Downloaded from jor.iranaict.ir at 18:12 +0330 on Wednesday February 11th 2026

فصلنامه علمی- پژوهشی
فناوری اطلاعات و ارتباطات ایران

سال دهم، شماره‌های ۳۵ و ۳۶، بهار و تابستان ۱۳۹۷
صص: ۱- ۱۴

الگوریتم رقابت استعماری آشوبی متعامد اصلاح شده و بکارگیری آن در بهبود بازشناسی الگو در شبکهٔ عصبی پرسپترون‌های چند لایه

پیمان معلم* مهرداد صادقی حریری** مهدی هاشمی***
* استاد، دانشگاه اصفهان، گروه مهندسی برق
** دانشجوی دکتری تخصصی مهندسی برق کنترل، دانشگاه آزاد اسلامی واحد علوم و تحقیقات تهران
*** مربی، موسسه آموزش عالی پیام گلپایگان، دانشکده مهندسی برق

تاریخ دریافت: ۱۳۹۶/۰۶/۱۹ تاریخ پذیرش: ۱۳۹۶/۱۲/۲۰

چکیده

علی‌رغم موفقیت الگوریتم رقابت استعماری (ICA) در حل مسائل بهینه‌سازی، این الگوریتم کماکان از به دام افتادن مکرر در کمینه محلی و سرعت پایین همگرایی رنج می‌برد. در این مقاله، نسخهٔ جدیدی از این الگوریتم، به نام رقابت استعماری آشوبی متعامد اصلاح شده (COICA)، پیشنهاد می‌شود. در سیاست جذب نسخه پیشنهادی، هرمستعمره از طریق تعریف بردار متعامد نوینی، فضای حرکت به سمت استعمارگر را جستجو می‌کند. همچنین احتمال انتخاب امپراطوری‌های قدرتمند، از طریق تابع توزیع بولتزمان تعریف شده و عمل انتخاب از طریق روش چرخ رولت انجام گرفته است. از الگوریتم پیشنهادی برای آموزش شبکه عصبی پرسپترون چند لایه[1] (MLP) جهت طبقه‌بندی مجموعه داده‌های استاندارد، از جمله یونسفر و سونار استفاده شده است. برای ارزیابی عملکرد این الگوریتم و بررسی میزان تعمیم‌پذیری شبکه عصبی آموزش دیده با نسخه پیشنهادی، از روش اعتبارسنجی متقابل K-Fold استفاده شده است. نتایج بدست آمده از شبیه‌سازی‌ها، کاهش خطای آموزش شبکه و همچنین بهبود تعمیم‌پذیری الگوریتم پیشنهادی را تایید می‌کند.

واژه‌های کلیدی: الگوریتم رقابت استعماری آشوبی متعامد، شبکه عصبی پرسپترون چند لایه، طبقه‌بندی داده

۱- مقدمه

روش‌های اکتشافی و فرا اکتشافی در حل مسائل بهینه‌سازی در علوم مختلف به کار می‌روند. این روش‌ها سعی در کشف یک راه حل بدون داشتن درک عمیقی از ساختار مساله دارند. به عبارت دیگر، مساله برای الگوریتم به عنوان یک جعبه سیاه به نظر می‌رسد. این روش‌ها معمولا بصورت تصادفی با استفاده از آمار بدست آمده از نمونه‌هایی از فضای جستجو عمل می‌کنند و یا بر اساس مدلی از برخی پدیده‌های

نویسنده عهده‌دار مکاتبات: پیمان معلم p_moallem@eng.ui.ac.ir

[1] Multilayer Perceptron

Downloaded from jor.iranaict.ir at 18:12 +0330 on Wednesday February 11th 2026

طبیعی یا فرآیندهای فیزیکی هستند. این الگوریتم‌ها شامل الگوریتم ژنتیک[۲] (GA) [۱]، شبیه‌سازی تبرید[۳] (SA) [۲]، [۲]، بهینه‌سازی ازدحام ذرات[۴] (PSO) [۳]، بهینه‌سازی جمعیت مورچگان[۵] (ACO) [۴]، الگوریتم تکاملی فرهنگی[۶] (CE) [۵]، تکامل تفاضلی[۷] (DE) [۶]، جستجوی هارمونی[۸] (HS) [۷]، الگوریتم جستجوی گرانشی[۹] (GSA) [۸]، الگوریتم تپه‌نوردی[۱۰] [۹]، الگوریتم سیستم جستجوی ذرات باردار[۱۱] (CSS) [۱۰] و غیره می‌باشد.

الگوریتم رقابت استعماری[۱۲] (ICA) [۱۱] از روش‌های فرااکتشافی بهینه‌سازی است که مبتنی بر پدیده‌های انسانی- اجتماعی عمل کرده و نشان داده شده که در مقایسه با الگوریتم‌های مبتنی بر تکامل پدیده‌های طبیعی، به سرعت بالاتری از همگرایی با متغیرهای مستقل زیاد دست یافته است [۱۲].

الگوریتم رقابت استعماری، یک الگوریتم چند عاملی است که هر عامل، یک کشور به عنوان مستعمره یا استعمارگر می‌باشد. این الگوریتم به فرآیند شکل گیری امپریالیسم، رشد و زوال آن، به عنوان مرحله‌ای از تکامل اجتماعی- سیاسی انسان نگاه می‌کند. کشورها تعدادی امپراطوری، در فضای جستجو را شکل می‌دهند. حرکت مستعمرات به سمت استعمارگر مربوطه و رقابت استعماری بین امپراطوری‌ها اساس و پایهٔ این الگوریتم را تشکیل می‌دهد. در طول این جابجایی‌ها، امپراطوری‌های قدرتمند تقویت یافته و امپراطوری‌های ضعیف تضعیف می‌شوند و به تدریج سقوط می‌کنند. در الگوریتم رقابت استعماری، مستعمرات با شعاع حرکتی تصادفی به سمت استعمارگر حرکت می‌کنند.

در سال‌های اخیر در راستای بهبود عملکرد الگوریتم رقابت استعماری، در حل مسائل بهینه‌سازی، تحقیقات مختلفی صورت گرفته است. کاوه و طلاطاهری [۱۳] در سال ۲۰۱۰ الگوریتم ICA را با افزودن یک بردار متعامد تصادفی بین استعمارگر و مستعمره در مرحلهٔ جذب، بهبود بخشیدند. این بردار بر راستای خط واصل مستعمره- استعمارگر عمود می‌باشد. بنابراین این الگوریتم، رقابت استعماری متعامد[۱۳] (OICA) نام گرفت. در این مقاله برای بهبود الگوریتم پایه، دو گام حرکتی تعریف شده است که عبارتند از: (۱) بکارگیری از مقادیر تصادفی مختلف برای مولفه‌های بردار پاسخ به جای استفاده از یک مقدار؛ (۲) ایجاد انحراف با استفاده از بردار متعامد بر خط واصل مستعمره- استعمارگر به جای استفاده از متغیر θ. در ادامه، در سال ۲۰۱۰ الگوریتم رقابت استعماری آشوبی[۱۴] (CICA) توسط بهرامی و همکارانش پیشنهاد شده است [۱۴] که موفق به بهبود عملکرد الگوریتم ICA شده‌اند. در این روش از نگاشت‌های آشوبی برای بروز رسانی زاویهٔ حرکتی مستعمرات به سمت موقعیت استعمارگر در راستای افزایش قابلیت گریز از دام اپتیمم محلی استفاده شده است. همچنین الگوریتم رقابت استعماری برای یادگیری شبکهٔ عصبی به کار برده شده است [۱۵].

