دوره ۱۳، شماره بهار و تابستان ۱۳۹۹

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

ارائه یک الگوریتم هیبریدی از جستجوی کلاغ بهینه‌شده با سیستم فازی و الگوریتم جستجوی گرانشی و به کارگیری آن در آموزش شبکه عصبی رو به جلو

آزاده آروین مهر [۱] مهدی یعقوبی *[۲] مهرداد جلالی [۳]

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

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

(۳) گروه کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران

(تاریخ دریافت : ۱۳۹۹/۷/۱۲ تاریخ پذیرش : ۱۴۰۰/۴/۲۶)

چکیده

مبنای اصلی الگوریتم جستجوی کلاغ (CSA)، پنهان کردن غذا است. این الگوریتم همگرایی کمی دارد و در بهینه‌های محلی گیر می‌کند. الگوریتم جستجوی گرانشی(GSA) یک الگوریتم جدید بهبود یافته ابتکاری مبتنی بر قانون جاذبه و تعاملات جرم است و توانایی خوبی در بهینه سراسری دارد، اما سرعت جستجوی آن کند است و در تکرارهای آخر رنج می‌برد.دراین تحقیق، بهبود روش الگوریتم کلاغ در دو مرحله صورت گرفته است: در مرحله اول با یک سیستم فازی پارامترهای CSA تنظیم شده است و در مرحله دوم از الگوریتم گرانشی استفاده شده است. الگوریتم جستجوی کلاغ با فازی(*FCSA*) اصلاح ومقادیر خروجی آن به عنوان جمعیت اولیه به *GSA* داده می‌شود و خروجی این دو، مقادیر اولیه شبکه عصبی رو به جلو(*FNNs*) را مقداردهی می‌کند. دراین مقاله، *GSA*و*FCSAGSA* *برای*(*FNNs*) به‌کارگرفته میشوندو دقت حاصل از *FNNs* با *PSO*،*GSA*و*PSOGSA* بررسی می‌شودو نتایج نشان می‌دهدکه روش پیشنهادی عملکردبهتری نسبت به*PSO*،*GSA*و*PSOGSA* برای آموزش*FNNs* از لحاظ سرعت هم‌گرایی و اجتناب ازبهینه محلی دارد.با شبیه سازی نشان می‌دهیم که یک*FNNs* آموزش‌دیده با*FCSAGSA*خطای کمتری نسبت به*FNNs* آموزش‌دیده با *GSA*دارد.

کلمات کلیدی: الگوریتم جستجوی گرانشی، الگوریتم جستجوی کلاغ، بهینه‌سازی، شبکه‌های عصبی، شبکه عصبی رو به جلو

*عهده‌دار مکاتبات:
مهدی یعقوبی
نشانی: گروه برق، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران
پست الکترونیکی: Yaghoobi@mshdiau.ac.ir

۱- مقدمه

فرآیند بهبود سازی، یعنی یافتن بهترین راه حل از مجموعه راه حل‌های ممکن با توجه به تابع هدف و ارضا کردن برخی محدودیت‌های مسئله مورد حل می‌باشد [۱].امروزه، بسیاری از مسائلی که شرکت‌ها مواجه‌اند نیازمند فرموله کردن مسائل بهبودسازی می‌باشند .کاربردهای زیادی با چنین مسائل پیچیده‌ای سروکار دارند و حل بسیاری از این مسائل سخت است. در این مسائل، فضای جستجو بطور نمایی با اندازه مسئله رشد می‌نمایند و روش‌های پیشین راه حل مناسبی را برای آن‌ها ارائه نمی‌کنند و در پیدا کردن مسائل بهبود یافته در زمان مناسب با شکست مواجه می‌شوند. بنابراین در دهه‌های اخیر، تکنیک‌های بهبود سازی الهام گرفته شده از طبیعت و بیولوژی برای حل این مسائل توجه زیادی را جذب کرده است که این الگوریتم‌ها مبتنی بر جمعیت هستند. بسیاری از الگوریتم‌های مبتنی بر جمعیت تقلید رفتار، عملکرد و مشخصه‌های سیستم‌های زندگی موجودات خاص می‌باشند. شبکه‌های عصبی سیستم‌ها و روش‌های محاسباتی نوین برای یادگیری ماشینی، نمایش دانش و در انتها اعمال دانش به دست آمده در جهت پیش‌بینی پاسخ‌های خروجی از سامانه‌های پیچیده هستند [۲]. ایده اصلی این گونه شبکه‌ها تا حدودی الهام‌گرفته از شیوه کارکرد سیستم عصبی زیستی برای پردازش داده‌هاو اطلاعات به منظور یادگیری و ایجاد دانش قرار دارد. عنصر کلیدی این ایده، ایجاد ساختارهایی جدید برای سامانه پردازش اطلاعات است.

الگوریتم جستجوی کلاغ (CSA)الگوریتمی،که توسط عسکرزاده در سال ۲۰۱۶ ارائه شد. در این الگوریتم، جستجوی کلاغ همگرایی کم دارد و در بهینه‌های محلی گیر می‌کند[۳]. احتمال p این است که کلاغ بر چه اساسی حرکت کند اگر به صورت تصادفی انجام شود در دام بهینه محلی گیر کرده و به همگرای زود راس می‌رسد. دقت کاهش پیدا کرده و مدت زمان بالا میرود. برای رفع مشکل آن برای احتمال p از رویکرد فازی استفاده شده است. چالش رویکرد کلاغ را مرتفع کردیم.گوپتا و همکارانش در سال ۲۰۱۸، از الگوریتم جستجوی کلاغ بهبودیافته برای تشخیص بیماری پارکینسون استفاده کردند. روش پیشنهادی *OCSA*می‌تواند در پیش‌بینی بیماری پارکینسون برای دقت ۱۰۰٪ و کمک به فرد در درمان مناسب در مرحله اولیه استفاده شود. عملکرد *OCSA*بر روی ۲۰ مجموعه‌داده معیار اندازه‌گیری شده و نتایج با الگوریتم جستجوی کلاغ استاندارد مقایسه شده است. نتایج نشان‌دهنده حداکثر دقت و به حداقل رساندن تعداد ویژگی‌های انتخاب‌شده و پایداری روش پیشنهادی است [۱]. اصلی و همکارانش در سال ۲۰۱۸، به تشریح کامل الگوریتم جستجوی کلاغ استاندارد ارائه شده توسط عسکرزاده در سال ۲۰۱۶ در یک فصل از کتاب خود پرداخته‌اند [۴]. نیاک و همکارانش در سال ۲۰۱۸، یک رویکرد مبتنی بر بهینه‌سازی جستجوی کلاغ برای یافتن مشکل طبقه‌بندی در داده‌کاوی پیشنهاد کردند. آنها الگوریتم جستجوی کلاغ با شبکه عصبی پیوندی را برای حل مسائل طبقه‌بندی مورد استفاده قرار دادند. نتایج روش پیشنهادی با سایر روش‌های مبتنی بر ازدحام مقایسه شد و نتایج تجربی نشان‌دهنده برتری روش پیشنهادی نسبت به سایر روش‌ها بوده است [۵].

