ICWR2022
May 11-12, 2022; Tehran, Iran
8th International Conference on Web Research

Investigating and Comparing the Performance of Meta-Heuristic Algorithms in Feature Selection and Software Fault Prediction

Ali Karimi[1], Mohsen Norouzi[2], Mohammad Hossein Darvishanpour*[3]

[1] Assistant Professor of Comprehensive Imam Hossein University, Tehran
a.karimi@ihu.ac.ir

[2] Educator and Researcher of Comprehensive Imam Hossein University, Tehran
m.norouzi@ihu.ac.ir

[3] Master student of Comprehensive Imam Hossein University, Tehran
g9813038390@ihu.ac.ir

Abstract
Meta-Heuristic algorithms are optimization techniques that provide the optimal solution through processes of repeated exploration and exploitation of the entire search space. Feature selection is also an important and prominent process in the field of machine learning that reduces data dimensions. This paper examines and compares nature-inspired meta-heuristic algorithms for feature selection to increase the accuracy of software fault prediction. Researchers can not easily select meta-heuristic algorithms as a suitable method for their research due to their great variety and multiplicity. In this paper, by describing the feature selection techniques and its methods, the application of meta-heuristic algorithms in different fields, such as swarm intelligence and binary methods of these algorithms has been investigated. Also, by introducing 18 meta-heuristic algorithms in 6 different categories and evaluating each of them, a suitable analysis has been provided to researchers so that they can easily and with the highest efficiency choose the appropriate algorithm and method of their work. In the papers presented so far, meta-heuristic algorithms have been studied from only one aspect, while in this article, while studying different types of research, they have tried to study and evaluate them from different aspects.

Keywords: Meta-Heuristic Algorithms, Swarm Intelligence, Feature Selection, Data-Dimensionality Reduction, Machine Learning.

ICWR2022
May 11-12, 2022; Tehran, Iran
8th International Conference on Web Research

بررسی و مقایسه عملکرد الگوریتم‌های فرااکتشافی در انتخاب ویژگی و پیش‌بینی خطای نرم‌افزار

علی کریمی[۱]، محسن نوروزی[۲] و محمد حسین درویشان‌پور*[۳]

[۱] استادیار دانشگاه جامع امام حسین (ع)، تهران،
a.karimi@ihu.ac.ir

[۲] مربی و پژوهشگر دانشگاه جامع امام حسین (ع)، تهران،
m.norouzi@ihu.ac.ir

[۳] دانشجوی کارشناسی ارشد دانشگاه جامع امام حسین (ع)، تهران،
g9813038390@ihu.ac.ir

چکیده

الگوریتم‌های فرااکتشافی فنون بهینه‌سازی هستند که با فرایندهای اکتشاف و بهره‌برداری مکرر از کل فضای جستجو، راه‌حل بهینه را ارائه می‌دهند. انتخاب ویژگی نیز یک فرایند مهم و برجسته در حوزه یادگیری ماشین است که باعث کاهش ابعاد داده می‌شود. این مقاله به بررسی و مقایسه الگوریتم‌های فرااکتشافیِ الهام‌گرفته از طبیعت برای انتخاب ویژگی در راستای افزایش دقت پیش‌بینی خطای نرم‌افزار می‌پردازد. پژوهشگران، الگوریتم‌های فرااکتشافی را به دلیل تنوع و تعدد زیاد، نمی‌توانند به راحتی و در زمانی کوتاه، به‌عنوان یک روش مناسب برای تحقیق موردنظر خود انتخاب کنند. در این مقاله سعی شده است با تشریح فنون انتخاب ویژگی و روش‌های آن، کاربرد الگوریتم‌های فرااکتشافی در حوزه‌های مختلف، از قبیل هوش جمعی و روش‌های دودویی کردن این الگوریتم‌ها مورد بررسی قرار گیرد. همچنین، با معرفی ۱۸ الگوریتم فرااکتشافی در ٦ دسته مختلف و ارزیابی هر کدام از آن‌ها، یک تحلیل مناسب در اختیار پژوهشگران قرار داده شده است تا به سادگی و با بیش‌ترین بازدهی بتوانند الگوریتم و روش مناسب کار خود را انتخاب نمایند. در مقالاتی که تاکنون ارایه شده است، الگوریتم‌های فرااکتشافی تنها از یک جنبه مورد بررسی قرار گرفته‌اند، در حالی که در این مقاله ضمن مطالعه انواع مختلفی از تحقیقات انجام‌شده، سعی شده است از جنبه‌های مختلف مورد بررسی و ارزیابی قرار گیرند.

کلمات کلیدی

الگوریتم‌های فرااکتشافی، بهینه‌سازی، هوش جمعی، انتخاب ویژگی، کاهش ابعاد داده، یادگیری ماشین.

۱- مقدمه

امروزه فراوانی داده‌ها باعث شده است که مدیریت داده‌ها به یک کار بسیار پیچیده و برجسته تبدیل شود. یک مجموعه داده شامل تعداد زیادی ویژگی است که البته برخی از آنها ضروری و برخی دیگر افزونه و زاید هستند. ویژگی‌های افزونه باعث کاهش عملکرد الگوریتم‌های یادگیری ماشین در حین کار با مجموعه داده‌ها می‌شوند. بنابراین فنون کاهش ابعاد و انتخاب ویژگی با هدف حفظ عملکرد به همراه حذف ویژگی‌های زاید و افزونه به دنبال افزایش عملکرد هستند [41]. برای رسیدن به این هدف نمی‌توان از روش‌های بهینه‌سازی قطعی استفاده کرد، چون آن‌ها توانایی حل مسئله را در یک بازه زمانی مشخص را ندارند. الگوریتم‌های فرااکتشافی به طور کلی برای مسائلی که هیچ الگوریتم رضایت‌بخشی ویژه‌ای برای حل آنها وجود ندارد، استفاده می‌شوند و برای حل مسائل پیچیده صنعت، تشخیص بیماری، بهینه‌سازی پرس‌وجو، پیش‌بینی سهام و انتخاب ویژگی کاربرد دارند. همگی آنها الهام‌گرفته از طبیعت هستند و از پارامترهای تصادفی استفاده می‌کنند که در سال‌های اخیر محبوبیت و کاربرد بسیاری پیدا کرده‌اند [42]. همچنین انتخاب یک الگوریتم فرااکتشافی مناسب بسیار حائز اهمیت است.

مانیک شارما و همکاران [1]، در مقاله خود ابتدا، به نقش فن انتخاب ویژگی در ده حوزه تحقیقاتی عمده و همچنین به کمترین و بیشترین زمینه‌های تحقیقاتی اشاره کرده‌اند. مشخص شده است که در ۱۰ سال گذشته بیشتر فعالیت‌های تحقیقاتی صورت‌گرفته با استفاده از کلونی زنبور و کلونی مورچه انجام شده است. الگوریتم‌های جستجوی میمون، بهینه‌سازی ازدحام ذرات و کرم شب‌تاب در دسته کم استفاده‌شدگان رتبه‌بندی شده‌اند. ویلی پودگورلس و همکاران [2]، این مقاله یک بررسی جامع درباره الگوریتم‌های هوش جمعی برای انتخاب ویژگی ارائه می‌کند که شکاف را در این زمینه تحقیقاتی کاهش می‌دهد. دکتر جیداد گیانی و همکاران [4]، در مقاله خود روش بهینه‌سازی کلونی مورچه‌گان را برای پیش‌بینی خطای نرم‌افزار با در نظر گرفتن مجموعه داده‌های مختلف به کار گرفته‌اند. نتایج نشان می‌دهد که روش پیشنهادی نتایج امیدوارکننده‌ای ارائه می‌دهد.

در این مقاله به بررسی و ارزیابی ۱۸ الگوریتم فرااکتشافی مختلف در ۶ دسته الهام‌گرفته از طبیعت با نام‌های الگوریتم کلونی موچه، کلونی زنبور مصنوعی، کرم شب‌تاب، خفاش، شیر مورچه، سنجاقک، شعله پروانه، ازدحام ذرات، تغذیه مصنوعی پرندگان، کلونی پنگوئن امپراتور، فیل، گرگ خاکستری، وال، ماهی بادبانی، گرده افشانی گل، علف هرز تهاجمی، ژنتیک و جستجوی صف با هدف استفاده و کاربرد آن در بحث انتخاب ویژگی پرداخته شده است. خواننده می‌تواند با مطالعه کامل مقاله، علاوه بر آشنایی و کسب اطلاعات درباره رفتار الگوریتم‌ها، با توجه به جدولی که در آخر مقاله ارائه شده است می‌تواند الگوریتم مناسب برای پروژه و تحقیق خود را انتخاب نماید. ساختار مقاله با این شرح است، در ابتدا توضیحی درباره فن انتخاب ویژگی و روش‌های آن ارائه می‌شود. سپس به کاربرد الگوریتم‌های فرااکتشافی و مبحث هوش جمعی پرداخته می‌شود. هر یک از الگوریتم‌های فرااکتشافی نامبرده تعریف می‌شوند و جدولی به منظور ارزیابی هر یک از آن‌ها ارائه می گردد. در آخر نتیجه‌گیری مطرح می‌شود.

۲- مفاهیم اولیه مورد نیاز

در این بخش از مقاله فن انتخاب ویژگی به همراه انواع روش‌های آن، کاربرد الگوریتم‌های فرااکتشافی، هوش جمعی و توابع انتقال که روشی برای دودویی کردن الگوریتم‌های فرااکتشافی است تشریح می‌شوند.

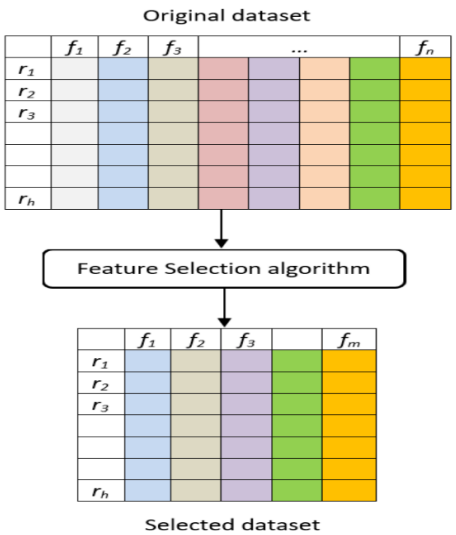
۲-۱- انتخاب ویژگی

انتخاب ویژگی یک فن مهم است که در تجزیه و تحلیل الگو استفاده می‌شود. همان‌طور که در شکل(۱) مشاهده می‌شود، می‌توان با حذف متغیرهای زاید و افزونه، ابعاد داده را کاهش داد. برای مثال اگر مجموعه داده‌ای دارای ۲۰ ویژگی باشد، می‌توان با استفاده از فنون انتخاب ویژگی و شناسایی و حذف ویژگی‌های زاید و افزونه و کاهش ویژگی‌ها، سرعت و دقت کار با مجموعه داده را افزایش داد. انتخاب ویژگی تلاش می‌کند تا بهترین زیرمجموعه ویژگی‌ها را پیدا کند، زیرمجموعه انتخاب شده باید مرتبط‌ترین ویژگی‌ها باشد. انتخاب ویژگی شامل سه روش است [3]:

- روش‌های فیلتر[۱]
- روش‌های پوشش[۲]
- روش‌های تعبیه‌شده[۳]

۲-۱-۱- روش فیلتر

روش‌های فیلتر ویژگی‌ها را بدون استفاده از الگوریتم‌های یادگیری ماشین و با تکیه بر ویژگی‌های داده‌ها ارزیابی می‌کنند. در ابتدا، ویژگی‌ها را بر اساس معیارهای خاصی رتبه‌بندی می‌کند، چه مستقل از یکدیگر و چه با در نظر گرفتن فضای ویژگی. سپس، ویژگی‌هایی با بالاترین رتبه برای ایجاد زیرمجموعه‌ای از ویژگی‌ها انتخاب می‌شوند. این روش سریع‌تر و از نظر محاسباتی ارزان‌تر از روش‌های پوشش هستند. هنگام برخورد با داده‌های با ابعاد بالا، استفاده از روش‌های فیلتر از لحاظ محاسباتی ارزان‌تر است [3].



شکل (۱) : فرایند انتخاب ویژگی [15]

۲-۱-۲- روش پوشش

همان‌طور که در شکل (۲) مشاهده می‌شود، روش‌های پوشش از یک الگوریتم یادگیری ماشین از پیش‌تعیین‌شده (به عنوان مثال، طبقه‌بند) برای ارزیابی کیفیت ویژگی‌ها استفاده می‌کنند و از تعصبات الگوریتم در فرایند انتخاب ویژگی اجتناب می‌شود. با این حال، آن‌ها مجبور هستند الگوریتم یادگیری ماشین را بارها برای ارزیابی کیفیت زیرمجموعه‌های انتخاب شده از ویژگی‌ها اجرا کنند، که از نظر محاسباتی بسیار گران است. بنابراین، از الگوریتم‌های هوش جمعی می‌توان برای بهینه‌سازی فرایند انتخاب ویژگی در یک مدل پوشش استفاده کرد، هدف آن‌ها این است که تعیین کنند کدام زیرمجموعه از ویژگی‌های ممکن بهترین عملکرد را هنگام استفاده از یک الگوریتم یادگیری ماشین از پیش‌تعیین‌شده ارائه می‌دهد. از آنجا که طبقه‌بندی متداول‌ترین وظیفه یادگیری ماشین است، از آن برای رویکرد انتخاب ویژگی استفاده می‌شود و هر روش یادگیری ماشین دیگری را می‌توان به طور مشابه استفاده کرد. با توجه به الگوریتم طبقه‌بندی از پیش‌تعریف‌شده، یک روش معمولی پوشش در دو مرحله زیر تکرار می‌شود تا به هدف مدنظر رسیده شود (یعنی دقت خاصی کسب شود) [2]:

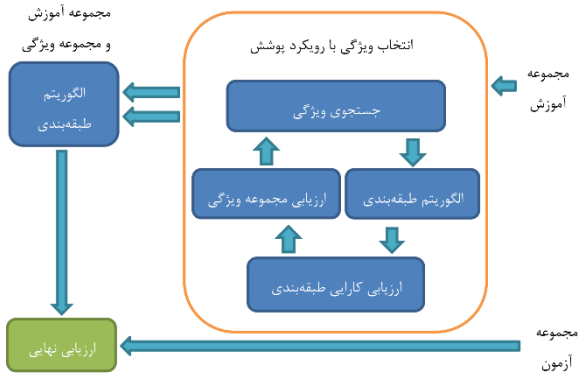
- زیرمجموعه‌ای از ویژگی‌ها انتخاب شود.
- با انجام الگوریتم طبقه‌بندی انتخاب شده، زیرمجموعه انتخاب شده ارزیابی شود.

۲-۱-۳- روش تعبیه‌شده

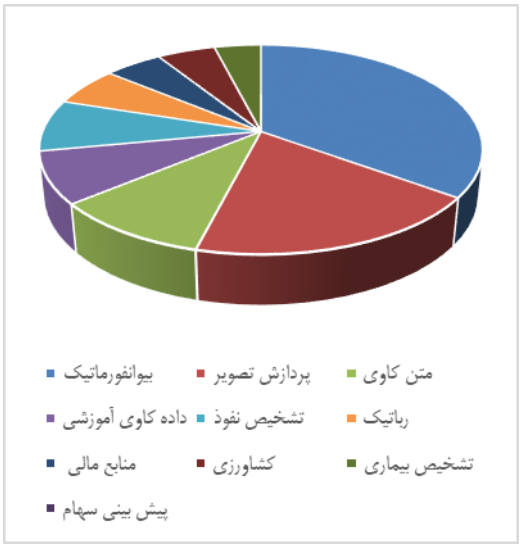
روش تعبیه‌شده یک روش جدید است که سعی می‌کند روش‌های فیلتر و پوشش را ترکیب کند، این روش‌ها با جستجو در فرایند یادگیری، مفید بودن زیرمجموعه‌های ویژگی را اندازه‌گیری می‌کنند. این بدان معناست که تعبیه‌شده شبیه روش پوشش با هزینه محاسباتی کمتری است [3].

۲-۲- کاربرد الگوریتم‌های فرااکتشافی

مسائل بهینه‌سازی، چند حالته، غیرخطی و پیچیده هستند و بنابراین، حل این نوع مسائل با استفاده از فنون قطعی مشکل است. به طور کلی، یک مسئله بهینه‌سازی را می‌توان با استفاده از روش‌های متفاوتی مانند مبتنی بر جمعیت، الگوریتم‌های تعیین‌کننده و تصادفی حل کرد. فرااکتشافی فنون بهینه‌سازی تصادفی هستند که عملکرد فنون ابتکاری را بهبود می‌بخشند. فنون فرااکتشافی مسئله بهینه محلی را حل کرده و با موفقیت در حل انواع مسائل بهینه‌سازی ازقبیل، تشخیص بیماری، بهینه‌سازی پرس‌وجو، انتخاب ویژگی، پیش‌بینی و متن کاوی که برخی از کاربردهای عمده این فنون هستند سهیم شده‌اند. همچنین در حوزه‌های مختلف دیگر ازقبیل، نجوم، کشاورزی، منابع مالی، مراقبت‌های بهداشتی، بیوانفورماتیک[۴]، فرایند صنعتی، علوم اجتماعی و رفتاری مورد توجه زیادی قرار گرفته‌اند. در شکل (۳) نقش انتخاب ویژگی در حوزه‌های مختلف ترسیم شده است. بیوانفورماتیک و پردازش تصویر بیشترین زمینه‌های مورد بررسی هستند. با این حال، پیش‌بینی و تشخیص بیماری تا آنجا که به مسئله انتخاب ویژگی مربوط می‌شوند، کمتر مورد توجه قرار گرفته است [1].



