

## بهینه‌سازی پایگاه قوانین سیستم فازی نوع TSK با استفاده از الگوریتم‌های ترکیبی فراابتکاری

حجت اله حمیدی

استادیار، گروه مهندسی فناوری اطلاعات، دانشکده مهندسی صنایع، دانشگاه صنعتی خواجه نصیرالدین طوسی

H\_hamidi@kntu.ac.ir

**چکیده:** یکی از چالش‌های اصلی در استفاده از سیستم‌های فازی، چگونگی طراحی پایگاه قواعد فازی با پارامترهای بهینه‌سازی شده است؛ به نحوی که منجر به عملکرد رضایت‌بخش سیستم شود. در این مقاله از روش آموزش ترکیبی تکامل تفاضلی مبتنی بر تضاد (ODE) و بهینه‌سازی انبوه ذرات (PSO) به منظور بهینه‌سازی پارامترهای توابع عضویت گوسی پایگاه قواعد در سیستم فازی نوع تاکاگی - سوگنو - کانگ (TSK) استفاده شده است. همچنین از الگوریتم ترکیبی پیشنهادی، برای آموزش سیستم فازی TSK مرتبه صفر به منظور کنترل دو پلنت غیرخطی استفاده شده است و نتایج به دست آمده بیانگر این است که برای کنترل پلنت‌های غیرخطی مدل، دقت شناسایی بهتری را نسبت به سایر رویکردهای آموزشی از خود نشان می‌دهد. همچنین در این مقاله از ترکیب الگوریتم‌های ODE و PSO استفاده شده است و آن را در دو مسئله طراحی سیستم فازی دقت گرا به کار می‌گیرد. در این دو مدل، همه پارامترهای آزاد سیستم فازی TSK مرتبه یک، از طریق HODEPSO بهینه می‌شوند. مدل‌های استفاده شده در این آزمایش‌ها، سری آشوبناک مکی گلس و یک مسئله اقتصادی واقعی هستند که مقادیر آینده آن‌ها پیش‌بینی می‌شود. نتایج به دست آمده بیانگر آن است که HODEPSO حداقل خطای متوسط تست و آموزش را در مقایسه با دیگر روش‌های آموزش دارد.

**واژه‌های کلیدی:** الگوریتم‌های فراابتکاری تکامل تفاضلی مبتنی بر تضاد (ODE)، بهینه‌سازی انبوه ذرات (PSO)، سیستم‌های فازی، هوش جمعی (SI).

### ۱- مقدمه

سراسری نزدیک باشند؛ به طوری که نظر تصمیم‌گیرنده را تا سطحی پذیرفتنی برآورده سازند. توجه به این نکته از این جهت با اهمیت است که در بیشتر مدل‌سازی‌ها، نمی‌توان تمامی پارامترهای مؤثر بر شرایط مسئله را وارد مدل کرد؛ بنابراین تضمینی وجود ندارد که بهترین حل به دست آمده مدل، همان حل مطلوب برای شرایط واقعی باشد [۱]. حال این سؤال پیش می‌آید که یک حل دقیق از یک مدل تقریبی بهتر است یا یک حل تقریبی از یک مدل دقیق. در پاسخ به این پرسش، رویکردهای فراابتکاری، این امکان را فراهم می‌سازند که یک حل تقریباً خوب از یک مدل کاملاً دقیق به دست آورده شود؛ زیرا افزایش پیچیدگی مسئله تأثیر چندانی در عملکرد روش‌های فراابتکاری نخواهند داشت.

در سال‌های اخیر، استفاده از الگوریتم‌های فراابتکاری گسترش چشمگیری در حل مسائل پیچیده محاسباتی داشته است و امروزه نیز کاربردهای بسیار زیادی در شاخه‌های مختلف علوم دارد. این رویکردها برخلاف روش‌های دقیق بهینه‌سازی، به دنبال نقاطی هستند که تا حد ممکن به بهینه

<sup>۱</sup> تاریخ ارسال مقاله : ۱۳۹۴/۰۶/۰۱

تاریخ پذیرش مقاله : ۹۵/۰۹/۰۳

نام نویسنده مسئول : حجت اله حمیدی

نشانی نویسنده مسئول : ایران، تهران، دانشگاه صنعتی خواجه نصیرالدین طوسی

به عبارت دیگر، روش‌های فراابتکاری روش‌هایی هستند که حل‌های نزدیک به بهینه را با هزینه محاسباتی پذیرفتنی جست‌وجو می‌کنند؛ ولی تضمینی برای رسیدن به حل بهینه ارائه نمی‌دهند. به روش‌های فراابتکاری، در اصطلاح، روش‌های غیردقیق نیز گفته می‌شود؛ زیرا مکانیزم‌های تصادفی در ایجاد ساختار آن‌ها نقش مهمی دارد. به‌طور کلی مبنای این رویکردها براساس نظم یا قواعد موجود در ارگانیسم‌های طبیعی است [۲]. تاکنون برای طراحی اتوماتیک سیستم‌های فازی، الگوریتم‌های فراابتکاری زیادی معرفی شده‌اند [۳-۷]. یکی از روش‌های مهم بهینه‌سازی، استفاده از الگوریتم‌های تکاملی است [۸]. این نوع الگوریتم‌ها، ابتکاری و احتمالی هستند و دربردارنده جمعیتی از افراد با رفتاری خاص مشابه پدیده‌های بیولوژیکی هستند. مشهورترین سیستم‌های فازی تکاملی، سیستم‌های فازی ژنتیکی هستند که در آن، طراحی سیستم فازی با استفاده از الگوریتم ژنتیک<sup>۱</sup> انجام شده است [۹-۱۶]. به تازگی الگوریتم تکامل تفاضلی<sup>۲</sup> به‌عنوان یک تکنیک نوین محاسبات تکاملی [۱۷]، [۱۸] برای حل مسائل بهینه‌سازی استفاده می‌شود. تکامل تفاضلی به بسیاری از دیگر روش‌های تکاملی همچون الگوریتم ژنتیک [۱۹]، [۲۰] و بهینه‌سازی انبوه ذرات<sup>۳</sup> ترجیح داده می‌شود و این به دلیل ویژگی‌های درخور توجه آن همچون مفهوم ساده، اجرای آسان و همگرایی سریع است [۱۷]، [۱۸].

روش دیگر طراحی سیستم‌های فازی، استفاده از مدل هوش ازدحامی<sup>۴</sup> است [۲۱]. یک مدل مشهور هوش ازدحامی، الگوریتم بهینه‌سازی انبوه ذرات است که برای طراحی سیستم‌های فازی به کار رفته است [۲۲-۲۶]. مدل مشهور دیگر هوش ازدحامی، الگوریتم بهینه‌سازی کلونی مورچه<sup>۵</sup> است که برای طراحی سیستم‌های فازی به کار رفته است [۲۷-۳۱]. روش دیگر طراحی سیستم‌های فازی، استفاده از الگوریتم‌های ترکیبی است. هر الگوریتم، مزایا و معایبی دارد؛ به همین دلیل برای داشتن مزایای چند الگوریتم، این الگوریتم‌ها با هم ترکیب می‌شوند [۳۲-۳۴]. در این مقاله، الگوریتم‌های تکامل تفاضلی مبتنی بر تضاد و بهینه‌سازی انبوه ذرات به‌صورت مؤثری با هم ترکیب شده است. ترکیب<sup>۱</sup>، به‌معنای به‌کارگیری هم‌زمان دو یا چند روش

برای استفاده از مزایا و کم‌رنگ‌کردن معایب آن‌ها است [۲]. در واقع، با ترکیب این دو الگوریتم، تنوع جمعیتی ایجاد می‌شود و از افتادن الگوریتم در دام بهینه محلی جلوگیری می‌شود. همچنین الگوریتم، سریع‌تر و با تقریب بهتری به جواب بهینه، همگرا می‌شود. برخی از روش‌های ترکیب الگوریتم تکامل تفاضلی و بهینه‌سازی انبوه ذرات در [۳۵] و [۳۶] پیشنهاد شده‌اند و برای حل مسائل بهینه‌سازی در زمینه‌های مختلفی به کار رفته‌اند. سیستم‌های فازی برای حل مسائل پیچیده که دچار عدم قطعیت هستند و توصیف ریاضی آن‌ها سخت است، بسیار مناسب هستند. چالش اصلی در استفاده از این نوع سیستم‌ها، چگونگی طراحی یک پایگاه قواعد با پارامترهای بهینه‌سازی شده است؛ به‌نحوی که منجر به عملکرد رضایت‌بخش سیستم شود. برای رویارویی با این مسئله به روش‌های هوشمند برای طراحی سیستم فازی ترجیح داده می‌شود. در این مقاله، پس از انجام مطالعات لازم در این زمینه، تلاش شده است با بهره‌گیری از الگوریتم‌های فراابتکاری، پارامترهای موجود در پایگاه قواعد فازی به روشی هوشمند، بهینه شوند؛ به‌نحوی که حتی با افزایش تعداد متغیرهای ورودی و افزایش تعداد قواعد فازی به‌صورت نمایی، پایگاه قواعد فازی طراحی شده همچنان عملکرد مناسب و رضایت‌بخش داشته باشد. به همین منظور در این مقاله، الگوریتم‌های تکامل تفاضلی مبتنی بر تضاد و بهینه‌سازی انبوه ذرات به‌صورت مؤثری با هم ترکیب شده‌اند. ترکیب، به‌معنای به‌کارگیری هم‌زمان دو یا چند روش برای استفاده از مزایا و کم‌رنگ‌کردن معایب آن‌ها است [۲]. در واقع با ترکیب این دو الگوریتم، تنوع جمعیتی ایجاد می‌شود و از افتادن الگوریتم در دام بهینه محلی جلوگیری می‌شود. همچنین الگوریتم، سریع‌تر و با تقریب بهتری به جواب بهینه، همگرا می‌شود.

این مقاله به شرح زیر سازماندهی شده است: در بخش دوم، کارهای انجام‌شده بررسی شده است. در بخش سوم، الگوریتم‌های تکامل تفاضلی مبتنی بر تضاد و بهینه‌سازی انبوه ذرات به‌کاررفته برای آموزش سیستم فازی ارائه شده است. در بخش چهارم، برای ارزیابی روش پیشنهادی، مثال‌های شبیه‌سازی شده با استفاده از رویکرد پیشنهادی بررسی شده است و نتایج تمامی تکنیک‌های موجود برای

مقدار Q مبتنی بر یادگیری فازی Q است. بهترین ترکیب مقادیر نتیجه‌دهنده یک سیستم رابط فازی طبق میزان فرمون و مقادیر Q جست‌وجو می‌شود. در پایان ACO-FQ به سه مسئله معروف، اعمال شده و نشان داده شده است که در مقایسه با دیگر روش‌های طراحی سیستم فازی تقویتی، عملکرد مناسب‌تری دارد.

در [۳۷]، یک شبکه فازی عصبی تکاملی ارائه شده است که با استفاده از پیوند عملیاتی شبکه فازی عصبی (FLNFN) و یک الگوریتم جدید یادگیری تکاملی طراحی شده است. این الگوریتم جدید یادگیری تکاملی مبتنی بر یک مشارکت بهینه‌سازی انبوه ذرات و الگوریتم فرهنگی است؛ بنابراین آن را بهینه‌سازی انبوه ذرات با مشارکت فرهنگی (CCPSO) می‌نامند. روش CCPSO ارائه‌شده از رفتار مشارکتی چندین جمع بهره می‌برد و می‌تواند با استفاده از فضای فرهنگی اجتماع، ظرفیت جست‌وجوی فرامحلی (سراسری) را افزایش دهد. رفتار مشارکتی دربردارنده جمعی از چندین جمعیت است که با تبادل اطلاعات تعامل می‌کند تا مسئله‌ای را حل کند. فضای فرهنگی اجتماع، گنجینه اطلاعاتی است که در آن، افراد قادرند تجارب خود را به گونه‌ای ذخیره کنند که دیگر افراد بتوانند به‌طور غیرمستقیم از آن‌ها یاد بگیرند. مدل FLNFN ارائه‌شده از پیوند عملیاتی شبکه‌های فازی عصبی به‌عنوان بخش نتیجه‌دهنده قوانین فازی استفاده می‌کند. این مطالعه از چندجمله‌ای‌های متعامد و توابع مستقل خطی در توسعه پیوند عملیاتی شبکه‌های فازی عصبی بهره می‌برد. مدل FLNFN می‌تواند بخش نتیجه‌دهنده ترکیب غیرخطی متغیرهای ورودی را تولید کند. درنهایت، FLNFN ارائه‌شده به همراه CCPSO (FLNFN-CCPSO) در چندین کاربرد پیش‌بینی اتخاذ شده‌اند و نتایج آزمایشگاهی نشان داده است روش CCPSO ارائه‌شده در پیش‌بینی مسائل مجموعه‌های زمانی، عملکرد خوبی را به نمایش می‌گذارد.

در [۳۸]، برای کاهش لرزش ساختمان‌ها با استفاده از میراکننده‌های مگنتورئولوژیکال (MR)، یک الگوریتم کنترل بهینه منطبق فازی ارائه شده است. میراکننده‌های MR

رسیدن به نتیجه‌گیری کلی به بحث گذاشته شده است و در بخش پنجم و ششم به ترتیب، بحث و نتیجه‌گیری شده و پیشنهادهایی جهت تحقیقات آینده مطرح شده است.

