



بهبود الگوریتم اثر نیلوفر آبی مبتنی بر سیستم استنتاج فازی

الهام دلیری نیا^(۱) مهرداد جلالی*^(۲) مهدی یعقوبی^(۳) حمید طباطبایی^(۴)

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

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

Applied Data Science and Analytics, SRH University Heidelberg, Heidelberg, Germany*

(۳) گروه مهندسی برق، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران

(۴) گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران

تاریخ دریافت: ۱۴۰۳/۰۸/۲۶ تاریخ پذیرش: ۱۴۰۳/۱۰/۰۴

چکیده

الگوریتم اثر نیلوفر آبی در سال ۲۰۲۴ ارائه شده است و برگرفته شده از گرده افشانی و حرکت بروی برگ‌های گل نیلوفر آبی می‌باشد. در این الگوریتم از مفاهیم هوش جمعی ایستا و پویا در قالب حرکت سنجاک‌ها استفاده شده که قدرت فرایند اکتشافی را در الگوریتم افزایش داده و همچنین قدرت استخراج با گرده‌افشانی محلی و حرکت آب بروی برگ‌های گل نیلوفر آبی در نظر گرفته شده است. اما در این الگوریتم مکانیسم دقیقی برای کنترل پارامترهای مهم در فرآیند استخراج و اکتشاف در نظر گرفته نشده است و حرکات سنجاک‌ها در همه شرایط بصورت تصادفی تعریف می‌شود و بدین خاطر، دقت و سرعت همگرایی این الگوریتم بهینه سازی، کاهش می‌یابد. در این مقاله یک سیستم استنتاج فازی در حرکت سنجاک‌ها ارائه شده تا دقت و سرعت همگرایی این الگوریتم با کنترل شعاع همسایگی، حرکت همترازی و انسجام افزایش یابد. نتایج بدست آمده الگوریتم پایه گل نیلوفر آبی در مقایسه با الگوریتم پیشنهادی بروی ۱۲ تابع تست در ابعاد بالا (۵۰ بعد)، نشان داد که روش فازی الگوریتم گل نیلوفر آبی دارای بیش از ۴۹ درصد بهبود در دقت همگرایی و بیش از ۹ درصد بهبود در سرعت همگرایی است. روش پیشنهادی در مسئله کاربردی بهبود کنترل فازی نیز نشان داد که توانسته در دو مسئله کنترل مخزن و پاندول معکوس نیز بهبود نتایج را به همراه داشته باشد.

*عهده‌دار مکاتبات:

مهرداد جلالی

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

Applied Data Science and Analytics, SRH University Heidelberg, Heidelberg, Germany

پست الکترونیکی: mehrdad.jalali@kit.edu

در حل مسائل بهینه‌سازی، هدف یافتن راه‌حل بهینه می‌باشد. به محدوده‌ی جواب‌های قابل قبول مسئله که جواب بهینه نیز یکی از زیر مجموعه‌های این محدوده است، فضای جستجو گفته می‌شود. هر نقطه از محدوده‌ی فضای جستجو، نشان دهنده‌ی یکی از راه‌حل‌های مسئله می‌باشد. در واقع جستجو برای پیدا کردن یک راه حل برای مسئله، برابر آن است که برای بدست آوردن مقادیر بهینه مربوط به مسئله بهینه‌سازی در محدوده فضای مسئله به جستجو پرداخته شود [1].

انسان در برخورد با مسائل تمایل دارد تا بهترین را تجسم و توصیف کند و به آن دست یابد، اما از آنجایی که می‌داند نمی‌تواند تمام شرایط حاکم بر بهترین را به خوبی شناسایی و تعریف کند، در بیشتر موارد به جای جواب بهترین یا بهینه مطلق و سراسری، به یک جواب رضایت‌بخش بسنده می‌کند. به همین دلیل سال‌هاست که محققان بدنال کشف بهترین الگوریتم‌ها برای حل یک مسئله هستند. در مقایسه الگوریتم‌های بهینه‌سازی دو معیار همگرایی و عملکرد مطرح می‌شود. بعضی از الگوریتم‌ها دارای همگرایی بوده ولی ممکن است عملکرد ضعیفی داشته باشند، یعنی فرایند بهبود آن‌ها از کارایی و سرعت لازم برخوردار نباشد، برعکس بعضی دیگر از الگوریتم‌ها همگرایی نداشته ولی عملکرد آن‌ها خیلی خوب است. از سال ۱۹۴۰ تا کنون روش‌ها و الگوریتم‌های بهینه‌سازی متعددی برای حل مسائل بهینه‌سازی بسیاری مطرح شده است. روش‌های ریاضی و کلاسیک بهینه‌سازی اغلب منجر به یک فرمول یا دستورالعمل خاص برای حل هر مسئله می‌شوند. این روش‌ها در حل مسائل پیچیده بهینه‌سازی، از توانایی مناسبی برخوردار نبودند و همچنین در یافتن بهینه کلی و سراسری مسئله قابلیت اطمینان را نداشتند. اغلب این روش‌ها نقطه بهینه محلی را بعنوان نقطه بهینه کلی در نظر می‌گیرند و نیز هر یک از این روش‌ها تنها برای مسأله خاصی کاربرد دارند [2].

با توجه به محدودیت‌ها و مشکلات روش‌های کلاسیک به منظور کاهش زمان پردازش و بهبود کیفیت راه‌حل‌ها، الگوریتم‌ها و روش‌های بهینه‌سازی نوین و هوشمندی پا به عرصه ظهور گذاشتند، که در حل مسئله‌های بهینه‌سازی مبتنی بر نوعی جستجوی هوشمند تدریجاً به سوی مکان‌های بهینه مسئله پیش می‌روند. امروزه الگوریتم‌های بهینه‌سازی الهام گرفته از طبیعت به عنوان روش‌های هوشمند بهینه‌سازی در کنار روش‌های کلاسیک، موفقیت قابل ملاحظه‌ای از خود نشان داده‌اند. این روش‌ها در حل بسیاری از مسائل بهینه‌سازی در حوزه‌های مختلفی چون تعیین مسیر بهینه عامل‌های خودکار، طراحی بهینه کنترل کننده برای پروسه‌های صنعتی، و نیز در طراحی عامل‌های هوشمند و غیره مورد استفاده قرار گرفته‌اند [3].

در استفاده از الگوریتم‌های بهینه‌سازی، ثابت شده است که یک الگوریتم نمی‌تواند در تمامی کاربردهای بهینه‌سازی به خوبی عمل کند و ارائه الگوریتم‌های بهینه‌سازی متنوع با ساختار و مکانیسم جستجوی متفاوت، می‌تواند در بهینه‌سازی مسائل مختلف کاربرد داشته باشد. در این مقاله یک راهکار بهبودی برای الگوریتم تکاملی جدید به نام اثر نیلوفر آبی (LEA) [4] با هدف ارتقای دقت و سرعت همگرایی مبتنی بر سیستم استنتاج فازی ارائه شده است. ساختار مقاله شامل مرور ادبیات تحقیق در بخش سه و توضیحات الگوریتم اثر نیلوفر آبی و روش پیشنهادی که الگوریتم اثر نیلوفر آبی فازی است در بخش چهارم آمده است. در بخش پنجم نتایج و در بخش ششم نتیجه‌گیری مقاله ارائه شده است.

۲- مرور ادبیات تحقیق

ادبیات تحقیق این مقاله بطور خلاصه به دو دسته از مقالات مرتبط است، دسته اول مقالاتی است که بر روی بهبود حرکت سنجاکک‌ها در الگوریتم سنجاکک ارائه شده است، زیرا در الگوریتم اثر نیلوفر آبی از حرکت سنجاکک استفاده شده است و دسته دیگر مقالاتی که در آن‌ها از سیستم استنتاج فازی برای بهبود الگوریتم‌های تکاملی مختلف استفاده شده است.

الگوریتم بهینه‌سازی سنجاکک با الهام از حرکت سنجاکک‌ها ارائه شده است [5] پس از ارائه این الگوریتم محققان در پژوهش‌های خود بر روی این الگوریتم کار کرده‌اند که از آن جمله در دو مقاله بطور مستقیم این الگوریتم بهبود یافته است که در [6] برای اپراتور حمله، از حمله دو طرفه استفاده کرده است و نشان داده شده است که اینگونه حرکت سنجاکک‌ها منجر به اکتشاف بهتر فضا شده است و همچنین از تابع نمایی برای حرکت سنجاکک‌ها بهره برده است که این مورد باعث حرکت‌های متنوع‌تری در این الگوریتم می‌شود، نتایج روش پیشنهاد شده در این مقاله نشان داده است که از نظر میزان دقت رسیدن به بهینه سراسری بهبود داشته است و از نظر سرعت همگرایی نیز روش پیشنهاد شده در تکرارهای کم‌تری به همگرایی می‌رسد.

و در [7] برای این الگوریتم حافظه در نظر گرفته شده است که برای ذخیره مکان بهترین سنجاکک پیدا شده استفاده می‌شود و در واقع الگوریتم گروه سنجاکک با الگوریتم ذرات ترکیب شده است. داشتن حافظه به الگوریتم پیشنهادی این مقاله کمک می‌کند تا به حرکت دسته جمعی شاپرک‌ها جهت دهی بهتری داده شود و این منجر به سرعت همگرایی بالای این روش شده است. نتایج روش پیشنهادی این مقاله نشان داده است که از نظر سرعت همگرایی نسبت

به الگوریتم سنجاکک پایه بهبود خوبی داشته است، اما دقت آن را بسیار بالاتر نبرده است و در مجموع اضافه کردن حافظه به این الگوریتم، بیشتر به سرعت همگرایی این الگوریتم کمک کرده است.

بروی الگوریتم سنجاکک مقالات دیگری نیز ارائه شده است که در این مقالات صرفاً از این الگوریتم بدون تغییراتی، در کاربردهای دیگر بهره گرفته شده است، که از آن جمله میتوان به مقاله [8] اشاره کرد که در آن از الگوریتم سنجاکک برای بهینه‌سازی پارامترهای روش ماشین بردار پشتیبان استفاده می‌شود و نشان داده است که دقت دسته‌بندی با روش ماشین بردار پشتیبان این مقاله نسبت به روش‌های بهبود یافته این روش با الگوریتم‌های تکاملی دیگر، بالاتر بوده است. در مقاله [9] از الگوریتم بهینه‌سازی سنجاکک در موقعیت‌یابی استفاده شده است و نشان داده است که قدرت جستجوی خوب این الگوریتم در فضای جستجو ف منجر به قدرت موقعیت‌یابی گره‌های بیسیم در انواع شبکه‌های بیسیم سیار و بیسیم سنسور می‌شود. در مقاله [10] از الگوریتم بهینه‌سازی سنجاکک در پیدا کردن بیشترین قدرت دنبال کننده در سیستم‌های قدرت استفاده شده است و هدف از استفاده از این الگوریتم بهینه‌سازی تابع سیستم قدرت است که به وسیله الگوریتم سنجاکک بیشینه می‌شود. نتایج نشان داده است که الگوریتم سنجاکک توانسته به عنوان الگوریتمی ساده در تنظیمات، بخوبی در این مسئله استفاده شود.

در مقاله [11] از الگوریتم بهینه‌سازی سنجاکک در خوشه‌بندی استفاده می‌شود و در آن خوشه‌بندی با این الگوریتم منجر به طول عمر شبکه‌های رادیو فرکانس میشود. این طول عمر شبکه با خوشه بندی دقیق گره‌های شبکه و مشخص کردن مرکز خوشه انجام می‌شود. هر مقدار مرکز خوشه دقیق‌تر انتخاب شود، مصرف انرژی در این شبکه‌ها پایین آمده و منجر به طول عمر شبکه می‌شود. نتایج نشان داده است که الگوریتم سنجاکک توانسته به عنوان الگوریتمی کارا در خوشه بندی مطرح شود و بخوبی در این مسئله بکار گرفته شود.

در مقاله [12] از الگوریتم بهینه‌سازی سنجاکک در بهینه‌سازی سیستم حرارت خورشیدی استفاده شده است و در آن تابع هدف در خصوص ارسال و ترکیب حرارت خورشیدی برای تولید برق و استفاده از آن در کنار برق شهری آمده است. نتایج نشان داده است که الگوریتم سنجاکک توانسته نقطه بهینه‌سازی چندهدفه این مسئله را با دقت مناسبی بدست آورد. همچنین در مقاله [13] از این الگوریتم در مسئله بهینه‌سازی چندهدفه دیگری که در برش صفحات تعریف می‌شود، بهره برده شده است که نشان دهنده نتایجی با دقت بالا در این مسئله بوده است.

