
بسم الله الرحمن الرحيم

دانشکده مهندسی کامپیوتر

پایان‌نامه کارشناسی ارشد مهندسی نرم افزار

یادگیری تابع انتقال در الگوریتم فراابتکاری باینری برای انتخاب ویژگی در مسائل دسته‌بندی

نگارنده: زهرا نصیری

اساتید راهنما

دکتر محسن بیگلری

دکتر حسام عمران پور

اسفند ۱۴۰۰

در این صفحه صورت جلسه دفاع را قرار دهید. لازم است پس از صحافی این صفحه مجدداً توسط دانشکده مهر گردد و استاد راهنما با امضای خود اصلاحات پایان‌نامه را تایید کند.

تقدیم اثر

تقدیم به

پدر و مادر عزیز و مهربانم

که در سختی‌ها و دشواری‌های زندگی همواره یاوری دلسوز و فداکار و پشتیبانی محکم و مطمئن برایم بوده‌اند.

تشکر و قدردانی

اکنون که به یاری پروردگار و یاری و راهنمایی اساتید بزرگ موفق به پایان این رساله شده‌ام وظیفه خود دانشته که نهایت سپاسگزاری را از تمامی عزیزانی که در این راه به من کمک کرده‌اند را به عمل آورم:

از استاد گرامی، جناب آقای دکتر محسن بیگلری که با ارائه راهنمایی‌ها، نظرات ارزنده و نکات مفید، اینجانب را در تدوین این پایان‌نامه کارشناسی ارشد راهنمایی نمودند و پشتیبان من در راهبرد این موضوع بودند. سپاسگزارم.

از استاد گرامی، جناب آقای دکتر حسام عمران پور برای تمام حمایت‌ها و زحمات بی‌دریغشان سپاسگزاری می‌کنم.

همچنین از پدر و مادر عزیز و مهربانم به خاطر زحماتی که در طول زندگی همواره برای پیروزی و شادکامی من به جان خریدند، تشکر می‌کنم.

تعهدنامه

اینجانب **زهرا نصیری** دانشجوی دوره کارشناسی ارشد رشته **مهندسی کامپیوتر گرایش نرم‌افزار** دانشکده مهندسی کامپیوتر دانشگاه صنعتی شاهرود نویسنده پایان‌نامه **یادگیری تابع انتقال در الگوریتم فراابتکاری باینری برای انتخاب ویژگی در مسائل دسته‌بندی** تحت راهنمائی **دکتر محسن بیگلری و دکتر حسام عمران پور** متعهد می‌شوم.

- تحقیقات در این پایان‌نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است .
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است .
- مطالب مندرج در پایان‌نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است .
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می‌باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید .
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان‌نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در کلیه مراحل انجام این پایان‌نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است .
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

۱۴۰۰/۱۱/۱۲

تـاریـخ **امضای دانـشجو**

مالکیت نتایج و حق نشر

چکیده

یکی از مسائل چالشی در شناسایی الگو، فرآیند انتخاب ویژگی داده است. انتخاب ویژگی نقش مهمی در حل مسائل با داده‌های با ابعاد بالا ایفا می‌کند و یک گام مهم و اساسی در پیش‌پردازش بسیاری از مسائل طبقه‌بندی و یادگیری ماشین است.

روش پیشنهادی این پایان‌نامه یک روش انتخاب ویژگی، مبتنی بر الگوریتم گرگ خاکستری است. در این الگوریتم جهت به کاربردن انتخاب ویژگی از روش دسته‌بند، استفاده شده است. تابع انتقال نیز بخش مهمی از الگوریتم گرگ خاکستری باینری است که برای نگاشت مقدار پیوسته به مقدار باینری ضروری است.

در تمامی تحقیقات قبلی، تنها از یک تابع انتقال برای کل الگوریتم استفاده شده است و همه گرگ‌ها در کل الگوریتم با این تابع انتقال سروکار داشتند. در این پایان‌نامه، الگوریتم با چندین تابع انتقال سروکار دارد و هر گرگ تابع انتقال خاص خود را خواهد داشت. این ایده از آنجا نشات می‌گیرد که الگوریتم‌ها فراابتکاری تکاملی هستند و می‌توانند خود را بهینه کنند، گرگ‌ها می‌تواند در کل الگوریتم در هر مرحله نقش داشته باشد و در عین حال خودشان را بهینه کرده و خود را با جامعه تطبیق دهند و فقط به یک تابع انتقال وابسته نباشند.

در روش پیشنهادی، هشت تابع انتقال استفاده می‌شود که به دو خانواده S شکل و V شکل تقسیم می‌شوند. دو رویکرد برای یادگیری تابع انتقال پیشنهاد می‌گردد.

در روش پیشنهادی، به هر گرگ یک تابع مخصوص به خود نسبت داده شده است. این ایده می‌تواند امکانی فراهم آورد که در هر تکرار الگوریتم با توجه به انتخاب گرگ آلفا و حرکت به سمت طعمه، تابع انتقال مناسب نیز انتخاب شود. ما در این نوآوری از دو ایده استفاده کردیم. در ایده اول سه و یا دو بیت باینری را به جمعیت اولیه اضافه می‌کنیم. اگر دو بیت اضافه گردد چهار حالت تابع انتقال و اگر سه بیت اضافه گردد هشت تابع انتقال قابل دستیابی می‌باشد. در ایده دوم از ۱۰ یا ۲۱ بیت باینری به جمعیت اولیه اضافه می‌شود. اگر ۱۰ بیت اضافه گردد یک نوع تابع انتقال خواهیم داشت . 2^{10} حالت ضریب برای شیب تابع انتقال قابل دسترس می‌باشد. اگر ۲۱ بیت اضافه گردد دو نوع تابع انتقال قابل دستیابی می‌باشد که 2^{10} حالت ضریب برای شیب تابع انتقال وجود خواهد داشت. در هر دو ایده، از بیت‌های اضافه شده ملاکی برای انتخاب تابع انتقال و نیز ضریب موثر بر شیب این توابع استفاده شده است. بعد از هر تکرار الگوریتم موقعیت گرگ آلفا بروزرسانی و تابع انتقال انتخاب می‌گردد. با تکرارهای بعدی، الگوریتم ضمن انتخاب تابع انتقال مناسب با شیب مناسب، یادگیری تابع انتقال را نیز انجام می‌دهد.

نتایج آزمایش بر روی ده مجموعه داده UCI، نشان می‌دهد که انتخاب زیرمجموعه ویژگی بدست آمده با دقت طبقه‌بندی بالا، موثر و کارآمد می‌باشد.

واژه‌های کلیدی: انتخاب ویژگی، داده‌های با بعد بالا، بهینه‌سازی گرگ خاکستری، توابع انتقال، باینری

لیست مقالات مستخرج از پایان نامه

عنوان	کنفرانس/ ژورنال	وضعیت
Learning the Transfer Function in Binary Metaheuristic Algorithm for Feature Selection in Classification Problems	Springer Journals, Neural Computing and Applications (NCAA)	Under review

فهرست مطالب

فهرست اشکال

فهرست جداول

فصل۱: کلیات تحقیق

۱-۱ مقدمه

توسعه فناوری اطلاعات تعداد زیادی پایگاه داده و داده‌های عظیمی را در زمینه‌های مختلف تولید کرده است. تحقیقات در پایگاه‌های اطلاعاتی و فناوری اطلاعات باعث ایجاد رویکردی برای ذخیره و دستکاری این داده‌های ارزشمند برای تصمیم‌گیری بیشتر شده است. داده‌کاوی فرآیند استخراج اطلاعات و الگوهای مفید از داده‌های عظیم است. همچنین به عنوان فرآیند کشف دانش، استخراج دانش از داده‌ها، استخراج دانش یا تجزیه و تحلیل داده / الگو نامیده می‌شود.

داده‌کاوی یک فرآیند منطقی است که برای جستجو در حجم زیادی از داده‌ها به منظور یافتن داده‌های مفید استفاده می‌شود. هدف این تکنیک یافتن الگوهایی است که قبلا ناشناخته بودند. هنگامی که این الگوها یافت می‌شوند، می‌توان از آن‌ها برای اتخاذ تصمیمات خاصی برای توسعه تجارت خود استفاده کرد.

سه مرحله درگیر در داده‌کاوی عبارتند از:

- اکتشاف
- شناسایی الگو
- استقرار

اکتشاف: در مرحله‌ی اول اکتشاف داده‌ها، داده‌ها پاک شده و به شکل دیگری تبدیل می‌شوند و متغیرهای مهم و سپس ماهیت داده‌ها بر اساس مسئله مشخص می‌شود.

شناسایی الگو: هنگامی که داده‌ها اکتشاف، پالایش و برای متغیرهای خاص تعریف شدند، مرحله دوم شناسایی الگو است. الگوهایی را که بهترین پیش بینی را انجام می‌دهند، شناسایی و انتخاب می‌شوند.

استقرار: الگوها برای نتیجه مطلوب به کار گرفته می‌شوند.

الگوریتم‌ها و تکنیک‌های مختلفی مانند طبقه‌بندی، خوشه‌بندی، رگرسیون، هوش مصنوعی، شبکه‌های عصبی، قوانین انجمن، درخت‌های تصمیم، الگوریتم‌های ژنتیک، روش نزدیک‌ترین همسایه و غیره برای

کشف دانش از پایگاه‌های داده استفاده می‌شوند.

طبقه‌بندی رایج‌ترین تکنیک داده‌کاوی است که از مجموعه‌ای از نمونه‌های از پیش طبقه‌بندی شده برای توسعه مدلی استفاده می‌کند که می‌تواند جمعیت رکوردها را به طور کلی طبقه‌بندی کند. کاربردهای تشخیص تقلب و ریسک اعتباری به ویژه برای این نوع تحلیل مناسب هستند. این رویکرد اغلب از درخت‌های تصمیم یا الگوریتم‌های طبقه‌بندی مبتنی بر شبکه عصبی استفاده می‌کند.

فرآیند طبقه‌بندی داده‌ها شامل یادگیری و طبقه‌بندی می‌شود. در یادگیری داده‌های آموزشی توسط الگوریتم طبقه‌بندی تجزیه و تحلیل می‌شوند. در آزمون طبقه‌بندی از داده‌ها برای تخمین دقت قوانین طبقه‌بندی استفاده می‌شود. اگر دقت قابل قبول باشد، قوانین را می‌توان به داده‌های جدید اعمال کرد. الگوریتم آموزش طبقه‌بندی از این مثال‌های از پیش طبقه‌بندی‌شده برای تعیین مجموعه پارامترهای مورد نیاز برای تمایز مناسب استفاده می‌کند. سپس الگوریتم این پارامترها را در مدلی به نام طبقه‌بندی کننده رمزگذاری می‌کند [۱].

طبقه‌بندی یک وظیفه مهم در یادگیری ماشین و داده کاوی است که هدف آن طبقه‌بندی هر نمونه در مجموعه داده‌ها به گروه‌های مختلف بر اساس اطلاعات توصیف شده توسط ویژگی‌های آن است. بدون دانش قبلی، تعیین اینکه کدام ویژگی‌ها مفید هستند دشوار است. زیرا معمولا تعداد زیادی از ویژگی‌ها در مجموعه داده‌ها وجود دارد که شامل ویژگی‌های مرتبط، نامربوط و افزونه هستند. این ویژگی‌ها به دلیل فضای جستجوی بزرگی که تحت عنوان "مصیبت ابعاد" شناخته می‌شود، حتی می‌توانند کارایی طبقه‌بندی را کاهش دهند. انتخاب مشخصه می‌تواند تنها با انتخاب ویژگی‌های مربوطه برای طبقه‌بندی به این مشکل بپردازد. با حذف / کاهش ویژگی‌های نامربوط و اضافی، انتخاب ویژگی می‌تواند تعداد ویژگی‌ها را کاهش دهد، زمان آموزش را کوتاه کند، طبقه‌بندی کننده‌های یاد گرفته شده را ساده کند، و / یا عملکرد طبقه‌بندی را بهبود بخشد، انتخاب ویژگی کار دشواری است زیرا تعامل پیچیده‌ای بین ویژگی‌ها وجود دارد. یک ویژگی فردی مرتبط (زاید یا نامربوط) ممکن است در هنگام کار با دیگر ویژگی‌ها، زاید (مرتبط) شود. بنابراین، یک زیرمجموعه ویژگی بهینه باید گروهی از ویژگی‌های مکمل

باشد که بر روی ویژگی‌های مختلف کلاس‌ها گسترده شده‌اند تا به درستی آن‌ها را از هم تفکیک کنند. روش‌های مختلف انتخاب ویژگی، شامل فیلتر، دسته بندی، تعبیه شده، ترکیبی و گروهی می‌گردد که در این پایان‌نامه از روش انتخاب ویژگی دسته‌بندی استفاده می‌شود.

برای انجام انتخاب ویژگی، فضای گسسته نتایج بهتری را ارائه می‌دهد. جهت نگاشت فضای پیوسته به فضای گسسته روش‌هایی وجود دارد که مفیدترین آن‌ها استفاده از توابع انتقال می‌باشد [۲].

۱-۲ تعریف مسئله

طبقه‌بندی[۱] و انتخاب ویژگی[۲] دو مرحله اساسی در داده‌کاوی[۳] هستند. محدود کردن تعداد ویژگی‌های[۴] ورودی در یک طبقه‌بندی کننده برای تولید یک مدل پیش‌بینی خوب و با پیچیدگی محاسباتی کم‌تر یک مزیت است. با یک زیرمجموعه ویژگی کوچک و مناسب به طور منطقی و ساده‌تر می‌توان در مورد طبقه‌بندی تصمیم گرفت.

در اینجا با استفاده از الگوریتم گرگ خاکستری و توابع انتقال اقدام به انتخاب ویژگی می‌شود، اما نکته‌ای که این کار را با سایر پژوهش‌ها متفاوت می‌کند استفاده از چندین تابع انتقال توام و متفاوت می‌باشد. الگوریتم در ادامه و پیشرفت در تکرارهای بعدی، با استفاده از تابع ارزیابی که در الگوریتم تعریف می‌شود، تابع انتقال بهینه را انتخاب می‌کند. الگوریتم بر روی مجموعه داده‌های مخزن UCI اعمال می‌گردد. از این مخزن ده مجموعه داده انتخاب شده است که تنوع تعداد ویژگی و تعداد نمونه مشهود باشد. خروجی این الگوریتم با توجه به مفهوم انتخاب تابع انتقال در طی تکرارهای آن، تعیین حداقل ویژگی‌های مفید از کل ویژگی‌های مجموعه داده، با دقت طبقه‌بندی بالا می‌باشد.

[1] Classification

[2] Feature Selection

[3] Data Mining

[4] Feature

۱-۳ اهداف تحقیق

در همه مقالات پیشین روش انتخاب ویژگی، با یک تابع انتقال سروکار دارند و انتخاب و یادگیری تابع انتقال مطرح نبوده است. بنابراین هدف این تحقیق، ارائه یک روش انتخاب ویژگی مبتنی بر یادگیری توابع انتقال است. برای رسیدن به این مقصود، تکنیکی ارائه می‌گردد که ضمن برآورده کردن این هدف، خطای طبقه‌بندی نیز کاهش یابد. آزمایشات انجام شده توسط این روش، مفید بودن آن را تایید می‌کند.

۱-۴ پیش‌فرض‌های تحقیق

فرض می‌کنیم الگوریتم باینری گرگ خاکستری[5] از مجموعه الگوریتم‌های فراابتکاری، برای بررسی روش پیشنهادی ما مناسب می‌باشد. همچنین در این مستند از دو نوع تابع انتقال S و V با شیب‌های مختلف استفاده شده است. ده مجموعه داده مورد استفاده از مخزن UCI، عبارتند از Breast Cancer Wisconsin با ۱۰ ویژگی و ۶۹۹ نمونه، Congressional Voting Records با ۱۶ ویژگی و ۴۳۵ نمونه، Statlog (Heart) با ۱۳ ویژگی و ۲۷۰ نمونه، Ionosphere با ۳۴ ویژگی و ۳۵۱ نمونه، Chess با ۳۶ ویژگی و ۳۱۹۶ نمونه، Lymphography با ۱۸ ویژگی و ۱۴۸ نمونه، Connectionist Bench با ۶۰ ویژگی و ۲۰۸ نمونه، SPECT Heart با ۲۲ ویژگی و ۲۶۷ نمونه، Tic-Tac-Toe Endgame با ۹ ویژگی و ۹۵۸ نمونه، Zoo با ۱۷ ویژگی و ۱۰۱ نمونه.

برای مقایسه نتایج حاصل از کار، مقادیر با یکی از مقالات مطرح در این حوزه مقایسه خواهد شد. محدوده تحقیق بر روی توابع انتقال در الگوریتم‌های باینری فراابتکاری قابل پیاده سازی است.

[5] Binary Gray Wolf Optimization Algorithm

۱-۵ روش تحقیق

در این تحقیق بعد از استخراج ویژگی‌ها، از یکی از مجموعه داده UCI، n بیت باینری به تعداد ویژگی‌های آن اضافه می‌شود. این بیت‌ها در اول الگوریتم مقداردهی می‌شود بنابراین از همان ابتدای الگوریتم، هر گرگ دارای تابع انتقال و شیب مخصوص به خود می‌باشند. از این رو، این بیت‌ها ملاکی برای انتخاب تابع انتقال و یا شیب تابع انتقال که از قبل تعیین شده است برای هر گرگ استفاده می‌شود. در این حالت، الگوریتم ۲^n حالت برای انتخاب تابع انتقال و یا شیب تابع انتقال در اختیار دارد. در طول اجرای الگوریتم با توجه به تابع ارزیابی و با توجه به یادگیری، که حین اجرای الگوریتم انجام می‌شود، گرگ‌ها موقعیت خود را بروز می‌کنند. بدیهی است با تغییر موقعیت گرگ‌ها و نتیجه حاصل از تابع ارزیابی امکان تغییر گرگ آلفا وجود دارد. با تغییر گرگ آلفا، تابع انتقال وابسته به این گرگ در آن تکرار تابع انتقال الگوریتم خواهد شد. با تکرارهای بعدی، الگوریتم ضمن انتخاب تابع انتقال مناسب با شیب مناسب، یادگیری تابع انتقال را نیز انجام می‌دهد، تا با کمترین خطا، تعداد ویژگی مناسب به بهترین نحو بدست آید.

نوآوری ما در این بخش همانطور که در بالا شرح داده شد ارائه روشی جدید برای یادگیری تابع انتقال و انتخاب شیب مناسب در حین اجرای الگوریتم می‌باشد که انتخاب ویژگی را محقق می‌کند.

۱-۶ ساختار پایان‌نامه

در فصل ابتدایی پایان‌نامه به کلیات تحقیق و تعریف مسئله و اهدافی که در این مسئله مدنظر می‌باشد، به صورت مختصر بیان شده است. فصل دوم ادبیات تحقیق و مرور کارهای پیشین است که به معرفی مفاهیم پایه مانند k نزدیکترین همسایه، انتخاب ویژگی و انواع آن، توضیح الگوریتم فراابتکاری و در نهایت به تشریح الگوریتم گرگ خاکستری باینری و مرور کارهای پیشین پرداخته خواهد شد. در فصل سوم، روش تحقیق می‌باشد که به تشریح کامل روش پیشنهادی می‌پردازد. فصل چهارم، نتایج

پیاده‌سازی و مقایسه نتایج با نتایج یکی از مقاله‌های معتبر و ارزیابی روش پیشنهادی، ارائه شده است.

فصل پنجم شامل جمع‌بندی و ارائه پیشنهادهایی برای انجام تحقیقات بیشتر خواهد بود.

فصل ۲: ادبیات تحقیق و مرور کارهای پیشین

۲-۱ مقدمه

بکارگیری مطالعات گذشته بر ارزش و صداقت بیشتر تحقیقات و موجب جلوگیری از تکرار و هدر رفتن وقت و آشنایی با ابزار بکار رفته و در نتیجه استفاده از مطالعات قبلی برای ارائه فرضیه‌های جدید می‌شود. پیشینه تحقیق در واقع خلاصه‌ای از تمامی مطالعات مهمی است که تاکنون در زمینه مورد نظر شما به انجام رسیده است. در بخش پیشینه تحقیق باید در مورد نظریه‌ها و مدل‌های اصلی به کار برده شده در تحقیق و حتی کارهای پیشین بحث شود. مهم‌ترین وظیفه در نوشتن پیشینه تحقیق این است که یک ارتباط منطقی میان آنچه انجام شده و آنچه قصد انجام آن را داشت، ایجاد شود.

در ادامه برای درک کارهای انجام شده به معرفی مفاهیم پایه مانند k نزدیکترین همسایه، انتخاب ویژگی و انواع آن، توضیح الگوریتم فراابتکاری گرگ خاکستری و به کارهای انجام شده در خصوص توابع انتقال و کاربرد آن‌ها در الگوریتم پرداخته خواهد شد.

۲-۲ k نزدیکترین همسایه (KNN[6])

K نزدیک‌ترین همسایه یک روش ساده و رایج برای طبقه‌بندی و یک الگوریتم یادگیری تحت نظارت است [۳]. با افزایش قدرت محاسباتی کامپیوترها، الگوریتم KNN محبوب گشت و یکی از کاربردهای رایج آن تشخیص الگو است. این روش مبتنی بر تخمین نزدیکترین همسایه می‌باشد. برای یک داده‌ی آزمایشی الگوریتم به دنبال k نمونه از نزدیکترین نمونه‌ها می‌گردد (k نمونه‌ی مشابه). نزدیکی دو نمونه، با بدست آوردن تشابه و یا فاصله‌ی میان این دو نمونه محاسبه می‌شود. هر نمونه می‌تواند از انواع داده‌ها تشکیل شده باشد که باید تشابه میان آن‌ها بررسی شود. منظور از فاصله، فاصله اقلیدسی می‌باشد که بصورت زیر تعریف می‌شود:

[6] K-Nearest Neighbour

$$Distance(x,y) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2} \quad (۱-۲)$$

d تعداد ویژگی‌هاست. در دسته‌بند KNN، برای تعیین برچسب نمونه M، روال زیر انجام می‌شود:

۱) k نمونه‌ای که به x از نظر معیار فاصله نزدیکتر هستند، تعیین می‌شوند.

۲) از این k نمونه برای تعیین برچسب نمونه x، رای‌گیری می‌شود. برچسب با تعداد حداکثر آرا، برابر با برچسب تخمین زده شده برای x خواهد بود.

پس از یافتن این k داده‌ی مشابه با نمونه‌ی آزمایشی، با رأی اکثریت برچسب کلاس داده‌ی آزمایشی انتخاب می‌شود. چناچه مقدار یک برای k تنظیم شود، در این صورت کلاس نزدیکترین داده به نمونه‌ی آزمایشی، به عنوان کلاس تخمینی ارائه می‌شود. اما به دلیل وجود داده‌های نویز و خارج از محدوده مقدار یک عدد مناسبی برای k نیست. می‌توان مقدار مناسب را به صورت تجربی به دست آورد. برای مثال با یک شروع و برای مجموعه‌ای آزمایشی نرخ خطا محاسبه می‌گردد. با افزایش مقدار k این کار تکرار می‌شود. مقداری از k که باعث حداقل نرخ خطا می‌شود، انتخاب مناسبی است. درکاربرد مقادیر سه و پنج برای k نتایج خوبی را به دنبال داشته است.

این الگوریتم همچنین می‌تواند برای داده‌هایی که برچسب کلاس آن‌ها از نوع پیوسته (عددی) است نیز استفاده شود. در این صورت پس از یافتن k همسایه، میانگین مقادیر حاصل از کلاس این k نمونه، به عنوان برچسب کلاس نمونه‌ی آزمایشی برگزیده می‌شود. هرگاه مقدار یا مقادیری از ویژگی در مجموعه داده اصلی یا آزمایشی ناقص باشند در محاسبه‌ی تشابه، کمترین و در محاسبه‌ی فاصله، بیشترین فاصله برای این ویژگی در نظر گرفته می‌شود. فرض کنید مقادیر ویژگی یک ویژگی عددی، به فاصله‌ی صفر تا یک نگاشت می‌شوند. می‌دانید که این عمل می‌تواند با یک نرمال سازی ساده انجام گردد. برای محاسبه فاصله‌ی میان دو نمونه و برای این ویژگی اگر هر دو مقدار ناقص باشند فاصله برابر یک (تشابه برابر با صفر) در نظر گرفته می‌شود. اما چناچه یکی از این مقادیر ناقص باشد و دیگری دارای ارزشی برابر با v، فاصله و تشابه از فرمول ۲-۲ به دست می‌آیند:

$$Dis(A_i, A_j) = Max(|I - v|, |0 - v|) \quad (۲-۲)$$
$$Sim(A_i, A_j) = Min(|I - v|, |0 - v|)$$

برای ویژگی غیرعددی کافی است حداقل یکی از آن‌ها ناقص باشد، تا فاصله برابر با یک و تشابه برابر با صفر تنظیم شود.

با انتساب وزن به هر یک از ویژگی‌ها درصد مشارکت ویژگی در محاسبه‌ی تشابه و یا فاصله‌ی میان نمونه‌ها کمتر یا بیشتر خواهد شد. بدین ترتیب ویژگی‌ای نامربوط و یا داده‌های نویز تاثیر کمتری در فرایند خواهند داشت.

به دلیل اینکه الگوریتم KNN برای یافتن برچسب کلاس داده‌های آزمایشی باید کلیه‌ی داده‌ها را پیمایش کند، این عمل در داده‌هایی با حجم بسیار بالا می‌تواند به شدت از کارایی الگوریتم بکاهد. تمهیداتی وجود دارند که پیچیدگی الگوریتم را بهبود می‌بخشند. ذخیره‌ی داده‌های اولیه در یک ساختار درختی می‌تواند پیچیدگی جستجو و پیمایش را لگاریتمی کند و یا با پیاده سازی موازی الگوریتم انتظار می‌رود سرعت بهتری از الگوریتم داشت.