## ۲- مروری بر ادبیات

در [۲۸]، کاربردهای بهینه‌سازی تجمع مورچگان (ACO) در طراحی یک کنترلر فازی (FC)، موسوم به ACO-FC، به‌منظور بهبود کارایی طراحی و عملکرد کنترلی، و نیز پیاده‌سازی سخت‌افزار ACO، ارائه شده است. بخش مقدم یک کنترلر فازی، یعنی بخش "IF" قوانین IF-THEN فازی تشکیل‌دهنده آن، در قسمت نوع شبکه قرار می‌گیرد و پس از آن، همه مقادیر نتیجه‌دهنده قوانین نامزد لیست می‌شوند. یک سفر مورچه به‌عنوان ترکیبی از مقادیر نتیجه‌دهنده انتخاب‌شده از هر قانون در نظر گرفته می‌شود. یک ماتریس فرمون از بین همه مقادیر نتیجه‌دهنده نامزد ایجاد می‌شود. جست‌وجو برای بهترین مقدار از بین همه ترکیبات مقادیر نتیجه‌دهنده قانون، عمدتاً مبتنی بر ماتریس فرمون است. از طریق مثال‌های شبیه‌سازی، نشان داده شده است که عملکرد ACO-FC ارائه‌شده، نسبت به دیگر روش‌های طراحی فراابتکاری بهتر است. ACO به‌کاررفته در ACO-FC مبتنی بر سیستم مشخصی از جمعیت مورچگان بوده است و سخت‌افزار روی یک تراشه آرایه گیت منطقی درخور برنامه‌ریزی (FPGA) عملی می‌شود. کاربرد تراشه ACO در کنترل فازی به‌وسیله یک مسئله کنترل دمای آب حمام شبیه‌سازی شده است و نتایج به‌دست‌آمده نشان‌دهنده کارایی تراشه طراحی شده است.

در [۳۱]، طراحی کنترلرهای فازی توسط بهینه‌سازی جمعیت مورچگان (ACO) در ترکیب با یادگیری فازی Q، موسوم به ACO-FQ، به همراه مقاوم‌سازی‌های مربوطه ارائه شده است. در ACO-FQ، سفر یک مورچه به‌صورت ترکیبی از عملیات نتیجه‌دهنده انتخاب‌شده از هر قانون در نظر گرفته می‌شود. جست‌وجوی بهترین مورچه از بین همه ترکیبات تا حدودی مبتنی بر تعقیب فرمون است. در این پژوهش برای هر نامزد موجود در بخش نتیجه‌دهنده قانون، یک مقدار Q متناظر اختصاص داده شده است. به‌روزرسانی

صحیح این روش ارزیابی، در داده‌های با ابعاد بالای مربوط به فرایندهای صنعتی و داده‌های اینترنت اعمال شده است و از منظر دقت و پیچیدگی سیستم‌های فازی با روش‌های معروف آموزش گروهی مقایسه شده است.

کسب دانش یک مسئله با سابقه طولانی در سیستم‌های مبتنی بر قانون فازی است. با وجود چندین روش در این زمینه، همچنان نیاز به تلاش زیادی جهت افزایش کارایی فرایند یادگیری است. در [۴۰]، یک روش جدید برای تکامل تدریجی قانون فازی ارائه شده است که شکل‌دهنده یک دانش سیستم خبره است. این راهبرد مبتنی بر استفاده از بهینه‌سازی انبوه ذرات (PSO) برای دستیابی به مقدمات، نتیجه‌دهنده‌ها و ارتباطات بین قوانین است. در این مقاله، برای آزمون امکان‌پذیری روش پیشنهادی، مسئله آونگ معکوس پیشنهاد شده است. در آخر نیز نشان داده شده است که نتایج شبیه‌سازی مؤید این حقیقت است که راهبرد ارائه‌شده از نظر نتایج نهایی و تلاش محاسباتی نسبت به روش‌های یادگیری کلاسیک عملکرد بهتری دارد. در [۴۱]، یک الگوریتم بهینه‌سازی انبوه ذرات چندگونه‌ای مبتنی بر دسته‌بندی سلسله مراتبی (HCMSPSO) برای بهینه‌سازی سیستم فازی ارائه شده است. هدف از این مقاله، یادگیری قوانین فازی نوع تاکاگی - سوگنو - کانگ (TSK) با دقت زیاد بوده است. در سیستم فازی (FS) طراحی شده با HCMSPSO، هر قانون مجموعه‌های فازی مربوط به خود را تعریف می‌کند؛ به این معنا که تعداد مجموعه‌های فازی برای هر متغیر ورودی با تعداد قوانین فازی برابر است. برای تأیید عملکرد HCMSPSO چندین شبیه‌سازی انجام شده است و در مقایسه با دیگر الگوریتم‌های یادگیری عصبی، ژنتیک و PSO، نشان داده شده است که عملکرد HCMSPSO نسبت به آن‌ها بهتر است. در [۴۲]، یک روش کنترل فازی در ترکیب با الگوریتم بهینه‌سازی انبوه ذرات (PSO) برای کنترل تضعیف شار موتور سنکرون با تحریک ترکیبی (HESM) ارائه شده است. برای تعیین جریان بهینه تحریک، جریان محورهای  $d$  و  $q$  در ناحیه سرعت پایین یک مدل کاهش از کل تلفات مسی ایجاد می‌شود. بزرگ‌ترین بهبود مشخصات دینامیکی و کاهش آثار منفی تغییرات پارامتر روی عملکرد

تجهیزات نیمه‌فعال هستند و با کمک منبع تغذیه خارجی پایش می‌شوند. در این مقاله، پایش ولتاژ میراکننده MR از طریق استفاده از سیستم فازی تکاملی انجام می‌گیرد و سیستم فازی با کمک الگوریتم‌های تکاملی (EA) بهینه می‌شود. الگوریتم میکروژنتیک ( $\mu$ -GA) و الگوریتم بهینه‌سازی انبوه ذرات (PSO) به منظور بهینه‌سازی پارامترهای FCL به کار می‌روند. در نهایت، عملکرد کنترل فازی بهینه‌شده به صورت غیرخطی بر روی یک مدل ساختمانی سه‌طبقه و تحت تحرکات ارتعاشی اعمال می‌شود. مزیت اصلی استفاده از FCL برای کنترل کردن ولتاژ میراکننده MR این است که یک تغییر تدریجی و صاف در ولتاژ ایجاد می‌کند؛ در نتیجه، روش ارائه‌شده، کنترل بهتری را برای ساختارهای تحت تنش فراهم می‌کند. در [۳۹]، یک روش جدید یادگیری مبتنی بر داده [سیستم‌های استنتاج فازی کم تعداد (SparseFIS)] برای سیستم‌های فازی تاکاگی - سوگنو (T-S) بررسی شده است که برای دربرگیری وزن‌های قوانین توسعه یافته است. روش یادگیری دربردارنده سه مرحله است: مرحله نخست، یک فرایند دسته‌بندی را در فضای مشخصه ورودی/خروجی با قیمت‌بخشی بردار تکرار انجام داده است و دسته‌های به‌دست‌آمده را روی محورهای یک بُعدی تصویر می‌کند تا مجموعه‌های فازی (مراکز و عرض‌ها) بخش‌های مقدمات قوانین را شکل دهد. مرحله دوم، از طریق اعمال یک رویه بهینه‌سازی شدیدترین نزول مقید به تنکی، وزن قوانین سیستم‌های فازی را نسبت به خطای حداقل مربعات اندازه‌گیری شده بهینه می‌کند. بسته به آستانه تنکی، وزن بسیاری یا تعدادی از قوانین را می‌توان صفر کرد تا به این ترتیب برخی قوانین از بین برود (انتخاب قانون). مرحله سوم، از طریق یک رویه بهینه‌سازی منظم‌شده مقید به تنکی به صورت مجزا به هر قانون، پارامترهای نتیجه‌دهنده خطی را حذف می‌کند (روش یادگیری محلی). قیود تنکی برای صفرکردن اجباری پارامترهای خطی به کار می‌روند که موجب یک سازوکار انتخاب ویژگی برای هر قانون را می‌شود. هرگاه پارامترهای خطی برخی از ویژگی‌ها در هر قانون برابر (نزدیک) صفر باشند، انتخاب ویژگی بهینه سراسری حاصل می‌شود. نهایتاً برای نشان‌دادن عملکرد

در [۴۴]، یک سیستم خبره فازی مبتنی بر بهینه‌سازی انبوه ذرات (PSO) برای تشخیص بیماری شریان اکلیلی (CAD) ارائه شده است. سیستم طراحی شده از اطلاعات واقعی پایگاه داده‌های بیماری‌های قلبی کلوند<sup>۷</sup> در مجارستان بهره برده است. از آنجاکه این پایگاه داده‌ها دربردارنده ویژگی‌های ورودی متعددی است، درخت تصمیم (DT) برای بازیابی خصوصیات که به تشخیص بیماری مربوطاند به کار گرفته شده است. برای تنظیم توابع عضویت (MFهای) فازی از بهینه‌سازی انبوه ذرات (PSO) استفاده شده است. مزیت اصلی این روش در مقایسه با دیگر روش‌ها، توانایی آن در تفسیر تصمیماتی است که از جانب سیستم خبره فازی تولید شده گرفته شده است.

در [۴۵]، نشان داده شده است که تنظیم بهینه کنترل‌کننده‌ها با استفاده از سه الگوریتم بهینه‌سازی تکاملی به یک کلاس از سیستم‌های servo اختصاص داده می‌شود (الگوریتم جست‌وجو گرانشی "GSA" الگوریتم بهینه‌سازی انبوه ذرات "PSO" و الگوریتم شبیه‌سازی حرارتی "SA"). فرآیندها در این سیستم servo با مدل‌های مرتبه دوم و یک جزء انتگرال و پارامترهای متغیر مشخص می‌شوند؛ بنابراین، توابع هدف در مسائل بهینه‌سازی دربردارنده توابع خروجی از مدل‌های تعریف شده با توجه به تغییرات پارامتری فرآیندها هستند. سیستم‌های servo با کنترل‌کننده‌های متناسب انتگرال - فازی (Takagi-Sugeno) T-S PI- FCS دربردارنده دو ورودی، توابع عضویت مثلثی، نه قانون در قانون پایه، عملگرهای SUM و PROD در موتور استنتاج و روش میانگین وزن دار شده در مازول غیرفازی ساز هستند. T-S PI-FCS به صورت کنترل‌کننده‌های فازی کم‌هزینه به دلیل ساختار ساده، از تنها سه پارامتر تنظیم به دلیل نداشت پارامترهای کنترل‌کننده متناسب انتگرال خطی (PI) بر روی پارامترهای فازی از نظر اصل هم‌ارزی معین و از روش بهینه‌سازی متقارن گسترش یافته پیاده‌سازی می‌شوند. نهایتاً با مقایسه الگوریتم‌های تکاملی در چارچوب یک مطالعه موردی متمرکز بر تنظیم بهینه T-S PI-FCS برای سیستم کنترل موقعیت تجهیزات آزمایشگاهی سیستم servo، نتایج حاصل بیانگر عملکرد رضایت‌بخش سیستم

HESM، یک کنترلر فازی در ناحیه سرعت زیاد به کار گرفته شده است و در همین حین برای بهبود راندمان و بهینه‌سازی جریان، الگوریتم PSO در طی عملکرد حالت دائم در کنترلر فازی اعمال می‌شود. در شبیه‌سازی نیز نشان داده شده است که سیستم HESM با راندمان بالاتر نسبت به سایر روش‌ها ارائه شده است.

برای تولید خودکار ساختار و پارامترهای اساس قانون فازی، در [۲۵]، یک الگوریتم بهینه‌سازی انبوه ذرات با طول مختلف ذرات (DLPPSO) ارائه شده است. مزیت اصلی این روش آن است که ساختار و پارامترهای اساس قانون فازی را می‌توان به طور خودکار با PSO ارائه‌شده تولید کرد. در این روش، بهترین سازگاری ( $f_{gbest}$ ) و بهترین تعداد قوانین فعال ( $N_{gbest}$ ) از بهترین ذره در جمعیت کنونی و بهترین سازگاری ( $f_{pbest}$ ) که ذره  $i$  ام تاکنون به دست آورده است و بهترین تعداد قوانین فعال آن ( $N_{pbest}$ ) موقع بروز بهترین موقعیت، به منظور تعیین قوانین فعال ذره  $i$  ام در هر نسل به کار می‌روند. برای افزایش تنوع ساختار، عملگر جهش برای تغییر تعداد قوانین فعال ذرات به کار گرفته می‌شود. این الگوریتم در مقایسه با دیگر PSOهای با طول مختلف ذرات، عملکرد تطبیقی بهتری دارد. نهایتاً برای نشان دادن کارایی الگوریتم بیان شده، یک تابع غیرخطی و دو مجموعه زمانی در آزمایش‌های شبیه‌سازی به کار می‌رود و نتایج به دست آمده نشان‌دهنده این است که روش ارائه‌شده توانسته است به طور مؤثرتری تابع غیرخطی را تقریب بزند و مجموعه‌های زمانی را پیش‌بینی کند.

الگوریتم ژنتیک به دلیل روش جست‌وجوی ابتکاری فوق‌العاده بسیار معروف است. سیستم فازی ژنتیک به دلیل توانایی بهینه‌سازی با سادگی مفهومی باعث تسهیل بهینه‌سازی قوانین می‌شود. در [۴۳]، بهینه‌سازی چهارده قانون فازی برای اتوماسیون تصمیم‌گیری نیمه خبره در مورد تخمین مدت زمان مبتنی بر فعالیت اولیه در مدیریت پروژه نرم‌افزاری ارائه شده است. هدف از این بهینه‌سازی، کاهش پیچیدگی عبارات زبانی و بهبود دقت تخمین مجموعه قانون فازی و در عین حال، حفظ میزان یکسانی از تفسیرپذیری بوده است.