در سال ۲۰۱۰ طی مقاله‌ای عبدچیری و همکارانش [۱۶] الگوریتم رقابت استعماری تطبیقی[۱۵] (AICA) را مطرح کردند. در این الگوریتم پیشنهادی، به منظور جستجوی موثر، سیاست جذب الگوریتم جهت بروز رسانی زاویهٔ حرکتی مستعمره به سمت موقعیت استعارگر به صورت دینامیکی تغییر یافته است. در این روش، مدلی احتمالی به کار رفته است که با بکارگیری اطلاعات موقعیت مستعمرات، اقدام به برقراری تعادل بین توانایی جستجو و بهره‌برداری

[2] Genetic Algorithm
[3] Simulated Annealing
[4] Particle Swarm Optimization
[5] Ant Colony Optimization
[6] Cultural Evolutionary Algorithm
[7] Differential Evolution
[8] Harmony Search
[9] Gravitational Search Algorithm
[10] Hill Climbing
[11] Charged System Search
[12] Imperialist Competitive Algorithm
[13] Orthogonal Imperialist Competitive Algorithm
[14] Chaotic Imperialist Competitive Algorithm
[15] Adaptive ICA

Downloaded from jor.iranaict.ir at 18:12 +0330 on Wednesday February 11th 2026

الگوریتم رقابت استعماری کرده است. در سال ۲۰۱۲ کوئلهو، آفونسو و آلوتو [۱۷]، الگوریتم جدیدی مبتنی بر الگوریتم ICA معرفی کرده‌اند. در این مقاله روش اصلاح شدهٔ ICA (MICA[۱۶]) مبتنی بر مفاهیم و اصول جاذبه و دافعه بین مستعمره و استعمارگر آن در طول جستجو برای پاسخ‌های بهتر ارائه شده است.

سلطانی سروستانی، لطفی و رمضانی در سال ۲۰۱۲ [۱۸]، الگوریتم تکاملی بهبود یافته‌ای مبتنی بر الگوریتم رقابت استعماری معرفی کرده‌اند. در الگوریتم ICA، کشورها به دو گروه استعمارگرها و مستعمرات تقسیم شده است. در حالی که در QCA[۱۷] دو نوع دیگر از کشورها که عبارتند از کشورهای مستقل و کشورهای به دنبال استقلال نیز به مجموعه کشورها افزوده شده است. در الگوریتم ICA موقعیت استعمارگرها ثابت است، در حالی که در الگوریتم QCA قادر به حرکت هستند. طلاطاهری و همکارانش در سال ۲۰۱۲ [۱۲] طی مقاله‌ای سرعت همگرایی الگوریتم ICA را با جایگزینی توابع آشوبی به جای توابع تصادفی در گام جذب الگوریتم، بهبود دادند. در این مقاله سه نسخه از الگوریتم CICA ارائه شده است، که در نهایت نسخهٔ سوم از الگوریتم‌های پیشنهادی را که می‌توان به عنوان رقابت استعماری آشوبی متعامد دانست، عملکرد بهتری نسبت به سایر الگوریتم‌ها از خود نشان داده است. در واقع این نسخه ترکیبی از روش متعامد و روش آشوبی می‌باشد. در الگوریتم پیشنهاد شده از چندین نگاشت آشوبی به کار رفته، توابع آشوبی سینوسی و منطقی نتایج بهتری نسبت به بقیه از خود نشان داده‌اند.

از مسائل کاربردی که می‌توان الگوریتم‌های بهینه‌سازی را به چالش کشید، آموزش شبکه‌های عصبی پرسپترون چند لایه است که علاوه بر نحوه رسیدن به پاسخ مطلوب، سرعت همگرایی و نرخ موفقیت اجرا، پارامتر عمومیت پذیری شبکه آموزش دیده در برخورد با داده‌های غیر آموزشی نیز به چالش کشیده می‌شود. از اینرو، بسیاری از الگوریتم‌های فرااکتشافی مدعی، عملکرد خود را در مقایسه با سایرین در محک آموزش شبکه‌های عصبی پرسپترون چند لایه قرار می‌دهند.

در بخش دوم این مقاله، ابتدا به جزییات الگوریتم رقابت استعماری پرداخته می‌شود. سپس در بخش بعد، اصلاحات پیشنهادی برای حل مشکلات این الگوریتم، مطرح شده و بلوک دیاگرام الگوریتم رقابت استعماری پیشنهادی مطرح می‌شود. بخش چهارم، به ارزیابی روش پیشنهادی در آموزش شبکه عصبی پرسپترون چند لایه برای کلاس بندی داده‌های یونسفر و سونار که از داده‌های پیچیده در کلاس بندی است، و مقایسه آن با سایر روش‌های مشابه می‌پردازد. در نهایت مقاله با ارائه نتایج در بخش پنجم به اتمام می‌رسد.

۲- الگوریتم رقابت استعماری

الگوریتم رقابت استعماری پایه، شامل ۶ مرحله است که در ادامه، تشریح می‌شود [۱۱].

۲-۱- تولید امپراطوری‌های اولیه

این الگوریتم مانند سایر الگوریتم‌های تکاملی از یک سری جمعیت اولیه تشکیل شده که کشور نامیده می‌شوند. در یک مساله N_{var} بعدی، یک کشور یک آرایه به طول $1 \times N_{\mathrm{var}}$ است. این آرایه به صورت رابطهٔ (۱) تعریف می‌شود. در واقع هر درایه نقش یک مشخصه از کشور مانند فرهنگ، زبان، ساختار اقتصادی و غیره را بازی می‌کند.

$$\mathrm{country} = [p_1, p_2, \ldots, p_{N\mathrm{var}}] \quad (۱)$$

هزینهٔ هر کشور از طریق ارزیابی تابع هزینهٔ f در متغیرهای $(P_1, P_2, P_3, \ldots, P_{N_{var}})$ بدست می‌آید که در رابطهٔ (۲) نشان داده شده است.

$$Cost = f(country) = f(P_1, P_2, P_3, \ldots, P_{N_{var}}) \quad (۲)$$

برای شروع الگوریتم بهینه‌سازی جمعیت اولیه به تعداد N_{pop} تولید می‌شود. به تعداد N_{imp} از قدرتمندترین کشورها به عنوان استعمارگر انتخاب می‌شوند که هر کدام یک امپراطوری را تشکیل می‌دهند. تعداد N_{col} جمعیت باقیمانده به عنوان مستعمره خواهد بود که بر حسب قدرت استعمارگرها بین امپراطوری‌ها توزیع می‌شوند. بنابراین کشورها به دو نوع مستعمره و استعمارگر تقسیم می‌شوند.

[16] Modified ICA
[17] Quad Countries Algorithm

Downloaded from jor.iranaict.ir at 18:12 +0330 on Wednesday February 11th 2026

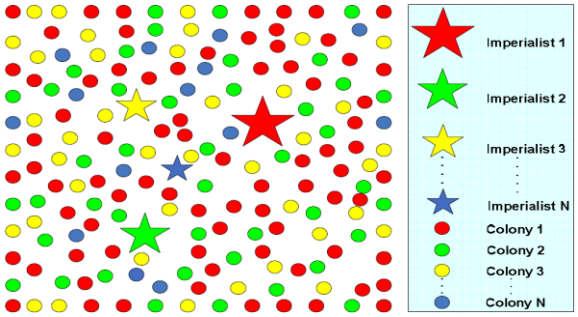
به منظور شکل‌گیری امپراطوری‌های اولیه، مستعمرات بین استعمارگرها بر اساس قدرتشان تقسیم می‌شوند. یعنی تعداد اولیهٔ مستعمرات هر امپراطوری به طور مستقیم با قدرت آن متناسب است. برای توزیع تناسبی مستعمرات در میان استعمارگرها، هزینهٔ نرمالیزه شدهٔ هر استعمارگر به صورت رابطه (۳) تعریف می‌شود:

$$C_n = c_n - \max_i\{c_i\} \quad (۳)$$

متغیر c_n هزینهٔ استعمارگر n ام و C_n هزینهٔ نرمالیزه شدهٔ آن است. با داشتن همهٔ هزینه‌های نرمالیزه شدهٔ کل استعمارگرها، قدرت نرمالیزه شدهٔ هر استعمارگر به صورت رابطه (۴) تعریف می‌شود:

$$p_n = \left| \frac{c_n}{\sum_{i=1}^{N_{imp}} c_i} \right| \quad (۴)$$

متغیر $N.C._n$ تعداد اولیهٔ مستعمرات امپراطوری nام و N_{col} تعداد کل مستعمرات را نشان می‌دهد. برای تقسیم مستعمرات، برای هر استعمارگر $N.C._n$ مستعمره به تصادف انتخاب شده و به آنها داده می‌شود. این مستعمرات به همراه استعمارگر، امپراطوری n ام را تشکیل خواهد داد. در شکل (۱) جمعیت اولیه هر امپراطوری نشان داده شده است. هر ستاره بیانگر یک استعمارگر است که ستاره‌های بزرگتر قدرت بیشتری را دارا می‌باشند. در شکل زیر Imperialist نشان دهندهٔ استعمارگر و Colony بیانگر مستعمرات است.