روش کاربردی دیگر جهت بهبود الگوریتم اولیه، محاسبه‌ی فاصله و یا تشابه میان زیرمجموعه‌ای از ویژگی به جای فضای کل است. بدین ترتیب که در ابتدا فاصله‌ی میان نمونه‌ی آزمایشی و داده‌های ذخیره شده با توجه به زیرمجموعه‌ای از ویژگی به جای کلیه‌ی آن‌ها محاسبه می‌شود. در این صورت چنانچه فاصله (تشابه) از مقدار تعریف شده‌ای بیشتر (کمتر) باشد، بدون محاسبه‌ی کامل فاصله یا تشابه به سراغ داده‌ی بعدی خواهیم رفت. به علاوه اگر به هر نحوی قادر باشیم برخی از نمونه‌ها را از فضای جستجو خارج کنیم بدون شک سرعت الگوریتم افزایش می‌یابد. جدول ۲-۱ حاوی مشخصات شش نقطه است که در دوکلاس A و B طبقه‌بندی شده‌اند.

جدول ۲-۱: مشخصات دکارتی شش نقطه همراه با کلاس هر یک [۴]

ردیف	X	Y	کلاس	p فاصله تا
۱	۲	۵	A	۳.۰۰
۲	۶	۳	B	۴.۱۲
۳	۴	۴	A	۲.۸۳
۴	۱	۲	B	۱.۰۰
۵	۱	۶	B	۴.۱۲
۶	۳	۲	A	۱.۰۰

با تنظیم عدد سه برای مقدار k می‌شود کلاس نقطه (۲,۲) P را با روش KNN بدست آورد. در ستون آخر جدول فاصله‌ی نقطه‌ی P با هر یک از نمونه‌ها نیز محاسبه شده است (فاصله‌ی اقلیدسی). از آنجا که مقدار k برابر با سه است، الگوریتم سه نمونه‌ی نزدیک به نقطه P را انتخاب خواهد کرد. نمونه‌های سوم، چهارم و ششم انتخاب می‌شوند. درمیان این سه نمونه، دو برچسب A و یک برچسب B قرار داردکه بدین ترتیب کلاس نقطه‌ی P برابر A تخمین زده می‌شود [۴].

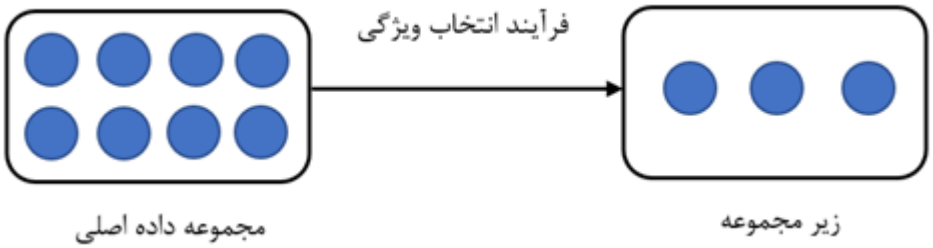
۲-۳ انتخاب ویژگی

با افزایش حجم روز افزون داده‌ها در سال‌های اخیر، مشکل داده‌های با ابعاد بالا بوجود آمده است که، باعث کاهش کارایی و دقت طبقه‌بندی در مبحث یادگیری ماشین و شناسایی الگو می‌شود [۵]. با این حال، استفاده از جستجو برای اکثر مسائل دنیای واقعی به دلیل داشتن بعد بالا به خصوص در مسائل NP - hard غیر ممکن است [۵]. بدیهی است که بررسی کل فضای مساله و ارزیابی همه حالت‌ها از نظر پیچیدگی محاسباتی و زمان واکنش بسیار پرهزینه هستند [۵]. انتخاب ویژگی در داده‌های با ابعاد بالا یکی از اساسی‌ترین مراحل یادگیری ماشین به حساب می‌آید. روش‌های انتخاب ویژگی برای مسائل

طبقه‌بندی به منظور انتخاب مجموعه ویژگی کاهش‌یافته به کار گرفته شده‌اند که طبقه‌بندی کننده را دقیق‌تر و سریع‌تر می‌کند [۶،۵].

انتخاب ویژگی راهی برای شـناسـایی ویژگی‌های مهم و حذف ویژگی‌های نامربوط (افزونه) از مجموعه داده‌ها فراهم می‌کند [۸،۷]. اهداف انتخاب ویژگی، کاهش بعد داده، بهبود عملکرد پیش‌بینی و درک خوب داده برای کاربردهای یادگیری ماشین است [۳،۷،۹].

روش‌های قدیمی انتخاب ویژگی، توانایی رویارویی با حجم زیاد این داده‌ها را ندارند و نمی‌توانند مانند داده‌های کلاسیک بر روی داده‌های با بعد بالا نیز موثر واقع شوند از این رو استفاده از روش‌های نوین که بتواند جوابگوی انتخاب ویژگی‌های موثر و حذف ویژگی‌های افزونه و نامرتبط در مجموعه داده‌های با ابعاد بالا باشد بسیار ضروری به نظر می‌رسد. بنابراین انتخاب مناسب زیرمجموعه ویژگی و تنظیم پارامترهای مدل تاثیر زیادی بر دقت طبقه‌بندی دارد [۱۰]. وقتی با حجم زیاد داده روبرو هستیم، یکی از روش‌های نوین و به روز که پاسخگوی ما در برخورد با مشکلات انتخاب ویژگی می‌باشد، استفاده از روش‌های نوین انتخاب ویژگی و الگوریتم‌های فراابتکاری است [۱۱،۵]. در ادامه به روش‌های مختلف انتخاب ویژگی می‌پردازیم:



شکل ۲-۱: مفهوم انتخاب ویژگی [۱۲]

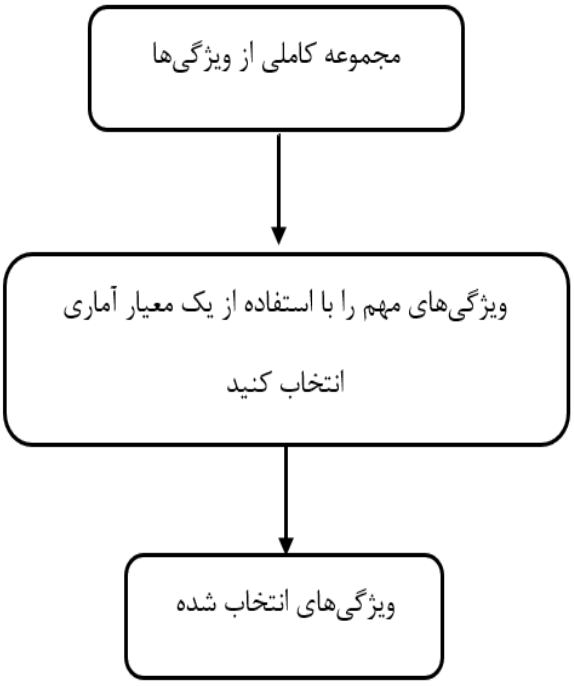
۲-۳-۱ روش فیلتر[۷]

یکی از روش‌های انتخاب ویژگی روش فیلتر می‌باشد. روش فیلتر مستقل از الگوریتم یادگیری عمل می‌کنند [۳،۲]. در این روش، به هر ویژگی با استفاده از معیارهای آماری از خصوصیات ذاتی داده یک

[7] Filter

مقدار امتیاز اختصاص داده می‌شود [۳]. ویژگی‌ها براساس امتیازات و تعیین رتبه‌بندی مشخصه‌ها به ترتیب نزولی سازماندهی می‌شوند. خروجی این روش، ویژگی‌هایی با بالاترین رتبه‌ها می‌باشد. از آنجا که همبستگی بین متغیرهای مستقل در زمان انتخاب ویژگی‌ها در نظر گرفته نمی‌شود، این امر منجر به انتخاب ویژگی‌های اضافی می‌شود. این روش دارای سرعت بالایی نسبت به سایر روش‌ها می‌باشند [۱۲] و به همین دلیل گزینه مناسبی برای استفاده در داده‌های با بعد بالا می‌باشد. روش فیلتر از نظر محاسباتی ارزان‌تر و کلی‌تر هستند [۲]، اما این روش از دقت مناسبی برخوردار نمی‌باشد. روش انتخاب ویژگی فیلتر ممکن است برخی ویژگی‌های اطلاعاتی را نادیده بگیرند [۱۳].

از جمله روش‌های فیلتر می‌توان به بهره اطلاعاتی[8] [۱۴،۱۵]، امتیاز فیشر[9] [۱۶]، Relief-F [۱۷]، روش فیلتر براساس همبستگی پیرسون[10] [۲۰،۱۹،۱۸] اشاره کرد.



شکل ۲-۲: روش فیلتر [۲۱]

[8] Information Gain

[9] F-Score

[10] Pearson’s correlation

جدول ۲-۲: مزایا و معایب روش فیلتر [۲۲،۱۲]

	مزایا	معایب
روش فیلتر	- کارآمد و ازنظر محاسباتی سریع‌تر - مستقل از الگوریتم یادگیری - از نظر محاسباتی سریع‌تر از روش‌های دسته‌بند[۱۱] و تعبیه‌شده[۱۲] است. - مناسب برای داده‌های با ابعاد بالا - روش‌های فیلتر قابلیت تعمیم خوب را دارند.	- این روش ارتباط بین طبقه‌بندی کننده‌ها را در نظر نمی‌گیرد. - در مرحله یادگیری، الگوها را به درستی تشخیص نمی دهد - روش‌های فیلتر ممکن است ویژگی‌هایی را که ممکن است به طور مستقل بی‌ربط باشند از دست بدهند اما هنگامی که با سایر ویژگی‌ها ترکیب شوند تاثیرگذارتر هستند. - این مساله در مقایسه با دیگر روش‌های پیشرفته انتخاب ویژگی مانند ترکیبی[۱۳] دقت کمتری دارد.

۲-۳-۲ روش دسته‌بند

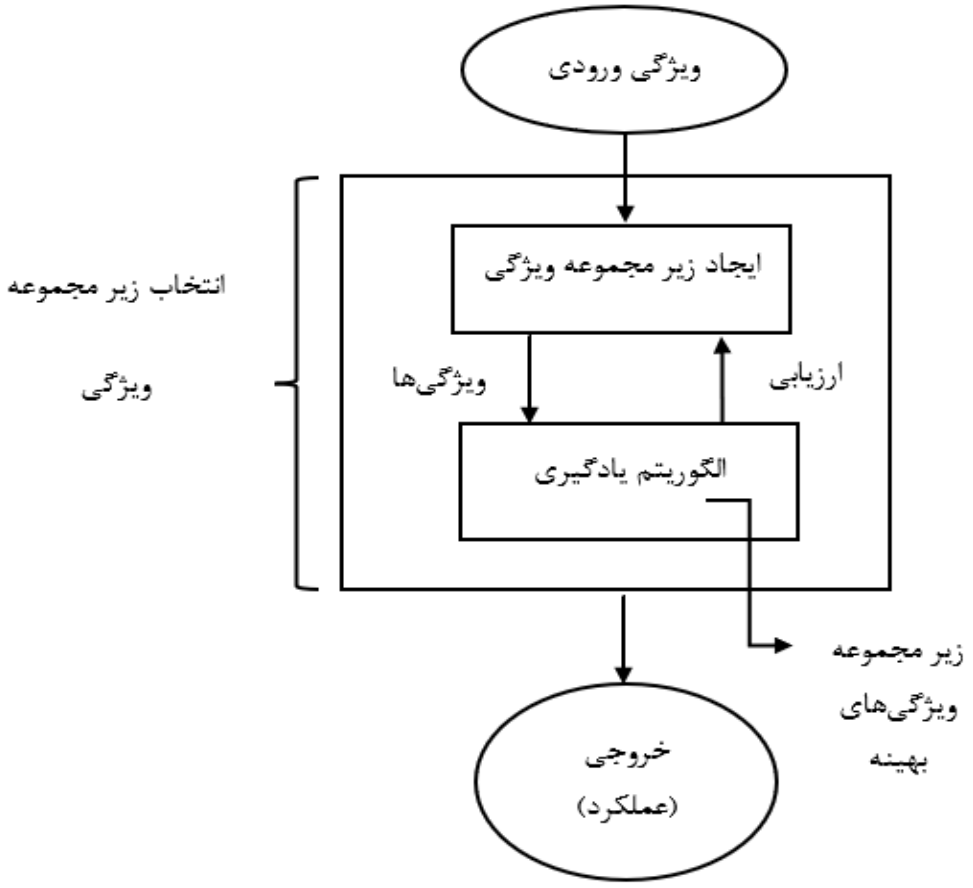
روش دسته‌بند تعاملی بین الگوریتم طبقه‌بندی و زیر مجموعه جستجو ایجاد می‌کند [۱۲]. در واقع عملکرد روش دسته‌بند به طبقه‌بندی کننده بستگی دارد [۲]. بهترین زیرمجموعه ویژگی‌ها براساس نتایج طبقه‌بندی کننده، انتخاب می‌شود [۳،۵]. این روش یک زیر روال را به عنوان یک تکنیک نمونه‌برداری آماری (به عنوان مثال، اعتبار سنجی متقابل) اجرا می‌کند که با استفاده از الگوریتم یادگیری برای تخمین دقت زیرمجموعه‌های ویژگی عمل می‌کند [۱۲]. روش دسته‌بند برتری خود را در وظایف طبقه‌بندی نشان داده است، زیرا آن‌ها در زمان حل مساله (دنیای واقعی) با بهینه‌سازی عملکرد طبقه‌بندی کننده به خوبی عمل می‌کنند [۱۲]. روش‌های دسته‌بند به دلیل مراحل یادگیری مکرر

[11] Wrapper

[12] Embedded

[13] Hybrid

[۲۳] و اعتبار متقابل، از نظر محاسباتی گران‌تر از روش‌های فیلتر هستند [۲۲]. با این حال، این روش‌ها دقیق‌تر از روش فیلتر هستند [۲۴،۲۵]. برخی از مثال‌های این روش، شامل حذف ویژگی‌های بازگشتی، الگوریتم‌های انتخاب ویژگی متوالی و الگوریتم‌های ژنتیک می‌باشد [۲۲]. یک مورد خاص از انتخاب ویژگی ترتیبی الگوریتم جستجوی حریصانه است که می‌تواند زیر مجموعه ویژگی را با انتخاب مکرر ویژگی‌ها براساس عملکرد یک طبقه‌بندی کننده تعیین کند. این الگوریتم با یک زیرمجموعه ویژگی صفر شروع می‌شود و یک ویژگی را یکی پس از دیگری اضافه می‌کند. در شکل ۲-۳، روش دسته‌بند با استفاده از یک طبقه‌بندی کننده از پیش تعریف شده، زیر مجموعه‌ای از ویژگی‌ها را استخراج می‌کند. سپس طبقه‌بندی کننده را برای اندازه‌گیری زیر مجموعه ویژگی‌های انتخاب‌شده اعمال می‌کند. انتخاب و اندازه‌گیری زیرمجموعه‌های ویژگی‌ها تا رسیدن به معیار مطلوب کیفیت ادامه می‌یابد [۱۲].

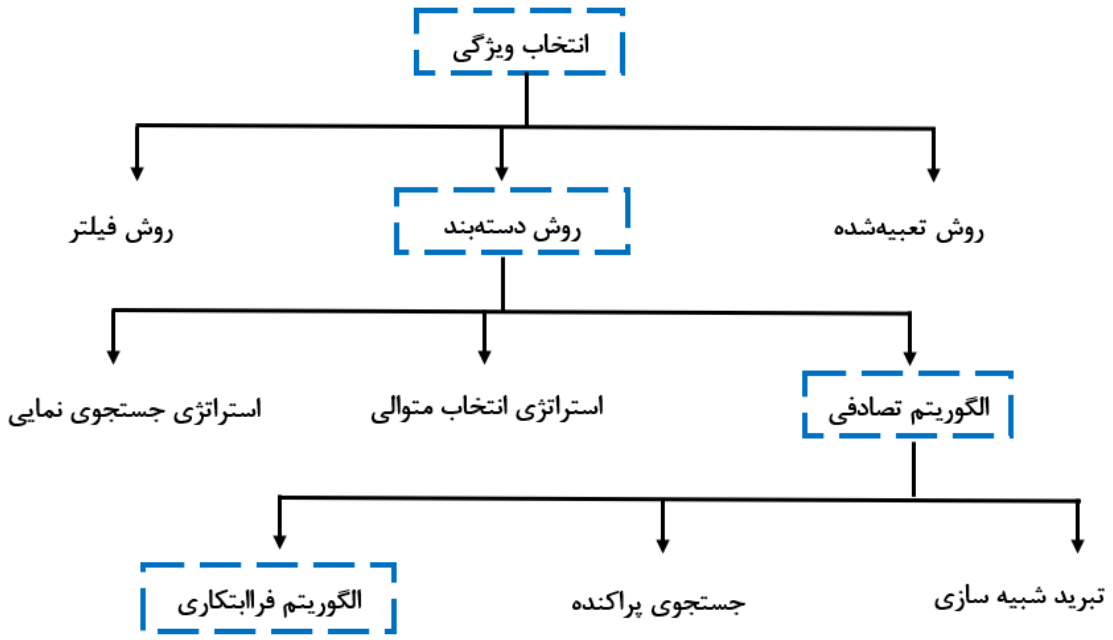


شکل ۲-۳: روش دسته‌بند[۱۲]

جدول ۲-۳: مزایا و معایب روش دسته‌بند [۲۲،۱۲]

	مزایا	معایب
روش دسته‌بند	- روش‌های دسته‌بند نسبت به روش‌های فیلتر تعمیم خوبی دارند. - روش‌های مبتنی بر دسته‌بند مجموعه ویژگی‌هایی را حفظ می‌کنند که بهترین دقت را به دست می‌دهند. - تعامل با طبقه‌بندی کننده برای انتخاب ویژگی - جستجوی دقیق‌تر فضای مجموعه ویژگی - این روش همبستگی بین ویژگی‌ها و برچسب‌های کلاس رادر نظر می‌گیرد. -دقیق‌تر از روش فیلتر است.	- این روش با توجه به این که الگوریتم باید به طور مکرر فراخوانی شود، کند است. - با افزایش تعداد ویژگی‌های ورودی، از نظر محاسباتی پرهزینه می‌شود. - نسبت به مجموعه داده‌های بزرگ متشکل از ویژگی‌های متعدد، مقیاس خوبی ندارند. - برخی از ویژگی‌ها ممکن است در مرحله ابتدایی برای ارزیابی در نظر گرفته نشوند باعث برازش بیش از حد می‌شود.

عموما الگوریتم‌های فراابتکاری در قالب روش‌های دسته‌بند بیان می‌گردد.



شکل ۲-۴: طبقه‌بندی روش‌های انتخاب ویژگی [۲۶]

۲-۳-۳ روش تعبیه‌شده

روش تعبیه‌شده متفاوت با روش‌های فیلتر و دسته‌بند می باشد، به این معنا که هنوز امکان تعامل با الگوریتم یادگیری برای انتخاب ویژگی را می‌دهند، اما زمان محاسباتی آن‌ها کمتر از روش دسته‌بند است [۲۷]. یک روش تعبیه‌شده زمان محاسباتی را که برای طبقه‌بندی مجدد زیرمجموعه‌های انتخابی مختلف در روش دسته‌بند صرف می‌شود، کاهش می‌دهد [۲۷]. رویکرد اصلی روش تعبیه‌شده، گنجاندن انتخاب ویژگی به عنوان بخشی از فرآیند آموزش است [۲۷]. به عبارت دیگر، انتخاب ویژگی مناسب و یادگیری مدل به طور همزمان انجام می‌شود و ویژگی‌ها در طول مرحله آموزش مدل انتخاب می‌شوند. به همین دلیل، هزینه محاسباتی این روش در مقایسه با روش دسته‌بند قطعا کم‌تر است. این روش، از آموزش مدل در هر زمان که یک انتخاب ویژگی جدید مورد بررسی قرار می‌گیرد، اجتناب می‌کند [۲۲]. روش تعبیه‌شده تلاش می‌کند تا معایب روش‌های فیلتر و دسته‌بند را جبران کند [۲۷]. نه تنها روابط بین یک ویژگی ورودی و ویژگی خروجی را در نظر می‌گیرد، بلکه به صورت محلی برای ویژگی‌هایی جستجو می‌کند که امکان تمایز محلی بهتر را فراهم می‌کند [۲۸]. از معیارهای مستقل برای تعیین زیرمجموعه‌های بهینه برای یک گروه شناخته شده استفاده می‌کند. پس از آن، از الگوریتم یادگیری برای انتخاب زیرمجموعه بهینه نهایی از میان زیرمجموعه های بهینه گروه‌های مختلف استفاده می‌شود [۲۹]. روش تعبیه‌شده را می‌توان تقریباً به سه دسته تقسیم کرد: مدل هرس، مدل مکانیزمی و مدل منظم‌سازی.

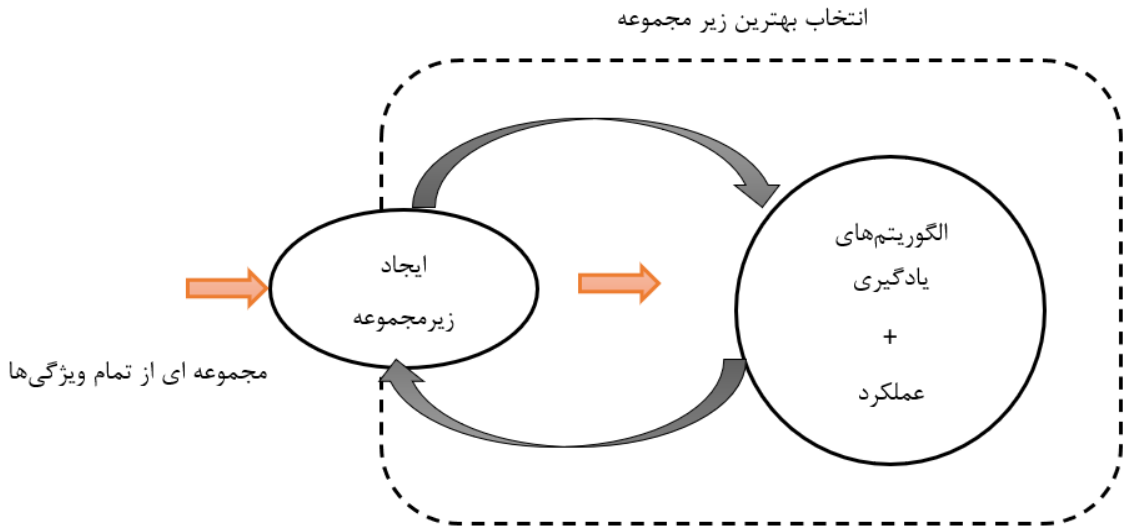
در روش مبتنی بر هرس، ابتدا تمام ویژگی‌ها برای ساخت مدل طبقه‌بندی وارد فرآیند آموزش می‌شوند و ویژگی‌هایی که مقدار ضریب همبستگی کمتری دارند، به صورت بازگشتی با استفاده از ماشین بردار پشتیبان[14] حذف می‌شوند.

در روش انتخاب ویژگی مبتنی بر مکانیزم داخلی، بخشی از مرحله آموزش الگوریتم‌های یادگیری نظارت

[14] Support Vector Machine

شده C4.5 و ID3 برای انتخاب ویژگی‌ها استفاده می‌شود.

در روش منظم‌سازی، خطاهای برازش با استفاده از توابع هدف به حداقل می‌رسند و ویژگی‌های با ضرایب رگرسیون نزدیک به صفر حذف می‌شوند. در شکل ۲-۵ شماتیک روش تعبیه‌شده به نمایش گذاشته شده است.



شکل ۲-۵: روش تعبیه‌شده [۱۲]

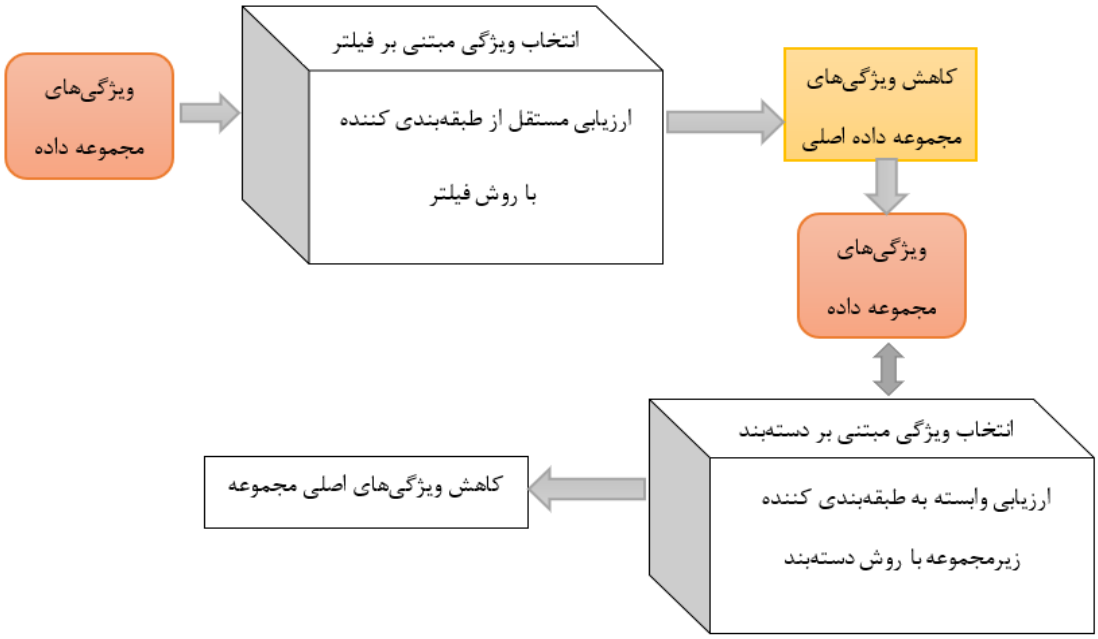
جدول ۲-۴: مزایا و معایب روش تعبیه‌شده [۲۲،۱۲]

	مزایا	معایب
روش تعبیه‌شده	- از نظر محاسباتی کارآمدتر از روش دسته‌بند است. - دقیق‌تر از روش فیلتر و دسته‌بند است.	- از نظر محاسباتی گران‌تر از روش فیلتر است. - برای داده‌های با ابعاد بالا مناسب نیست.