ارائه شده است.

با کاهش زمان اجرای حجم کاری و تخصیص هوشمند سیستم‌های فازی، باعث بهبود عملکرد شبکه می‌شود. تصمیمات برنامه‌ریزی براساس دانش سیستم فازی از این روش گرفته می‌شود. نهایتاً این مقاله نتایج مقایسه عملکرد SFSSهای مختلف را ارائه داده است و SFSS و سیستم‌های فازی ژنتیک را مقایسه کرده است. نتایج شبیه‌سازی بیانگر این است که SFSS می‌تواند به همگرایی سریع‌تر و با کیفیت بالاتر و کاهش تعداد پارامترهای کنترلی (که آن‌ها را جایگزین خوبی برای سیستم‌های فازی ژنتیک می‌سازد) دست یابد. PSO در کاربردهای دیگر از تحقیقات اخیر به‌طور وسیعی استفاده شده است [53, 54].

### ۳- الگوریتم‌های بهینه‌سازی فراابتکاری

روش‌های بهینه‌سازی، روش‌های جست‌وجویی هستند که اهداف آن‌ها یافتن جوابی برای مسئله بهینه‌سازی است؛ به‌گونه‌ای که کمیت مورد بررسی بهینه شود. براساس شواهد و سوابق نتایج، به‌طور متوسط، بهترین تقابل کیفیت و زمان برای بهینه‌سازی سیستم‌های فازی با استفاده از الگوریتم‌های فراابتکاری فراهم می‌شود.

#### ۳-۱- الگوریتم تکامل تفاضلی

الگوریتم تکامل تفاضلی (DE) یکی از روش‌های مؤثر بر مبنای جست‌وجو است [۱۳]. همانند سایر الگوریتم‌های تکاملی، این الگوریتم نیز با ایجاد یک جمعیت اولیه شروع به کار می‌کند. سپس با اعمال عملگرهایی شامل ترکیب، جهش و تقاطع نسل، نوزاد تشکیل می‌شود و در مرحله بعد که مرحله انتخاب نام دارد، نسل نوزاد با نسل والد برای تعیین میزان شایستگی (که تابع هدف آن را می‌سنجد) مقایسه می‌شود. سپس بهترین اعضا به‌عنوان نسل بعدی وارد مرحله بعد می‌شوند. این روند تا رسیدن به نتایج مطلوب ادامه می‌یابد. در این قسمت به‌ترتیب، مراحل عملکرد این الگوریتم بیان شده است.

#### ایجاد جمعیت اولیه:

تعداد متغیرهای مسئله در این الگوریتم با  $D$  نشان داده شده است. هر یک از این متغیرها یک حد بالا و یک حد

انتخاب پروژه، فرآیند تصمیم‌گیری پیچیده‌ای است که با اهداف متعدد و اغلب متضاد تحت تأثیر قرار می‌گیرد. پیچیدگی مشکل انتخاب پروژه با توجه به بالابودن تعداد پروژه‌ها است؛ زمانی که از یک مجموعه از پروژه‌ها، نمونه مناسب باید انتخاب شود.

در [۴۶]، استراتژی جست‌وجوی پارامترهای سیستم فازی T-S جدید به نام OptiFel با یک PSO ناهمگن و چند گروه (MsPSO) به‌منظور افزایش عملکرد جست‌وجو توسعه داده شده است. MsPSO کل جمعیت را به زیرگروه‌های اشتراکی متعدد گروه‌بندی می‌کند تا روش‌های مختلف جست‌وجو را برای راه‌حل‌های بالقوه انجام می‌دهد. در این مقاله، مشخص شده است که استراتژی زیرگروه‌های متعدد پیشنهادی تا حد زیادی برای پیدا کردن پارامترهای بهینه مناسب برای زیرفضاهای مدل فازی T-S مفید است. اثبات نظری این موضوع نیز نشان داده است که اشتراک میان زیرگروه‌ها می‌تواند تعادلی بین اکتشاف و بهره‌برداری برای اطمینان از ذرات همگرا برای نقاط پایدار را حفظ کند. نتایج تجربی بیانگر این است که MsPSO به‌طور چشمگیری بهتر از الگوریتم‌های سنتی PSO در شش تابع معیار، عمل می‌کند و رویکرد OptiFel می‌تواند یک مدل سیستم فازی خوب و با دقت بالا و توانایی تعمیم قوی را تولید کند.

در [۴۷]، استفاده از کسب دانش بیولوژیکی برای سیستم‌های فازی متکی بر گروه بهینه‌سازی انبوه اطلاعات ذرات (SI-PSO) پیشنهاد شده است. مدل‌های مبتنی بر جمعیت به وجود دانش به‌صورت ذرات توجه می‌کنند که در فضا برای رسیدن به کیفیت بالاتر حرکت می‌کنند. سیستم‌های فازی SI-PSO برای کسب دانش در این کار به‌صورت سیستم‌های فازی انبوه (SFSS)، دسته‌بندی می‌شوند. به‌طور خاص، دو روش آموزش KASIA (با استفاده از پایگاه‌های قانون به‌صورت ذرات در PSO) و KARP (با استفاده از قوانین به‌صورت ذرات در PSO) معرفی می‌شوند. عملکرد SFSS به‌صورت یک مسئله مهم عملی با مجموعه‌های داده و آموزش خوشه‌بندی فازی در شبکه‌های محاسباتی مطالعه شده است. خوشه‌بندی فازی،

$$Z_{i,g+1} = \arg \max(f(z_{i,g}), z_{i,g} + 1) \quad (۴)$$

در این رابطه، اندیس  $g$  نشان‌دهنده نسل است و  $Z_{i,g+1}$  جمعیت نسل جدید (نوزادها) و  $Z_{i,g}$  جمعیت نسل قبل (والدها) هستند. تابع  $f$  همان تابع هدف مسئله است.

تکرار:

انجام مراحل ۲ و ۳ تا رسیدن به ماکزیمم تکرار یا همگرایی کل جمعیت.

### ۳-۲- الگوریتم تکامل تفاضلی مبتنی بر تضاد:

به‌طورکلی در رویکردهای بهینه‌سازی الگوریتم تکاملی، یک حدس تصادفی متحد برای جمعیت اولیه در نظر گرفته می‌شود. در هر نسل، هدف، شامل حرکت به سمت راه‌حل مطلوب است و روند تحقیق، زمانی خاتمه می‌یابد که برخی از معیارهای از پیش تعیین شده، قانع‌کننده باشند. زمان محاسبه معمولاً به حدس اولیه بستگی دارد؛ یعنی هرچه فاصله میان حدس اولیه و راه‌حل مطلوب بیشتر باشد، زمان بیشتری برای خاتمه طول می‌کشد و برعکس. یادگیری مبتنی بر تضاد، شانس شروع با جمعیت اولیه بهتر را از طریق بررسی راه‌حل‌های مخالف بیشتر می‌کند. رویکرد مشابه این می‌تواند نه تنها در راه‌حل‌های اولیه استفاده شود، بلکه می‌تواند در جمعیت حاضر برای هر راه‌حلی به‌طور مداوم به کار گرفته شود [۱۹].

تعریف عدد مخالف: فرض کنید  $x \in [a, b]$  یک عدد واقعی باشد، عدد مخالف  $\tilde{x}$  است که  $\tilde{x} = a + b - x$  آن را تعریف می‌کند.

تعریف نقطه مخالف:

فرض کنید  $p = (x_1, x_2, \dots, x_d)$  نقطه‌ای در فضای  $D$  بعدی باشد که در آن  $x_1, x_2, \dots, x_d \in R$  و  $x_i \in [a_i, b_i]$  نقطه مخالف عبارت است از:  $\tilde{p} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_d)$  where  $\tilde{x}_i = a_i + b_i - x_i$

پایین دارند. جمعیت اولیه در اندازه  $N_p$  در  $D$  به‌صورت تصادفی مطابق رابطه (۱) تشکیل می‌شود.

$$X_{i0} = X_i \min + \text{round}(\delta_i \cdot (X_i \max - X_i \min)) \quad (۱)$$

$i = 1, 2, \dots, N_p$

که  $\delta_i$  عددی تصادفی در بازه  $(0, 1]$ ،  $X_i \max$  و  $X_i \min$  حدود بالا و پایین متغیرهای مسئله و  $N_p$  تعداد اعضا هستند.

### جهش و همبری:

در این الگوریتم می‌توان از پنج استراتژی برای ترکیب و تولید نوزاد بهره جست [۱۳]. در این مقاله از استراتژی شخص بهترین - شخص رندوم - شخص رندوم، برای انجام جهش به‌صورت زیر استفاده شده است.

$$Z_{i,G} = X_{best,G} + F \cdot (X_{r1,G} - X_{r2,G} + X_{r3,G} - X_{r4,G}) \quad (۲)$$

که در این رابطه،  $F$  فاکتور معیار نامیده می‌شود.  $X_r$  ها اعضای انتخابی به‌صورت تصادفی هستند. همچنین بهترین عضو جمعیت حاضر است.

برای هر متغیر از هر عضو جمعیت یک عدد رندوم  $K$  در بازه  $[1, D]$  و یک عدد تصادفی  $u$  در بازه  $[0, 1]$  انتخاب شده اس. همبری مطابق با رابطه زیر انجام می‌شود:

$$\text{if } u \leq CR \text{ or } j = k$$

$$\text{then } Z_{i,j} = X_{r1,j} + F \cdot (X_{r3,j} - X_{r2,j}) \quad (۳)$$

$$\text{else } Z_{i,j} = X_{i,j}$$

که  $j$  شماره هر متغیر از  $i$  امین عضو جمعیت است. همچنین عدد  $CR$  ثابت همبری است و عددی بین ۰ و ۱ انتخاب می‌شود.

مرحله تخمین و انتخاب:

در این مرحله، نوزادها و والدها با توجه به تابع هدف، ارزش‌دهی می‌شوند و در صورتی که نوزاد، ارزشی بیش از والد داشته باشد، جایگزین والد می‌شود. در غیر این صورت، والد، خود به همراه نسل بعدی به مرحله بعد می‌رود.

## بهینه‌سازی مبتنی بر تضاد:

می‌گردند. در این محدوده فقط یک تکه غذا وجود دارد و پرندگان هم از محل این غذا خبر ندارند؛ ولی در هر لحظه، فاصله خود با محل غذا را می‌دانند. در این حالت، یک استراتژی مناسب برای یافتن محل دقیق غذا، دنبال کردن آن پرنده‌ای است که از بقیه پرندگان به غذا نزدیک‌تر است. در واقع، PSO هم از چنین سناریویی الهام گرفته است و راه‌حلی برای مسائل بهینه‌سازی ارائه می‌دهد. در PSO هر پرنده یک راه‌حل برای مسئله است. تمام پاسخ‌های موجود، یک مقدار شایستگی دارند که آن را از روی تابع شایستگی تعریف شده برای مسئله به دست می‌آورند. هدف این تکنیک، یافتن محلی است که بهترین مقدار شایستگی را در فضای مسئله دارد. این مقدار شایستگی در جهت و سرعت حرکت این پرندگان (پاسخ‌های مسئله) به سمت محل غذا (پاسخ بهینه) تأثیر مستقیم دارد. PSO با تعدادی پاسخ اولیه (ذرات) شروع به کار می‌کند و با حرکت دادن این پاسخ‌ها در تکرارهای متوالی، به دنبال یافتن جواب بهینه برای مسئله است. در هر تکرار دو مقدار  $P_{Best}$  و  $G_{Best}$  مشخص می‌شوند.

$P_{Best}$ : مکان بهترین مقدار شایستگی که هر ذره در طول حرکت خود، به آن رسیده است.

$G_{Best}$ : مکان بهترین شایستگی ذره در جمعیت فعلی. پس از یافتن مقادیر فوق، سرعت حرکت ذرات از رابطه (۴) و نیز مکان بعدی هر ذره از رابطه (۵) محاسبه می‌شود.

$$V_{it+1} = w.v_{it-1} + c_1.r_1.(P_{best_t} - P_{it}) \quad (5)$$

$$+ c_2.r_2.(G_{best_t} - P_{it}) \quad (6)$$

$$P_{t+1} = P_t + v_t$$

در این روابط، مقادیر  $r_1$  و  $r_2$  اعدادی تصادفی بین صفر و یک هستند و ضرایب  $c_1$  و  $c_2$  هم که ضرایب یادگیری نامیده می‌شوند را معمولاً برابر دو مقداردهی می‌شود.

در هر تکرار از الگوریتم، سرعت حرکت ذرات (میزان تغییرات برای هر ذره) در هر بعد را می‌توان با یک مقدار از پیش تعیین شده  $V_{max}$  محدود کرد. در این حالت اگر

فرض کنید  $p = (x_1, x_2, \dots, x_d)$  یک نقطه در فضای  $d$  بعدی است؛ یعنی یک راه‌حل انتخابی در نظر بگیرد  $F(0)$  تابع تناسب است که برای اندازه‌گیری تناسب انتخاب‌شده‌ها به کار گرفته می‌شود. بر طبق تعریف نقطه مخالف  $\tilde{p} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_d)$ ، اکنون اگر  $p = (x_1, x_2, \dots, x_d)$  است. می‌تواند به وسیله  $f(\tilde{p}) \geq f(p)$  باشد، آنگاه نقطه  $p$  جایگزین شود؛ در غیر اینصورت با  $p$  ادامه می‌دهیم. از این‌رو، نقطه و نقطه مخالف آن به‌طور هم‌زمان ارزیابی می‌شوند تا با آن‌هایی که مناسب‌تر هستند ادامه دهیم.