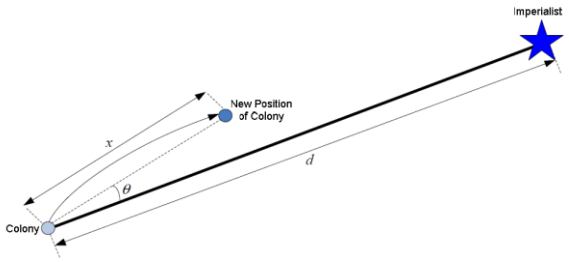
شکل ۱- تولید امپراطوری‌های اولیه؛ استعمارگر قدرتمند با ستارهٔ بزرگتر نشان داده شده است [۱۱]

۲-۲- حرکت مستعمرات یک امپراطوری به سمت استعمارگر

در این الگوریتم کشورهای استعمارگر شروع به توسعهٔ مستعمرات خود می‌کنند. این فرآیند با حرکت تمام مستعمرات به سمت استعمارگر مدل شده است. در راستای این سیاست، کشور مستعمره، به اندازه x واحد در جهت خط واصل مستعمره به استعمارگر، حرکت کرده و به موقعیت جدید، کشانده می‌شود. x عددی تصادفی با توزیع یکنواخت است (و یا هر توزیع مناسب دیگر) که طبق رابطهٔ (۵) می‌باشد. اگر فاصله میان استعمارگر و مستعمره با d نشان داده شود، معمولاً برای d داریم.

$$x \sim U(0, \beta \times d) \quad (۵)$$

که در آن β عددی بزرگتر از یک و نزدیک به ۲ می‌باشد. یک انتخاب مناسب می‌تواند $\beta = 2$ باشد. وجود ضریب $\beta \geq 1$ باعث می‌شود تا کشور مستعمره در حین حرکت به سمت کشور استعمارگر، از جهت‌های مختلف به آن نزدیک شود. همچنین در کنار این حرکت، یک انحراف زاویه‌ای کوچک نیز با توزیع یکنواخت به مسیر حرکت افزوده می‌شود. یک نمای گرافیکی از اعمال سیاست جذب در الگوریتم رقابت استعماری در صفحه دو بعدی در شکل (۲) نشان داده شده است.



شکل ۲- حرکت مستعمرات به سمت استعمارگر مربوطه با زاویه انحراف تصادفی [۱۱]

که مقدار زاویهٔ θ توسط رابطهٔ (۶) تعریف می‌شود:

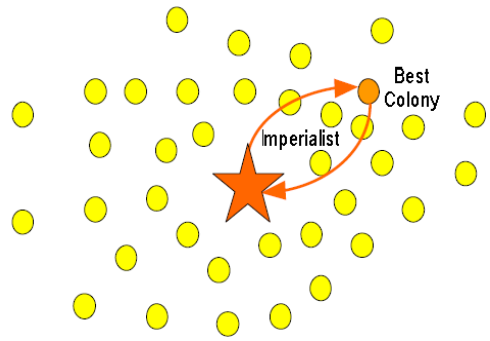
$$\theta \sim U(-\gamma, \gamma) \quad (۶)$$

پارامتر γ در رابطهٔ بالا جهت تنظیم انحراف از مسیر اصلی اعمال می‌شود. در شکل بالا θ یک عدد تصادفی با توزیع یکنواخت است.

۲-۳-جابجایی موقعیت بین استعمارگر و یک مستعمره

اگر مستعمره‌ای در حین حرکت به سمت استعمارگر، مقدار هزینهٔ کمتری نسبت به استعمارگر آن امپراطوری بدست آورد، جای مستعمره و استعمارگر با یکدیگر تعویض خواهد شد. شکل (۳) این فرآیند را به تصویر کشیده است.

Downloaded from jor.iranaict.ir at 18:12 +0330 on Wednesday February 11th 2026



شکل ۳- تعویض موقعیت مستعمره‌ای با وضعیتی بهتر با استعمارگر [۱۱]

۲-۴- قدرت کل یک امپراطوری

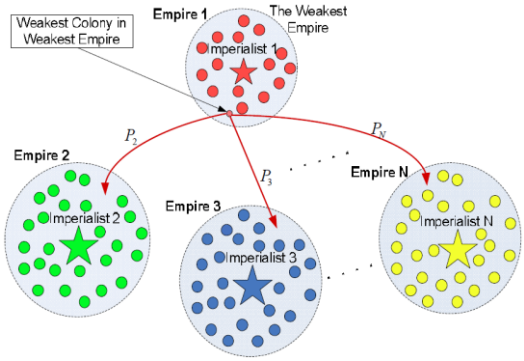
قدرت کل یک امپراطوری اساسا تحت تاثیر قدرت کشور استعمارگر است. البته قدرت مستعمرات نیز اثر خواهد گذاشت، هر چند که در مقابل قدرت کل امپراطوری قابل اغماض است. در نهایت هزینهٔ کل بصورت رابطه (۷) مدل شده است:

$$T.C._n = Cost(imperialist_n) + \xi mean\{Cost(colonies\ of\ empire_n)\} \quad (۷)$$

بطوریکه $T.C._n$ نشان‌دهنده هزینهٔ کل امپراطوری n ام و ξ عدد مثبتی است که کمتر از یک در نظر گرفته می‌شود. افزایش مقدار ξ نقش مستعمرات در تعیین قدرت کل امپراطوری را بیشتر می‌کند.

۲-۵- رقابت استعماری

همهٔ امپراطوری‌ها سعی در تصاحب مستعمرات سایر امپراطوری‌ها دارند. در این رقابت استعماری چندین امپراطوری قدرتمند برای تصاحب مستعمرات ضعیف امپراطوری ضعیف با هم رقابت می‌کنند. طی این فرآیند امپراطوری‌های قدرتمند، قوی‌تر و امپراطوری‌های ضعیف، ضعیف‌تر می‌شوند. شکل (۴) این رقابت را به تصویر می‌کشد.



شکل ۴- رقابت استعماری [۱۱]

برای شروع رقابت، احتمال مالکیت هر امپراطوری بر اساس قدرت کل آن تعیین می‌شود. هزینهٔ کل نرمالیزه شده بصورت رابطهٔ (۸) بیان می‌شود:

$$N.T.C._n = T.C._n - \max_i\{T.C._i\} \quad (۸)$$

که متغیرهای $T.C._n$ و $N.T.C._n$ به ترتیب نشان‌دهنده هزینهٔ کل و هزینهٔ کل نرمالیزه شدهٔ امپراطوری n ام است. با استفاده از هزینهٔ کل نرمالیزه شده، احتمال مالکیت هر امپراطوری از رابطهٔ (۹) بدست می‌آید:

$$P_{p_n} = \left|\frac{N.T.C._n}{\sum_{i=1}^{N_{imp}} N.T.C._i}\right| \quad (۹)$$

برای توزیع مستعمرات ذکر شده در میان استعمارگرها بر اساس احتمال هر یک، بردار P بصورت رابطهٔ (۱۰) تشکیل می‌یابد:

$$P = [p_{p_1}, p_{p_2}, p_{p_3}, \dots, p_{p_{N_{imp}}}] \quad (۱۰)$$

سپس برداری با ابعاد برابر با بردار P تولید خواهیم کرد که مولفه‌های آن اعداد تصادفی با توزیع یکنواخت باشد که در رابطه (۱۱) نشان داده شده است:

$$R = [r_1, r_2, r_3, \dots, r_{N_{imp}}]$$
$$r_1, r_2, r_3, \dots, r_{N_{imp}} \sim U(0,1) \quad (۱۱)$$

بردار D از طریق کسر بردار R از بردار P بصورت رابطه (۱۲) بدست می‌آید:

$$D = P - R = \left[D_1, D_2, D_3, \dots, D_{N_{imp}}\right] = [p_{p_1} - r_1, p_{p_2} - r_2, p_{p_3} - r_3, \dots, p_{p_{N_{imp}}} - r_{N_{imp}}] \quad (۱۲)$$

با رجوع به بردار D مستعمرات مذکور به امپراطوری‌ای که اندیس مربوطهٔ آن در بردار D بیشینه است، تحویل داده می‌شود.