۲-۳-۴ روش ترکیبی

از آنجایی که هر کدام از روش‌های فیلتر یا دسته‌بند، یافتن بهترین جواب را تضمین نکرده و هر کدام دارای مزایا و نقص‌هایی هستند، می‌توان آن‌ها را به صورت مکمل در کنار هم استفاده کرد که به این روش، روش ترکیبی می‌گویند. به عبارت دیگر، روش ترکیبی از دو طبقه تشکیل شده است [۳۰،۳۱].

در طبقه اول، روش فیلتر قرار دارد که به کاهش ابعاد داده می‌پردازد و پس از آن در طبقه دوم توسط روش دسته‌بند، بهترین زیرمجموعه ویژگی، انتخاب خواهد شد [۳۲]. در نتیجه، احتمال حذف شدن ویژگی‌های مطلوب در روش ترکیبی نسبت به روش فیلتر کمتر است [۳۳]. استفاده از روش ترکیبی می‌تواند یکی از بهترین گزینه‌ها برای انتخاب ویژگی در داده‌های با بعد بالا باشد، زیرا حذف ویژگی‌های نامرتبط و افزونه بدون کاهش سرعت و افزایش نه چندان زیاد پیچیدگی محاسباتی انجام می‌شود [۳۴].



شکل ۲-۶: روش ترکیبی [۱۲]

جدول ۲-۵: مزایا و معایب روش ترکیبی [۱۲]

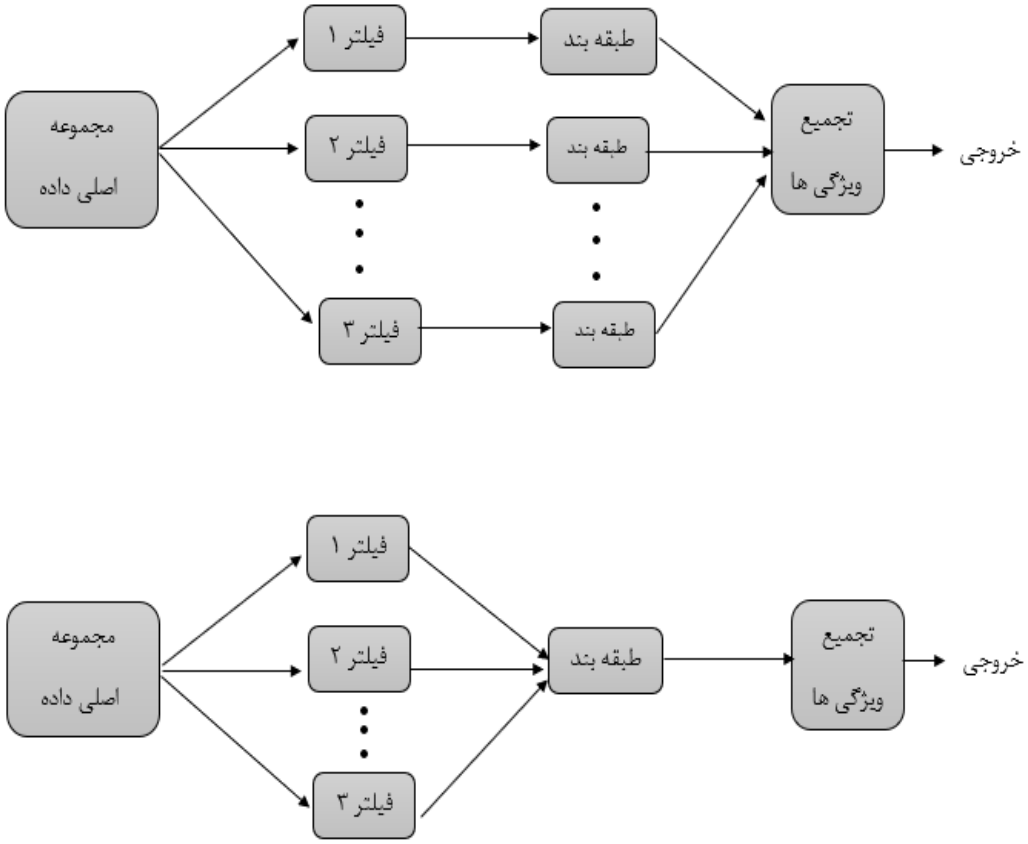
	مزایا	معایب
روش ترکیبی	- کم‌تر مستعد مشکلات بیش برازش است. -آن‌ها همچنین در دستیابی به راه‌حل‌های بهینه سریع‌تر عمل می‌کنند. - تعامل ویژگی‌ها را درک می‌کند	- تکنیک توسعه روش‌های مبتنی بر ترکیبی ممکن است بسیار وحشتناک باشد زیرا نیاز به ترکیب بیش از یک روش دارد.

۲-۳-۵ روش گروهی[۱۵]

داده‌ها ممکن است از نظر تعداد ویژگی‌ها و نمونه‌ها خیلی بزرگ بوده و همچنین از نظر نویز، افزونگی و غیرخطی بودن دارای مشکلاتی باشند. در میان روش‌های موجود، روش دسته‌بند نرخ صحت طبقه‌بندی مطلوبی دارند ولی به دلیل سرعت کار پایین و پیچیدگی محاسباتی بالا، نمی‌توان به خوبی و به تنهایی از آن‌ها در داده‌های با ابعاد بالا بهره برد. از طرفی، روش فیلتر سرعت بسیار بالاتری نسبت به روش دسته‌بند دارند اما دارای نرخ صحت طبقه‌بندی کمتری نسبت به روش دسته‌بند است. در روش ترکیبی وقتی روش فیلتر به عنوان طبقه اول بر روی مجموعه داده‌ها اعمال شود، این احتمال وجود دارد که بعضی ویژگی‌های مفید به علت خطا، حذف شوند.

اخیرا روش‌های انتخاب ویژگی به نام روش گروهی [۳۵] مورد توجه محققان قرار گرفته است. روش گروهی مانند روش ترکیبی است با این تفاوت که در طبقه اول از چند تا روش فیلتر استفاده می‌شود زیرا وجود چند تا روش فیلتر موجب می‌شود که در صورت حذف یک ویژگی مفید توسط یک روش، این ویژگی مفید در روش‌های دیگر دیده شود بدین ترتیب ویژگی مفیدی در مراحل انتخاب ویژگی از بین نمی‌رود در مرحله دوم خروجی‌های حاصل شده از مرحله اول به صورت مجزا و یا با روش‌های مختلفی با هم ترکیب شده و ویژگی‌های منتخب نهایی را در خروجی ایجاد می‌کنند [۳۶]. در شکل ۲-۷ دیاگرام دو رویکرد انتخاب ویژگی به روش گروهی نشان داده شده است.

[15] Embedded



شکل ۲-۷: روش گروهی [۳۶،۳۵]

۲-۴ الگوریتم فراابتکاری

الگوریتم‌های فراابتکاری از طبیعت، رفتار اجتماعی و رفتار بیولوژیکی (حیوانات، پرندگان، خفاش، وال، گرگ‌ها، و غیره) الهام گرفته شده‌اند. بسیاری از محققان روش‌های محاسباتی متفاوتی را برای تقلید رفتار این گونه‌ها در جستجوی غذا (راه‌حل بهینه) پیشنهاد کرده‌اند [۳۷]. روش‌های مختلف فراابتکاری رفتار سیستم‌های فیزیکی و بیولوژیکی را در طبیعت تقلید می‌کنند و به عنوان روش‌های قوی برای بهینه‌سازی پیشنهاد شده‌اند. در الگوریتم‌های فراابتکاری، شیوه‌های فراابتکاری به چهار دلیل اصلی، سادگی، انعطاف‌پذیری، مکانیسم بدون مشتق و اجتناب از بهینه‌سازی محلی به طور قابل‌توجهی رایج شده‌اند [۳۸].

اول، شیوه‌های فراابتکاری نسبتا ساده هستند. آن‌ها عمدتا از مفاهیم بسیار ساده الهام گرفته‌اند. الهامات معمولا به پدیده‌های فیزیکی، رفتارهای حیوانات و یا مفاهیم تکاملی مربوط می‌شوند. سادگی به دانشمندان کامپیوتر اجازه می‌دهد تا مفاهیم طبیعی مختلف را شبیه‌سازی کنند، فراابتکاری جدید را پیشنهاد دهند، دو یا چند فراابتکاری را ترکیب کنند، یا فراابتکاری‌های فعلی را بهبود بخشند. علاوه بر این، سادگی به دانشمندان دیگر کمک می‌کند تا به سرعت شیوه‌های فراابتکاری را یاد بگیرند و از آن‌ها برای مشکلات خود استفاده کنند.

دوم، انعطاف‌پذیری، به قابلیت اجرای فراابتکاری در مسائل مختلف بدون هیچ تغییر خاصی در ساختار الگوریتم اشاره دارد. فراابتکاری به آسانی برای انواع مختلف مشکلات را از آنجا که به صورت جعبه سیاه در نظر می‌گیرند، قابل کاربرد هستند. به عبارت دیگر، تنها ورودی‌ها و خروجی‌های یک سیستم برای یک فراابتکاری مهم هستند. بنابراین، تمام نیازهای یک طراح این است که بداند چگونه مشکل خود را برای شیوه‌های فراابتکاری نشان دهد.

سوم، اکثر فراابتکاری‌ها دارای مکانیسم‌های بدون مشتق هستند. برخلاف روش‌های بهینه‌سازی گرادیان محور، فراابتکاری‌ها مسائل را به طور تصادفی بهینه می‌کنند. فرآیند بهینه‌سازی با راه‌حل تصادفی شروع می‌شود و نیازی به محاسبه مشتق فضاهای جستجو برای یافتن بهینه نیست.

در نهایت، فراابتکاری‌ها توانایی بالایی برای جلوگیری از بهینه‌سازی محلی در مقایسه با تکنیک‌های بهینه‌سازی مرسوم دارند. این امر به دلیل ماهیت تصادفی فراابتکاری‌ها است که به آن‌ها اجازه می‌دهد تا از رکود در راه‌حل‌های محلی جلوگیری کرده و کل فضای جستجو را به طور گسترده جستجو کنند. فضای جستجو برای مسائل موجود معمولا ناشناخته و با تعداد زیادی از بهینه‌سازی محلی بسیار پیچیده است [۴۳-۳۹]. بنابراین شیوه‌های فراابتکاری گزینه‌های خوبی برای بهینه‌سازی این مسائل موجود چالش برانگیز هستند.

به طور کلی، فراابتکاری‌ها را می‌توان به دو دسته اصلی تقسیم کرد: مبتنی بر راه‌حل واحد[۱۶] و مبتنی

[16] Single solution

بر جمعیت[۱۷] [۳۸].

در گروه مبتنی بر راه‌حل واحد، فرآیند جستجو با یک راه‌حل انتخابی شروع می‌شود. سپس این راه‌حل تک کاندید در طول دوره تکرار بهبود می‌یابد. با این حال، روش‌های فراابتکاری مبتنی بر جمعیت، بهینه‌سازی را با استفاده از مجموعه‌ای از راه‌حل‌ها (جمعیت) انجام می‌دهند. در این حالت، فرآیند جستجو با یک جمعیت اولیه تصادفی (راه‌حل‌های چندگانه) شروع می‌شود و این جمعیت در طول دوره تکرار افزایش می‌یابد.

یکی از شاخه‌های جالب فراابتکاری مبتنی بر جمعیت، هوش ازدحام[۱۸] است [۳۸]. مفاهیم هوش ازدحام برای اولین بار در سال ۱۹۹۳ مطرح شد [۳۸]. برخی از رایج‌ترین تکنیک‌های هوش ازدحام، الگوریتم بهینه‌سازی کلونی مورچه[۱۹] [۴۴]، روش بهینه‌سازی ازدحام ذرات[۲۰] [۴۵] و کلونی زنبور عسل[۲۱] می‌باشد[۴۶].

برخی از مزایای الگوریتم‌های هوش ازدحام عبارتند از [۳۸]:

۱. اغلب از حافظه برای ذخیره بهترین راه‌حل به‌دست‌آمده تاکنون استفاده می‌کنند.
۲. معمولا پارامترهای کمتری برای تنظیم دارند.
۳. در مقایسه با روش‌های تکاملی دارای عملگرهای کمتری هستند.
٤. پیاده‌سازی الگوریتم‌های هوش ازدحام آسان است.

صرف‌نظر از تفاوت‌های میان روش‌های فراابتکاری، یک ویژگی مشترک دارند، فرآیند جستجوی همه آن‌ها به دو مرحله اکتشاف[۲۲] و بهره‌برداری[۲۳] تقسیم می‌شوند [۳۸].

[17] Population
[18] Swarm Intelligence (SI)
[19] Ant Colony Optimization
[20] Particle swarm optimization
[21] Artificial bee colony
[22] Exploration
[23] Exploitation

فاز اکتشاف به فرآیند بررسی نواحی‌های فضای جستجو تا حد ممکن اشاره دارد. یک الگوریتم باید دارای عملگرهای تصادفی باشد تا به طور تصادفی و سراسری[۲۴] فضای جستجو را در فاز اکتشاف جستجو کند. بهره‌برداری به قابلیت جستجوی محلی در اطراف مناطق به دست آمده در مرحله اکتشاف اشاره دارد. در فاز اکتشاف به علت کاوش در کل فضای جستجو میل به همگرایی کمتر می‌باشد و این فاز زمان‌بر خواهد بود و کارایی کمتر است. در فاز بهره‌برداری همگرایی قوی تر است ولی احتمال به دام افتادن در بهینه‌های محلی بیشتر است. بنابراین یافتن تعادل مناسب بین این دو فاز به دلیل ماهیت تصادفی فراابتکاری یک کار چالش برانگیز محسوب می‌شود [۳۸].

۲-۵ الگوریتم بهینه‌سازی گرگ خاکستری[۲۵]

بهینه‌سازی گرگ خاکستری یک الگوریتم تکاملی است. دو گرگ خاکستری (نر و ماده) موقعیت بالاتری دارند و بقیه گرگ‌ها را در گروه مدیریت می‌کنند. یک نسخه باینری جدید از بهینه‌سازی گرگ خاکستری برای پیدا کردن مناطق بهینه از فضای جستجوی پیچیده پیشنهاد شده است. بهینه‌سازی گرگ خاکستری یکی از آخرین تکنیک‌های الهام‌گرفته از زیست‌شناسی است که فرآیند شکار گروهی از گرگ‌های خاکستری در طبیعت را شبیه‌سازی می‌کند. بهینه‌سازی پیوسته گرگ خاکستری نوع جدید آن باینری بهینه‌سازی گرگ خاکستری برای انتخاب ویژگی است.

۲-۵-۱ بهینه‌سازی پیوسته گرگ خاکستری

گرگ‌ها گروهی زندگی می‌کنند. اندازه گروه به طور متوسط ۵ تا ۱۲ است. آن‌ها قوانین بسیار سختی در سلسله مراتب غالب اجتماعی دارند. چهار نوع گرگ خاکستری مانند آلفا، بتا، دلتا و امگا برای شبیه‌سازی سلسله مراتب رهبری مانند شکل ۲-۸ استفاده می‌شوند [۴۷،۲۹].

[24] Globaly

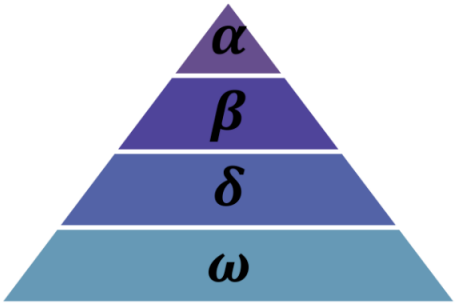
[25] Grey Wolf Optimization

۱. آلفاها گروه را رهبری می‌کنند، گرگ‌های آلفا مسئول برای تصمیم‌گیری هستند.

۲. بتاها فرمانبردار و مطیع گرگ‌های آلفا هستند و در تصمیم‌گیری یا فعالیت‌های دیگر به الفاها کمک می‌کنند. بتا می‌تواند مرد یا زن باشد و او احتمالا بهترین کاندید برای آلفا است.

۳. امگا نقش قربانی را بازی می‌کند و باید تسلیم گرگ‌های حکمران باشند. آن‌ها آخرین گرگ‌هایی هستند که اجازه خوردن دارند.

٤. دلتاها مجبورند تسلیم آلفا و بتاها شوند، اما آن‌ها بر امگا تسلط دارند. پاسبان‌ها، نگهبانان، شکارچیان و مراقبان از این دسته هستند و مسئول مراقبت از مرزهای منطقه هستند.



شکل ۲-۸: سلسله مراتب گرگ خاکستری (سلطه از بالا به پایین کاهش می‌یابد) [۳۸]

مراحل اصلی شکار گرگ خاکستری شامل، ردیابی، تعقیب کردن و نزدیک شدن به طعمه، تعقیب کردن، محاصره کردن، حمله کردن شکار تا توقف آن و حمله به شکار است [۳۸].

این مراحل در شکل ۲-۹ نمایش داده شده است. (A) تعقیب، نزدیک شدن و ردیابی طعمه و (B - D) تعقیب، آزار و دور زدن، محاصره کردن و حمله و (E) ثابت ماندن و حمله کردن [۴۸].

شکل ۲-۹: رفتار شکار گرگ‌های خاکستری [۳۸]

۲-۵-۱-۱ مدل ریاضی و الگوریتم

در این بخش مدل‌های ریاضی سلسله مراتب اجتماعی، ردیابی، محاصره و حمله به طعمه ارائه می‌شوند. سپس الگوریتم گرگ خاکستری مشخص می‌شود.

محاصره طعمه

همان طور که ذکر شد، مدل‌سازی ریاضی، رفتار محاصره کردن گرگ خاکستری در حین شکار به صورت، معادلات ۲-۳ و ۲-۴ پیشنهاد شده‌اند:

$$\vec{X}(t+1) = \overrightarrow{X_P}(t) + \vec{A}.\vec{D} \quad (۲-۳)$$

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_P}(t) - \vec{X}(t)\right| \quad (۲-۴)$$

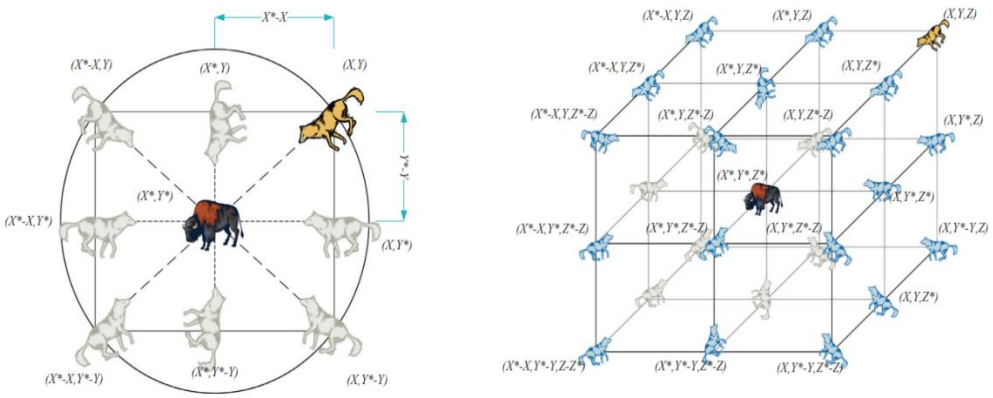
که در آن t تکرار فعلی، A و C بردار ضرایب، X_P بردار موقعیت طعمه و X بردار موقعیت یک گرگ خاکستری را نشان می‌دهد. در شکل ۲-۱۰ بردارهای موقعیت و مکان‌های احتمالی بعدی گرگ به صورت دو بعدی و سه بعدی نشان داده شده‌اند.

بردارهای A و C به صورت معادله ۲-۵ و ۲-۶ تعریف می‌شوند:

$$\vec{A} = 2a.\overrightarrow{r_1} - a \quad (۲-۵)$$

$$\vec{C} = 2\overrightarrow{r_2} \quad (۲-۶)$$

که در آن مولفه‌های $\vec{a}$ به صورت خطی از دو به صفر در طول تکرارها و مولفه r_1 ، r_2 بردارهای تصادفی در $[۰،۱]$ هستند.



شکل ۲-۱۰: بردارهای موقعیت دو بعدی و سه بعدی و مکان‌های احتمالی بعدی آن‌ها [۳۸]

شکار

گرگ‌ها توانایی تشخیص مکان شکار و محاصره کردن آن‌ها را دارند. شکار معمولا توسط آلفا هدایت می‌شود. بتا و دلتا نیز ممکن است گاهی در شکار شرکت کنند. با این حال، در یک فضای جستجوی انتزاعی هیچ ایده‌ای در مورد مکان بهینه (طعمه) وجود ندارد. به منظور شبیه‌سازی ریاضی رفتار شکار گرگ‌های خاکستری، فرض می‌شود که گرگ آلفا (بهترین راه‌حل کاندید) بتا و دلتا دانش بهتری در مورد مکان بالقوه شکار دارند.

بنابراین، سه راه‌حل اول را که تاکنون بدست آمده است ذخیره می‌شود و دیگر عوامل جستجو (از جمله متدها) را ملزم می‌کند تا موقعیت خود را با توجه به موقعیت بهترین عوامل جستجو به روز کنند.

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3} \quad (۷-۲)$$

$$\overrightarrow{X_1} = \left|\overrightarrow{X_\alpha} - \overrightarrow{A_1}.\overrightarrow{D_\alpha}\right| \quad (۸-۲)$$

$$\overrightarrow{X_2} = \left|\overrightarrow{X_\beta} - \overrightarrow{A_2}.\overrightarrow{D_\beta}\right|$$

$$\overrightarrow{X_3} = \left|\overrightarrow{X_\delta} - \overrightarrow{A_3}.\overrightarrow{D_\delta}\right|$$

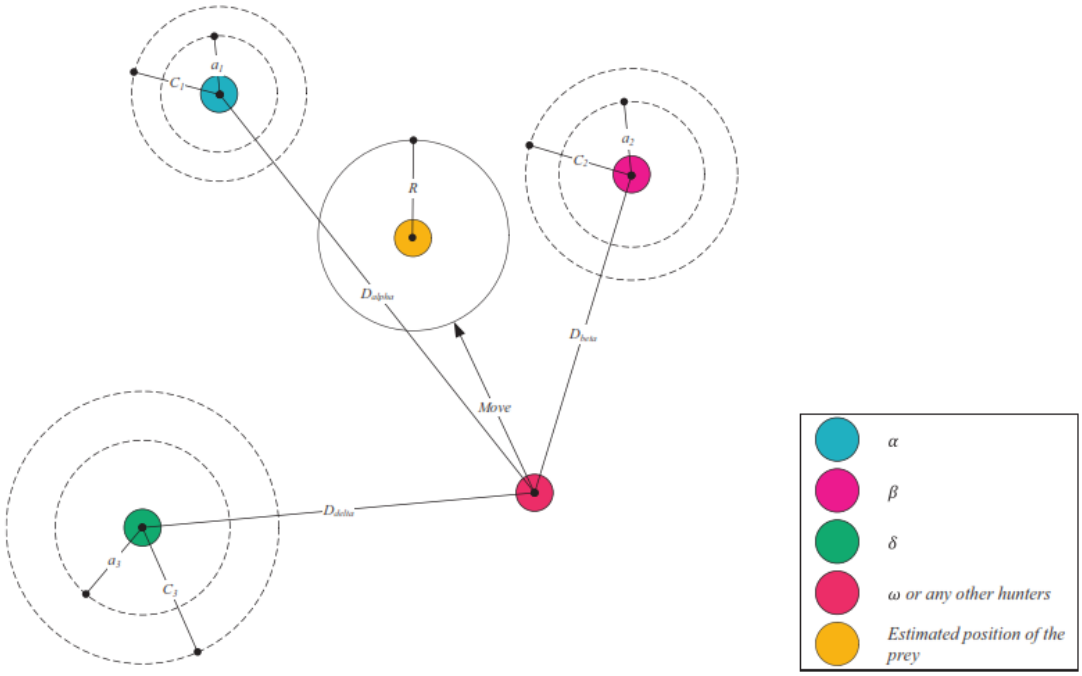
$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \vec{X}\right| \quad (۹-۲)$$

$$\overrightarrow{D_\beta} = \left|\overrightarrow{C_2}.\overrightarrow{X_\beta} - \vec{X}\right|$$

$$\overrightarrow{D_\delta} = \left|\overrightarrow{C_3}.\overrightarrow{X_\delta} - \vec{X}\right|$$

شکل ۲-۱۱ نشان می‌دهد که چگونه یک عامل جستجو موقعیت خود را براساس آلفا، بتا و دلتا در یک فضای جستجوی دو بعدی به روز می‌کند.

می‌توان مشاهده کرد که موقعیت نهایی در یک مکان تصادفی در یک دایره است که توسط موقعیت‌های آلفا، بتا و دلتا در فضای جستجو تعریف می‌شود. به عبارت دیگر آلفا، بتا و دلتا موقعیت طعمه را تخمین می‌زنند و دیگر گرگ‌ها موقعیت خود را به طور تصادفی در اطراف طعمه به روز می‌کنند.