الگوریتم جستجوی گرانشی (GSA) یکی از جدیدترین الگوریتم‌های جستجوی اکتشافی جمعیت محور می‌باشد. این الگوریتم براساس وزن اجسام میباشد. نیاز به الگوریتمی است که بتواند رویکرد و پارامترهایی را براساس وزن به دست بیاورد. وزن‌ها یا همان توده‌های سنگین با راه حل‌های خوب مسئله ارتباط دارند.جمعیت بهبود یافته از طریق الگوریتم کلاغ فازی را به الگوریتم گرانشی داده شده است. تا افتادن در دام بهینه محلی و همگرایی زود راس را بر طرف کند. هوانگ در سال ۲۰۱۵ از الگوریتم جستجوی گرانشی و *Back-propagation*برای آموزش شبکه‌های عصبی رو به جلو استفاده کرد. الگوریتم جستجوی گرانشی همانند اکثر الگوریتم‌های دیگر، این الگوریتم دارای توانایی خوبی برای جستجوی مطلوب جهانی است، اما سرعت جستجوی آن کم است. برعکس، الگوریتم*Back-propagation*می‌تواند به سرعت همگرایی را در اطراف مطلوب جهانی به دست آورد. نتایج تجربی نشان می‌دهد که الگوریتم ترکیبی پیشنهاد شده از *GSA*و *BP*آموزش*FNNs*بهتر می‌کند[۶].شبکه عصبی رو به جلو (*FNN*) یک شبکه عصبی مصنوعی است، که در آن اتصال میان واحدهای تشکیل دهنده آن یک چرخه را تشکیل نمی‌دهند. در واقع این شبکه متفاوت از شبکه‌های عصبی بازگشتی می‌باشد. شبکه عصبی رو به جلو اولین و ساده‌ترین نوع شبکه عصبی مصنوعی می‌باشد. در این شبکه اطلاعات تنها از یک مسیر حرکت می‌کند که جهت آن رو به جلو می‌باشد. در واقع اطلاعات باشروع از گره (نورون)های ورودی و گذر از لایه‌های پنهان (درصورت وجود) به سمت گره‌های خروجی می‌روند. همانطور که گفته شد در این شبکه حلقه یا دوری وجود ندارد. با این حال در این شبکه‌ها به دلیل انتخاب تصادفی اولیه وزن ورودی ممکن است شبکه به خوبی آموزش نبیند و خروجی آن جواب بهبود یافته ای نباشد. شبکه‌های رو به جلو دارای ساختاری لایه به لایه هستند و در این ساختار همه اتصالات از یک لایه به لایه دیگر است و هیچ راهی از لایه به خودش وجود ندارد [۷]. در مباحث مربوط به شبکه‌های عصبی، قواعد یادگیری، رویه‌ای برای اصلاح وزن‌ها و بایاس‌ها تعریف می‌کنیم [۸]. قاعده یادگیری در راستای آموزش شبکه برای انجام کار خاصی مورد استفاده قرار می‌گیرد. در فاز آموزش به دلیل مقداردهی تصادفی ممکن است خطا صورت دهد. به همین منظور استفاده از روش ترکیبی الگوریتم جستجوی کلاغ فازی و الگوریتم جستجوی گرانشی را برای این منظور پیشنهاد می‌دهیم.

میرجلالی و همکارانش در سال ۲۰۱۸ الگوریتمی جدید برای رفع مشکلات الگوریتم بهینه‌سازی پروانه (افتادن در دام بهینه محلی و همگرایی زودرس) ارائه کردند. آن‌ها همچنین از الگوریتم ارائه شده خود برای آموزش شبکه‌های عصبی رو به جلو استفاده کردند. نتایج نشان دهنده برتری روش پیشنهادی و اثبات شایستگی این الگوریتم برای حل مشکلات چالش برانگیز است [۲]. میرجلالی و همکارانش در سال ۲۰۱۸ یک الگوریتم آموزش جدید را با نام نهنگ آبی که اخیرا پیشنهاد شده است را برای برای آموزش شبکه‌های عصبی مورد استفاده قرار دادند. ثابت شده است که الگوریتم بهینه سازی نهنگ آبی توانایی حل طیف گسترده‌ای از مشکلات بهینه‌سازی را دارد و از الگوریتم‌های فعلی بهتر عمل می‌کند. نتایج کمی و کیفی ثابت می‌کند که روش پیشنهاد شده می‌تواند الگوریتم‌های فعلی را در اکثریت مجموعه داده‌ها از لحاظ سرعت اجتناب ازبهینه محلی و

سرعت همگام سازی محلی بهتر کند [۷]. ژو و همکارانش در سال ۲۰۱۶ از الگوریتم *Symbiotic organisms search* برای آموزش شبکه‌های عصبی رو به جلو استفاده کردند. یک الگوریتم قدرتمند و فرا ابتکاری است که باعث تحریک استراتژی‌های همبستگی متقابل بین ارگانیسم‌ها برای زنده ماندن و گسترش در اکوسیستم می‌شود. نتایج نشان می‌دهد که *SOS* بهتر از الگوریتم‌های دیگر برای آموزش *FNN*ها از لحاظ سرعت همگام‌سازی است. همچنین ثابت شده است که یک *FNN* که توسط روش *SOS* آموزش دیده است دقت بیشتری نسبت به اکثر الگوریتم‌های مقایسه شده دارد [۸].

ادامه ی مقاله به این صورت است. بخش ۲ خلاصه ای از الگوریتم جستجوی کلاغ، الگوریتم جستجوی گرانشی، شبکه‌های عصبی رو به جلو را نشان می‌دهد. بخش ۳ روش پیشنهادی را ارائه می‌دهد. بخش ۴ نتایج آزمایش را تحلیل می‌کند و بخش ۵ نتیجه را گزارش می‌دهد.

۲- تعاریف اولیه

با افزایش ابعاد و محدودیت‌های مسائل بهینه‌سازی، مدل‌های ریاضی آن‌ها نیز پیچیده‌تر شده و حل آن‌ها به کمک ابزارهای معمول، سخت‌تر و بسیار زمان‌بر شده است. به همین دلیل الگوریتم‌های تکاملی از جمله الگوریتم ژنتیک، الگوریتم کلونی مورچگان، الگوریتم زنبور عسل و ... برای حل مسائل بهینه‌سازی در طی سال‌های اخیر ارائه شده است. در این بخش ابتدا به تشریح الگوریتم جستجوی کلاغ، الگوریتم جستجوی گرانشی، شبکه‌های عصبی رو به جلو و کارهای صورت گرفته در سال‌های اخیر می‌پردازیم.

۲-۱ الگوریتم جستجوی کلاغ

الگوریتم جستجوی کلاغ (*CSA*) الگوریتمی فرا ابتکاری است که توسط عسکرزاده در سال ۲۰۱۶ ارائه شد. مبنای اصلی این الگوریتم مکانیسم جستجوی کلاغ برای پنهان کردن غذا است. کلاغ یکی از باهوش‌ترین پرندگان است. نسبت به اندازه‌ی بدنشان مغز بزرگی دارند. کلاغ‌ها باهوش هستند. مغز آنها از مغز انسان کمی کمتر است. بعلاوه آنها تست آینه را رد کرده‌اند؛ می‌توانند چهره‌ها را به یاد بیاورند. وقتی چهره‌ی غیردوستانه بیایند، به شیوه‌ای پیچیده به دیگر کلاغ‌ها اطلاع می‌دهند. همچنین، می‌توانند غذای خود را پنهان و پس از چند ماه بیاد بیاورند. کلاغ‌ها را دزد می‌خوانند زیرا غذا را از دیگر پرندگان می‌دزدند. آن‌ها از تجربه‌ی خود به عنوان دزدی را برای پیش‌بینی رفتار دزدها استفاده می‌کنند. آن‌ها بسیار محتاط هستند. وقتی کلاغ دزدی را اعلام می‌کند، آنها مکان پنهان کردن غذا را تغییر می‌دهند. این کار باعث می‌شود آن‌ها ضرر نکنند [۳].

۴ اصل *CSA* عبارتند از [۴]:

- کلاغ‌ها به صورت گروهی زندگی می‌کنند.
- کلاغ‌ها مخفیگاه غذا را حفظ می‌کنند.
- هر عضو گروه فرد دیگر را هنگام دزدی دنبال می‌کند.
- کلاغ‌ها در هنگام دزدی محتاطند؛ آن‌ها از مخفیگاه خود محافظت می‌کنند.

۲-۱-۱ مدل ریاضی الگوریتم جستجوی کلاغ

M تعداد کلاغ‌ها ست. D تعداد کل بُعدها است، $y^{j,t}$ موقعیت کلاغ j را در تکرار t در فضای جستجو نشان می‌دهد که در آن $j = 1,2,\dots M$ تعداد $tMax$ بیشینه‌ی تکرارها است [۵].

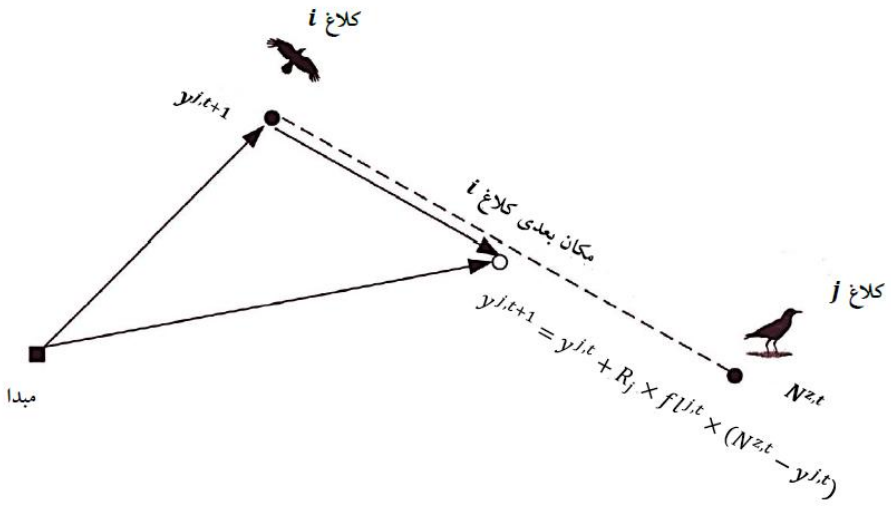
هر کلاغ حافظه‌ای دارد که در آن موقعیت مخفیگاه وجود دارد. $N^{j,t}$ موقعیت مخفیگاه در تکرار t برای کلاغ j است.