۲-۶- سقوط امپراطوری‌های ضعیف

زمانی که یک امپراطوری تمام مستعمرات خود را به سایر امپراطوری‌ها واگذار کند و هیچ مستعره‌ای نداشته باشد، سقوط می‌کند که این عمل با حذف آن امپراطوری مدل می‌شود. البته در نهایت استعمارگر آن امپراطوری نیز به عنوان یک مستعمره به امپراطوری قدرتمندی ملحق می‌شود.

Downloaded from jor.iranaict.ir at 18:12 +0330 on Wednesday February 11th 2026

۳- اصلاحات پیشنهادی در الگوریتم رقابت استعماری

علی رغم تحقیقات موفقی که در زمینهٔ توسعه و بهبود الگوریتم رقابت استعماری انجام شده است، همچنان برخی از مسائل چالش برانگیز در آن وجود دارد، که از آن جمله می‌توان به (۱) افتادن در دام بهینه محلی، (۲) سرعت کم همگرایی در برخی از مسائل، (۳) پیچیدگی محاسباتی بسیار بالا برای توابعی با متغیرهای مستقل زیاد، (۴) عدم تضمین رسیدن به پاسخ مطلوب اشاره کرد که در این مقاله سه پیشنهاد برای بهبود عملکرد الگوریتم رقابت استعماری پیشنهاد شده است.

برای مقابله با چالش‌های بالا، الگوریتم رقابت استعماری آشوبی متعامد اصلاح شده پیشنهاد شده است. در این الگوریتم در دو گام بسیار حیاتی الگوریتم ICA یعنی سیاست جذب و رقابت استعماری، تغییراتی اعمال شده است. در ابتدا از یک تابع توزیع احتمال جدیدی برای امپراطوری‌ها استفاده خواهد شد. سپس از روش نمونه برداری چرخ رولت برای عملیات انتخاب استفاده کرده‌ایم. همچنین روش جدیدی جایگزین روش بردار متعامد پیشنهاد خواهیم کرد. در نهایت روش اعتبارسنجی متقابل $K-Fold$ را برای بررسی تعمیم پذیری الگوریتم پیشنهادی به کار برده‌ایم. در ادامه به تغییرات اعمال شده به تفصیل خواهیم پرداخت.

۳-۱- تعریف تابع توزیع بولتزمان

یکی از اساسی‌ترین ایراداتی که می‌توان بر الگوریتم ICA پایه گرفت، نوع نمونه‌برداری در انتخاب امپراطوری قدرتمند برای تصاحب ضعیف‌ترین مستعمره از ضعیف‌ترین امپراطوری است. اگرچه شاید روش به کار رفته یک روش کاملاً سرراست و ساده‌ای باشد، اما از نظر مفاهیم یکنواخت نیست، یعنی الزاما کسی که قوی‌ترین است انتخاب نمی‌شود و همچنین رفتار این روش شدیداً غیرخطی است.

در گام رقابت استعماری الگوریتم ICA، بایستی یکی از امپراطوری‌های قدرتمند جهت جذب مستعمرهٔ ضعیف در ضعیف‌ترین امپراطوری، انتخاب شود. بنابراین برای انجام چنین کاری، قدرت هر امپراطوری توسط تابع توزیع احتمال بولتزمان تعریف می‌شود. از رابطهٔ (۱۳) برای محاسبهٔ احتمال انتخاب امپراطوری‌ها استفاده شده است:

$$P_n = e^{\left(-\alpha \times TotalCost_n / \max(TotalCost)\right)} \quad (۱۳)$$

متغیر P_n احتمال انتخاب امپراطوری n ام، و $TotalCost_n$ مقدار هزینهٔ کل مربوط به امپراطوری n ام است و عبارت $\max(TotalCost)$ بیشترین مقدار تابع هزینهٔ کل بین تمام امپراطوری‌ها را شامل می‌شود. مقدار α پارامتر فشار انتخاب نامیده می‌شود. هر چه α بزرگتر انتخاب شود، احتمال انتخاب امپراطوری‌های قدرتمند بیشتر و احتمال انتخاب امپراطوری‌های ضعیف، کمتر خواهد شد. با اعمال این فاکتور، به آسانی می‌توان بر سرعت همگرایی تاثیر گذاشت. البته در کل باید بین فرآیند بهره‌برداری و جستجو تعادلی برقرار ساخت.

۳-۲- روش پیشنهادی جایگزین بردار متعامد در رقابت استعماری متعامد

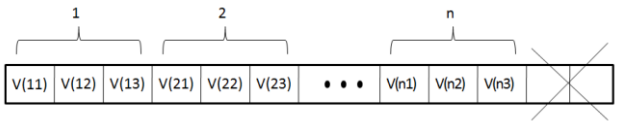
در مقالهٔ کاوه و طلاطاهری [۱۹] نسخه‌های مختلفی از الگوریتم ICA مطرح کرده و عملکرد آنها را با یکدیگر مورد بررسی قرار داده‌اند. یکی از روش‌های بهبود قابلیت جستجو، افزودن بردار متعامد به راستای حرکتی مستعمره به سمت استعمارگر می‌باشد که این بردار با V_2 در رابطهٔ (۱۴) نشان داده شده است [۱۲].

$$\{x\}_{new} = \{x\}_{old} + \beta \times d \times \{cm\} \otimes \{V_1\} + cm \times \tan(\theta) \times d \times \{V_2\}$$
$$\{V_1\} \cdot \{V_2\} = 0, \quad \|\{V_2\}\| = 1 \quad (۱۴)$$

یکی دیگر از مشکلاتی که هنگام بالا بودن تعداد متغیرهای مستقل تابع (متغیرهای تصمیم) مطرح است، ایجاد بردار متعامد در گام الگوریتم رقابت استعماری متعامد می‌باشد. در واقع تولید چنین بردار تصادفی متعامد در ابعاد بالا کار بسیار دشواری است. ما در این مقاله روشی جایگزینی برای روش بالا ارائه خواهیم کرد که شبه کد آن در ادامه به صورت زیر می‌باشد. روش پیشنهادی سرعت تولید بردار متعامد را تسریع بخشیده و فضای جستجوی الگوریتم را هدف‌دار می‌سازد.

گام اول: تقسیم بردار به بردارهای سه درایه‌ای که در شکل (۵) به تصویر کشیده شده است. همان طور که در شکل نشان داده شده است درایه‌های اضافی را در گام اول حذف می‌کنیم.