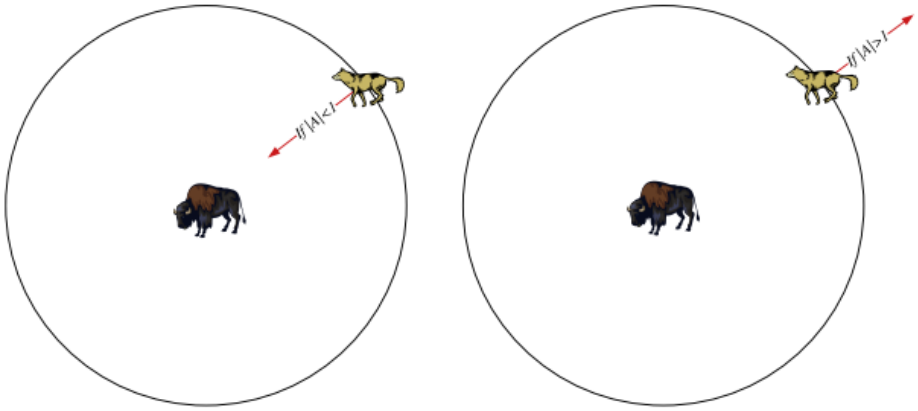
شکل ۲-۱۱: بروزرسانی مکان در GWO [۳۸]

۲-۵-۱-۲ مرحله بهره‌برداری[۲۶] (حمله به طعمه)

همانطور که در بالا گفته شد گرگ‌های خاکستری با حمله به طعمه هنگام توقف حرکت، شکار را انجام می‌دهند. به منظور مدل ریاضی نزدیک شدن به طعمه، مقدار $\vec{a}$ را کاهش داده می‌شود. توجه داشته باشید که دامنه نوسان $\vec{A}$ نیز با $\vec{a}$ کاهش می‌یابد. به عبارت دیگر $\vec{A}$ یک مقدار تصادفی در فاصله $[-2a.2a]$ است که در طول تکرار a از دو به صفر کاهش می‌یابد. هنگامی که مقادیر تصادفی $\vec{A}$ در [-۱،۱] است، موقعیت بعدی عامل جستجو می‌تواند در هر موقعیتی بین موقعیت فعلی آن و موقعیت طعمه باشد. شکل ۲-۱۲ نشان می‌دهد که $|A| < 1$ گرگ‌ها را وادار می‌کند که به طعمه حمله کنند. با اپراتورهایی که تاکنون پیشنهاد شده است، الگوریتم گرگ خاکستری به عوامل جستجوگر خود اجازه می‌دهد تا موقعیت خود را بر اساس موقعیت آلفا، بتا و دلتا به روز کنند و به سمت طعمه حمله کنند. با این حال، الگوریتم گرگ خاکستری در حال ثبت راه‌حل‌های محلی با این اپراتورها است. مکانیسم

[26] Exploitation

محاصره شده ارائه شده اکتشاف را تا حدی نشان می‌دهد، اما الگوریتم گرگ خاکستری برای تأکید بر اکتشاف به اپراتورهای بیشتری احتیاج دارد.



شکل ۲-۱۲: حمله به طعمه در مقابل جستجوی طعمه [۳۸]

۲-۵-۱-۳ مرحله اکتشاف[۲۷] (جستجوی طعمه)

گرگ‌ها معمولا به موقعیت آلفا، بتا و دلتا توجه می‌کنند. آن‌ها از یکدیگر فاصله می‌گیرند تا به دنبال طعمه بگردند و به سمت طعمه حمله کنند. به منظور مدل ریاضی واگرایی، از $\vec{A}$ با مقادیر تصادفی بزرگ‌تر از ۱ یا کم‌تر از ۱- استفاده می‌شود تا نماینده جستجو را وادار به انحراف از طعمه شود. این امر بر اکتشاف تاکید دارد و به الگوریتم گرگ خاکستری اجازه می‌دهد که اگر $|A| > 1$ باشد عمل جستجو به طور همگانی انجام شود.

قابلیت مقدار تطبیقی A موجب می‌شود، نیمی از تکرارها به اکتشاف $(|A| \geq 1)$ و بقیه به بهره‌برداری $(|A| < 1)$ اختصاص یابد. این مکانیزم به الگوریتم گرگ خاکستری کمک می‌کند تا اکتشاف بسیار خوب، اجتناب مینیمم محلی و بهره‌برداری را به طور همزمان فراهم کند. یک نکته نهایی در مورد بهینه‌ساز گرگ خاکستری به روزرسانی پارامتر a است که توازن بین اکتشاف و بهره‌برداری را کنترل می‌کند. پارامتر a به صورت خطی در هر تکرار از دو تا صفر به روز می‌شود که رابطه خطی آن به صورت معادله ۲-۱۰ می‌باشد:

[27] Exploration

$$a = 2 - t\frac{2}{MaxIter} \quad (۱۰-۲)$$

که در آن t تعداد تکرار و $MaxIter$ تعداد کل تکرار مجاز برای بهینه‌سازی است.

در شکل ۲-۱۳ شبه کد الگوریتم بهینه‌سازی پیوسته گرگ خاکستری نمایش داده شده است.

Input: n Number of gray wolves in the pack,
Number of iterations for optimization.N_{Iter}
Output: Optimal gray wolf position,x_α
Best fitness value.$f(x_\alpha)$

1. Initialize a population of n gray wolves positions randomly.
2. Find the $\alpha, \beta, and\ \delta$ solutions based on their fitness values.
3. While stopping criteria not met do
   For each $Wolf_i \in pack$ do
   Update current wolf's position
   End
   Update a, A and c. .I
   Evaluate the positions of individual wolves. .II
   Update $\alpha, \beta, and\ \delta$ .III
   End

شکل ۲-۱۳: الگوریتم بهینه‌سازی پیوسته گرگ خاکستری [۳۸]

۲-۵-۲ بهینه‌سازی گرگ خاکستری باینری

در گرگ خاکستری موقعیت مکان‌های گرگ در هر نقطه در فضای پیوسته واقع شده‌اند. در نتیجه معادلات به‌روزرسانی می‌توانند به راحتی اجرا شوند. در بهینه‌سازی گرگ خاکستری باینری، فضای جستجو یک ابرمکعب در نظر گرفته می‌شود و موقعیت گرگ‌ها تنها در مقادیر صفر یا یک قرار دارد. گرگ‌ها مکان خود را تغییر می‌دهند اما فقط در این ابرمکعب قادر به حرکت می‌باشند، بنابراین با استفاده از همان معادلات پیوسته نمی‌توان آن را به‌روز کرد. گرگ خاکستری باینری همان استراتژی را برای به دست آوردن ارزش‌های آلفا، بتا و دلتا دارد و از معادله ۲-۹ برای محاسبه $\overrightarrow{D_\alpha}$، $\overrightarrow{D_\beta}$ و $\overrightarrow{D_\delta}$ استفاده می‌کند. سپس با استفاده از توابع انتقال مانند S و V روابطی مانند زیر به دست می‌آید.

$$s_1^d = \frac{1}{1 + e^{-10(A^d.D_\alpha^d - 0.5)}} \quad (۱۱-۲)$$

$$s_2^d = \frac{1}{1+e^{-10(A^d.D_\beta^d-0.5)}} \quad (۱۲-۲)$$

$$s_3^d = \frac{1}{1+e^{-10(A^d.D_\delta^d-0.5)}} \quad (۱۳-۲)$$

که d امین بعد یک عامل (گرگ) است.

مقادیر $bstep_1^d$ ، $bstep_2^d$ و $bstep_3^d$ و با استفاده از معادلات ۲-۱۴ و ۲-۱۵ و ۲-۱۶ محاسبه می‌شوند. بعد از این مرحله، نتیجه یک مقدار باینری خواهد بود و دیگر یک مقدار پیوسته نخواهد بود. سپس از تابع انتقال برای تغییر استفاده می‌کند، همانطور که در معادلات ۲-۱۱ و ۲-۱۲ و ۲-۱۳ دیده شده است. مقادیر صفر و یک برای مقایسه با اعداد تصادفی مورد نیاز هستند.

$$bstep_1^d = \begin{cases} 1 & if\ (s_1^d \geq randn) \\ 0 & else \end{cases} \quad (۱۴-۲)$$

$$bstep_2^d = \begin{cases} 1 & if\ (s_2^d \geq randn) \\ 0 & else \end{cases} \quad (۱۵-۲)$$

$$bstep_3^d = \begin{cases} 1 & if\ (s_3^d \geq randn) \\ 0 & else \end{cases} \quad (۱۶-۲)$$

که در آن $randn$ یک عدد تصادفی بین [۰، ۱] است. $bstep_1^d$ ، $bstep_2^d$ و $bstep_3^d$ فواصلی هستند که گرگ i به سمت گرگ آلفا، بتا و دلتا حرکت می‌کند. سپس x_1، x_2 و x_3 با معادلات زیر محاسبه می‌شوند:

$$X_1^d = \begin{cases} 1 & if\ (X_\alpha^d + bstep_1^d \geq 1) \\ 0 & else \end{cases} \quad (۱۷-۲)$$

$$X_2^d = \begin{cases} 1 & if\ (X_\beta^d + bstep_2^d \geq 1) \\ 0 & else \end{cases} \quad (۱۸-۲)$$

$$X_3^d = \begin{cases} 1 & if\ (X_\delta^d + bstep_3^d \geq 1) \\ 0 & else \end{cases} \quad (۱۹-۲)$$

در پایان، یک تقاطع تصادفی ساده را برای به روز رسانی موقعیت X_i در تکرار بعدی، همانطور که در معادله ۲-۲۰ نشان داده شده است، اتخاذ می‌کند.

$$X_i^d(nt) = \begin{cases} X_1^d & if(rand < 1/3) \\ X_2^d & elseif(1/3 \leq rand < 2/3) \\ X_3^d & else \end{cases} \quad (۲۰-۲)$$

همانگونه که مشاهده می‌شود، تابع انتقال نقش کلیدی در تبدیل فضای پیوسته به باینری بازی می‌کند. لذا لازم است در ادامه بیشتر به این توابع پرداخته شود.

۲-۶ تابع ارزیابی

از بهینه‌سازی گرگ خاکستری باینری برای جستجوی تطبیقی فضای ویژگی برای بهترین ترکیب ویژگی استفاده می‌شود. بهترین ترکیب ویژگی، ترکیبی است که بیش‌ترین عملکرد طبقه‌بندی و کم‌ترین تعداد ویژگی‌های انتخاب شده را داشته باشد. تابع ارزیابی مورد استفاده در بهینه‌سازی گرگ خاکستری باینری برای ارزیابی موقعیت انفرادی گرگ خاکستری در معادله ۲-۲۱ نشان داده شده است.

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|C-R|}{|C|} \quad (۲۱-۲)$$

که در آن $\gamma_R(D)$ کیفیت طبقه‌بندی مجموعه ویژگی وضعیت R نسبت به تصمیم D، R طول زیرمجموعه ویژگی انتخابی، C تعداد کل ویژگی‌ها، α و β دو پارامتر متناظر با اهمیت کیفیت طبقه‌بندی و طول زیر مجموعه می‌باشد، α در بازه $[۰,۱]$ و $\beta = 1-\alpha$ هستند. می‌توانیم ببینیم $\gamma_R(D)$ تابع ارزیابی ماکزیمم کیفیت طبقه‌بندی و $\frac{|C-R|}{|C|}$ نسبت ویژگی‌های انتخاب نشده به تعداد کل ویژگی‌ها است. معادله بالا را می‌توان به سادگی به یک مساله کمینه‌سازی تبدیل کرد. مساله کمینه‌سازی را می‌توان مانند معادله ۲-۲۲ فرموله کرد.

$$Fitness = \alpha E_R(D) + \beta\frac{|R|}{|C|} \quad (۲۲-۲)$$

که در آن $E_R(D)$ میزان خطا برای طبقه‌بندی مجموعه ویژگی حالت، R طول زیرمجموعه ویژگی انتخابی و C تعداد کل ویژگی‌ها است. $\alpha \in [0,1]$ و $\beta = 1-\alpha$ ثوابت برای کنترل اهمیت دقت طبقه‌بندی و کاهش ویژگی هستند.

۲-۷ توابع انتقال

به گفته میرجلیلی و لوئیس در [۴۹]، یک تابع انتقال بخش مهمی در نسخه‌های باینری فراابتکاری‌ها است. این امر به طور قابل‌توجهی بر اجتناب از بهینه‌سازی محلی و تعادل بین اکتشاف و بهره‌برداری تاثیر می‌گذارد [۴۹]. تابع انتقال مقادیر پیوسته را به [۰، ۱] نگاشت می‌کند و سپس آن‌ها را با توجه به احتمال به صفر و یک مجزا می‌کند [۴۹،۵۰]. توابع انتقال، ساختار و دیگر عملیات الگوریتم را حفظ می‌کند و موقعیت‌های جمعیت را در یک فضای باینری حرکت می‌دهد.

برخی مفاهیم باید برای انتخاب یک تابع انتقال به منظور نگاشت مقادیر سرعت به مقادیر احتمال به صورت زیر در نظر گرفته شوند [۴۹]:

- محدوده تابع انتقال باید در بازه [۰، ۱] محدود شود، تا نشان‌دهنده احتمال تغییر موقعیت ذره باشد.
- یک تابع انتقال باید احتمال بالایی از تغییر موقعیت را برای یک مقدار مطلق بزرگ سرعت فراهم کند. ذراتی که مقادیر مطلق بزرگی برای سرعتشان دارند احتمالا از بهترین راه‌حل دور هستند، بنابراین باید موقعیت خود را در تکرار بعدی تغییر دهند.
- تابع انتقال همچنین باید احتمال کمی از تغییر موقعیت را برای یک مقدار مطلق کوچک سرعت ارائه دهد.
- مقدار بازگشت یک تابع انتقال باید با افزایش سرعت افزایش یابد. قسمت‌هایی که از بهترین راه‌حل دور می‌شوند، باید احتمال بیشتری برای تغییر بردارهای موقعیت خود به منظور بازگشت به موقعیت‌های قبلی خود داشته باشند.
- مقدار بازگشت یک تابع انتقال باید با کاهش سرعت کاهش یابد.

این مفاهیم تضمین می‌کنند که یک تابع انتقال قادر به نگاشت فرآیند جستجو از یک فضای جستجوی پیوسته به یک فضای جستجوی باینری است. عملکرد الگوریتم‌های فراابتکاری باینری ممکن است با

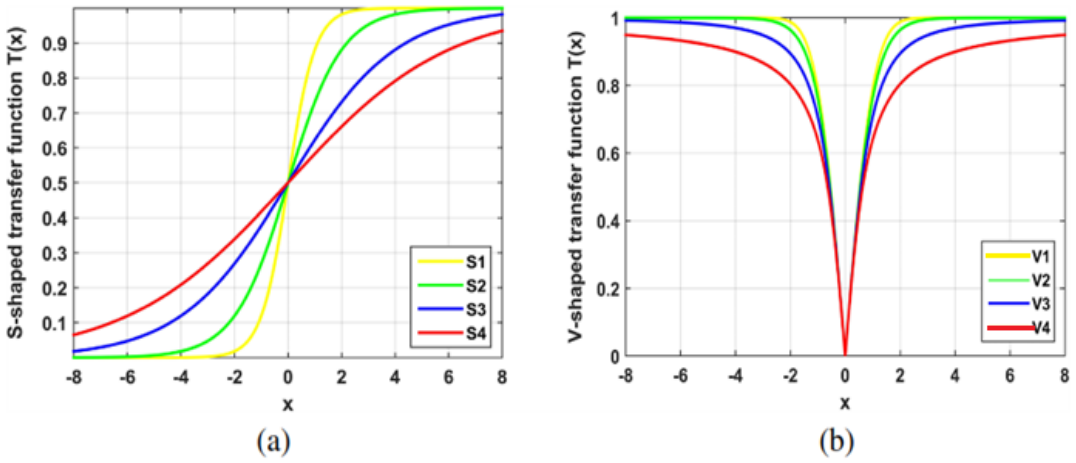
استفاده از یک تابع انتقال نامناسب کاهش یابد. بنابراین تابع انتقال نقش مهمی در عملکرد الگوریتم دارد و تابع انتقال باید به درستی انتخاب شود.

برای بهبود عملکرد الگوریتم های فراابتکاری باینری، الگوریتم‌های اصلی یا توابع انتقال در مقالات بهبود یافته‌اند [۵۳-۵۸]، با این حال، تابع انتقال نقشی کلیدی در کارایی الگوریتم‌های باینری ایفا می‌کند [۵۹،۶۰].

در ادامه به دو نوع متداول تابع انتقال S شکل [۴۹،۵۱] و V شکل [۴۹،۵۲] می‌پردازیم. فرمول‌ها و تصاویری از توابع انتقال S و V در جدول ۲-۶ و شکل ۲-۱۴ مشاهده می‌کنید.

جدول ۲-۶: توابع انتقال s شکل و v شکل [۴۹]

خانواده V شکل		خانواده S شکل	
تابع انتقال	نام تابع انتقال	تابع انتقال	نام تابع انتقال
$T(x) = \left\lvert \text{erf}\left(\frac{\sqrt{\pi}}{2}x\right)\right\rvert = \left\lvert \frac{\sqrt{2}}{\pi}\int_0^{\left(\sqrt{\pi}/2\right)x} e^{-t^2}dt \right\rvert$	V1	$T(x) = \frac{1}{1+e^{-2x}}$	S1
$T(x) = \lvert\tanh(x)\rvert$	V2	$T(x) = \frac{1}{1+e^{-x}}$	S2
$T(x) = \left\lvert \frac{(x)}{\sqrt{1+x^2}} \right\rvert$	V3	$T(x) = \frac{1}{1+e^{(-x/2)}}$	S3
$T(x) = \left\lvert \frac{2}{\pi} arc\tan\left(\frac{\pi}{2}x\right) \right\rvert$	V4	$T(x) = \frac{1}{1+e^{(-x/3)}}$	S4



شکل ۲-۱۴: توابع انتقال (a) S شکل و (b) V شکل [۴۹]

مشاهده می‌شود که S1 با افزایش سرعت به شدت افزایش می‌یابد و به اشباع خود می‌رسد و مقدار آن بسیار بیشتر از S2 افزایش می‌یابد، در حالی که اشباع S3 و S4 دیرتر از S2 شروع می‌شود. لازم به ذکر است که احتمال تغییر مقادیر بردارهای موقعیت با افزایش شیب این توابع انتقال افزایش می‌یابد، بنابراین بیش‌ترین احتمال را در میان آن‌ها برای همان مقدار سرعت باز می‌گرداند، در حالی که S4 کم‌ترین مقدار را فراهم می‌کند [۴۹]. با توجه به شکل ۲-۱۴ (a) ، این منحنی‌ها، آن‌ها توابع انتقال s شکل نامیده می‌شوند. در نتیجه، گروه آن‌ها خانواده توابع انتقال "S - شکل" نیز نامیده می‌شود.

در شکل ۲-۱۴ (b) ، توابع و گروه خانواده توابع انتقالی (V شکل) دیده می‌شوند. چهار تابع انتقال V شکل به نام‌های V1، V2، V3 و V4 با استفاده از معادلات ریاضی متنوع ارائه شده‌اند. این توابع انتقال در جدول ۲-۶ و شکل ۲-۱۴ (b) ، نشان داده شده‌اند. دیده شود که اشباع V1 با مقادیر مطلق پایین‌تر سرعت در مقایسه با V2 شروع می‌شود. این امر باعث می‌شود که V1 بتواند احتمال بالاتری از تغییر موقعیت ذرات را نسبت به V2 برای همان سرعت فراهم کند. در مقابل، اشباع توابع انتقال V3 و V4 پس از V1 و V2 آغاز می‌شود که احتمال تغییر کمتری را برای سرعت‌های مشابه فراهم می‌کند. فارغ از فرم ریاضی توابع V شکل، این شیب شکل است که نقش اصلی را بازی می‌کند. در شیب کوچک‌تر موقعیت ذره به آرامی تغییر می‌کند و با افزایش شیب تابع انتقال، سرعت سوییچینگ موقعیت افزایش می‌یابد. به عبارت دیگر تابع انتقال سرعت تغییر موقعیت را کنترل می‌کند.

۲-۸ توابع انتقال S و V در الگوریتم بهینه‌سازی ازدحام ذرات باینری

در سال ۲۰۱۳، میرجلیلی و لوئیس از توابع انتقال S و V برای الگوریتم بهینه‌سازی ازدحام ذرات باینری[۲۸] استفاده کردند. آن‌ها در [۴۹] آورده‌اند که بهینه‌سازی ازدحام ذرات یک تکنیک محاسبه تکاملی است که از رفتار اجتماعی پرنده الهام‌گرفته شده و توسط کندی و ابرهارت ارائه شده است [۵۱].

[28] Binary Particle Swarm Optimization

این الگوریتم از تعدادی ذره (راه‌حل‌های کاندید) استفاده می‌کند که در فضای جستجو به پرواز در می‌آیند تا بهترین راه‌حل را پیدا کنند. در همین حال، همه آن‌ها بهترین مکان (بهترین راه‌حل) را در مسیرهای خود ردیابی می‌کنند. به عبارت دیگر، ذرات بهترین راه‌حل خود و همچنین بهترین راه حلی که جمعا تاکنون به دست آورده است را در نظر می‌گیرند. هر ذره در روش بهینه‌سازی ازدحام ذرات باید موقعیت فعلی، سرعت جریان، فاصله تا بهترین راه‌حل شخصی و بهترین راه‌حل کلی را برای اصلاح موقعیت خود در نظر بگیرد که از روابط زیر استفاده می‌شود:

$$v_i^{t+1} = wv_i^t + c_1 * rand * (pbest_i - x_i^t) + c_2 * rand * (gbest - x_i^t) \quad (۲۳-۲)$$

$$x_i^{t+1} = v_i^t + v_i^{t+1} \quad (۲۴-۲)$$

که در آن v_i^t سرعت ذره i در تکرار t، w یک تابع وزنی، c_i ضریب شتاب، $rand$ یک عدد تصادفی بین صفر و یک است، x_i^t موقعیت فعلی ذره i در تکرار t، $pbest_i$ بهترین راه حلی انفرادی است و $gbest$ بهترین راه حل کلی است که جمعیت تاکنون به دست آورده است.

میرجلیلی و لوئیس برای تبدیل فضای پیوسته به باینری از مفهوم احتمال کمک گرفتند و از تابع سیگموئید زیر استفاده کردند [۶۱].

$$T\left(v_i^k(t)\right) = \frac{1}{1 + e^{-v_i^k(t)}} \quad (۲۵-۲)$$

که در آن $v_i^k(t)$ سرعت ذرهi در تکرار t در بعد k ام را نشان می‌دهد.

پس از تبدیل سرعت‌ها به مقادیر احتمال، بردارهای موقعیت می‌توانند با احتمال سرعتشان به صورت زیر به‌روز شوند:

$$X_i^k(t+1) = \begin{cases} 0 & if\ rand < T(v_i^k(t+1)) \\ 1 & if\ rand \geq T(v_i^k(t+1)) \end{cases} \quad (۲۶-۲)$$

اما میرجلیلی از تابع انتقال V نیز استفاده کرد که به چهار شکل زیر ارائه شده است.

جدول ۲-۷: توابع انتقال V شکل [۳۸]

خانواده V شکل	
نام تابع انتقال	**تابع انتقال**
V1	$\boldsymbol{T(x) = \left\lvert \mathbf{erf}\ (\frac{\sqrt{\pi}}{2}x) \right\rvert = \left\lvert \frac{\sqrt{2}}{\pi} \int_0^{(\sqrt{\pi}/2)x} e^{-t^2} dt \right\rvert}$
V2	$\boldsymbol{T(x) = \lvert \tanh(x) \rvert}$
V3	$\boldsymbol{T(x) = \left\lvert \frac{(x)}{\sqrt{1+x^2}} \right\rvert}$
V4	$\boldsymbol{T(x) = \left\lvert \frac{2}{\pi} arc\ \tan\ (\frac{\pi}{2}x) \right\rvert}$

برای این توابع انتقال، قوانین به روزرسانی موقعیت متفاوت می‌باشد و معادله ۲-۲۷ به شکل زیر تغییر می‌کند.

$$x_i^k(t+1) = \begin{cases} (x_i^k(t))^{-1} & if\ rand < T(v_i^k(t+1)) \\ x_i^k(t) & if\ rand \geq T(v_i^k(t+1)) \end{cases} \quad (۲۷-۲)$$

آن‌ها نشان دادند که خانواده V شکل از توابع انتقال S به طور قابل‌توجهی عملکرد بهینه‌سازی ازدحام ذرات باینری را بهبود می‌بخشد [۳۸].

۲-۹ تابع انتقال V در الگوریتم خفاش باینری

میرجلیلی و همکارانش در سال ۲۰۱۴ [۶۲]، از تابع $arctan$ خانواده تابع انتقال V شکل برای ارائه الگوریتم باینری خفاش بهره بردند. الگوریتم خفاش[۲۹] یکی از الگوریتم‌های فراابتکاری است که از رفتار مکان‌یابی خفاش‌ها برای انجام بهینه‌سازی تقلید می‌کند [۶۳]. خفاش‌ها از سونار طبیعی برای انجام این کار استفاده می‌کنند. دو ویژگی اصلی خفاش‌ها در هنگام یافتن طعمه در طراحی الگوریتم خفاش اتخاذ شده است. خفاش‌ها هنگام تعقیب طعمه تمایل به کاهش بلندی صدا و افزایش سرعت صدای

[29] Bat Algorithm

اولتراسونیک منتشر شده دارند. این رفتار به صورت ریاضی به صورت زیر مدل‌سازی شده است. بردارهای موقعیت، بردار سرعت و بردار فرکانس خفاش در طول دوره تکرار به صورت زیر به روز می‌شوند:

$$v_i(t+1) = v_i(t) + (x_i(t) - Gbest)F_i \quad (۲۸-۲)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (۲۹-۲)$$

که در آن $Gbest$ بهترین راه‌حل به‌دست‌آمده تاکنون است و F_i فرکانس خفاش i ام را نشان می‌دهد که در هر دوره از تکرار به صورت زیر به روز می‌شود:

$$F_i = F_{min} + (F_{max} - F_{min})\beta \quad (۳۰-۲)$$

که در آن β یک عدد تصادفی از یک توزیع یکنواخت در $[۰،۱]$ است.