شکل (۲) : چارچوب روش انتخاب ویژگی مبتنی بر پوشش [2]



شکل (۳) : نقش انتخاب ویژگی در حوزه‌های مختلف [1]

۲-۳- هوش جمعی

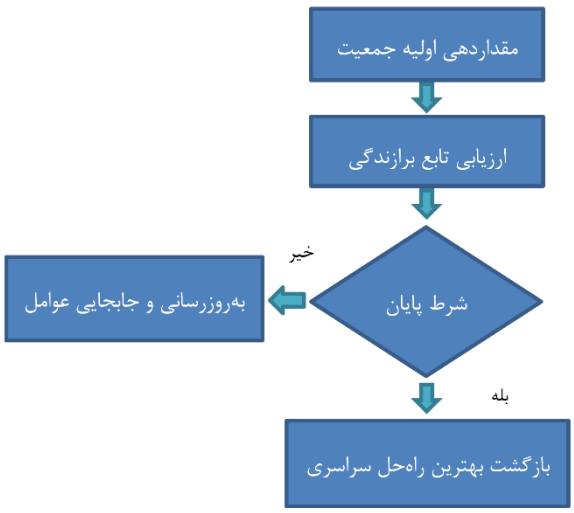
هوش جمعی[۵] به عنوان یکی از شاخه‌های هوش محاسبات پذیرفته شده است و یکی از واضح‌ترین مزایای آن خودمختاری است. هیچ مدیریت خارجی ندارد، اما هر عامل در ازدحام رفتار خود را به طور مستقل کنترل می‌کند. عامل نشان‌دهنده راه‌حل احتمالی برای یک مسئله معین است و بر این اساس، می‌توان دومین مزیت را که خود سازماندهی است، استنباط کرد. هوش بر عامل فردی تمرکز نمی‌کند، بلکه در خود گروه ظاهر می‌شود. بنابراین، راه‌حل‌ها از قبل مشخص نیستند، اما در زمان اجرای برنامه تغییر می‌کنند. خودسازمان‌دهی نقش مهمی در سازگاری دارد. دومین مزیت، در تغییر محیط‌هایی که نمایندگان به خوبی به تغییرات ذکر شده پاسخ می‌دهند، رفتار خود را تغییر داده و به طور مستقل با آنها سازگار می‌شود [2].

علاوه بر این، هماهنگی مرکزی وجود ندارد، چون هیچ نقطه شکستی وجود ندارد و ازدحام باعث افزونگی می‌شود که در آن دو مزیت دیگر پنهان است. اولین مورد مقیاس‌پذیری است، به این معنا که گروه می‌تواند شامل چندین هزار عامل باشد، زیرا هیچ عامل واحدی برای گروه وجود ندارد، انعطاف‌پذیری که مزیت هوش جمعی است کاملاً برآورده می‌شود. هر الگوریتم هوش جمعی، باید از برخی مراحل اساسی پیروی کند. بنابراین، شکل

(۴) را می‌توان تعریف کرد. در ابتدا جمعیت اولیه به صورت تصادفی تولید و مقداردهی می‌شود. سپس شرط توقف و پایان برنامه تعیین می‌شود که می‌تواند رسیدن به مقدار معینی از تکرارها یا کسب بهترین مقدار بهینه باشد. سپس هر کدام از راه‌حل‌های بدست آمده توسط تابع برازندگی محاسبه می‌شوند. عوامل دارای مقدار برازندگی ضعیف حذف شده و با راه‌حل‌های بهتری جایگزین می‌شوند. این مراحل آنقدر تکرار می‌شوند که به شرط پایان رسیده شود و بهترین مقدار بدست آمده تعیین شود [2].

۲-٤- توابع انتقال

برای حل مسائل بهینه‌سازی دودویی می‌توان از قوانین به اصطلاح دوتایی‌سازی استفاده کرد. در این راستا، توابع انتقال ریاضی مختلف همراه با قوانین دوتایی‌سازی برای تبدیل مقادیر ورودی واقعی به دودویی معرفی شده‌اند [8]. برای دستیابی به این هدف، هشت تابع انتقال مختلف از دو گروه S و V در جدول (۱) تشکیل داده شده است. چندین محقق از این رویکرد برای تبدیل الگوریتم‌های پیوسته به نسخه دودویی استفاده کردند [9].



شکل (٤) : چارچوب هوش‌جمعی [2]

جدول (۱) : توابع انتقال شکل S و V [15]

S-shaped family		V-shaped family	
Name	Transfer function	Name	Transfer function
S1	$T(x) = \frac{1}{1+e^{-2x}}$	V1	$T(x) = \left\lvert erf\left(\frac{\sqrt{\Pi}}{2}x\right)\right\rvert = \lvert\frac{\sqrt{2}}{\Pi}\int_0^{\left(\frac{\sqrt{\Pi}}{2}\right)x} e^{-t^2}\,dt\rvert$
S2	$T(x) = \frac{1}{1+e^{-x}}$	V2	$T(x) = \lvert\, tanh(x)\,\rvert$
S3	$T(x) = \frac{1}{1+e^{(-\frac{x}{2})}}$	V3	$T(x) = \lvert(x)/\sqrt{1+x^2}\rvert$
S4	$T(x) = \frac{1}{1+e^{(-\frac{x}{3})}}$	V4	$T(x) = \lvert\frac{2}{\Pi}\, arc\, tan(\frac{\Pi}{2}x)\rvert$

۳- طرح مساله

با توجه به اهمیت الگوریتم‌های فرااکتشافی در فن انتخاب ویژگی، پیش‌بینی خطا نرم‌افزار و تعدد بسیار زیاد الگوریتم‌های فرااکتشافی تاکنون پیشنهاد شده و همچنین روند رو به رشد این نوع الگوریتم‌ها، انتخاب یک الگوریتم موثر برای یک محقق کار بسیار دشواری است چون نیازمند زمان زیادی برای آزمون و خطا است که بسیار خسته‌کننده است.

جدول (۲) : الگوریتم‌های الهام‌گرفته از حشرات و خزندگان

اختصار	الگوریتم	سال
ACO	Ant Colony Optimization	۱۹۹۹
ABC	Artifcial Bee Colony	۲۰۰۷
FA	Firefy Algorithm	۲۰۰۹
BA	Bat Algorithm	۲۰۱۰
ALO	Ant Lion Optimizer	2015
DA	Dragonfy algorithm	۲۰۱۵
MFO	Moth-Flame Optimization	۲۰۱۵

جدول (۳) : الگوریتم‌های الهام‌گرفته از پرندگان

اختصار	الگوریتم	سال
PSO	Particle Swarm Optimization	۱۹۹۵
AFB	Artifcial Feeding Birds	۲۰۱۹
EPC	Emperor Penguins Colony	۲۰۱۹

جدول (٤) : الگوریتم‌های الهام‌گرفته از حیوانات

اختصار	الگوریتم	سال
GWO	Grey Wolf Optimizer	۲۰۱۴
EHO	Elephant Herding Optimization	۲۰۱۵

جدول (٥) : الگوریتم‌های الهام‌گرفته از جانوران دریایی

اختصار	الگوریتم	سال
WOA	Whale Optimization Algorithm	۲۰۱۶
SFO	Sail Fish Optimizer	۲۰۱۹

جدول (٦) : الگوریتم‌های الهام‌گرفته از گیاهان

اختصار	الگوریتم	سال
IWO	Invasive Weed Optimization	۲۰۱۰
FPA	Flower Pollination Algorithm	۲۰۱۲

جدول (۷) : الگوریتم‌های الهام‌گرفته از انسان‌ها

اختصار	الگوریتم	سال
GA	Genetic Algorithm	۱۹۸۸
QS	Queuing Search	۲۰۱۸

٤- راه‌حل پیشنهادی

در این مقاله سعی شده است که به بررسی ۱۸ الگوریتم فرااکتشافی الهام‌گرفته از طبیعت پرداخته و آنها به منظور آشنایی و درک بهتر مقایسه و ارزیابی شوند. در نتیجه محقق می‌تواند با زمان کمتر و آگاهی بالاتر الگوریتم مناسب تحقیق خود را انتخاب کند. همان طور که در جداول شماره (۲) تا (۷) قابل مشاهده است، الگوریتم‌های فرااکتشافی مطرح شده در ۶ دسته، حشرات و خزندگان، پرندگان، حیوانات، جانوران دریایی، گیاهان، انسان‌ها تقسیم‌بندی شده‌اند.

٤-١- الگوریتم‌های فرااکتشافی الهام‌گرفته از حشرات و خزندگان

٤-١-١- الگوریتم کلونی مورچه

الگوریتم کلونی مورچه[۶]، رفتار مورچه‌ها را در جستجوی غذا شبیه‌سازی می‌کند. در فرایند جستجوی غذا، مورچه‌های مختلف با گذاشتن نوعی ماده شیمیایی به نام فِرمون[۷] در مسیر خود، مسیرهای متفاوتی را برای رسیدن به منبع غذا از لانه خود و بازگشت به لانه انتخاب می‌کنند. مورچه‌ها در کوتاه‌ترین مسیر قدم می‌زنند و در مقایسه با مورچه‌هایی که طولانی‌ترین مسیر را طی کرده‌اند، به سرعت به لانه می‌رسند. با افزایش زمان، ماده شیمیایی رسوب شده در مسیر ممکن است تبخیر شود و در نتیجه فِرمون بیشتری در کوتاهترین مسیر رسوب می‌کند. همه مورچه‌های دیگر که از لانه شروع به حرکت می‌کنند، مسیری را دنبال می‌کنند که حاوی مقدار بیشتری از فِرمون با رسوب مواد شیمیایی خود در مسیر است. بنابراین فِرمون رسوب شده در طولانی‌ترین مسیر ممکن است پس از زمان معینی ناپدید شود و بنابراین هیچ مورچه‌ای آن مسیر را انتخاب نمی‌کند. به این ترتیب می‌توان از رفتار مورچه‌ها برای یافتن مسیر بهینه با کوتاهترین مسیر استفاده کرد. از آنجا که مورچه‌ها بر روی یک سیستم بسیار پویا کار می‌کند، می‌توان رفتار مورچه‌ها را در یافتن راه‌حل برای مسایل بهینه‌سازی شبیه‌سازی کرد [۴].

این الگوریتم نیاز به یک مسئله دارد تا به صورت گراف[۸] نمایش داده شود. در شکل (۵)، ویژگی‌ها به شکل گره نشان داده شده است، لبه‌های بین آنها نشان‌دهنده انتخاب ویژگی بعدی است. سپس جستجو برای زیرمجموعه ویژگی بهینه، یک پیمایش مورچه‌ای از طریق گراف است که در آن حداقل تعداد گره‌ها بازدید می‌شوند. مورچه در حال حاضر در گره a قرار دارد و انتخاب دارد که کدام ویژگی را به مسیر خود اضافه کند (خطوط نقطه چین). ویژگی b، سپس c و سپس d را انتخاب می‌کند. پس از رسیدن به d، زیر مجموعه فعلی {a; b; c; d} برای برآورده کردن معیار توقف پیمایش تعیین می‌شود (به عنوان مثال دقت الگوریتم طبقه‌بندی مناسبی با این زیر مجموعه به دست آمده است). مورچه پیمایش خود را خاتمه می‌دهد و این زیرمجموعه ویژگی را به عنوان مجموعه ویژگی کاهش داده شده به عنوان خروجی اعلام می‌کند [۵].

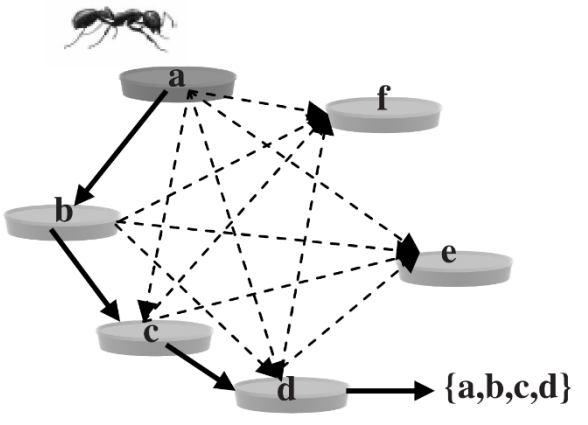
٤-١-٢- الگوریتم کلونی زنبور مصنوعی

الگوریتم کلونی زنبور مصنوعی[۹] یک الگوریتم هوش ازدحام است که رفتار زنبورهای شاغل، زنبورهای ناظر و زنبورهای پیشاهنگ را در فرایند تغذیه شبیه‌سازی می‌کند. به عنوان یک الگوریتم ازدحام، این الگوریتم دارای خود سازماندهی و تقسیم کار است. تقسیم کار به عنوان تعیین نقش‌ها و مسئولیت‌های مختلف با عوامل موجود در ازدحام مشخص می‌شود، به طوری که زنبورها به عنوان زنبورهای شاغل، ناظر و پیشاهنگ طبقه‌بندی می‌شوند. زنبورهای شاغل وظیفه بهره‌برداری را بر عهده دارند و همچنین مسئول آگاه‌سازی زنبورهای تماشاچی در مورد مکان و کیفیت منبع غذایی با حرکت رقص‌گونه هستند. زنبورهای ناظر، رقص زنبورهای شاغل را تماشا می‌کنند و منبع غذایی را برای جستجو و بهره‌برداری تعیین می‌کنند. زنبورهای پیشاهنگ بسته به انگیزه درونی یا بر اساس سرنخ‌های دیگر احتمالی، به کشف منابع جدید در محیط اختصاص داده شده‌اند. این نقش‌ها می‌توانند بر اساس شرایط کندو به طور هوشمند تغییر کنند [۶].

زنبور پیشاهنگ وقتی منبع جدیدی پیدا می‌کند و زنبور شاغل وقتی منبعش تمام می‌شود، تبدیل به زنبور پیشاهنگ می‌شوند. یک زنبور عادی در کندو منتظر است و رقص‌ها را تماشا می‌کند و هنگامی که منبعی را که درباره آن اطلاعات جمع‌آوری کرده است انتخاب می‌کند، به زنبور شاغل تبدیل می‌شود [۶]. الگوریتم کلونی زنبور مصنوعی برای حل مسائل مختلف بهینه‌سازی در حوزه‌های مختلف استفاده می‌شود. شکل (۶) در قسمت عملگرها، فرایندهای انتخاب، جایگزینی و بازتولید زنبورها مطرح شده است. انتخاب زنبورهای پیشاهنگ، حداکثر چرخه تکرار و انتخاب منابع غذایی به عنوان پارامترهای کنترلی مطرح شده است. همچنین حوزه‌های استفاده از این الگوریتم نشان داده شده است و می‌توان به مسائل مسیریابی، مسائل زمان‌بندی، تقسیم‌بندی تصویر و مسائل تخصیص افزونگی و غیره اشاره داشت [۷].

٤-١-٣- الگوریتم کرم شب‌تاب

الگوریتم کرم شب‌تاب[۱۰] یک فن جستجو و همچنین بهینه‌سازی است و اساس آن فعالیت چشمک‌زن کرم شب‌تاب است. کرم شب‌تاب با یکدیگر از



شکل (۵) : رویکرد الگوریتم کلونی مورچه برای انتخاب ویژگی [۵]

طریق درخشش زیستی درخشان ارتباط برقرار می‌کند و در مقایسه با جستجوی خودسرانه، فضای عملکردی را بسیار کارآمدتر جستجو می‌کند. هر کرم شب‌تاب با توجه به روشنایی کرم شب‌تاب به سوی یکدیگر جذب می‌شود. قوانین اولیه به این شرح است [8]:

- هر کرم شب‌تاب باید از جنس یکسان باشد (تک جنسیتی).
- اگر بیش از ۲ کرم شب‌تاب وجود داشته باشد، کرم شب‌تاب با نور تیره‌تر به سمت کرم شب‌تاب شدیدتر جذب می‌شود.
- وقتی کرم شب‌تاب روشن‌تری وجود ندارد، آن‌ها خودسرانه پرواز می‌کنند.