Downloaded from jor.iranaict.ir at 18:12 +0330 on Wednesday February 11th 2026



شکل ۵- تقسیم بردار اصلی V به زیر بردارهای سه درایه‌ای

که بردارهای V_1 و V_2 و ... به صورت روابط (۱۵) تعریف می‌شوند:

$$V_1 = [V(11)\ V(12)\ V(13)]$$
$$V_2 = [V(21)\ V(22)\ V(23)]$$
$$V_i = [V(i1)\ V(i2)\ V(i3)] \qquad i = 1,2,\dots,n \quad (۱۵)$$

گام دوم: ضرب بردار تفکیک شده در بردار تصادفی جهت تولید بردار متعامد کوچک می‌باشد، که عمل ضرب خارجی بین دو بردار مطابق رابطه (۱۶) انجام می‌گیرد.

$$U \times V = (u_1 i + u_2 j + u_3 k) \times (v_1 i + v_2 j + v3k = u2v3 - u3v2i + u3v1 - u1v3j + u1v2 - u2v1k \quad (۱۶)$$

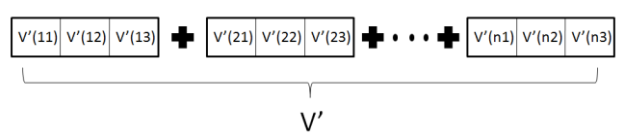
با استفاده از ضرب برداری بالا بردار متعامد V' بر بردار V و بردار تصادفی W بدست می‌آید که طبق روابط (۱۷) تعریف می‌شود:

$$W = rand(1,3)$$
$$V_i' = V_i \otimes W = [V(i1)\ V(i2)\ V(i3)] \otimes W \qquad i = 1,2,\dots,n \quad (۱۷)$$

گام سوم: بردار تولیدی V_i' بر بردار V_i و W عمود است که تعامد بین بردارهای V_i' و V_i حائز اهمیت است که به این معنی است که ضرب داخلی دو بردار V_i' و V_i مطابق رابطه (۱۸) برابر با صفر خواهد بود.

$$V_i' \perp V_i, W$$
$$V_i' \cdot V_i = 0 \qquad i = 1,2,\dots,n \quad (۱۸)$$

گام چهارم: با کنار هم قرار دادن بردارهای V_i' بردار نهایی V' را بدست می‌آوریم که این گام در شکل (۶) به تصویر کشیده شده است:

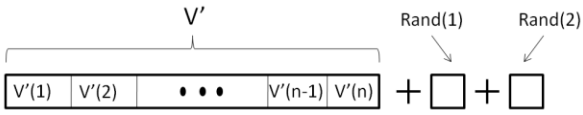


شکل ۶- تشکیل بردار V' از به هم پیوستن ریز بردارهای V_i'

گام پنجم: بردار حاصل از گام قبلی باید بر بردار اصلی اولیه عمود باشد، یعنی باید رابطه‌ی (۱۹) همواره برقرار باشد:

$$V'.V = 0 \quad \Rightarrow \quad V' \perp V \quad (۱۹)$$

گام ششم: در صورتی که تعداد درایه های بردار V' و V با هم برابر نباشند ضرب بالا عملا ممکن نخواهد بود. بنابراین برای این منظور برای برابرسازی تعداد درایه های بردار V' با بردار V به تعداد لازم درایه به انتهای بردار V' اضافه خواهیم کرد. شکل (۷) چگونگی این گام را نشان می‌دهد.



شکل ۷- برابرسازی تعداد درایه‌های بردار V' با تعداد درایه‌های بردار اصلی V

همان طور که در شکل بالا مشاهده می‌شود درایه‌های تصادفی $rand(1)$ و $rand(2)$ به منظور برابرسازی تعداد درایه‌ها به بردار V' اضافه می‌کنیم. در صورتی که تعداد درایه‌های بردار V برابر با $3k+1$ باشد تنها به اضافه کردن $rand(1)$ و در صورتی که برابر با $3k+2$ باشد با اضافه کردن هر دو مقادیر $rand(1)$ و $rand(2)$ تعداد درایه‌های بردار V' را با تعداد درایه‌های بردار V یکسان می‌کنیم.

گام هفتم: در نهایت حاصل ضرب داخلی دو بردار V و V' برابر با مقدار کوچکی خواهد بود و از این رو بردار V' برداری نزدیک به بردار عمود بر V خواهد بود.

۳-الگوریتم رقابت استعماری پیشنهادی در آموزش شبکه عصبی پرسپترون چند لایه

در این مقاله برای بررسی توانایی طبقه‌بندی شبکه عصبی پرسپترون چند لایه که توسط الگوریتم رقابت استعماری آشوبی متعامد اصلاح شده، از دو مجموعه داده با ابعاد مختلف استفاده شده است. مورد اول مجموعه دادهٔ یونسفر با ابعاد متوسط و مورد دوم مجموعه دادهٔ سونار با ابعاد بالا می‌باشد. جدول زیر ویژگی‌های مجموعه داده‌های بکار رفته را به طور جامع به تصویر می‌کشد. در جدول (۱) تعداد داده‌های آموزش و تست با توجه به روش ارزیابی $k-Fold$ بیان شده است.

Downloaded from jor.iranaict.ir at 18:12 +0330 on Wednesday February 11th 2026

جدول ۱- مشخصات و ویژگی‌های داده‌های به کار رفته در آموزش شبکه عصبی

تعداد الگو	نوع ابعاد داده	ویژگی مجموعه داده	نوع مجموعه داده	تعداد خروجی	نوع ورودی	تعداد ورودی	نام داده
۳۵۱	با بعد متوسط	چند متغیره	طبقه‌بندی	۱	اعداد صحیح و حقیقی	۳۴	دادهٔ یونسفر
۲۰۸	با بعد بالا	چند متغیره	طبقه‌بندی	۱	اعداد حقیقی	۶۰	دادهٔ سونار

مجموعه دادهٔ یونسفر شامل طبقه‌بندی سیگنال‌های بازگشتی رادار از الکترون‌های آزاد یک یونسفر است. این مجموعه شناسایی نوع خاصی از ساختار در یونوسفر را شامل می‌شود [۲۰]. در این مجموعه داده ۳۴ مشخصه ورودی وجود دارد و تعداد کل الگوها ۳۵۱ عدد می‌باشد. با اعمال این مجموعه داده به شبکه عصبی، ساختار پرسپترون چند لایه ۱-۳-۳۴ خواهد بود. در نتیجه ورودی‌های شبکه عصبی شامل اعداد صحیح و حقیقی است و تعداد وزن‌های کل شبکه که توسط الگوریتم پیشنهادی نیاز به بروز رسانی خواهد داشت، ۱۰۹ وزن است.

مجموعه دادهٔ سونار شامل ۲۰۸ الگو با ۶۰ ویژگی می‌باشد. هدف ما آموزش یک شبکه عصبی پرسپترون ۳ لایه به منظور تشخیص و تفکیک سیگنال‌های سونار منعکس شده از یک استوانهٔ فلزی و سیگنال‌های منعکس شده از تخته سنگی تقریبا استوانه‌ای می‌باشد [۲۱]. ساختار تعیین شده برای لایه‌های شبکه عصبی MLP طبق ۱-۳-۶۰ است. در جدول (۲) ویژگی‌های شبکه MLP آموزش داده شده، نشان داده شده است.

جدول ۲- مشخصات شبکه عصبی به کار رفته در آموزش

محدودهٔ بروزرسانی	تعداد وزنهای بروزرسانی شده	نوع توابع فعال‌سازی	ساختار لایه-های شبکه	
(-۱,۱)	۱۰۹	تانژانت هایپربولیک سیگموید	۱-۳-۳۴	دادهٔ یونسفر
(-۱,۱)	۱۸۷	تانژانت هایپربولیک سیگموید	۱-۳-۶۰	دادهٔ سونار

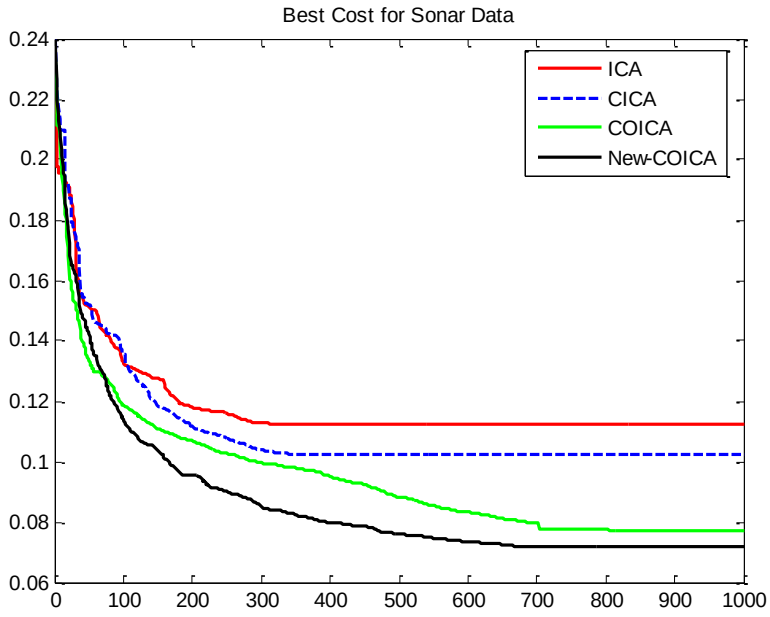
۳-۱-مقایسه روش پیشنهادی با سایر نسخه‌های الگوریتم رقابت استعماری