توجه داشته باشید که هم بلندی و هم سرعت، زمانی به روز می‌شوند تا راه‌حل‌های جدید بهبود یابند و تضمین شود که خفاش‌ها به سمت بهترین راه‌حل حرکت می‌کنند. جهت باینری کردن الگوریتم خفاش به الگوریتم خفاش باینری از تابع انتقال $arctan$ از خانواده V شکل استفاده گردید.

نتایج بدست آمده از الگوریتم پیشنهادی الگوریتم خفاش باینری در مقایسه با بهینه‌سازی ازدحام ذرات باینری و ژنتیک نشان داد که الگوریتم خفاش باینری شایستگی بیشتری دارد.

۲-۱۰ تابع انتقال S در الگوریتم گرگ خاکستری باینری

در ادامه میرجلیلی و همکارانش در سال ۲۰۱۴ [۳۸]، الگوریتم گرگ خاکستری را معرفی کردند و در سال ۲۰۱۵، Emary و همکارانش الگوریتم گرگ خاکستری باینری را همراه تابع انتقال S ارائه دادند [۷]. از آنجا که الگوریتم گرگ خاکستری، الگوریتم مورد بحث این پایان‌نامه می‌باشد و توضیح آن در بخش قبلی داده شده است از توضیح مکرر آن پرهیز می‌شود.

نسخه باینری معرفی‌شده در مقاله Emary و همکارانش با استفاده از دو رویکرد مختلف اجرا شده است. در رویکرد اول معادله به روزرسانی اصلی در معادله ۲-۳۱ نشان داده شده است.

$$X_i^{t+1} = Crossover(x_1, x_2, x_3) \quad (۳۱-۲)$$

$$X_1^d = \begin{cases} 1 & if\ (X_\propto^d + bstep_\alpha^d \geq 1) \\ 0 & else \end{cases} \quad (۳۲-۲)$$

که در آن $X_\propto^d$ بردار موقعیت گرگ آلفا در بعد d و $bstep_\alpha^d$ یک گام باینری در بعد d است که می‌توان آن را مانند رابطه ۲-۳۳ محاسبه کرد.

$$bstep_\alpha^d = \begin{cases} 1 & if\ (cstep_\alpha^d \geq rand) \\ 0 & else \end{cases} \quad (۳۳-۲)$$

که در آن rand یک عدد تصادفی است که در $[۰،۱]$ توزیع یکنواخت شده است و $cstep_\alpha^d$ اندازه گام با مقدار پیوسته برای بعد d است و می‌تواند با استفاده از تابع سیگموئیدی مانند معادله ۲-۳۴ محاسبه شود.

$$cstep_\alpha^d = \frac{1}{1 + e^{-10(A_1^d . D_\propto^d - 0.5)}} \quad (۳۴-۲)$$

که در آن A_1^d و $D_\propto^d$ با استفاده از معادلات ۲-۵ و ۲-۹ در بعد d محاسبه می‌شوند.

$$X_2^d = \begin{cases} 1 & if\ (X_\beta^d + bstep_2^d \geq 1) \\ 0 & else \end{cases} \quad (۳۵-۲)$$

که در آن X_β^d بردار موقعیت گرگ بتا در بعد d و $bstep_\beta^d$ گام باینری در بعد d است که می‌توان آن را مانند رابطه ۲-۳۶ محاسبه کرد.

$$bstep_\beta^d = \begin{cases} 1 & if\ (cstep_\beta^d \geq rand) \\ 0 & else \end{cases} \quad (۳۶-۲)$$

که در آن $rand$ یک عدد تصادفی است که از $[۰،۱]$ توزیع یکنواخت گرفته شده است و $cstep_\beta^d$ اندازه گام با مقدار پیوسته برای بعد d است و می‌تواند با استفاده از تابع سیگموئیدی مانند معادله ۲-۳۷ محاسبه شود.

$$cstep_\beta^d = \frac{1}{1 + e^{-10(A_1^d . D_\beta^d - 0.5)}} \quad (۳۷-۲)$$

که در آن A_1^d و D_β^d با استفاده از معادلات ۲-۵ و ۲-۹ در بعد d محاسبه می‌شوند.

$$X_3^d = \begin{cases} 1 & if\ (X_\delta^d + bstep_\delta^d \geq 1) \\ 0 & else \end{cases} \quad (۳۸-۲)$$

که در آن X_δ^d بردار موقعیت گرگ دلتا در بعد، d و $bstep_\delta^d$ گام باینری در بعد d است که می‌توان آن را مانند رابطه ۲-۳۹ محاسبه کرد.

$$bstep_\delta^d = \begin{cases} 1 & if\ (cstep_\delta^d \geq rand) \\ 0 & else \end{cases} \quad (۳۹-۲)$$

که در آن $rand$ یک عدد تصادفی است که از $[۰،۱]$ توزیع یکنواخت گرفته شده است و $cstep_\delta^d$ اندازه گام با مقدار پیوسته برای بعد d است و می‌تواند با استفاده از تابع سیگموئیدی مانند معادله ۲-۴۰ محاسبه شود.

$$cstep_\delta^d = \frac{1}{1 + e^{-10(A_1^d.D_\delta^d - 0.5)}} \quad (۴۰-۲)$$

که در آن A_1^d و D_δ^d با استفاده از معادلات ۲-۵ و ۲-۹ در بعد d محاسبه می‌شوند.

همانطور که در معادله ۲-۴۱ نشان داده شده است، یک استراتژی crossover به ازای هر بعد برای راه‌حل‌های a، b، c اعمال می‌شود.

$$x_d = \begin{cases} a_d & if\ (rand < {}^1/_3) \\ b_d & elseif\ ({}^1/_3 \leq rand < {}^2/_3) \\ c_d & else \end{cases} \quad (۴۱-۲)$$

که در آن a_d ، b_d و c_d مقادیر باینری برای اولین، دومین و سومین پارامتر در بعد d، x_d خروجی $crossover$ در بعد d و $rand$ یک عدد تصادفی است که از توزیع یکنواخت در دامنه $[۰،۱]$ گرفته شده است.

در رویکرد دوم تنها بردار موقعیت گرگ خاکستری به روز شده مجبور به باینری شدن می‌شود. معادله به روزرسانی اصلی در رابطه ۲-۴۲ نشان داده شده است.

$$X_d^{t+1} = \begin{cases} 1 & if\ sigmoid(\frac{x_1 + x_2 + x_3}{3}) \geq rand \\ 0 & else \end{cases} \quad (۴۲-۲)$$

که در آن $rand$ یک عدد تصادفی است که از $[۰،۱]$ توزیع یکنواخت، X_d^{t+1} به روز شده موقعیت باینری در بعد d در تکرار t و سیگموئید (a) در رابطه ۲-۴۳ است.

$$sigmoid(a) = \frac{1}{1 + e^{-10(x-0.5)}} \quad (۴۳-۲)$$

Emary و همکارانش از تابع انتقال S برای باینری کردن الگوریتم گرگ خاکستری استفاده کردند. ثابت کردند که شکل باینری الگوریتم گرگ خاکستری عملکردی بهتر از الگوریتم بهینه‌سازی ازدحام ذرات و الگوریتم ژنتیک از خود نشان می‌دهند.

۲-۱۱ توابع انتقال S و V در الگوریتم سنجاقک باینری

میرجلیلی در سال ۲۰۱۵ [۶۴]، الگوریتم سنجاقک[30] را پیشنهاد داد و برای باینری کردن آن از دو تابع سیگموئید و تابع V (تابع نوع V3 در این مقاله) استفاده کرد.

بنا به گفته رینولدز، رفتار گروها از سه اصل ابتدایی پیروی می‌کند [۹،۵۲]. جداسازی، که به اجتناب از برخورد استاتیک افراد از افراد دیگر در همسایگی اشاره دارد و هم ترازی، که انطباق سرعت افراد با افراد دیگر در همسایگی را نشان می‌دهد و انسجام که به تمایل افراد به سمت مرکز گروه همسایگی اشاره دارد.

هدف اصلی هر گروه زنده ماندن است، بنابراین تمام افراد باید به سمت منابع غذایی جذب شوند و از دشمنان خارجی دور شوند.

با در نظر گرفتن این دو رفتار، پنج عامل اصلی در به روزرسانی موقعیت سنجاقک‌ها به صورت گروهی وجود دارد. هر یک از این رفتارها از نظر ریاضی به صورت زیر مدل‌سازی شده‌اند:

$$s_i = -\sum_{j=1}^{N} x - x_j \quad (۴۴-۲)$$

که X موقعیت فرد فعلی است، x_j موقعیت فرد همسایه j ام را نشان می‌دهد و N تعداد افراد همسایه است.

$$A_i = \frac{\sum_{j=1}^{N} v_j}{N} \quad (۴۵-۲)$$

[30] Dragonfly algorithm

که در آن v_j سرعت فرد همسایه j ام را نشان می‌دهد.

همبستگی به صورت زیر محاسبه می‌شود:

$$C_i = \frac{\sum_{j=1}^{N} x_j}{N} - x \quad (۴۶-۲)$$

که X موقعیت فرد فعلی، N تعداد همسایگی‌ها، و x_j موقعیت فرد همسایه j ام را نشان می‌دهد.

جذب به یک منبع غذایی به صورت زیر محاسبه می‌شود.

$$F_i = x^+ - x \quad (۴۷-۲)$$

که در آن x موقعیت فرد فعلی است، و x^+ موقعیت منبع غذایی را نشان می‌دهد.

دور شدن از دشمن به صورت زیر محاسبه می‌شود:

$$E_i = x^- + x \quad (۴۸-۲)$$

که x موقعیت فرد فعلی است و x^- موقعیت دشمن را نشان می‌دهد.

بردار ΔX جهت حرکت سنجاقک‌ها را نشان می‌دهد و به صورت زیر تعریف می‌شود.

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \quad (۴۹-۲)$$

که در آن s وزن جداسازی را نشان می‌دهد، S_i که نشان‌دهنده جداسازی فرد i ام، a وزن هم ترازی، A_i همترازی فرد i ام، c وزن همبستگی، c_i پیوستگی فرد i ام، f فاکتور غذایی، f_i منبع غذایی فرد iام، e فاکتور دشمن، E_i موقعیت دشمن از فرد i ام، w وزن اینرسی است.

پس از محاسبه بردار جهت حرکت، بردارهای موقعیت به صورت زیر محاسبه می‌شوند:

$$X_{t+1} = X_t + \Delta X_{t+1} \quad (۵۰-۲)$$

که در آن t تکرار فعلی است.

همسایگان سنجاقک‌ها بسیار مهم هستند، بنابراین یک همسایگی (دایره در یک فضای دوبعدی، کره در یک فضای سه‌بعدی، یا ابرکره در یک فضای) با شعاع مشخص در اطراف هر سنجاقک مصنوعی فرض می‌شود.

هنگام اکتشاف همراستایی کم و همبستگی بالا و در هنگام بهره‌برداری، به سنجاق‌هایی با همراستایی بالا و وزن همبستگی پایین اختصاص داده می‌شود. برای انتقال بین اکتشاف و بهره‌برداری، شعاع همسایگی‌ها متناسب با تعداد تکرارها افزایش می‌یابد. روش دیگر برای برقراری تعادل در اکتشاف و بهره‌برداری، تنظیم سازگار ضرایب جمعیت (w ،e ،f ،c ،a ،s) در طول بهینه‌سازی است.

برای تبدیل یک تکنیک هوش ازدحام پیوسته به یک الگوریتم باینری بدون تغییر ساختار، استفاده از یک تابع انتقال است. آن‌ها با استفاده از دو نوع تابع انتقال S شکل و V شکل اقدام به این کار کردند [۶۶]. به گفته سارمی و همکارانش [۶۶]، توابع انتقال V شکل بهتر از توابع انتقال S شکل هستند چون ذرات را مجبور به گرفتن مقادیر صفر یا یک نمی‌کنند. الگوریتم سنجاقک باینری، تابع انتقال زیر مورد استفاده قرار داده است [۴۹].

$$T(\Delta x) = \left|\frac{\Delta x}{\sqrt{\Delta x^2 + 1}}\right| \qquad (۵۱-۲)$$

این تابع انتقال ابتدا برای محاسبه احتمال تغییر موقعیت برای تمام سنجاقک‌ها مورد استفاده قرار می‌گیرد. سپس فرمول بهنگام‌سازی موقعیت برای به روزرسانی موقعیت عامل‌های جستجو در فضای جستجوی باینری به کار گرفته می‌شود:

$$X_{t+1} = \begin{cases} \neg X_t & r < T(\Delta x_{t+1}) \\ X_t & r \geq T(\Delta x_{t+1}) \end{cases} \qquad (۵۲-۲)$$

که در آن r عددی در فاصله $[۰،۱]$ است.

با تابع انتقال و معادلات به روزرسانی موقعیت جدید، الگوریتم سنجاقک باینری قادر خواهد بود تا مسائل باینری را به راحتی با در نظر گرفتن فرمول‌بندی مناسب مساله حل کند.

جهت بررسی نتایج، سه گروه از توابع محک (توابع تک نمایی، چند نمایی و مرکب) با ویژگی‌های مختلف برای محک زدن عملکرد الگوریتم سنجاقک از دیدگاه‌های مختلف استفاده شد. این نتایج ثابت می‌کند که الگوریتم سنجاقک بهتر از الگوریتم ژنتیک و الگوریتم سنجاقک باینری اکتشاف و بهره‌برداری بالایی را از الگوریتم سنجاقک به دلیل استفاده از تابع انتقال V شکل دارا می‌باشد.

۲-۱۲ تابع انتقال V در الگوریتم وال کوهان‌دار باینری

بار دیگر میرجلیلی در سال ۲۰۱۶ [۶۷]، الگوریتم دیگری به نام الگوریتم وال کوهان‌دار[31] را با الهام از رفتار وال‌ها پیشنهاد کردند و در سال ۲۰۱۷، Abdelazim و همکاران از تابع انتقال قدرمطلق تانژانت هیپربولیک برای باینری کردن الگوریتم وال کوهان‌دار استفاده کردند [۶۸].

جالب‌ترین چیز در مورد وال‌های گوژپشت، روش شکار ویژه آن‌ها است. این رفتار کاوشی تغذیه حباب تله نامیده می‌شود. از نظر ریاضی، الگوریتم وال کوهان‌دار رفتار جمعیت وال‌ها را به صورت زیر شبیه‌سازی می‌کند:

$$\vec{D} = \left|\vec{C}.\vec{X}^*(t) - \vec{X}(t)\right| \quad (۵۳-۲)$$

$$\vec{X}(t+1) = \vec{X}^*(t+1) - \vec{A}.\vec{D} \quad (۵۴-۲)$$

هنگامی که t شماره تکرار فعلی باشد، X بردار موقعیت است، X^* بردار موقعیتی است که با بهترین راه‌حل یافت شده مطابقت داده می‌شود، A و C بردارهای ضرایب هستند. A و C در دو معادله زیر تعریف شده‌اند:

$$\vec{A} = 2\vec{a}.\vec{\mathrm{r}} - \vec{a} \quad (۵۵-۲)$$

$$\vec{C} = 2\vec{r} \quad (۵۶-۲)$$

که در آن r به صورت تصادفی در محدوده $[۰،۱]$ قرار می‌گیرد و a به صورت خطی از دو به صفر در طی تکرارها کاهش می‌یابد. این الگوریتم مانند بسیاری از الگوریتم‌های بهینه‌سازی دارای دو مرحله است: اکتشاف و بهره‌برداری.

در مرحله اکتشاف، عوامل جستجو تشویق می‌شوند تا برای جستجو یا اکتشاف مناطق مختلف در فضای جستجو حرکت کنند. در حالی که در مرحله بهره‌برداری، عوامل به صورت محلی حرکت می‌کنند تا راه‌حل‌های فعلی

را بهبود بخشند.

[31] Whale Optimization Algorithm

مرحله بهره‌برداری به دو مرحله تقسیم می‌شود:

۱) مکانیزم کوچک شدن: این مساله را می‌توان با معادله ۲-۵۶ بدست آورد.

۲) موقعیت به روز رسانی حلزونی: این روش فاصله بین وال و طعمه را محاسبه می‌کند. معادله آن برای تقلید حرکت مارپیچ به صورت معادله ۲-۵۷ به کار می‌رود:

$$\vec{X}(t+1) = \vec{D}^{l} e^{bl}.\cos(2\pi l)) + .\vec{\mathrm{X}}^{*}(t) \qquad (۵۷-۲)$$

که در آن l یک عدد تصادفی در محدوده [۱،۱-] و b یک ثابت است. یک احتمال ۵۰ ٪ برای انتخاب بین مکانیزم جمع شدگی در نظر گرفته شده است. در نتیجه، مدل ریاضی به شرح معادله ۲-۵۸ است:

$$\vec{X}(t+1) = \begin{cases} \vec{\mathrm{X}}^{*}(t) - \vec{A}.\vec{D} & if\ p < 0.5 \\ \vec{D}^{l} e^{bl}.\cos(2\pi l)) + .\vec{\mathrm{X}}^{*}(t) & if\ p \geq 0.5 \end{cases} \qquad (۵۸-۲)$$

که در آن p یک عدد تصادفی در یک توزیع یکنواخت است.

مرحله اکتشاف: از طرف دیگر، در مرحله اکتشاف، A از مقادیر تصادفی در محدوده $-۱ < A < ۱$ استفاده کرده است تا عامل را مجبور کند تا از این مکان دور شود و از نظر ریاضی به صورت زیر فرمول‌بندی شده است:

$$\vec{D} = \left|\vec{C}.\vec{X}_{rand} - \vec{x}\right| \qquad (۵۹-۲)$$

$$\vec{X}(t+1) = X_{rand} - \vec{A}.\vec{D} \qquad (۶۰-۲)$$

در الگوریتم پیشنهادی، تابع انتقال تانژانت هیپربولیک از توابع V شکل مطابق با معادلات ۲-۶۱ و ۲-۶۲ ارائه شده است.

$$y^{k} = |tanh x^{k}| \qquad (۶۱-۲)$$

$$X_{i}^{d} = \begin{cases} 0 & if\ rand < S(x_{i}^{k}(t+1)) \\ 1 & else \end{cases} \qquad (۶۲-۲)$$

به منظور تبدیل نسخه اصلی الگوریتم وال کوهان‌دار به نسخه باینری، توابع انتقال V شکل به کار گرفته می‌شوند. برای بررسی عملکرد BWOA - V، آزمایش‌ها بر روی ۱۱ مجموعه داده معیار از مجموعه داده UCI اعمال شده است و علاوه بر این، پنج معیار ارزیابی و همچنین آزمون آماری غیرپارامتری رتبه - مجموع ویلکوکس برای ارزیابی جنبه‌های مختلف الگوریتم‌های مقایسه شده انجام شد و نتایج آزمایش‌ها

ثابت کرد که الگوریتم باینری پیشنهادی قادر به حداقل رساندن تعداد ویژگی‌های انتخاب شده و همچنین به حداکثر رساندن دقت طبقه‌بندی در دامنه انتخاب ویژگی می‌باشد [۶۹].

۲-۱۳ توابع انتقال S و V در الگوریتم گرگ خاکستری باینری

Pei Hu و همکارانش در سال ۲۰۲۰، از توابع انتقال S و V برای الگوریتم گرگ خاکستری استفاده کردند و با اصلاحاتی در برخی از توابع انتقال V شکل، باینری کردن الگوریتم گرگ خاکستری را بهینه‌تر کردند [۵۸].

در جدول ۲-۸ اصلاحات مورد نظر دیده می‌شود:

جدول ۲-۸: جزئیات توابع انتقال بهبود یافته [۴۹]

نام تابع انتقال	تابع انتقال
$V_{2a}(x)$	$\lvert\tanh(x)/\tanh(4)\rvert$
$V_{3a}(x)$	$\left\lvert\sqrt{17}*x/(4*\sqrt{1\_+x^2})\right\rvert$
$V_{4a}(x)$	$\left\lvert\arctan(\frac{2}{\pi}x)/\arctan(2\pi)\right\rvert$

در واقع هر تابع انتقالی به ماکزیمم مقدار خود تقسیم شده، تا آن را نرمالایز کند، زیرا در الگوریتم گرگ خاکستری باینری ، مقدار x در محدوده [۴،۴-] است و مقادیر بیشینه V1(x) ، V2(x)،V3(x)و V4(x) به ترتیب ۱/۰۰۰۰، ۰/۹۹۹۳، ۰/۹۷۰۱ و ۰/۸۹۵۵ هستند [۵۹]. هدف اصلی داشتن احتمال بالا تا یک شود. اکنون حتی اگر x به مقدار بیشینه برسد، V2(x)،V3(x)و V4(x) احتمال ۰/۰۷٪، ۲/۹۹ ٪ و ۱۰/۴۵ ٪ دارند که به یک نرسیده‌اند و این در تضاد با هدف اصلی است [۵۹]. به منظور جلوگیری از این وضعیت، توابع انتقال برای حالت خاص الگوریتم گرگ خاکستری باینری نرمالایز می‌شوند.

تغییر دیگری که در الگوریتم پایه داده شد رابطه مربوط به مقدار a می‌باشد و به صورت معادله ۲-۶۳ ارائه شد:

$$a = 2 * {it}/{MAX\_IT} \qquad (۶۳-۲)$$

از معادله ۲-۶۳، یک افزایش خطی از صفر به دو خواهد داشت. در شروع الگوریتم، a یک مقدار عددی کوچک است. این بدان معنی است که مقدار x ، (AD) کوچک است و الگوریتم گرگ خاکستری باینری احتمال زیادی برای تغییر موقعیت دارد. با تکامل الگوریتم، a بزرگ می‌شود و AD احتمال زیادی برای به دست آوردن مقدار بیشینه دارد. از این رو، الگوریتم گرگ خاکستری باینری احتمال کمی برای تغییر موقعیت دارد. این یک تعادل عالی بین اکتشاف و بهره‌برداری است.

توانایی‌های کلی اکتشاف و بهره‌برداری از روش‌های پیشنهادی توسط ۲۹ تابع معیار مورد بررسی قرار گرفت. توابع انتقال با توجه به محدوده مقادیر AD بهبود یافت. این روش‌ها عملکرد قابل‌توجهی در توابع بنچ مارک نشان می‌دهند و هم‌گرایی سریعی دارند.

۲-۱۴ نتیجه‌گیری

در این فصل به بیان مفاهیم اولیه لازم جهت درک پایان‌نامه پرداخته شده است و در ادامه به تشریح طبقه‌بندی و مفاهیم انتخاب ویژگی و توضیح انواع آن و به بیان الگوریتم‌های فراابتکاری و توضیح الگوریتم گرگ خاکستری و به شرح توابع انتقال و نقشی که در الگوریتم اجراء می‌کند پرداخته شد.

توابع انتقال، ساختار و دیگر عملیات الگوریتم را حفظ می‌کند و موقعیت‌های جمعیت را در یک فضای باینری حرکت می‌دهد. توابع انتقال بخش کلیدی الگوریتم‌های فراابتکاری باینری محسوب می‌شوند. با توجه بکارگیری آسان آن‌ها برای کنترل مراحل اکتشاف و بهره‌برداری در الگوریتم، محققان به دفعات از این توابع استفاده کرده‌اند.

فصل ۳: روش تحقیق

۳-۱ مقدمه

در مقالات الگوریتم فرابتکاری باینری پیشین، غالبا یکی از توابع انتقال بکار رفته است. این سوال می‌تواند مطرح شود، چرا در این پایان‌نامه از این تابع انتقال استفاده شده است و چرا از نوع دیگر استفاده نشده است و یا اگر بجای این تابع انتقال از تابع انتقال دیگری استفاده می‌شد، نتیجه بهتری به دست نمی‌آمد؟

روش پیشنهادی برای حل این چالش، الگوریتمی را پیاده کرده است که در حین یادگیری، در مورد انتخاب تابع انتقال نیز تصمیم گیری می‌کند. به این شکل، تابع انتقال تابعی از شرایط تحمیلی نخواهد بود.

۳-۲ روش پیشنهادی

در این تحقیق از ده مجموعه داده UCI، استفاده شده است. بعد از انتخاب یکی از مجموعه داده‌های UCI، اقدام به استخراج ویژگی می‌گردد. سپس n بیت باینری به ویژگی‌های آن اضافه می‌شود. از این بیت‌ها ملاکی برای انتخاب تابع انتقال و یا شیب تابع انتقال که از قبل تعیین شده است برای هر گرگ استفاده می‌شود. به عبارت دیگر، الگوریتم ۲^n حالت برای انتخاب تابع انتقال و یا شیب تابع انتقال در اختیار دارد. بنابراین هر گرگ تابع انتقال و شیب مختص خود را دارد. در طول اجرای الگوریتم با توجه به تابع ارزیابی و با توجه به یادگیری، که حین اجرای الگوریتم انجام می‌شود، گرگ‌ها موقعیت خود را بروز می‌کنند. بعد از هر تکرار الگوریتم، موقعیت گرگ آلفا بروزرسانی و تابع انتقال انتخاب می‌گردد. با تکرارهای بعدی، الگوریتم ضمن انتخاب تابع انتقال مناسب با شیب مناسب، یادگیری تابع انتقال را نیز انجام می‌دهد، تا با کمترین خطا، تعداد ویژگی مناسب به بهترین نحو بدست آید.