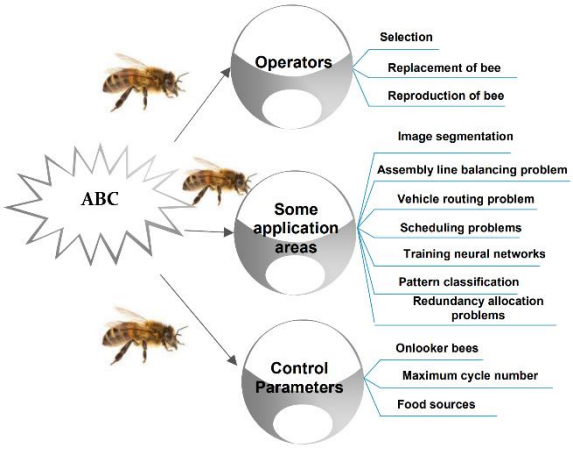
٤-١-٤- الگوریتم خفاش

خفاش‌ها[11] تنها پستاندارانی هستند که بال دارند و تخمین زده می‌شود که حدود ۹۹۶ گونه مختلف وجود دارد که تا ۲۰ درصد از کل گونه‌های پستانداران را تشکیل می‌دهند. خفاش‌ها حشره‌خوار هستند و از نوعی ردیاب آوایی برای تشخیص طعمه، اجتناب از موانع و تعیین محل شکاف‌های خود در تاریکی استفاده می‌کنند. این خفاش‌ها یک صدای صوتی بسیار بلند منتشر می‌کنند و به انعکاسی که از اجسام اطراف بازمی‌گردد گوش می‌دهند. صدای آنها از نظر خواص متفاوت است و بسته به گونه می‌تواند با رویکرد شکار آنها مرتبط باشد. اگر بتوان برخی از ویژگی‌های خفاش‌ها را ایده‌آل در نظر گرفت، می‌توان الگوریتم‌های الهام‌گرفته از خفاش را توسعه داد. برای سادگی، از قوانین تقریبی یا ایده‌آل مطرح شده استفاده می‌شود [9]:

۱- همه خفاش‌ها از انعکاس صدا برای حس کردن فاصله همچنین تفاوت بین مواد غذایی، شکار و موانع استفاده می‌کنند.

۲- خفاش‌ها به طور تصادفی با سرعت، موقعیت، فرکانس ثابت، طول موج متغیر و بلندی صدای مشخص شده برای جستجوی طعمه پرواز می‌کنند. آنها می‌توانند به طور خودکار طول موج (یا فرکانس) صداهای ساطع شده خود را تنظیم کرده و بسته به نزدیکی هدفشان، نرخ انتشار صدا را تنظیم کنند.

۳- اگرچه بلندی صدا می‌تواند از بسیاری جهات متفاوت باشد، فرض می‌شود که بلندی صدا از مقداری بزرگ تا حداقل مقدار متغیر است.



شکل (٦) : الگوریتم کلونی زنبور مصنوعی [۷]

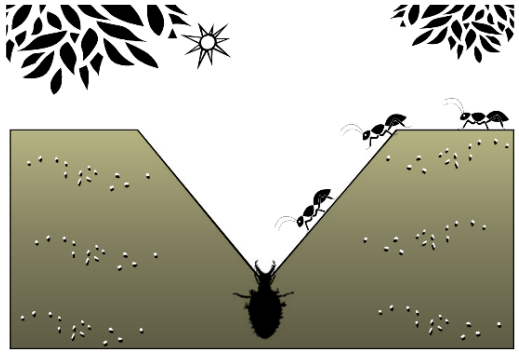
در مسئله انتخاب ویژگی، هر زیرمجموعه ویژگی به عنوان یک رشته دودویی از ۱ و ۰ کدگذاری می‌شود که در آن ۱ نشان دهنده حضور و ۰ نشان دهنده عدم حضور یک ویژگی است. برای ترسیم مسئله در این الگوریتم، موقعیت هر خفاش به عنوان مقادیر دودویی در نظر گرفته شده که منجر به تشکیل یک فضای جستجوی دودویی می‌شود. خفاش‌ها می‌توانند با چرخاندن تعداد زیادی بیت[12] به موقعیت جدیدی حرکت کنند. بنابراین، نسخه دودویی این الگوریتم باید از رویکرد متفاوتی برای به‌روزرسانی سرعت و موقعیت خود استفاده کند. بنابراین، از یک تابع انتقال برای تغییر موقعیت خفاش از صفر به یک یا بالعکس استفاده می‌شود [10].

٤-١-٥- الگوریتم بهینه‌سازی شیر مورچه

یک الگوریتم الهام‌گرفته از طبیعت جدید به نام بهینه‌ساز شیرمورچه[13] وجود دارد. پنج مرحله اصلی این الگوریتم شامل، شکار طعمه مانند قدم زدن تصادفی مورچه‌ها، ساختن تله، به دام انداختن مورچه‌ها در تله، گرفتن طعمه، و بازسازی تله انجام می‌شود. الگوریتم پیشنهادی در سه مرحله معیار گذاری شده‌است. این الگوریتم همچنین طرح‌های بهینه بهتری را برای اکثر مسائل مهندسی به کار رفته پیدا می‌کند، که نشان می‌دهد این الگوریتم در حل مسائل محدود با فضاهای جستجوی متنوع شایستگی دارد. شکل‌های بهینه به‌دست‌آمده برای پروانه کِشتی، قابلیت کاربرد الگوریتم پیشنهادی در حل مسائل واقعی با فضاهای جستجوی ناشناخته را نیز نشان می‌دهد [11].

مجموعه‌ای از شرایط را می‌توان برای شکار شیرمورچه، با موارد زیر مطرح کرد و در شکل (۷) قابل مشاهده می‌باشد [12].

- شکارها (مورچه‌ها) به طور تصادفی در فضای جستجو حرکت می‌کنند. این حرکات تحت تاثیر تله‌های شیرمورچه‌ها قرار می‌گیرد.
- شیرمورچه دارای بیشترین مقدار تابع برازندگی گودال بزرگتری می‌سازد.
- شکار مورچه توسط شیرمورچه متناسب با مقدار برازندگی آن شیرمورچه انجام می‌شود.
- هر شیرمورچه می‌تواند در هر تکرار یک مورچه را شکار کند.
- برای شبیه‌سازی مورچه‌ها به سمت شیرمورچه‌ها، دامنه راه رفتن‌های تصادفی به صورت تطبیقی کاهش می‌یابد.
- اگر مورچه توسط شیرمورچه بیهوش شود، به این معنی است که توسط شیرمورچه گرفته شده و زیر خاک کشیده می‌شود.



شکل (۷) : تله مخروطی شکل و رفتار شکار مورچه‌ها [12]

❖ یک شیرمورچه خود را به آخرین طعمه صید شده تغییر مکان می‌دهد و گودالی می‌سازد تا احتمال گرفتن طعمه دیگر پس از هر شکار افزایش یابد.

٤-١-٦- الگوریتم سنجاقک

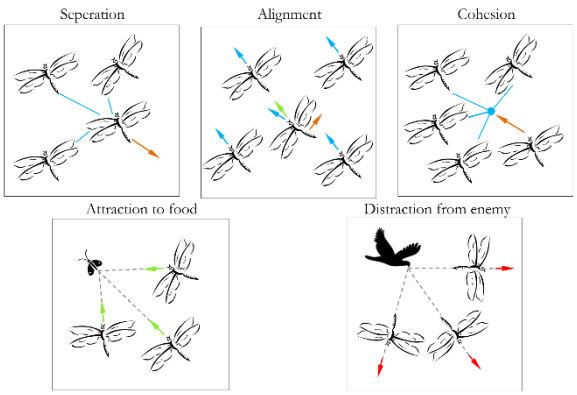
یک روش جدید بهینه‌سازی هوش ازدحام به نام الگوریتم سنجاقک[١٤] وجود دارد که الهام‌گرفته از رفتار ازدحامی ایستا و پویا سنجاقک‌ها است و از طبیعت سرچشمه گرفته است. این الگوریتم به پنج پارامتر برای کنترل انسجام، هم‌ترازی، جدایی، جذب (به سمت منابع غذایی) و حواس‌پرتی (دشمنان به بیرون) در ازدحام مجهز است. دو مرحله اساسی بهینه‌سازی، اکتشاف و بهره‌برداری، با مدل‌سازی روابط اجتماعی در تعامل سنجاقک‌ها در مسیریابی، جستجوی غذا و اجتناب از دشمنان هنگام ازدحام کردن به صورت پویا یا آماری طراحی شده‌اند. نتایج شکل ساده و دودویی آن ثابت می‌کند که الگوریتم‌های پیشنهادی قادر به بهبود جمعیت تصادفی اولیه برای یک مسئله معین، همگرا شدن به سمت بهینه جهانی و ارائه نتایج بسیار رقابتی در مقایسه با سایر الگوریتم‌های شناخته شده در این حوزه می‌باشد. نتایج همچنین نشان می‌دهد که این الگوریتم تمایل به یافتن تقریب‌های بسیار دقیق از راه‌حل‌های بهینه با توزیع یکنواخت بالا برای مسائل چندهدفه دارد [13]. رفتار دسته‌جمعی از سه اصل اولیه پیروی می‌کند.

❖ جدایی[١٥] به اجتناب از برخورد سنجاقکی به سنجاقک دیگر در همسایگی اشاره دارد.
❖ تراز[١٦] که نشان‌دهنده تطابق سرعت یک سنجاقک با سایر سنجاقک‌ها در همسایگی است.
❖ انسجام[١٧] که به گرایش سنجاقک‌ها به سمت مرکز توده اشاره دارد [١٣].

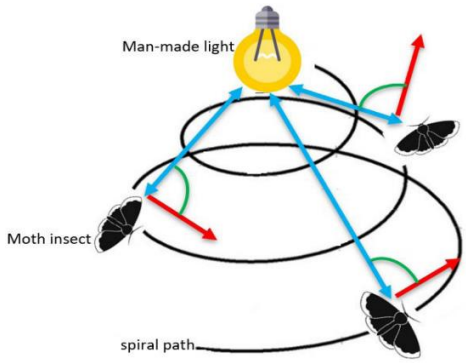
هدف اصلی هر ازدحام، بقا است، بنابراین همه افراد باید به سمت منابع غذایی جذب شوند و دشمنان ظاهری منحرف شوند. با توجه به این دو رفتار، پنج عامل اصلی در به‌روزرسانی موقعیت سنجاقک‌ها وجود دارد که در شکل (٨) قابل مشاهده است [١٣].

٤-١-٧- الگوریتم شعله پروانه

ایده اصلی الگوریتم شعله پروانه[١٨] از روند اکتشاف پروانه‌ها در هنگام جستجوی نور در طبیعت ناشی می‌شود. بسته به نوری که از ماه یا از نور مصنوعی ساخته می‌شود، هنگام حرکت به سمت نور در شب جهت‌گیری عرضی نامیده می‌شود. موقعیت پروانه‌ها بر اساس زاویه حرکت ثابت در مورد نور ناهموار کنترل می‌شود. پروانه‌ها به شکل مارپیچ حرکت می‌کنند، این رویکرد مسیر پروازی مارپیچ مهلکی را برای پروانه‌ها ایجاد می‌کند که در شکل (٩) قابل مشاهده است. برای افزایش عملکرد اکتشاف و بهره‌برداری این الگوریتم، از رویکرد شعله پروانه و روش تنوع جمعیت با استفاده از پویایی جمعیت تکاملی می‌توان استفاده کرد. رویکرد پویایی جمعیت تکاملی با



شکل (٨) : الگوهای اولیه بین سنجاقک‌ها در یک ازدحام [١٣]



شکل (٩) : پرواز مارپیچی پروانه‌ها در اطراف منبع نور [14]

موفقیت توسط چندین محقق برای کنترل تنوع جمعیتی چندین الگوریتم فراابتکاری استفاده شده است. این رویکرد ماتریس[١٩] پروانه‌ها را به دو نیمه بهترین نیمه و بدترین نیمه تقسیم می‌کند. برای بالا نگه داشتن تنوع جمعیت، بدترین نیمه با اعمال یک عملگر متقاطع یکنواخت[٢٠] بین هر راه‌حل در بدترین نیمه با یک راه‌حل تصادفی انتخاب شده از بهترین نیمه جایگزین می‌شود. این فرآیند نسبت کاوش را برای الگوریتم شعله پروانه افزایش می‌دهد [14].

٤-٢- الگوریتم‌های فرااکتشافی الهام‌گرفته از پرندگان

٤-٢-١- الگوریتم بهینه‌سازی ازدحام ذرات

بهینه‌سازی ازدحام ذرات[٢١] یک فن بهینه‌سازی تصادفی مبتنی بر جمعیت است و رفتار اجتماعی موجوداتی از قبیل، دسته پرندگان و ماهی‌ها را شبیه‌سازی می‌کند. این رویکرد را می‌توان به عنوان یک سیستم به‌روزرسانی خودکار توصیف کرد. در این روش، هر راه‌حل انتخاب شده می‌تواند یک ذره در فضای جستجو در نظر گرفته شود. هر ذره برای یافتن بهترین راه‌حل از حافظه و دانش خود استفاده می‌کند. همه ذرات دارای مقادیر برازندگی هستند که توسط یک تابع برازندگی برای بهینه‌سازی ارزیابی می‌شوند. در حین حرکت، هر ذره با تغییر سرعت بر اساس تجربه خود و با توجه به تجربه ذره همسایه، موقعیت خود را تنظیم می‌کند، بنابراین از بهترین موقعیتی که خود و همسایه‌اش با آن مواجه می‌شوند، استفاده می‌کند. ذرات با دنبال کردن جریانی از ذرات بهینه در

فضای مسئله حرکت می‌کنند. سپس این فرآیند به تعداد ثابت، بارها یا تا زمانی که حداقل خطای از پیش تعیین شده به دست آید، تکرار می‌شود [16].

این روش در ابتدا به عنوان یک فن بهینه‌سازی برای فضاهای اعداد واقعی معرفی شد و با موفقیت در بسیاری از زمینه‌ها ازقبیل، بهینه‌سازی عملکرد، آموزش شبکه عصبی مصنوعی، کنترل سیستم فازی و سایر مسائل کاربردی استفاده شده است [16]. بهینه‌سازی ازدحام ذرات شامل یک مفهوم بسیار ساده است و نمونه آن را می‌توان در چند خط کد کامپیوتری پیاده‌سازی کرد. فقط به عملگرهای ریاضی اولیه نیاز دارد و از نظر محاسباتی، حافظه مورد نیاز و هم از نظر سرعت ارزان است. در آزمایش‌های متعدد انجام شده ثابت شده است که در حل چندین نوع مسئله مؤثر بوده است [15].

٤-٢-٢- الگوریتم تغذیه مصنوعی پرندگان

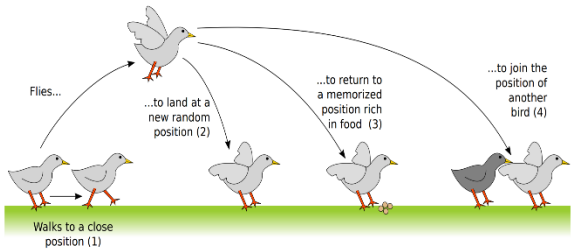
تغذیه مصنوعی پرندگان[۲۲] یک روش فراابتکاری جدید است که از رفتار بسیار پیش‌پاافتاده پرندگان در جستجوی غذا الهام گرفته است. این روش بسیار ساده و در عین حال کارآمد است و می‌تواند به راحتی با مسائل مختلف بهینه‌سازی سازگار شود. کبوترها پرندگان بسیار رایجی در شهرهای اروپایی هستند و به راحتی قابل مشاهده هستند. آنها با نوک زدن به دانه‌ها یا خرده‌های غذا روی زمین تغذیه می‌کنند. هنگامی که هیچ غذایی در دسترس ندارند، آنها با استفاده از دو حالت حرکتی پیاده‌روی و پرواز محیط خود را کشف می‌کنند. مشاهده شده است که یک کبوتر هنگام جستجوی غذا چهار نوع حرکت انجام می‌دهد، که در شکل (۱۰) قابل مشاهده است [17].

- حرکت اول، راه رفتن به موقعیت جدید نزدیک به موقعیت فعلی خود (زیرا آنها به آرامی راه می‌روند).
- حرکت دوم، پرواز و فرود در یک نیمه دلخواه (موقعیت تصادفی).
- حرکت سوم، پرواز و بازگشت به موقعیت حفظ شده غنی از غذا.
- حرکت چهارم، پرواز و فرود نزدیک به کبوتر دیگری. به طور معمول، یک کبوتر برای جستجوی غذا راه می‌رود (یک یا چند حرکت اول).

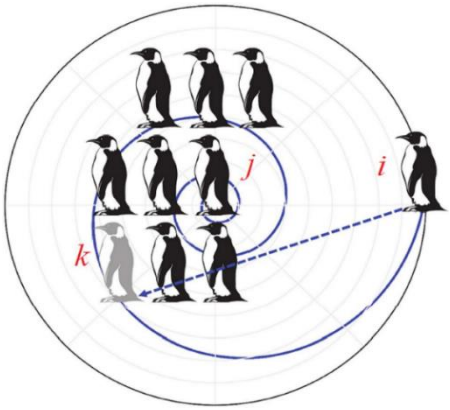
پس از مدتی، اگر غذایی پیدا نشد، پرواز می‌کند و به یک مکان تصادفی (حرکت دوم)، به یک موقعیت حفظ شده (حرکت سوم) یا پیوستن به کبوتر دیگری (حرکت چهارم) می‌رود. سپس دوباره شروع به راه رفتن می‌کند (حرکت اول) و غیره [17].

