توانند به جواب بهینه نزدیک شوند. الگوریتم‌های فراابتکاری، زمان محاسبه را به طور قابل ملاحظه‌ای کاهش می‌دهند. الگوریتم‌های فراابتکاری، به طور وسیع در مسائل بهینه‌سازی علوم مختلف، از قبیل علوم کامپیوتر، هوش مصنوعی و مهندسی و مدیریت، کاربرد دارند. یکی از کاربردهای مهم الگوریتم‌های متاهوریستیک، در مسائل بهینه‌سازی ترکیبیاتی است. به این دسته از مسائل، مسائل بهینه‌سازی گسسته نیز می‌گویند. بسیاری از این مسائل از نوع مسائل سخت NP-Hard می‌باشند. مسائل ترکیبیاتی معمولاً ظاهری ساده دارند، اما به‌سختی می‌توانیم آن‌ها را حل کنیم.

کلمات کلیدی: الگوریتم‌های فراابتکاری، الگوریتم‌های ابتکاری، بهینه‌سازی.



۱ - مقدمه

سیستماتیک برای خارج شدن از بهینه‌های محلی یا جلوگیری از قرار گرفتن در بهینه‌های محلی با این مشکل برخورد می‌کنند. ویژگی مشترک الگوریتم‌های فراابتکاری، استفاده از مکانیزم‌های خروج از بهینه‌ی محلی است [2].

۳ - الگوریتم‌های فراابتکاری

الگوریتم‌های فراابتکاری یکی از انواع الگوریتم‌های ابتکاری هستند که در مسائل گوناگون کاربرد دارند. این الگوریتم‌ها که نوعی از الگوریتم‌های تصادفی هستند برای یافتن پاسخ بهینه به کار برده می‌شوند. بخش عظیمی از این الگوریتم‌ها که از زندگی اجتماعی حشرات الهام می‌گیرد بسیار قدرتمند و قوی هستند و در برخی کلونی آنها بطور مداوم در حال رشد و توسعه می‌باشند و یا در برخی از آنها بر اثر مسائلی مانند شکار کاهش می‌یابند و یا بر اثر تقسیم کلونی تجزیه می‌شوند. اما چیزی که مهم است این است که بطور معمول حشرات با تغییر شرایط در عرضه مواد غذایی و عادت‌های خود کنار می‌آیند. این حشرات بسیار کوچک بوده و به تنهایی قدرت آنچنانی ندارند اما جمع آنها بسیار قوی و قدرتمند خواهد بود. معیارهای مختلفی جهت طبقه‌بندی الگوریتم‌های فراابتکاری وجود دارد که بصورت زیر می‌باشند [2]:

- **الگوریتم‌های با حافظه:** این الگوریتم‌ها دارای حافظه هستند و اطلاعات بدست آمده را در حین جستجو در خود ذخیره می‌کند.
- **الگوریتم‌های بی‌حافظه:** این دسته از الگوریتم‌های فراابتکاری فاقد حافظه هستند، این بدین معنی است که، این نوع الگوریتم‌ها اطلاعات بدست آمده در حین جستجو را ذخیره نمی‌کند و از آن استفاده نمی‌کند.
- **الگوریتم‌های قطعی:** مسئله را با استفاده از تصمیمات قطعی حل می‌کند.
- **الگوریتم‌های غیرقطعی:** از یک سری قوانین احتمالی در حین جستجو استفاده می‌کند.

روش‌ها و الگوریتم‌های بهینه‌سازی به دو دسته کلی الگوریتم‌های قطعی و الگوریتم‌های تقریبی تقسیم‌بندی می‌شوند. الگوریتم‌های قطعی قادر به یافتن جواب بهینه به صورت دقیق هستند اما در مورد مسائل بهینه‌سازی سخت کارایی آنها به شدت کاهش می‌یابد و به دو دسته مستقیم و غیرمستقیم تقسیم می‌شوند. الگوریتم‌های تقریبی قادر به یافتن جواب‌های نزدیک به بهینه در زمان حل کوتاه برای مسائل بهینه‌سازی سخت هستند و به دو دسته الگوریتم‌های ابتکاری و فراابتکاری بخش‌بندی می‌شوند. دو مشکل اصلی الگوریتم‌های ابتکاری قرار گرفتن در بهینه محلی و ناتوانی برای کاربرد در مسائل گوناگون است. بنابراین الگوریتم‌های فراابتکاری برای رفع مشکلات الگوریتم‌های ابتکاری ارائه شدند [1]