$N^{j,t}$ عبارت است از بهترین موقعیت بدست آمده توسط کلاغ j.

در تکرار t، کلاغ j می‌خواهد مخفیگاه کلاغ Z را دنبال کند. در این صورت دو احتمال وجود دارد [۹]:

۱)کلاغ Z نمی‌داند کلاغ j او را دنبال می‌کند، کلاغ j به مخفیگاه کلاغ z نزدیک می‌شود. موقعیت کلاغ j عبارت است از:

$$y^{j,t+1} = y^{j,t} + R_j \times fl^{j,t} \times \left(N^{z,t} - y^{j,t}\right) \qquad (1)$$

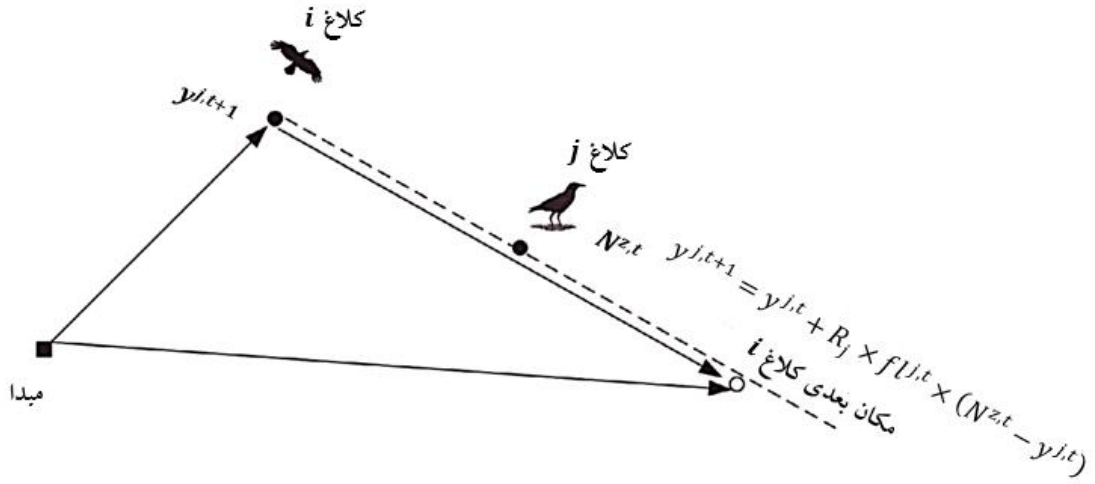


شکل ۱ : $fl < 1$ [۱۳]

که در آن fl طول پرواز را نشان می‌دهد. R_j عددی تصادفی $[۰,۱] \in$ است. fl تاثیر زیادی بر جستجوی محلی دارد در حالی که حجم بالا منجر به جستجوی سراسری می‌شود.

۲) کلاغ z می‌داند کلاغ j او را دنبال می‌کند، کلاغ z موقعیتش را در فضای جستجو تغییر می‌دهد تا از مخفیگاه حفاظت کند. دو مورد قبلی به صورت ریاضی می‌شوند [۱۰]:

$$y^{j,t+1} = \begin{cases} y^{j,t} + R_j \times fl^{j,t} \times \left(N^{z,t} - y^{j,t}\right), R_z \geq AP^{j,t} \\ Choose\ a\ rand\ Postition, \quad otherwise \end{cases} \qquad (2)$$



شکل ۲ : $fl < 1$ [۱۳]

که در آن R_z عدد تصادفی $[۰,۱] \in$ و $AP^{j,t}$ احتمالی آگاهی کلاغ z در تکرار t است. AP بالانس بین جستجو و کاوش را کنترل می‌کند. مقادیر کم AP باعث جستجو در موقعیت‌های محلی می‌شود (کاوش) در حالی که مقادیر بالا باعث جستجوی سراسری در فضای جستجو می‌شود (اکتشاف).

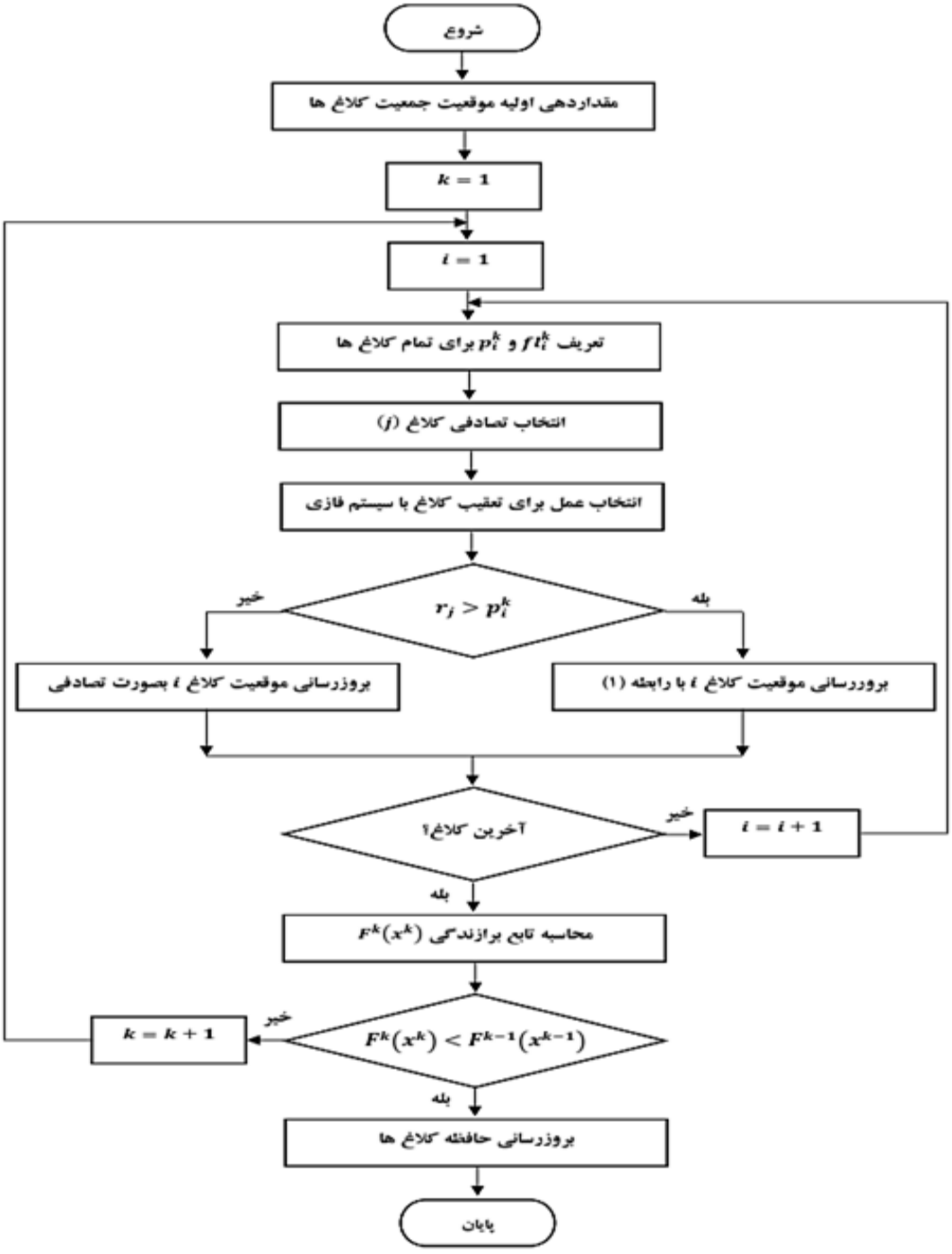
الگوریتم جستجو کلاغ با ایجاد محدوده‌های $D, tMax, M, AP$ و fl شروع می‌شود. موقعیت هر کلاغ y بطور تصادفی در فضای جستجو مقداردهی می‌شود. در آغاز، کلاغ‌ها تجربه‌ای برای پنهان کردن غذا ندارند. پس غذاها را در موقعیت‌های اولیه N مخفی می‌کنند. در طول اجرای الگوریتم، هر کلاغ با استفاده از تابع تناسب از پیش تعریف شده ارزیابی می‌شود. سپس طبق مقادیر تناسب، کلاغ‌ها موقعیت‌های خود را با معادله‌ی ۳ بروزرسانی می‌کنند. هر موقعیت جدید برای احتمال چک می‌شود [۱۱]. کلاغ‌ها حافظه ی خود را به این صورت بروز رسانی می‌کنند:

$$N^{j,t+1} = \begin{cases} y^{j,t+1}, Fn\left(y^{j,t+1}\right)\ is\ better\ than\ Fn(N^{j,t}) \\ N^{j,t}, \quad ohterwise \end{cases} \qquad (3)$$

که در آن $Fn()$ تابع هدف است.

