

بهبود الگوریتم KPSO با استفاده از سیستم ایمنی مصنوعی

زهره افصحی^۱ ، محمدرضا میدی^۲

۱ دانشکده مهندسی کامپیوتر، دانشگاه آزاد قزوین

Afsahi_ai@yahoo.com

۲ دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر

mmeybodi@aut.ac.ir

چکیده:

در این مقاله تکنیکی ارائه شده است که الگوریتم گروه ذرات بهینه‌های چند تابی را در توابع دارای چند بهینه مکانیابی کند. در این مقاله الگوریتم گروه ذرات بهینه‌ساز مبتنی بر کلاس‌بندی ارائه شده است. استفاده از الگوریتم کلاس-بندی $k-means$ به منظور مشخص کردن تطبیقی **Niche**ها، به الگوریتم گروه ذرات بهینه‌ساز کمک می‌کند تا بتواند تمامی راه حل‌های بهینه را پیدا کند. در هر **Niche** از الگوریتم سیستم ایمنی مصنوعی برای به دست آوردن ذرات آن استفاده خواهد شد. نتایج نشان می‌دهد که الگوریتم پیشنهادی می‌تواند تمامی بهینه‌های عمومی را در توابع تست با موفقیت مکانیابی کند.

کلمات کلیدی:

الگوریتم گروه ذرات بهینه‌ساز، الگوریتم **Niching** و سیستم ایمنی مصنوعی.

بنابراین ضروری به نظر می‌رسد که یا باید الگوریتم‌های خاصی طراحی شوند و یا ساختار همان الگوریتم‌ها به نحوی تغییر کند که قادر به پیدا کردن تمامی راه حل‌ها باشند. الگوریتم‌های بهینه‌سازی مبتنی بر فرضیه تکامل با معروفی مفهوم Niching تغییر پیدا خواهند کرد. زمانیکه تکنیک‌های Niching به یک الگوریتم جستجو اعمال می‌شوند، الگوریتم را قادر می‌سازند تا فضای جستجو را به زیر فضاهایی تقسیم کند و در هر یک از این زیر فضاهای به جستجو بپردازد. در این مقاله، تمرکز روی الگوریتم جستجوی گروه ذرات بهینه‌ساز است و یک متد جدید از این الگوریتم همراه با تکنیک Niching و مبتنی بر کلاس‌بندی معرفی شده است. در مدل پیشنهادی به منظور تعیین **Niche**ها از الگوریتم $k-means$ و برای مشخص کردن اعضای هر **Niche** از الگوریتم سیستم ایمنی مصنوعی استفاده شده است. این مقاله با معرفی الگوریتم گروه ذرات بهینه ساز شروع می‌شود. در ادامه و در بخش سوم به مفهوم Niching و کاربرد آن در الگوریتم‌های تکاملی خواهیم پرداخت. در بخش چهارم به تفصیل در مورد کلاس‌بندی و به کارگیری آن در تکنیک Niching و همچنین کاربرد آن در KPSO می‌پردازیم. در ادامه و در بخش پنجم مختصراً در مورد سیستم

۱- مقدمه

طراحی الگوریتم‌های بهینه‌سازی به گونه‌ای است که معمولاً به یک راه حل بهینه همگرا می‌شوند. از این‌رو، در بسیاری از شرایط امر ایده‌آلی نیست. بعضی از مسائل بهینه‌سازی بیش از یک راه حل بهینه دارند. در چنین مواردی یک الگوریتم بهینه‌سازی ایده‌آل باید بتواند تمامی راه حل‌ها را پیدا کند. زیرا هر یک از این راه حل‌ها با توجه به زمینه‌ای که الگوریتم بهینه‌سازی مورد استفاده قرار می‌گیرد، می‌تواند راه حل اصلی مسئله باشد. برای مثال یک راه حل خاص با اضافه کردن یک ضابطه انتخاب می‌شود، یا برخی از راه حل‌ها در شرایط متفاوت ترجیح داده می‌شوند، یا ترکیب تمامی راه حل‌ها می‌تواند برای یک مسئله خاص مورد استفاده قرار بگیرد. زمانیکه توابع بیش از یک بهینه عمومی و یا بیش از یک بهینه محلی که ارزشش خیلی نزدیک به بهینه عمومی است، دارند، رفتار الگوریتم‌های بهینه‌سازی ایده‌آل نیست. در برخی از موارد و با توجه به طراحی این الگوریتم‌ها، در محیط‌های چند بهینه‌ای یا فقط یکی از این راه حل‌ها برگردانده می‌شود و یا به علت حضور بیش از یک بهینه دچار سردرگمی می‌شود و در نتیجه در همگرا شدن به جواب با شکست مواجه می‌شود (در برخی از الگوریتم‌های ژنتیک [2]).