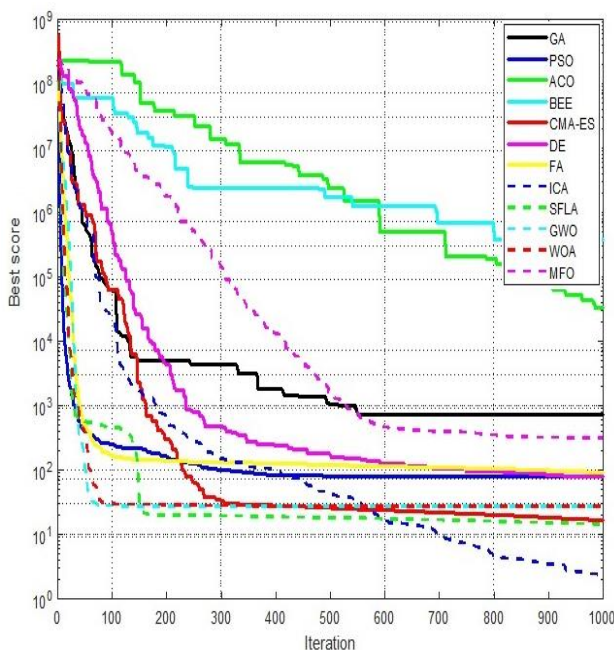
۲ - الگوریتم‌های ابتکاری

مسائل زیادی در علوم مختلف، از قبیل کامپیوتر، ریاضیات، مهندسی و مدیریت، از نوع مسائل بهینه‌سازی هستند. در این نوع مسائل، ما از میان مجموعه‌ای از جواب‌های ممکن، می‌خواهیم بهترین جواب، که تابع هدف را، بیشینه و یا کمینه کند، پیدا کنیم. خیلی از مسائل بهینه‌سازی که در دنیای واقعی با آن مواجه هستیم، دارای ابعاد بزرگ و پیچیدگی هستند. در این مواقع، رسیدن به جواب بهینه‌ی قطعی، بسیار زمانبر است. الگوریتم‌های ابتکاری، معمولاً به دنبال رسیدن به جواب‌های خوب، یعنی جواب‌های نزدیک به بهینه، در زمان محاسباتی قابل قبول هستند ولی قادر به تضمین بهینه بودن جواب‌های به دست آمده نیستند. روش‌های ابتکاری مختلفی برای حل مسائل بهینه‌سازی ترکیبی ابداع شده است که در آنها، جواب نزدیک به بهینه تولید می‌شود. مشکل این روش‌ها، این است که اکثر آنها فقط در مورد یک مسأله‌ی به خصوص قابل کاربرد هستند.

یکی از عیب‌های الگوریتم‌های ابتکاری، تولید یک جواب با تعداد محدودی از جواب‌ها و توقف آنها در بهینه محلی با کیفیت پایین است. الگوریتم‌های فراابتکاری، برای حل این مشکلات و عیب‌های روش‌های ابتکاری پیشنهاد شده‌اند. الگوریتم‌های فراابتکاری، با معرفی اصول



- الگوریتم‌های مبتنی بر یک جواب: الگوریتم‌هایی که یک جواب برای مسئله پیدا می‌کنند.
- الگوریتم‌های مبتنی بر جمعیت: این دسته از الگوریتم‌ها برای پیدا کردن جواب بهینه از یک جمعیت اولیه شروع می‌کند. از الگوریتم‌های فراابتکاری شناخته شده در این دسته می‌توان به الگوریتم‌های ژنتیک، کلونی مورچگان، کلونی زنبورها، کرم شبتاب، روش بهینه‌سازی ازدحام ذرات، رقابت استعماری، فاخته، گرگ خاکستری و غیره اشاره کرد.
- الگوریتم ژنتیک GA
- الگوریتم ازدحام ذرات PSO
- الگوریتم کلونی مورچگان ACO
- الگوریتم زنبورعسل مصنوعی BEE
- الگوریتم استراتژی تکاملی انطباق ماتریس کوواریانس CMA-ES
- الگوریتم تفاضل تکاملی DE
- الگوریتم کرم شب‌تاب FA
- الگوریتم جهش قورباغه SFLA
- الگوریتم رقابت استعماری ICA
- الگوریتم گرگ خاکستری GWO
- الگوریتم وال یا نهنگ WOA
- الگوریتم شمع و پروانه MFO