در مقاله [14] الگوریتم بهینه‌سازی سنجاکک بهبود داده شده است و این بهبود با اضافه کردن اپراتوری برای ایجاد تنوع در این الگوریتم بوده است. روش پیشنهادی در این مقاله در زمینه تست نرم‌افزار و در هنگام ورود کاربران استفاده شده است. نتایج نشان می‌دهد که نسخه ارائه شده در این مقاله قدرت اکتشاف بالایی دارد و در کاربرد تست نرم‌افزار در تمام اجراها به دقت بالایی دست پیدا می‌کند.

از بهبودهای دیگر این الگوریتم در مقاله [15] مشاهده می‌شود که از نسخه گسسته آن برای بهینه‌سازی در کاهش ابعاد مجموعه داده‌ها و مسئله انتخاب ویژگی بهره‌گیری شده است و نشان داده است که در مجموعه داده‌هایی با ابعاد بالا نیز بخوبی می‌تواند به تعداد ویژگی مطلوب همگرا شود. در مقاله [16] نیز کاربرد دیگری از نسخه گسسته این الگوریتم در حل مسئله کوله پشتی به شکل ۰ و ۱ آمده است که نشان می‌دهد این الگوریتم در مسائل گسسته به خوبی می‌تواند استفاده شود.

از ویژگی‌های الگوریتم سنجاک تفاوت اصلی بین آن و گروه ذرات است، در نظر گرفتن مفاهیم تفکیک، هم‌ترازی، انسجام، جاذبه یا دافعه در این الگوریتم از مهم‌ترین تفاوت‌ها می‌باشند. اگرچه در کارهای گذشته تلاش برای ادغام مفاهیم تفکیک، هم‌ترازی و انسجام با الگوریتم گروه ذرات شده است اما الگوریتم سنجاک رفتار هوش جمعی سنجاک‌ها را با در نظر گرفتن همه فاکتورهای ممکن و اعمال آن‌ها به افراد در یک هوش جمعی مدل کرده است. همچنین مفاهیم هوش جمعی‌های ایستا و پویا در این الگوریتم کاملاً جدید می‌باشد که در حالت کلی باعث شده است این الگوریتم دارای دقت و سرعت همگرایی نسبتاً مناسبی باشد.

در این الگوریتم مکانیسم دقیقی برای کنترل پارامترهای مهم در فرایند استخراج و اکتشاف در نظر گرفته نشده است همچنین در این الگوریتم موقعیت سنجاک‌ها و میزان همگرایی آن‌ها برای کنترل پارامترها و شعاع هر سنجاک نادیده گرفته شده است که این عوامل در کنار حرکت تصادفی سنجاک‌ها و بدون کنترل، باعث می‌شود این الگوریتم ارزیابی تابع زیادی باشد و در واقع سرعت همگرایی بسیار مناسب نباشد. همچنین نتایج این الگوریتم در مقاله [9] و همچنین آزمایش‌ها نشان می‌دهد که این الگوریتم در توابع تست با ۱۰ بعد در هر دور اجرا به دقت بسیار بالای در همگرایی دست پیدا نکرده است که این نشان دهنده ضعف این الگوریتم در بهینه‌سازی توابع با ابعاد بالا است.

ساختار الگوریتم سنجاک برگرفته از الگوریتم گروه ذرات است و مراجع زیادی در زمینه استفاده از منطق فازی در الگوریتم‌های مختلف از جمله الگوریتم گروه ذرات وجود دارند. در هرکدام از مقاله‌ها که برای تنظیم پارامترها از منطق فازی استفاده شده است، با توجه به آزمایش‌ها و نتایج تجربی، مزایایی را به همراه داشته است و همچنین نتایج مثبت بوده است. در مقاله [17] یک استراتژی جدید بنام FAOPSO معرفی شده است. در ابتدا برای جلوگیری از افتادن در بهینه محلی روشی معرفی شده است نتایج آزمایش‌ها در این مقاله نشان می‌دهد که رویکرد رقابتی موجود باعث همگرایی بهتر و در نتیجه بدست آمدن نتایج بهتر می‌شود و همچنین در مقاله دیگری [18] در بهینه‌سازی ذرات از منطق فازی استفاده شده است، در این مقاله برای حل دو مشکل افتادن در دام بهینه محلی و پایین بودن سرعت همگرایی از منطق فازی استفاده شده است. در این مقاله دو الگوریتم ترکیبی با استفاده از منطق فازی ارائه می‌گردد. الگوریتم‌های ارائه شده در این مقاله سعی

می‌کند مشکلات عنوان شده در PSO را حل نماید. در این مقاله دو روش ارائه شده است که در واقع یک تابع غیرخطی برای ضریب اینرسی W و تابع دیگری برای ضرایب شتاب $c1$ و $c2$ می‌باشند که از افتادن در بهینه محلی جلوگیری می‌کند. یکی از مشکلات دیگری که مطرح شد همگرایی خطی الگوریتم بود. این کار باعث می‌شود که سرعت همگرایی به مراتب کاهش بیابد برای حل این مشکل نیز از توابع غیرخطی برای پارامتر وزن میانی استفاده شده است. چون در این سیستم‌ها چندین پارامتر برای تصمیم‌گیری وجود دارد، بنابراین توابع فازی می‌تواند از جایگاه ویژه‌ای برخوردار باشد. با توجه به نتایج بدست آمده می‌توان گفت که استفاده از توابع فازی رفتار الگوریتم PSO استاندارد را به مراتب بهبود می‌بخشد و این امکان را برای الگوریتم PSO فراهم می‌کند که در مورد موقعیت‌های غیرعادی، عکس العمل مناسبی نشان دهد. همچنین چون چندین پارامتر در تصمیم‌گیری دخیل می‌باشد، منطق فازی یکی از بهترین روش‌ها برای توابع غیرخطی در PSO می‌باشد.

همچنین در دسته دیگری از مقالات به بهبود الگوریتم گروه ذرات با استفاده از سیستم استنتاج فازی پرداخته شده است که برخی از مقالات اخیر آن [19] است. در این مقالات مشاهده می‌شود که قدرت اکتشاف و استخراج الگوریتم گروه ذرات می‌تواند با کنترل پارامتر c در این روش به خوبی کنترل شود. در این مقالات کارایی روش پیشنهادی در توابع تست و چندین مسئله کاربردی آمده است و نشان می‌دهد که استفاده از سیستم استنتاج فازی برای کنترل پارامتر الگوریتم گروه ذرات می‌تواند به سرعت این الگوریتم بسیار کمک نماید. با توجه به ساختار الگوریتم سنجاچک که برگرفته از الگوریتم گروه ذرات است، استفاده از سیستم استنتاج فازی در الگوریتم سنجاچک یکی از ایده‌های این پایان نامه بوده است.

در مرور مقالات مشاهده می‌شود که سیستم استنتاج فازی به عنوان سیستمی برای کنترل فرایند اکتشاف و استخراج در الگوریتم‌های تکاملی مختلف استفاده شده است که از آن جمله در مقالاتی به بهبود الگوریتم ژنتیک برای کاربردهای مختلف پرداخته شده است که از جمله مقالات میتوان به [20,21] اشاره نمود. در این دو مقاله نرخ تقاطع و جهش به صورت فازی تنظیم می‌شود و در ورودی های سیستم فازی تکرار الگوریتم و میزان همگرایی قرار دارد. نتایج استفاده از سیستم فازی نشان می‌دهد که سرعت همگرایی در الگوریتم ژنتیک بسیار تحت تأثیر قرار گرفته و کاهش می‌یابد.

با بررسی روش‌های تکاملی دیگر نیز مشاهده می‌شود سیستم استنتاج فازی بخوبی توانسته تقریباً در تمام این الگوریتم‌ها برای کنترل دقیق‌تر پارامترها استفاده شود و نتایج را از نظر سرعت و دقت همگرایی بهبود دهد. از جمله این مقالات به الگوریتم رقابت استعماری فازی [22] میتوان اشاره کرد که در این مقاله مشاهده می‌شود، حرکت کشورهای مستعمره به سمت کشور امپراطوری و همچنین زاویه حرکتی آن‌ها می‌تواند بخوبی با سیستم استنتاج فازی

کنترل شود که این خود منجر به بالابردن قدرت استخراج این الگوریتم شده است زیرا یکی عیب‌های الگوریتم رقابت استعماری ضعف در استخراج مخصوصاً در توابع با ابعاد بالا است.

در پژوهش دیگری، الگوریتم فاخته فازی [23] ارائه می‌شود در این الگوریتم با کمک سیستم استنتاج فازی حرکت فاخته‌ها به ناحیه بهتر غذا کنترل می‌شود. یکی از پارامترهای مهم در الگوریتم فاخته حرکت مهاجرت است که باعث حرکت دسته‌جمعی جواب‌ها می‌شود. در سیستم استنتاج فازی پیشنهادی در این مقاله این حرکت از نظر طول گام حرکتی کنترل می‌شود و نشانه داده شده است که دقت همگرایی الگوریتم فاخته بهبود پیدا کرده است.

در الگوریتم مورچگان فازی [24] به ارائه راهکاری برای تولید حرکت‌های جدید با استفاده از سیستم استنتاج فازی پرداخته شده است و میزان تولید فرمون در راه حل‌های مناسب با استفاده از سیستم استنتاج فازی به‌خوبی کنترل می‌شود و نتایج نشان دهنده بالابردن سرعت همگرایی در این روش نسبت به نسخه پایه الگوریتم مورچگان است.

الگوریتم میمون عنکبوتی فازی [25] که برگرفته از رفتار این حیوان است از نظر اپراتوری بسیار شبیه به الگوریتم عنکبوت بوده و در آن کنترل پارامتر a در میزان اکتشاف و استخراج این الگوریتم مهم است که در این مقاله با استفاده از سیستم استنتاج فازی راهکاری برای کنترل آن ارائه می‌شود و نشان داده شده که نسبت به نسخه پایه الگوریتم میمون عنکبوتی دارای دقت رسیدن به بهینه سراسری بالاتری است.

در مقالات دیگری نیز از سیستم استنتاج فازی برای کنترل پارامترهای الگوریتم‌های تکاملی استفاده می‌شود که از آن جمله الگوریتم جنگل فازی [26] در کنترل پارامتر تولید درخت، الگوریتم پنگوئن فازی [27] در کنترل نحوه حرکت پنگوئن به سمت محل غذا، الگوریتم زنبور عسل فازی [28] در رقص زنبور و نشان داده محل غذا به دیگر زنبورها و الگوریتم تکامل تفاضلی فازی [29] در تنظیم استراتژی‌های اکتشاف و استخراج در این الگوریتم و غیره اشاره نمود. همچنین بهبود الگوریتم بهینه‌سازی طوفان مغزی [30]، الگوریتم شاهین‌های هریس [31] از دیگر روش‌هایی است که در سال‌های اخیر با تکنیک فازی به ارتقاء نتایج آن‌ها پرداخته شده است. با توجه به مرور مراجع در حوزه فازی مشاهده شد که هیچ بهبودی به وسیله الگوریتم فازی بر روی حرکت سنجاقک‌ها انجام نشده است. سیستم استنتاج فازی دارای مزایای زیادی می‌باشد و استفاده از آن در الگوریتم‌های مختلف نشان می‌دهد که میتوان از این مزایا برای بهبود الگوریتم‌های مختلف استفاده شود. بنابراین در این مقاله نشان داده شده است که استفاده از منطق فازی در بهبود الگوریتم اثر نیلوفر آبی که در آن از حرکت سنجاقک‌ها استفاده شده است، باعث بهبود نتایج می‌شود.

معرفی الگوریتم اثر نیلوفر آبی: در الگوریتم بهینه سازی اثر نیلوفر آبی [4]، فرایند گرده افشانی گل و خاصیت برگه‌ای آن، منبع الهام الگوریتم بوده است. در گرده افشانی دو فرآیند عمده وجود دارد که شامل فرآیند بیولوژیک و گرده افشانی متقابل و فرایند غیربیولوژیک و خودگرده افشانی است.

در گرده افشانی بیولوژیکی، گرده‌ها از یک گل به گل دیگر در گیاهان مختلف توسط گرده افشان‌هایی مانند حشرات، پرندگان و ... منتقل می‌شود. گرده افشانی متقابل در فاصله طولانی رخ دهد و می‌تواند به عنوان یک روند گرده افشانی سراسری و جستجو تمام فضای جستجو (اکتشاف) در نظر گرفته شود. جهت مدل‌سازی این بخش از مدل ریاضی الگوریتم بهینه سازی سنجاکک، استفاده شده است.

در فرآیند خود گرده افشانی خود باروری، لقاح یک گل از گرده‌های گل‌های دیگر همان گیاه است. در این نوع گرده افشانی، باد و انتشار در آب به گرده افشانی این گونه گیاهان گلدار کمک می‌کند. خود گرده افشانی به عنوان گرده افشانی محلی در نظر گرفته می‌شود و جستجو جهت استخراج بهینه صورت می‌گیرد. با توجه به خاصیت برگه‌ای این گیاه که آب بدون جذب در برگ، از روی برگ می‌لغزد. در این خصوص یک جستجوی محلی در حرکت قطرات بروی برگ می‌تواند در نظر گرفته شود.