شکل (۱) نشان‌دهنده شبه کد الگوریتم تکامل تفاضلی مبتنی بر تضاد است [۲۱]. در این الگوریتم به‌منظور تعیین اعضای جمعیت اولیه، ابتدا برای هر یک از ژن‌های تشکیل‌دهنده اعضا، جمعیت  $x_i(t)$  همانند ژن  $x_{ij}(t)$  یک حد بالا  $x_{max}$  و یک حد پایین  $x_{min}$  تعریف می‌شود؛ سپس ژن‌ها به‌طور تصادفی بین حد بالا و حد پایین به صورت  $x_{ij}(t) \approx U(x_{min}, x_{max})$  تعریف می‌شوند. برای استفاده از اعداد متضاد در طول آغاز جمعیت، به‌ازای هر  $x_{ij}(t)$  یک  $\tilde{x}_{ij}(t)$  در نظر می‌گیریم که با  $\tilde{x}_{ij}(t) = x_{min} + x_{max} - x_{ij}(t)$  تعریف می‌شود. اگر تابع هزینه  $\tilde{x}_{ij}(t)$  کمتر از تابع هزینه  $x_{ij}(t)$  شد (یعنی اگر  $f(x_{ij}(t)) \geq f(\tilde{x}_{ij}(t))$  باشد)، آنگاه نقطه  $x_{ij}(t)$  می‌تواند به‌وسیله  $\tilde{x}_{ij}(t)$  جایگزین شود؛ در غیر اینصورت با  $x_{ij}(t)$  ادامه می‌دهیم. از این‌رو، ژن و ژن مخالف آن به‌طور هم‌زمان ارزیابی می‌شوند تا با آن‌هایی که مناسب‌تر هستند، ادامه دهیم.

## ۳-۳- الگوریتم بهینه‌سازی انبوه ذرات

الگوریتم بهینه‌سازی انبوه ذرات (PSO) براساس رفتار اجتماعی دسته‌های پرندگان عمل می‌کند [۱۴]. برای درک بهتر این تکنیک، سناریوی زیر را در نظر بگیرید: «دسته‌ای پرنده به‌طور تصادفی در محدوده خاصی به دنبال غذا



به معنای به‌کارگیری هم‌زمان دو یا چند روش برای استفاده از مزایا و کم‌رنگ‌کردن معایب آن‌ها است. در الگوریتم پیشنهادی، جمعیت اولیه در ابتدا به صورت تصادفی تولید می‌شوند؛ سپس با اعمال مفهوم یادگیری مبتنی بر تضاد، راه‌حل‌هایی از همین جمعیت نیز، بررسی و مناسب‌ترین اعضا در مرحله اول با مقایسه مقادیر تابع هزینه انتخاب می‌شوند. این موضوع، شانس بازدید و یا حتی بازبینی مناطق کشف‌نشده از فضای جست‌وجو را فراهم می‌آورد و منجر به سرعت‌بخشیدن به میزان همگرایی و دستیابی به یک تقریب بهتر در الگوریتم تکامل تفاضلی خواهد شد؛ در واقع، زمان محاسبه در الگوریتم‌های تکاملی به حدت اولیه بستگی دارد و هرچه فاصله میان حدت اولیه و راه‌حل مطلوب کمتر باشد، زمان کمتری برای خاتمه طول می‌کشد. یادگیری مبتنی بر تضاد، شانس شروع با جمعیت اولیه بهتر را از طریق بررسی راه‌حل‌های متفاوت بیشتر می‌کند. به این صورت که پس از ارزیابی جمعیت اولیه و اعمال مفهوم مبتنی بر تضاد، جمعیت جدید به وجود می‌آید و با ارزیابی مجدد تابع هزینه، مناسب‌ترین اعضا انتخاب می‌گردند. سپس تعدادی از اعضای جمعیت نهایی به‌عنوان والد، استفاده می‌شوند و عملگرهای جهش و تقاطع الگوریتم تکامل تفاضلی بر روی آن‌ها اعمال می‌شوند و جمعیت فرزندان به وجود می‌آید. با در نظر گرفتن مفهوم یادگیری مبتنی بر تضاد به روش قبل و مشخص شدن مناسب‌ترین فرزندان در این مرحله، در ادامه، بهترین اعضای جمعیت والدین و فرزندان (هر عضوی که دارای مقدار تابع هزینه کمتری باشد) برای حضور در مرحله بعدی انتخاب می‌شوند. سپس عملگرهای الگوریتم بهینه‌سازی انبوه ذرات با الگوی جست‌وجوی محلی بر روی این جمعیت جدید اعمال می‌شوند و همین چرخه تا حصول شرایط خاتمه رویکرد پیشنهادی تکرار می‌شود. جزئیات دقیق گام‌های اجرایی، فلوجارت الگوریتم و شبه کد الگوریتم ترکیبی تکامل تفاضلی مبتنی بر تضاد و بهینه‌سازی انبوه ذرات در شکل (۲) نشان داده شده است [۵۱].

سرعت هر ذره در هر بعد، از این حد فراتر رفت، با  $V_{max}$  جایگزین می‌شود.

شمارشگر تعداد تکرارها را برابر صفر قرار می‌دهیم.  $t=0$

پارامترهای کنترلی  $P_r$ ،  $\beta$  را مقداردهی می‌کنیم.

تعداد  $n_s$  عضو را به وجود می‌آوریم و آن را به‌عنوان اعضای جمعیت اولیه  $C(0)$  در نظر می‌گیریم.

تا هنگامی که شرایط خاتمه برآورده نشده‌اند، مراحل زیر را انجام می‌دهیم:

به ازای هر عضو  $X_i(t)$  که عضو  $C(t)$  است، یک  $\tilde{X}_i(t)$  به دست می‌آوریم و مراحل زیر را انجام می‌دهیم:

تابع هزینه  $f(X_i(t))$  و  $f(\tilde{X}_i(t))$  را محاسبه می‌کنیم و با مناسب‌ترین اعضا مراحل زیر را ادامه می‌دهیم:

بردار آزمون  $u_i(t)$  را با استفاده از عملگر جهش به وجود می‌آوریم.

عضو فرزند  $X'_i(t)$  را با استفاده از عملگر تقاطعی به دست می‌آوریم.

فرزندان و متضادشان را در تابع هزینه با یکدیگر مقایسه می‌کنیم و بهترین جواب را به دست می‌آوریم.

اگر تابع هزینه فرزند  $f(X'_i(t))$  از تابع هزینه والد  $f(X_i(t))$  بهتر بود، آنگاه:

عضو فرزند  $X'_i(t)$  را به اعضای نسل بعدی  $C(t+1)$  اضافه می‌کنیم.

در غیر این صورت:

عضو والد  $X_i(t)$  را به اعضای نسل بعدی  $C(t+1)$  اضافه می‌کنیم.

پایان اگر.

پایان حلقه تکرار (به ازای).

پایان حلقه تکرار (تا هنگامی که).

ژن  $J$  را از مجموعه  $(0, n_x-1)$  به طور تصادفی انتخاب می‌کنیم و آن را به مجموعه  $J$  اضافه می‌کنیم.

تا هنگامی که  $U(0,1) < p_r // |J| < n_x$  است، مراحل زیر را تکرار می‌کنیم.

ژن  $J+1$  را به مجموعه  $J$  اضافه می‌کنیم.

باقی‌مانده ژن  $J+1$  بر  $n_x$  را به‌عنوان ژن  $J$  قرار می‌دهیم.

پایان حلقه تکرار (تا هنگامی که).

شکل (۱): شبه کد الگوریتم تکامل تفاضلی مبتنی بر تضاد

### ۳-۴- رویکرد ترکیب الگوریتم‌های تکامل

#### تفاضلی مبتنی بر تضاد و بهینه‌سازی انبوه ذرات

در این مقاله، ترکیب الگوریتم‌های تکامل تفاضلی مبتنی بر تضاد و بهینه‌سازی انبوه ذرات<sup>۱</sup> ارائه شده است. ترکیب

### ۴- طراحی سیستم فازی با بهره‌گیری از

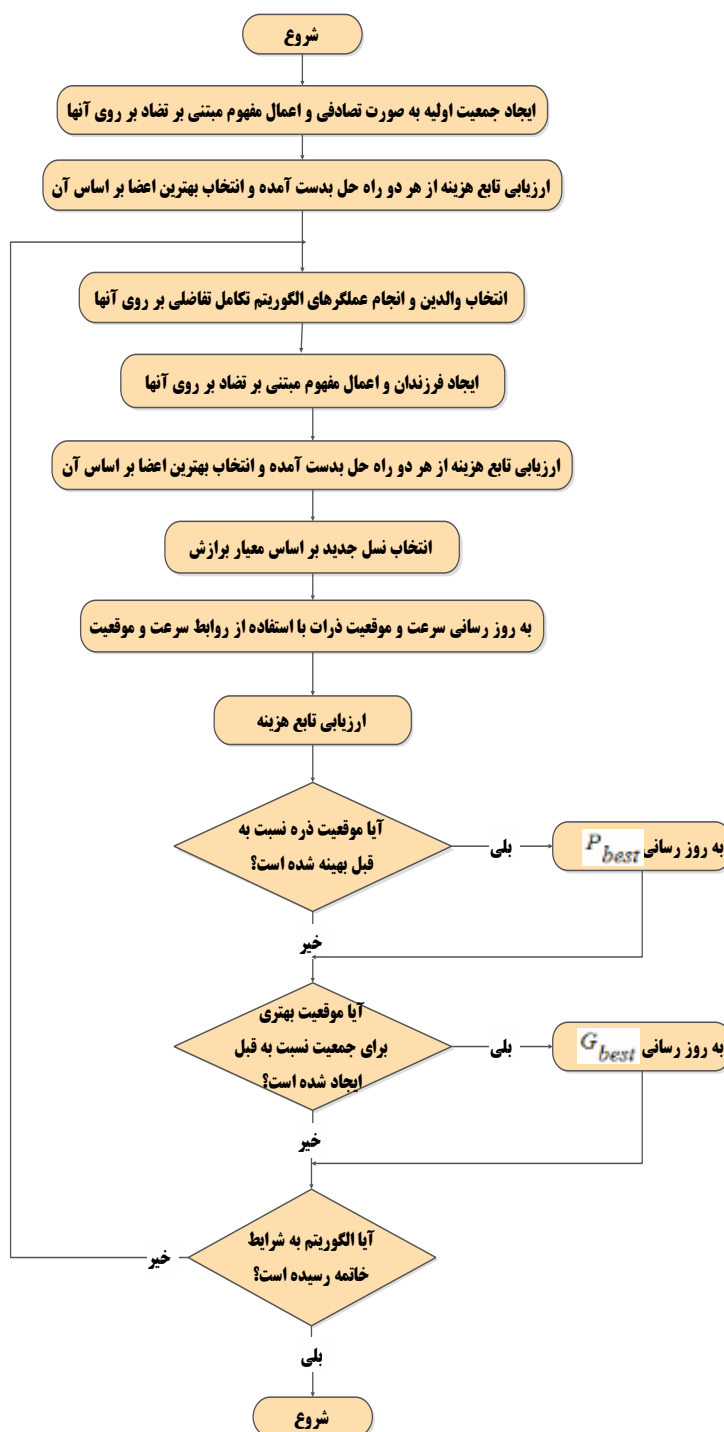
#### رویکرد پیشنهادی

#### ۴-۱- سیستم فازی به‌کار برده شده

سیستم فازی به‌کار برده‌شده از نوع سوگنو است. در این سیستم فازی،  $\hat{A}$  امین قاعده فازی با  $R_i$ ، مشخص و

به‌صورت زیر بیان می‌شود:

$$R_i : \text{If } x_1 \text{ is } A_{i1} \text{ and } \dots \text{ and } x_n \text{ is } A_{in} \text{ Then } y \text{ is } f_i(x_1, \dots, x_n) \quad (V)$$



شکل (۲): فلوجارت الگوریتم ترکیبی تکامل تفاضلی مبتنی بر تضاد و بهینه‌سازی انبوه ذرات

برای طراحی سیستم فازی توضیح داده شده است که در بردارنده خوشه‌بندی کاهشی و الگوریتم ترکیبی تکامل تفاضلی مبتنی بر تضاد و بهینه‌سازی انبوه ذرات است.

طراحی سیستم فازی از داده‌های آموزش در بردارنده تعیین قواعد فازی و بهینه‌سازی همه پارامترها در هر قاعده فازی است. سیستم فازی معادله (۷) که در بردارنده  $r_i$  متغیر ورودی و  $r$  قاعده فازی است، در کل دارای  $r(2n+1)$  پارامتر است که باید بهینه‌سازی شوند. این پارامترها می‌توانند با بردار زیر توصیف شوند:

$$\bar{s} = [m_{11}, \sigma_{11}, \dots, m_{1n}, \sigma_{1n}, a_1, \dots, m_{r1}, \sigma_{r1}, \dots, m_{rn}, \sigma_{rn}, a_r] \in \mathfrak{R}^{r(2n+1)} \quad (11)$$

مراحل روش پیشنهادی دو مرحله‌ای برای طراحی سیستم فازی در زیر بیان شده است:

**مرحله اول:** ابتدا داده‌ها از سیستم به دست می‌آید و سپس ورودی‌های سیستم فازی تعیین می‌شود و در ادامه، این داده‌ها با تکنیک خوشه‌بندی کاهشی به چندین خوشه تقسیم می‌شوند (برای تفکیک‌بندی فضای ورودی از تکنیک خوشه‌بندی کاهشی استفاده می‌شود) که هر خوشه بیان‌کننده یک قاعده فازی در سیستم فازی است. با اتمام فرآیند خوشه‌بندی، مراکز خوشه‌ها می‌توانند برای تشکیل مراکز توابع عضویت در سیستم فازی استفاده شوند. در این مقاله، مراکز توابع عضویت با استفاده از الگوریتم ترکیبی تکامل تفاضلی مبتنی بر تضاد و بهینه‌سازی انبوه ذرات تولید شده‌اند. در این مرحله سیستم فازی خام به دست می‌آید.