شکل ۲: مقایسه کلی الگوریتم‌های فراابتکاری مبتنی بر جمعیت

یکی دیگر از مسائل مهمی که در الگوریتم‌های فراابتکاری در نظر گرفته می‌شود، مسئله پیاده‌سازی آنها می‌باشد. فرایند طراحی و پیاده‌سازی الگوریتم‌های فراابتکاری دارای سه مرحله متوالی است که هر کدام از آنها دارای گام‌های متفاوتی است. در هر گام فعالیت‌هایی باید انجام شود تا آن گام کامل شود. مرحله اول آماده‌سازی است که در آن باید شناخت دقیقی از مسئله‌ای که می‌خواهیم حل کنیم بدست آوریم، و اهداف طراحی الگوریتم‌های فراابتکاری برای آن باید با توجه به روش‌های حل موجود برای این مسئله بطور واضح و شفاف مشخص شود. مرحله بعدی ساخت نام دارد. مهمترین هدف این مرحله انتخاب استراتژی حل، تعریف معیارهای اندازه‌گیری عملکرد و طراحی الگوریتم برای استراتژی حل انتخابی می‌باشد و آخرین مرحله پیاده‌سازی است که در آن پیاده‌سازی الگوریتم‌های طراحی شده در مرحله قبل، شامل تنظیم پارامترها، تحلیل عملکرد و در نهایت تدوین و تهیه گزارش نتایج باید انجام شود. در سال‌های اخیر الگوریتم‌های فراابتکاری جدیدی نیز بوجود آمده اند که در ادامه به برخی از مهمترین آنها اشاره خواهد شد.

در شکل ۲ به مقایسه ۱۲ الگوریتم بهینه‌سازی و نتایج حاصل از آنها در دستیابی به نقاط بهینه در تکرارهای کوتاه پرداخته شده است. این الگوریتم‌ها مبتنی بر جمعیت یا population based بوده که با الهام‌گیری از طبیعت و محیط پیرامون اقدام به حل مسئله می‌کنند.

برخی از معروف ترین این الگوریتم‌ها عبارتند از [2] :

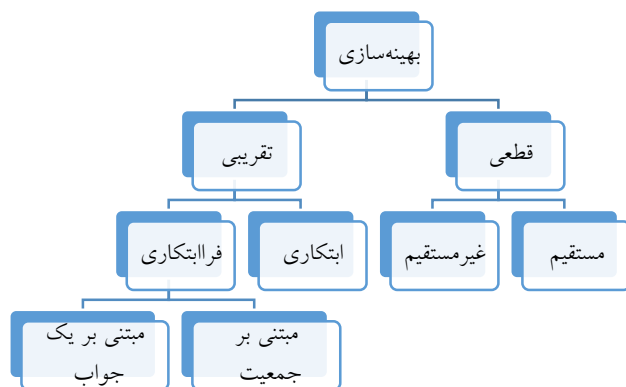


۴- روش‌های حل مسائل بهینه‌سازی

مسائل بهینه‌سازی به دو دسته‌ی مسائل بهینه‌سازی ترکیبی و پیوسته تقسیم‌بندی می‌شوند. در زمینه‌ی مسائل پیوسته، روش‌های کلاسیک بسیاری برای بهینه‌سازی سراسری ارائه شده است، اما این تکنیک‌ها در صورتی که تابع هدف دارای ویژگی‌های مشخصی (نظیر محدب بودن^۱) نباشد، معمولاً کارایی ندارند. روش‌های حل مسائل بهینه‌سازی پیوسته، به دو دسته‌ی روش‌های بهینه‌سازی خطی و غیرخطی تقسیم‌بندی می‌شوند. روش‌های حل مسائل بهینه‌سازی ترکیبی را نیز می‌توان به دو قسمت الگوریتم‌های دقیق^۲ و الگوریتم‌های ابتکاری^۳ تقسیم‌بندی کرد. بهینه‌سازی غیرخطی پیوسته و روش‌های تقریبی مسائل گسسته، جزو مسائل سخت محسوب می‌شوند [1].

الگوریتم‌های دقیق برای حل مسائل بهینه‌سازی ترکیبی، نظیر روش شاخه و حد^۴، روش‌هایی هستند که رسیدن به جواب بهینه را تضمین می‌کنند. روش‌های دقیق، معمولاً برای حل مسائل با اندازه‌های کوچک یا متوسط، می‌توانند به کار گرفته شوند. برای حل مسائل واقعی که معمولاً با ابعاد بزرگ و پیچیده هستند، امکان استفاده از روش‌های دقیق وجود ندارد. بنابراین، از روش‌های ابتکاری، غالباً برای حل مسائل بهینه‌سازی ترکیبی بزرگ استفاده می‌شود. بیشتر مسائل بهینه‌سازی ترکیبی، در کلاس مسائل NP-hard قرار می‌گیرند. بنابراین برای حل آنها باید از روش‌های تقریبی که جواب‌های نزدیک به بهینه را در زمانی قابل قبول به دست می‌آورند، استفاده کرد. الگوریتم‌هایی از این دست، الگوریتم‌های ابتکاری نام دارند. الگوریتم‌های ابتکاری، معمولاً به دنبال رسیدن به جواب‌های خوب، یعنی جواب‌های نزدیک به بهینه، در زمان محاسباتی قابل قبول هستند ولی قادر به تضمین بهینه بودن جواب‌های به دست آمده نیستند. روش‌های ابتکاری مختلفی برای حل مسائل بهینه‌سازی ترکیبی ابداع شده است که در آنها، جواب نزدیک به بهینه تولید می‌شود. مشکل این روش‌ها، این است که اکثر آنها فقط در مورد یک مسأله‌ی به خصوص قابل کاربرد هستند.