۲-۱-۱- فاز اکتشاف

گرده افشانی سراسری با کمک سنجاکک‌ها: برای شبیه‌سازی رفتار هوشمندانه سنجاکک‌ها، سه اصل اولیه از توده‌های حشرات و همچنین دو مفهوم غذا و مهاجم در نظر گرفته شده است.

تفکیک: که اشاره به اجتناب از برخورد یک فرد با سایر افراد همسایه دارد.

هم تراز: که نشان دهنده تنظیم سرعت افراد با توجه به سایر افراد همسایه می‌باشد.

انسجام: که اشاره به تمایل افراد به سمت مرکز ثقل همسایه‌ها دارد.

هدف اصلی در هر توده و گروه، حفظ بقا می‌باشد، در نتیجه همه افراد باید به سمت منابع غذایی جذب شوند و از دشمن‌ها دور بمانند. با در نظر گرفتن ای دو رفتار، ۵ فاکتور در بروزرسانی موقعیت افراد در توده وجود دارد. هریک از مفاهیم فوق را میتوان بصورت ریاضی مدل کرد:

تفکیک را میتوان بصورت رابطه ۱ محاسبه کرد:

$$S_i^t = - \sum_{j=1}^N (X_i^t - X_j^t) \quad (1)$$

که در آن X_i^t موقعیت فرد جاری با شماره i در تکرار t را نشان می‌دهد و X_j موقعیت j امین فرد در همسایگی در تکرار t را نشان می‌دهد و N نیز تعداد افراد موجود در همسایگی را نشان می‌دهد.

هم ترازوی بصورت رابطه ۲ محاسبه می‌شود:

$$A_i^t = \frac{\sum_{j=1}^N X_j^t}{N} \quad (2)$$

که X_j سرعت j امین فرد در همسایگی در تکرار t را نشان می‌دهد.

انسجام به صورت رابطه ۳ محاسبه می‌شود:

$$C_i^t = \left(\frac{\sum_{j=1}^N X_j^t}{N} \right) - X_i^t \quad (3)$$

که X_i^t موقعیت فرد جاری با شماره i در تکرار t می‌باشد و N تعداد همسایه‌ها و X_j موقعیت j امین فرد در

همسایگی در تکرار t را نشان می‌دهد. جذب شدن به سمت منبع غذایی به صورت رابطه ۴ محاسبه می‌شود:

$$F_i^t = X_+^t - X_i^t \quad (4)$$

که در آن X_i^t موقعیت فرد جاری با شماره i در تکرار t و X_+^t موقعیت منبع غذا است که از تکرار فعلی یعنی t

مشخص شده است و بهترین جواب پیدا شده است.

فرار از دشمن به صورت رابطه ۵ محاسبه می‌شود:

$$E_i^t = X_-^t + X_i^t \quad (5)$$

که در آن X_i^t موقعیت فرد جاری با شماره i در تکرار t و X_-^t موقعیت دشمن که از دور تکامل فعلی یعنی t

مشخص شده است و بدترین جواب پیدا شده است. بردار طول گام یا سرعت، جهت حرکت سنجاقک‌ها را نشان

می‌دهد و بصورت زیر تعریف می‌شود:

$$\Delta X_i^{t+1} = (sS_i^t + aA_i^t + cC_i^t + fF_i^t + eE_i^t) + w\Delta X_i^t \quad (6)$$

که در آن s ضریب تفکیک را نشان می‌دهد و S_i^t میزان تفکیک مربوط به فرد i ام در دور تکامل t می‌باشد و a

ضریب مرتبط با هم ترازوی می‌باشد و A_i^t مقدار هم ترازوی مربوط به فرد i ام می‌باشد، c نیز ضریب انسجام می‌باشد

و G_i^t نیز مقدار انسجام مربوط به i امین فرد می‌باشد. مقدار f فاکتور تغذیه می‌باشد و F_i^t نیز منبع غذای مربوط به i امین فرد و e فاکتور دشمنی می‌باشد و E_i^t موقعیت دشمن مربوط به i امین فرد و w وزن اینرسی بوده و t نیز شمارنده تکرار الگوریتم است.

بعد از محاسبه بردار گام، بردارهای موقعیت به صورت رابطه ۷ محاسبه می‌شوند:

$$X_i^{t+1} = X_i^t + \Delta X_i^{t+1} \quad (7)$$

که در آن t شمارنده تکرار الگوریتم می‌باشد.

به منظور بهبود رفتارهای تصادفی و اتفاقی در اکتشاف سنجاقک‌های مصنوعی، آن‌ها نیاز دارند تا در زمانی که هیچ راه‌حلی در همسایگی آن‌ها وجود ندارد در اطراف فضای جستجو با استفاده از یک طول گام تصادفی پرواز کنند. در این حالت، موقعیت سنجاقک‌ها با استفاده از رابطه رابطه ۸ بروزرسانی می‌شود:

$$X_i^{t+1} = X_i^t + Levy(d) \times X_i^t \quad (8)$$

که t شمارنده تکرار فعلی می‌باشد و d بعد های بردار موقعیت می‌باشد و مقدار $levy$ با استفاده از رابطه رابطه ۹

محاسبه می‌شود:

$$Levy(x) = 0.01 \times \frac{r_1 \times \sigma}{|r_2|^{\frac{1}{\beta}}} \quad (9)$$

که r_1 و r_2 دو عدد تصادفی در بازه صفر و یک می‌باشند و بتا یک ثابت و β الفبا با استفاده از رابطه رابطه ۱۰

محاسبه می‌شود:

$$\sigma = \left(\frac{\mathcal{L}(1 + \beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\mathcal{L}\left(\frac{1 + \beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \quad (10)$$

که در آن:

$$\mathcal{L}(x) = (x - 1)! \quad (11)$$

۲-۱-۲- فاز استخراج

گرده افشانی محلی: در این گرده افشانی یک ضریب در نظر گرفته شده است که اندازه ناحیه رشد برای هر گل را در اطراف بهترین گل پیدا شده مشخص می‌کند. در این حرکت بهترین جواب پیدا شده ملاک حرکت است و دیگر

جوابها به سمت آن حرکت می‌نمایند. در ابتدای الگوریتم حرکت‌ها با گام بلندتر و در انتها با گام کوچک انجام می‌شود.

$$X_i^{t+1} = X_i^t + R (X_i^t - g^*) \quad (12)$$

X_{t+1} مکان راه حل در تکرار $t+1$ ام و g^* مکان بهترین موقعیت در تمام تکرارهای انجام شده، است. R ناحیه رشد است که با توجه به تکرارهای الگوریتم کاهش می‌یابد و در واقع گام‌های حرکت در ابتدای الگوریتم بیشتر بوده و با نزدیک شدن به انتهای الگوریتم کاهش می‌یابد تا اینکه همگرایی به بهینه مسئله انجام شود.

$$R = 2 e^{-\left(\frac{4t}{L}\right)^2} \quad (13)$$

در آن t شماره تکرار تکامل فعلی الگوریتم و L تعداد تکرار نهایی الگوریتم است.

جهت مدل‌سازی حرکت آب بروی برگ‌ها از حرکت قطرات آب به اولین گودال‌ها بروی برگ و پر شدن گودال‌ها و سرریز شدن آب از روی برگ استفاده می‌شود.

در مدل‌سازی این حرکت از قطره استفاده می‌شود. هر عضو از جمعیت یا راه حل یک قطره نام دارد که با یک موقعیت و سرعت اولیه در فضای جستجوی مساله برای یافتن راه‌حل بهینه حرکت می‌کند.

یک قطره پس از شکل‌گیری در برگ، توسط الگوریتم تپه‌نوردی در نزدیک‌ترین بهینه محلی به خود قرار خواهد گرفت. این بهینه محلی یگ گودال نامیده می‌شود. هر گودال با توجه به عمق (*fitness*) خود دارای یک گنجایش قرارگیری قطرات می‌باشد که در حین اجرای الگوریتم با توجه به ظرفیت خود، قطرات را در خود جای می‌دهد. عمیق‌ترین گودال به با ارزش‌ترین گودال (گودالی با بهترین *fitness*) در هر دور از اجرا گفته می‌شود.

در این مدل‌سازی جستجوی محلی هر قطره یک بردار سرعت در نظر گرفته می‌شود که مقدار اولیه آن، برابر با طول قدم اولیه‌ای می‌باشد که از ورودی دریافت می‌شود. در هر تکرار بردار جابجایی یک قطره با بردار سرعت آن جمع می‌شود و سپس به مقدار بردار سرعت آن اضافه می‌شود. معادله حرکت یک قطره بصورت معادله (۱۷) می‌باشد. در این مدل‌سازی جریان آب در سطح برگ وجود دارد. همانطور که بیان شد هر گودال دارای یک گنجایش می‌باشد که هر چه عمق یک گودال بیشتر باشد (دارای *fitness* به‌تری باشد) دارای گنجایش بیشتری خواهد بود. در هر دور از اجرا، گنجایش تمامی گودال‌ها بصورت زیر محاسبه خواهد شد:

$$c_i^t = \frac{(|f_i^t - f_{Max}|) \times const}{(|f_{Min} - f_{Max}|)} \quad (14)$$

که در آن f_i^t $fitness$ گودال i ام در تکرار t می‌باشد و f_{Max} اندازه بزرگ‌ترین $fitness$ بین گودال‌ها و f_{Min} اندازه کوچک‌ترین $fitness$ بین گودال‌ها می‌باشد. $Const$ یک عدد ثابت می‌باشد که نشان دهنده حداکثر ظرفیت یک گودال برای یک تابع هدف می‌باشد و c_i^t نیز گنجایش گودال i ام در تکرار t می‌باشد.

در هر دور از اجرا به یک مقدار رندم از میزان میانگین گنجایش گودال‌ها، به میزان قطرات موجود در هر گودال اضافه می‌شود. اگر میزان قطرات موجود در یک گودال از گنجایش آن گودال بیشتر بود آن گودال از بین گودال‌ها حذف می‌شود و میزان آب این گودال در سطح جاری می‌شود.

جهت حرکت آب سرازیر شده به سمت یک گودال با گنجایش بیشتر نسبت به گودالی که آب از آن سرازیر شده است، می‌باشد. این گودال از بین گودال‌های موجود با گنجایش بیشتر بصورت رندم با یک اولویت مشخص می‌شود. این اولویت به اینصورت می‌باشد که هرچه میزان گنجایش یک گودال بیشتر باشد احتمال انتخاب آن گودال بیشتر خواهد بود. معادله (۱۵) بیانگر احتمال انتخاب یک گودال بین گودال‌های موجود با گنجایش بیشتر می‌باشد:

$$Select_i^t = \frac{c_i^t}{\sum_{j=0}^k c_j^t} \quad (15)$$

که در آن $Select_i^t$ برابر احتمال انتخاب یک گودال در تکرار t ، c_i^t گنجایش گودال i ام در تکرار t و k برابر با تعداد گودال‌هایی می‌باشد که دارای گنجایش بیشتر نسبت به گودال شامل آب سرازیر شده می‌باشد.

پس از انتخاب گودال، آب سرازیر شده به سمت گودال انتخابی حرکت می‌کند. اگر در طی مسیر حرکت به گودالی برخورد که دارای گنجایش بیشتری نسبت به گودال مبدأ بود، به میزان آب سرازیر شده، به مقدار آب آن گودال اضافه می‌شود و حرکت قطع خواهد شد، در غیراینصورت آنقدر حرکت می‌کند که به گودال انتخاب شده برسد و به میزان آب سرازیر شده به مقدار آب آن گودال اضافه می‌شود. در پایان حرکت گودال مبدأ حذف خواهد شد.

هر چه اندازه گنجایش یک گودال بیشتر باشد با احتمال بیشتری برای دفعات بیشتر در اجرای الگوریتم مؤثر خواهد بود (حذف نخواهد شد) و همچنین با احتمال بیشتری جزء گودال انتخابی خواهد شد که آب‌های سرازیری به سمت آن حرکت می‌کنند.

در اینجا بر خلاف الگوریتم پرنندگان، ذرات (قطرات) دارای حافظه برای نگهداری بهترین موقعیت خود در حین جستجو نیستند و دانش راه‌حل‌های خوب برای هر قطره حفظ نمی‌شود. هر قطره قابلیت دستیابی به مکان فعلی هر قطره دیگر را دارد، بنابراین در هر لحظه مکان بهترین قطره مشخص می‌باشد.