تا زمانی که معیارهای *termination* رعایت می‌شوند، بهترین موقعیت به عنوان راه حل بهینه گزارش می‌شود.

روند نمای الگوریتم جستجوی کلاغ در شکل (۳) ارائه شده است [۱۲]:



شکل(۳) :روند نمای الگوریتم جستجوی کلاغ

۲-۲ الگوریتم جستجوی گرانشی

در سال ۲۰۰۹ رشیدی و همکارانش الگوریتم جستجوی گرانشی را ارائه کردند [۶]. الگوریتم جستجوی گرانشی از جمله الگوریتم‌های ابتکاری است که الهام گرفته از نیروی جاذبه و قوانین حرکتی نیوتن می‌باشد. پارامترهای مورداستفاده دراین الگوریتم عبارتند از: جرم، نیروی گرانشی، بهترین و بدترین مقدار (هدف مینیمم‌سازی باشد بیشترین برازندگی بدترین وکمترین برازندگی بهترین درنظرگرفته می‌شود و اگر هدف ماکزیمم‌سازی باشد برعکس می‌باشد). هرعامل، محل و وضعیت سایرعامل‌هارا ازطریق قانون جاذبه گرانشی درک می‌کند.قوانین گرانشی ابزاری برای تبادل اطلاعات میان عامل‌هاست. جرم عامل‌ها باتوجه به تابع هدف تعیین می‌شود. میزان شباهت هرعامل به سایرین به صورت فاصله بیان می‌شود. این الگوریتم در برخی از مسائل بهینه‌سازی دچار همگرایی زودرس شده و در بهینه محلی گیر می‌کند و پیشرفتی برای پیدا کردن جواب بهینه ندارد که این مشکل جزو معایب این الگوریتم محسوب می‌شود.

توده‌های سنگین یا راه‌حل‌های خوب مسئله ارتباط دارند موقعیت عامل نشان‌دهنده یک پاسخ پتانسیل مسئله می‌باشد و حجم آن با استفاده از تابع تناسب مشخص می‌گردد. با گذشت زمان توده‌ها توسط سنگین‌ترین حجم جذب‌شده که مربوط به یک راه‌حل بهینه در فضای جستجو می‌گردند.

$$F_i^d(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon}(4)$$

که M_{aj} توده فعال گرانشی مرتبط با عامل j و M_{pi} توده گرانشی منفعل که در ارتباط با عامل i است و $G(t)$ پدر ثابت گرانشی در زمان t است و ε یک ثابت کوچک است و $R_{ij}(t)$ فاصله اقلیدسی بین دو عامل iو j است.

$$G(t) = G_0 \times \exp\left(-\alpha \times \frac{iter}{maxiter}\right)(5)$$

که α ضریب نزولی، G_0 ثابت گرانشی اولیه و $iter$ تکرار فعلی و $maxiter$ ماکزیمم تکرار است.

$$R_{ij}(t) = \left\|X_i(t).X_j(t)\right\|_2(6)$$

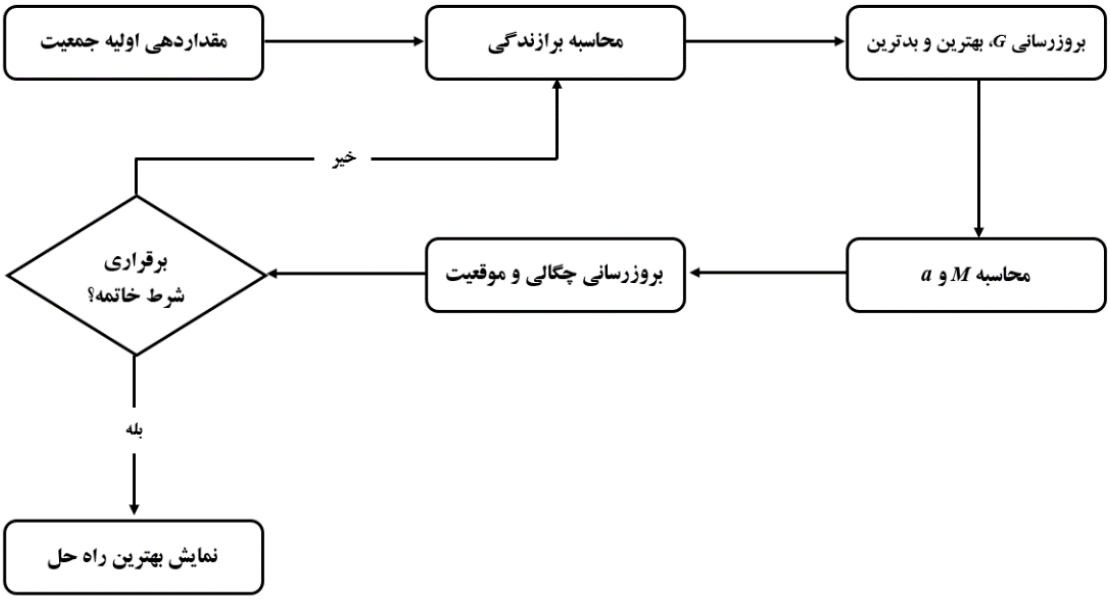
$$a_i^d = \frac{F_i^d(t)}{M_{ii}(t)'}(7)$$

d ابعاد مسئله و t زمان خاص و M_{ii} توده شی i است.

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)}(8)$$

$$m_i(t) = \frac{fit_i - worst\,(t)}{best(t) - worst(t)}(9)$$

که fit_i ارزش برازندگی عامل i در زمان t است، $best(t)$ بهترین عامل در زمان t است و $worst\ (t)$ بدترین عامل در زمان t است.



شکل ۴ روند نمایی الگوریتم جستجوی گرانشی[۶]

۲-۳ شبکه‌های عصبی رو به جلو

شبکه‌های رو به جلو شبکه‌هایی هستند که مسیر پاسخ در آنها، همواره رو به جلو پردازش می‌شود و به نورون‌های لایه قبل باز نمی‌گردد. این نوع شبکه‌ها به سیگنال‌ها اجازه می‌دهند تنها از مسیر یک طرفه یعنی از ورودی به خروجی عبور کنند. در دسته اول ساده‌ترین فرم شبکه عصبی رو به جلو، شبکه‌های عصبی رو به جلو تک لایه وجود دارد که یک لایه ورودی از گره‌های منبع وجود دارد که در نهایت از طریق وزن‌ها به نورون‌های لایه خروجی متصل می‌شوند. شبکه تک لایه نمونه‌ای از این شبکه است که هر دو لایه ورودی و خروجی ۴ نورون دارند . که اصطلاح تک لایه، به لایه خروجی اشاره دارد [۷]. دسته دوم شبکه‌های عصبی رو به جلو، شبکه عصبی رو به جلو چندلایه است که شامل یک یا چند لایه پنهان است که نورون‌های این لایه‌ها، نورون‌های مخفی نامیده می‌شوند. با اضافه کردن چندلایه پنهان شبکه قابلیت تجزیه و تحلیل‌هایی با درجه پیچیدگی بالاتر را خواهد داشت . ساده‌ترین و مشهورترین نوع این مدل، شبکه‌های ELM هستند که آموزش آنها نیز بسیار ساده است. توابع فعال‌سازی در لایه خروجی، عمدتا براساس نیاز کاربر انتخاب می‌شوند. هر نورون در هر لایه از شبکه به واسطه وزن به هر کدام از نورون‌های لایه بعدی خود متصل شده است [۸] .