**مرحله دوم:** در این مرحله برای بهینه‌سازی پارامترهای بخش مقدم و تالی سیستم فازی خام از الگوریتم ترکیبی تکامل تفاضلی مبتنی بر تضاد و بهینه‌سازی انبوه ذرات توسعه داده شده استفاده می‌شود.

برای مثال، اگر پس از فرآیند خوشه‌بندی دو خوشه به دست آید، این سیستم فازی دارای دو قاعده فازی  $R_1$  و  $R_2$  است که به صورت زیر تعریف می‌شود:

$$\begin{cases} R_1 : \text{If } x_1 \text{ is } A_{11} \text{ and } x_2 \text{ is } A_{12} \text{ Then } y \text{ is } a_1 \\ R_2 : \text{If } x_1 \text{ is } A_{21} \text{ and } x_2 \text{ is } A_{22} \text{ Then } y \text{ is } a_2 \end{cases} \quad (12)$$

که در آن  $x_1, \dots, x_n$  متغیرهای ورودی،  $y$  متغیر خروجی،  $A_{ij}$  یک مجموعه فازی و تابع  $f_i$  در بخش تالی یک سیستم فازی نوع TSK مرتبه صفر به صورت  $a_{i0}$  (مقدار قطعی) و مرتبه یک به صورت  $a_{i0} + \sum_{j=1}^n a_{ij} x_j$  است. مجموعه فازی  $A_{ij}$  به کاررفته، تابع عضویت گوسی است که به صورت زیر تعریف می‌شود:

$$A_{ij}(x_j) = \exp \left\{ - \left( \frac{x_j - m_{ij}}{\sigma_{ij}} \right)^2 \right\} \quad (8)$$

که در آن  $m_{ij}$  مرکز و  $\sigma_{ij}$  پهنای مجموعه فازی گوسی هستند.

اگر در موتور استنتاج، عملگر AND فازی با حاصل ضرب جبری در تئوری فازی پیاده‌سازی شود، برای ورودی معین  $\bar{x} = (x_1, \dots, x_n)$  نیروی آتش یا وزن تأثیر قاعده  $i$  به صورت زیر محاسبه می‌شود:

$$\begin{aligned} \varphi_i(\bar{x}) &= \prod_{j=1}^n A_{ij}(x_j) = \\ &= \exp \left\{ - \sum_{j=1}^n \left( \frac{x_j - m_{ij}}{\sigma_{ij}} \right)^2 \right\} \end{aligned} \quad (9)$$

اگر سیستم فازی دارای  $r$  قاعده باشد، خروجی سیستم فازی با روش غیرفازی سازی میانگین وزن دار شده به صورت زیر محاسبه می‌شود:

$$y = \frac{\sum_{i=1}^r \varphi_i(\bar{x}) f_i}{\sum_{i=1}^r \varphi_i(\bar{x})} \quad (10)$$

که در آن  $\varphi_i(\bar{x})$  نیروی آتش یا وزن تأثیر قاعده  $i$  و  $f_i$  مقدار تالی قاعده فازی در معادله (۷) است.

## ۴-۲- روش پیشنهادی دو مرحله‌ای برای طراحی سیستم فازی

در این بخش، جزئیات روش پیشنهادی دو مرحله‌ای

معادله زیر توصیف شده است:

$$y(k+1) = \frac{y(k)}{1+y^2(k)} + u^3(k) \quad (13)$$

که در آن  $-2 \leq y(k) \leq 2$ ،  $y(0) = 0$ ، سیگنال کنترل و  $u(k) \in [-1, 1]$  است. هدف کنترل ردیابی خروجی مطلوب زیر:

$$y_d(k) = \sin\left(\frac{\pi k}{50}\right) \cos\left(\frac{\pi k}{30}\right), \quad 1 \leq y(k) \leq 250 \quad (14)$$

به وسیله یک کنترل‌کننده فازی است.

کنترل‌کننده فازی طراحی شده برای این مثال، دو متغیر ورودی و یک متغیر خروجی دارد که متغیرهای ورودی این کنترل‌کننده به صورت زیر تعریف می‌شوند:

$$\begin{aligned} \text{Input 1} &= y_d(k+1) \\ \text{Input 2} &= y(k) \end{aligned} \quad (15)$$

مقدار خروجی سیستم فازی  $u(k)$ ، ابتدا به کمک رابطه (۱۰)، محاسبه و سپس برای کنترل خروجی پلنت ارسال می‌شود تا مسیر حرکت مطلوب رابطه (۱۴) تعقیب شود.

پارامترهای سیستم فازی با استفاده از الگوریتم HODEPSO بهینه‌سازی شده‌اند. برای ارزیابی عملکرد کنترل‌کننده فازی طراحی شده، از RMSE استفاده شد که به صورت زیر تعریف می‌شود:

$$RMSE = \sqrt{\frac{\sum_{k=0}^{249} (y_d(k+1) - y(k+1))^2}{250}} \quad (16)$$

پارامترهای الگوریتم HODEPSO برای این مثال در جدول (۱) ارائه شده است. جدول (۲)، RMSE میانگین ۵۰ اجرا را نشان می‌دهد. در شکل (۳)، RMSE میانگین در هر ارزیابی را نشان می‌دهد. شکل (۴) بیان‌کننده بهترین نتیجه کنترلی است که در آن مسیرهای حرکت خروجی کنترل‌شده و مطلوب بسیار به هم نزدیک‌اند.

که در آن  $x_1$  و  $x_2$  متغیرهای ورودی،  $y$  متغیر خروجی،  $A_{ij}$  یک مجموعه فازی و  $a_i$  یک مقدار قطعی است. مجموعه فازی  $A_{ij}$  به کاررفته، تابع عضویت گوسی است که به صورت رابطه (۸) تعریف می‌شود که در آن  $m_{ij}$  مرکز و  $\sigma_{ij}$  پهنای تابع عضویت گوسی هستند.

سیستم فازی معادله (۱۲) که دربردارنده  $n = 2$  متغیر ورودی و  $r = 2$  قاعده فازی است، در کل دارای  $r(2n+1)$  پارامتر است که باید بهینه‌سازی شوند  $(2(2 \times 2 + 1) = 10)$ .

#### ۴-۳- شبیه‌سازی

در این بخش، سیستم فازی طراحی شده با الگوریتم بهینه‌سازی ترکیبی، شبیه‌سازی شده است. همچنین برای ارزیابی عملکرد سیستم فازی طراحی شده با الگوریتم بهینه‌سازی ترکیبی، مقایسه‌ای با سایر روش‌ها انجام شده است.

در این بخش، چهار مثال برای نشان دادن عملکرد روش پیشنهادی ارائه شده است. در مثال اول و دوم، سیستم فازی طراحی شده به عنوان کنترل‌کننده فازی به کار رفته است. در این دو مثال، فرآیندها غیر خطی هستند و داده‌های آموزش به صورت آنلاین در پیشروی تولید می‌شوند و هدف، کنترل و ردیابی مسیر مطلوب است. همچنین سیستم فازی به کار برده شده در مثال اول و دوم، نوع سوگنو مرتبه صفر است. در ادامه، دو مثال مشهور که برای ارزیابی عملکرد و قابلیت‌های انواع مدل‌های فازی به کار می‌روند، ارائه شده است. در مثال سوم، سیستم فازی طراحی شده با روش پیشنهادی برای پیش‌بینی مقادیر آینده سری زمانی آشوبناک مکی-گلس<sup>۱۱</sup> و در مثال چهارم، برای پیش‌بینی نرخ رهن خانه<sup>۱۱</sup> به کار رفته است. درخور ذکر است در مثال‌های سوم و چهارم، سیستم فازی به کار برده شده از نوع سوگنو مرتبه یک است.

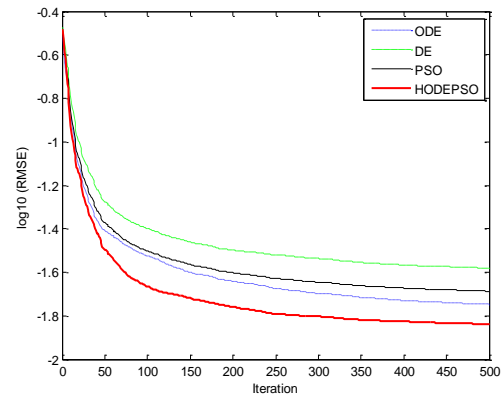
#### ۴-۳-۱- مثال اول: کنترل ردیاب پلنت غیرخطی

فرآیند تحت کنترل در مراجع [۳۱] و [۳۲] و [۴۸] با

جدول (۲): مقایسه نتایج روش‌های مختلف برای مثال کنترل

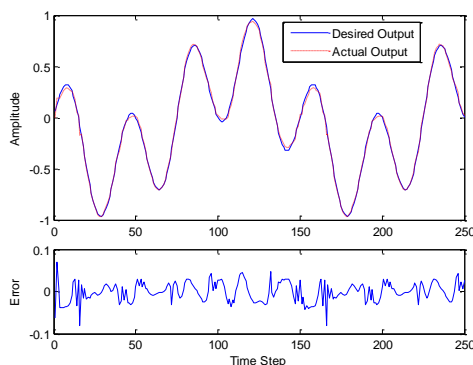
ردیاب پلنت غیرخطی

میانگین RMSE	تعداد قوانین	روش
0.0260	5	[31]
0.0330	5	[32]
0.0223	5	[48]
0.0202	4	Our Method (HODEPSO)
0.0348	4	Method (ODE)
0.0367	4	Method (PSO)
0.0417	4	Method (DE)



شکل (۳): مقادیر RMSE به دست آمده در هر تکرار با

الگوریتم ترکیبی، الگوریتم بهینه‌سازی انبوه ذرات، الگوریتم تکامل تفاضلی و الگوریتم تکامل تفاضلی مبتنی بر تضاد برای مثال کنترل ردیاب پلنت غیرخطی



شکل (۴): نتیجه کنترل کننده فازی با استفاده از روش پیشنهادی

برای مثال کنترل ردیاب پلنت غیرخطی

عملکرد HODEPSO با عملکرد ODE، PSO و DE مقایسه شد که در مسائل کنترلی یکسانی به کار رفتند. در کارهای قبلی انجام شده در مراجع [۳۱] و [۳۲] و [۴۸] بر روی این مثال، تعداد قواعد فازی برابر با ۵، تعداد کل پارامترهای بهینه‌سازی برابر با ۲۵ و کمترین RMSE به دست آمده برای آن‌ها برابر با ۰.۰۲۲۳ بود؛ ولی در روش پیشنهادی، تعداد قواعد فازی برابر با ۴، تعداد کل پارامترهای بهینه‌سازی برابر با ۲۰ است.

در روش پیشنهادی برای این مثال، علاوه بر سادگی سیستم فازی (کاهش تعداد قواعد فازی)، مقدار RMSE نیز به ۰.۰۲۰۲ کاهش یافت. با توجه به نتایج ارائه شده در جدول ۲، روش پیشنهادی به RMSE متوسط کمتری در مقایسه با سایر روش‌ها دست یافته است.

۴-۳-۲- مثال دوم: کنترل درجه حرارت آب حمام

کنترل درجه حرارت آب حمام در مراجع [۳۱] و [۳۲]

جدول (۱): پارامترهای روش پیشنهادی برای مثال کنترل

ردیاب پلنت غیرخطی

پارامترهای HODEPSO برای مثال ۱		
1	PopSize	10
2	Beta_min	0.2
3	beta_max	0.8
4	P <sub>r</sub>	0.2
5	C <sub>1</sub>	1.49
6	C <sub>2</sub>	1.49
7	W	0.72
8	MaxIt	500

با توجه به نتایج شبیه‌سازی در شکل (۳) مشاهده می‌شود که الگوریتم ترکیبی با سرعت بیشتری به جواب بهینه همگرا می‌شود و عملکرد بهتری نسبت به الگوریتم تکامل تفاضلی مبتنی بر تضاد، الگوریتم بهینه‌سازی انبوه ذرات و الگوریتم تکامل تفاضلی دارد.

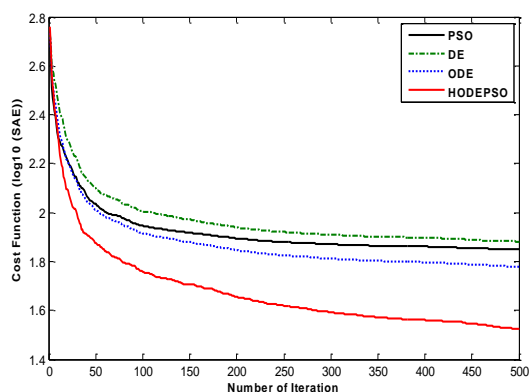
همین مثال در مراجع [۳۱] و [۳۲] و [۴۸] مطالعه شده است. مقایسه نتایج به دست آمده از این روش‌ها و روش پیشنهادی در جدول (۲) ارائه شده است. همچنین نتیجه کنترل کننده فازی طراحی شده با روش پیشنهادی در شکل (۴) نشان داده شده است.