٤-٢-٣- الگوریتم کلونی پنگوئن‌های امپراتور

کلونی پنگوئن‌های امپراتور[۲۳] گونه‌ای از پرندگان هستند که در قطب جنوب زندگی می‌کنند. دمای هوا در محل زندگی آنها معمولاً در حدود ۴۰- درجه سانتیگراد است. همان‌طور که در شکل (۱۱) قابل مشاهده است، آنها جمعی متشکل از صدها پنگوئن را تشکیل می‌دهند تا گرم بمانند، با بادهای شدید و سرمای شدید محیط زندگی مقابله کنند. پس از آن، برای دستیابی به تعادل حرارتی، حرکات مارپیچی شکل انجام می‌دهند. به این ترتیب اعضای مستقر در محیط به مرکز محیط هدایت می‌شوند. ایده‌های اساسی این الگوریتم نحوه تابش گرما و نحوه انجام حرکت مارپیچی شکل است. برای مثال چند پنگوئن در اطراف پراکنده شده‌اند، آنها تصمیم می‌گیرند برای در امان ماندن از سرما،



شکل (۱۰) : چهار نوع حرکتی پرندگان هنگام جستجوی غذا [17]



شکل (۱۱) : حرکت مارپیچی‌شکل پنگوئن‌های امپراطور [18]

جمعی تشکیل دهند. همچنین شدت گرما و جذابیت هر پنگوئن بر اساس شدت تابش گرمای بدن آن محاسبه می‌شود. در رویکرد این الگوریتم، ابتدا

نمونه‌هایی از جمعیت دریافت می‌شود و سپس محاسبات هزینه روی جمعیت جدید انجام می‌گیرد. یعنی ارزیابی در حلقه درونی انجام می‌شود. در خارج از حلقه، جمعیت اولیه (هزینه‌های اولیه) با جمعیت جدید (هزینه‌های جدید) ادغام شده و بهترین‌ها انتخاب می‌شوند و این فرایند ادامه دارد [18].

٤-٣- الگوریتم‌های فرااکتشافی الهام‌گرفته از حیوانات

٤-٣-١- الگوریتم بهینه‌سازی گرگ خاکستری

بهینه‌سازی گرگ خاکستری[۲۴] یک الگوریتم تکاملی است که پیشنهاد می‌کند که گرگ‌های خاکستری بیشتر از شکار در گله، تولید مثل موفقی داشته باشند. دو گرگ خاکستری (نر و ماده) جایگاه بالاتری دارند و سایر گرگ‌های دسته را مدیریت می‌کنند. بهینه‌ساز گرگ خاکستری یکی از فنون الهام گرفته از طبیعت است که روند شکار دسته‌ای از گرگ‌های خاکستری را در طبیعت شبیه‌سازی می‌کند [20]. گرگ‌های خاکستری بیشتر ترجیح می‌دهند در یک گله زندگی کنند و اندازه گله به طور متوسط به تعداد ۵ تا ۱۲ گرگ است. نکته جالب توجه این است که آنها دارای یک سلسله مراتب اجتماعی بسیار دقیق هستند. رهبران یک گرگ نر و یک گرگ ماده هستند که گروه آلفا[۲۵] نامیده می‌شوند [19].

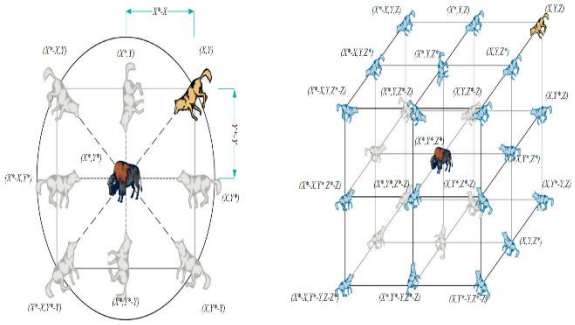
آلفا بیشتر مسئول تصمیم‌گیری در مورد شکار، محل خواب، زمان بیدار شدن و غیره است و تصمیمات آلفا به گله دیکته می‌شود. با این حال، نوعی

رفتار دموکراتیک[۲۶] نیز مشاهده شده است که در آن یک آلفا از دیگر گرگ‌های گله پیروی می‌کند. در گردهمایی‌ها، کل اعضای گله با پایین نگه داشتن دُم، آلفا را تصدیق می‌کنند. به گرگ آلفا گرگ غالب نیز گفته می‌شود، زیرا دستورات او باید توسط گله انجام شود. گرگ‌های آلفا فقط مجاز به جفت‌گیری در گله هستند. جالب اینجا است که آلفا لزوما قوی‌ترین عضو گله نیست، بلکه از نظر مدیریت گله بهترین است. این نشان می‌دهد که سازماندهی و نظم و انضباط یک گله بسیار مهمتر از قدرت آن است. همان‌طور که در شکل (۱۲) قابل مشاهده است، یک گرگ خاکستری می‌تواند موقعیت خود را در کنار فضای اطراف طعمه در هر مکان تصادفی به‌روز کند [19].

٤-٣-٢- الگوریتم جستجوی فیل

فیل‌ها[۲۷] علیرغم شخصیت رام‌شان، ساختار اجتماعی و رفتاری پیچیده‌ای دارند که در طول تاریخ تکامل یافته‌اند تا به بهترین نحو با محیط طبیعی سازگار شوند. هدف اصلی از شبکه‌سازی واحدهای خانواده در فیل‌ها، حفاظت گروهی و پرورش نسل‌های جوان است. فیل‌های ماده بالغ از گروه‌های مختلف خانواده به طور داوطلبانه به نوزادان آموزش‌هایی مانند حرکات راه رفتن، جستجوی غذا، تجربیات اجتماعی با دیگران ارائه می‌دهند. فیل‌های نر بالغ از شبکه ماده جدا می‌شوند و ماجراجویی خود را در جای دیگری دنبال می‌کنند. در هر صورت یک فیل که گروه خود را ترک کند، دیگر هرگز در گروه خود پذیرفته نخواهد شد. این الگوریتم از از مزایای الگوریتم‌های ازدحام ذرات و تکامل نسل الگوریتم ژنتیک بهره می‌برد. شکل (۱۳) عملیات جستجوی فیل را نشان می‌دهد و همانند دیگر الگوریتم‌های مطرح شده، یک فیل ماده پیشرو با بهترین مقدار برازندگی وجود دارد که سایر فیل‌های ماده به سمت آن حرکت می‌کنند. علاوه بر این، مفروضات فنی به شرح زیر برای قادر ساختن الگوریتم کارآمد مطرح شده است [21].

- هر فیل یک محدوده بصری دارد که با فاصله اقلیدسی[۲۸] محاسبه می‌شود. فاصله بصری فیل نر از ماده دورتر است. در محدوده بصری خود، فیل به طور تصادفی در جستجوی غذا یا زمین بهتر حرکت می‌کند که به معنای موقعیت بهتر در فضای جستجو است.
- هنگامی که دو فیل نر در محدوده بصری خود با یکدیگر روبرو می‌شوند، مقادیر برازندگی فعلی خود را با هم مقایسه می‌کنند. فیلی که مقدار برازندگی کمتری دارد عقب می‌نشیند و منحرف می‌شود تا از فیل برنده دور شود. جهت معکوس هر چیزی تصادفی است جز جهت فیل برنده.



شکل (۱۲) : بردارهای موقعیت دو بعدی و سه بعدی [19]

- برای سادگی، در نسخه اولیه، تنها یک گروه فیل ماده در کل الگوریتم وجود دارد و این گروه از هم جدا نخواهد شد.
- برای سادگی، مجدداً وقتی یک فیل می‌میرد، یک بچه فیل جدید از همان جنسش متولد می‌شود تا جنسیت گروه را متعادل نگه دارد و تعداد جمعیت گروه را حفظ کند.

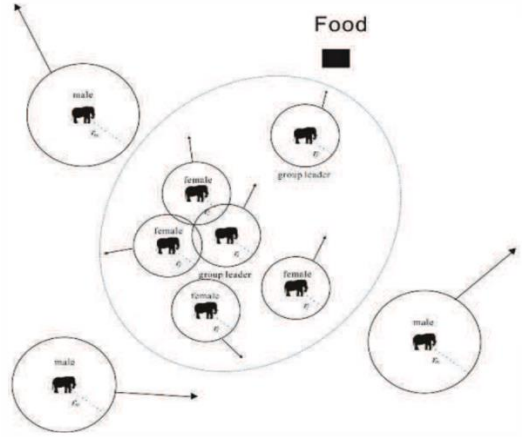
٤-٤- الگوریتم‌های فرااکتشافی الهام‌گرفته از جانوران دریایی

٤-٤-١- الگوریتم بهینه‌سازی وال

وال[۲۹] یکی از بزرگترین پستانداران جهان است که به صورت گروهی زندگی می‌کند. همان‌طور که در شکل (۱۴) قابل مشاهده است، وال‌ها رویکرد درخشانی برای شکار مانند، ماهی کوچک، ماهی مرکب دارند و هنگام جستجوی غذای خود از یک رویکرد جستجو به نام تغذیه حباب‌دار استفاده می‌کنند، جایی که مجموعه‌ای از حباب‌ها در مسیر شنا به صورت مارپیچی به سمت بالا در اطراف منبع غذا ایجاد می‌شود. در طبیعت، وال‌ها برای تعیین محل طعمه شروع به تولید حباب توری شکل می‌کنند و قبل از حمله شروع به حرکت به سمت هدف به شکل مارپیچ می‌کنند. هنگام حرکت به سمت مکانی که نشان‌دهنده مکان مطلوب است، گروهی از وال‌ها را تقلید می‌کند (هر وال نشان‌دهنده یک راه‌حل است). وال‌ها در مسیری مارپیچ شنا می‌کنند که با دمیدن تور حباب ایجاد می‌شود. از آنجا که وال یک الگوریتم مبتنی بر جمعیت است، اولین مرحله ایجاد جمعیت اولیه (یعنی گروهی از وال‌ها) است [22].

٤-٤-٢- الگوریتم بهینه‌سازی بادبان‌ماهی (نیزه ماهی)

یکی از نمونه‌های جالب رفتار اجتماعی در گروه‌های، ماهی‌ها، پرندگان و پستانداران شکار گروهی است. در شکار گروهی، شکارچیان در مقایسه با زمانی که به تنهایی شکار می‌کنند، برای کشتن طعمه به تلاش زیادی نیاز ندارند. در ساده‌ترین شکل شکار گروهی، شکارچیان سعی می‌کنند طعمه را با هماهنگی یا بدون هماهنگی بکشند، در حالی که در شکل پیچیده شکار گروهی، شکارچیان از نقش‌های خاصی برای گله‌داری و گرفتن طعمه استفاده می‌کنند. بادبان ماهی[۳۰] سریعترین ماهی در اقیانوس است که می‌تواند حداکثر



شکل (۱۳) : عملیات جستجوی فیل‌ها در تصویر انتزاعی [21]

سرعتی در حدود ۱۰۰ کیلومتر در ساعت داشته باشد. آنها به صورت گروهی شکار می‌کنند و ماهی‌ها را به سمت سطح آب می‌برند [23].

قابلیت مانور و شتاب ماهی‌ها در طول حمله برای ماهی بادبانی بسیار چالش برانگیز است. ماهی بادبانی با نیزه خود یک حرکت بریده بریده انجام می‌دهد و چندین ماهی را زخمی می‌کند یا به یک ماهی ضربه می‌زند و آن را بدون حرکت برای خوردن درمی‌آورد که در شکل(۱۵) قابل مشاهده است. بهینه‌سازی بادبان ماهی یک الگوریتم فراابتکاری مبتنی بر جمعیت است. در این الگوریتم فرض می‌شود که بادبان‌ماهی‌ها راه‌حل‌های انتخاب شده و متغیرهای مسئله، موقعیت بادبان‌ماهی‌ها در فضای جستجو هستند. بر این اساس، جمعیت به طور تصادفی تولید می شود. برخی از نکات قابل توجه برای نشان دادن توانایی این الگوریتم برای تقریب نقطه بهینه جهانی مسائل بهینه‌سازی را می‌توان به شرح زیر مطرح کرد [23]:

- رویکرد احاطه‌ای پیشنهادی، یک همسایگی در اطراف راه‌حل‌ها فراهم می‌کند که فرایند اکتشاف را برای فضای جستجو تضمین می‌کند.
- قابلیت مانور ماهی‌ها در اطراف بهترین عامل در زمانی که شمارشگر تکرار افزایش می‌یابد باعث افزایش فرایند بهره‌برداری می‌شود.
- به‌روزرسانی موقعیت ماهی‌ها باعث ایجاد رفتارهای حرکتی متنوع برای ماهی‌ها می‌شود که در اطراف ماهی بادبانی موقعیت اکتشاف شده را متنوع می‌کند.
- کاهش تدریجی پارامترهای این الگوریتم، باعث ایجاد تعادل پویا بین فرایند اکتشاف و بهره‌برداری می‌شود.
- همگرایی این الگوریتم زمانی تضمین می‌شود که شدت قدرت حمله ماهی بادبانی به طور تطبیقی در هر تکرار کاهش یابد.
- از آنجایی که ماهی بادبانی به موقعیت ماهی مناسب منتقل می‌شود، مناطق امیدوارکننده فضای جستجو در طول بهینه‌سازی ذخیره می‌شوند.

٤-٥- الگوریتم‌های فرااکتشافی الهام‌گرفته از گیاهان

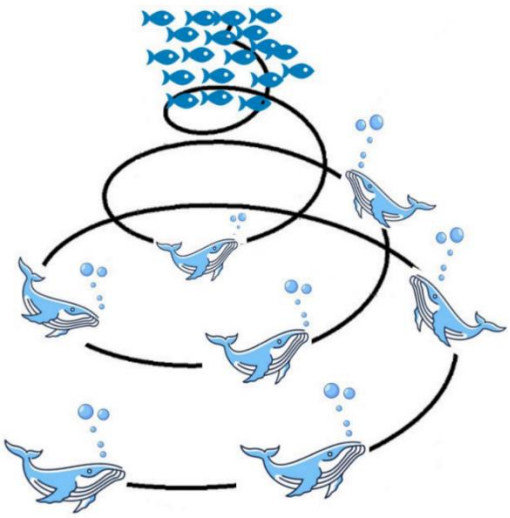
٤-٥-١- الگوریتم بهینه‌سازی علف‌های هرز تهاجمی

بهینه‌سازی علف‌های هرز تهاجمی[۳۱] با الهام از پدیده استعمار علف‌های هرز مهاجم در طبیعت، بر پایه بیولوژی[۳۲] علف‌ها است. نشان داده شده است که بررسی خواص علف‌های هرز مهاجم، منجر به یک الگوریتم بهینه‌سازی قدرتمند می‌شود. رفتار استعمار علف‌های هرز در یک مزرعه زراعی را می‌توان تشریح کرد. در ابتدا علف‌های هرز به وسیله ایجاد پراکندگی به یک سیستم کشت (مزرعه) حمله می‌کنند و فضاهای بین محصولات کشاورزی را اشغال می‌کنند. هر علف هرز مهاجم منابع بلااستفاده مزرعه را می‌گیرد و به علف هرز گلدار تبدیل می‌شود و به طور مستقل علف‌های هرز جدیدی تولید می‌کند [24].

تعداد علف‌های هرز جدید تولید شده توسط هر علف هرز گلدار به مقدار برازندگی علف هرز گلدار در آن مجموعه بستگی دارد. آن دسته از علف‌های هرز که سازگاری بهتری با محیط دارند و منابع استفاده نشده بیشتری را مصرف کرده‌اند، سریعتر رشد می‌کنند و بذر بیشتری تولید می‌کنند. علف‌های هرز تولید شده جدید به طور تصادفی در سطح مزرعه پخش شده و به علف‌های هرز گلدار تبدیل می‌شوند. این روند تا رسیدن به حداکثر تعداد علف‌های هرز در مزرعه به دلیل محدودیت منابع ادامه می‌یابد. اکنون فقط آن دسته از علف‌های هرز با مقدار برازندگی بهتر می‌توانند زنده بمانند و علف‌های هرز جدید تولید کنند. این رقابت بین علف‌های هرز باعث می‌شود که آنها در طول زمان به خوبی با محیط سازگار شده و بهبود پیدا کنند [24].

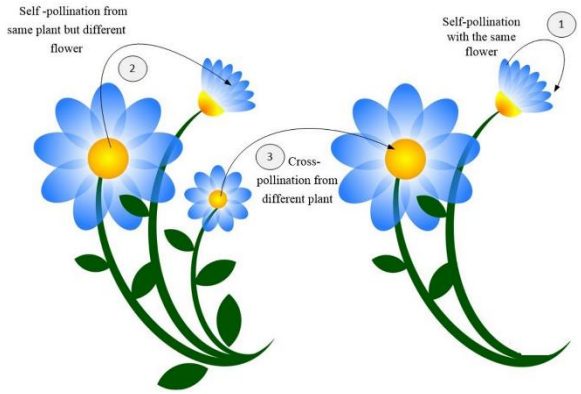
٤-٥-٢- الگوریتم گرده افشانی گل

الگوریتم گرده افشانی گل[۳۳] یکی از جدیدترین الگوریتم‌های فراابتکاری است که از رفتار گرده افشانی گیاهان گلدار الهام گرفته شده است. گرده افشانی شامل انتقال دانه‌های گرده از قسمت نر گل به تخمک‌هایی است که در قسمت ماده از طریق گرده افشان‌هایی مانند پرندگان، پروانه‌ها، زنبورها و خفاش‌ها منتقل می‌شوند. با توجه به فرایندهای انتقال گرده، گرده افشانی می‌تواند دو نوع زنده و غیرزنده باشد. گرده افشانی زنده به انتقال گرده از طریق گرده افشان‌ها (یعنی حشرات یا سایر حیوانات) اشاره دارد. در مقابل، گرده افشانی غیرزنده به هیچ گرده افشانی برای انتقال گرده نیاز ندارد (یعنی از حامل‌های غیر حیوانی مانند باد و آب استفاده می‌کند). علاوه بر این، گرده افشانی را می‌توان با گرده افشانی خود یا گرده افشانی متقابل انجام داد. خود گرده افشانی زمانی اتفاق می‌افتد که گرده از قسمت نر به قسمت ماده همان گل یا به گل دیگری از همان گیاه منتقل شود. گرده افشانی متقاطع به انتقال گرده از گل یک گیاه به گل گیاه دیگر اشاره دارد که در شکل (۱۶) قابل مشاهده است [25].