۳- روش پیشنهادی

همانطور که در بخش گذشته به آن اشاره شد، الگوریتم جستجوی کلاغ همگرایی کم دارد و در دام بهینه‌های محلی گرفتار می‌شود. الگوریتم جستجوی گرانشی در برخی از مسائل بهینه‌سازی دچار همگرایی زودرس شده و در بهینه محلی گیر می‌کند و پیشرفتی برای پیدا کردن جواب بهینه ندارد. شبکه‌های عصبی رو به جلو به دلیل مقداردهی اولیه تصادفی ممکن است به خوبی آموزش ندیده و این امر سبب بروز خطا در فرآیند آموزش گردد.

بدین منظور در این فصل قصد داریم، برای برطرف ساختن محدودیت‌های فوق از روش ترکیبی الگوریتم جستجوی کلاغ فازی و الگوریتم جستجوی گرانشی را برای مقداردهی اولیه وزن‌های ورودی شبکه عصبی رو به جلو پیشنهاد می‌دهیم.

مراحل روش پیشنهادی به شرح زیر است:

۱- بهبود پارامترهای الگوریتم جستجوی کلاغ با رویکرد فازی

۲- بهبود الگوریتم جستجوی گرانشی با استفاده از الگوریتم جستجوی کلاغ فازی

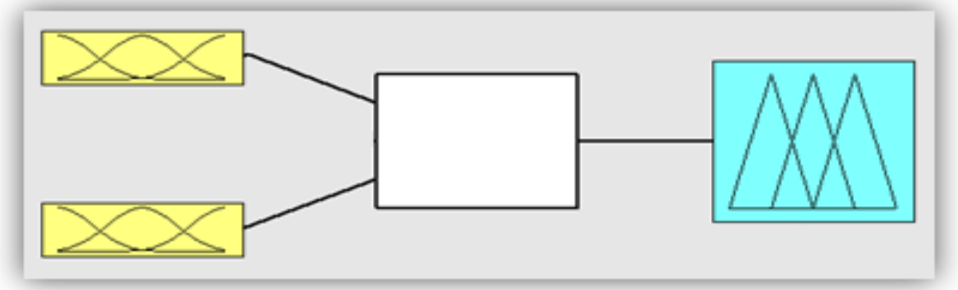
۳- بهبود فرآیند آموزش شبکه‌های عصبی رو به جلو با استفاده از الگوریتم ترکیبی جستجوی کلاغ فازی- جستجوی گرانشی

۳-۱ بهبود پارامترهای الگوریتم جستجوی کلاغ با رویکرد فازی

الگوریتم جستجوی کلاغ (CSA) یک الگوریتم طبیعی جدید پیشنهادی توسط عسکرزاده در سال ۲۰۱۶ است. مانند بیشتر الگوریتم‌های بهینه‌سازی، CSA همگرایی کم دارد و در بهینه‌های محلی گیر می‌کند.

ازین رو در روش پیشنهادی از قوانین‌فازی برای محاسبه احتمال فرآیند الگوریتم جستجوی کلاغ و درنهایت رسیدن به نقطه بهبود یافته استفاده خواهیم کرد.

شمای سیستم فازی طراحی شده برای روش پیشنهادی بصورت زیر خواهد بود:



شکل ۵ سیستم فازی در الگوریتم جستجوی کلاغ

در سیستم فازی پیشنهادشده، دو متغیر به‌عنوان ورودی سیستم و یک پارامتر به‌عنوان خروجی سیستم فازی در نظر گرفته‌شده است. متغیرهای ورودی سیستم فازی شامل شماره تکرار و میزان برازندگی کلاغ i و پارامتر خروجی این سیستم احتمال انتخاب عمل برای تعقیب کلاغ است (۱- حالتی که کلاغ می‌داند در حال تعقیب شدن است و ۲- حالتی که کلاغ نمی‌داند در حال تعقیب شدن است). مقدار پارامتر خروجی در هر تکرار با توجه به شرایط ورودی‌های سیستم فازی تعیین می‌شود و قوانین استنتاج سیستم فازی می‌تواند به عنوان مثال بصورت زیر باشد:

اگر برازندگی کلاغ بالا باشد و تکرارهای آخر باشد بهتر این است که احتمال p کم شود.

اگر برازندگی کلاغ کم باشد و تکرارهای آخر باشد بهتر این است که احتمال p زیاد شود.

و ...

جزئیات ورودی‌های سیستم فازی به شرح زیر است:

○ متغییر ورودی تکرار

در روش پیشنهادی، سیستم استنتاج فازی احتمال انتخاب عمل برای تعقیب کلاغ را محاسبه می‌نماید. محاسبه احتمال انتخاب عمل برای تعقیب کلاغ در هر تکرار الگوریتم انجام می‌شود که برای اندازه‌گیری متغیر ورودی تکرار از روند نرمال‌سازی تکرارها استفاده می‌کند؛ یعنی در شروع الگوریتم مقدار تکرارها کم (نزدیک به صفر) و در پایان الگوریتم مقدار تکرارها زیاد (یک) در نظر گرفته می‌شود و مقدار متغیر تکرار در بازه $[0,1]$ تعریف شده است که این مسئله به‌صورت رابطه‌ی (۱۰) بیان می‌شود:

$$Iteration = \frac{Current\ Iteration}{Maximum\ of\ Iteration} (10)$$

○ متغیر ورودی برازندگی

محاسبه برازندگی هر کلاغ با توجه به تابع معیار ریاضی موردنظر انجام می‌شود. برای متغیر برازندگی، نیاز به نرمال‌سازی است که رابطه‌ی (۱۱) برای نرمال‌سازی برازندگی هر کلاغ بکار برده می‌شود.

$$Fitness_{Norm_i} = \frac{Fitness_i - min(Fitness)}{\max(Fitness) - \min(Fitness)} (11)$$

که $Fitness_Norm_i$ مقدار نرمال شده برازندگی کلاغ i-ام است و $Fitness_i$ برازندگی کلاغ i-ام را نشان می‌دهد و $min(Fitness)$ مقدار برازندگی بدترین کلاغ موجود در جمعیت کلاغ‌ها و $max(Fitness)$ مقدار برازندگی بهترین کلاغ موجود در جمعیت است.

۳-۲بهبود الگوریتم جستجوی گرانشی با استفاده از الگوریتم جستجوی کلاغ فازی

همانطور که پیش تر گفته شد الگوریتم جستجوی گرانشی در برخی از مسائل بهینه‌سازی دچار همگرایی زودرس شده و در بهینه محلی گیر می‌کند و پیشرفتی برای پیدا کردن جواب بهینه ندارد که این مشکل جزو معایب این الگوریتم محسوب می‌شود. بدین منظور جمعیت اولیه این الگوریتم را با استفاده از الگوریتم جستجوی کلاغ فازی مقداردهی می‌نماییم تا مشکلات آن بر طرف گردد. در الگوریتم کلاغ ، احتمال [P] این است که کلاغ بر چه اساسی حرکت می‌کند. وقتی این عمل به صورت تصادفی انجام شود ، در دام بهینه محلی گیر می‌کند و به همگرای زود راس می‌رسدونتیجه به دست آمده یک نتیجه غیر واقعی است و دقت کاهش پیدا کرده و مدت زمان بالا میرود. و ما با استفاده از رویکرد فازی مشکلی که به صورت تصادفی(در دام بهینه محلی می‌افتد و دچار همگرایی زود راس میشود) ایجاد شده را برطرف می‌کنیم. برای بحث پارامتر وزن نیاز به الگوریتمی است که با استفاده از آن بتوانیم وزن را به دست آوریم . الگوریتم جستجوی گرانشی بر اساس وزن اجسام است. هر چه میزان وزن اجسام بیشتر ، جرم آن بیشتر است . اجسام با جرم کمتر کشش به سمت اجسام با جرم بیشتری دارند.. همین حالت بر روی الگوریتم کلاغ نیز وجود دارد .این جرم همان میزان برازندگی را مشخص میکند . هرچه جرم بیشتر باشد برازندگی هم بهتر است و به همین خاطر از الگوریتم جستجوی گرانشی استفاده می‌کنیم. و مقادیر به دست آمده از الگوریتم کلاغ فازی به الگوریتم جستجوی گرانشی داده می‌شودتا الگوریتم جستجوی گرانشی وزن‌های بهبود و بهینه شده را تحویل دهد.. تا افتادن در دام بهینه محلی و همگرایی زود راس را بر طرف کند. شبکه‌ها به دلیل انتخاب تصادفی اولیه وزن ورودی ممکن است شبکه به خوبی آموزش نبیند و خروجی آن جواب بهبود یافته ای نباشد. شبکه‌های رو به جلو دارای ساختاری لایه به لایه هستند و در این ساختار همه اتصالات از یک لایه به لایه دیگر است و هیچ راهی از لایه به خودش وجود ندارد در مباحث مربوط به شبکه‌های عصبی، قواعد یادگیری، رویه‌ای برای اصلاح وزن‌ها و بایاس‌ها تعریف می‌کنیم قاعده یادگیری در راستای آموزش شبکه برای انجام کار خاصی مورد استفاده قرار می‌گیرد. در فاز آموزش به دلیل مقداردهی تصادفی ممکن است خطا صورت دهدبه همین منظور استفاده از روش ترکیبی الگوریتم جستجوی کلاغ فازی و الگوریتم جستجوی گرانشی را برای این منظور پیشنهاد می‌دهیم. الگوریتم گرانشی در مقاله پایه از PSO و GSA برای بهبود شبکه عصبی رو به جلو به صورت مجزا به کاربرده است و نتایج خوبی به دست نیامده است . ما نیز به طبع از مقاله پایه ، در روش پیشنهادی از این دو الگوریتم به صورت ترکیبی برای شبکه عصبی رو به جلو استفاده نموده ایم و نتیجه بهتری را کسب کرده ایم .