تضاد و بهینه‌سازی انبوه ذرات در مرحله دوم روش پیشنهادی (بهینه‌سازی پارامترهای سیستم فازی)، هر یک از این الگوریتم‌ها به صورت جداگانه به کار رفتند. نتایج به‌کارگیری این الگوریتم‌ها در مرحله دوم روش پیشنهادی با میانگین 50 بار اجرا در شکل 5 نشان داده شده است.

جدول (3): پارامترهای روش پیشنهادی برای مثال کنترل درجه

حرارت آب حمام

پارامترهای HODEPSO برای مثال 2		
1	PopSize	30
2	beta_min	0.2
3	beta_max	0.8
4	P <sub>r</sub>	0.2
5	C <sub>1</sub>	1.49
6	C <sub>2</sub>	1.49
7	W	0.72
8	MaxIt	500



شکل (5): مقادیر SAE به دست آمده در هر تکرار با الگوریتم ترکیبی، الگوریتم بهینه‌سازی انبوه ذرات و الگوریتم تکامل تفاضلی و الگوریتم تکامل تفاضلی مبتنی بر تضاد برای مثال کنترل درجه حرارت آب حمام

همین مثال در مراجع [31] و [32] و [48] مطالعه شده است. مقایسه نتایج به دست آمده از این روش‌ها و روش پیشنهادی در جدول (4) ارائه شده است. همچنین نتیجه کنترل کننده فازی طراحی شده با روش پیشنهادی در شکل (6) نشان داده شده است.

و [48] با معادله زیر توصیف شده است:

$$y(k+1) = \alpha(T_s)y(k) + \frac{b(T_s)}{1+e^{0.5y(k)-\gamma}}u(k) + [1-\alpha(T_s)]Y_0 \quad (17)$$

که در آن

$$\alpha(T_s) = e^{-\alpha T_s} \quad (18)$$

$$b(T_s) = \frac{\beta}{\alpha}(1-e^{-\alpha T_s}) \quad (19)$$

که  $u(k)$  ورودی کنترل پلنت و  $0 \leq u(k) \leq 5$  است. پارامترهای این مثال به صورت  $\alpha = 1.00151 \times 10^{-4}$ ،  $\beta = 8.67973 \times 10^{-3}$ ،  $\gamma = 40$ ،  $Y_0 = 25^\circ C$  و تناوب نمونه برداری  $T_s = 30$  ثانیه است. درجه حرارت آب مرجع  $y_d(k)$  به صورت زیر توصیف شده است:

$$y_d(k) = \begin{cases} 34, & k \leq 30 \\ 34 + 0.5(k-30), & 30 < k \leq 50 \\ 44 + 0.8(k-50), & 50 < k \leq 70 \\ 60 + 0.5(k-70), & 70 < k \leq 90 \\ 70, & 90 < k \leq 120 \end{cases} \quad (20)$$

کنترل کننده فازی طراحی شده برای این مثال، دو متغیر ورودی و یک متغیر خروجی دارد. متغیرهای ورودی این کنترل کننده به صورت زیر تعریف شده‌اند:

$$\begin{aligned} \text{Input 1} &= (y(k+1) - 50) / 20 \\ \text{Input 2} &= (y_d(k+1) - 50) / 20 \end{aligned} \quad (21)$$

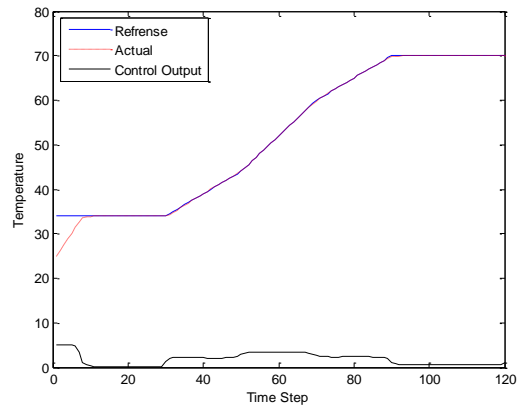
و متغیر خروجی آن  $u(k)$  است که در آن، متغیرهای ورودی به بازه  $[-2, 2]$  و متغیر خروجی به بازه  $[0, 5]$  محدود شده است. برای ارزیابی عملکرد کنترل کننده فازی طراحی شده با روش پیشنهادی از  $SAE^{12}$  استفاده شد که به صورت زیر تعریف می‌شود:

$$SAE = \sum_{k=0}^{119} |y(k+1) - y_d(k+1)| \quad (22)$$

پارامترهای روش پیشنهادی برای این مثال در جدول (3) ارائه شده است. برای نشان دادن کارایی الگوریتم ترکیبی نسبت به تک تک الگوریتم‌های تکامل تفاضلی مبتنی بر

جدول (۴): مقایسه نتایج روش‌های مختلف برای مثال کنترل درجه حرارت آب حمام

	روش	تعداد قوانین	میانگین SAE
1	[31]	9	38.9
2	[32]	9	78.5
3	[48]	9	35.4
4	Our Method (HODEPSO)	8	29.6
5	Method (ODE)	8	54.3
6	Method (PSO)	8	74.8
7	Method (DE)	8	79.8



شکل (۶): نتیجه کنترل‌کننده فازی با استفاده از روش پیشنهادی برای مثال کنترل درجه حرارت آب حمام

مقادیر آینده، یک سری زمانی آشوبناک مکی-گلوس به کار رفته است. این سری زمانی با استفاده از معادله دیفرانسیل تأخیری مکی - گلوس به صورت زیر تولید شد:

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \quad (23)$$

مسئله پیش‌بینی سری زمانی، مبتنی بر معادله دیفرانسیل مکی-گلوس، یک معیار مشهوری برای مقایسه قابلیت‌های مدل‌های فازی مختلف است. از سری زمانی آشوبناک مکی - گلوس  $x(t)$ ، 1000 جفت داده ورودی - خروجی استخراج شد. 500 جفت داده اول برای آموزش سیستم فازی به کار رفتند؛ درحالی‌که مابقی 500 جفت داده دیگر به عنوان داده‌های آزمون برای ارزیابی عملکرد مدل فازی در پیش‌بینی به کار رفتند. مدل فازی طراحی شده در این مثال دارای چهار متغیر ورودی همانند مراجع [۴۹] و [۴۸] به صورت زیر است:

$$\begin{aligned} \text{Input 1} &= x(t-24) \\ \text{Input 2} &= x(t-18) \\ \text{Input 3} &= x(t-12) \\ \text{Input 4} &= x(t-6) \end{aligned} \quad (24)$$

به منظور ارزیابی عملکرد مدل فازی طراحی شده از RMSE استفاده شد که به صورت زیر تعریف شده است:

$$RMSE = \sqrt{\frac{1}{n} \sum_{k=1}^n (y_k - \hat{y}_k)^2} \quad (25)$$

که در آن  $r_i$  تعداد داده‌ها،  $y_k$  خروجی واقعی و  $\hat{y}_k$  خروجی مدل فازی هستند.

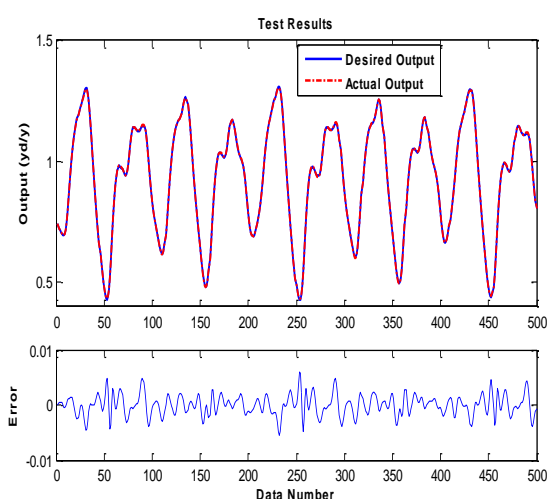
پارامترهای روش پیشنهادی برای این مثال در جدول (۵) ارائه شده است. برای نشان دادن کارایی الگوریتم ترکیبی نسبت به تک‌تک الگوریتم‌های تکامل تفاضلی مبتنی بر تضاد، بهینه‌سازی انبوه ذرات و تکامل تفاضلی در مرحله دوم روش پیشنهادی (بهینه‌سازی پارامترهای سیستم فازی)، هر یک از این الگوریتم‌ها به صورت جداگانه به کار رفتند. نتایج به‌کارگیری این الگوریتم‌ها در مرحله دوم، روش پیشنهادی با میانگین 30 بار اجرا در شکل (۷) نشان داده شده است.

در کارهای قبلی انجام شده در مراجع [۳۱] و [۳۲] و [۴۸] برای این مثال، تعداد قواعد فازی برابر با 9، تعداد کل پارامترهای بهینه‌سازی برابر 45 و کمترین SAE به دست آمده برابر 35.4 بود؛ ولی در روش پیشنهادی برای این مثال، تعداد قواعد فازی برابر با 8 و تعداد کل پارامترهای بهینه‌سازی برابر 40 است و در روش پیشنهادی، علاوه بر سادگی سیستم فازی (کاهش تعداد قواعد فازی)، مقدار SAE نیز به 29.6 کاهش یافت. با توجه به نتایج ارائه شده در جدول ۴، روش پیشنهادی به SAE متوسط کمتری در مقایسه با سایر روش‌ها دست یافته است.

۴-۳-۳- مثال سوم: پیش‌بینی مقادیر آینده سری زمانی آشوبناک مکی - گلوس

در این مثال، سیستم فازی طراحی شده برای پیش‌بینی

همین مثال در مراجع [49] و [48] مطالعه شده است. مقایسه نتایج به دست آمده از این روش‌ها و روش پیشنهادی در جدول (6) ارائه شده است. مقایسه میان خروجی واقعی و خروجی مدل فازی برای داده‌های آموزش در شکل 8 و مقایسه میان خروجی واقعی و خروجی مدل فازی برای داده‌های آزمون در شکل (9) نشان داده شده است.



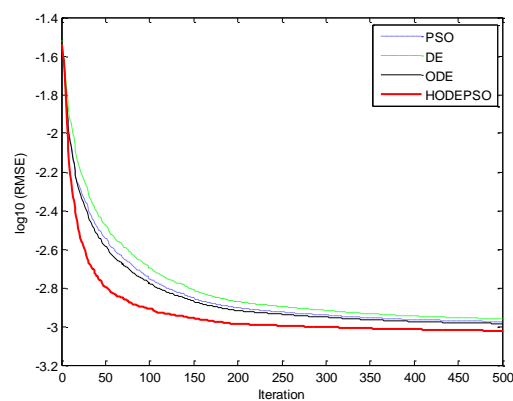
شکل (8): مقایسه میان خروجی واقعی و خروجی مدل فازی برای داده‌های آموزش در مثال پیش‌بینی مقادیر آینده سری زمانی آشوبناک مکی - گلس

جدول (6): مقایسه نتایج روش‌های مختلف برای مثال پیش‌بینی مقادیر آینده سری زمانی آشوبناک مکی - گلس

	روش	تعداد قوانین	RMSE (آموزش)	RMSE (تست)
1	[49]	3	5.380E-3	5.350E-3
2	[48]	4	9.400E-3	6.100E-3
3	[48]	10	---	3.900E-3
4	Our Method (HODEPSO)	10	9.230E-4	9.810E-4
5	Method (ODE)	10	6.510E-3	6.640E-3
6	Method (PSO)	10	9.460E-3	8.520E-3
7	Method (DE)	10	9.780E-3	9.950E-3

جدول (5): پارامترهای روش پیشنهادی برای مثال پیش‌بینی مقادیر آینده سری زمانی آشوبناک مکی - گلس

پارامترهای HODEPSO برای مثال 3		
1	$r_a$	0.45
2	PopSize	30
3	beta_min	0.2
4	beta_max	0.8
5	$P_r$	0.2
6	$C_1$	1.49
7	$C_2$	1.49
8	W	0.72
9	MaxIt	500



شکل (7): مقادیر RMSE به دست آمده در هر تکرار با الگوریتم ترکیبی، الگوریتم بهینه‌سازی انبوه ذرات و الگوریتم تکامل تفاضلی و الگوریتم تکامل تفاضلی مبتنی بر تضاد برای مثال پیش‌بینی مقادیر آینده سری زمانی آشوبناک مکی - گلس

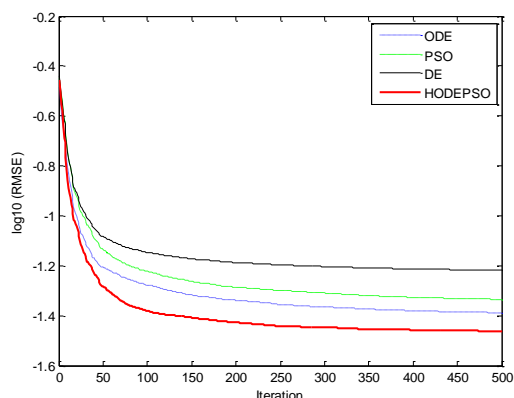
مدل فازی طراحی شده در این مثال، دارای 10 قاعده فازی و هر قاعده فازی دارای 13 پارامتر بهینه‌سازی در بخش مقدم و تالی است؛ بنابراین، تعداد کل پارامترهای بهینه‌سازی برابر 130 است. با توجه به نتایج ارائه شده در جدول (6)، روش پیشنهادی برای داده‌های آموزش به



دوم روش پیشنهادی (بهینه‌سازی پارامترهای سیستم فازی)، هر یک از این الگوریتم‌ها به صورت جداگانه به کار رفتند. نتایج به‌کارگیری این الگوریتم‌ها در مرحله دوم روش پیشنهادی با میانگین 10 بار اجرا در شکل (۱۰) نشان داده شده است.