رابطه ۱۶ بیانگر سرعت و موقعیت حرکت قطرات در هنگام جستجوی محلی و رابطه ۱۷ بیانگر سرعت و موقعیت حرکت قطرات در هنگام حرکت در سطح (سرریز آب از گودال) می‌باشد.

$$V_i^{t+1} = q \times V_i^t \quad \text{و} \quad X_i^{t+1} = X_i^t + V_i^{t+1} \quad (16)$$

$$V_i^{t+1} = V_i^t + \text{Rand}(X_{deep\ pit}^t - X_i^t) \quad \text{و} \quad X_i^{t+1} = X_i^t + V_i^{t+1} \quad (17)$$

در رابطه‌های ۱۶ و ۱۷، $X_{deep\ pit}^t$ موقعیت فعلی عمیق‌ترین گودال در تکرار t ، V_i^t سرعت فعلی قطره i در تکرار t و X_i^t موقعیت فعلی قطره i در تکرار t هستند. q ضریب افزایش سرعت می‌باشد.

مراحل الگوریتم اثر نیلوفر آبی در ادامه آمده است:

- ۱- تولید جمعیت گل‌های اولیه بصورت تصادفی
- ۲- ارزیابی موقعیت هر راه حل و محاسبه شایستگی آن
- ۳- بهترین راه حل (برازندگی بهتر) با g^* مشخص شود
- ۴- تعیین احتمال گرده‌افشانی (P) انجام می‌شود. به احتمال P بروزرسانی موقعیت راه حل با گرده‌افشانی سراسری و به احتمال $1-P$ بروزرسانی موقعیت راه حل با گرده‌افشانی محلی انجام شود.
- ۵- تا هنگامیکه به حداکثر تعداد تکرار الگوریتم نرسیده است مراحل زیر انجام شود، در غیراینصورت g^* نشان دهنده راه حل مسئله است.

۵-۱- برای تمام گل‌ها از ۱ الی n مراحل زیر انجام شود:

۵-۱-۱- تولید عدد تصادفی (rand) در بازه ۰ و ۱

۵-۱-۲- اگر rand کوچکتر از P است گرده افشانی سراسری با حرکت سنجاقک:

۵-۱-۲-۱- مقدار پارامترهای e, w, s, f, a, c را بروز شود.

۵-۱-۲-۲- موقعیت غذا و مهاجم بروز شود.

۵-۱-۲-۳- مقادیر S و A و C و F و E محاسبه و بروزرسانی شود. از طریق معادله ۱ الی ۵

۵-۱-۲-۴- بروزرسانی شعاع همسایگی راه حل‌ها

۵-۱-۲-۵- اگر راه حل مورد نظر در شعاع همسایگی بود

۵-۱-۲-۵-۱- بروزرسانی بردار سرعت با معادله ۶

۲-۵-۱-۲-۵- بروزرسانی بردار موقعیت با معادله ۷

۲-۶-۱-۲-۵- اگر راه حل مورد نظر در شعاع همسایگی نبود

۱-۲-۶-۱-۵- بروزرسانی بردار موقعیت با معادله ۸

۲-۷-۱-۲-۵- اگر میزان شایستگی راه حل جدید بهتر از منبع غذا می‌باشد، راه حل جدید را به عنوان منبع غذا قرار بده و اگر میزان شایستگی راه حل جدید بدتر از دشمن است، آن را جایگزین دشمن کن.

۲-۸-۱-۲-۵- اگر راه حلی از فضای جستجو خارج شد، حذف شده و مجدد تولید می‌شود.

۳-۱-۲-۵- اگر rand بزرگ‌تر مساوی P است گرده افشانی محلی:

۱-۳-۱-۵- بروزرسانی ناحیه رشد راه حل‌ها

۲-۳-۱-۵- موقعیت راه حل جدید از معادله ۱۲ مشخص می‌شود

۴-۱-۵- محاسبه برازندگی تابع بهینه‌سازی به ازاء موقعیت راه حل‌های جدید تولید شده

۵-۱-۵- اگر راه حل‌های جدید تولید شده از حالت قبل خود برازندگی بهتری دارند که جایگزین شده و راه حل حالت قبل حذف می‌شود در غیراینصورت راه حل جدید حذف می‌شود.

۶-۱-۵- برای تمام جواب‌های ممکن مراحل زیر انجام شود:

۱-۶-۱-۵- جستجوی محلی با معادله ۱۶ و محاسبه گنجایش هر گودال با معادله ۱۴.

۲-۶-۱-۵- اضافه کردن تصادفی آب به گودال‌ها

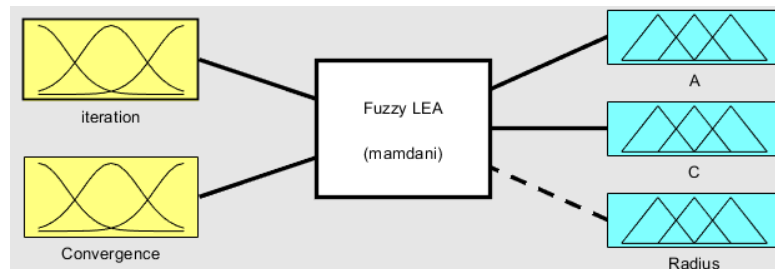
۳-۶-۱-۵- اگر میزان آب در گودال از ظرفیت آن بیشتر شد یک گودال بر اساس شایستگی با استفاده از معادله ۱۵ انتخاب می‌شود سپس آب به سمت گودال انتخابی حرکت می‌کند با معادله ۱۷.

۲-۵- بروزرسانی بهترین جواب پیدا شده (برازندگی بهتر) با g^*

۳- روش پیشنهادی (الگوریتم اثر نیلوفر آبی فازی)

در الگوریتم اثر نیلوفر آبی، نرخ اکتشاف و استخراج در حرکت سنجاقک‌ها با استفاده از شعاع، تنظیم می‌شود و این شعاع فقط وابسته به تعداد دور تکامل الگوریتم بوده و برای تنظیم آن پارامتر دیگری در نظر گرفته نشده است اما در مدل پیشنهادی پارامتر میزان همگرایی سنجاقک‌ها به نقطه بهینه نیز در نظر گرفته می‌شود. (میزان همگرایی بالای سنجاقک‌ها به یک ناحیه نشان دهنده آن است که در آن ناحیه بهینه سراسری قرار دارد و باید به استخراج بها داده شود و برعکس آن، همگرایی پایین سنجاقک‌ها به یک ناحیه در فضای جستجو نشان دهنده آن است که باید به اکتشاف نیز پرداخته شود). از طرفی دو حرکت اصلی که تأثیر مستقیم بر اکتشاف و استخراج دارد حرکت همترازی و حرکت

انسجام است. در حرکت سنجاکک ها پارامترهای a (ضریب برای هم تراز)، c (ضریب برای انسجام) و شعاع همسایگی از پارامترهای مهم در کنترل فرایند استخراج و اکتشاف هستند یعنی اینکه در فاز اکتشاف بایستی هم ترازى بالا، انسجام کم و شعاع کم باشد تا نواحی مختلف فضا جستجو شود و در فاز بهره برداری باید هم ترازى کم، انسجام بالا و شعاع بالا باشد تا همگرایی به بهینه مسئله با دقت بهتری ممکن شود، در روش پیشنهادی، پیشنهاد می شود که با استفاده از یک سیستم استنتاج فازی این سه پارامتر با توجه به تعداد تکرارهای الگوریتم و همچنین میزان همگرایی سنجاککها کنترل شود.



شکل ۱: ساختار کلی سیستم استنتاج فازی پیشنهادی

در سیستم استنتاج فازی پیشنهادی، میزان پراکندگی با استفاده از رابطه ۱۸ محاسبه می شود:

مقادیر تابع سنجاککها و همچنین مقدار بهترین سنجاکک بین صفر و یک نرمال شده و سپس محاسبه مقدار همگرایی (*Convergence*) با میانگین گیری و اختلاف مقدار هر سنجاکک تا مقدار بهترین سنجاکک محاسبه می شود:

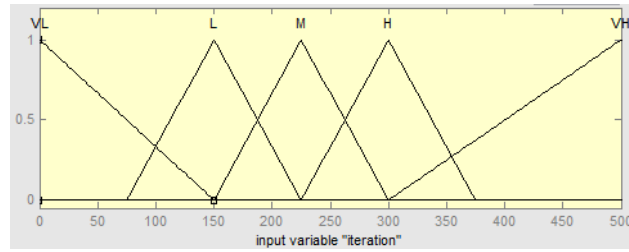
$$Convergence = 1 - \sum_{i=1}^n \left(\frac{\left| fitness\left(\frac{X_i^t - W}{T - W}\right) - fitness\left(\frac{T - W}{T - W}\right) \right|}{n} \right) \quad (18)$$

مقدار همگرایی با نزدیک شدن مقادیر تابع سنجاککها به مقدار بهترین سنجاکک، عدد نزدیک تری به یک را نشان می دهد و برعکس.

توابع عضویت ورودی ها:

ورودی اول: تکرار الگوریتم (*iteration*)

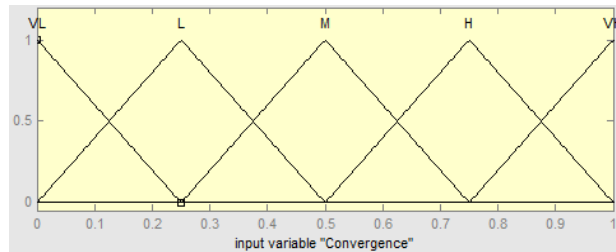
در ۵ حالت: خیلی کم (*VL*)، کم (*L*)، متوسط (*M*)، زیاد (*H*) و خیلی زیاد (*VH*)



شکل ۲: تابع عضویت تکرار الگوریتم به عنوان ورودی اول سیستم استنتاج فازی

ورودی دوم: میزان پراکندگی سنجاکها (*Convergence*)

در ۵ حالت: خیلی کم (VL)، کم (L)، متوسط (M)، زیاد (H) و خیلی زیاد (VH)

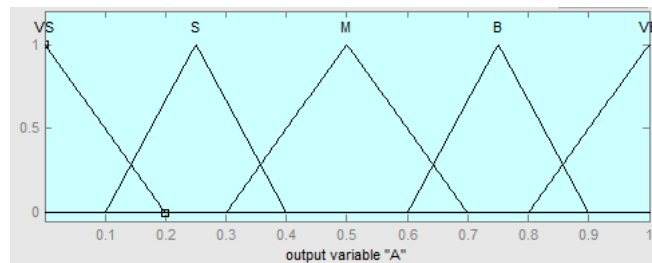


شکل ۳: تابع عضویت میزان پراکندگی سنجاکها به عنوان ورودی دوم سیستم استنتاج فازی

توابع عضویت خروجی ها:

خروجی اول: ضریب هم ترازی A

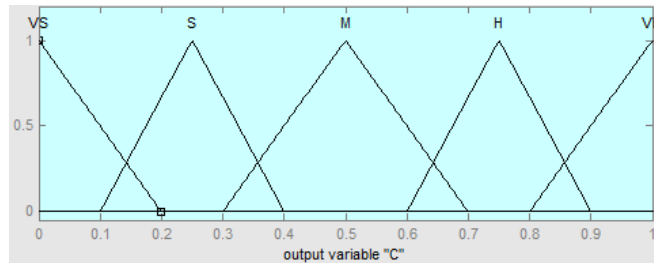
در ۵ حالت: خیلی بزرگ (VB)، بزرگ (B)، متوسط (M)، کوچک (S) و خیلی کوچک (VS)



شکل ۴: تابع عضویت ضریب هم ترازی A بعنوان خروجی اول سیستم استنتاج فازی

خروجی دوم: ضریب انسجام C

در ۵ حالت: خیلی بزرگ (VB)، بزرگ (B)، متوسط (M)، کوچک (S) و خیلی کوچک (VS)

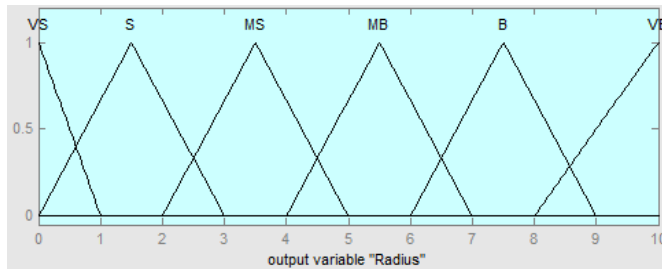


شکل ۵: تابع عضویت ضریب انسجام C بعنوان خروجی دوم سیستم استنتاج فازی

خروجی سوم: شعاع اطراف هر سنجاک

در ۶ حالت: خیلی بزرگ (VB)، بزرگ (B)، متوسط طرف بزرگ (MB)، متوسط طرف کوچک (MS)، کوچک (S)

و خیلی کوچک (VS)



شکل ۶: تابع عضویت شعاع اطراف هر سنجاک بعنوان خروجی سوم سیستم استنتاج فازی

نحوه تنظیم قوانین به این صورت است که در ورودی ها سیستم استنتاج فازی، میزان همگرایی و تکرار الگوریتم قرار دارد و میزان همگرایی دارای اولویت بیشتری از نظر تأثیر روی پارامترهای خروجی است زیرا میزان همگرایی نشان دهنده موقعیت سنجاکها و میزان همگرایی است ولی تعداد تکرار فقط با این پیش فرض است که چون تعداد تکرار رو به پایان است پس بایستی همگرایی اتفاق افتد. بنابراین تنظیم قوانین با تأثیر بیشتر از میزان همگرایی لحاظ شده است و بدین صورت که اگر میزان همگرایی کم است پس بایستی ضریب هم ترازوی بیشتر و ضریب انسجام و شعاع کمتر باشد و در صورتی که میزان همگرایی بالا است بایستی ضریب هم ترازوی کمتر و ضریب انسجام و شعاع بیشتر باشد.