یکی از عیب‌های الگوریتم‌های ابتکاری، تولید یک جواب با تعداد محدودی از جواب‌ها و توقف آنها در بهینه محلی با کیفیت پایین است. الگوریتم‌های فراابتکاری، برای حل این مشکلات و عیب‌های روش‌های ابتکاری پیشنهاد شده‌اند. الگوریتم‌های فراابتکاری، با معرفی اصول سیستماتیک برای خارج شدن از بهینه‌های محلی یا جلوگیری از قرار گرفتن در بهینه‌های محلی با این مشکل برخورد می‌کنند. ویژگی مشترک الگوریتم‌های فراابتکاری، استفاده از مکانیزم‌های خروج از بهینه محلی است. در شکل ۱ روش‌های بهینه‌سازی به طور خلاصه نشان داده شده است.



شکل ۱: روش‌های بهینه‌سازی

الگوریتم‌های قطعی قادر به یافتن جواب بهینه به صورت دقیق هستند اما در مورد مسائل بهینه‌سازی سخت کارایی کافی ندارند و زمان اجرای آنها متناسب با ابعاد مسائل به صورت نمایی افزایش می‌یابد. الگوریتم‌های تقریبی قادر به یافتن جواب‌های خوب (نزدیک به بهینه) در زمان حل کوتاه برای مسائل بهینه‌سازی سخت هستند. الگوریتم‌های تقریبی نیز به سه دسته الگوریتم‌های ابتکاری^۵ و فراابتکاری^۶ و فوق‌ابتکاری^۷ بخش‌بندی می‌شوند. دو مشکل اصلی الگوریتم‌های ابتکاری، گیر افتادن آنها در نقاط بهینه محلی و همگرایی زودرس به این نقاط است. الگوریتم‌های فراابتکاری برای حل این

^۵ heuristic

^۶ meta-heuristic

^۷ hyper heuristic

^۱.Convexity

^۲.Exact Algorithms

^۳.Heuristic Algorithms

^۴.Branch & Bound Method



نامرغوب‌تری است.

۵- مروری بر الگوریتم‌های فراابتکاری

در ادامه به شرح مختصری از برخی الگوریتم‌های فراابتکاری خواهیم پرداخت.

- **الگوریتم کلونی پنگوئن‌های امپراتور (EPC)**

این الگوریتم مبتنی بر ازدحام موجودات و الهام گرفته از طبیعت می‌باشد که توسط تابش گرمای بدن پنگوئن‌ها و حرکت مارپیچی آنها در مستعمره خود کنترل می‌شود. این الگوریتم، یک الگوریتم حافظه‌دار می‌باشد و برای حل مسایل چند متغیره مناسب است [3].

- **الگوریتم سنجاقک (DA)**

ایده اصلی الگوریتم سنجاقک از رفتار گروهی سنجاقک‌ها در حالت ایستا (استراحت) و پویا (حرکت یا جنب‌وجوش) در طبیعت الهام گرفته است. دو مرحله اساسی در بهینه‌سازی، اکتشاف و بهره‌برداری است که از کار گروهی در راهبری، جستجو غذا و اجتناب از دشمنان در حالت‌های ایستا و پویا الگوبرداری شده است [2].

- **الگوریتم جستجوی جغد (OSA)**

الگوریتم جستجوی جغد یک الگوریتم بهینه‌سازی الهام گرفته از طبیعت است که برای حل مشکلات بهینه‌سازی جهانی طراحی شده است. این الگوریتم یک روش مبتنی بر جمعیت است که براساس مکانیزم شکار کردن جغدها در تاریکی مدل‌سازی شده است [4].

- **الگوریتم بهینه‌سازی نهنگ یا وال (WOA)**

ساختار این الگوریتم از شیوه شکار حباب خالص وال‌ها الهام گرفته است. نهنگ کوهان‌دار ترجیح می‌دهد تا گروهی از کرپل‌ها یا ماهیان کوچک که نزدیک به سطح آب هستند را شکار کند. مدل ریاضی الگوریتم بهینه‌سازی نهنگ که مبتنی بر روش تغذیه حباب خالص است شامل مراحل محاصره طعمه، مانور تغذیه حباب خاص به صورت مارپیچی و به دام انداختن شکار می‌باشد [5].