جدول (۷): پارامترهای روش پیشنهادی برای مثال مدل‌سازی

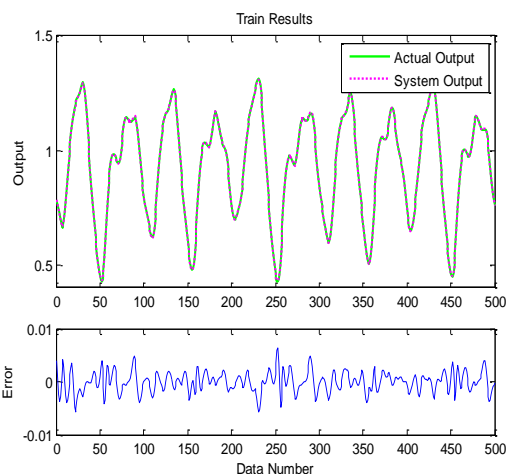
پیش‌بینی رهن خانه		
پارامترهای HODEPSO برای مثال ۴		
1	ra	0.35
2	PopSize	40
3	beta_min	0.2
4	beta_max	0.8
5	Pr	0.2
6	C1	1.49
7	C2	1.49
8	W	0.72
9	MaxIt	500



شکل (۱۰): مقادیر RMSE به دست آمده در هر تکرار با الگوریتم ترکیبی، الگوریتم بهینه‌سازی انبوه ذرات و الگوریتم تکامل تفاضلی و الگوریتم تکامل تفاضلی مبتنی بر تضاد برای مثال مدل‌سازی پیش‌بینی رهن خانه

همین مثال مدل‌سازی در مرجع [۴۹] مطالعه شده است. مقایسه نتایج به دست آمده از این روش‌ها و روش پیشنهادی در جدول (۸) ارائه شده است.

مقایسه میان خروجی واقعی و خروجی مدل فازی برای داده‌های آموزش در شکل (۱۱) و مقایسه میان خروجی واقعی و خروجی مدل فازی برای داده‌های آزمون در شکل (۱۲) نشان داده شده است.



شکل (۹): مقایسه میان خروجی واقعی و خروجی مدل فازی برای داده‌های آزمون در مثال پیش‌بینی مقادیر آینده سری زمانی آشوبناک مکی - گلس

#### ۴-۴-۴- مثال چهارم: پیش‌بینی نرخ رهن خانه

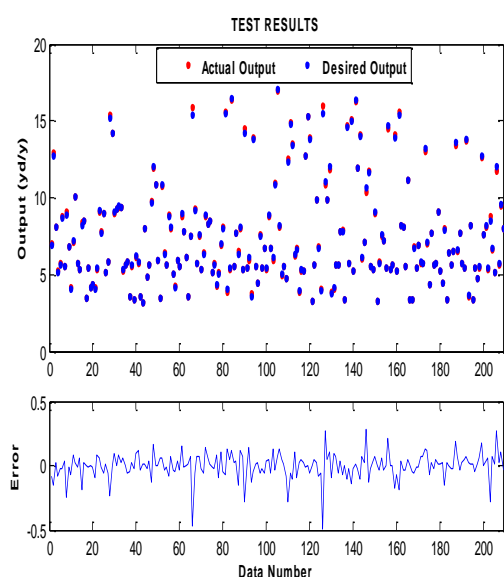
این مثال، یک مسئله اقتصادی واقعی است که نرخ رهن مرسوم ۳۰ ساله در آمریکا را پیش‌بینی می‌کند. در مدل پیش‌بینی، ۱۵ ورودی و یک خروجی وجود دارد. یک پایگاه داده واقعی، دربردارنده ۱۰۴۹ مثال بوده است و آن را می‌توان از سایت Keel به آدرس <http://www.keel.es/> دانلود کرد. ۸۴۰ نمونه اول به عنوان داده‌های آموزش و ۲۰۹ نمونه باقی مانده به عنوان داده‌های تست به کار رفتند. مدل فازی طراحی شده در این مثال، همانند مدل ارائه شده در مرجع [۴۹] است.

برای ارزیابی عملکرد مدل فازی طراحی شده از  $RMSE^{13}$  استفاده شد که به صورت زیر تعریف می‌شود:

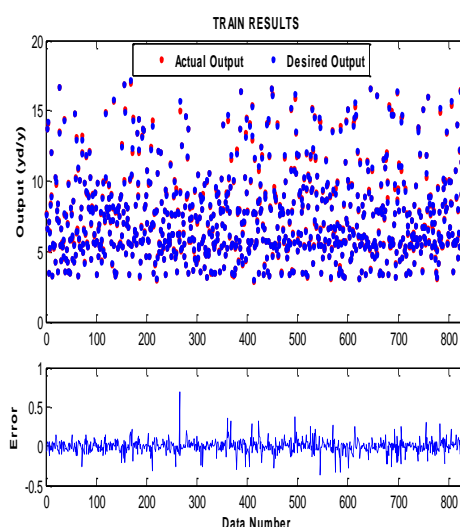
$$RMSE = \sqrt{\frac{1}{n} \sum_{k=1}^n (y_k - \hat{y}_k)^2} \quad (26)$$

که در آن  $r_i$  تعداد داده‌ها،  $y_k$  خروجی واقعی و  $\hat{y}_k$  خروجی مدل فازی هستند.

پارامترهای روش پیشنهادی برای این مثال در جدول (۷) ارائه شده است. برای نشان دادن کارایی الگوریتم ترکیبی نسبت به تک تک الگوریتم‌های تکامل تفاضلی مبتنی بر تضاد، بهینه‌سازی انبوه ذرات و تکامل تفاضلی، در مرحله



شکل (۱۲): مقایسه میان خروجی واقعی و خروجی مدل فازی برای داده‌های آزمون در مثال مدل‌سازی پیش‌بینی نرخ رهن خانه



شکل (۱۱): مقایسه میان خروجی واقعی و خروجی مدل فازی برای داده‌های آموزش در مثال مدل‌سازی پیش‌بینی نرخ رهن خانه

## ۵- بحث

از آنجایی که سیستم‌های فازی می‌توانند توابع غیرخطی را با هر پیچیدگی دلخواهی مدل کنند، روش بسیار قدرتمندی هستند. در مقایسه با سایر روش‌های تقریب، قدرت اصلی آن‌ها این است که براساس تجربه افراد خبره ساخته می‌شوند. تعریف یک سیستم مبتنی بر دانش تخصصی با سیستم‌های فازی، بسیار آسان‌تر از سایر روش‌های منطقی است. قدرت سیستم‌های فازی از قابلیت آن‌ها برای بیان مفاهیم مبهم و برقراری روابطی میان آن‌ها نشأت می‌گیرد. با اینکه سیستم‌های فازی روش بسیار قدرتمندی هستند، برخی محدودیت‌ها را نیز دارند. یکی از محدودیت‌های اصلی، مشکل ابعاد است و هنگامی مطرح می‌شود که تعداد متغیرهای ورودی بیش از حد زیاد باشد. توجه داشته باشید که مشکل از تعریف اصلی سیستم‌های فازی نیست. سیستم‌های فازی با هر تعداد متغیرهای ورودی می‌توانند تعریف شوند. مشکل از آنجایی به وجود می‌آید که اگر تعداد متغیرهای ورودی بیش از حد زیاد باشد، استخراج مجموعه قواعد لازم برای طراحی سیستم فازی توسط تجربه سخت می‌شود. به هر حال، تعداد قواعد فازی در تعریف کلاسیک سیستم‌های فازی با افزایش تعداد

مدل فازی طراحی شده، 5 قاعده فازی دارد و مجموع تعداد پارامترهای بهینه‌سازی در بخش مقدم و تالی برابر با 66 است. با توجه به نتایج ارائه شده در جدول (۸)، روش پیشنهادی برای داده‌های آموزش به RMSE، 0.0614 و برای داده‌های آزمون به RMSE، 0.0828 دست یافته است. با توجه به نتایج به دست آمده مشاهده می‌شود که مدل فازی طراحی شده با روش پیشنهادی عملکرد تقریباً بهتری نسبت به مدل‌های فازی ارائه شده در مراجع [۴۹] و [۴۸] دارد و به RMSE کمتری در مقایسه با آن دست یافته است.

جدول (۸): مقایسه نتایج روش‌های مختلف برای مثال

مدل‌سازی پیش‌بینی رهن خانه

	روش	تعداد قوانین	RMSE (آموزش)	RMSE (تست)
1	[49]	5	0.0763	0.0913
2	Our Method (HODEPSO)	5	0.0614	0.0828
3	Method (ODE)	5	0.0921	0.1117
4	Method (PSO)	5	0.1100	0.1245
5	Method (DE)	5	0.1136	0.1302

متغیرهای ورودی به صورت نمایی افزایش می یابد و باعث می شود تقریباً فرآیند تعریف مجموعه قواعد فازی، مدیریت نشود.

در این مقاله از سیستم فازی سوگنو استفاده شده است. مدل های فازی سوگنو که در بردارنده قواعد اگر - آنگاه فازی است، به صورت گسترده ای برای حل مسائل مدل سازی و کنترل به کار می روند. با توجه به ساختار ساده و غیرخطی آن، مدل فازی سوگنو قادر به تقریب و پیش بینی سیستم های پیچیده با استفاده از تعداد متوسط قواعد است. مشکل اصلی در طراحی مدل های فازی، پیدا کردن یک پایگاه قواعد فازی مناسب برای یک عمل خاص است. روش آزمون و خطا با بار محاسباتی بالا و کارایی پایین به صورت گسترده ای در اکثر موارد استفاده می شود. به طور کلی طراحی سیستم فازی می تواند به دو مرحله آموزش ساختار و تنظیم پارامتر تقسیم شود. مرحله آموزش ساختار در بردارنده تعیین تعداد قواعد فازی و پارتیشن بندی فضای ورودی است. مرحله دوم، در بردارنده تعیین توابع عضویت بخش مقدم و محاسبه پارامترهای تالی است. الگوریتم های فراابتکاری تکنیک های جست و جوی سراسری برای حل مسائل بهینه سازی هستند. این الگوریتم ها با توجه به مقاوم بودن و قابلیت جست و جو، برای طراحی سیستم های فازی به کار برده شده اند.

باعث فرار الگوریتم از دام بهینه محلی و ایجاد یک جست و جوی مؤثر در فضای راه حل می شود. همچنین ضمن حفظ دقت پاسخ ها الگوریتم سریع تر به جواب بهینه همگرا می شود. در ادامه برای ارزیابی عملکرد روش پیشنهادی چهار مثال شبیه سازی ارائه شده است. در مثال اول و دوم از سیستم فازی طراحی شده به عنوان کنترل کننده فازی و در مثال سوم برای پیش بینی مقادیر آینده، سری زمانی آشوبناک مکی - گلس و در مثال چهارم برای پیش بینی نرخ رهن استفاده شده است. ابتدا برای نشان دادن کارایی الگوریتم ترکیبی نسبت به تک تک الگوریتم های تکامل تفاضلی مبتنی بر تضاد و بهینه سازی ازدحام ذرات، هر کدام از این الگوریتم ها به صورت جداگانه در مرحله دوم روش پیشنهادی (تنظیم پارامترهای سیستم فازی) به کار رفتند. نتایج، برتری الگوریتم ترکیبی را نسبت به تک تک الگوریتم های تکامل تفاضلی مبتنی بر تضاد و بهینه سازی ازدحام ذرات نشان می دهد. در نهایت، برای ارزیابی عملکرد روش پیشنهادی با استفاده از الگوریتم ترکیبی، نتایج حاصل از روش پیشنهادی با برخی کارهای انجام شده در این زمینه مقایسه شده است. نتایج، نشان دهنده عملکرد مناسب روش پیشنهادی نسبت به سایر روش ها است.

برای پیشنهادهایی جهت ادامه کار، موارد زیر بیان می شوند:

- انجام فرآیند با مدل های آنلاین و به کارگیری روش پیشنهادی در مسائل واقعی متنوع، با توجه به نتایج و کیفیت روش پیشنهاد شده در افزایش دقت سیستم فازی.
- در روش پیشنهاد شده بیشتر بر روی دقت مدل فازی تأکید شده است؛ اما در این روش، قابلیت تفسیر مدل فازی نادیده گرفته شده است. برای مثال، مفاهیم زبان شناختی مجموعه های فازی ممکن است در طی فرآیند بهینه سازی بدون محدودیت خراب شوند.
- استفاده از روش پیشنهادی مبتنی بر الگوریتم ترکیبی تکامل تفاضلی مبتنی بر تضاد و بهینه سازی ازدحام ذرات برای به دست آوردن مدل های دقیق و با داشتن قابلیت تفسیر بهتر به صورت هم زمان.