۳-۳بهبود فرآیند آموزش شبکه‌های عصبی رو به جلو با استفاده از الگوریتم ترکیبی جستجوی کلاغ فازی- جستجوی گرانشی

همانطور که گفته شد انتخاب صحیح وزن بین لایه‌ها سبب کاهش خطا و افزایش دقت در شبکه‌های عصبی رو به جلو می‌شود. در این راستا در فاز آموزش شبکه‌های عصبی رو به جلو مقداردهی اولیه وزن بین لایه‌های ورودی را توسط

الگوریتم ترکیبی جستجوی کلاغ فازی-جستجوی گرانشی انجام خواهیم داد تا میانگین خطای مربعات شبکه را کاهش دهیم. فرآیند آموزش شبکه‌های عصبی رو به جلو بصورت زیر است:

۱- انتخاب پارامترهای آموزش و اینرسی

۲- مقداردهی وزن بین نرون‌های شبکه با اعداد کوچک

۳- به ازای هر یک از نمونه‌های آموزشی مراحل ذیل انجام شود:

۳-۱ پاسخ نرون‌های لایه خروجی به ازای بردار ورودی نمونه مورد نظر محاسبه شود. o_j^M

۳-۲ پارامتر دلتا برای کلیه نرون‌های لایه خروجی برآورد شود.

$$\delta_j^M = \left(d_j - o_j^M\right)\left(1 - o_j^M\right)o_j^M (12)$$

۳-۳ پارامتر دلتا برای کلیه نرون‌های لایه‌های میانی برآورد شود.

$$\delta_j^k = \frac{-\partial E}{\partial o_j^k}\left(1 - o_j^k\right)o_j^k = \left(1 - o_j^k\right)o_j^k \sum_l \delta_j^{k+1} w_{j,l}^{k,k+1} (13)$$

۳-۴ برای کلیه پارامترهای وزن شبکه، تصحیحات مورد نیاز محاسبه شود.

$$\Delta w_{i,j}^{k-1,k} = \alpha \delta_j^k o_i^{k-1} (14)$$

۳-۵ تصحیح برآورد شده با تصحیحات قبلی که به ازای سایر نمونه‌ها حاصل شده است، جمع شود.

$$\Delta w_{i,j}^{k-1,k} = \Delta w_{i,j}^{k-1,k} + \Delta^P w_{i,j}^{k-1,k} (15)$$

۴- تصحیحات نهایی دوره آموزش با اعمال پارامتر اینرسی برآورد شود.

$$\Delta w_{i,j}^{k-1,k}(t+1) = \Delta w_{i,j}^{k-1,k} + \beta \Delta w_{i,j}^{k-1,k}(t) (16)$$

۵- پارامترهای وزن شبکه براساس تصحیحات فوق، بهنگام شوند.

$$w_{i,j}^{k-1,k}(t+1) = w_{i,j}^{k-1,k}(t) + \Delta w_{i,j}^{k-1,k}(t+1) (17)$$

۶- مراحل ۳، ۴ و ۵ تا مرحله همگرایی آموزش تکرار شوند.

۴- نتایج شبیه سازی

آنچه در این بخش بررسی خواهد شد شامل جزییات مربوط به پیاده‌سازی روش پیشنهادی می‌باشد. دادگان استفاده‌شده در این بخش و نحوه‌ی جمع‌آوری آن‌ها ذکر خواهد شد و برای شفاف‌شدن موضوع داده‌های استفاده‌شده، در ادامه درج می‌شود.

۴-۱ مجموعه داده

برای بررسی عملکرد روش پیشنهادی، عملکرد یک گیت منطقی سه ورودی نوع *XOR* شبیه سازی می‌شود. چگونگی عملکرد یک گیت *XOR* در جدول ۱ آمده است:

جدول ۱ مسئله *XOR*

خروجی	ورودی
0	0 0 0
1	0 0 1
1	0 1 0
0	0 1 1
1	1 0 0
0	1 0 1
0	1 1 0
1	1 1 1

۴-۲ معیار ارزیابی

در ریاضیات و آمار، خطای میانگین مربعات (*MSE*) روشی برای برآورد میزان خطاست که در واقع تفاوت بین مقادیر تخمینی و آنچه تخمین زده شده، است. *MSE* به دو دلیل تقریباً همه جا مثبت است (صفر نیست) یک اینکه تصادفی است و دوم به این دلیل که تخمین‌گر اطلاعاتی که قابلیت تولید تخمین دقیق‌تری دارد را حساب نمی‌کند. پس این شاخص که مقداری همواره نامنفی دارد، هرچقدر مقدار آن به صفر نزدیکتر باشد، نشان دهنده میزان کمتر خطاست.

MSE شامل واریانس تخمین‌گر و بایاس (سوگیری) است. برای یک برآوردگر غیر بایاس، *MSE* همان واریانس برآوردگر است. مثل واریانس، *MSE* همان واحدهای اندازه‌گیری را به عنوان مربع مقادیر تخمین زده شده، دارد.

در نتیجه به منظور ارزیابی روش پیشنهادی از معیار میانگین خطای مربعات بهره خواهیم برد که بصورت زیر است:

$$MSE = \sqrt{\frac{1}{N}\left[\sum_{i=1}^{N}\left(f^{E}(x_i) - f(x_i)\right)\right]^2} \quad (18)$$

۴-۳ ارزیابی نتایج

در این بخش به ارزیابی روش پیشنهادی به منظور بدست آوردن کمترین میزان میانگین خطای مربعات، برای مجموعه داده‌ی مورد نظر، خواهیم پرداخت.

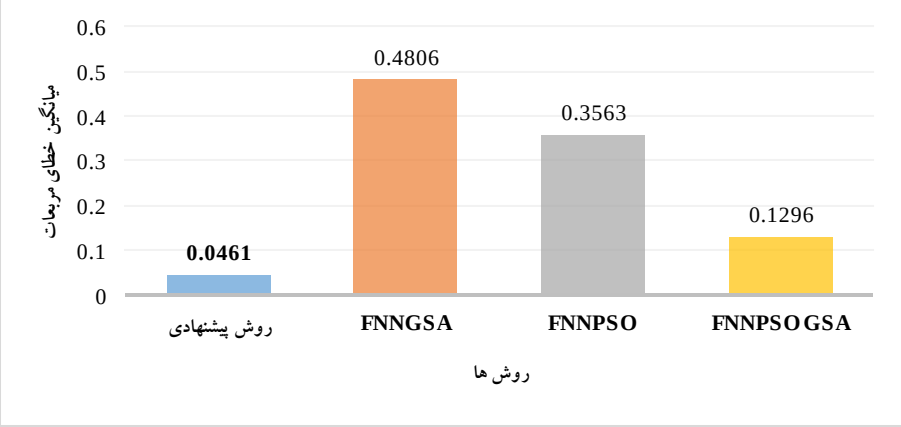
روش پیشنهادی توسط نرم افزار متلب توسط معیار ارزیابی مورد ارزیابی قرار گرفته است.