در این قسمت از مقاله، شبکه عصبی MLP را با الگوریتم پیشنهادی به همراه سه نسخهٔ دیگر از الگوریتم رقابت استعماری شامل الگوریتم استاندارد (ICA)، رقابت استعماری آشوبی (CICA) و نسخهٔ اولیهٔ الگوریتم COICA آموزش داده و منحنی خطای میانگین مربعات را در یک شکل بدست آورده‌ایم. در این شبیه‌سازی‌ها تعداد تکرارهای هر الگوریتم در ۱۰۰۰ تکرار محدود شده است. به منظور مقایسهٔ صحیح الگوریتم‌ها با یکدیگر، شرایط اولیه برای شروع الگوریتم‌ها را یکسان فرض می‌کنیم. برای این منظور، جمعیت اولیه‌ای را ایجاد و ذخیره کرده‌ایم. برای آموزش شبکه توسط الگوریتم مربوطه، همان جمعیت اولیهٔ تعیین شده را فراخوانی کرده‌ایم. جدول (۳) پارامترهای مختلف و قابل تنظیم الگوریتم‌های مذکور را نشان می‌دهد.

Downloaded from jor.iranaict.ir at 18:12 +0330 on Wednesday February 11th 2026

جدول ۲- پارامترهای قابل تنظیم الگوریتم‌های مورد بررسی

نام پارامتر	ویژگی پارامتر	مقدار انتخاب شده	الگوریتمهای بکار رفته
VarSize	طول بردار اصلی	بسته به نوع مساله	تمام الگوریتم‌ها
VarMin	کران پایین بردار اصلی	-۱	تمام الگوریتم‌ها
VarMax	کران بالا بردار اصلی	+۱	تمام الگوریتم‌ها
MaxIt	ماکزیمم مقدار تکرارها	۱۰۰۰	تمام الگوریتم‌ها
Npop	تعداد جمعیت اولیه	۸۰	تمام الگوریتم‌ها
nEmp	تعداد امپراطوری‌های اولیه	۱۰	تمام الگوریتم‌ها
Ncol	تعداد مستعمرات	۷۰	تمام الگوریتم‌ها
Alpha	فاکتور فشار	۱	تمام الگوریتم‌ها
Beta	ضریب جذب	۲	تمام الگوریتم‌ها
pRevolution	احتمال انقلاب	۰,۱	تمام الگوریتم‌ها
Mu	نرخ انقلاب	۰,۰۵	تمام الگوریتم‌ها
zeta	ضریب میانگین هزینه مستعمرات	۰,۱	تمام الگوریتم‌ها



شکل ۸- مقایسه نتایج نسخه‌های مختلف ICA برای مجموعه دادهٔ سونار

Downloaded from jor.iranaict.ir at 18:12 +0330 on Wednesday February 11th 2026

در نمودار شکل (۸) محور عمودی نشان‌دهندهٔ خطای میانگین مربعات[18] آموزش شبکه (MSE) و محور افقی تعداد تکرارها می‌باشد. در شکل بالا می‌توان مشاهده کرد که خطای میانگین مربعات آموزش شبکهٔ عصبی MLP با استفاده از الگوریتم پیشنهادی (New-COICA) نسبت به سایر روش‌ها بیشتر کاهش یافته است. از آنجایی که هدف از آموزش شبکهٔ عصبی، طبقه‌بندی الگوهای جدید می‌باشد، میزان درصد خطای طبقه‌بندی نیز برای هر الگوریتم در جدول (۴) آورده شده است.

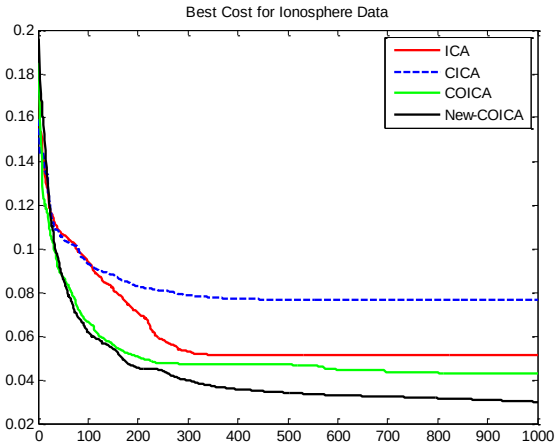
جدول ۳- مقایسه نتایج آماری شبیه‌سازی‌ها برای نسخه‌های مختلف ICA در ۱۰ تکرار برای دادهٔ سونار

	Train MSE	Mean of Classification Error	Std of MSEtr	Std of Error
ICA	0.1125	14.2857	0.0029	4.7619
CICA	0.1038	17.4603	0.0059	5.4986
COICA	0.0803	17.4603	0.0241	2.7493
New-COICA	0.0640	9.5238	0.0088	8.2479

در جدول (۴) پارامتر Train MSE نشان‌دهندهٔ خطای آموزش از طریق روش میانگین مربعات خطا می‌باشد، پارامتر بعدی Mean Classification Error نشان‌دهندهٔ درصد خطای طبقه‌بندی است که در واقع درصد خطای خروجی را نشان می‌دهد. پارامتر Std of MSEtr بیان‌کنندهٔ انحراف معیار خطای آموزش شبکه عصبی مصنوعی است و در نهایت پارامتر Std of Error انحراف استاندارد خطای طبقه‌بندی خروجی را نشان می‌دهد.

در جدول (۴)، پارامترهای انحراف استاندارد خطای آموزش و درصد خطای طبقه‌بندی نیز برای ۱۰ بار تکرار شبیه‌سازی‌ها نشان داده شده است. انحراف استاندارد خطای طبقه‌بندی نسبت به سایر روش‌ها کاهش نیافته است، اما میزان میانگین درصد خطای طبقه‌بندی نسبت به سایر نسخه‌ها کاهش چشم‌گیری یافته است یعنی به مقدار ۹٫۵٪ رسیده است و همچنین در خطای آموزش شبکه نیز این برتری از آن الگوریتم پیشنهادی می‌باشد. انحراف معیار درصد خطای طبقه‌بندی نسبت به سایر الگوریتم‌ها با اختلاف اندکی بیشتر است، از این رو الگوریتم پیشنهادی قابلیت اطمینان بر الگوریتم‌ها را بهبود نبخشیده است.

با اعمال شبیه‌سازی‌ها بر روی مجموعه دادهٔ یونسفر نیز نتایج زیر بدست آمده است. شکل (۹) خطای میانگین مربعات برای روش‌های مختلف را به تصویر می‌کشد. در این شکل محور عمودی خطای MSE آموزش شبکه و محور افقی تعداد تکرارها را نشان می‌دهد.



شکل ۹. مقایسه نتایج نسخه‌های مختلف ICA برای مجموعه دادهٔ یونسفر

و نتایج آماری بدست آمده از ۱۰ بار تکرار در شبیه‌سازی‌ها در جدول (۵) نشان داده شده است.