روش پیشنهادی در این بخش همانطور که در بالا شرح داده شد ارائه روشی جدید و بکر برای یادگیری تابع انتقال و انتخاب شیب مناسب در حین اجرای الگوریتم می‌باشد که انتخاب ویژگی را محقق می‌کند.

۳-۳ ارزیابی و تابع شایستگی

در فرآیند یادگیری ماشین، نیاز به یک مدل است و این مدل باید به نوعی ارزیابی گردد. جهت این کار مدل به وسیله بخشی از داده‌ها آموزش داده می‌شود و به وسیله بخشی دیگر از داده‌ها ارزیابی می‌گردد تا مشخص شود مدل تا چه اندازه در طبقه‌بندی موفق عمل می‌کند.

در اعتبارسنجی متقابل $K-fold$ ، نمونه اصلی به‌طور تصادفی به زیرنمونه‌های فرعی با اندازه k تقسیم می‌شود. از زیرنمونه‌های فرعیk، یک زیرنمونه منفرد به‌عنوان داده‌های اعتبارسنجی برای آزمایش مدل حفظ می‌شود و زیرنمونه‌های $k-1$ به‌عنوان داده‌های آموزشی استفاده می‌شوند. سپس فرایند اعتبارسنج متقابل، که k بار تکرار می‌شود، با هر یک از نمونه‌های k به‌طور دقیق یک بار به‌عنوان داده‌های اعتبارسنجی مورد استفاده قرار می‌گیرد. پس از آن نتایج k بار تکرار می‌تواند برای تولید یک برآورد واحد به‌طور میانگین قرار بگیرد. در این روش همه مشاهدات برای هر دو آموزش و اعتبار مورد استفاده قرار می‌گیرند، و هر مشاهده برای اعتبارسنجی به‌طور دقیق استفاده می‌شود. ما در این پایان‌نامه k را مطابق مقاله [۵۹] را برابر دو در نظر گرفته می شود تا شرایط آزمایش یکسان گردد.

جهت انتخاب ویژگی، تابع ارزیابی الگوریتم می‌تواند به دو شکل زیر مورد بررسی قرار گیرد:

$$fitness = kfoldLoss = 1 - Validation\ Accurecy \quad (۱-۳)$$

$$fitness = m * kfoldLoss + n * {|S|}/{|C|} = m * (1 - Validation\ Accurecy) + n * {|S|}/{|C|} \quad (۲-۳)$$

که در آن $kfoldfLoss$ خطای، اعتبارسنجی متقابل $K-fold$ است. S تعداد ویژگی‌ها برای زیر مجموعه داده و C تعداد ویژگی‌ها برای کل مجموعه داده است. m و n دو ضریب هستند. در این پایان‌نامه مطابق مقاله [۵۹] مقادیر m برابر ۰/۹۹ و n برابر ۰/۰۱ در نظر گرفته می‌شود تا شرایط آزمایش یکسان گردد.

۳-۴ توابع انتقال در روش پیشنهادی

توابع انتقال زیادی مانند U، Z، X، S و یا V وجود دارد. اما با توجه به اینکه توابع انتقال S و V جزء توابع پایه بوده و در اکثر مقالات از آن‌ها استفاده شده است و مقاله [۵۹] نیز از این توابع استفاده کرده است در این پایان نامه نیز از این نوع توابع انتقال استفاده می‌شود. اما باید در نظر داشت در روش پیشنهادی از هر نوع تابع انتقال می‌توان استفاده کرد.

Pei Hu و همکاران در ۲۰۲۰ [۵۹] توابع $V_{2a}(x)$، $V_{3a}(x)$ و $V_{4a}(x)$ را به شکل جدول ۳-۱ بهبود دادند.

جدول ۳-۱: جزئیات توابع انتقال در روش پیشهنادی [۴۹]

نام تابع انتقال	تابع انتقال
$V_{2a}(x)$	$\lvert\tanh(x)/\tanh(4)\rvert$
$V_{3a}(x)$	$\left\lvert\sqrt{17} * x/(4 * \sqrt{1\_ + x^2})\right\rvert$
$V_{4a}(x)$	$\left\lvert\arctan(\frac{2}{\pi}x)/\arctan(2\pi)\right\rvert$

که $tanh(4)$، $4/\sqrt{17}$ و $arctan(2\pi)$ به ترتیب ماکزیمم مقادیر توابع انتقال $V_{2a}(x)$، $V_{3a}(x)$ و $V_{4a}(x)$ می‌باشند و معادله ۲-۱۰ را نیز به معادله ۳-۳ تغییر دادند:

$$a = 2 * \frac{it}{MAX\_IT} \quad (۳-۳)$$

در الگورتیم پیشنهادی نیز از روابط فوق جهت یکسان سازی با آزمایشات و قیاس با نتایج آن مقاله از آن‌ها استفاده می شود.

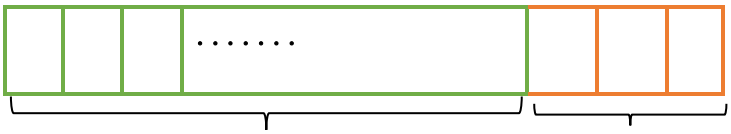
۳-۵ شرح روش پیشنهادی

برای پیاده سازی روش انتخاب تابع انتقال، نیاز به تکنیکی است که طی مراحل یادگیری، تابع انتقال را انتخاب کرده و مورد استفاده قرار دهد. برای برآورده کردن این هدف، ضمن ساختن جمعیت اولیه،

تعدادی سلول باینری به آن اضافه می‌شود. چگونگی استفاده از این سلول‌ها با مطرح کردن دو ایده و با سه روش اجرایی برای هر ایده، توضیح داده خواهد شد.

۳-۵-۱ روش پیشنهادی، ایده اول

در ابتدای شروع الگوریتم، هم زمان که مقادیر X را برای ذرات (موقعیت گرگ‌ها) مقداردهی می‌شود، مقادیر این سلول‌ها را نیز مقداردهی می‌گردد. به عبارت دیگر هر ذره (گرگ) همانطور که یک موقعیت خاص خود دارد، از یک تابع انتقال خاص خود استفاده می‌کند. اگر تعداد بیت‌ها را برابر سه در نظر گرفته شود در این وضعیت می‌توان به هشت تابع انتقال اشاره کرد. بنابراین در روش اول ABGWO-SV-Idea1، با ترکیب چهار تابع S شکل و چهار تابع V شکل، هر گرگ می‌تواند فقط یکی از این توابع انتقال را داشته باشد.



شکل ۳-۱: شمایی از رمزنگاری ایده اول با سه بیت برای چهار تابع S شکل و V شکل

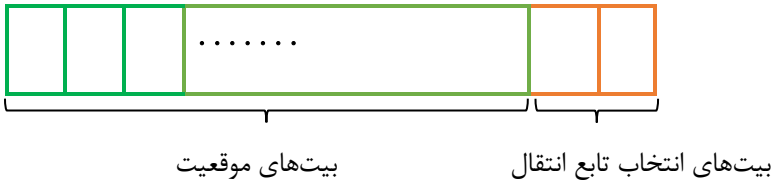
به جدول ۳-۲ به بیت‌های انتخاب تابع انتقال توجه کنید.

جدول ۳-۲: بیت‌های انتخاب تابع انتقال با سه بیت

تابع انتقال	بیت‌ها
S1	۰۰۰
S2	۰۰۱
S3	۰۱۰
S4	۰۱۱
V1	۱۰۰
V2	۱۰۱
V3	۱۱۰
V4	۱۱۱

همانطور که مشاهده می‌شود برای بیت‌های ۰۰۰ تابع انتقال S1 انتخاب می‌شود و برای بیت‌های ۰۰۱ تابع انتقال S2 انتخاب می‌شود و همینطور ادامه دارد تا بیت ۱۱۱ که تابع انتقال V4 انتخاب می‌شود. هر گرگی با توجه به بیت‌هایی که در ابتدای الگوریتم مقداردهی شده است تابع خاص خود را خواهد داشت.

اگر تعداد بیت‌های مورد نظر برابر دو بیت در نظر گرفته شود در این وضعیت می‌توان به چهار تابع انتقال اشاره کرد. در روش دوم ABGWO-S-Idea1، چهار تابع انتقال می‌تواند چهار تابع S شکل باشد. در روش سوم ABGWO-V-Idea1، چهار تابع انتقال می‌تواند چهار تابع V شکل باشد.



شکل ۳-۲: شمایی از رمزنگاری ایده اول با دو بیت برای چهار تابع S شکل یا V شکل

در این حالت به جدول ۳-۳ به دو بیت انتخاب تابع انتقال توجه کنید.

جدول ۳-۳: بیت‌های انتخاب تابع انتقال با دو بیت

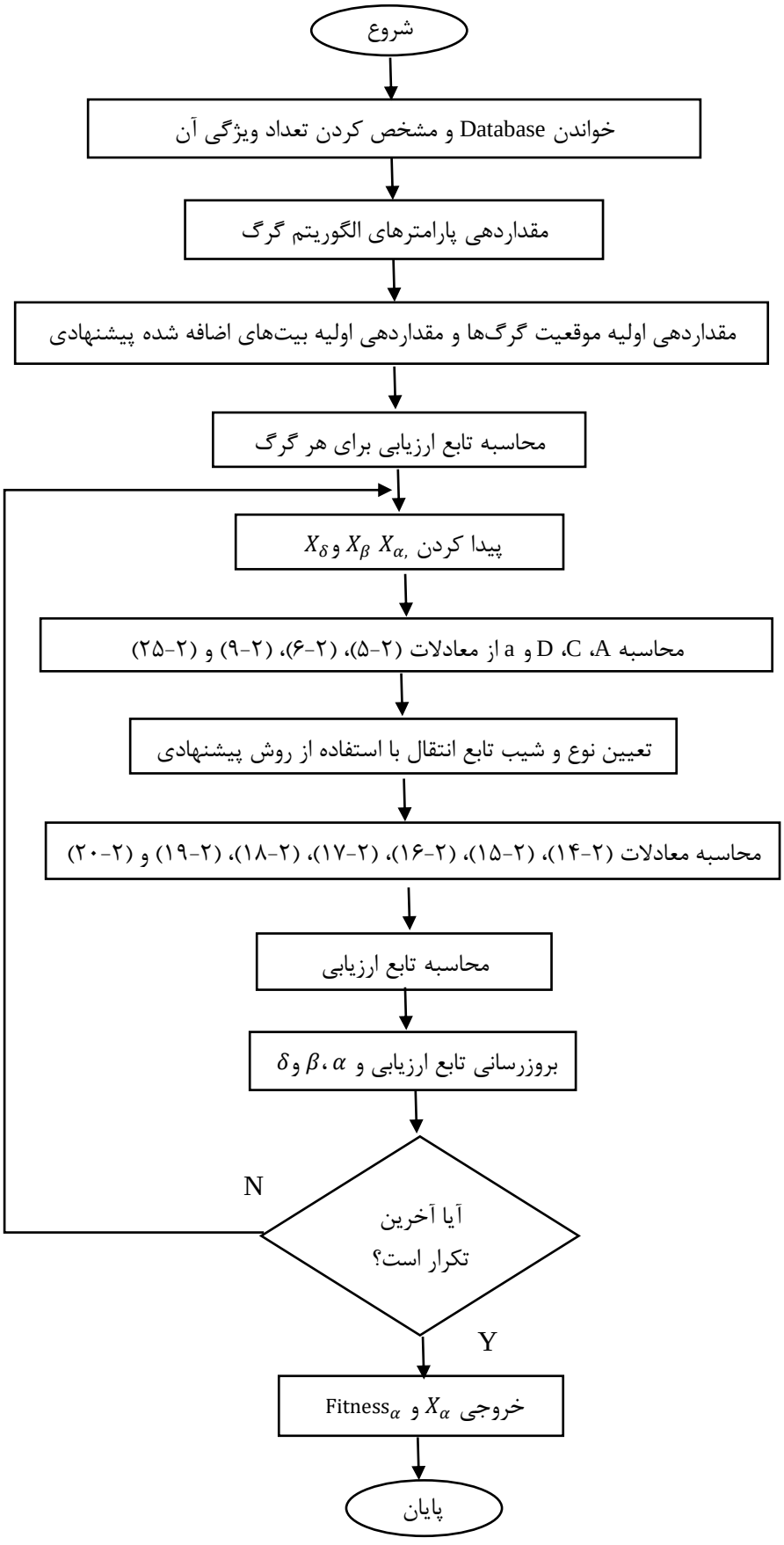
تابع انتقال	بیت‌ها
S1 OR V1	۰۰
S2 OR V2	۰۱
S3 OR V3	۱۰
S4 OR V4	۱۱

همانطور که مشاهده می‌شود برای بیت‌های ۰۰ تابع انتقال S1(V1) انتخاب می‌شود و برای بیت‌های ۰۱ تابع انتقال S2(V2) انتخاب می‌شود و برای بیت‌های ۱۰ تابع انتقال S3(V3) انتخاب می‌شود برای بیت‌های ۱۱ تابع انتقال S4(V4) انتخاب می‌شود. هر گرگی با توجه به بیت‌هایی که در ابتدای الگوریتم مقداردهی شده است تابع خاص خود را خواهد داشت.

چهار تابع S یا چهار تابع V شیب‌های متفاوتی از توابع S یا V شکل ارائه می‌دهند. بنابراین روش پیشنهادی هم نوع تایع انتقال و هم شیب مطلوب را در طی مراحل اجراء انتخاب و یاد می‌گیرد. شبه کد الگوریتم پیشنهادی در شکل ۳-۳ و فلوچارت آن در شکل ۳-۴ به تصویر کشیده شده است.

Initialize the related parametere of porposed algorithm
Randomly generate the positions of the wolves and porposed columns
Compute the fitness of each wolf
Find $X\_ \propto, X\_\beta \ \ and \ X\_\delta$
For $i = 1: MAX\_IT$ do
Update a , A and C by Eqs. (3-3), (2-5) and (2-6)
Determinate Kind of transition function
Calculate the position of each wolf by Eqs. (2-11)-(2-20)
Compute the fitness of each wolf
Update $X\_ \propto, X\_\beta \ \ and \ X\_\delta$
End for
Output $X\_ \propto$

شکل ۳-۳: الگوریتم بهینه‌سازی گسسته گرگ خاکستری - روش پیشنهادی



شکل ۳-۴: فلوچارت الگوریتم بهینه‌سازی گسسته گرگ خاکستری - روش پیشنهادی

۳-۵-۲ روش پیشنهادی، ایده دوم

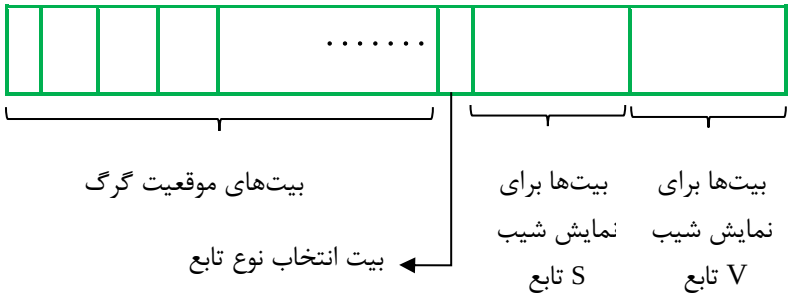
در ایده اول، چهار تابع انتقال از نوع S و چهار تابع انتقال از نوع V با شیب‌های مشخص سروکار داشت. در ایده دوم پاسخ همین الگوریتم را، زمانی که با شیب‌های مختلفی روبرو می‌شود مورد بررسی قرار می‌گیرد. برای همین منظور از توابع انتقال زیر استفاده می‌شود. ضریب x آن مقداری بین صفر تا یک می‌باشد که می‌تواند شیب‌های متفاوتی ارائه دهد.

$$S = \frac{1}{1 + e^{-\gamma x}} \quad (۳-۳)$$

$$V = |\tanh(\gamma x)| \quad (۳-۴)$$

γ بین صفر تا ۱ می‌باشد.

در ایده اول، برای اجرای طرح از دو یا سه بیت استفاده شد. در اینجا ۱۰ بیت برای نمایش شیب‌های مختلف تابع انتقال S شکل و ۱۰ بیت برای نمایش شیب‌های مختلف تابع انتقال V شکل استفاده می‌شود. این ۱۰ بیت اضافه شده مقادیر γ را تعیین می‌کنند. بنابراین در این ایده تا اینجا دو روش ABGWO-S-Idea2 و ABGWO-V-Idea2 مشخص گردید. اما اگر گرگ‌ها اجازه داشته باشند در طی اجرای الگوریتم، نوع تابع انتقال و نیز شیب تابع را انتخاب کنند، پس به ۱۰ بیت برای نمایش شیب‌های مختلف تابع انتقال S شکل و ۱۰ بیت برای نمایش شیب‌های مختلف تابع انتقال V شکل و یک بیت برای انتخاب نوع تابع، نیاز خواهند داشت که به روش ABGWO-SV-Idea2 معرفی می گردد. بنابراین در اینجا سه روش بیان شد. ۱۰ بیتی که برای شیب در نظر گرفته می‌شود 2^{10} حالت انتخاب برای شیب قابل دسترس می‌باشد.



شکل ۳-۵: شمایی از رمزنگاری ایده دوم با ۱۰ بیت برای تابع S شکل و ۱۰ بیت برای V شکل و یک بیت برای انتخاب نوع تابع

در بیت انتخاب نوع تابع اگر مقدار بیت صفر باشد تابع انتقال از نوع S و اگر مقدار بیت یک باشد تابع انتقال از نوع V خواهد بود.

معادله ۳-۵ فرمول ده بیتی برای نشان دادن شیب تابع S یا V است.

$$\gamma = \frac{\sum_{i=0}^{9} bit(i) * 2^i}{1023} \qquad (۵-۳)$$

نتیجه این کسر مقداری بین صفر و یک است. این مقدار همان مقدار γ در معادله ۳-۳ و ۳-۴ است.

برای شش روش پیشنهادی سایر روند الگوریتم، تغییری نمی‌کند. به عبارت دیگر، الگوریتم بعد از محاسبه تابع ارزیابی [۳۲]برای همه گرگ‌ها با استفاده از معادله‌های ۳-۱ یا ۳-۲ به صورت صعودی مرتب می‌شوند. با سه مقدار اولی به ترتیب گرگ‌های آلفا، بتا و دلتا مشخص می‌شوند. با مشخص شدن گرگ آلفا تابع انتقال و مقدار شیب تابع انتقال متعلق به این گرگ انتخاب می‌شود. با محاسبه فرمول‌های مربوط به گرگ خاکستری و استفاده از تابع انتقال جهت باینری کردن مقادیر پیوسته، موقعیت‌های جدید گرگ‌ها بدست می‌آیند. به عبارت دیگر موقعیت گرگ‌ها بروزرسانی می‌شوند. دوباره تابع ارزیابی برای همه گرگ‌ها با استفاده از معادله‌های ۳-۱ یا ۳-۲ محاسبه و به صورت صعودی مرتب می‌شوند. با سه مقدار اولی به ترتیب گرگ‌های آلفا، بتا و دلتا مشخص می‌شوند. با مشخص شدن گرگ آلفا تابع انتقال و مقدار شیب تابع انتقال متعلق به این گرگ انتخاب می‌شود. این کار تا پایان اجرای الگوریتم ادامه دارد. باید توجه شود که تابع انتقال در حین اجرای الگوریتم انتخاب می‌شود و بروزرسانی برای آن انجام نمی‌شود.

[32] Fitness Function

فصل ۴: نتایج و تفسیر آن ها

۴-۱ مقدمه

در این بخش نتایج حاصل از روش پیشنهادی و تحلیل و تفسیر آن‌ها ارائه می‌شود.

۴-۲ ارزیابی نتایج تجربی

شش روش پیشنهادی، یک بار با معادله ۳-۱ و بار دیگر با معادله ۳-۲ پیاده‌سازی شد. نتایج عددی به دست آمده به ترتیب در جداول ۴-۳ و ۴-۶ قابل مشاهده است. پارامترهای تنظیمی برای الگوریتم در جدول ۴-۱ آورده شده است. بر این اساس تعداد جمعیت اولیه را برابر ۳۰ و تعداد تکرار الگوریتم، برابر ۵۰۰ در نظر گرفته شده است. کل الگوریتم به علت کاهش خطا ۱۰ بار تکرار شده است.

جدول ۴-۱: تنظیم پارامترها برای آزمایشات

پارامترها	مقادیر
پارامتر K برای طبقه بندی کننده KNN	۵
پارامتر k برای اعتبارسنجی متقابل K-fold	۲
تعداد افراد جمعیت	۳۰
تعداد تکرار	۵۰۰
تعداد زمان اجرا	۱۰
پارامتر m در تابع ارزیابی	۰.۹۹
پارامتر n در تابع ارزیابی	۰.۰۱

الگوریتم توسط پردازنده Intel Core i5 2.6Ghz، با حافظه Ram 8 GB و با سیستم عامل Windows 10 بر روی نرم‌افزار MATLAB R2020b اجراء شده است.

با استفاده از ده مجموعه داده UCI مورد اشاره در جدول ۴-۲، الگوریتم مورد بررسی قرار گرفته است. در این جدول نام ده مجموعه داده به همراه تعداد نمونه‌ها و تعداد ویژگی‌ها آورده شده است.

جدول ۴-۲: جزئیات مجموعه داده‌های آزمایش [۵۹]

ردیف	مجموعه داده	نمونه	ویژگی	نوع ویژگی	مقدار از دست رفته
۱	Breast Cancer Wisconsin (Original)	۶۹۹	۱۰	عدد صحیح	بله
۲	Congressional Voting Records	۴۳۵	۱۶	غیر عددی	بله
۳	Statlog (Heart)	۲۷۰	۱۳	غیر عددی، حقیقی	خیر
۴	Ionosphere	۳۵۱	۳۴	عدد صحیح، حقیقی	خیر
۵	Chess (King–Rook vs. King–Pawn)	۳۱۹۶	۳۶	غیر عددی	خیر
۶	Lymphography	۱۴۸	۱۸	غیر عددی	خیر
۷	Connectionist Bench (Sonar, Mines vs. Rocks)	۲۰۸	۶۰	حقیقی	خیر
۸	SPECT Heart	۲۶۷	۲۲	غیر عددی	خیر
۹	Tic-Tac-Toe Endgame	۹۵۸	۹	غیر عددی	خیر
۱۰	Zoo	۱۰۱	۱۷	غیر عددی، عدد صحیح	خیر

نتایج عددی اجرای روش‌های پیشنهادی و دو الگوریتم BGWO از مقاله [۶۸] و ABGWO از مقاله [۵۹] مبتنی بر معادله ۳-۱ در جدول ۴-۳ آمده است.

جدول ۴-۳ با اعداد نتایج برای تأکید بر عملکرد برتر و نشان دادن اعتبار آن برجسته شده است. جدول ۴-۳ نشان می‌دهد که ۶ روش ارائه شده در این مقاله، نتایج بهتری نسبت به دو روش BGWO و ABGWO ارائه شده در مقاله [۵۹] دارند.

در ۶ روش پیشنهادی، روشی که فقط از توابع S شکل استفاده می‌شود نتایج ضعیف‌تری داشته و روشی که فقط از توابع V شکل استفاده می‌شود نتایج قابل توجه‌تری را نشان داده است.