- **حالت(۱): تعداد گره‌های لایه پنهان برابر یا پنج باشد**

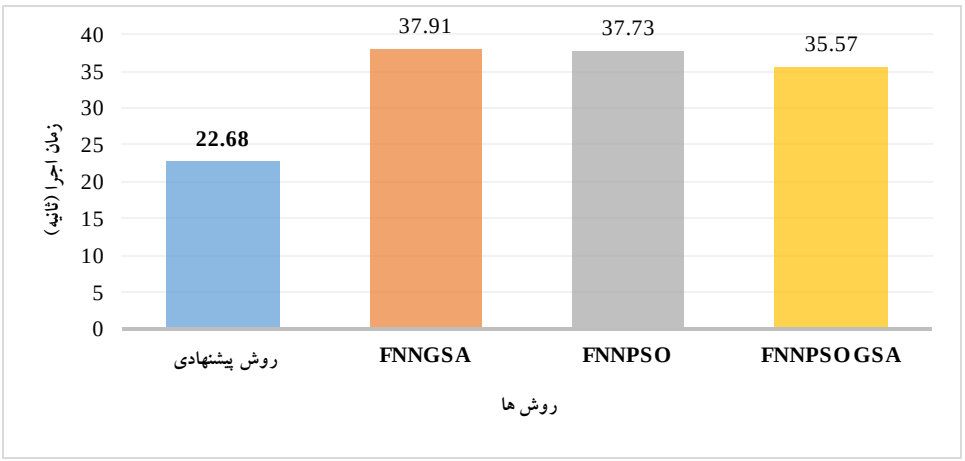
مقایسه نتایج بدست‌آمده از روش پیشنهادی بر روی حالت (۱) توسط معیارهای ارزیابی با سایر روش‌ها در جدول (۲) به شرح زیر است:

جدول ۲ مقایسه نتایج بدست آمده از روش پیشنهادی با سایر روش‌ها در حالت (۱)

	روش‌ها	میانگین خطای مربعات	زمان اجرا (ثانیه)
حالت (۱)	روش پیشنهادی	*0.0461*	*22.68*
	FNNGSA	*0.4806*	*37.91*
	FNNPSO	*0.3563*	*37.73*
	FNNPSOGSA	*0.1296*	*35.57*



نمودار ۱ مقایسه میانگین خطای مربعات روش پیشنهادی با سایر روش‌ها در حالت (۱)

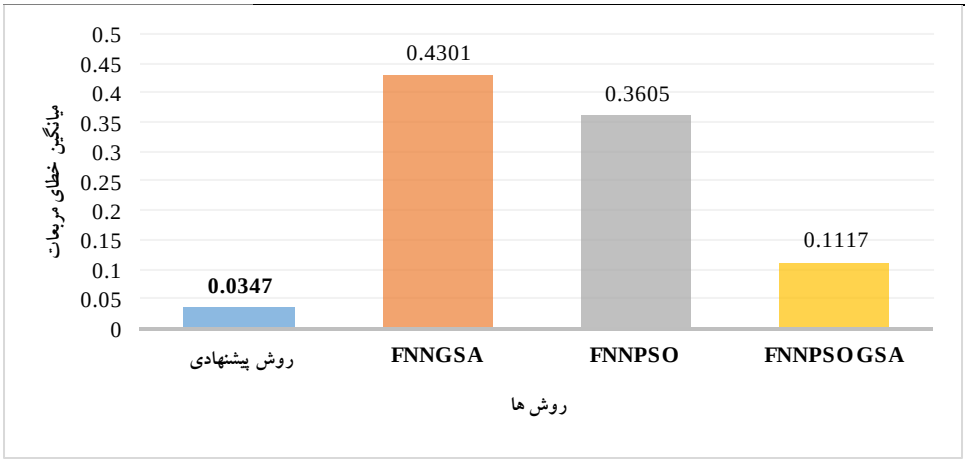


نمودار ۲ مقایسه زمان اجرای روش پیشنهادی با سایر روش‌ها در حالت (۱)

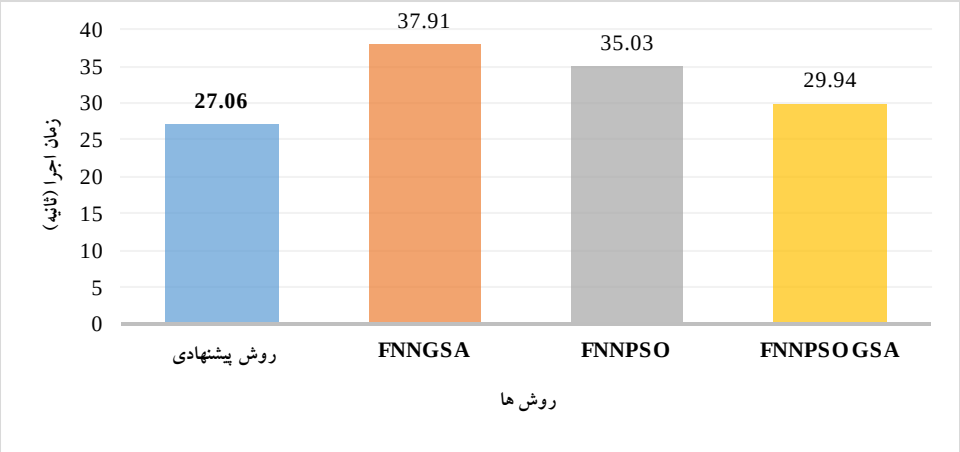
- **حالت(۲): تعداد گره‌های لایه پنهان برابر یا ده باشد** مقایسه نتایج بدست‌آمده از روش پیشنهادی بر روی حالت (۲) توسط معیارهای ارزیابی باسایر روش‌ها در جدول (۳) به شرح زیراست:

جدول ۳ مقایسه نتایج بدست آمده از روش پیشنهادی با سایر روش‌ها در حالت (۲)

	روش‌ها	میانگین خطای مربعات	زمان اجرا (ثانیه)
حالت (۲)	روش پیشنهادی	*0.0347*	*27.06*
	FNNGSA	*0.4301*	*37.91*
	FNNPSO	*0.3605*	*35.03*
	FNNPSOGSA	*0.1170*	*29.94*



نمودار ۳ مقایسه میانگین خطای مربعات روش پیشنهادی با سایر روش‌ها در حالت (۲)



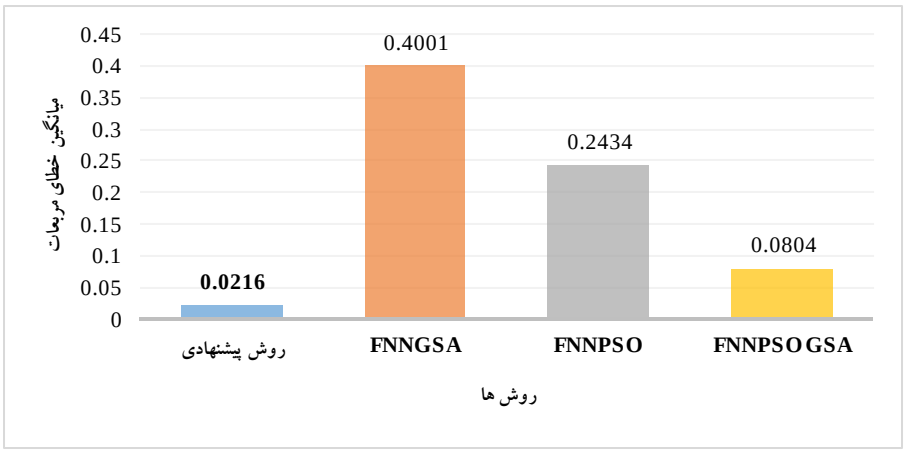
نمودار ۴ مقایسه زمان اجرای روش پیشنهادی با سایر روش‌ها در حالت (۲)