مشکلات الگوریتم‌های ابتکاری ارائه شده‌اند. در واقع الگوریتم‌های فراابتکاری، یکی از انواع الگوریتم‌های بهینه‌سازی تقریبی هستند که دارای راهکارهای برون‌رفت از نقاط بهینه محلی هستند و قابلیت کاربرد در طیف گسترده‌ای از مسائل را دارند. رده‌های گوناگونی از این نوع الگوریتم در دهه‌های اخیر توسعه یافته است که همه این‌ها زیر مجموعه الگوریتم فراابتکاری می‌باشند. تا به امروز تعریف مشخص و جامعی از اصطلاح متاهوریستیک یا فراابتکاری صورت نگرفته است و تعاریف مختلفی برای این اصطلاح ارائه شده است.

برخلاف روش‌های ابتکاری، هدف اصلی این روش‌ها، جستجوی مؤثر و کارای فضای جواب به جای یافتن صرف جواب‌های بهینه یا نزدیک بهینه می‌باشد؛ روش‌های فراابتکاری سیاست‌ها و راهکارهایی هستند که فرآیند جستجو را هدایت می‌کنند؛ روش‌های فراابتکاری تقریبی بوده و اغلب غیرقطعی (تصادفی) می‌باشند؛ این روش‌ها ممکن است با استفاده از مکانیزم‌هایی از به دام افتادن فرآیند جستجو در بهینه‌های موضعی جلوگیری کنند؛ الگوریتم‌های فراابتکاری، برخلاف روش‌های ابتکاری وابسته به نوع مساله نیستند، به عبارت دیگر می‌توان آنها را برای حل طیف گسترده‌ای از مسائل بهینه‌سازی مورد استفاده قرار داد.

روش‌های فراابتکاری پیشرفته‌تر، از تجربیات و اطلاعات به دست آمده در طول فرآیند جستجو به شکل حافظه برای هدایت جستجو به نواحی پرامیدتر فضای جواب استفاده می‌کنند؛ به طور خلاصه می‌توان گفت که الگوریتم‌های فراابتکاری، راهکارهای پیشرفته و کلی جستجو می‌باشند و گام‌ها و معیارهایی را پیشنهاد می‌کنند که در فرار از دام بهینه‌های موضعی بسیار مؤثر هستند. عامل مهم در این روش‌ها، تعادل پویا بین استراتژی‌های تنوع بخشی و پر قدرت سازی است. تنوع بخشی به جستجوی گسترده در فضای جواب اشاره دارد و پر قدرت سازی به معنی بهره‌برداری از تجربیات به دست آمده در فرآیند جستجو و تمرکز بر نواحی پرامیدتر فضای جواب می‌باشد. بنابراین با ایجاد تعادل پویا بین این دو استراتژی از یک طرف، جستجو به سمت محدوده‌هایی از فضای جواب سوق داده می‌شود که جواب‌های بهتری در آنها یافت شده است و از طرف دیگر موجب عدم اتلاف زمان بیشتر در بخشی از فضای جواب می‌شود که پیش از این بررسی شده و یا شامل جواب‌های



نسل جاری و موقعیت بهترین ملخ وابسته است. از ویژگی‌های این الگوریتم می‌توان به سادگی و دارا بودن فقط یک پارامتر تنظیم، ارجاع کرد[9].

• الگوریتم گرده افشانی گل‌ها (FPA)

این الگوریتم از نحوه گرده افشانی گل‌ها الهام گرفته است. برخی گل‌ها و حشرات با یک مشارکت بسیار حرفه‌ای گرده افشانی را انجام می‌دهند. به عنوان مثال، برخی از گل‌ها فقط می‌توانند به یک گونه خاص حشرات برای گرده افشانی موفق جذب شوند[10].

• الگوریتم جستجوی کلاغ‌ها (CSA)

این الگوریتم از رفتار هوشمند کلاغ‌ها الهام گرفته و همچنین یک تکنیک مبتنی بر جمعیت است. ایده اصلی کلاغ این است که کلاغ‌ها غذای مازاد خود را مخفی کرده و ذخیره می‌کنند و در صورت نیاز آن را پیدا می‌کنند. کلاغ‌ها می‌توانند از ابزارها استفاده کنند و با روش‌های پیچیده‌ای ارتباط برقرار کنند و مکان‌های مخفی غذا خود را تا چند ماه بعد به یاد آورند. آنها مکان‌هایی را که پرندگان دیگر غذای خود را پنهان می‌کنند، مشاهده می‌کنند و زمانی که صاحب آن محل را ترک می‌کند آنها را سرقت می‌کنند[11].

الگوریتم بهینه‌سازی شیر (LOA)

این الگوریتم که از رفتار اجتماعی شیرها الهام گرفته شده است یک الگوریتم مبتنی بر جمعیت می‌باشد. شیرها در دو نوع ساکن و کوچ‌نشین زندگی می‌کنند. شیرهای نر بالغ آزادانه گروه را ترک کرده و در اطراف حرکت می‌کنند. شیرهای ساکن بعنوان عوامل جستجوی محلی و شیرهای کوچ‌نشین به عنوان عوامل جستجوی جهانی برای شناسایی و اکتشاف مدل می‌شوند. سبک زندگی خاص شیرها و ویژگی‌های همکاری آنها انگیزه اصلی برای توسعه این الگوریتم بهینه‌سازی بوده است[12].