جدول ۴-۳: مقایسه عددی نتایج الگوریتم‌های پیشنهادی با BGWO و ABGWO مبتنی بر معادله ۳-۱

ردیف	ABGWO-V Idea2		ABGWO-V Idea1		ABGWO-S Idea2		ABGWO-S Idea1		ABGWO-SV Idea2		ABGWO-SV Idea1		ABGWO		BGWO	
	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا
۱	۷	۰/۰۱۸۹	۶/۵	۰/۰۱۸۹	۸/۸	۰/۰۱۹۵	۸/۸	۰/۰۱۹۹	۷/۴	۰/۰۱۸۹	۷/۵	۰/۰۱۸۹	۶/۸۵	۰/۰۲۱۷	۷/۳	۰/۰۲۱۴
۲	۴/۵	۰/۰۲۹۵	۵	۰/۰۲۸۵	۱۰/۶	۰/۰۳۲۷	۹/۲	۰/۰۳۰۸	۶/۷	۰/۰۲۸۳	۶/۵	۰/۰۲۶۵	۸/۳۵	۰/۰۱۵۸۳	۱۰/۳۵	۰/۰۱۵۶۸
۳	۵/۸	۰/۱۲۹۶	۵/۸	۰/۱۳۹۳	۹	۰/۱۴۲۲	۸	۰/۱۴	۷/۲	۰/۱۴۵۲	۶/۴	۰/۱۴۰۴	۷/۰۵	۰/۱۸۴۶	۸/۴	۰/۱۸۳۵
۴	۱۴/۲	۰/۱۱۱۴	۱۴/۷	۰/۱۰۹۷	۳۱	۰/۱۳۴۵	۲۹/۸	۰/۱۳۲۸	۱۷/۷	۰/۱۱۷۱	۱۷/۶	۰/۱۱۷۱	۱۶/۴۵	۰/۱۲۷۸	۲۲/۳۵	۰/۱۳۴
۵	۱۸/۳	۰/۰۵۳۴	۲۰/۴	۰/۰۵۲۸	۳۳/۶	۰/۰۷۱۱	۳۱/۴	۰/۰۶۱۳	۲۴/۶	۰/۰۵۴۱	۱۹	۰/۱۱۲۳	۱۹/۸۵	۰/۰۶۶۳	۲۷/۳	۰/۰۶۸۸
۶	۸/۹	۰/۱۳۷۲	۹/۹	۰/۱۲۹۷	۱۵/۸	۰/۱۴۶۶	۱۵/۵	۰/۱۴۷۳	۱۱/۳	۰/۱۳۱۱	۱۰/۹	۰/۱۲۲۳	۱۰	۰/۵۳۲۴	۱۲/۵۵	۰/۵۲۹۷
۷	۲۸/۸	۰/۱۳۹۹	۲۷/۳	۰/۱۴۱۸	۵۷/۱	۰/۱۹۵۷	۵۳	۰/۱۸۲۲	۳۱/۹	۰/۱۵	۳۷/۹	۰/۱۵۲۴	۲۹/۵	۰/۱۷۰۴	۳۸/۹۵	۰/۱۸۸۵
۸	۱۳	۰/۱۴۱۲	۱۲/۷	۰/۱۳۹	۲۱/۳	۰/۱۴۰۸	۲۱/۳	۰/۱۴۰۴	۱۸	۰/۱۳۶۳	۱۵/۵	۰/۱۳۹	۱۲/۳	۰/۲۵۰۴	۱۶/۲۵	۰/۲۵۳
۹	۵/۶	۰/۲۱۵۲	۶	۰/۲۰۹۴	۹	۰/۱۷۱۶	۹	۰/۱۷۱۸	۹	۰/۱۷۱۲	۹	۰/۱۷۱۲	۶/۱	۰/۲۲۴	۶/۶	۰/۲۱۸۸
۱۰	۹/۸	۰/۰۳۳۷	۱۰/۸	۰/۰۳۶۶	۱۴/۶	۰/۰۴۱۶	۱۴/۵	۰/۰۴۰۶	۱۲/۴	۰/۰۳۶۶	۱۱/۹	۰/۰۳۴۷	۹/۷	۰/۰۶۴۹	۱۱/۷۵	۰/۰۶۵۳

نتایج آماری جدول ۴-۳ جهت معنادار بودن مقایسه شش روش پیشنهادی از دو ایده با BGWO و ABGWO به ترتیب در جداول ۴-۴ و ۴-۵ ارائه شده است. در این جداول علامت "+" نشان‌دهنده آن است که روش پیشنهادی برتر از BGWO یا ABGWO می‌باشد و علامت "-" نشان‌دهنده آن است که

روش پیشنهادی ضعیف تر از BGWO یا ABGWO می‌باشد و علامت "=" به این معنی است که نتایج روش پیشنهادی با BGWO یا ABGWO یکسان است.

همانگونه که در جدول ۴-۴ مقایسه روش‌های پیشنهادی با BGWO مبتنی بر معادله ۳-۱ مشاهده می‌شود، ABGWO-SV-Idea1 از نظر خطا، در نه مجموعه داده و از نظر تعداد ویژگی‌های انتخاب شده در هفت مجموعه داده نسبت به BGWO از خود برتری نشان می‌دهند و ABGWO-SV-Idea2 از نظر خطا، در همه ده مجموعه داده و از نظر تعداد ویژگی‌های انتخاب شده در شش مجموعه داده نسبت به BGWO از خود برتری نشان می‌دهند ولی ABGWO-V-Idea1 و ABGWO-V-Idea2 در همه ده مجموعه داده برتری چشمگیری نسبت به BGWO از نظر خطا و از نظر تعداد ویژگی‌های انتخاب شده دارا می‌باشند.

جدول ۴-۴: مقایسه الگوریتم‌های پیشنهادی از لحاظ خطای طبقه‌بندی و تعداد ویژگی‌های انتخاب شده با BGWO مبتنی بر معادله ۳-۱

ردیف	ABGWO-V Idea2		ABGWO-V Idea1		ABGWO-S Idea2		ABGWO-S Idea1		ABGWO-SV Idea2		ABGWO-SV Idea1		BGWO	
	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا
۱	+	+	+	+	-	+	-	+	-	+	-	+	۷/۳	۰/۰۲۱۴
۲	+	+	+	+	-	+	+	+	+	+	+	+	۱۰/۳۵	۰/۱۵۶۸
۳	+	+	+	+	-	+	+	+	+	+	+	+	۸/۴	۰/۱۸۳۵
۴	+	+	+	+	-	-	-	+	+	+	+	+	۲۲/۳۵	۰/۱۳۴
۵	+	+	+	+	-	-	-	+	+	+	+	-	۲۷/۳	۰/۰۶۸۸
۶	+	+	+	+	-	+	-	+	+	+	+	+	۱۲/۵۵	۰/۵۲۹۷
۷	+	+	+	+	-	-	-	+	+	+	+	+	۳۸/۹۵	۰/۱۸۸۵
۸	+	+	+	+	-	+	-	+	-	+	+	+	۱۶/۲۵	۰/۲۵۳
۹	+	+	+	+	-	+	-	+	-	+	-	+	۶/۶	۰/۲۱۸۸
۱۰	+	+	+	+	-	+	-	+	-	+	-	+	۱۱/۷۵	۰/۰۶۵۳
+	۱۰	۱۰	۱۰	۱۰	۰	۷	۲	۱۰	۶	۱۰	۷	۹		
-	۰	۰	۰	۰	۱۰	۳	۸	۰	۴	۰	۳	۱		
=	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰		

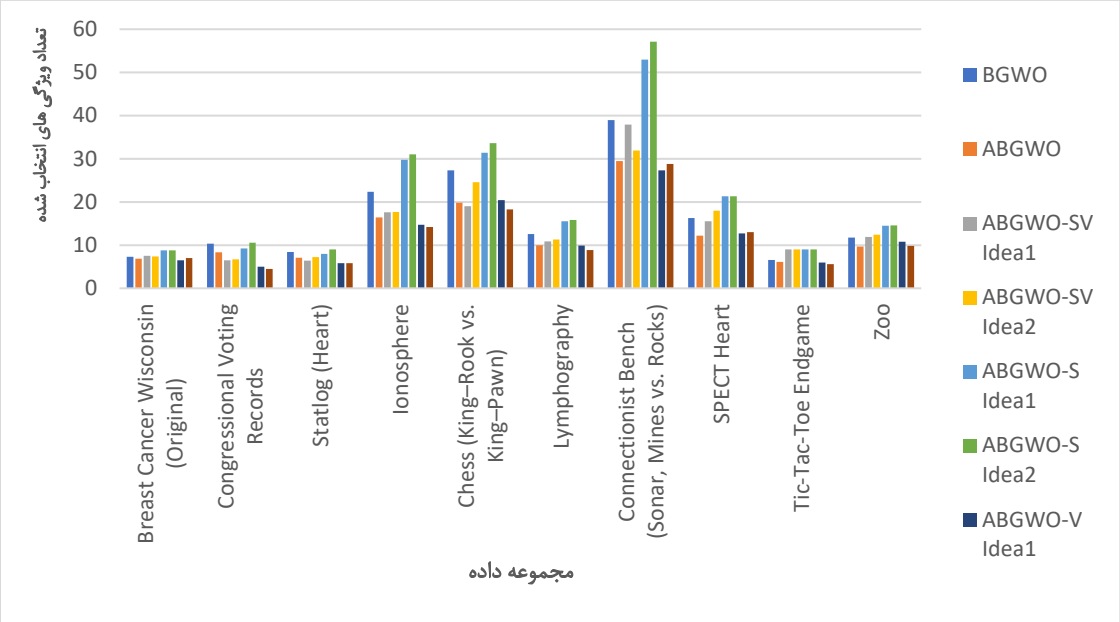
همانگونه که در جدول ۴-۵ مقایسه روش‌های پیشنهادی با ABGWO مبتنی بر معادله ۳-۱ مشاهده می‌شود، ABGWO-V-Idea1 و ABGWO-V-Idea2 در همه ده مجموعه داده برتری چشمگیری از نظر خطا نسبت به ABGWO دارند و از نظر تعداد ویژگی‌های انتخاب شده در هفت مجموعه داده برتری خود را نسبت به ABGWO نشان می‌دهند.

جدول ۴-۵: مقایسه الگوریتم‌های پیشنهادی از لحاظ خطای طبقه‌بندی و تعداد ویژگی‌های انتخاب شده با ABGWO مبتنی بر معادله ۳-۱

ردیف	ABGWO-V Idea2		ABGWO-V Idea1		ABGWO-S Idea2		ABGWO-S Idea1		ABGWO-SV Idea2		ABGWO-SV Idea1		ABGWO	
	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا
۱	-	+	+	+	-	+	-	+	-	+	-	+	۶/۸۵	۰/۰۲۱۷
۲	+	+	+	+	-	+	-	+	+	+	+	+	۸/۳۵	۰/۱۵۸۳
۳	+	+	+	+	-	+	-	+	-	+	+	+	۷/۰۵	۰/۱۸۴۶
۴	+	+	+	+	-	-	-	-	-	+	-	+	۱۶/۴۵	۰/۱۲۷۸
۵	+	+	-	+	-	-	-	+	-	+	+	-	۱۹/۸۵	۰/۰۶۶۳
۶	+	+	+	+	-	+	-	+	-	+	-	+	۱۰	۰/۵۳۲۴
۷	+	+	+	+	-	-	-	-	-	+	-	+	۲۹/۵	۰/۱۷۰۴
۸	-	+	-	+	-	+	-	+	-	+	-	+	۱۲/۲	۰/۲۵۰۴
۹	+	+	+	+	-	+	-	+	-	+	-	+	۶/۱	۰/۲۲۴
۱۰	-	+	-	+	-	+	-	+	-	+	-	+	۹/۷	۰/۰۶۴۹
+	۷	۱۰	۷	۱۰	۰	۷	۰	۸	۱	۱۰	۳	۹		
-	۳	۰	۳	۰	۱۰	۳	۱۰	۲	۹	۰	۷	۱		
=	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰		

از مقایسه نتایج بدست آمده از جداول ۴-۴ و ۴-۵ نتیجه می‌شود که روش‌های پیشنهادی ABGWO-V-Idea1 و ABGWO-V-Idea2 از نظر خطا و تعداد ویژگی انتخاب شده نسبت به ABGWO و BGWO برتری دارند. نتایج جداول ۴-۳، ۴-۴ و ۴-۵ براساس فرمول ۳-۱ می‌باشد که تابع ارزیابی در این حالت متاثر از خطای اعتبار سنجی می‌باشد و تعداد ویژگی ها نقشی در آن ندارند.

در شکل ۴-۱ نتایج گرافیکی حاصل از مقادیر عددی جدول ۴-۳ به تصویر کشیده شده است. در این نمودار تعداد ویژگی‌های انتخاب شده برحسب مجموعه داده‌ها برای روش‌های پیشنهادی و روش‌های ABGWO و BGWO مشاهده می‌شود. به وضوح مشخص است که الگوریتم‌های ABGWO-V-Idea1 و ABGWO-V-Idea2 نسبت به سایر الگوریتم‌ها از نظر تعداد حداقل ویژگی‌ها موفقیت بیشتری کسب کرده است.



شکل ۴-۱: مقایسه الگوریتم‌های پیشنهادی از لحاظ تعداد ویژگی‌های انتخاب‌شده با BGWO و ABGWO مبتنی بر معادله ۳-۱

نتایج عددی اجرای روش‌های پیشنهادی و دو الگوریتم BGWO از مقاله [۶۸] و ABGWO از مقاله [۵۹] مبتنی بر معادله ۳-۲ در جدول ۴-۶ آمده است.

جدول ۴-۶ با اعداد نتایج برجسته شده است تا بر عملکرد برتر تاکید و اعتبار آن را نشان دهد. جدول ۴-۶ نشان می دهد که ۶ روش ارائه شده در این مقاله، نتایج بهتری نسبت به دو روش BGWO و ABGWO ارائه شده در مقاله [۵۹] دارند.

در ۶ روش پیشنهادی، روشی که فقط از توابع S شکل استفاده می شود نتایج ضعیف تری داشته و روشی که فقط از توابع V شکل استفاده می شود نتایج قابل توجه تری را نشان داده است.

جدول ۴-۶: مقایسه عددی نتایج الگوریتم‌های پیشنهادی با BGWO و ABGWO مبتنی بر معادله ۳-۲

ردیف	ABGWO-V Idea2		ABGWO-V Idea1		ABGWO-S Idea2		ABGWO-S Idea1		ABGWO-SV Idea2		ABGWO-SV Idea1		ABGWO		BGWO	
	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا
۱	۶/۱	۰/۰۱۹۰	**۵/۴**	۰/۰۲۰۲	۸/۸	۰/۰۱۹۳	۷/۷	**۰/۰۱۸۸**	۶/۷	۰/۰۱۹۰	۶/۱	۰/۰۱۹۰	۵/۷۵	۰/۰۳۳۳	۷	۰/۰۳۵۳
۲	۵/۵	۰/۰۲۸۸	**۵/۲**	۰/۰۲۹۸	۹/۶	۰/۰۳۴۲	۸/۱	۰/۰۳۲۴	۶/۲	۰/۰۲۶۲	۵/۳	**۰/۰۲۵۹**	۸/۲۵	۰/۱۵۹۸	۹/۹	۰/۱۶۵۵
۳	۵/۹	۰/۱۳۳۳	**۵/۷**	۰/۱۴۱۸	۹/۳	۰/۱۵۳۴	۹	۰/۱۳۷۸	۶/۹	۰/۳۵۵	۶/۷	**۰/۱۲۷۴**	۶/۵	۰/۱۷۳۸	۷/۳	۰/۱۸۱۴
۴	**۱۱**	۰/۱۰۶۸	۱۱/۵	**۰/۱۰۵۷**	۲۸/۹	۰/۱۳۴۱	۲۷/۱	۰/۱۲۶۲	۱۵/۶	۰/۱۱۷۷	۱۴/۷	۰/۱۱۰۸	۱۶/۷۵	۰/۱۳۱	۲۱	۰/۱۴۰۵
۵	**۱۹/۴**	۰/۰۵۵۰	۲۰/۵	۰/۰۶۵۵	۳۳/۱	۰/۰۶۶۶	۳۲/۶	۰/۰۶۴۲	۲۰/۵	**۰/۰۴۷۸**	۲۲/۴	۰/۰۵۰۵	۲۰/۷۵	۰/۰۷۴۱	۲۷	۰/۰۸۴
۶	**۸/۳**	۰/۱۳۱۸	۱۶/۴	۰/۱۳۰۴	۱۶/۴	۰/۱۵۰۰	۱۶/۱	۰/۱۴۷۳	۱۱	۰/۱۳۳۸	۱۰/۵	**۰/۱۱۴۲**	۸/۸۵	۰/۵۲۷۳	۱۱/۸۵	۰/۵۳۷۶
۷	**۲۵/۴**	۰/۱۴۸۱	۵۶/۲	**۰/۱۳۹۵**	۵۶/۲	۰/۱۸۹۴	۵۶/۷	۰/۱۸۹۴	۳۳	۰/۱۴۸۶	۳۰/۶	۰/۱۵۰۵	۳۰/۴۵	۰/۱۸۷۷	۳۶/۶	۰/۱۹۷۸
۸	۱۱/۹	۰/۱۴۴۶	۲۰۶	۰/۱۴۳۰	۱۶/۴	۰/۱۴۰۸	۱۹/۸	۰/۱۳۶۷	۱۶/۴	**۰/۱۳۵۲**	۱۵/۲	۰/۱۳۶۴	۱۱/۱۵	۰/۲۵۳۱	۱۵/۴۵	۰/۲۶۶۹
۹	**۱۱/۴**	۰/۲۳۳۴	۵/۴	۰/۲۱۹۰	۹	۰/۱۷۱۳	۹	**۰/۱۷۰۶**	۸/۶	۰/۱۷۵۵	۸/۲	۰/۱۷۹۶	۵/۹	۰/۲۳۸۵	۶/۸	۰/۲۲۵۴
۱۰	۹/۷	۰/۰۴۰۶	**۸/۴**	۰/۰۴۸۵	۱۳/۶	۰/۰۳۵۷	۱۴/۵	۰/۰۳۹۶	۱۱/۸	۰/۰۳۴۶	۱۰/۷	**۰/۰۳۱۷**	۹/۲	۰/۰۷۷۲	۱۱/۴	۰/۱۰۳۱

نتایج آماری جدول ۴-۶ جهت معنادار بودن مقایسه شش روش پیشنهادی از دو ایده با BGWO و ABGWO به ترتیب در جداول ۴-۷ و ۴-۸ ارائه شده است. در این جداول علامت "+" نشان‌دهنده آن است که روش پیشنهادی برتر از BGWO یا ABGWO می‌باشد و علامت "-" نشان‌دهنده آن است که

روش پیشنهادی ضعیف تر از BGWO یا ABGWO می‌باشد و علامت "=" به این معنی است که نتایج روش پیشنهادی با BGWO یا ABGWO یکسان است.

همانگونه که در جدول ۴-۷ مقایسه روش‌های پیشنهادی با BGWO مبتنی بر معادله ۳-۲ مشاهده می‌شود، ABGWO-SV-Idea1 از نظر خطا، در همه ده مجموعه داده و از نظر تعداد ویژگی‌های انتخاب شده در نه مجموعه داده نسبت به BGWO از خود برتری نشان می‌دهند و ABGWO-SV-Idea2 از نظر خطا، در همه ده مجموعه داده و از نظر تعداد ویژگی‌های انتخاب شده در هفت مجموعه داده نسبت به BGWO از خود برتری نشان می‌دهند ولی ABGWO-V-Idea1 و ABGWO-V-Idea2 در همه ده مجموعه داده برتری چشمگیری نسبت به BGWO از نظر خطا و از نظر تعداد ویژگی‌های انتخاب شده دارا می‌باشند.

جدول ۴-۷: مقایسه الگوریتم‌های پیشنهادی از لحاظ خطای طبقه‌بندی و تعداد ویژگی‌های انتخاب شده با BGWO مبتنی بر معادله ۳-۲

ردیف	ABGWO-V Idea2		ABGWO-V Idea1		ABGWO-S Idea2		ABGWO-S Idea1		ABGWO-SV Idea2		ABGWO-SV Idea1		BGWO	
	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا
۱	+	+	+	+	-	+	-	+	+	+	+	+	۷	۰/۰۳۵۳
۲	+	+	+	+	+	+	+	+	+	+	+	+	۹/۹	۰/۱۶۵۵
۳	+	+	+	+	-	+	-	+	+	+	+	+	۷/۳	۰/۱۸۱۴
۴	+	+	+	+	-	+	-	+	+	+	+	+	۲۱	۰/۱۴۰۵
۵	+	+	+	+	-	+	-	+	+	+	+	+	۲۷	۰/۰۸۴
۶	+	+	+	+	-	+	-	+	+	+	+	+	۱۱/۸۵	۰/۵۳۷۶
۷	+	+	+	+	-	+	-	+	+	+	+	+	۳۶/۶	۰/۱۹۷۸
۸	+	+	+	+	-	+	-	+	-	+	+	+	۱۵/۴۵	۰/۲۶۶۹
۹	+	+	+	+	-	+	-	+	+	+	-	+	۶/۸	۰/۲۲۵۴
۱۰	+	+	+	+	-	+	-	+	-	+	+	+	۱۱/۴	۰/۱۰۳۱
+	۱۰	۱۰	۱۰	۱۰	۱	۱۰	۱	۱۰	۷	۱۰	۹	۱۰		
-	۰	۰	۰	۰	۹	۰	۹	۰	۳	۰	۱	۰		
=	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰		

همانگونه که مقایسه روش‌های پیشنهادی با ABGWO مبتنی بر معادله ۳-۲ در جدول ۴-۸ نشان داده شده است، ABGWO-V-Idea1 و ABGWO-V-Idea2 در همه ده مجموعه داده برتری چشمگیری نسبت به ABGWO از نظر خطا دارند و از نظر تعداد ویژگی‌های انتخاب شده ABGWO-V-Idea1 در نه مجموعه داده و ABGWO-V-Idea2 در شش مجموعه داده برتری خود را نسبت به ABGWO نشان می‌دهند.

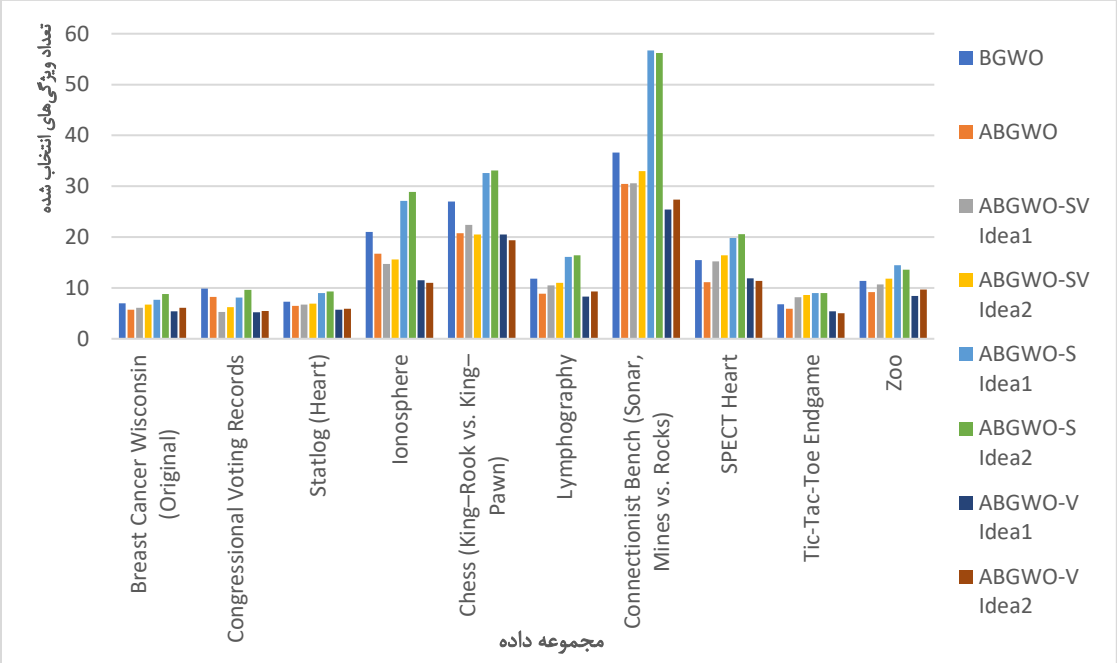
جدول ۴-۸: مقایسه الگوریتم‌های پیشنهادی از لحاظ خطای طبقه‌بندی و تعداد ویژگی‌های انتخاب‌شده با ABGWO مبتنی بر معادله ۳-۲

ردیف	ABGWO-V Idea2		ABGWO-V Idea1		ABGWO-S Idea2		ABGWO-S Idea1		ABGWO-SV Idea2		ABGWO-SV Idea1		ABGWO	
	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا	تعداد	خطا
۱	-	+	+	+	-	+	-	+	-	+	-	+	۵/۷۵	۰/۰۳۳۳
۲	+	+	+	+	-	+	+	+	+	+	+	+	۸/۲۵	۰/۱۵۹۸
۳	+	+	+	+	-	+	-	+	-	+	-	+	۶/۵	۰/۱۷۳۸
۴	+	+	+	+	-	-	-	+	+	+	+	+	۱۶/۷۵	۰/۱۳۱
۵	+	+	+	+	-	+	-	+	+	+	-	+	۲۰/۷۵	۰/۰۷۴۱
۶	-	+	+	+	-	+	-	+	-	+	-	+	۸/۸۵	۰/۵۲۷۳
۷	+	+	+	+	-	-	-	-	-	+	-	+	۳۰/۴۵	۰/۱۸۷۷
۸	-	+	-	+	-	+	-	+	-	+	-	+	۱۱/۱۵	۰/۲۵۳۱
۹	+	+	+	+	-	+	-	+	-	+	-	+	۵/۹	۰/۲۳۸۵
۱۰	-	+	+	+	-	+	-	+	-	+	-	+	۹/۲	۰/۰۷۲۲
+	۶	۱۰	۹	۱۰	۰	۸	۱	۹	۳	۱۰	۲	۱۰		
-	۴	۰	۱	۰	۱۰	۲	۹	۱	۷	۰	۸	۰		
=	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰		

از مقایسه نتایج بدست آمده از جداول ۴-۷ و ۴-۸ نتیجه می‌شود که روش‌های پیشنهادی ABGWO-V-Idea1 و ABGWO-V-Idea2 از نظر خطا و تعداد ویژگی انتخاب شده نسبت به ABGWO و BGWO برتری دارند. البته عملکرد ABGWO-V-Idea1 نسبت به ABGWO-V-Idea2 بهتر می‌باشد.

نتایج جداول ۴-۶، ۴-۷ و ۴-۸ براساس فرمول ۳-۲ می‌باشد که تابع ارزیابی در این حالت متاثر از خطای اعتبار سنجی و تعداد ویژگی‌ها می‌باشد.

در شکل ۴-۲ نتایج گرافیکی حاصل از مقادیر عددی جدول ۴-۶ به تصویر کشیده شده است. در این نمودار تعداد ویژگی‌های انتخاب شده برحسب مجموعه داده‌ها برای روش‌های پیشنهادی و روش‌های ABGWO و BGWO مشاهده می‌شود. به وضوح مشخص است که الگوریتم‌های ABGWO-V-Idea1 و ABGWO-V-Idea2 نسبت به سایر الگوریتم‌ها از نظر تعداد حداقل ویژگی‌ها موفقیت بیشتری کسب کرده است.



شکل ۴-۲: مقایسه الگوریتم‌های پیشنهادی از لحاظ تعداد ویژگی‌های انتخاب‌شده با BGWO و ABGWO مبتنی بر معادله ۳-۲

تا اینجا با استفاده از روش‌های پیشنهادی، انتخاب ویژگی برای ده مجموعه داده از مخزن UCI انجام گردید. با استفاده از همبستگی[۳۳] بین هر ویژگی و کلاس برچسب نیز می‌توان انتخاب ویژگی را انجام داد.