- **حالت(۳): تعداد گره‌های لایه پنهان برابر یا پانزده باشد**

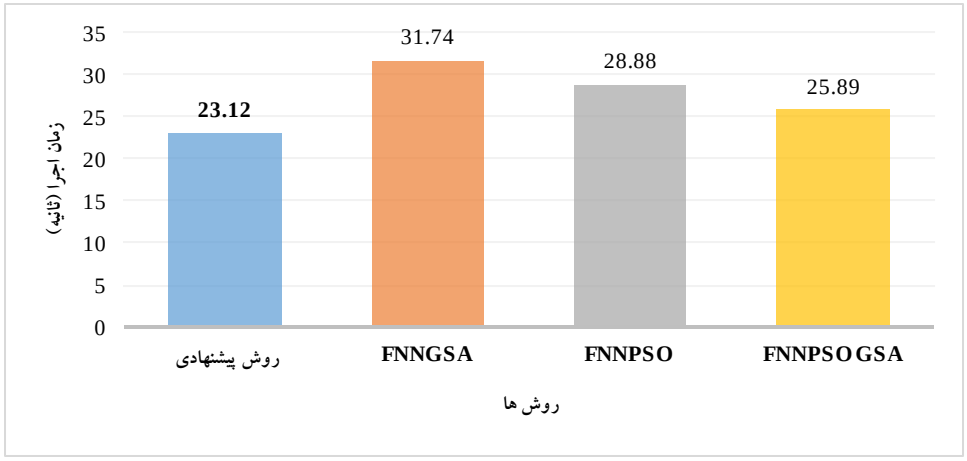
مقایسه نتایج بدست‌آمده از روش پیشنهادی بر روی حالت (۳) توسط معیارهای ارزیابی با سایر روش‌ها در جدول (۴) به شرح زیر است:

جدول ۴ مقایسه نتایج بدست آمده از روش پیشنهادی با سایر روش‌ها در حالت (۳)

	روش‌ها	میانگین خطای مربعات	زمان اجرا (ثانیه)
حالت (۳)	روش پیشنهادی	*0.0216*	*23.12*
	FNNGSA	*0.4001*	*31.74*
	FNNPSO	*0.2434*	*28.88*
	FNNPSOGSA	*0.0804*	*25.89*



نمودار ۵ مقایسه میانگین خطای مربعات روش پیشنهادی با سایر روش‌ها در حالت (۳)



نمودار ۶ مقایسه زمان اجرای روش پیشنهادی با سایر روش‌ها در حالت (۳)

به طور کلی، در تمام نتایج تولیدشده، می‌توان مشاهده کرد که *FNNGSA* به دلیل روند جستجوی کُند *GSA*، که بر توانایی استخراج *FNNGSA* تأثیر دارد، عملکرد خوبی ندارد. با این‌حال، *GSA* در بین همه الگوریتم‌های تکاملی توانایی اکتشاف قوی دارد. *FNN* نه تنها به توانایی اکتشافی قوی بلکه به توانایی استخراج دقیق نیاز دارد. با مراجعه به نتایج حاصل از دقت طبقه‌بندی بدست‌آمده توسط الگوریتم‌های مبتنی بر *FNN*، نشان داده می‌شود که *FNNPSO* به دلیل توانایی استخراج دقیق‌تر از *PSO* عملکرد بهتری نسبت به *FNNGSA* دارد، اما همچنان از مشکل افتادن در دام حداقل‌های محلی رنج می‌برد. این ضعف بدین معنی است که *FNNPSO* از عملکرد ناپایدار برخوردار است. نتایج به دست آمده توسط روش پیشنهادی ثابت می‌کند که هم از استخراج قوی و هم توانایی اکتشافی خوبی برخوردار است. به عبارت دیگر، از قدرت الگوریتم جستجوی کلاغ فازی والگوریتم جستجوی گرانشی با موفقیت استفاده شده است و عملکرد خوبی در آموزش *FNN* دارد. این بدان معنی است که روش پیشنهادی قادر به حل مشکل افتادن در دام حداقل‌های محلی است و سرعت همگرایی سریع را ارائه می‌دهد.

۵- نتیجه‌گیری و کارهای آینده

در این مقاله، الگوریتم آموزش جدید به نام *FCSAGSA* با استفاده از *GSA* و *FCSA* معرفی و مورد بررسی قرار گرفته است.کارایی الگوریتم‌های ارائه شده برای آموزش شبکه عصبی بر روی مشکل معیار *3-bit XOR* ارزیابی شد. نتایج با سایر روش‌ها مقایسه شد. برای حل مسئله *3-bit XOR*، روش پیشنهادی از نظر میزان همگرایی و جلوگیری از حداقل محلی، عملکرد بهتری را نشان می‌دهد. مشاهده شد که *FNNPSO* خطای کمتری را نسبت به *FNNGSA* نشان می‌دهد. بنابراین، می‌توان نتیجه گرفت که روش پیشنهادی، مشکل افتادن در دام بهینه حداقل‌های محلی و سرعت همگرایی را بهبود می‌بخشد. نتایج *FNNGSA* همچنین ثابت می‌کند که *GSA* به دلیل سرعت جستجوی آهسته، برای آموزش *FNN* مناسب نیست. به طور خلاصه، نتایج ثابت می‌کند که روش پیشنهادی مشکل افتادن در دام حداقل محلی را بهبود داده و سرعت همگرایی را در مقایسه با الگوریتم‌های یادگیری موجود برای *FNN*ها افزایش می‌دهد. در این پژوهش تلاش شد تا گام کوچکی در جهت بهبود نتایج حاصل در تحقیقات داده‌کاوی در حوزه استخراج آموزش شبکه‌های عصبی، برداشته شود. در این حوزه مقداردهی بهینه اولیه وزن‌های ورودی و آموزش صحیح شبکه عصبی می‌تواند تاثیر حیاتی در روند طبقه‌بندی، داشته باشد. با وجود نتایج دلگرم‌کننده، برخی از محدودیت‌ها باید در کارهای آینده رسیدگی شود. اول از همه، استفاده از یک ضریب تنظیم برای جلوگیری از بروز مشکل بیش‌برازش و دوم اینکه برای مسائل داده‌های با ابعاد زیاد برای تجزیه و تحلیل روش ارائه شده در بحث طبقه‌بندی استفاده گردد.

مراجع

[1] Gupta, Deepak, Shirsh Sundaram, Ashish Khanna, Aboul Ella Hassanien, and Victor Hugo C. de Albuquerque. "Improved diagnosis of Parkinson's disease using optimized crow search algorithm." Computers & Electrical Engineering 68 (2018): 412-424.

[2]Faris, Hossam, Ibrahim Aljarah, and Seyedali Mirjalili. Improved monarch butterfly optimization for unconstrained global search and neural network training. *Applied Intelligence* 48, no. 2 (2018): 445-464.

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

[4] Zolghadr-Asli, Babak, Omid Bozorg-Haddad, and Xuefeng Chu. "Crow Search Algorithm (CSA)." In Advanced Optimization by Nature-Inspired Algorithms, pp. 143-149. Springer, Singapore, 2018.

[5] Naik, Bighnaraj, and Janmenjoy Nayak. Crow Search Optimization-Based Hybrid Metaheuristic for Classification: A Novel Approach. In Progress in Computing, Analytics and Networking, pp. 775-783. Springer, Singapore, 2018.

[6] Do, Quang Hung. "A hybrid gravitational search algorithm and back-propagation for training feedforward neural networks." In *Knowledge and Systems Engineering*, pp. 381-392. Springer, Cham, 2015.

[7] Aljarah, Ibrahim, Hossam Faris, and Seyedali Mirjalili. Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Computing* 22, no. 1 (2018): 1-15.

[8] Wu, Haizhou, Yongquan Zhou, Qifang Luo, and Mohamed Abdel Basset. "Training feedforward neural networks using symbiotic organisms search algorithm." *Computational intelligence and neuroscience* 2016 (2016).

[9] Allaoui, Mohcin, Belaïd Ahiod, and Mohamed El Yafrani. A hybrid crow search algorithm for solving the DNA fragment assembly problem. *Expert Systems with Applications* 102 (2018): 44-56.

[10] Liu, Dong, Chunlei Liu, Qiang Fu, Tianxiao Li, Khan M. Imran, Song Cui, and Faiz M. Abrar. "ELM evaluation model of regional groundwater quality based on the crow search algorithm." Ecological Indicators 81 (2017): 302-314.

[11] Sayed, Gehad Ismail, Ashraf Darwish, and Aboul Ella Hassanien. Chaotic crow search algorithm for engineering and constrained problems. In Computer Engineering and Systems (ICCES), 2017 12th International Conference on, pp. 676-681. IEEE, 2017.

[12] Sayed, Gehad Ismail, Aboul Ella Hassanien, and Ahmad Taher Azar. "Feature selection via a novel chaotic crow search algorithm." Neural Computing and Applications (2017): 1-18.

[13] Huang, Ko-Wei, Ze-Xue Wu. CPO: A Crow Particle Optimization Algorithm. International Journal of Computational Intelligence Systems 12, no. 1 (2018): 426-435.