شماره قوانین	ورودی ۱: تکرار الگوریتم	ورودی ۲: میزان همگرایی	خروجی ۱: ضریب هم ترازی	خروجی ۲: ضریب انسجام	خروجی ۳: شعاع
۱	VL	VL	VB	VS	VS
۲	VL	L	VB	S	VS
۳	VL	M	B	M	S
۴	VL	H	M	B	MS
۵	VL	VH	S	VB	MB
۶	L	VL	VB	VS	VS
۷	L	L	B	S	S
۸	L	M	B	M	MS
۹	L	H	M	B	MB
۱۰	L	VH	S	VB	B
۱۱	M	VL	B	VS	VS
۱۲	M	L	M	S	S
۱۳	M	M	M	M	MS
۱۴	M	H	M	B	B
۱۵	M	VH	S	VB	VB
۱۶	H	VL	B	VS	S
۱۷	H	L	M	S	MS
۱۸	H	M	S	M	MB
۱۹	H	H	S	B	B
۲۰	H	VH	VS	VB	VB
۲۱	VH	VL	B	VS	MS
۲۲	VH	L	M	S	MB
۲۳	VH	M	S	M	B
۲۴	VH	H	VS	B	VB
۲۵	VH	VH	VS	VB	VB

مراحل الگوریتم اثر نیلوفر آبی فازی در ادامه بیان می شود:

- ۱- تولید جمعیت گل های اولیه به صورت تصادفی
- ۲- ارزیابی موقعیت هر گل و محاسبه شایستگی آن
- ۳- بهترین گل (برازندگی بهتر) با g^* مشخص شود

۴- تعیین احتمال گرده افشانی (P) انجام می‌شود. به احتمال P بروزرسانی موقعیت گل با گرده افشانی سراسری و به احتمال $1-p$ بروزرسانی موقعیت گل با گرده افشانی محلی انجام شود.

۵- تا هنگامی که به حداکثر تعداد تکرار الگوریتم نرسیده است مراحل زیر انجام شود، در غیراینصورت g^* نشان دهنده راه حل مسئله است.

۵-۱- برای تمام گل‌ها از ۱ الی n مراحل زیر انجام شود:

۵-۱-۳- تولید عدد تصادفی (rand) در بازه ۰ و ۱

۵-۱-۴- اگر rand کوچکتر از p است گرده افشانی سراسری با حرکت سنجاک:

۵-۱-۴-۱- مقدار پارامترهای e, w, s, f را بروز کن و مقدار همگرایی سنجاک‌ها را از معادله ۱۸ محاسبه شود و

پارامترهای a و c و شعاع همسایگی را با سیستم استنتاج فازی بروزرسانی شود و موقعیت غذا و مهاجم سنجاک‌ها بروز شود.

۵-۱-۴-۲- مقادیر S و A و C و F و E محاسبه و بروزرسانی شود. از طریق معادله ۱۱ الی ۵

۵-۱-۴-۳- بروزرسانی شعاع همسایگی سنجاک‌ها

۵-۱-۴-۴- اگر سنجاک مورد نظر در شعاع همسایگی بود

۵-۱-۲-۵-۱- بروزرسانی بردار سرعت با معادله ۶

۵-۱-۲-۵-۲- بروزرسانی بردار موقعیت با معادله ۷

۵-۱-۴-۵- اگر سنجاک مورد نظر در شعاع همسایگی نبود

۵-۱-۲-۶-۱- بروزرسانی بردار موقعیت با معادله ۸

۵-۱-۴-۶- اگر میزان شایستگی سنجاک جدید بهتر از منبع غذا می‌باشد، سنجاک جدید را بعنوان منبع غذا قرار

بده و اگر میزان شایستگی سنجاک جدید بدتر از دشمن است، آنرا جایگزین دشمن کن.

۵-۱-۴-۷- اگر سنجاکی (محل گل) از فضای جستجو خارج شد، حذف شده و مجدد تولید می‌شود. تمامی

موقعیت سنجاک‌ها بعنوان موقعیت جدید گل‌ها در نظر گرفته می‌شود.

۵-۱-۵- اگر rand بزرگتر مساوی p است گرده افشانی محلی:

۵-۱-۵-۱- بروزرسانی ناحیه رشد گل‌ها

۵-۱-۵-۲- موقعیت گل جدید از معادله ۱۲ مشخص می‌شود

۵-۱-۶- محاسبه برازندگی تابع بهینه سازی به ازاء موقعیت گل‌های جدید تولید شده

۵-۲-۵- اگر گلهای جدید تولید شده از حالت قبل خود برازندگی به تری دارند که جایگزین شده و گل حالت قبل حذف می شود در غیراینصورت گل جدید حذف می شود.

۵-۲-۶- برای تمام جواب های ممکن مراحل زیر انجام شود:

۴-۱-۶-۵- جستجوی محلی با معادله ۱۶ و محاسبه گنجایش هر گودال با معادله ۱۴.

۵-۱-۶-۵- اضافه کردن تصادفی آب به گودال ها

۶-۱-۶-۵- اگر میزان آب در گودال از ظرفیت آن بیشتر شد یک گودال بر اساس شایستگی با استفاده از معادله ۱۵

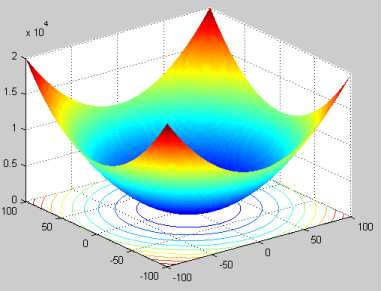
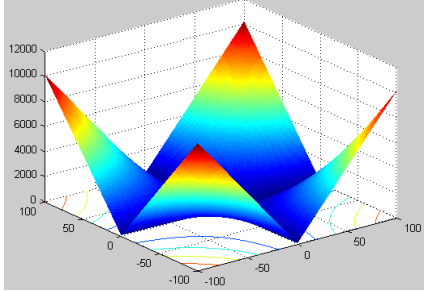
انتخاب می شود سپس آب به سمت گودال انتخابی حرکت می کند با معادله ۱۷.

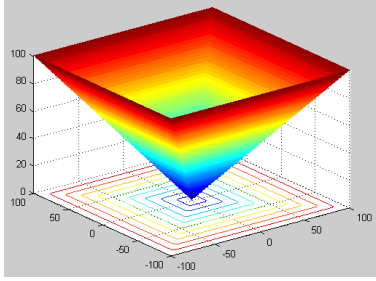
۳-۵- بروزرسانی بهترین جواب پیدا شده (برازندگی بهتر) با g^*

۴- نتایج

جهت آزمایش نتایج از دوازده تابع آزمایش استاندارد در ابعاد ۵۰ بعد استفاده شده است. در جدول ۱ شکل توابع و تابع آمده است. در شبیه سازی ها جمعیت اولیه هر دو روش ۳۰ عامل و تعداد تکرارنهایی الگوریتم ۵۰۰ بوده و نتایج بصورت میانگین از ۳۰ اجرای مختلف است.

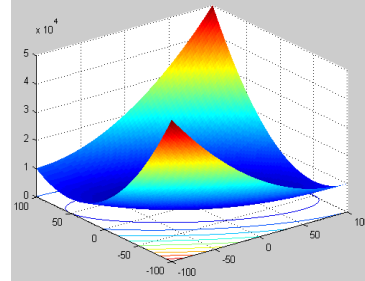
جدول ۱: شکل توابع آزمایش و معادله مربوطه

تابع بهینه سازی اول	تابع بهینه سازی دوم
 $F_1(x) = \sum_{i=1}^n x_i^2$	 $F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
تابع بهینه سازی سوم	تابع بهینه سازی چهارم



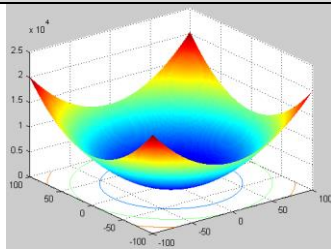
$$F_4(x) = \max\{|x_i|, 1 \leq i \leq n\}$$

تابع بهینه سازی ششم



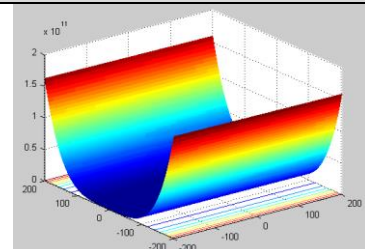
$$F_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^n x_j \right)^2$$

تابع بهینه سازی پنجم



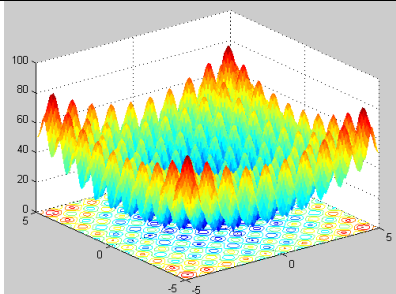
$$F_6(x) = \sum_{i=1}^n [(x_i + 0.5)]^2$$

تابع بهینه سازی هشتم



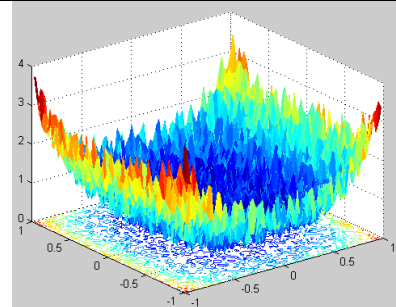
$$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

تابع بهینه سازی هفتم



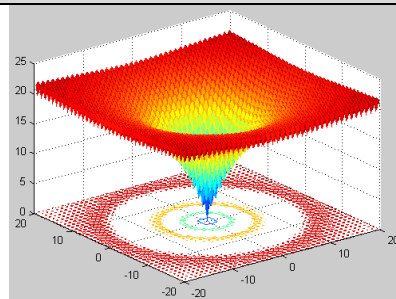
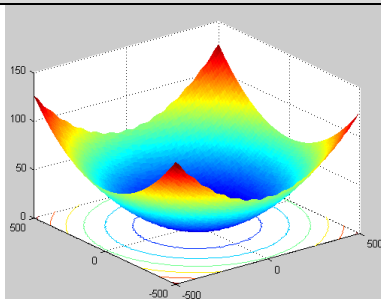
$$F_8(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$$