[33] Correlation

برای نمونه همبستگی بر روی یک مجموعه داده مانند Statlog Heart انجام داده می‌شود و با روش پیشنهادی ABGWO-V-idea2 که بر روی همین مجموعه داده اجراء گردیده است مقایسه می‌گردد. جدول ۴-۹ مشخصات مجموعه داده Statlog Heart از سایت UCI را نمایش می‌دهد. تعداد ویژگی‌ها ۱۳ بعلاوه یک برچسب کلاس می‌باشد.

جدول ۴-۹ جزئیات مجموعه داده statlog heart با ۱۳ ویژگی

ردیف	مقدار	ویژگی	توضیحات
۱	۲۹-۶۲	Age	سن بیمار بر حسب سال.
۲	۱ = مرد ۰ = زن	Sex	جنسیت.
۳	مقدار ۱ = آنژین معمولی مقدار ۲ = آنژین غیر معمول مقدار ۳ = درد غیر آنژین مقدار ۴ = بدون علامت	Cp	نوع درد قفسه سینه.
۴	مقدار عددی (۱۴۰ mm/Hg)	trestbps	فشار خون در حالت استراحت بر حسب میلی متر جیوه در هنگام پذیرش در بیمارستان اندازه گیری شد.
۵	مقدار عددی (۲۸۹ mg/dl)	Chol	کلسترول سرم بیمار بر حسب میلی گرم در دسی لیتر اندازه گیری شد.
۶	مقدار ۱ = درست مقدار ۰ = نادرست	Fbs	قند خون ناشتا بیمار. اگر بیشتر از ۱۲۰ mg/dl باشد، مقدار مشخصه ۱ (درست) است، در غیر این صورت مقدار مشخصه ۰ (نادرست) است.
۷	مقدار ۰ = معمولی مقدار ۱ = داشتن ST-T مقدار ۲ = هیپرتروفی	restecg	نتایج الکتروکاردیوگرافی استراحت برای بیمار.
۸	۱۴۰,۱۷۳.	thalach	حداکثر ضربان قلب بیمار.
۹	مقدار ۱ = بله. مقدار ۰ = خیر.	Exang	آنژین ناشی از ورزش
۱۰	مقدار عددی.	Oldpeak	افسردگی ST ناشی از ورزش نسبت به استراحت.
۱۱	مقدار ۱ = شیب دار. مقدار ۲ = تخت. مقدار ۳ = سرازیری.	Slope	اندازه گیری شیب برای تمرین اوج
۱۲	۰-۳	Ca	تعداد عروق اصلی (۰-۳) رنگ آمیزی شده توسط فلوروسکوپی.
۱۳	مقدار ۱ = معمولی. مقدار ۱ = رفع نقص. مقدار ۱ = نقص برگشت پذیر.	Thal	نشان دهنده ضربان قلب بیمار است.

در جدول ۴-۱۰ نتایج روش پیشنهادی ABGWO-V-idea2 بر روی مجموعه داده Statlog Heart نمایش داده شده است.در هر تکرار ویژگی‌هایی که انتخاب شده‌اند با عدد یک و ویژگی‌هایی که انتخاب نشده‌اند با عدد صفر نمایش داده شده است و در یک ردیف دیگر درصد میانگین ویژگی‌های انتخاب شده محاسبه شده است. همبستگی بین هر ویژگی و برچسب کلاس محاسبه و در سطر پایانی نمایش داده شده است.

همانگونه که مشاهده می‌شود ویژگی‌های cp, ca, thal, oldpeak, exang مربوط به ویژگی‌های ۳، ۹، ۱۰، ۱۲ و ۱۳ دارای بیشترین درصد میانگین (انتخاب ویژگی) از اجرای الگوریتم پیشنهادی می‌باشد و همین طور مشاهده می‌شود که همبستگی همین ویژگی‌ها با برچسب کلاس، دارای بیشترین مقدار می‌باشد. مشابه همین نتایج در مقاله [۷۰] نیز بدان اشاره شده است.

جدول ۴-۱۰ مقایسه همبستگی بین ستون ویژگی و ستون برچسب با درصد میانگین ویژگی انتخاب شده

ردیف ویژگی	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳
تکرار ۱	۰	۱	۱	۰	۰	۱	۱	۰	۰	۱	۰	۱	۱
تکرار ۲	۰	۰	۱	۰	۰	۱	۰	۰	۱	۰	۱	۱	۱
تکرار ۳	۰	۰	۱	۰	۰	۰	۰	۰	۱	۱	۰	۱	۱
تکرار ۴	۰	۰	۱	۰	۰	۰	۰	۰	۱	۰	۱	۱	۱
تکرار ۵	۰	۰	۱	۰	۰	۰	۱	۰	۰	۱	۱	۱	۱
تکرار ۶	۰	۰	۱	۰	۰	۱	۰	۰	۰	۱	۱	۱	۱
تکرار ۷	۰	۰	۱	۰	۰	۱	۱	۰	۰	۱	۰	۱	۱
تکرار ۸	۰	۰	۱	۰	۰	۰	۰	۰	۱	۱	۱	۱	۰
تکرار ۹	۰	۱	۱	۰	۰	۰	۰	۰	۱	۰	۰	۱	۱
تکرار ۱۰	۰	۰	۱	۰	۰	۰	۰	۰	۱	۱	۰	۱	۱
درصد میانگین ویژگی انتخاب شده													
	۹۰	۱۰۰	۵۰	۷۰	۶۰	۰	۳۰	۴۰	۰	۰	۱۰۰	۲۰	۰
همبستگی بین ستون ویژگی و ستون برچسب													
	۰/۵۲۵۰	۰/۴۵۵۳	۰/۳۳۷۶	۰/۴۱۸۰	۰/۴۱۹۳	۰/۴۱۸۵	۰/۱۸۲۱	۰/۰۱۶۳	۰/۱۱۸۰	۰/۱۵۵۴	۰/۴۱۷۴	۰/۲۹۷۷	۰/۲۱۲۳

فصل ۵: جمع‌بندی و پیشنهادها

۵-۱ جمع‌بندی

در شبیه‌سازی، شش روش ABGWO-SV-Idea1، ABGWO-SV-Idea2، ABGWO-S-Idea1، ABGWO-S-Idea2، ABGWO-V-Idea1 و ABGWO-V-Idea2 که با ۵۰۰ تکرار بر روی ۳۰ گرگ (جمعیت) انجام شده است هر جمعیت تابع انتقال خود را همراهی می‌کند به نحوی که بعد از هر تکرار و بروزرسانی موقعیت گرگ آلفا تابع انتقال نیز انتخاب می‌گردد. با تکرارهای بعدی، الگوریتم ضمن انتخاب تابع انتقال مناسب با شیب مناسب، یادگیری تابع انتقال را نیز انجام می‌دهد. روشی پیشنهادی منتسب کردن یک تابع انتقال متفاوت به هر گرگ، تاکنون در هیچ یک از مقالات مشاهده نشده است. در این حالت به الگوریتم این امکان داده می شود که در فرآیند بروزرسانی پارامترهای مساله و موقعیت گرگ آلفا، تابع انتقال را انتخاب کند، بنابراین الگوریتم قدرت مانور بیشتری از خود نشان می‌دهد.

با توجه به نتایج بدست آمده، از آزمایش‌های مختلفی که در جدول ۴-۳ برای فرمول ۳-۱ که فقط خطای اعتبار سنجی را در نظر می‌گیرد و جدول ۴-۶ برای فرمول ۳-۲ که خطای اعتبارسنجی و تعداد ویژگی‌ها را در نظر می‌گیرد، مشخص می‌باشد که الگوریتم‌های ABGWO-V-Idea1 و ABGWO-V-Idea2 نسبت به الگوریتم‌های ABGWO و BGWO توانایی بیشتری برای رسیدن به حداقل ویژگی انتخاب شده با حداقل خطای (بیشترین دقت) طبقه‌بندی را دارد. البته در تمام شرایط الگوریتم ABGWO-V-Idea2 نسبت به همه الگوریتم‌ها عملکرد چشمگیری از خود نشان داده است.

۵-۲ پیشنهاد برای کار آینده

با توجه به اینکه روش پیشنهادی، یک روش کلی برای یادگیری و انتخاب توابع انتقال با شیب‌های متفاوت در طول اجرای الگوریتم می‌باشد. لذا پیشنهاد می‌گردد این روش برای ترکیب بیشتری از توابع انتقال بکار گرفته شود، و یا می‌توان با افزایش تعداد بیت‌ها برای افزایش تعداد حالت‌های شیب توابع انتقال و میل فضای باینری شیب‌ها به پیوسته به بحث و بررسی نتایج آن اقدام کرد و یا می‌توان همین

روش را برای روش چند هدفه بهینه سازی به کار برد که بتواند هم خطا و هم تعداد ویژگی انتخاب شده را با هم کاهش داد.

مراجع

مراجع

[1] Bharati M. Ramageri. (2010). "**DATA MINING TECHNIQUES AND APPLICATIONS**"**,** Indian Journal of Computer Science and Engineering, Vol. 1 No. 4, pp.301-305.

[2] Bing Xue, Mengjie Zhang, Will N. Browne. (2013). "**Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach**", IEEE Transactions on Cybernetics , Vol.43, No.6, pp .1656-1671.

[3] Rui Zhang, Feiping Nie, Xuelong Li, Xian Wei. (2019). "**Feature Selection with Multi-view Data: A Survey**". Information Fusion**,** Vol.50, pp.158-167.

[4] Jiawei Han, Micheline Kamber, Jian Pei. (2012), "**Data maining concepts and techniques**", 3rd ed.

[5] Hoda Zamani, Mohammad-Hossein Nadimi-Shahraki. (2016), "**Feature Selection Based on Whale Optimization Algorithm for Diseases Diagnosis**", International Journal of Computer Science and Information Security, Vol.14, No.9, pp. 1243-1247.

[6] H.-H. Hsu, C.-W. Hsieh, and M.-D. Lu. (2011). "**Hybrid feature selection by combining filters and wrappers**", Expert Systems with Applications, Vol.38, NO.7, pp.8144-8150.

[7] E. Emary, Hossam M. Zawbaa, Aboul Ella Hassanien. (2015). "**Binary Gray Wolf Optimization Approaches for Feature Selection**", Neurocomputing**,** Vol.172, No. C, pp 371–381.

[8] Sharma, N., & Saroha, K. (2015). "**Study of dimension reduction methodologies in data mining**", In International Conference on Computing, Communication & Automation, pp.133-137.

[9] Van der Maaten, L. J. P. (2007). "**An introduction to dimensionality reduction using matlab**". Computer Science, 1201(07-07), pp. 62.

[10] C.-L. Huang, C.-J. Wang. (2006). "**A GA-based attribute selection and parameter optimization for support vector machine**", Expert Syst, Vol.31, No.2, pp.231–240.

[11] S. Kashef and H. Nezamabadi-pour. (2015). "**An advanced ACO algorithm for feature subset selection**", Neurocomputing, Vol.147, pp.271- 279.

[12] Esther Omolara Abiodun, Abdullah Alabdulatif, Abdulatif Alabdulatif, Rami S. Alkhawaldeh. (2021), "**A systematic review of emerging feature selection optimization methods for optimal text classification: the present state and**

prospective opportunities", Neural Computing and Applications, Vol. 33, No.22, pp. 15091–15118.

[13] Rahmaninia M, Moradi P. (2017). "**OSFSMI: online stream feature selection method based on mutual information**", Applied Soft Computing, Vol. 68, pp. 733-746.

[14] Hancer E, Xue B, Zhang M. (2018). "**Differential evolution for filter feature selection based on information theory and feature ranking**", Knowledge-Based Systems, Vol.140, No. C, pp.103–19.

[15] M. A. Hall and L. A. Smith. (1998), "**Practical feature subset selection for machine learning**", Proc. of the 21st Australasian Computer Science Conference ACSC'98, pp.181-191, Perth, Australia.

[16] Gu Q, Li Z, Han J. (2012). "**Generalized fisher score for feature selection**", Computer Science.

[17] I. Kononenko. (1994.). "**Estimating attributes: analysis and extensions of RELIEF**", Proc. European Conf. on Machine Learning, Vol.784, pp.171- 182.

[18] Hira ZM, Gillies DF. (2015). "**A review of feature selection and feature extraction methods applied on microarray data**", Advances in Bioinformatics, Vol.5, pp.1-13.

[19] Jacek Biesiada, Wlodzisław Duch. (2007). "**Feature Selection for High-Dimensional Data – A Pearson Redundancy Based Filter**", In Advances in Soft Computing, Vol.45, pp.242-249.

[20] Joseph Lee Rodgers; W. Alan Nicewander. (1988). "**Thirteen Ways to Look at the Correlation Coefficient**", The American Statistician, Vol.42, No.1, pp.59-66.

[21] K. Pavya, Dr. B. Srinivasan. (2018). "**REVIEW OF LITERATURE ON FILTER AND WRAPPER METHODS FOR FEATURE SELECTION**", INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY, Vol 7, No.1.

[22] B. Venkatesh, J. Anuradha. (2019). "**A Review of Feature Selection and Its Methods**", Cybernetics and Information Technologies, Vol.19, No.1.

[23] Sanz H, Valim C, Vegas E, Oller JM, Reverter F. (2018). "SVM-RFE: selection and visualization of the most relevant features through non-linear kernels", BMC bioinformatics, Vol.19, No.1, 432.

[24] Faris H, Mafarja MM, Heidari AA, Aljarah I, Ala'm AZ, Mirjalili S, Fujita H. (2018). "**An efficient binary salp swarm algorithm with crossover scheme for feature selection problems**", Knowledge Based Systems, Vol.154, pp.43-67.

[25] Gehad Ismail Sayed, Alaa Tharwat, Aboul Ella Hassanien. (2019). "**Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection**". Applied Intelligence, Vol.49, No.1, pp 188–205.

[26] Prachi Agrawal; Hattan F. Abutarboush; Talari Ganesh . (2021). "**Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009-2019)**", IEEE Access**,** Vol. 9.

[27] Rabia Aziz, C.K. Verma , Namita Srivastava. (2017). "**Dimension reduction methods for microarray data: a review**", AIMS Bioengineering, Vol. 4**,** No. 2**,** pp. 179-197.

[28] L.Yu and H.Liu. (2003). "**Feature selection for high-dimensional data: a fast correlation-based filter solution**", in Proc. of the 20th Int. Conf. on Machine Learning, ICML'03, Washington DC, USA, Vol. 2, pp.856-863,

[29] Jeng-Shyang Pan, Pei Hu, Shu-Chuan Chu. (2019). "**Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power**", Processes, Vol.7, No.11, 845.

[30] Wu Y, Liu Y, Wang Y, Shi Y, Zhao X. (2018). "**JCDSA: a joint covariate detection tool for survival analysis on tumor expression profiles**", BMC bioinformatics.; Vol. 19, No.1, 187.

[31] Yang R, Zhang C, Zhang L, Gao R. (2018). "A two-step feature selection method to predict Cancerlectins by Multiview features and synthetic minority oversampling technique", BioMed Research International**,** Vol.20, No.170.

[32] Yosef Masoudi, Sobhanzadeh, Habib Motieghader and Ali Masoudi-Nejad. (2019). "**FeatureSelect: a software for feature selection based on machine learning approaches**", BMC Bioinformatics, Vol.20, No.170.

[33] Metin SK. (2018). "**Feature selection in multiword expression recognition**", Expert Systems with Applications, Vol.92, No. C, pp.106–123.

[34] V. Bolon-Canedo, N. Sanchez-Marono, and A. Alonso-Betanzos. (2012). "**An ensemble of filters and classifiers for microarray data classification**", Pattern Recognition, Vol.45, No.1, pp.531-539.

[۳۵] روحی، ا و نظام آبادی پور، ح. (۱۳۹۶)، **"یک روش انتخاب ویژگی ترکیبی برای داده‌های با بعد بالا مبتنی بر خرد جمعی"**، نشریه مهندسی برق و مهندسی کامپیوتر ایران، ب- مهندسی کامپیوتر، شماره ۴، زمستان.

[36] V. Bolon-Canedo, N. Sanchez-Marono, A. Alonso-Betanzos, J. M.Benitez, and F.Herrera. (2014), "**A review of microarray datasets and applied feature selection methods**", Information Sciences, Vol.282, pp.111-135.

[37] S. Shoghian, M. Kouzehgar. (2012). "**A Comparison among Wolf Pack Search and Four other Optimization Algorithms**", World Academy of Science, Engineering and Technology, Vol.6.

[38] Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis. (2014). "**Grey Wolf Optimizer**", Advances in Engineering Software, Vol.69, pp.46-61.

[39] Hui Wang, Shahryar Rahnamayan, Hui Sun, Mahamed GH Omran. (2013). "**Gaussian bare-bones differential evolution**", IEEE Transactions *on* Cybernetics, Vol.43, No.2.

[40] Xingsi Xue, Junfeng Chen, Xin Yao. (2018). "**Efficient user involvement in semiautomatic ontology matching**", IEEE Transactions *on* Emerging Topics *in* Computational Intelligence, Vol.5, No.2.

[41] Zhenyu Meng, Jeng-Shyang Pan, Kuo-Kun Tseng. (2019). "*An* enhanced Differential Evolution algorithm *with* novel control parameter adaptation schemes *for* numerical optimization, Knowledge-Based Systems, Vol.168, pp.80–99.

[42] Chaoli Sun, Yaochu Jin, Ran Cheng, Jinliang Ding, Jianchao Zeng. (2017). "**Surrogateassisted cooperative swarm optimization of high-dimensional expensive problems**", IEEE Transactions on Evolutionary Computation, Vol.21, pp 644–660.

[43] Saber Salehi, Ali Selamat, M. Reza Mashinchi, Hamido Fujita. (2015). "**The synergistic combination of particle swarm optimization and fuzzy sets to design granular classifier**", Knowledge-Based Systems, Vol.76, pp.200–218.

[44] Dorigo M, Birattari M, Stutzle T. (2006). "**Ant colony optimization**", IEEE Computational Intelligence Magazine,Vol.1, No.1.

[45] Kennedy, J. and Eberhart, R. (1995). "**Particle Swarm Optimization**", Proceedings of the IEEE International Conference on Neural Networks, Vol.4, pp.1942-1948.

[46] Basturk B, Karaboga D. (2006). "**An artificial bee colony (ABC) algorithm for numeric function optimization**". IEEE swarm intelligence symposium, pp.12–4.

[47] Pei Hu, Jeng-Shyang Pan, Shu-Chuan Chu, Qing-Wei Chai, Tao Liu, ZhongCui Li, (2019). "**New hybrid algorithms for prediction of daily load of power network**", Appl. Sci. Vol.9, No.21, 4514.

[48] Muro C, Escobedo R, Spector L, Coppinger R. (2011). "**Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations**", Behav Process, Vol.88, No.3, pp.192–7.

[49] Mirjalili, S. and Lewis. (2013). "**A. S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization**", Swarm and Evolutionary Computation, Vol.9**,** pp.1-14.

[50] Majdi Mafarja, Derar Eleyan, Salwani Abdullah, Seyedali Mirjalili. (2017). "**S-Shaped vs. V Shaped Transfer Functions for Ant Lion Optimization Algorithm in Feature Selection Problem**", Proceedings of the International Conference on Future Networks and Distributed Systems, Article No.21, pp.1–7.

[51] J. Kennedy, R.C. Eberhart. (1997). "**A Discrete Binary Version of the Particle Swarm Algorithm**". Proc. Journal Pre-proof Journal Pre-proof 49 IEEE Int. Conf. Syst. Man, Cybern, IEEE Computer Society, Washington, DC, USA, pp.4104–4108.

[52] M. Nezamabadi-pour, Hossein, Rostami-Shahrbabaki, M. Maghfoori-Farsangi. (2008). "**Binary particle swarm optimization: challenges and new solutions**", On Computer Science *and* Engineering (JCSE), Vol.*6*, No.(1-A), pp.21-32,

[53] Z. Beheshti, S.M. Shamsuddin. (2014) . "**CAPSO: Centripetal accelerated particle swarm optimization**", Information Sciences , Vol. 258, pp. 54–79.

[54] Z. Beheshti, S.M. Shamsuddin, S. Hasan, N.E. Wong. (2016). "**Improved centripetal accelerated particle swarm optimization**", Advance Soft Compu, Vol.8, No.2, pp.1–26.

[55] Y. Zhang, D. Gong, X. Gao, T. Tian, X. Sun.(2020). "**Binary differential evolution with self-learning for multi-objective feature selection**", Information Sciences, Vol.507, pp.67–85.

[56] E. Pashaei, E. Pashaei, N. Aydin. (2019) . "**Gene selection using hybrid binary black hole algorithm and modified binary particle swarm optimization**", Genomics. Vol.111, No.4 , pp.669–686.

[57] Z. Yang, K. Li, Y. Guo, S. Feng, Q. Niu, Y. Xue, A. Foley. (2019). "**A binary symmetric based hybrid metaJournal Pre-proof Journal Pre-proof 50 heuristic method for solving mixed integer unit commitment problem integrating with significant plug-in electric vehicles**", Energy, Vol.170, pp.889–905.

[58] J. Gholami, F. Pourpanah, X. Wang. (2020). "**Feature selection based on improved binary global harmony search for data classification**", Applied Soft Computing Journal, Vol.93, No.106402.

[59] P. Hu, J.-S. Pan, S.-C. Chu. (2020). "**Improved Binary Grey Wolf Optimizer and Its application for feature selection**", Knowledge-Based Systems**,** Vol.195 , No.105746.

[60] Z. Beheshti. (2020). "**A time-varying mirrored S-shaped transfer function for binary particle swarm optimization**", Information Sciences**,** Vol.512 , pp.1503–1542.

[61] R. Eberhart, J. Kennedy. (1995). "**A new optimizer using particle swarm theory**", Conference MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp.39–43.

[62] Yang XS. (2010). "**A new metaheuristic bat-inspired algorithm**", Nature inspired cooperative strategies for optimization nicso, Springer, Vol.284, pp.65–74 .

[64] Seyedali Mirjalili. (2015). "**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.

[65] Reynolds CW. (1987). "**Flocks, herds and schools: a distributed behavioral model**", ACM SIGGRAPH Computer Graphics, Vol. 21, No.4, pp.25–34

[66] Saremi S, Mirjalili S, Lewis A. (2014). "**How important is a transfer function in discrete heuristic algorithms**", Neural Computing and Applications, Vol.26, No.3, pp.625-640.

[67] S. Mirjalili and A. Lewis.(2016). "**The Whale Optimization Algorithm**", Advances in Engineering Software, Vol. 95, pp. 51-67.

[68] Eid Emary, Hossam M. Zawbaa, Aboul Ella Hassanien. (2016). "**Binary grey wolf optimization approaches for feature selection**", Neuro computing, Vol.17.

[69] Abdelazim G. Hussien; Essam H Houssein; Aboul Ella Hassanien. (2017). "**A binary whale optimization algorithm with hyperbolic tangent fitness function for feature selection**", Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), Vol. 1, pp. 166 – 172.

[70] S.Chellammal, R. Sharmila. (2019). "**Recommendation of Attributes for Heart Disease Prediction using Correlation Measure**", International Journal of Recent Technology and Engineering (IJRTE), Vol. 8, No. 2S3.

Abstract

One of the challenges in pattern recognition is the data attribution selection process. Feature selection plays an important role in solving problems with high-dimensional data and is an important and fundamental step in pre-processing many classification and machine learning problems.

The proposed method of this dissertation is a feature selection method based on the gray wolf algorithm. In this algorithm, the Wrapper method is used to select the feature selection. The transfer function is also an important part of the binary gray wolf algorithm, which is necessary for mapping a continuous value to a binary value.

In all previous research, only one transfer function was used for the whole algorithm, and all wolves in the whole algorithm dealt with this transfer function. In this dissertation, the algorithm deals with several transfer functions and each wolf will have its own transfer function. The idea stems from the fact that algorithms are evolutionary meta-innovations and can optimize themselves. Do not depend on the transfer function.

In the proposed method, eight transfer functions are used, which are divided into two families, S-shaped and V-shaped. Two approaches are proposed for learning the transfer function.

In the proposed method, each wolf is assigned a specific function. This idea can make it possible to select the appropriate transfer function in each iteration of the algorithm according to the selection of the alpha wolf and the movement towards the prey. We used two ideas in this innovation. In the first idea, we add three or two binary bits to the initial population. If two bits are added, four modes of the transfer function are available, and if three bits are added, eight transfer functions are available. In the second idea, 10 or 21 binary bits are added to the initial population. If 10 bits are added, we will have a type of transfer function. 2^{10} coefficient modes are available for the slope of the transfer function. If 21 bits are added, two types of transfer functions are available, of which there will be 2^{10} coefficient modes for the slope of the transfer function. In both ideas, the added standard bits are used to select the transfer function as well as the coefficient affecting the slope of these functions. After each iteration, the alpha wolf position algorithm is updated and the transfer function is selected. With subsequent iterations, the algorithm, while selecting the appropriate transfer function with the appropriate slope, also learns the transfer function. Experimental results on ten UCI datasets show that the selection of the obtained feature subset with high classification accuracy is effective and efficient.

Keywords: Feature Selection, High Dimensional Data, Binary Gray Wolf Optimization, Transfer Functions, Binary

Shahrood University of Technology

Faculty of Computer Engineering

M.Sc. Thesis in Software Engineering

Learning The Transfer Function in Binary Metaheuristic Algorithm for Feature Selection in Classification Problems

By: Zahra Nassiri

Supervisors:
Dr. Mohsen Biglari
Dr. Hesam Omranpour

March 2022