• الگوریتم بهینه‌سازی بویایی کوسه (SSO)

این الگوریتم براساس توانایی کوسه به عنوان یک شکارچی برتر در طبیعت برای یافتن طعمه طراحی و مدل‌سازی شده است که از حس بویایی کوسه و حرکت آن به سمت منبع بو الهام گرفته است. برای نشان دادن کارایی این روش در حل مسایل مهندسی واقعی از این روش برای بهینه‌سازی کنترل فرکانس بار در سیستم‌های برق استفاده شده است. نتایج به دست آمده اعتبار الگوریتم بهینه‌سازی پیشنهادی را تأیید می‌کند[6].

• الگوریتم بهینه‌سازی شیر مورچه (ALO)

این الگوریتم از نحوه خاص شکار مورچه‌ها توسط شیر مورچه الهام گرفته است. شیر مورچه تله خود را به صورت گودال مخروطی شکل می‌سازد و به محض اینکه از حضور مورچه در تله آگاه می‌شود؛ دانه‌های ماسه را به طرف لبه گودال پرتاب می‌کند و در نهایت مورچه را شکار می‌کند[7].

• الگوریتم بهینه‌سازی گرگ خاکستری (GWO)

الگوریتم گرگ خاکستری یک الگوریتم فراکاوشی الهام گرفته از طبیعت است که اساس آن بر پایه ساختار سلسله مراتبی و رفتار اجتماعی گرگ‌ها در زمان شکار می‌باشد. این الگوریتم فرایند ساده‌ای را در تنظیمات دارد و به راحتی قابلیت تعمیم به مسایل با ابعاد بزرگ را دارا می‌باشد. در پیاده‌سازی این الگوریتم، چهار نوع از گرگ‌های خاکستری مانند آلفا، بتا، دلتا و امگا برای شبیه‌سازی سلسله مراتب رهبری استفاده شده است که در آن سه گام اصلی از شکار، جستجو برای طعمه، محاصره طعمه و حمله به طعمه، اجرا می‌شوند. این الگوریتم تنها دارای دو پارامتر تنظیم می‌باشد[8].

• الگوریتم بهینه‌سازی ملخ (GOA)

این الگوریتم با الهام از رفتار اجتماعی ملخ‌ها و نحوه تأثیرپذیری هر ملخ از محیط پیرامونش طراحی شده است. در این الگوریتم بروزرسانی موقعیت هر ملخ به فاصله هر ملخ از تمام جمعیت ملخ‌ها در



را بیابد [16].

• الگوریتم بهینه‌سازی پروانه (ABO)

الگوریتم بهینه‌سازی پروانه یک الگوریتم جدید الهام گرفته از طبیعت است که از رفتار جستجوی غذا و زاد و ولد پروانه‌ها برای حل مسایل بهینه‌سازی جهانی تقلید می‌کند. این چارچوب عمدتاً بر مبنای استراتژی تغذیه پروانه‌ها است که از حس بویایی برای تعیین موقعیت شهد یا زاد و ولد استفاده می‌کنند [13].

• الگوریتم جستجوی سنجاب (SSA)

این الگوریتم یک روش الهام گرفته از طبیعت می‌باشد که از شیوه خاص پرواز سنجاب‌های جنوبی و روش حرکت آنها معروف به گایدینگ (بدون نیرو پرواز کردن) تقلید می‌کند. الگوریتم جستجوی سنجاب پاسخ‌های دقیق‌تر را با سرعت همگرایی بالاتر ارائه می‌دهد. پریدن مکانیسم موثری است که توسط پستانداران کوچک برای پیمودن مسافت‌های طولانی استفاده می‌شود [14].

• الگوریتم بهینه‌سازی کریل (KH)

الگوریتم بهینه‌سازی کریل، یک الگوریتم بهینه‌سازی بیولوژیکی است که از رفتار اجتماعی گروهی کریل‌ها برای حل مسایل بهینه‌سازی سراسری تقلید می‌کند. حداقل فاصله هر کریل از غذا و از بالاترین تراکم دسته کریل‌ها به عنوان تابع هدف در نظر گرفته می‌شود. موقعیت وابسته به زمان هر کریل توسط سه عامل اصلی حرکت ناشی از حضور افراد دیگر، جستجوی غذا و انتشار تصادفی بررسی می‌شود [15].