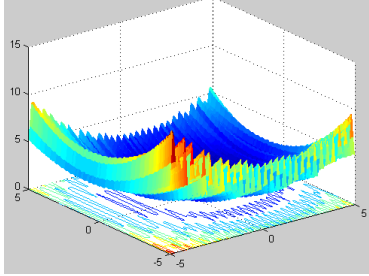
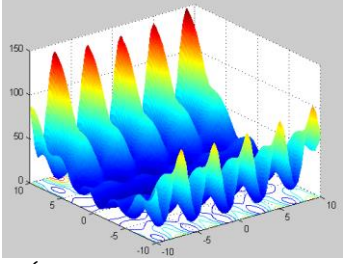
تابع بهینه سازی دهم



$$F_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}(0,1)$$

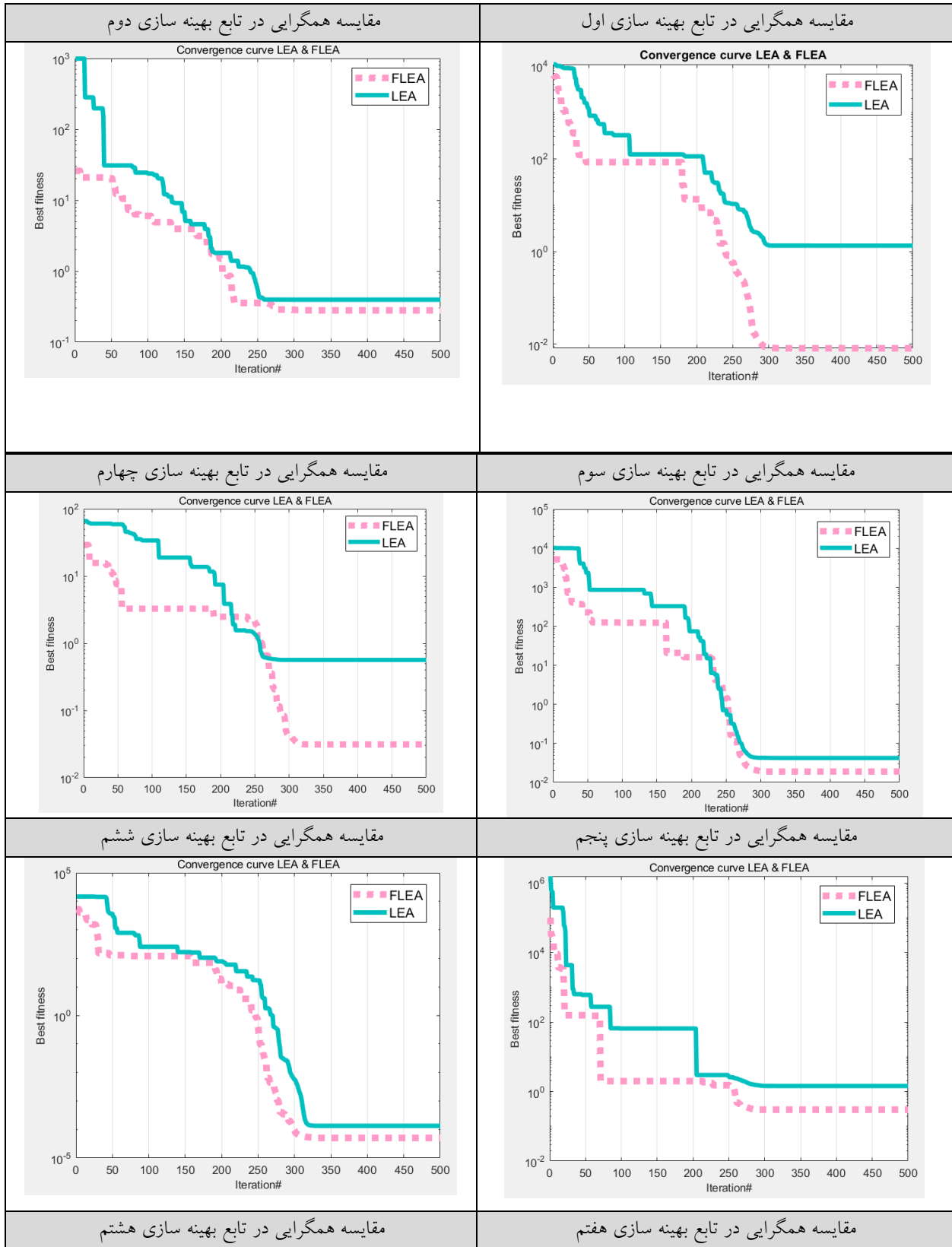
تابع بهینه سازی نهم

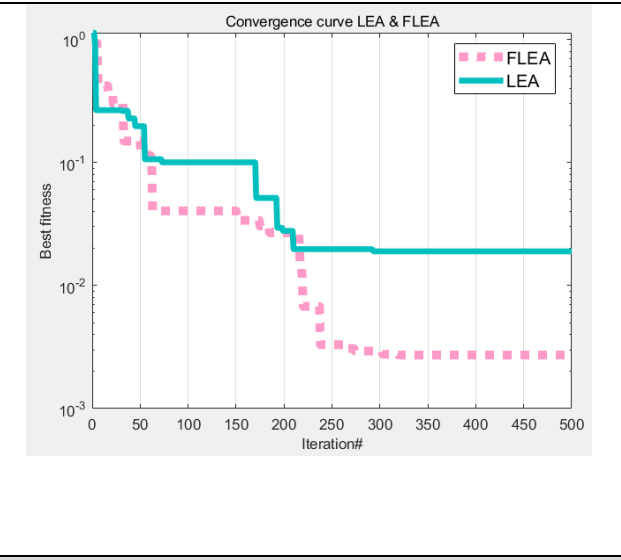
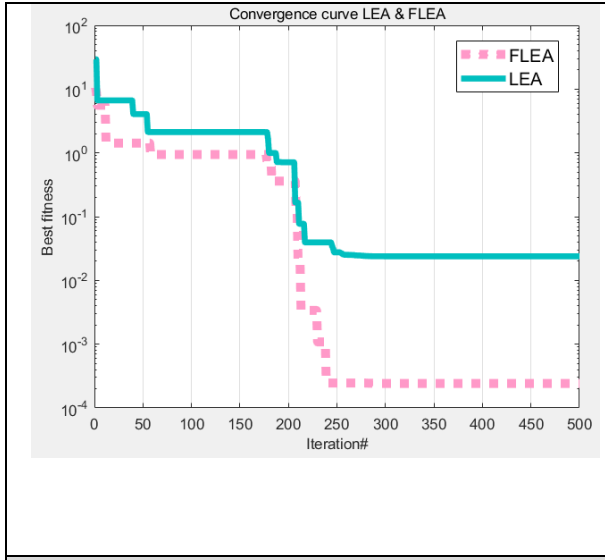


$F_{10}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$F_9(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$
<p>تابع بهینه سازی دوازدهم</p>	<p>تابع بهینه سازی یازدهم</p>
<div style="text-align: center;">  </div> $F_{12}(x) = 0.1 \left\{ \begin{aligned} &\sin^2(3\pi x_1) \\ &+ \sum_{i=1}^n (x_i - 1)^2 [1 \\ &+ \sin^2(3\pi x_i + 1)] \\ &+ (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \end{aligned} \right\} \\ + \sum_{i=1}^n u(x_i, 5, 100, 4)$	<div style="text-align: center;">  </div> $F_{11}(x) = \frac{\pi}{n} \left\{ \begin{aligned} &10 \sin(\pi y_1) \\ &+ \sum_{i=1}^{n-1} (y_i - 1)^2 [1 \\ &+ 10 \sin^2(\mu y_{i+1})] + (y_n - 1)^2 \end{aligned} \right\} \\ + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$

در جدول ۲ برخی از نمودارهای همگرایی روش پیشنهادی در مقایسه با الگوریتم اثر نیلوفر آبی آمده است

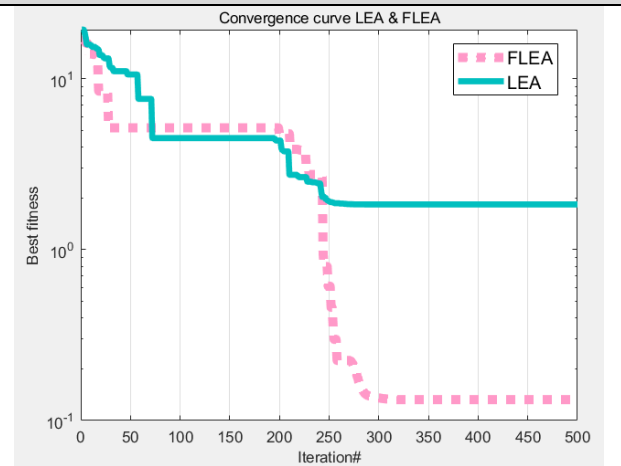
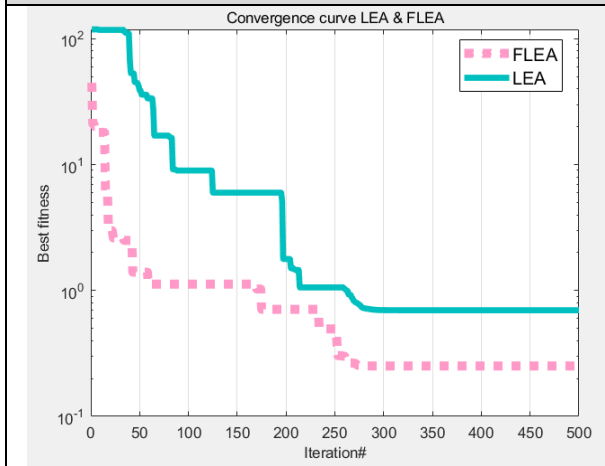
جدول ۲: نمونه از نمودارهای همگرایی روش پیشنهادی در مقایسه با الگوریتم اثر نیلوفر آبی





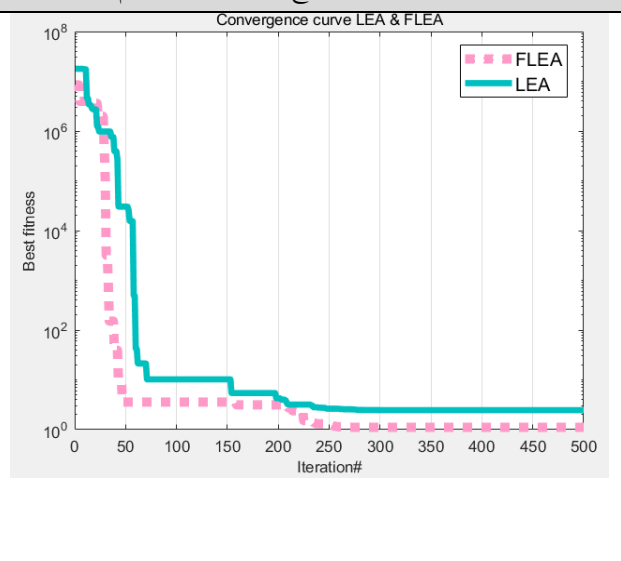
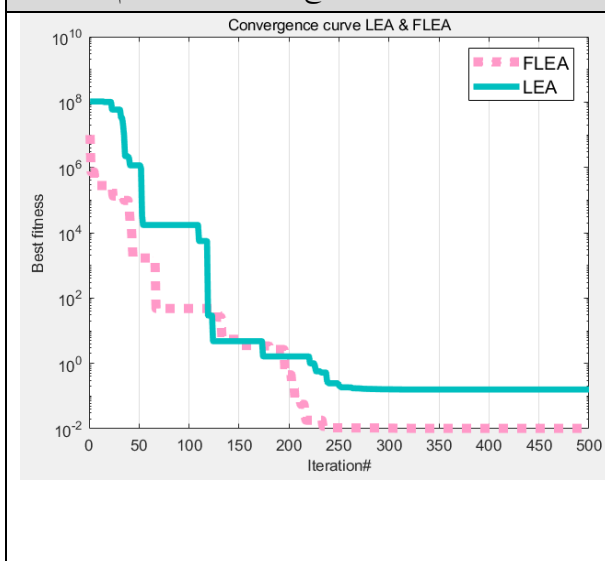
مقایسه همگرایی در تابع بهینه سازی دهم

مقایسه همگرایی در تابع بهینه سازی نهم



مقایسه همگرایی در تابع بهینه سازی دوازدهم

مقایسه همگرایی در تابع بهینه سازی یازدهم



بر اساس جدول ۲، نمودار همگرایی در روش پیشنهادی (FLEA) نسبت به روش LEA دارای قدرت همگرایی بالاتری است زیرا در آخرین تکرار به کمینه‌ترین مقدار همگرا شده است، همچنین بعلاوه این که در تکرارهای کمتری به دقت بهتر رسیده است، میزان سرعت همگرایی در روش پیشنهادی (FLEA) نسبت به روش LEA بالاتری بوده است.

میزان بهبود روش پیشنهادی در بدست آوردن مقدار دقیق نقطه بهینه و میزان ارزیابی تابع بر اساس مقادیر میانگین نسبت به روش LEA در جدول ۳ ارائه شده است.

جدول ۳: مقایسه نتایج روش پیشنهادی نسبت به روش LEA

روشها تابع	دقت بهینه سازی			میزان ارزیابی تابع بهینه سازی		
	LEA[۴]	FLEA	میزان بهبود به درصد	LEA[۴]	FLEA	میزان بهبود به درصد
تابع اول	0.015	0.0091	60.4	7661.25	7590.12	0.93
تابع دوم	0.0203	0.0095	46.3	7855.62	6313.14	24.43
تابع سوم	0.009	0.0013	14.12	8430.22	8112.6	3.91
تابع چهارم	0.0941	0.0104	11.12	8150.21	8050.47	1.23
تابع پنجم	0.1051	0.0904	86.2	8825.21	7625.94	15.72
تابع ششم	0.0211	0.0045	21.02	9895.02	9140.02	8.26
تابع هفتم	0.0058	0.0024	41.02	6885.02	6850.21	0.50
تابع هشتم	0.9024	0.2504	27.6	7501.22	6124.21	22.48
تابع نهم	0.8409	0.5032	59.1	7560.99	7440.65	1.61
تابع دهم	0.9057	0.8119	89.01	8101.6	7202.3	12.48
تابع یازدهم	1.0552	1.0021	96.01	6410.39	6321.5	1.40
تابع دوازدهم	1.9221	0.8011	41.01	7901.02	6452.21	22.45
		میانگین	49.25		میانگین	9.62

در جدول ۳ میزان بهبود نتایج با توجه به تغییرات دو مقدار بدست آمده برای هر دو روش محاسبه شده است و هر چه فاصله دو مقدار بیشتر باشد میزان بهبود، عدد بزرگتری خواهد بود. از نظر زمان اجرا بعلاوه این که بیشترین زمان محاسبه مربوط به اپراتورهای هر دو الگوریتم است و زمان انجام سیستم فازی در روش پیشنهادی زیاد نیست، بنابراین

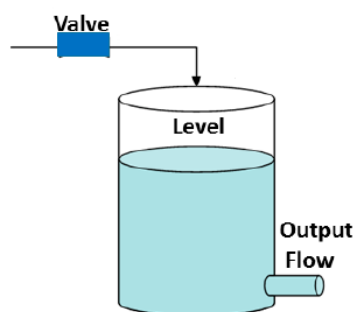
بطور میانگین زمان اجرای روش پیشنهادی نسبت به روش پایه الگوریتم اثر نیلوفر آبی ۷ درصد بیشتر است.