شکل (١٤) : روش تغذیه حباب‌دار وال‌ها [22]

شکل (١٥) : وضعیت رفتاری شکار گروهی بادبان‌ماهی‌ها [23]



شکل (١٦) : روش‌های گرده افشانی گل [25]

فرآیند گرده افشانی، پایداری گل و رفتار گرده افشان‌ها را می‌توان در قوانین زیر ایده‌آل کرد [26]:

- قاعده ۱: گرده افشانی زنده و متقابل را می‌توان به عنوان گرده افشانی جهانی تشخیص داد که گرده افشان‌ها از توزیع لِوی[٣٤] پیروی می‌کنند.
- قاعده ۲: غیرزنده و خود گرده افشانی را می‌توان به گرده افشانی موضعی تعبیر کرد.
- قاعده ۳: خاصیت پایداری گل را می‌توان نسبت تولید مثلی در نظر گرفت که متناسب با درجه تشابه دو گل است.
- قاعده ۴: به دلیل مجاورت فیزیکی و باد، گرده افشانی محلی نسبت به گرده افشانی جهانی برتری جزئی دارد.

٤-٦- الگوریتم‌های فرااکتشافی الهام‌گرفته از انسان

٤-٦-١- الگوریتم ژنتیک

در الگوریتم ژنتیک[٣٥]، فرایند بهینه‌سازی با تولید مجموعه‌ای از راه‌حل‌ها که نمایانگر جمعیت اولیه هستند شروع می‌شود، سپس عملگرهای ژنتیکی (به عنوان مثال، انتخاب، تقاطع و جهش) بر روی راه‌حل‌های انتخاب شده از جمعیت اعمال می‌شوند. این مرحله تا زمانی که شرایط توقف را برآورده نکرده است، به طور مکرر تکرار می‌شود. به عنوان مثال، راه‌حل نهایی و تقریباً مطلوب یافت می‌شود یا تعداد تکرارهای از پیش تعیین شده به دست می‌آید. در هر تکرار، راه‌حل‌ها با استفاده از یک تابع برازندگی که اهمیت مقدار به دست آمده را برآورد می‌کند، ارزیابی می‌شوند. تقاطع و جهش عملگرهای اصلی تکاملی در الگوریتم ژنتیک هستند [27].

در طول فرآیند بهینه‌سازی، این عملگرها برای تولید راه‌حل‌های جدید، که منعکس‌کننده عملکرد ژنتیک است، به افراد در جامعه اعمال می‌شود. ایجاد شیوه‌های جدید با انتخاب دو والدین از گروه جمعیت، بر اساس رویکرد روش‌های انتخاب (به عنوان مثال، تصادفی، رولت یا مسابقات) شروع می‌شود، سپس عملگر تقاطع (به عنوان مثال، تک نفره، دو نفره، یکنواخت) روی این والدین اعمال می‌شود و سپس دو فرزند به وجود می‌آورد. در عملیات جهش، تغییرات محلی (به عنوان مثال، به طور تصادفی) برای هر دو فرزند اعمال می‌شود. سپس با در نظر گرفتن فرزندان تولید شده که جایگزین برخی راه‌حل‌ها در جمعیت بر اساس رویکرد جایگزینی نخبه‌گرایی هستند، باعث به‌روز شدن جمعیت می‌شود. سپس جمعیت به‌روز شده برای تکرار بعدی در نظر گرفته می‌شود [27].

٤-٦-٢- الگوریتم جستجوی صف

حضور در صف یک وضعیت رایج در زندگی مردم است، مانند صف بستن برای انجام معاملات تجاری، خرید و فروش و غیره. کارکنان می‌توانند در صف‌های مختلف به مشتریان خود خدمات ارائه دهند، اما اغلب مشتریان صفی را انتخاب می‌کنند که در کمترین زمان بتوانند خدمات دریافت کنند. به طور کلی، کارکنان با قابلیت‌های تجاری قوی‌تر به مشتریان بیشتری خدمت می‌کنند. هر خدمت مشتری عمدتاً تحت تأثیر کارکنان یا خود مشتری است. با الهام از فرآیند صف‌بندی تشریح‌شده، یک الگوریتم فراابتکاری جدید به نام الگوریتم جستجوی صف[٣٦] پیشنهاد شده است. پنج اصل این الگوریتم به شرح زیر مطرح شده است [28]:

- اصل ۱: تعداد مشتریان در صف محدود و ثابت است. این اصل تفسیر می‌شود که تعداد افراد مورد استفاده در جستجوی راه‌حل بهینه در الگوریتم از پیش تنظیم و ثابت شده است.
- اصل ۲: برخی از مشاغل باید مدیریت شوند در حالی که برخی دیگر اختیاری هستند. برای مثال در معاینه فیزیکی، معاینات اولیه ضروری است اما برخی از معاینات خاص اختیاری هستند. این اصل به این معنی است که برخی از رویه‌های به‌روزرسانی برای افراد ضروری هستند در حالی که بقیه قابل انتخاب هستند.
- اصل ۳: کارکنان با قابلیت‌های تجاری قوی‌تر در مدت زمان مشابه به مشتریان بیشتری خدمت خواهند کرد. پس از هر تکرار در الگوریتم، چند نفر اول با توجه به مقدار برازندگی هر فرد مورد مقایسه قرار می‌گیرند. هر یک از چند نفر اول بهترین یک منطقه جستجوی محلی را تعیین می‌کنند. این اصل به این معنی است که منطقه جستجوی محلی تعیین شده توسط یکی از چند نفر اول بهترین‌ها با مقدار برازندگی بهتر به افراد بیشتری در الگوریتم توزیع می‌شود.
- اصل ۴: هر خدمات مشتری عمدتاً تحت سلطه خود مشتری یا کارکنان است. این اصل به این معنی است که به‌روزرسانی یک فرد می‌تواند بر اساس خودش یا چند نفر اول بهترین‌ها در الگوریتم محقق شود.
- اصل ۵: نوسانات هر خدمات مشتری می‌تواند توسط خود مشتری، کارکنان و سایر مشتریان ایجاد شود. به عنوان مثال، عدم رعایت دقیق نظم صف باعث ایجاد درجه خاصی از بی‌نظمی می‌شود. با بدتر شدن این اختلال، تأثیر سایر مشتریان بر یکی از خدمات مشتری افزایش می‌یابد. در نهایت، این خدمات مشتری عمدتاً تحت تأثیر سایر مشتریان قرار خواهد گرفت. تحت برخی شرایط خاص، افراد دیگر می‌توانند تأثیر مهمی بر به‌روزرسانی یک فرد داشته باشند.

جدول (۸) : بررسی نقاط ضعف و قوت الگوریتم‌های فرااکتشافی

ردیف	الگوریتم	نقاط قوت	نقاط ضعف	مراحل اکتشاف و بهره برداری	مراجع
۱	بهینه‌سازی کلونی مورچه	۱- برای مسئله فروشنده دوره گرد کارآمد است. ۲- دارای موازی‌سازی ذاتی است.	۱- به طور مکرر تصمیمات تصادفی می‌گیرد. ۲- به سختی می‌توان تحلیل نظری مطرح کرد.	مسیریابی از طریق ماده فِرمون، مسیریابی در محیط گراف	[29, 4]
۲	کلونی زنبور مصنوعی	۱- دارای خود سازماندهی است. ۲- از داده های هوشمند جمعی استفاده می‌کند.	۱- تعداد پارامتر زیاد. ۲- جستجوی محلی ضعیف.	رقص زنبورهای پیشاهنگ، ارزیابی غذا، پرواز	[30, 29]
۳	کرم شب‌تاب	۱- درک مفاهیم رابطه‌ها و کدنویسی آن آسان است.	۱- تعداد پارامتر زیاد. ۲- همگرایی آهسته. ۳- نمونه مقالات کمی وجود دارد.	جاذبه، حرکت	[33, 29]
۴	خفاش	۱- ساختار ساده ۲- پارامترهای کم ۳- استحکام قوی	۱- سرعت همگرایی کند است. ۲- دقت بهینه‌سازی کم است.	تنظیم طول موج، پرواز	[34, 30]
۵	بهینه‌سازی شیرموچه	۱- پیاده‌سازی آسان ۲- زمان اجرای معقول ۳- پارامترهای بسیار کمی برای تنظیم دارد و انعطاف‌پذیز است.	۱- از همگرایی زودرس رنج می‌برد. ۲- مشخصه همگرای نظری وجود ندارد. ۳- توزیع احتمال بر اساس نسل‌ها تغییر می‌کند.	قدم زدن تصادفی مورچه‌ها، ساخت تله، به دام انداختن مورچه‌ها در تله، گرفتن طعمه و بازسازی تله	[36, 12]
۶	سنجاقک	۱- اجرای بسیار ساده و آسان. ۲- استفاده از پارامترهای کم. ۳- زمان همگرایی الگوریتم معقول است.	۱- حافظه داخلی ندارد که بتواند منجر به همگرایی زودهنگام به بهینه محلی شود. ۲- به راحتی در بهینه محلی گیر می‌کند زیرا نرخ بهره‌برداری بالایی دارد.	کنترل انسجام، هم‌ترازی، جدایی، جذب (به سمت منابع غذایی) و حواس‌پرتی (دفع دشمنان)	[37, 14]
۷	شلعه پروانه	۱- سادگی در فهم رویکرد الگوریتم و پیاده‌سازی آن. ۲- توانایی جستجوی سریع. ۲- ترکیب ساده با دیگر الگوریتم‌ها.	۱- در بهینه‌سازی بد محلی گیر می‌کند زیرا به جای اکتشاف، به بهره‌برداری تمرکز می‌کند.	به دور شعله چرخیدن و جهت‌گیری عرضی	[38, 15]
۸	بهینه‌سازی ازدحام ذرات	۱- تعادل مناسب بین فرایند اکتشاف و بهره‌برداری ایجاد می‌کند. ۲- سرعت جستجوی بالایی دارد. ۳- با در نظر گرفتن موقعیت بهینه جهانی و بهترین موقعیت یک ذره، دارای حافظه است.	۱- همگرایی زودرس. ۲- به دام انداختن بهینه محلی در مسائل چند وجهی آسان است. ۳- تنظیم پارامتر به مسائل مختلفی بستگی دارد.	هر راه‌حل یک ذره در فضای جستجو است، تغییر موقعیت راه‌حل با تغییر سرعت ذره امکان‌پذیر است.	[32, 17]
۹	تغذیه مصنوعی پرندگان	۱- سادگی در فهم روابط و پیاده‌سازی. ۲- کارآمد است و می‌تواند به راحتی با مسائل مختلف بهینه‌سازی سازگار شود.	1- نمونه مقالات کمی وجود دارد.	راه‌رفتن، پرواز کردن، جستجوی غذا	[19]
۱۰	کلونی پنگوئن‌های امپراتور	۱- انواع مختلفی از محدودیت‌ها را بسیار کارآمد مدیریت می‌کند و راه‌حل‌های بهتری ارائه می‌دهد.	۱- نمونه مقالات کمی وجود دارد.	حرکات مارپیچی، نحوه تابش گرما.	[39, 20]
۱۱	بهینه‌سازی فیل	۱- بهینه‌سازی سراسری ۲- خوشه‌بندی داده ۳- موقعیت‌یابی گره‌های اثربخش	۱- عدم تحمل‌پذیری خطا	حفاظت گروهی و پرورش نسل، حرکت تصادفی برای پیداکردن غذا یا محیط بهتر	[35, 21]

ردیف	الگوریتم	نقاط قوت	نقاط ضعف	مراحل اکتشاف و بهره برداری	مراجع
۱۲	بهینه‌سازی گرگ خاکستری	۱- تعداد پارامتر کم. ۲- استفاده زیاد در انتخاب ویژگی. ۳- بهینه‌ساز چند هدفه	۱- عدم تحمل‌پذیری خطا	ردیابی، احاطه کردن، حمله، حرکت و جابجایی گرگ	[35, 30]
۱۳	بهینه‌سازی وال	۱- به پارامترهای کمی نیاز دارد. ۲- توانایی جستجوی قدرتمند. ۳- رویکرد یادگیری مبتنی بر مخالفت می‌تواند توانایی کاوش آن را بهبود بخشد.	۱- همگرایی آهسته. ۲- همگرایی زودرس در مسئله چند وجهی.	محاصره کردن، جستجوی طعمه، مانور دادن	[32, 30]
۱۴	بهینه‌سازی ماهی بادبانی	۱- بادبان‌ماهی‌ها مناطق امیدوارکننده فضای جستجو را کاوش می‌کنند. ۲- از مناطق بهینه جهانی بسیار دقیق بهره‌برداری می‌کنند.	۱- نمونه مقالات کمی وجود دارد.	مانور و شتاب ماهی‌ها، زخمی‌کردن ماهی‌ها توسط نیزه ماهی بادبانی	[25]
۱۵	گرده افشانی گل	۱- می‌تواند همگرایی جهانی را تضمین کند.	۱- هیچ اطلاعاتی در مورد میزان همگرایی ارائه نمی‌دهد. ۲- نمونه مقالات کمی وجود دارد.	گرده افشانی گیاهان گلدار، درجه تشابه دو گل	[40, 26]
۱۶	علف‌های هرز تهاجمی	۱- نرخ همگرایی و همچنین سطح خطای نهایی مناسبی دارد. ۲- در برابر مقادیر پارامترهای مختلف بسیار پایدار و کارآمد است.	۱- نمونه مقالات کمی وجود دارد.	پر کردن فضای بین محصولات کشاورزی، تولید علف‌های هرز جدید توسط علف‌های هرز گلدار	[28]
۱۷	جستجوی صف	۱- به تعداد پارامتر کمی نیاز دارد. ۲- به سادگی قابل پیاده‌سازی برای مسائل جدید است. ۳- انتخاب مناسبی برای حل مسائل بهینه‌سازی مهندسی است.	۱- نمونه مقالات کمی وجود دارد.	تعداد مشتریان، قابلیت کارکنان، رعایت نظم و دقت	[30]
۱۸	ژنتیک	۱- از یک رویکرد احتمال تصادفی استفاده می‌کند. ۲- از رویکردهای سنتی دقیق‌تر است. ۳- به سادگی با الگوریتم‌های دیگر ترکیب می‌شود.	۱- به سادگی در دام بهینه محلی می‌افتد. ۲- مقدار پارامتر به طور جدی بر کیفیت راه‌حل تأثیر می‌گذارد. ۳- در نسخه‌های بهبودیافته به منابع محاسباتی بیشتری نیاز است.	انتخاب، تقاطع، جهش، به‌روزرسانی جمعیت، روش‌های انتخاب تصادفی	[32, 29]

۵-تایج ارزیابی کارایی الگوریتم‌ها

ارزیابی کارایی کمی برای پیش‌بینی خطای نرم‌افزار به عنوان نمونه، عملکرد سه الگوریتم انتخاب ویژگی دودویی، شعله پروانه توسعه‌یافته، کلونی زنبور و وال توسعه‌یافته با به کارگیری مدل یادگیری ماشین جنگل تصادفی و انتخاب ۲۰ مجموعه داده پرومیس مورد آزمایش و بررسی قرار گرفته است. ۲۰ مجموعه داده پرومیس در جدول (۹) نشان داده شده است.

اطلاعات جمع‌آوری‌شده از اندازه‌گیری به موقع و مستمر از فرآیندهای مختلف و محصول نرم‌افزاری، ویژگی نرم‌افزاری نام دارد. ویژگی‌های نرم‌افزار همچنین می‌توانند برای ایجاد یک موضوع نرم‌افزاری بهبودیافته، که رویه مرتبط با آن را بهبود می‌بخشد، مورد استفاده قرار گیرد. همچنین می‌توان به این نکته اشاره کرد که ویژگی‌های نرم‌افزار باید از ابتدا جمع‌آوری شوند، همچنین برآورد هزینه و تلاش محصول با احتیاط انجام شود. این امر به نظارت بر قابلیت اطمینان و قابلیت نگهداری محصول نهایی کمک می‌کند [۴۳]. ویژگی‌های مجموعه داده پرومیس در جدول (۱۰) قابل مشاهده است.