• الگوریتم مگس میوه (FFOA)

الگوریتم مگس میوه یک روش مکاشفه‌ای جدید مبتنی بر رفتار غذایابی مگس میوه است. مگس میوه نسبت به سایر حشرات دیگر دارای حس بویایی و بینایی قوی‌تری می‌باشد به طوری که می‌تواند بوی انواع میوه‌ها را در هوا تشخیص دهد. این حشره پس از استشاق بوی میوه و بعد از این که به موقعیت میوه نزدیک شد می‌تواند با استفاده از حس بینایی خود و همکاری با سایرین موقعیت دقیق میوه

• الگوریتم پژواک صدای دلفین (DE)

این الگوریتم بر اساس مکان‌یابی پژواک صدای دلفین طراحی شده است. بررسی‌ها نشان داده‌اند که دلفین‌ها می‌توانند در ذهن خود برای صداها گوناگون تجسم‌های گوناگونی پدید آورند به گونه‌ای که هر صدا در ذهنشان بیان‌گر و برابر با یک شی باشد. در هنگام بهره‌گیری از پژواک‌یابی، دلفین‌ها اغلب صدای برگشتی از یک جانور یا اشیای اطراف را در ذهن خود تحلیل کرده و هر صدای برگشتی را مترادف با یک شی تعریف می‌کنند. این «دیدن از طریق شنیدن» نشان می‌دهد که چگونه دلفین‌های با توانایی دید بسیار پایین می‌توانند در مغز خود جهان را شناسایی کنند [17].

• الگوریتم جستجوی صاعقه (LSA)

این الگوریتم بر اساس پدیده طبیعی رعد و برق و مکانیزم انتشار رعد پیشگام با استفاده از مفهوم ذرات سریع (ذکر شده به عنوان پرتابه) طراحی شده است و نرخ همگرایی بالا ارائه می‌دهد. سه نوع پرتابه برای تشکیل گروه رهبری نشان داده شده است. پرتابه‌ها در این الگوریتم به معنای دیگر نشان دهنده مقدار اولیه جمعیت هستند [18].

• الگوریتم جلبک مصنوعی (AAA)

الگوریتم بهینه‌سازی جلبک مصنوعی، الهام گرفته شده از رفتارهای زندگی ریز جلبک‌ها گونه‌های فتوسنتزی است. این الگوریتم بر اساس فرآیند تکاملی، فرآیند تطبیق و حرکت ریز جلبک‌ها طراحی شده است [19].

• الگوریتم ازدحام سالپ (SSA)

در عمق اقیانوس سالپ‌ها یک ازدحام به نام زنجیره سالپ تشکیل می‌دهند. علت اصلی تشکیل این زنجیره‌ها برای مسیریابی و یافتن آذوقه می‌باشد. سالپ‌های دنبال کننده به دنبال سردسته می‌باشند و سالپ سردسته به سمت منبع غذایی حرکت می‌کند. اگر منبع غذایی



۸- نتیجه گیری

پیچیدگی مدل‌های ریاضی، افزایش نمایی زمان حل بسیاری از روش‌ها، عدم دسترسی به اطلاعات گرادیان و همگرایی به بهینه محلی، از جمله مشکلاتی هستند که الگوریتم‌های بهینه‌سازی کلاسیک در حل مسایل پیچیده با آن‌ها مواجه هستند. به منظور رفع این مشکلات از الگوریتم‌های فراابتکاری به طور گسترده برای حل مسائل پیچیده و چندمتغیره استفاده می‌شود. انتخاب بهترین و مناسبترین الگوریتم به دلیل تنوع بالای آنها کاری دشوار است. در این مقاله تعدادی از پرکاربردترین الگوریتم‌های بهینه‌سازی فراابتکاری از سال ۲۰۱۲ تا ۲۰۱۹ معرفی شد. شرح عملکرد کلی، پارامترهای اصلی، کاربرد در مسایل واقعی و منبع الهام هر کدام از این روش‌ها به اختصار بیان شده است. تمامی این روش‌ها الگوریتم‌های مبتنی بر جمعیت بوده‌اند و از طریق تولید یک جمعیت تصادفی و تکامل آن برای بهینه‌سازی استفاده شده است. در این میان تنها دو الگوریتم کلونی پنگوئن امپراتور و الگوریتم جستجوی کلان‌ها جزو الگوریتم‌های حافظه دار بوده و سایر الگوریتم‌ها بدون حافظه می‌باشند.

با پاسخ بهینه کلی تعویض شود به صورت خودکار زنجیر سالپ‌ها به سمت بهینه کلی حرکت خواهد کرد. این الگوریتم به تخمین بهینه کلی توسط قراردادی چندین سالپ با موقعیت اتفاقی می‌پردازد، سپس سازگاری هر سالپ محاسبه می‌شود، سالپ با بهترین سازگاری کشف شده و موقعیت بهترین سالپ به متغیر F بعنوان منبع غذایی اختصاص می‌یابد تا توسط زنجیر سالپ تعقیب شود آن سالپ در صورت خروج هر سالپ از فضای جستجو به مرزهای تعیین شده بازگردانده می‌شود. تمام گام‌های بالا به غیر از گام جایگذاری اولیه تا زمان ارضای شرایط توقف تکرار می‌شود [20].