## ۶- نتیجه گیری و پیشنهادات

در این مقاله، روش دو مرحله ای مبتنی بر خوشه بندی کاهش و الگوریتم ترکیبی تکامل تفاضلی مبتنی بر تضاد و بهینه سازی ازدحام ذرات برای طراحی مدل فازی سوگنو ارائه شده است. در مرحله اول، روش پیشنهادی خوشه بندی کاهش برای پارتیشن بندی فضای ورودی و تعیین تعداد قواعد فازی به کار رفته است. در مرحله دوم، برای تنظیم پارامترهای بخش مقدم و تالی سیستم فازی از الگوریتم ترکیبی تکامل تفاضلی مبتنی بر تضاد و بهینه سازی ازدحام ذرات توسعه داده شده استفاده شده است. با استفاده از ترکیب این دو الگوریتم، تنوع جمعیتی ایجاد می شود و

## مراجع

- new encoding method”, *Fuzzy Sets and Systems* 152 617–635, 2005.
- [14] Stavrakoudis, D. G., Galidaki, G. N., Gitas, I. Z., and Theocharis, J. B. “A Genetic Fuzzy-Rule-Based Classifier for Land Cover Classification From Hyperspectral Imagery”, *IEEE Trans on Geoscience and Remote Sensing*, Vol. 50, No. 1, Jan, 130-148, 2012.
- [15] Lin, C. j. “A GA-based neural fuzzy system for temperature control”, *Fuzzy Sets and Systems* 143, 311–333, 2004.
- [16] Angelov, P.P., Buswell, R.A., “Automatic generation of fuzzy rule-based models from data by genetic algorithms”, *Information Sciences* 150, 17–31, 2003.
- [17] Storn, R. “System design by constraint adaptation and differential evolution.”, *IEEE Trans. Evol. Comput.* 3, pp. 22–34, 1999.
- [18] Ilonen, J., Kamarainen, J.K., Lampinen, J. “Differential evolution training algorithm for feed forward neural networks.” *Neural Proc. Lett.* 17, pp. 93–105, 2003.
- [19] Goldberg, E., Richardson, J., “Genetic algorithms with sharing for multimodal function optimization.” in: J. Richardson (Ed.), *Genetic Algorithms and their Applications (ICGA’87)*, pp. 41–49, 1987.
- [20] Kristinsson, K.M., Dumont, G.A., “System identification and control using genetic algorithms.” *IEEE Trans. Syst. Man Cybernet.* 22, pp. 1033–1046, 1992.
- [21] Kennedy, J., Eberhart, R. and Shi, Y. “Swarm Intelligence”, San Mateo, CA: Morgan Kaufmann, 2001.
- [22] Zhao, L., Yang, Y., Zeng, Y. “Eliciting compact T–S fuzzy models using subtractive clustering and coevolutionary particle swarm optimization”, *Neurocomputing* 72, 2569–2575, 2009.
- [23] Li, C., Wu, T. “Adaptive fuzzy approach to function approximation with PSO and RLSE”, *Expert Systems with Applications* 38, 13266–13273, 2011.
- [24] Chen, D., Zhao, C. “Data-driven fuzzy clustering based on maximum entropy principle and PSO”, *Expert Systems with Applications* 36, 625–633, 2009.
- [25] Chen, D., Wang, J., Zou, F., Zhang, H., Hou, W. “Linguistic fuzzy model identification based on PSO with different length of particles”, *Applied Soft Computing* 12, 3390–3400, 2012.
- [26] Melin, P., Olivas, F., Castillo, O., Valdez, F., Soria, J., Valdez, M. “Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic”, *Expert Systems with Applications* 40, 3196–3206, 2013.
- [27] Mucientes, M., Casillas, J. “Quick design of fuzzy controllers with good interpretability in mobile robotics,” *IEEE Trans. Fuzzy Syst.*, Vol. 15, No. 4, pp. 636–651, Aug. 2007.
- [28] Juang, C. F., Lu, C. M., Lo, C., Wang, C. Y. “Ant colony optimization algorithm for fuzzy controller design and its FPGA implementation”, *IEEE*
- [1] Hashemzadeh, E., Hamidi, H. “Using a Data Mining Tool and FP-growth Algorithm Application for Extraction of the Rules in Two Different Dataset.” *International Journal of Engineering (IJE)*, *TRANSACTIONS C: Aspects* Vol. 29, No. 6, June 2016.
- [2] Hamidi, H., Daraei, A. “Analysis of Pre-processing and Post-processing Methods and Using Data Mining to Diagnose Heart Diseases.” *International Journal of Engineering (IJE)*, *TRANSACTIONS B: Applications* Vol. 29, No. 8, August 2016.
- [3] Abadpour, M., Hamidi, H. “Stabilization of V94.2 Gas Turbine Using Intelligent Fuzzy Controller Optimized by the Genetic Algorithm.” *International Journal of Applied and Computational Mathematics*, pp 1–14, 15 November 2016. DOI: 10.1007/s40819-016-0276-2
- [4] Zadeh, L. A. “Fuzzy sets”, *Inform. And Contr.* Vol. 8, pp. 338-353, 1965.
- [5] Mamdani, E. H. and Assilian, S. “An experiment in linguistic synthesis with a fuzzy logic controller”, *Int. J. Man-Machine Studies*, Vol. 7, No. 1, pp. 1-13, 1975.
- [6] Takagi T. and Sugeno, M. “Fuzzy identification of systems and its applications to modeling and control”, *IEEE Trans. Systems, Man, and Cybernetics*, vol. 15, pp. 116-132, 1985.
- [7] Procyk, T.J., Mamdani, E.M. “A linguistic self-organizing process controller”, *Automatica* 15 (1), 15–30, 1979.
- [8] Yao, X., Ed., “Evolutionary Computation-Theory and Applications”, Singapore: World Scientific, 1999.
- [9] Fazzolari, M., Alcalá, R., Nojima, Y., Ishibuchi, H. and Herrera, F. “A Review of the Application of Multiobjective Evolutionary Fuzzy Systems: Current Status and Further Directions”, *IEEE Trans on Fuzzy Syst.*, Vol. 21, No. 1, Feb, 45-65, 2013.
- [10] Cordon, O., Gomide, F., Herrera, F., Hoffmann, F., Magdalena, L., “Ten years of genetic fuzzy systems: current framework and new trends”, *Fuzzy Sets and Systems* 141 (1), 5–31, 2004.
- [11] Papadakis, S. E., Theocharis, J. B. “A GA-based fuzzy modeling approach for generating TSK models”, *Fuzzy Sets and Systems*, 131, 121-152, 2002.
- [12] Giordano, V., Naso, D. Turchiano, B. “Combining genetic algorithms and Lyapunov-based adaptation for online design of fuzzy controllers”, *IEEE Trans. Systems Man Cybernet., Part B: Cybernet.* 36 (5) 1118–1127, 2006.
- [13] Chiou, Y. C., Lan, L. W. “Genetic fuzzy logic controller: an iterative evolution algorithm with

- Swarm Optimization”, IEEE Trans. Fuzzy Syst., vol. 18, no. 6, pp. 1083-1097, December. 2010.
- [41] Juang, C., Hsiao, C. and Hsu, C. “Hierarchical Cluster-Based Multispecies Particle-Swarm Optimization for Fuzzy-System Optimization”, IEEE Trans. Fuzzy Syst., Vol. 18, No. 1, pp. 14-26, February. 2010.
- [42] Huang, M., Lin, H., Yunkai, H., Jin, P., and Guo, Y., “Fuzzy Control for Flux Weakening of Hybrid Exciting Synchronous Motor Based on Particle Swarm Optimization Algorithm”, IEEE Trans. Magnetics, Vol. 48, No. 11, pp. 2989-2992, November. 2012.
- [43] Tana, C.H., Yap, K.S., Yap, H.J., “Application of genetic algorithm for fuzzy rules optimization on semi expert judgment automation using Pittsburg approach”, Elsevier. Applied Soft Computing, 12, pp. 2168–2177, 2012.
- [44] Muthukaruppan, S., Er, M.J. “A hybrid particle swarm optimization based fuzzy expert system for the diagnosis of coronary artery disease”, Elsevier. Expert Systems with Applications, 39, pp. 11657–11665, 2012.
- [45] Xiong, N. “Fuzzy rule-based similarity model enables learning from small case bases”, Elsevier. Applied Soft Computing, 13, pp. 2057–2064, 2013.
- [46] Cheung, N. J., Ding X. , and Shen, H., “OptiFel: A Convergent Heterogeneous Particle Swarm Optimization Algorithm for Takagi–Sugeno Fuzzy Modeling”, IEEE Trans. Fuzzy Syst, Vol. 22, No. 4, pp. 919-933, August. 2014.
- [47] García-Galán, S., Prado, R. P., and Muñoz Expósito, J. E. “Swarm Fuzzy Systems: Knowledge Acquisition in Fuzzy Systems and Its Applications in Grid Computing”, IEEE Trans. Knowledge and Data Engineering, Vol. 26, No. 7, pp. 1791-1804, July. 2014.
- [48] Juang, C. F., Hung, C. W., Hsu, C. H. “Rule-Based Cooperative Continuous Ant Colony Optimization to Improve the Accuracy of Fuzzy System Design”, IEEE Trans. Fuzzy Systems., Vol. 22, No. 4, pp. 723–735, Agu. 2014.
- [49] Zhao, W., Niu, Q., Li, K. and Irwin, G. W. “A Hybrid Learning Method for Constructing Compact Rule-Based Fuzzy Models”, IEEE Trans. Cybernetics, Vol. 43, No. 6, pp. 1807-1821, December. 2013.
- [50] Hamidi, H. “A Combined Fuzzy Method for Evaluating Criteria in Enterprise Resource Planning Implementation.” International Journal of Intelligent Information Technologies (IJIT), 12(2), pp. 25-52, 2016.
- [51] Vafaei, A, Hamidi, H. and Monadjemi, S.A. “Evaluation and Check pointing of Fault Tolerant Mobile Agents Execution in Distributed Systems,” Journal of Networks, VOL. 5, NO. 7. 2010.
- [52] Vafaei, A, Hamidi, H. and Monadjemi, S.A. “A Framework for Fault Tolerance Techniques in the Analysis and Evaluation of Computing Systems” International Journal of Innovative Computing, Trans. Ind. Electron., vol. 55, no. 3, pp. 1453–1462, Mar. 2008.
- [29] Juang, C. F., Kang, Y. P., Lo, C., “Fuzzy Controller Design by Clustering-Aided Ant Colony Optimization”, Proceedings of the 17th World Congress the International Federation of Automatic Control Seoul, Korea, 12297-12302, July 6-11, 2008.
- [30] Chang, Y. H., Chang, C. W. , Tao, C. W., Lin, H. W., Taur, J. S. “Fuzzy sliding-mode control for ball and beam system with fuzzy ant colony optimization”, Expert Systems with Applications 39, 3624–3633, 2012.
- [31] Juang, C. F., and Chang, P. H. “Designing Fuzzy-Rule-Based Systems Using Continuous Ant-Colony Optimization”, IEEE Trans on Fuzzy Syst., Vol. 18, No. 1, 138-149, Feb 2010.
- [32] Juang, C. F., Lo, C., “Zero-order TSK-type fuzzy system learning using a two-phase swarm intelligence algorithm”, Fuzzy Sets and Systems 159, 2910 – 2926, 2008.
- [33] Lee, C. H., Kuo, C. T., Chang, H. H., Chien, J. C., and Chang, F. K. “A Hybrid Algorithm of Electromagnetism-like and Genetic for Recurrent Neural Fuzzy Controller Design”, Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS 2009, Hong Kong, March 18 - 20, 2009.
- [34] Juang, C. F. “A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design”, IEEE Trans on Syst., man, and Cybernetics-PART B: Cybernetics, Vol. 34, No 2, 997-1006, APRIL 2004.
- [35] Kannan, S., Slochanal, S.M.R., Subbaraj, p., and Padhy, N.PP, “Application of Particle Swarm Optimization Technique and its Variants to Generation Expansion Planning”, Electric Power Systems Research, 70(30): 203-210, 2004.
- [36] Zhang, W-J., Xie, X-F., “DEPSO: Hybrid Particle Swarm with Differential Evolution Operator”, In: Proceedings of the IEEE International Conference on System, man, and Cybernetics, vol. 4, pp 3816-3821, 2003.
- [37] Lin, C., Chen, C., and Lin, C., “A Hybrid of Cooperative Particle Swarm Optimization and Cultural Algorithm for Neural Fuzzy Networks and Its Prediction Applications”, IEEE Trans. Systems Man Cybernet. Part C: Applications and Reviews. Vol. 39, No. 1, pp. 55-68, Jan. 2009.
- [38] Faruque Ali, Sk., Ramaswamy, A., “Optimal fuzzy logic control for MDOF structural systems using evolutionary algorithms”, Elsevier. Engineering Applications of Artificial Intelligence, 22, pp. 407–419, 2009.
- [39] Lughofer, E. and Kindermann, S., “SparseFIS: Data-Driven Learning of Fuzzy Systems With Sparsity Constraints”, IEEE Trans. Fuzzy Syst., Vol. 18, No. 2, pp. 396-411, April. 2010.
- [40] Prado, R. P., García-Galán, S., Muñoz Expósito, J. E., and Yuste, A. J., “Knowledge Acquisition in Fuzzy-Rule-Based Systems With Particle-

Information Systems” Journal of Information Systems and Telecommunication, Vol. 4, No. 1, January-March 2016, pp. 19-26.

Information and Control (IJICIC), Vol.8, No.7, July 2012

[53] Hamidi, H. “Selecting Enterprise Resource Planning System Using Fuzzy Analytic Hierarchy Process Method.” Journal of Information Systems and Telecommunication, Vol. 3, No. 4, October-December 2015, pp. 205-215.

[54] Hamidi, H. “A New Method for Transformation Techniques in Secure

- 
- <sup>1</sup> Genetic Algorithm (GA)  
<sup>2</sup> Differential Evolution (DE)  
<sup>3</sup> Particle Swarm Optimization (PSO)  
<sup>4</sup> Swarm Intelligence (SI)  
<sup>5</sup> Ant Colony Optimization (ACO)  
<sup>6</sup> Hybridization  
<sup>7</sup> Cleveland  
<sup>8</sup> HODEPSO  
<sup>9</sup> Crisp  
<sup>10</sup> Chaotic Mackey-Glass Time Series  
<sup>11</sup> Mortgage  
<sup>12</sup> Sum of Absolute Error  
<sup>13</sup> Root Mean Squares Error