در جدول (۱۱) به منظور دستابی به مقدار مطلوب ناحیه زیر منحنی، الگوریتم جنگل تصادفی استفاده شده و برای افزایش کارایی آن می‌توان با تنظیم پارامترها به نتیجه بهتری رسید. پارامترهای مختلفی برای این الگوریتم وجود دارد و به طور کلی از سه پارامتر تعداد درخت تخمین‌زن ، حداکثر ویژگی ، حداکثر عمق استفاده شده است. سه مقدار ۱۰ و ۵۰ و ۲۰۰ برای پارامتر اول با نام تعداد درخت تخمین‌زن مورد استفاده قرار گرفته است و در نتیجه بهترین مقدار به دست آمده از آن حفظ می‌شود. همان‌طور که قابل مشاهده است یک مقدار به تنهایی نمی‌تواند در تمامی مجموعه داده‌ها بهترین عملکرد را داشته

جدول (۹) : مجموعه داده‌های استفاده‌شده در تحقیق [22, 14]

مجموعه داده	نسخه	تعداد نمونه‌ها	تعداد خطا	نرخ خطا
Ant	1/7	۷۴۵	۱۶۶	0/223
Camle	1/0	۳۳۹	۱۳	0/038
Camel	1/2	۶۰۸	۲۱۶	0/355
Camel	1/4	۸۷۲	۱۴۵	0/166
Camel	1/6	۹۶۵	۱۸۸	0/195
Jedit	3/2	۲۷۲	۹۰	0/331
Jedit	4/0	۳۰۶	۷۵	0/245
Jedit	4/1	۳۱۲	۷۹	0/253
Jedit	4/2	۳۶۷	۴۸	0/131
Jedit	4/3	۴۹۲	۱۱	0/022
Log4j	1/0	۱۳۵	۳۴	0/252
Log4j	1/1	۱۰۹	۳۷	0/339
Log4j	1/2	۲۰۵	۱۸۹	0/922
Lucene	2/0	۱۹۵	۹۱	0/467
Lucene	2/2	۲۴۷	۱۴۴	0/583
Lucene	2/4	۳۴۰	۲۰۳	0/597
Xalan	2/4	۷۲۳	۱۱۰	0/152
Xalan	2/5	۸۰۳	۳۸۷	0/842
Xalan	2/6	۸۸۵	۴۱۱	0/464
Xalan	2/7	۹۰۹	۸۹۸	0/988

جدول (۱۰) : ویژگی‌های مجموعه داده‌های پرومیس [27]

ویژگی	توضیح
wmc	تعداد توابع تعریف‌شده در داخل یک کلاس را شمارش می‌کند.
dit	عمق یک کلاس در سلسله مراتب کلاس از ریشه وراثت را محاسبه می‌کند.
noc	تعداد فرزندان مستقیم یک کلاس را شمارش می‌کند.
cbo	تعداد کلاس‌های مرتبط با کلاس را می‌شمارد.
rfc	تعداد توابع متمایز فراخوانی‌شده، توسط یک کلاس در پاسخ به پیام دریافتی را می‌شمارد.
lcom	تعداد توابعی که، یک فیلد را به اشتراک نمی‌گذارند، با جفت روش‌هایی که دارای اشتراک هستند، شمارش می‌کند.
ca	تعداد کلاس‌های وابسته، برای یک کلاس مشخص را شمارش می‌کند.
ce	تعداد کلاس‌هایی که یک کلاس، به آن‌ها وابستگی دارد می‌شمارد.
npm	تعداد توابع عمومی تعریف‌شده در یک کلاس را شمارش می‌کند.
lcom3	تعداد اجزای متصل، در یک گراف را می‌شمارد.
loc	تعداد کل خطوط کد یک کلاس را می‌شمارد.
dam	نسبت ویژگی‌های خصوصی در یک کلاس را محاسبه می‌کند.
moa	تعداد اعضای داده اعلام‌شده به عنوان نوع کلاس را می‌شمارد.
mfa	کسری از توابع به ارث رسیده توسط یک کلاس را، به توابع قابل دسترسی توسط توابع تعریف‌شده در کلاس نشان می‌دهد.
cam	انسجام بین توابع یک کلاس را بر اساس لیست پارامترها محاسبه می‌کند.
ic	تعداد کلاس‌های اجداد متصل‌شده به یک کلاس را شمارش می‌کند.
cbm	تعداد توابع جدید یا تعریف‌شده را که با توابع ارث‌بری‌شده همراه شده‌اند، می‌شمارد.
amc	میانگین اندازه تابع را برای هر کلاس اندازه‌گیری می‌کند.
max_cc	حداکثر تعداد مسیرهای منطقی مستقل در یک تابع را محاسبه می‌کند.
avg_cc	متوسط تعداد، مسیرهای منطقی مستقل در یک تابع را محاسبه می‌کند.

جدول (۱۱) : پارامترهای الگوریتم جنگل تصادفی

مجموعه داده	n_estimators			max_features			max_depth		
	۱۰	۵۰	۲۰۰	Auto	Sqrt	Log2	۱	۷	۲۰
Ant-1.7	۰/۷۲۶۱	۰/۷۴۳۸	۰/۷۴۵۹	۰/۷۳۸۴	۰/۷۴۶۹	۰/۷۴۸۱	۰/۷۶۱۵	۰/۷۵۵۱	۰/۷۴۲۲
Camle-1.0	۰/۶۲۵۳	۰/۵۶۴۷	۰/۶۲۷۱	۰/۶۲۷۲	۰/۶۲۳۳	۰/۶۲۵۲	۰/۶۸۳۰	۰/۶۳۱۳	۰/۵۶۴۶
Camel-1.2	۰/۵۸۶۷	۰/۶۲۴۲	۰/۶۳۸۱	۰/۶۴۱۹	۰/۶۳۸۴	۰/۶۴۸۸	۰/۵۶۶۶	۰/۶۴۰۵	۰/۶۳۳۴
Camel-1.4	۰/۶۰۸۲	۰/۶۳۱۸	۰/۶۴۲۶	۰/۶۴۲۳	۰/۶۱۵۹	۰/۶۳۵۶	۰/۶۵۹۶	۰/۶۷۱۹	۰/۶۳۵۴
Camel-1.6	۰/۶۰۷۴	۰/۶۳۰۲	۰/۶۳۷۹	۰/۶۳۰۱	۰/۶۳۸۵	۰/۶۴۲۴	۰/۶۰۵۴	۰/۶۶۰۸	۰/۶۳۱۲
Jedit-3.2	۰/۷۴۴۳	۰/۷۴۸۳	۰/۷۵۸۵	۰/۷۵۳۶	۰/۷۵۶۲	۰/۷۵۸۱	۰/۷۴۱۱	۰/۷۶۱۴	۰/۷۳۹۳
Jedit-4.0	۰/۷۲۸۱	۰/۷۱۹۲	۰/۷۳۲۹	۰/۷۲۸۴	۰/۷۲۹۳	۰/۷۲۷۴	۰/۷۲۱۸	۰/۷۲۴۰	۰/۷۱۷۱
Jedit-4.1	۰/۶۹۳۴	۰/۷۷۸۸	۰/۷۸۳۰	۰/۷۸۷۱	۰/۷۹۸۵	۰/۷۷۹۹	۰/۷۵۸۴	۰/۷۹۲۰	۰/۷۸۶۱
Jedit-4.2	۰/۶۶۰۱	۰/۷۲۹۳	۰/۷۳۰۷	۰/۷۵۰۵	۰/۷۶۵۶	۰/۷۴۰۷	۰/۸۰۵۵	۰/۷۵۸۴	۰/۷۴۳۸
Jedit-4.3	۰/۶۴۷۳	۰/۶۴۷۳	۰/۶۴۹۰	۰/۶۵۰۸	۰/۶۴۹۱	۰/۶۵۰۶	۰/۶۴۲۶	۰/۶۳۶۸	۰/۶۴۷۴
Log4j-1.0	۰/۶۷۱۸	۰/۶۹۹۶	۰/۷۰۳۴	۰/۶۶۶۸	۰/۶۶۳۰	۰/۶۸۸۸	۰/۷۴۰۵	۰/۶۷۹۲	۰/۶۸۷۶
Log4j-1.1	۰/۷۴۲۶	۰/۷۵۱۳	۰/۷۵۴۱	۰/۷۷۶۶	۰/۷۶۲۹	۰/۷۷۵۹	۰/۷۸۲۱	۰/۷۳۶۲	۰/۷۴۲۱
Log4j-1.2	۰/۶۹۹۷	۰/۷۱۶۰	۰/۷۲۲۸	۰/۷۵۰۵	۰/۶۸۸۲	۰/۷۱۶۰	۰/۷۳۴۶	۰/۷۲۷۳	۰/۷۱۲۷
Lucene-2.0	۰/۶۲۶۰	۰/۶۷۱۵	۰/۶۸۰۴	۰/۶۹۶۱	۰/۶۹۳۹	۰/۶۹۳۳	۰/۶۹۲۵	۰/۶۷۶۳	۰/۶۹۳۹
Lucene-2.2	۰/۵۹۸۵	۰/۶۲۵۳	۰/۶۰۶۵	۰/۶۲۵۴	۰/۵۹۴۹	۰/۶۱۶۸	۰/۶۲۲۷	۰/۶۳۳۹	۰/۶۲۲۲
Lucene-2.4	۰/۷۱۱۳	۰/۷۱۷۲	۰/۷۳۵۳	۰/۷۳۸۶	۰/۷۲۸۴	۰/۷۳۹۶	۰/۶۶۸۰	۰/۷۲۱۵	۰/۷۲۲۷
Xalan-2.4	۰/۷۰۹۴	۰/۷۰۶۹	۰/۷۳۶۸	۰/۷۳۱۴	۰/۷۱۵۶	۰/۷۴۰۸	۰/۶۷۱۷	۰/۷۲۰۳	۰/۷۳۷۰
Xalan-2.5	۰/۶۲۸۷	۰/۶۹۳۵	۰/۷۱۵۴	۰/۷۱۰۰	۰/۷۰۴۸	۰/۷۱۸۱	۰/۶۰۰۴	۰/۷۰۲۱	۰/۷۱۹۴
Xalan-2.6	۰/۷۵۶۰	۰/۷۷۹۱	۰/۷۸۴۰	۰/۷۸۱۱	۰/۷۷۲۸	۰/۷۷۶۰	۰/۷۳۰۴	۰/۷۶۷۰	۰/۷۸۳۳
Xalan-2.7	۰/۷۹۶۸	۰/۷۹۶۸	۰/۷۹۶۸	۰/۷۹۶۸	۰/۷۹۶۸	۰/۷۹۶۸	۰/۸۰۰۵	۰/۸۶۸۳	۰/۷۹۶۸
میانگین	۰/۶۸۲۶	۰/۶۹۶۹	۰/۷۰۸۸	۰/۷۱۰۳	۰/۷۰۴۲	۰/۷۱۱۰	۰/۶۹۹۵	۰/۷۱۳۳	۰/۷۰۲۹

باشد و در هر مجموعه داده عملکرد متفاوتی دارد. با انجام میانگین‌گیری به این نتیجه رسیده شد که عدد ۲۰۰ مقدار مطلوبی است.

در جدول (۱۲) هر مجموعه داده با الگوریتم طبقه‌بند جنگل تصادفی به همراه پارامترهای بهبودیافته و رویکرد انتخاب ویژگی پیشنهادی در ۱۰۰ تکرار با تعداد جمعیت ۴۰، مورد آزمایش قرار گرفت. به دلیل استفاده از فن اعتبارسنجی متقابل ۱۰ قسمت، بعد از هر آزمایش ۱۰ قسمت وجود دارد که برای کسب نتیجه بین آنها میانگین‌گیری انجام می‌شود. برای نمایش دقیق‌تر نتایج تحقیق، مقادیر حاصله، بیشترین، کمترین، میانگین و میانه ناحیه زیر منحنی برای تمامی مجموعه داده‌ها ثبت شده است.

در شکل (۱۷) نمودار میله‌ای میانگین مقدار ناحیه زیر منحنی برای هر مجموعه داده قابل نمایش است. در محور افقی اسامی هر ۲۰ مجموعه داده پرومیس و در محور عمودی مقادیر میانگین مقدار ناحیه زیر منحنی مطرح شده است. همانطور که قابل مشاهده است، بیشترین میانگین مقدار ناحیه زیر منحنی به مجموعه داده xalan-2.7 و کمترین میانگین مقدار ناحیه زیر منحنی به مجموعه داده Camel-1.2 تعلق دارد. با مشاهده این نمودار می‌توان نتیجه گرفت که نتایج ۲۰ مجموعه داده پرومیس در بازه ۰/۶۵ و ۰/۹۰ است.

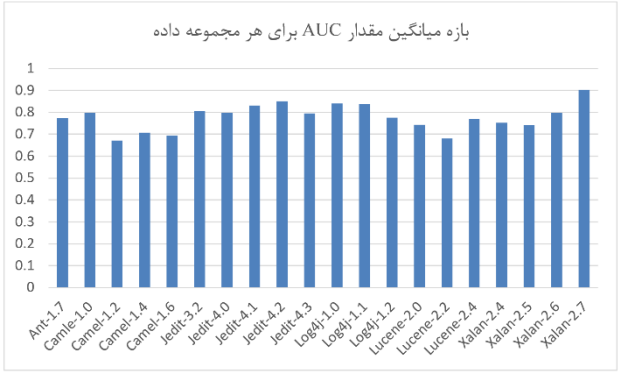
در جدول (۱۳) اهمیت مبحث انتخاب ویژگی توسط الگوریتم‌های فرااکتشافی مطرح‌شده و ویژگی‌های انتخاب‌شده برای تمامی ۲۰ مجموعه داده پرومیس به شکل آرایه‌ای از مقادیر دودویی که صفر به معنی عدم انتخاب ویژگی و یک به معنی انتخاب ویژگی، قابل مشاهده است. همچنین تعداد ویژگی‌های استفاده‌شده در هر مجموعه داده ثبت شده است که کمترین

مقدار آن عدد ۷ و بیشترین مقدار عدد ۱۷ است. به طور معمول تعداد ویژگی ۱۲ بین مجموعه داده‌های مطرح‌شده پرتکرار است. در قسمت الگوریتم‌ها نام آخرین الگوریتم فرااکتشافی تاثیرگذار در آخرین تکرار برنامه ثبت شده است. میانگین مقدار ناحیه زیر منحنی برای هر مجموعه داده با توجه به تعداد ویژگی استفاده‌شده ثبت شده است.

در شکل (۱۸) ویژگی اول در تمامی مجموعه داده‌ها wmc نام دارد و به همین ترتیب تمامی این ویژگی‌ها دارای نام هستند. تعداد انتخاب تمامی ویژگی‌ها، بین ویژگی‌های مطرح‌شده در جدول (۱۳) شمارش شده است. این اعداد پس از شمارش در جدولی به صورت صعودی مرتب می‌شوند که عددی در بازه ۹ و ۱۸ است. به طور واضح‌تر، ویژگی ca با ۹ بار انتخاب و ویژگی ic با ۱۸ بار انتخاب بین تمامی مجموعه داده‌ها به ترتیب کم‌ارزش‌ترین و

جدول (۱۲) : نتایج الگوریتم جنگل تصادفی با انتخاب ویژگی

مجموعه داده	نتایج آزمون با انتخاب ویژگی						
	میانگین AUC	بیشترین AUC	کمترین AUC	میانه AUC	sensitivity	Specificity	Precisions
Ant-1.7	۰/۷۷۴۴	۰/۸۲۵	۰/۷۴۲۱	۰/۷۶۴۸	۰/۸۰۶۰	۰/۷۴۲۹	۰/۴۷۲۵
Camle-1.0	۰/۷۹۹۶	۰/۹۰۹۱	۰/۶۵۶۳	۰/۸۴۵۹	۰/۸۴۳۷	۰/۷۵۵۴	۰/۱۳۷۱
Camel-1.2	۰/۶۷۰۴	۰/۷۲۶۹	۰/۶۱۲۲	۰/۶۶۴۳	۰/۶۰۴۰	۰/۷۳۶۸	۰/۵۶۱۰
Camel-1.4	۰/۷۰۷۰	۰/۸۰۳۳	۰/۵۹۰۸	۰/۷۱۵۷	۰/۵۸۴۶	۰/۸۲۹۵	۰/۴۰۰۹
Camel-1.6	۰/۶۹۲۱	۰/۸۱۵۶	۰/۵۹۵۱	۰/۶۷۲۴	۰/۶۲۰۸	۰/۷۶۳۴	۰/۳۹۰۳
Jedit-3.2	۰/۸۰۵۸	۰/۹۳۴۸	۰/۶۳۸۹	۰/۸۲۴۷	۰/۷۹۱۹	۰/۸۱۹۸	۰/۶۸۰۲
Jedit-4.0	۰/۷۹۸۱	۰/۹۱۰۷	۰/۶۶۶۷	۰/۸۱۲۶	۰/۷۷۲۷	۰/۸۲۳۵	۰/۵۵۶۲
Jedit-4.1	۰/۸۳۰۶	۰/۹۲۵۹	۰/۶۹۷۷	۰/۸۴۶۸	۰/۷۹۱۷	۰/۸۶۹۴	۰/۶۶۴۷
Jedit-4.2	۰/۸۵۰۱	۰/۹۱۹۴	۰/۷۵۲۷	۰/۸۴۰۴	۰/۸۹۹۹	۰/۸۰۰۴	۰/۳۹۴۶
Jedit-4.3	۰/۷۹۴۹	۰/۹۸۹۶	۰/۴۷۹۶	۰/۸۰۱۴	۰/۶۲۵	۰/۹۶۴۷	۰/۴۴۷۲
Log4j-1.0	۰/۸۳۹۸	۰/۹۵۴۵	۰/۶۲۵	۰/۸۶۹۳	۰/۸۶۶۷	۰/۸۱۳۰	۰/۵۸۹۵
Log4j-1.1	۰/۸۳۹۲	۰/۹۲۸۶	۰/۵۳۳۳	۰/۸۸۱۹	۰/۸۶۲۳	۰/۸۱۵۰	۰/۷۰۵۵
Log4j-1.2	۰/۷۷۶۱	۰/۹۷۲۲	۰/۵۳۳۳	۰/۸۲۰۴	۰/۷۳۹۸	۰/۸۱۲۵	۰/۹۷۱۰
Lucene-2.0	۰/۷۴۳۱	۰/۸۸۱۰	۰/۵۳۳۳	۰/۷۳۵۷	۰/۷۱۲۷	۰/۷۷۳۶	۰/۷۳۶۴
Lucene-2.2	۰/۶۷۹۳	۰/۸۱۹۴	۰/۵۲۲۷	۰/۶۶۹۸	۰/۷۰۱۲	۰/۶۵۷۴	۰/۷۴۶۴
Lucene-2.4	۰/۷۶۹۵	۰/۹۱۱۸	۰/۷۰۱۸	۰/۷۶۳۲	۰/۷۷۸۷	۰/۷۶۰۳	۰/۸۳۰۹
Xalan-2.4	۰/۷۵۲۱	۰/۸۸۶۴	۰/۶۷۵۰	۰/۷۳۲۹	۰/۶۴۰۶	۰/۸۶۳۶	۰/۴۵۴۵
Xalan-2.5	۰/۷۴۰۸	۰/۸۷۵	۰/۶۸۸۶	۰/۷۲۰۶	۰/۷۴۹۰	۰/۷۳۲۶	۰/۷۲۰۲
Xalan-2.6	۰/۷۹۸۲	۰/۸۶۲۲	۰/۷۱۹۹	۰/۷۹۶۱	۰/۷۴۳۰	۰/۸۵۳۳	۰/۸۱۲۶
Xalan-2.7	۰/۹۰۲۴	۱/۰	۰/۷۴۴۴	۰/۹۹۴۴	۰/۹۹۵۲	۰/۸۰۹۵	۰/۹۹۵۲