۵- نتایج بر روی مسائل کاربردی

در ادامه روش پیشنهادی بر روی دو مسئله کاربردی در بهینه سازی علوم مهندسی اعمال شده و نتایج آن با روش الگوریتم تکامل تفاضل فازی (FDE) [32]، مقایسه شده است. مسائل کاربردی شامل دو مسئله کنترل پارامترهای فازی در کنترلرهای مربوط به مسئله کنترل مخزن آب و مسئله پاندول معکوس می باشد.

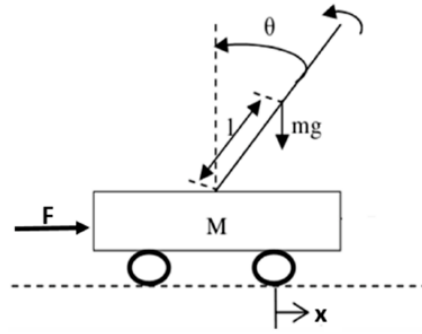
۵-۱- کنترل مخزن آب

در مسئله کنترل مخزن آب همانند شکل ۷ یک مخزن آب با سیستم فازی کنترل می شود (شکل ۸). در این کنترل مقدار دقیق مراکز توابع عضویت یک مسئله بهینه سازی است که روش پیشنهادی بهترین نقاط را برای این مسئله مشخص میکند تا بدین شکل کنترل مخزن بهینه تر باشد.

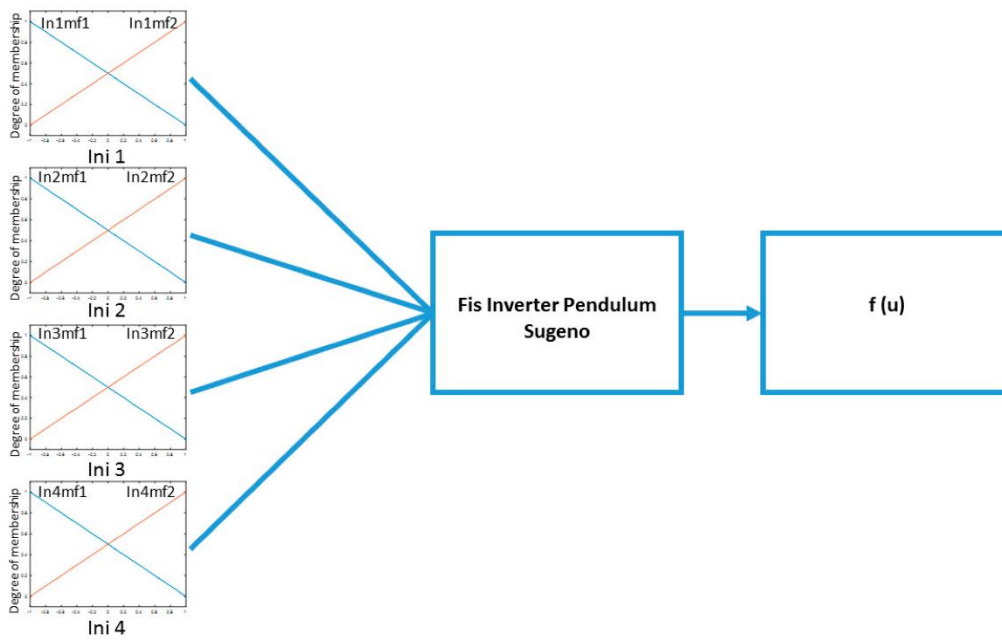


شکل ۷: شمایی از مسئله کنترل مخزن آب

همانطور که در شکل ۸ مشخص است ورودی فازی "سطح" دارای سه تابع عضویت مثلثی است که در مجموعه ۶ مرکز (مراکز به پایه های تابع مثلثی گفته می شود) برای این سه تابع عیت مثلثی وجود دارد. همچنین برای ورودی "نرخ" نیز همین تعداد وجود دارد و برای خروجی فازی که "مقدار" است، ۵ تابع مثلثی وجود دارد که در مجموع ۱۰ مرکز عضویت دارد.



شکل ۱۰: نمایی از مسئله سیستم پاندول معکوس



شکل ۱۱: شمایی از توابع عضویت مسئله سیستم پاندول معکوس

همانطور که در شکل ۱۱ مشخص است ورودی فازی "In1" تا "In4" بوده و سیستم فازی سوگینو است که در آن ورودی فازی و خروجی عدد است. بنابراین هر ورودی در مجموعه ۲ مرکز (پایه های تابع عضویت) برای این چهار ورودی در مجموعه ۸ مرکز وجود دارد و همچنین برای خروجی یک عدد است. بنابراین یک جواب ممکن در الگوریتم پیشنهادی یک آرایه با ۹ خانه است که شامل کلیه مراکز توابع عضویت و مقدار خروجی است (شکل ۱۲).

مراکز تابع عضویت ورودی اول	مراکز تابع عضویت ورودی دوم	مراکز تابع عضویت ورودی سوم	مراکز تابع عضویت ورودی چهارم	خروجی

شکل ۱۲: ساختار یک جواب ممکن در روش پیشنهادی برای مسئله کنترل مخزن آب

در نتایج با توجه به مقاله [32] در دو مسئله کنترل معیار، از ایجاد نویز نیز استفاده شده است و طبق معادله ۱۹ میزان خطا نشان دهنده دقت در بهینه سازی و کنترل بهینه دو مسئله است.

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (x_t - \hat{x}_t)^2} \quad (19)$$

نتایج گرفته شده در سی اجرای مختلف بصورت میانگین و انحراف معیار در مقایسه با بهترین نتایج مقاله الگوریتم تکامل تفاضل فازی (FDE) [32] در جدول ۴ و ۵ گزارش شده است.

جدول ۴: مقایسه نتایج از نظر میزان بهبود روش پیشنهادی در سی اجرا مختلف در مسئله مخزن آب

روشها معیار	دقت بهینه سازی بدون در نظر رفتن نویز			دقت بهینه سازی با در نظر گرفتن نویز		
	FDE	FLEA	میزان بهبود به درصد	FDE	FLEA	میزان بهبود به درصد
میانگین	۰,۰۹۴۴	۰,۰۸۹۴	۵,۵۹	۰,۳۶۷	۰,۳۴۲	۷,۳۱
انحراف معیار	۰,۰۳۳۴	۰,۰۳۱۴	۶,۳۷	۰,۵۶	۰,۵۱۱	۹,۵۹

جدول ۵: مقایسه نتایج از نظر میزان بهبود روش پیشنهادی در سی اجرا مختلف در مسئله پاندول معکوس

روشها معیار	دقت بهینه سازی بدون در نظر رفتن نویز			دقت بهینه سازی با در نظر گرفتن نویز		
	FDE	FLEA	میزان بهبود به درصد	FDE	FLEA	میزان بهبود به درصد
میانگین	۰,۲۱۵	۰,۱۹۸	۸,۵۹	۰,۳۷۵	۰,۳۵۶	۵,۳۴
انحراف معیار	۰,۲۷۵	۰,۲۵۱	۹,۵۶	۰,۱۹	۰,۱۷۹	۶,۱۵

همانطور که در جدول ۴ و ۵ مشخص است روش پیشنهادی توانسته در دو حالت بدون نویز و با نویز در مسئله مخزن آب بطور میانگین در حدود ۶,۵ درصد بهبود در دقت نتایج نسبت به الگوریتم تکامل تفاضلی فازی داشته باشد.

این بهبود در مسئله پاندول معکوس نیز بطور میانگین در حدود ۷ درصد است. از نظر انحراف معیار نتایج میتوان گفت که روش پیشنهادی با اختلاف کمتری نسبت به میانگین توانسته در سی اجرای مختلف همگرا شود.

۶- نتیجه گیری

در این مقاله راهکاری جدید برای بهبود نتایج الگوریتم اثر نیلوفر آبی ارائه شد و در آن از سیستم استنتاج فازی برای تاثیر بر فرآیند اکتشاف و استخراج استفاده شد. میزان شعاع و حرکت همترازی و حرکت انسجام در سیستم استنتاج فازی با توجه به تکرارهای الگوریتم و محاسبه میزان همگرایی عاملهای جستجو کنترل شده و بدین شکل میزان دقت و سرعت همگرایی الگوریتم اثر نیلوفر آبی ارتقاء یافت. در شبیه سازی ها ابعاد توابع آزمایش بالا (۵۰ بعد) در نظر گرفته شد تا مقایسه کارایی روش پیشنهادی مشخص باشد. نتایج الگوریتم اثر نیلوفر آبی بروی دوازده تابع آزمایش نشان داده که این در هر دور اجرا با دقت بسیار بالا به بهینه مسئله همگرا نشده است اما در شبیه سازی روش پیشنهادی مشاهده می شود که نسبت به روش پایه دارای دقت و سرعت همگرایی بهتری است، و این بهبود در دقت تا بیش از ۴۹ درصد و در سرعت همگرایی تا بیش از ۹ درصد کاهش در تعداد ارزیابی تابع بوده است. بهینه سازی در ابعاد بالا در مسائل بهینه سازی کاربردهای مختلف، یک چالش در حوزه بهینه سازی است و روش پیشنهادی نشان داده است که توانسته در این حوزه بخوبی استفاده گردد. همچنین روش پیشنهادی برای تنظیم توابع عضویت در دو مسئله کاربردی در بهینه سازی علوم مهندسی، دو مسئله کنترلی مخزن آب و پاندول معکوس آزمایش شد و براساس نتایج نسبت به روش تکامل تفاضلی فازی [۳۲] توانست در این دو مسئله بیش از ۶,۵ درصد بهبود حاصل نماید.

منابع

- [1] Y. J. Zheng, "Water wave optimization: A new nature-inspired metaheuristic," *Computers & Operations Research*, vol. 55, pp. 1–11, 2015.
- [2] S.W. Mahfoud, "A comparison of parallel and sequential niching method", in: *Proceedings of the Sixth International Conference on Genetic Algorithms, ICGA*, pp. 136–143, 1995.
- [3] Ali Asghar Heidari, Seyedali Mirjalili, Hossam Faris, Ibrahim Aljarah d, Majdi Mafarja, Huiling Chen, "Harris hawks optimization: Algorithm and applications", *Future Generation Computer Systems*, 2019.

- [4] Dalirinia, E., Jalali, M., Yaghoobi, M. et al,” Lotus effect optimization algorithm (LEA): a lotus nature-inspired algorithm for engineering design optimization” ,J Supercomput 80, pp.761–799, 2024.
- [5] Yassine Meraihi, Amar Ramdane-Cherif, Dalila Acheli & Mohammed Mahseur,” Dragonfly algorithm: a comprehensive review and applications” , Neural Computing and Applications, 2020.
- [6] Sree Ranjini,K.S.aS.Murugan, “Memory based Hybrid Dragonfly Algorithm for numerical optimization problems” Expert Systems with Applications, vol. 1, no.1, pp. 111-119, 2017.
- [7] Singh,”Elite opposition learning and exponential function steps-based dragonfly algorithm for global optimization”, Expert Systems with Applications, vol. 2, no.1, pp. 51-60, 2017.
- [8] Alaa Tharwat ,Thomas Gabel,Aboul Ella Hassanien ,” Parameter Optimization of Support Vector Machine Using Dragonfly Algorithm”, International Conference on Advanced Intelligent Systems and Informatics ,pp. 309-319, 2017.
- [9] Philip T. Daely , Soo Y. Shin,” Range based wireless node localization using Dragonfly Algorithm”, Eighth International Conference on Ubiquitous and Future Networks (ICUFN),pp. 55-61, 2016.
- [10] Gururaghav Raman,Gurupraanesh Raman,Chakkarapani Manickam,Saravana Ilango GanesanT,” Dragonfly Algorithm Based Global Maximum Power Point Tracker for Photovoltaic Systems”, International Conference in Swarm Intelligence,pp. 211-219,2016.
- [11] C. Hema , Sharmila Sankar, Sandhya,” Energy efficient cluster based protocol to extend the RFID network lifetime using dragonfly algorithm”, International Conference on Communication and Signal Processing (ICCSP), pp.111-121, 2016.
- [12] Velamuri Suresh ,Sreejith, “Generation dispatch of combined solar thermal systems using dragonfly algorithm” , Computing, pp.59–80, 2016.
- [13] MohammadJafari,Mohammad HosseinBayati Chaleshtari,” Using dragonfly algorithm for optimization of orthotropic infinite plates with a quasi-triangular cut-out”, European Journal of Mechanics - A/Solids,pp. 1-14, 2017.