جدول ۴. مقایسه نتایج آماری شبیه‌سازی‌ها در ۱۰ تکرار برای نسخه‌های مختلف ICA در داده یونسفر

	Train MSE	Mean of Classification Error	Std of MSEtr	Std of Error
ICA	0.1028	15.7143	0.0056	9.7590
CICA	0.0778	17.8571	0.0124	9.4401
COICA	0.0520	12.1429	0.0055	7.1429
New-COICA	0.0491	8.5450	0.0219	3.2991

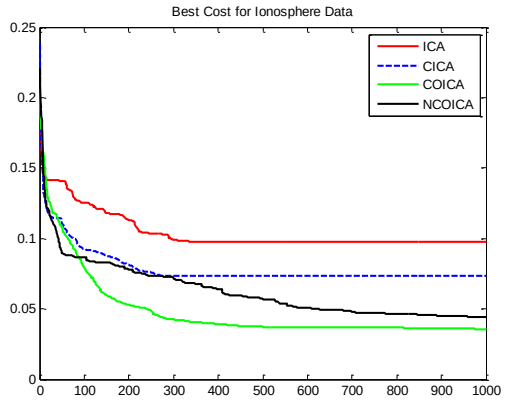
[18] Mean Square Error

Downloaded from jor.iranaict.ir at 18:12 +0330 on Wednesday February 11th 2026

در جدول بالا نیز پارامتر Train MSE نشان دهندهٔ خطای آموزش از طریق روش میانگین مربعات خطا می‌باشد، پارامتر بعدی Mean Classification Error نشان‌دهندهٔ درصد خطای طبقه‌بندی است. پارامتر Std of MSEtr بیان‌کنندهٔ انحراف معیار خطای آموزش شبکه است و پارامتر Std of Error انحراف استاندارد خطای طبقه‌بندی خروجی را نشان می‌دهد.

خطای میانگین مربعات برای آموزش شبکه برای این مجموعه داده نیز با استفاده از الگوریتم پیشنهادی نسبت به سایر نسخه‌ها به مقدار کمتری همگرا شده است و این در حالی است که درصد خطای طبقه‌بندی نیز مقدار ۸٫۵٪ بدست آمده که کمترین مقدار در بین این الگوریتم‌ها می‌باشد. انحراف استاندارد برای درصد خطای طبقه‌بندی در روش پیشنهادی مقدار کمتری نسبت به سایر نسخه‌ها بدست آمده است و این در حالی است که انحراف استاندارد خطای MSE برای الگوریتم پیشنهادی نسبت به سایر الگوریتم‌ها بیشترین مقدار را دارد.

در بعضی از اجراها الگوریتم پیشنهادی نسبت به بعضی از الگوریتم‌ها، در کاهش خطای MSE آموزش توانایی کمتری از خود نشان می‌دهد. شکل (۱۰) نمونه‌ای از این اجرا را نشان می‌دهد. در شکل مذکور محور عمودی بیانگر خطای میانگین مربعات آموزش شبکه و محور افقی بیانگر تعداد تکرارها می‌باشد.

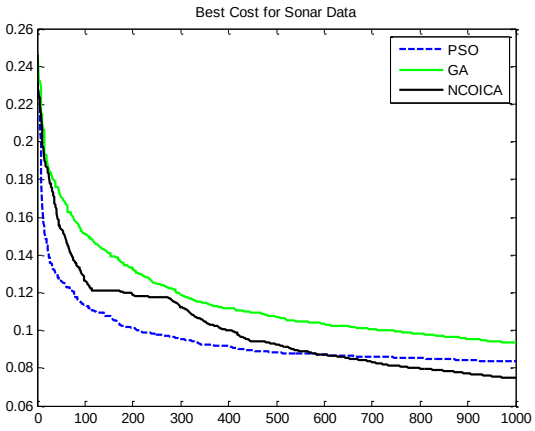


شکل ۱۰- وقوع پدیدهٔ بیش برازش در آموزش شبکه عصبی برای مجموعه دادهٔ یونسفر در نسخهٔ قبلی

با مشاهده شکل (۱۰) انتظار می‌رود که درصد خطای طبقه‌بندی برای الگوریتم COICA سابق نسبت به الگوریتم پیشنهادی مقدار کمتری بدست آید، در حالیکه درصد خطای طبقه‌بندی برای الگوریتم COICA برابر ۲۰٪ و برای الگوریتم پیشنهادی ۵/۷٪ بدست آمده است. می‌توان نتیجه گرفت که شبکه عصبی در حین آموزش توسط الگوریتم COICA دچار پدیدهٔ بیش برازش شده است. می‌توان دید که در اغلب موارد الگوریتم پیشنهادی با وجود خطای آموزش بیشتر، به درصد خطای طبقه‌بندی کمتری دست یافته است. در واقع عمومیت پذیری آموزش توسط الگوریتم پیشنهادی بهبود یافته است.

۳-۲- مقایسهٔ الگوریتم پیشنهادی با سایر الگوریتم‌های تکاملی

در این بخش از مقاله، عملکرد الگوریتم پیشنهادی را نسبت به سایر الگوریتم‌های تکاملی بررسی می‌کنیم. الگوریتم ژنتیک و الگوریتم ازدحام ذرات جزو محبوب‌ترین و پرکاربردترین الگوریتم‌های بهینه‌سازی می‌باشند. به این منظور در ادامه به بررسی نتایج این الگوریتم‌ها با الگوریتم پیشنهادی خواهیم پرداخت. شکل (۱۱) منحنی همگرایی خطای میانگین مربعات آموزش برای الگوریتم‌های GA و PSO با الگوریتم پیشنهادی را برای مجموعه دادهٔ سونار نشان می‌دهد. در این شکل محور عمودی بیانگر خطای MSE آموزش شبکه و محور افقی نشان‌دهنده تعداد تکرارها است.



شکل ۱۱. مقایسه نتایج روش پیشنهادی با سایر الگوریتم‌های تکاملی برای دادهٔ سونار

و نتایج آماری برای ۱۰ تکرار برای هر الگوریتم در جدول (۶) نشان داده شده است.

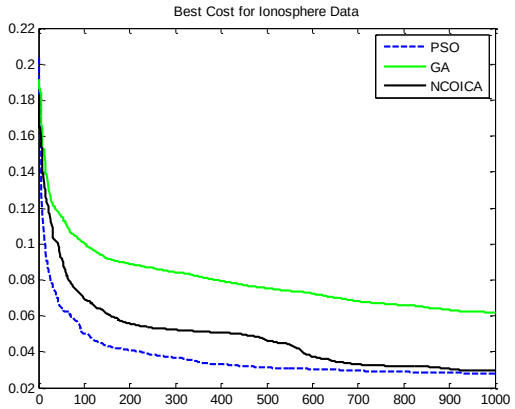
Downloaded from jor.iranaict.ir at 18:12 +0330 on Wednesday February 11th 2026

جدول ۵. مقایسه نتایج آماری شبیه‌سازی‌ها در ۱۰ تکرار برای هر الگوریتم در دادهٔ سونار

	Train MSE	Mean Classification Error	Std of MSEtr	Std of Error
GA	0.0926	22.4603	0.0043	10.9971
PSO	0.0774	20.6349	0.0055	5.4986
New-COICA	0.0742	19.0476	0.0053	4.7619

در جدول (۶) نیز پارامتر Train MSE خطای آموزش از طریق روش میانگین مربعات خطا را نشان می‌دهد، پارامتر Mean Classification Error نشان‌دهندهٔ درصد خطای طبقه بندی و پارامتر Std of MSEtr بیان‌کنندهٔ انحراف معیار خطای آموزش شبکه و پارامتر Std of Error انحراف استاندارد خطای طبقه‌بندی خروجی را نشان می‌دهد.

منحنی خطای MSE آموزش برای دادهٔ یونسفر نیز به صورت شکل (۱۲) بدست آمده است. در شکل زیر خطای میانگین مربعات آموزش شبکه توسط محور عمودی نشان داده شده است و محور افقی تعداد تکرارها را نشان می‌دهد.



شکل ۱۲- مقایسه روش پیشنهادی با سایر الگوریتم‌های تکاملی برای دادهٔ یونسفر

و نتایج آماری شبیه‌سازی‌ها برای مجموعه دادهٔ یونسفر در ۱۰ تکرار به اختصار در جدول (۷) بدست آمده است.