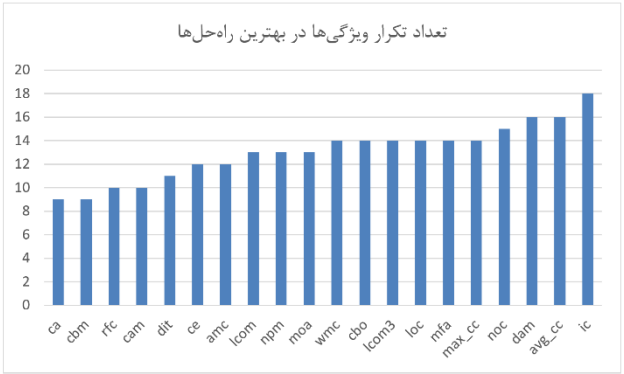
شکل (۱۷) : بازه میانگین مقدار ناحیه زیر منحنی هر مجموعه داده

پرارزش‌ترین ویژگی‌ها نامیده می‌شوند. می‌توان به این نتیجه رسید که با اعمال فرایند انتخاب ویژگی به عنوان یک روش کاهش ابعاد برای حذف ویژگی‌های زاید و افزونه با استفاده از الگوریتم‌های فرااکتشافی، بهبود مطلوبی در مقدار ناحیه زیر منحنی مطرح می‌شود.

در شکل (۱۹) به طور کلی مقدار تابع برازندگی برای مجموعه داده Ant-1.7 در طی ۱۰۰ بار تکرار آورده شده است. با توجه به تابع برازندگی مطرح‌شده، هر چه این مقدار کمتر باشد به دلیل افزایش مقدار ناحیه زیر منحنی بهتر است. با مشاهده این نمودارها می‌توان دریافت که با استفاده از روش انتخاب ویژگی پیشنهادی و سه الگوریتم فرااکتشافی مطرح‌شده، مقادیر

جدول (۱۳) : ویژگی‌های انتخاب‌شده برای هر مجموعه داده

مجموعه داده	ویژگی‌های انتخاب شده	تعداد	الگوریتم	AUC
Ant-1.7	[1. 1. 1. 1. 0. 1. 0. 0. 0. 1. 0. 0. 1. 1. 0. 1. 0. 0. 0. 1]	۱۰	BABC	۰/۷۷۴۴
Camle-1.0	[0. 1. 1. 0. 0. 1. 0. 0. 0. 1. 0. 0. 1. 1. 0. 1. 1. 0. 1. 1]	۱۰	TBWOA	۰/۷۹۹۶
Camel-1.2	[1. 1. 1. 1. 0. 0. 0. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 0. 0]	۱۴	BABC	۰/۶۷۰۴
Camel-1.4	[1. 1. 1. 1. 1. 0. 1. 1. 1. 0. 1. 1. 1. 1. 0. 0. 0. 1. 1. 0. 0]	۱۳	TBWOA	۰/۷۰۷۰
Camel-1.6	[0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1. 0. 1. 1. 1. 1. 0. 1. 0. 0. 0. 1]	۷	EBMFO	۰/۶۹۲۱
Jedit-3.2	[1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 1. 1. 1. 1. 1. 1. 0. 1. 0. 1]	۱۳	TBWOA	۰/۸۰۵۸
Jedit-4.0	[1. 0. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1]	۱۶	EBMFO	۰/۷۹۸۱
Jedit-4.1	[0. 0. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1]	۱۶	EBMFO	۰/۸۳۰۶
Jedit-4.2	[1. 1. 0. 1. 1. 1. 1. 0. 0. 1. 0. 1. 1. 1. 0. 0. 1. 1. 1. 0. 0. 1]	۱۲	TBWOA	۰/۸۵۰۱
Jedit-4.3	[1. 0. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 0. 1. 1]	۱۵	EBMFO	۰/۷۹۴۹
Log4j-1.0	[1. 1. 1. 1. 0. 0. 0. 0. 1. 1. 0. 0. 0. 1. 1. 1. 1. 1. 0. 1. 1. 0]	۱۲	TBWOA	۰/۸۳۹۸
Log4j-1.1	[1. 0. 0. 1. 0. 0. 0. 0. 1. 1. 1. 1. 1. 1. 0. 0. 0. 1. 1. 1. 1. 1]	۱۲	EBMFO	۰/۸۳۹۲
Log4j-1.2	[0. 0. 1. 1. 1. 1. 0. 0. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 0. 1. 1. 1]	۱۴	EBMFO	۰/۷۷۶۱
Lucene-2.0	[0. 1. 1. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1]	۱۶	TBWOA	۰/۷۴۳۱
Lucene-2.2	[0. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 1. 0. 1. 0. 1. 1. 1]	۱۴	BABC	۰/۶۷۹۳
Lucene-2.4	[1. 0. 1. 1. 0. 1. 1. 1. 1. 0. 0. 1. 0. 1. 1. 1. 0. 1. 0. 1. 1. 0]	۱۲	BABC	۰/۷۶۹۵
Xalan-2.4	[1. 1. 0. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 0. 1. 0. 1. 1. 1. 1]	۱۵	BABC	۰/۷۵۲۱
Xalan-2.5	[1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 1]	۱۷	TBWOA	۰/۷۴۰۸
Xalan-2.6	[1. 0. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1]	۱۲	TBWOA	۰/۷۹۸۲
Xalan-2.7	[1. 0. 0. 0. 0. 0. 1. 1. 1. 0. 1. 1. 0. 0. 0. 1. 1. 1. 0. 0. 1. 1]	۱۱	BABC	۰/۹۰۲۴



شکل (۱۸) : تعداد تکرار هر ویژگی در بهترین راه‌حل‌ها

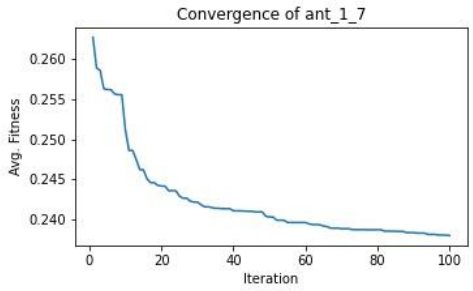
برازندگی در طی ۱۰۰ تکرار به مرور کاهش پیدا کرده و بهبود لازم حاصل می‌شود. محور افقی نشان‌دهنده تعداد تکرارهای انجام‌شده در اجرای برنامه است. محور عمودی نشان‌دهنده محدوده مقدار تابع برازندگی است. برای مثال در مجموعه داده Ant-1.7 با محاسبه مقدار برازندگی راه‌حل‌های تصادفی که تقریبا ۰/۲۷ است و با طی تقریبا ۱۰ تکرار توسط سه الگوریتم فرااکتشافی به صورت چرخشی به مقدار ۰/۲۵ کاهش می‌یابد. همین طور که ادامه داده می‌شود، به مرور زمان به ۱۰۰ تکرار که شرط پایان برنامه است می‌رسد، که توانسته است مقدار تابع برازندگی را تقریبا به ۰/۲۳ برساند و نسبت به مقدار اولیه بهبود خوبی حاصل شده است.

در جدول (۱۴) نتایج روش انتخاب ویژگی پیشنهادی به همراه الگوریتم طبقه‌بندی جنگل تصادفی با پارامترهای بهبودیافته با میانگین مقدار ۰/۷۷۸۲ برای تمامی ۲۰ مجموعه داده ثبت شده است. مقاله بعدی برای سال ۲۰۲۱ است و از الگوریتم طبقه‌بندی درخت تصمیم و ۱۷ مجموعه داده پرومیس استفاده کرده است. این روش، مقدار میانگین ۰/۸۰۱۴ را ثبت کرده است. مقاله بعدی برای سال ۲۰۲۰ است که از دو نوع تابع انتقال برای دودویی کردن الگوریتم شعله پروانه توسعه‌یافته استفاده می‌کند. این مقاله برای ارزیابی روش خود از ۱۵ مجموعه داده پرومیس استفاده کرده است و دارای مقدار میانگین ۰/۷۳۷۸ و ۰/۷۴۹۴ است.

مقاله بعدی برای سال ۲۰۱۸ است که برای ارزیابی روش خود از ۱۸ مجموعه داده پرومیس استفاده کرده و دارای مقدار میانگین ۰/۵۱۳۵ است. مقاله بعدی برای سال ۲۰۱۷ است که روش خود را با دو نوع الگوریتم طبقه‌بندی بیز ساده و C4.5 مورد آزمایش قرار داده است. از ۷ مجموعه داده پرومیس برای ارزیابی روش خود استفاده نموده و به ترتیب دارای مقدار میانگین ۰/۶۸۷۱ و ۰/۶۳۵۷ است. برای مثال در مجموعه داده Ant-1.7 روش پیشنهادی مطرح‌شده از مقاله‌های سال ۲۰۲۱، ۲۰۲۰ و ۲۰۱۸ بهتر عمل کرده ولی در کل، مقاله ۲۰۱۷ دارای بهترین نتیجه است. به عنوان مثالی دیگر، در مجموعه داده Camel-1.0 نتایج روش پیشنهادی مطرح‌شده از تمامی نتایج مقالات بهتر است. روش پیشنهادی مطرح‌شده به طور کلی توانست تمامی ۱۵ نتیجه مقاله ۲۰۲۰ به همراه دو تابع انتقال را بهبود بخشد. همچنین روش پیشنهادی توانست تمامی ۱۸ نتیجه مقاله ۲۰۱۸ را بهبود دهد. در مقاله سال ۲۰۱۷ نیز روش پیشنهادی تمامی مجموعه داده‌ها غیر از Ant-1.7را بهبود داده است. در مقاله سال ۲۰۲۱ نیز روش پیشنهادی در ۱۷ مجموعه داده توانست نتیجه ۷ مجموعه داده را بهبود دهد.

۷- نتیجه‌گیری

روش‌های مبتنی بر فیلتر دارای پیچیدگی زیادی هستند و از روش‌های آماری برای ارزیابی راه‌حل پیشنهادی استفاده می‌کنند و هیچ‌گونه وابستگی به روش‌های یادگیری ماشین ندارند. بنابراین از سرعت اجرایی بالایی برخوردار هستند. روش‌های مبتنی بر پوشش به دلیل استفاده از الگوریتم‌های یادگیری ماشین دارای پیچیدگی کمتری هستند ولی سرعت اجرای کُندی دارند. اگر تعداد ویژگی‌های یک مجموعه داده زیاد نباشد می‌توان برای کسب کارایی بهتر به دلیل استفاده از الگوریتم‌های یادگیری ماشین، به عنوان الگوریتم‌های طبقه‌بندی، از روش‌های مبتنی بر پوشش استفاده کرد ولی اگر تعداد ویژگی‌ها



شکل (۱۹) : تغییرات مقدار تابع برازندگی در طی اجرای برنامه

جدول (۱۴) : مقایسه نتایج تحقیق با مقالات مشابه اخیر

	نتایج روش پیشنهادی	Hassouneh [۲۲] , ۲۰۲۱	Tumar [۱۴] , ۲۰۲۰		Turabiieh [۲۷] , ۲۰۱۸	Shatnawi [۴۴] , ۲۰۱۷	
مجموعه داده	RF	DT	TF=S2	TF=V3	L-RNN	NB	C4.5
Ant-1.7	۰.۷۷۴۴	۰.۷۳۷	۰.۷۵۵۲	۰.۷۶۱۵	۰.۵۲۳۴	**۰.۷۹۰**	۰.۷۴۰
Camle-1.0	۰.۷۹۹۶	---	۰.۷۴۸۹	۰.۷۵۵۲	۰.۶۰۷۴	---	---
Camel-1.2	۰.۶۷۰۴	**۰.۷۰۳**	۰.۶۲۱۷	۰.۶۲۱۵	۰.۴۴۳۲	۰.۵۶۰	۰.۵۲۰
Camel-1.4	**۰.۷۰۷۰**	۰.۶۸۶	۰.۶۸۴۵	۰.۶۹۱۸	۰.۵۲۱۴	۰.۶۷۰	۰.۶۰۰
Camel-1.6	**۰.۶۹۲۱**	۰.۶۷۷	۰.۶۷۲۷	۰.۶۵۸۰	۰.۵۰۴۵	۰.۵۹۰	۰.۵۴۰
Jedit-3.2	**۰.۸۰۵۸**	۰.۷۹۰	۰.۸۰۲۸	۰.۸۰۵۳	---	---	---
Jedit-4.0	۰.۷۹۸۱	**۰.۸۲۳**	۰.۷۲۵۳	۰.۷۱۶۱	۰.۵۴۹۴	۰.۷۰۰	۰.۷۲۰
Jedit-4.1	**۰.۸۳۰۶**	۰.۷۲۴	۰.۷۸۱۴	۰.۷۸۶۸	---	۰.۷۵۰	۰.۶۹۰
Jedit-4.2	**۰.۸۵۰۱**	۰.۸۴۷	۰.۷۹۲۷	۰.۷۹۷۳	۰.۵۱۸۵	۰.۷۵۰	۰.۶۴۰
Jedit-4.3	**۰.۷۹۴۹**	---	۰.۷۰۲۸	۰.۷۴۹۹	۰.۶۸۳۸	---	---
Log4j-1.0	۰.۸۳۹۸	**۰.۹۰۴**	۰.۷۶۸۲	۰.۷۹۳۷	۰.۵۳۵۴	---	---
Log4j-1.1	۰.۸۳۹۲	**۰.۹۸۲**	۰.۷۶۵۹	۰.۷۹۸۶	۰.۵۲۲۲	---	---
Log4j-1.2	۰.۷۷۶۱	۱.۰۰۰	۰.۶۹۱۰	۰.۷۱۰۲	۰.۳۶۱۴	---	---
Lucene-2.0	۰.۷۴۳۱	**۰.۸۱۸**	---	---	۰.۵۵۳۱	---	---
Lucene-2.2	۰.۶۷۹۳	**۰.۷۶۵**	---	---	۰.۴۶۳۰	---	---
Lucene-2.4	۰.۷۶۹۵	**۰.۸۱۱**	---	---	۰.۴۸۵۱	---	---
Xalan-2.4	۰.۷۵۲۱	**۰.۷۷۳**	۰.۷۲۶۶	۰.۷۴۹۵	۰.۵۳۶۴	---	---
Xalan-2.5	۰.۷۴۰۸	**۰.۷۸۸**	---	---	۰.۵۰۹۸	---	---
Xalan-2.6	**۰.۷۹۸۲**	۰.۷۹۶	---	---	۰.۵۰۶۴	---	---
Xalan-2.7	**۰.۹۰۲۴**	---	۰.۸۲۶۳	۰.۸۴۵۶	۰.۴۲۰۱	---	---
میانگین	۰.۷۷۸۲	۰.۸۰۱۴	۰.۷۳۷۸	۰.۷۴۹۴	۰.۵۱۳۵	۰.۶۸۷۱	۰.۶۳۵۷

زیاد باشد، روش‌های مبتنی بر پوشش زمان اجرای بسیار زیادی می‌گیرند و به جای آن از روش‌های مبتنی بر فیلتر می‌توان استفاده کرد.