- [14] Shounak Rushikesh Sugave , Suhas Haribhau Patil , B. Eswara Reddy,” DDF: Diversity Dragonfly Algorithm for cost-aware test suite minimization approach for software testing Sign In or Purchase”, International Conference on Intelligent Computing and Control Systems (ICICCS),pp. 34-45,2017.
- [15] Majdi M. Mafarja ; Derar Eleyan ; Iyad Jaber ; Abdelaziz Hammouri ; Seyedali Mirjalili,” Binary Dragonfly Algorithm for Feature Selection”, International Conference on New Trends in Computing Sciences (ICTCS), pp.87-95,2017.
- [16] Mohamed Abdel-Basset,Qifang Luo,Miao,Yongquan Zhou,” Solving 0–1 Knapsack Problems by Binary Dragonfly Algorithm”, International Conference on Intelligent Computing,pp. 491-502,2017.
- [17] Shikha Agrawal,Jitendra Agrawal,” A comparative study of fuzzy PSO and fuzzy SVD-based RBF neural network for multi-label classification”, Neural Computing and Applications, pp.60-67,2017.
- [18] Bharat Singh Rana , Laxmi Srivastava,” Optimal location and sizing of STATCOM using Fuzzy-PSO approach”, IEEE 6th International Conference on Power Systems , pp.47-56,2017.
- [19] YoucefSoufi,” Fuzzy-PSO controller design for maximum power point tracking in photovoltaic system”, International Journal of Hydrogen Energy, pp.30-36,2017.
- [20] Dey, Swati,” Rough-Fuzzy-GA-based design of Al alloys having superior cryogenic performance”, Materials and Manufacturing Processes, pp.87-95,2017.
- [21] Anuja S. Desai , D. P. Gaikwad,” Real time hybrid intrusion detection system using signature matching algorithm and fuzzy-GA”, pp.100-106,2016.
- [22] Kumar Abhishek , Saurav tta , Manoj Masanta , Siba Sankar Mahapatra,” Fuzzy embedded imperialist competitive algorithm (ICA) for multi-response optimization during machining of CFRP (Epoxy) composites”, International Conference on Advances in Mechanical, Industrial, Automation and Management Systems (AMIAMS), pp.33-39,2017.
- [23] EhsanAmiri,” Efficient protocol for ta clustering by fuzzy Cuckoo Optimization Algorithm”, Applied Soft Computing, pp.93-101,2016.
- [24] R.J.KuoB.S.,WibowoabF.E.Zulvia,” Application of a fuzzy ant colony system to solve the dynamic vehicle routing problem with uncertain service time”, Applied Mathematical Modelling, pp.111-121,2016.

[25] JoydipDhara,SurbhiArora," Designing fuzzy rule base using Spider Monkey Optimization Algorithm in cooperative framework", Future Computing and Informatics Journal, pp.1-6,2017.

[26] ArashChaghari," Fuzzy clustering based on Forest optimization algorithm", Journal of King Saud University - Computer and Information Sciences, pp.43-49,2016.

[27] JiaZheng PeiYiXin SunHong Zhang," Fuzzy energy management strategy for parallel HEV based on pigeon-inspired optimization algorithm", Science China Technological Sciences, pp.13-17,2017.

[28] Leticia Amador-AnguloOscar Castillo," Comparative Study of Metrics That Affect in the Performance of the Bee Colony Optimization Algorithm Through Interval Type-2 Fuzzy Logic Systems", North American Fuzzy Information Processing Society Annual Conference, pp.50-56,2017.

[29] M.Salehpour,A.Jamali," A new aptive differential evolution optimization algorithm based on fuzzy inference system", Engineering Science and Technology, an International Journal, pp.187-195,2017.

[30] Yan Huang, Wei Liu, Xiaogang Zang," Improving Fuzzy Rule Classifier with Brain Storm Optimization and Rule Modification", Artificial Intelligence (cs.AI); Neural and Evolutionary Computing (cs.NE).2024.

[31] D. Jing, "Harris Hawks Optimization Based Clustering With Fuzzy Routing for Lifetime Enhancing in Wireless Sensor Networks," in IEEE Access, vol. 12, pp. 12149-12163, 2024

[32] Castillo, O., Valdez, F., Soria, J., Amador-Angulo, L., Ochoa, P., & Peraza, C , " Comparative Study in Fuzzy Controller Optimization Using Bee Colony, Differential Evolution, and Harmony Search Algorithms. Algorithms," 12(1), 9.2018.

[1] Y.-J. Zheng, "Water wave optimization: a new nature-inspired metaheuristic," *Computers & Operations Research*, vol. 55, pp. 1-11, 2015.

- [2] S. W. Mahfoud, "A comparison of parallel and sequential niching methods," in *Conference on genetic algorithms*, 1995, vol. 136: Citeseer, p. 143.
- [3] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future generation computer systems*, vol. 97, pp. 849-872, 2019.
- [4] E. Dalirinia, M. Jalali, M. Yaghoobi, and H. Tabatabaee, "Lotus effect optimization algorithm (LEA): a lotus nature-inspired algorithm for engineering design optimization," *The Journal of Supercomputing*, vol. 80, no. 1, pp. 761-799, 2024.
- [5] Y. Meraihi, A. Ramdane-Cherif, D. Acheli, and M. Mahseur, "Dragonfly algorithm: a comprehensive review and applications," *Neural Computing and Applications*, vol. 32, no. 21, pp. 16625-16646, 2020.
- [6] S. R. KS and S. Murugan, "Memory based hybrid dragonfly algorithm for numerical optimization problems," *Expert Systems with Applications*, vol. 83, pp. 63-78, 2017.
- [7] J. Song and S. Li, "Elite opposition learning and exponential function steps-based dragonfly algorithm for global optimization," in *2017 IEEE International Conference on Information and Automation (ICIA)*, 2017: IEEE, pp. 1178-1183.
- [8] A. Tharwat, T. Gabel, and A. E. Hassanien, "Parameter optimization of support vector machine using dragonfly algorithm," in *Proceedings of the international conference on advanced intelligent systems and informatics 2017*, 2018: Springer, pp. 309-319.
- [9] P. T. Daely and S. Y. Shin, "Range based wireless node localization using dragonfly algorithm," in *2016 eighth international conference on ubiquitous and future networks (ICUFN)*, 2016: IEEE, pp. 1012-1015.
- [10] G. Raman, G. Raman, C. Manickam, and S. I. Ganesan, "Dragonfly algorithm based global maximum power point tracker for photovoltaic systems," in *Advances in Swarm Intelligence: 7th International Conference, ICSI 2016, Bali, Indonesia, June 25-30, 2016, Proceedings, Part I 7*, 2016: Springer, pp. 211-219.
- [11] C. Hema and S. Sankar, "Energy efficient cluster based protocol to extend the RFID network lifetime using dragonfly algorithm," in *2016 international conference on communication and signal processing (ICCSP)*, 2016: IEEE, pp. 0530-0534.
- [12] V. Suresh and S. Sreejith, "Generation dispatch of combined solar thermal systems using dragonfly algorithm," *Computing*, vol. 99, no. 1, pp. 59-80, 2017.
- [13] M. Jafari and M. H. B. Chaleshtari, "Using dragonfly algorithm for optimization of orthotropic infinite plates with a quasi-triangular cut-out," *European Journal of Mechanics-A/Solids*, vol. 66, pp. 1-14, 2017.

- [14] S. R. Sugave, S. H. Patil, and B. E. Reddy, "DDF: Diversity dragonfly algorithm for cost-aware test suite minimization approach for software testing," in *2017 international conference on intelligent computing and control systems (ICICCS)*, 2017: IEEE, pp. 701-707.
- [15] M. M. Mafarja, D. Eleyan, I. Jaber, A. Hammouri, and S. Mirjalili, "Binary dragonfly algorithm for feature selection," in *2017 International conference on new trends in computing sciences (ICTCS)*, 2017: IEEE, pp. 12-17.
- [16] M. Abdel-Basset, Q. Luo, F. Miao, and Y. Zhou, "Solving 0–1 knapsack problems by binary dragonfly algorithm," in *Intelligent Computing Methodologies: 13th International Conference, ICIC 2017, Liverpool, UK, August 7-10, 2017, Proceedings, Part III 13*, 2017: Springer, pp. 491-502.
- [17] S. Agrawal, J. Agrawal, S. Kaur, and S. Sharma, "A comparative study of fuzzy PSO and fuzzy SVD-based RBF neural network for multi-label classification," *Neural Computing and Applications*, vol. 29, no. 1, pp. 245-256, 2018.
- [18] B. S. Rana and L. Srivastava, "Optimal location and sizing of STATCOM using Fuzzy-PSO approach," in *2016 IEEE 6th International Conference on Power Systems (ICPS)*, 2016: IEEE, pp. 1-6.
- [19] Y. Soufi, M. Bechouat, and S. Kahla, "Fuzzy-PSO controller design for maximum power point tracking in photovoltaic system," *International Journal of hydrogen energy*, vol. 42, no. 13, pp. 8680-8688, 2017.
- [20] S. Dey, P. Dey, and S. Datta, "Rough-Fuzzy-GA-based design of Al alloys having superior cryogenic performance," *Materials and Manufacturing Processes*, vol. 32, no. 10, pp. 1075-1081, 2017.
- [21] A. S. Desai and D. Gaikwad, "Real time hybrid intrusion detection system using signature matching algorithm and fuzzy-GA," in *2016 IEEE international conference on advances in electronics, communication and computer technology (ICAECCT)*, 2016: IEEE, pp. 291-294.
- [22] K. Abhishek, S. Datta, M. Masanta, and S. S. Mahapatra, "Fuzzy embedded imperialist competitive algorithm (ICA) for multi-response optimization during machining of CFRP (Epoxy) composites," in *2017 International conference on advances in mechanical, industrial, automation and management systems (AMIAMS)*, 2017: IEEE, pp. 100-103.
- [23] E. Amiri and S. Mahmoudi, "Efficient protocol for data clustering by fuzzy cuckoo optimization algorithm," *Applied Soft Computing*, vol. 41, pp. 15-21, 2016.

- [24] R. Kuo, B. Wibowo, and F. Zulvia, "Application of a fuzzy ant colony system to solve the dynamic vehicle routing problem with uncertain service time," *Applied Mathematical Modelling*, vol. 40, no. 23-24, pp. 9990-10001, 2016.
- [25] J. Dhar and S. Arora, "Designing fuzzy rule base using spider monkey optimization algorithm in cooperative framework," *Future Computing and Informatics Journal*, vol. 2, no. 1, pp. 31-38, 2017.
- [26] A. Chaghari, M.-R. Feizi-Derakhshi, and M.-A. Balafar, "Fuzzy clustering based on Forest optimization algorithm," *Journal of King Saud university-computer and information sciences*, vol. 30, no. 1, pp. 25-32, 2018.
- [27] J. Pei, Y. Su, and D. Zhang, "Fuzzy energy management strategy for parallel HEV based on pigeon-inspired optimization algorithm," *Science China Technological Sciences*, vol. 60, pp. 425-433, 2017.
- [28] L. Amador-Angulo and O. Castillo, "Comparative study of metrics that affect in the performance of the Bee Colony Optimization Algorithm through interval type-2 fuzzy logic systems," in *Fuzzy Logic in Intelligent System Design: Theory and Applications*, 2018: Springer, pp. 61-72.
- [29] M. Salehpour, A. Jamali, A. Bagheri, and N. Nariman-Zadeh, "A new adaptive differential evolution optimization algorithm based on fuzzy inference system," *Engineering Science and Technology, an International Journal*, vol. 20, no. 2, pp. 587-597, 2017.
- [30] Y. Huang, W. Liu, and X. Zang, "Improving Fuzzy Rule Classifier with Brain Storm Optimization and Rule Modification," *arXiv preprint arXiv:2410.01413*, 2024.
- [31] D. Jing, "Harris hawks optimization based clustering with fuzzy routing for lifetime enhancing in wireless sensor networks," *IEEE Access*, 2024.
- [32] O. Castillo, F. Valdez, J. Soria, L. Amador-Angulo, P. Ochoa, and C. Peraza, "Comparative study in fuzzy controller optimization using bee colony, differential evolution, and harmony search algorithms," *Algorithms*, vol. 12, no. 1, p. 9, 2018.