جدول ۷- مقایسه نتایج آماری شبیه‌سازی‌ها در ۱۰ تکرار برای هر الگوریتم در دادهٔ یونسفر

	Train MSE	Mean Classification Error	Std of MSEtr	Std of Error
GA	0.0628	14.2857	0.0051	10.3016
PSO	0.0303	13.3333	0.0038	3.2991
New-COICA	0.0335	11.4286	0.0036	5.7143

در جدول بالا نیز پارامترهای مطرح شده در جدول از سمت چپ به راست به ترتیب عبارت است از خطای میانگین مربعات خطای آموزش شبکه، درصد خطای طبقه‌بندی شبکهٔ آموزش دیده، انحراف معیار خطای آموزش شبکه و انحراف استاندارد خطای طبقه‌بندی خروجی.

با مشاهدهٔ نتایج بالا مشاهده می‌شود، که خطای میانگین مربعات آموزش با استفاده از الگوریتم PSO نسبت به الگوریتم پیشنهادی به مقدار کمتری کاهش یافته است و این در حالی است که الگوریتم پیشنهادی به درصد خطای طبقه‌بندی بهتری دست یافته است. بهبود درصد خطای طبقه‌بندی را می‌توان به علت توانایی بهتر و بیشتر در جستجو توسط الگوریتم پیشنهادی دانست. این ویژگی باعث کاهش احتمال رخداد پدیدهٔ بیش برازش در روند آموزش شبکه گشته و در نتیجه باعث بهبود قابل توجه طبقه‌بندی داده‌ها توسط شبکه عصبی مصنوعی شده است. انحراف استاندارد خطای آموزش برای الگوریتم پیشنهادی در مقایسه با دو الگوریتم دیگر کاهش یافته است و انحراف استاندارد درصد خطای طبقه‌بندی نسبت به الگوریتم ژنتیک بهبود یافته و با اختلاف اندکی از الگوریتم PSO بیشتر است.

۴- نتیجه‌گیری

در این مقاله با اعمال سه تغییر اصلی در الگوریتم رقابت استعماری آشوبی متعامد سعی در بهبود عملکرد الگوریتم مزبور کرده‌ایم. در الگوریتم پیشنهادی از تابع توزیع بولتزمان جهت تعیین احتمال تصاحب ضعیف‌ترین مستعمره در ضعیف‌ترین امپراطوری استفاده شده است. همچنین جهت انتخاب امپراطوری قدرتمند برای عملیات ذکر شده، از روش انتخاب چرخ رولت استفاده شده است. در نهایت با استفاده از روش جدیدی، تغییراتی در راستای حرکت مستعمره به

Downloaded from jor.iranaict.ir at 18:12 +0330 on Wednesday February 11th 2026

سمت استعمارگر مربوطه اعمال کرده‌ایم. اجرای شبیه‌سازی‌ها بر روی دو مجموعه دادهٔ سونار و یونسفر انجام گرفت که شامل دو بخش، یعنی بررسی با سایر نسخه‌های رقابت استعماری و بررسی با سایر الگوریتم‌های تکاملی است. نتایج بدست آمده نشان می‌دهد که الگوریتم پیشنهادی در مقایسه با سایر نسخه‌های رقابت استعماری به خطای میانگین مربعات کمتری در آموزش شبکه رسید و همچنین درصد خطای طبقه‌بندی کمتری نسبت به سایر نسخه‌ها از خود نشان داد. همچنین تعمیم‌پذیری الگوریتم پیشنهادی نسبت به سایر نسخه‌ها بهبود یافت. در مقایسه با سایر الگوریتم‌های تکاملی، الگوریتم پیشنهادی نسبت به الگوریتم ژنتیک به خطای آموزش کمتری همگرا شد و تنها با اختلاف اندکی نسبت به الگوریتم ازدحام ذرات دارای خطای آموزش بیشتری است و این در حالی است که با وجود خطای آموزش بیشتر به درصد خطای طبقه‌بندی بهتری دست یافته است. در کل عمومیت‌پذیری الگوریتم پیشنهادی نسبت به سایر روش‌ها بهبود یافت. انحراف استاندارد درصد خطای طبقه‌بندی و یا خطای آموزش شبکه به طور کلی کاهش محسوسی نیافته است.

منابع

1.M. Melanie, “An Introduction to Genetic Algorithms,” Massachusett's MIT Press, 34(7):pp.1-9, 1999.

2.L. A. Ingber, “Simulated annealing: practice versus theory,” J. Math. Comput. Modell., 18(11):pp. 29–57, 1993.

3.Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. Proceedings of IEEE, 1942– 1948 (1995)

4.M. Dorigo, V. Maniezzo and A. Colorni, “The ant system: optimization by a colony of cooperating agents,” IEEE Transaction System Man Cybern, B 26(1):pp. 29–41, 1996.

5.B. Franklin and M. Bergerman, “Cultural Algorithms: Concepts and Experiments,” In Proceedings of the IEEE Congress on Evolutionary Computation, 2: pp. 1245–1251, 2000.

6.R. Storn and K. Price, “Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces,” Journal of Global Optimization, 11(4):pp. 341–359, 1997.

7.K. Lee and Z. Geem, “A new structural optimization method based on the harmony search algorithm,” Computers and Structures, 82:781-98, 2004.

8.E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, “A Gravitational Search Algorithm,” Information Science, Special Section on High Order Fuzzy Sets, 179(13): pp. 2232-2248, 2009.

9.Brownlee, J., Clever Algorithms: Nature-Inspired Pro Recipes, LuLu Enterprises Incorporated, (1st Edition), 2011.

10.Kaveh A, Talatahari S. Novel heuristic optimization method: Charged system search, Acta Mechanica, doi: 10.1007/s00707-009-0270-4, 2010.

11.Atashpaz-Gargari, E. and Lucas, C., "Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition", IEEE Congress on Evolutionary Computation (CEC), pp. 4661-4667, 2007.

12.Talatahari, S. and Farahmand Azar, B. and Sheikholeslami, R. and Gandomi, A.H., "Imperialist competitive algorithm combined with chaos for global optimization", Commun. Nonl. Sci. Numer. Simulat., vol. 17, pp. 1312-1319, 2012.

13.Kaveh, A. and Talataheri, S., "Optimum design of skeletal structures using imperialist competitive algorithm", computers & structures journal, vol. 88(21), pp. 1220-1229, 2010.

Downloaded from jor.iranaict.ir at 18:12 +0330 on Wednesday February 11th 2026

14.H. Bahrami, K. Faez, M. Abdechiri, “Imperialist Competitive Algorithm using Chaos Theory for Optimization,” UKSim-AMSS 12th International Conference on Computer Modeling and Simulation, 2010.

15.M. Abdechiri, K. Faez and H. Bahrami, “Neural Network Learning based on Chaotic Imperialist Competitive Algorithm,” The 2nd International Workshop on Intelligent System and Applications (ISA2010), 2010.

16.Abdechiri, M. and Faez, K. and Bahrami, H., "Adaptive Imperialist Competitive Algorithm (AICA)", 9th IEEE International Conference on Cognitive Informatics (ICCI), pp. 940-945, 2010.

17.Coelho, L. D. and Afonso, L. and Alotto, P., "A Modified Imperialist Competitive Algorithm for Optimization in Electromagnetics", IEEE Transactions on Magnetics, vol. 48, pp. 579-582, 2012

18.Soltani-Sarvestani, M. A. and Lotfi, S. and Ramezani, F., "Quad Countries Algorithm (QCA)", Lecture Notes in Computer Science, vol. 7198, pp. 119-129, 2012.

19.Kaveh, A. and Talataheri, S., "Imperialist Competitive Algorithm For Engineering Design Problems", Asian Journal Of Civil Engineering, vol. 11, pp. 675-697, 2010.

20.Sigillito, V.G., Wing, S.P., Hutton, L.V., and Baker, K.B. (1989), ‘Classification of Radar Returns from the Ionosphere using Neural Networks’, Johns Hopkins APL Technical Digest, 10, 262–266.

21.Gorman, R.P., and Sejnowski, T.J. (1988), ‘Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets’, Neural Networks, 1, 75–89.