این مقاله یک مرور و بررسی اجمالی در مورد ۱۸ الگوریتم فرااکتشافی و دودویی‌سازی آن‌ها برای انتخاب ویژگی مبتنی بر پوشش ارایه کرده است. علاوه‌بر بررسی فنون حل مسائل مرتبط با انتخاب ویژگی، این الگوریتم‌ها در ۶ دسته مختلف در بین سال‌های ۱۹۸۸ تا ۲۰۱۹ مورد ارزیابی قرار گرفته‌اند. در جدول شماره ۸، نقاط قوت و ضعف هر کدام از این ۱۸ الگوریتم مطرح‌شده و مراحل اکتشاف و بهره‌برداری آن‌ها به‌طور خلاصه بیان شده است. در برخی الگوریتم‌ها کاربرد آن‌ها، تعداد پارامترها، درک و فهم ساده و موارد دیگر نیز مورد بررسی قرار گرفته است. نتایج حاصل گویای این حقیقت است که

الگوریتم‌های سنجاقک، تغذیه مصنوعی پرندگان، کلونی پنگوئن‌های امپراتور، فیل، بادبان ماهی و جستجوی صف، نسبت به دیگر روش‌ها جدیدتر هستند و مقالات کمتری در مورد آن‌ها وجود دارد، لذا نمی‌توان نقاط ضعف خاصی برای آنها مطرح کرد و البته جای کار بیشتری برای آینده دارند.

برای انتخاب هر کدام از این نوع الگوریتم‌ها می‌توان معیارهای مختلفی را مطرح کرد. برای مثال، اگر به دنبال استفاده از الگوریتم‌هایی هستید که به سادگی پیاده‌سازی شوند، می‌توانید از الگوریتم‌های کرم شب‌تاب، خفاش، بهینه‌سازی شیر مورچه، سنجاقک، شعله پروانه، تغذیه مصنوعی پرندگان، وال، جستجوی صف و ژنتیک استفاده کنید. اگر به دنبال استفاده از روش‌هایی پر استفاده و با عملکرد بالا هستید، می‌توانید از الگوریتم‌های کلونی مورچگان، کلونی زنبور، ازدحام ذرات، گرگ خاکستری، وال، علف‌های هرز تهاجمی و ژنتیک استفاده کنید. اگر به دنبال رفع نقص‌های این نوع الگوریتم‌ها هستید می‌توانید از نسخه بهبودیافته آنها استفاده کنید و یا خودتان آنها را بهبود بخشید. حتی می‌توانید با ترکیب ۲ یا ۳ نوع از این الگوریتم‌ها و استفاده در طی تکرارهای مختلف در یک برنامه، به بهبود آنها کمک کنید.

در ادامه اثربخشی ترکیب سه الگوریتم فرااکتشافی شعله پروانه توسعه‌یافته، کلونی زنبور و وال توسعه‌یافته بر روی ۲۰ مجموعه داده پیش‌بینی خطای نرم‌افزار پرومیس به همراه الگوریتم یادگیری ماشین جنگل تصادفی بهبودیافته، فن اعتبارسنجی متقابل ۱۰ قسمت و فن نمونه‌برداری مورد آزمایش قرار گرفته است. نتیجه آزمایش‌ها گویای آن است که استفاده و ترکیب این نوع الگوریتم‌ها می‌تواند سبب بهبود مقدار ناحیه زیر منحنی نسبت به روش بدون انتخاب ویژگی شود. استفاده از الگوریتم‌های توسعه‌یافته در مدل‌های ترکیبی در مقایسه با نسخه ساده، به دلیل کشف سریع‌تر راه‌حل‌های بهینه باعث کاهش هزینه‌های محاسباتی می‌شود.

با هر بار آزمایش مقادیر مختلف جمعیت، نتایج معیارهای ارزیابی همانند ناحیه زیر منحنی تغییر می‌کنند. پارامترهای جنگل تصادفی به نام‌های تعداد درخت تخمین و حداکثر ویژگی و حداکثر عمق با مقادیر مختلف مورد آزمایش قرار گرفت، که به ترتیب، مقادیر ۲۰۰، log2 و ۷ در اکثر موارد دارای مقدار ناحیه زیر منحنی بالاتری هستند و به دلیل اینکه در اکثر مقالات حوزه انتخاب ویژگی بیش‌ترین استفاده را داشته‌اند، بنابراین آزمایش‌ها در جنگل تصادفی با همین مقادیر انجام‌گرفته و توانسته است مقدار بهبود بیشتری به همراه داشته باشد. همچنین چهار معیار ارزیابی عمکرد مدل یادگیری ماشین به نام‌های ناحیه زیر منحنی، دقت، حساسیت و اختصاصی بودن مورد ارزیابی قرار گرفت و می‌توان به این نتیجه رسید که با استفاد از فن اعتبارسنجی متقابل ۱۰ قسمت و میانگین‌گیری به نتیجه منطقی نسبت به دیگر آزمایش‌های بدون استفاده از این فن دست یافت.

همچنین تعداد ویژگی‌های استفاده‌شده در هر مجموعه داده ثبت شده که کمترین مقدار آن عدد ۷ و بیشترین مقدار عدد ۱۷ می‌باشد. به طور معمول تعداد ۱۲ ویژگی بین مجموعه داده‌های مطرح‌شده پرتکرار است. ویژگی ca با ۹ بار انتخاب و ویژگی ic با ۱۸ بار انتخاب بین تمامی مجموعه داده‌ها به ترتیب کم‌ارزش‌ترین و پرارزش‌ترین ویژگی نامیده می‌شوند. مقاله وال توسعه‌یافته در برخی مجموعه داده‌ها دارای نتیجه بالایی است و این در حالی است که از اعتبارسنجی متقابل استفاده نکرده و نتایج حاصله از نظر فنی منطقی نیستد و اگر از فن اعتبارسنجی متقابل استفاده می‌شد، قطعا نتیجه ضعیف‌تری بدست می‌آورد. در این صورت تعداد بهبود روش پیشنهادی این تحقیق نیز بیشتر می‌شد.

برای کار آینده، می‌توان از معیارهای ارزیابی مطرح دیگر مانند f-score جهت ارزیابی عملکرد روش پیشنهادی بهره برد. از روش پیشنهادی مطرح‌شده می‌توان در حوزه‌های دیگر از قبیل پزشکی، به‌طور مثال در جهت افزایش دقت تشخیص سرطان استفاده کرد. می‌توان تاثیر عملکرد روش‌های مختلف طبقه‌بندی ازقبیل، یادگیری گروهی، شبکه عصبی، یادگیری عمیق و روش‌های مختلف ماشین بردار پشتیبان را به همراه روش انتخاب ویژگی پیشنهادی مورد بررسی قرار داد. می‌توان از الگوریتم‌های فرااکتشافی بهبودیافته جدید برای جایگزین کردن در این مدل ترکیبی به منظور افزایش مقدار ناحیه زیر منحنی بهره برد تا یک مدل کامپیوتری ساخته شود که قادر به پیش‌بینی دقیق‌تر کلاس هدف (مستعد خطا بودن ماژول) براساس ویژگی‌های انتخاب‌شده به همراه مقدار ناحیه زیر منحنی باشد.

مراجع

[1] Sharma, M., Kaur, P., "A Comprehensive Analysis of Nature-Inspired Meta-Heuristic Techniques for Feature Selection Problem", Springer, Archives of Computational Methods in Engineering, CIMNE, Barcelona, 2020.

[2] Brezocnik, L., Fister, I., Jr and Podgorelec, V., "Swarm Intelligence Algorithms for Feature Selection: A Review", mdpi, Journals Applied Sciences, vol. 8, pp. 1-31, 2018.

[3] Balogun, A., O., Basri, S., Abdulkadir, S., J. and Hashim, A., S., "Performance Analysis of Feature Selection Methods in Software Defect Prediction: A Search Method Approach", MDPI, Applied Sciences Journals, vol. 9, pp.1-20, doi:10.3390/app9132764, July 2019.

[4] Kumar B, K., Gyani, J., N. G, "Software Defect Prediction using Ant Colony Optimization", International Journal of Applied Engineering Research, vol. 13, Number 19, pp.14291-14297, 2018.

[5] Kanan, H., R., Faez, K. and Taheri, S., M., "Feature Selection Using Ant Colony Optimization (ACO): A New Method and Comparative Study in the Application of Face Recognition System", Advances in Data Mining. Theoretical Aspects and Applications, Berlin, Heidelberg, pp. 63–76, 2007.

[6] Bennis, F., Bhattacharjya, R., K., *Nature-Inspired Methods for Metaheuristics Optimization*, Springer Nature Switzerland, vol.16, 2020.

[7] Rajan, C., Geetha, K., Rasi Priya, C., Sasikala, R., "Investigation on Bio-Inspired Population Based Metaheuristic Algorithms for Optimization Problems in Ad Hoc Networks", International Journal of Mathematical, Computational, Natural and Physical Engineering, vol.9, no:3, pp.111-118, 2015.

[8] Anbu, M., Anandha Mala, G. S., "Feature selection using firefly algorithm in software defect prediction", Springer, Cluster Computing, vol. 22, pp. 10925–10934, 2017.

[9] Yang, XS., "A New Metaheuristic Bat-Inspired Algorithm", Springer, Nature Inspired Cooperative Strategies for Optimization, vol. 284, pp. 65–74, 2010.

[10] Ramasamy, R., Rani, S., "Modified Binary Bat Algorithm for Feature Selection in Unsupervised Learning", The International Arab Journal of Information Technology, vol.15, no.6, pp.1060-1067, November 2018.

[27] Turabieh, H., Mafarja, M., Li, X., "Iterated feature selection algorithms with layered recurrent neural network for software fault prediction", Elsevier, Expert Systems With Applications, vol.122, pp.27-42, 2019.
[28] Zhang, J., Xiao, M., Gao, L., Pan, Q., "Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems", Elsevier, Applied Mathematical Modelling, vol.63, pp.464-490, 2018.
[29] Mishan, M.T., Fadzil, A.F.A., Samah, K.A.F.A., Baharin, N.F., Anuar, N., "Business Intelligence for Paintball Tournament Matchmaking Using Particle Swarm Optimization", ijeecs, Indonesian Journal of Electrical Engineering and Computer Science, vol.11, pp.599-606, August 2018.
[30] Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., Cosar, A., "A survey on new generation metaheuristic algorithms", Elsevier, Computers & Industrial Engineering, vol.137, pp.1-29, November 2019.
[31] Cuevas, E., Fausto, F. and González, A., "An Introduction to Nature-Inspired Metaheuristics and Swarm Methods", Springer, New Advancements in Swarm Algorithms: Operators and Applications, vol.160, pp.1-41, 2019.
[32] Li, S., Gong, W., Gu, Q., "A comprehensive survey on meta-heuristic algorithms for parameter extraction of photovoltaic models", Elsevier, Renewable and Sustainable Energy Reviews, vol.141, pp.1-22, 2021.
[33] Tan, W.S., Hassan, M.Y., Majid, M.S., Rahman, H.A., "Optimal distributed renewable generation planning: A review of different approaches", Elsevier, Renewable and Sustainable Energy Reviews, vol.18, pp.626-645, 2013.
[34] Yang, L., Yang, X., Wu, Y., Liu, X., "Applied Research on Distributed Generation Optimal Allocation Based on Improved Estimation of Distribution Algorithm", MDPI, energies, vol.11, pp.1-17, 2018.
[35] De, D., Mukherjee, A., Das, S.K., Dey, N., *Nature Inspired Computing for Wireless Sensor Networks*, Springer Tracts in Nature-Inspired Computing, 2020.
[36] Abualigah, L., Shehab, M., Alshinwan, M., Mirjalili, S., Elaziz, M.A., "Ant Lion Optimizer: A Comprehensive Survey of Its Variants and Applications", Springer, Archives of Computational Methods in Engineering, vol.28, pp.1397-1416, April 2020.
[37] Rahman, C.M., Rashid, T.A., "Dragonfly Algorithm and Its Applications in Applied Science Survey", Hindawi, Computational Intelligence and Neuroscience, vol.2019, pp.1-21, 2019.
[38] Shehab, M., Abualigah, L., Hamad, H.A, Alabool, H., Alshinwan, M., Khasawneh, A.M., "Moth–flame optimization algorithm: variants and applications", Springer, Neural Computing and Applications, vol.32, pp.9859-9884, 2019.
[39] Dhiman. G., Kumar, V., "Emperor penguin optimizer: A bio-inspired algorithm for engineering problems", Elsevier, Knowledge-Based Systems, vol. 159, pp. 20-50, November 2018.
[40] Agrawal, P., Abutarboush, H.F., Ganesh, T., Mohamed, Q.W., "Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009-2019)", IEEE Access, vol.9, pp.26766-26791, 2021.
[41] Alyasseri, Z.A.A., Khader, A.T., Al-Betar, M.A., Awadallah, M.A., Yang, X.S., "Variants of the Flower Pollination Algorithm: A Review", Springer, Studies in Computational Intelligence, vol.744, pp.91-118, 2018.
[11] Mirjalili, S., "The Ant Lion Optimizer", Elsevier, Advances in Engineering Software, vol.83, pp.80–98, May, 2015.
[12] Mafarja, M. S., Mirjalili, S., "Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection", Springer, Soft Computing, vol.23, pp.6249-6265, June 2018.
[13] Mirjalili, S., "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems", springer, Neural Computing and Applications, vol.27, pp.1053-1073, 2016.
[14] Tumar, I., Hassouneh, Y., Turabieh, H. and Thaher, T., "Enhanced Binary Moth Flame Optimization as a Feature Selection Algorithm to Predict Software Fault Prediction", IEEE Access, vol.8, pp.8041-8055, 2020.
[15] Kennedy, J. and Eberhart, R., "Particle Swarm Optimization", IEEE, International Conference on Neural Networks, Perth, WA, Australia, 1995.
[16] Chuang, L.Y., Chang, H.W., Tu., C.J., Yang, C. H., "Improved binary PSO for feature selection using gene expression data", Elsevier, Computational Biology and Chemistry, vol.32, pp.29-38, 2008.
[17] Lamy, J.B., "Artificial Feeding Birds (AFB): a new metaheuristic inspired by the behavior of pigeons", springer, Advances in nature-inspired computing and applications, pp. 43–60, 2019.
[18] Harif, S., Mohammadzadeh, J., Khalilian, M., Ebrahimnejad, S., "Hybrid-EPC: an Emperor Penguins Colony algorithm with crossover and mutation operators and its application in community detection", Springer, Progress in Artifcial Intelligence, vol.10, pp.181-193, 2021.
[19] Mirjalili, S., Mirjalili, S. M., Lewis, A., "Grey Wolf Optimizer", Elsevier, Advances in Engineering Software, vol.69, pp.46-61, 2014.
[20] Emary, E., Zawbaa, H.M., Hassanien, A.E., "Binary Gray Wolf Optimization Approaches for Feature Selection", Elsevier, Neurocomputing, vol.172, pp.371-381, 2016.
[21] Deb, S., Fong, S., Tian, Z., "Elephant Search Algorithm for Optimization Problems", IEEE, The Tenth International Conference on Digital Information Management, Jeju, Korea (South), 2015.
[22] Hassouneh, Y., Turabieh, H., Thaher, T., Tumar, I., Chantar, H. and Too, J., "Boosted Whale Optimization Algorithm With Natural Selection Operators for Software Fault Prediction", IEEE, Journals & Magazines, vol.9, pp.14239-14258, 2021.
[23] Shadravan, S., Naji, H.R., Bardsiri, V.K., "The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems", Elsevier, Engineering Applications of Artificial Intelligence, vol.80, pp.20-34, April 2019.
[24] Karimkashi, S., Kishk, A.A, "Invasive Weed Optimization and its Features in Electromagnetics", IEEE, Transactions on antennas and propagation, vol.58, no.4, pp.1269-1278, April 2010.
[25] Nasser, A.B., Zamli, K.Z., Alsewari, A.R.A., Ahmed, B.S., "Hybrid flower pollination algorithm strategies for t-way test suite generation", PLoS, vol.13, pp.1-24, 2018.
[26] Nabil, E., "A Modified Flower Pollination Algorithm for Global Optimization", Elsevier, Expert Systems With Applications, vol. 57, pp. 192-203, 2016.

[42] Boussaïd, I., Lepagnot, J., Siarry, P., "A survey on optimization metaheuristics", Elsevier, Infor mation Sciences, vol. 237, pp. 82-117, 2013.
[43] A. Kumar, A. Bansal, “Software Fault Proneness Prediction Using Genetic Based Machine Learning Techniques”, IEEE, 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 2019.
[44] R. Shatnawi, “The application of ROC analysis in threshold identification, data imbalance and metrics selection for software fault prediction”, Springer Nature Switzerland, Advances in Computing and Data Sciences, CCIS, vol.1440, pp.443-451, October 2021.

زیرنویس‌ها

[1] Filter method
[2] Wrapper method
[3] Embedded method
[4] Bioinformatics
[5] Swarm Intelligence
[6] Ant Colony Optimization (ACO)
[7] Pheromone
[8] Graph
[9] Artifcial Bee Colony (ABC)
[10] Firefly Algorithm (FA)
[11] Bats
[12] Bit
[13] Ant Lion Optimizer (ALO)
[14] Dragonfy algorithm
[15] Separation
[16] Alignment
[17] Cohesion
[18] Moth Flame Optimization (MFO)
[19] Matrix
[20] Uniform crossover
[21] Particle Swarm Optimization (PSO)
[22] Artifcial Feeding Birds (AFB)
[23] Emperor Penguins Colony (EPC)
[24] Grey Wolf Optimizer (GWO)
[25] Alpha
[26] Democratic
[27] Elephants
[28] Euclidean distance
[29] Whale
[30] Sailfish
[31] Invasive Weed Optimization (IWO)
[32] Biology
[33] Flower pollination algorithm (FOA)
[34] Levy
[35] Genetic Algorithm (GA)
[36] Queuing search algorithm (QSA)