• بهینه‌سازی کفتار خالدار (SHO)

ایده اصلی این الگوریتم از رفتار اجتماعی کفتار خالدار و رابطه بین آنها گرفته شده است. مفهوم اصلی پشت این الگوریتم، رابطه اجتماعی است یعنی کفتارهای لکه‌دار و رفتار جمعی آن‌ها شامل سه مرحله اساسی جستجوی شکار، احاطه کردن طعمه و حمله به طعمه می‌باشد و برای هر سه مرحله، مدل ریاضی تعریف شده است [21].

• الگوریتم بهینه‌سازی شاهین هریس (HHO)

الگوریتم بهینه‌سازی شاهین هریس مبتنی بر جمعیت و الهام از طبیعت ارائه شده است که به آن بهینه‌سازی هریس هاوکس (HHO) گفته می‌شود. الهام بخش اصلی این الگوریتم از رفتار مشارکتی و سبک تعقیب طعمه توسط گروه شاهین هریس در طبیعت با روش گشت و گذار غافلگیرانه استنباط شده است. در این استراتژی هوشمند چندین شاهین با همکاری و حمله از جهات مختلف تلاش برای غافلگیر کردن طعمه می‌کنند. شاهین هریس می‌تواند الگوهای مختلف تعقیب را بر اساس ماهیت سناریوهای پویا و الگوهای گریز طعمه ارایه دهد. این روش به طور محاسباتی از الگوها و رفتارهای پویا تقلید می‌کند تا یک الگوریتم بهینه‌سازی را بوجود آورد [22].

مراجع

- [1] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja and H. Chen, "Harris hawks optimization: Algorithm and applications," Future Generation Computer Systems, vol. 97, pp. 849-872, 2019.
- [2] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," Neural Computing and Applications, vol. 27, pp. 1053-1073, 2016.
- [3] S. Harifi, M. Khalilian, S. Ebrahimnejad and J. Mohammadzadeh, "Emperor Penguins Colony: a New Metaheuristic Algorithm for Optimization," Evolutionary Intelligence, 2019.
- [4] J. Mohita, M. Shubhamb and A. Rani, "Owl search algorithm: A novel nature-inspired heuristic paradigm for global optimization," Journal of



- [17] A. Kaveh and N. Farhodi, "A new optimization method: Dolphin echolocation," *Advances in Eng Software*, 2013.
- [18] H. Shareef, A. A. Ibrahim and A. H. Mutlag, "Lightning search algorithm," *Applied Soft Computing*, 2015.
- [19] S. A. Uymaz, G. Tezel and E. Yel, "Artificial algae algorithm (AAA) for nonlinear global optimization," *Applied Soft Computing*, 2015
- [20] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris and S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems," *Advances in Eng Software*, 2017.
- [21] G. Dhiman and V. Kumar, "Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications," *Advances in Eng Software*, 2017.
- [22] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja and H. Chen, "Harris hawks optimization: Algorithm and applications," in *Future Generation Computer Systems*, 2019.
- [5] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Eng Soft*, vol. 95, 2016.
- [6] O. Abedinia, A. Ghasemi and N. Amjady, "A new metaheuristic algorithm based on shark smell optimization", *Complexity*, 2016.
- [7] S. Mirjalili, "The Ant Lion Optimizer," *Advances in Engineering Software*, vol. 83, pp. 80-98, 2015.
- [8] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey Wolf Optimizer," *Advances in Eng Software*, 2014.
- [9] S. Saremi, S. Mirjalili and A. Lewis, "Grasshopper Optimisation Algorithm: Theory and application," *Advances in Engineering Software*, 2017.
- [10] X.-S. Yang, "Flower Pollination Algorithm for Global Optimization," in *International Conference on Unconventional Computing and Natural Computation*, 2012.
- [11] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm", *Computers & Structures*, 2016.
- [12] M. Yazdani and F. Jolai, "Lion Optimization Algorithm (LOA): A nature-inspired metaheuristic algorithm," *Journal of Computational Design and Engineering*, 2016.
- [13] X. Qi, Y. Zhu and H. Zhang, "A new meta-heuristic butterfly-inspired algorithm," *Journal of Computational Science*, vol. 23, pp. 226-239, 2017.
- [14] M. Jain, V. Singh and A. Rani, "A novel nature-inspired algorithm for optimization: Squirrel search algorithm," *Swarm and Evolutionary Computation BASE DATA*, 2018.
- [15] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, 2012.
- [16] W.-T. Pan, "A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example," *Knowledge- Based Systems*, 2012.