موازی و ترتیبی [6]. در Niching موازی تعدادی از Niche ها در جمعیت به طور همزمان مشخص و نگه-داری می‌شوند. در حالیکه متدهای Niching ترتیبی راه حل های چندتایی را با تکرار متدهای Niching در فضای مسئله پیدا می‌کند. این پروسه تا زمانی که یک راه حل بالقوه در هر تکرار مشخص شود که تضمینی باشد برای جلوگیری از جستجوی دوباره، ادامه پیدا می‌کند. تکنیک-های Niching چه موازی و چه ترتیبی به طور گسترده‌ای در الگوریتم‌های تکاملی از جمله الگوریتم ژنتیک مورد مطالعه قرار گرفته [16,17] شده و انواع مختلفی از آن موجود است [19]. به تازگی تکنیک‌های Niching در الگوریتم گروه ذرات بهینه‌ساز نیز وارد شده‌اند [12] تا این الگوریتم را قادر سازند که تمامی راه حل های بهینه را در توابع چند بهینه‌ای مشخص کند. این تکنیک‌ها عبارتند از:

۱-۳- بسط دادن تابع شایستگی

این روش به عنوان یک تکنیک Niching ترتیبی برای الگوریتم گروه ذرات به کار می‌رود. زمانیکه الگوریتم گروه ذرات بهینه ساز یک ماکریم محلی (x^*) را مشخص می‌کند، تابع شایستگی بسط داده می‌شود، برای هر نقطه X جاییکه $(x^*) > f(x)$ ، $f(x) < f(x^*)$ تاثیر نمی‌پذیرد. سایر ذرات که $x^* \leq f(x) \leq f(x^*)$ طوری بسط داده می‌شوند، که x^* یک مینیمم محلی شود. سپس تمامی ذرات به طور تصادفی جایگذاری می‌شوند. ون دن برگ اشاره دارد که تکنیک بسط دادن فضای جستجو ممکن است این فضا را با معروفی بهینه‌های اشتباه دگرگون کند، در نتیجه الگوریتم گروه ذرات بهینه ساز قادر به همگرایی به تمامی راه حل‌ها نباشد. الگوریتم مشابه دیگری در سال ۲۰۰۱ ارائه شد [11] که نیاز به تغییری در الگوریتم گروه ذرات بهینه ساز ندارد و نتایج قابل قبولی در توابع با چندین راه حل بهینه، ارائه کرده است. به هر حال اثر متدهای بسط دادن تابع شایستگی روی همه توابع شایستگی یکنواخت نیست. البته در بعضی از موارد بهینه‌های اشتباه را معروفی می‌کند که این متدهای غیر قابل اعتماد نشان می‌دهد. [9,10]

۲-۳- گروه ذرات بهینه‌ساز $nbest$

در سال ۲۰۰۲، بریتس مدل $nbest$ را معرفی کرد [9]. این متدهای گروه ذرات بهینه‌ساز شده Niching موازی در الگوریتم گروه ذرات بهینه‌ساز است و به این الگوریتم کمک می‌کند تا تمامی راه حل‌های توابع چند بهینه‌ای را

ایمنی مصنوعی و چگونگی استفاده از آن در تعیین ذرات h Niches توضیحاتی داده شده است. در بخش ششم به بررسی پارامترهای اضافی و ارائه نتایج می‌پردازیم. در پایان و در بخش هفتم استنتاج خود را بر اساس نتایج بدست آمده ارائه خواهیم کرد.

۲- الگوریتم گروه ذرات بهینه‌ساز

که $pbest_i$ بهترین موقعیت ملاقات شده توسط ذره آم و $gbest$ بهترین موقعیت ملاقات شده توسط کل گروه است. $rand1_i^d$ و $rand2_i^d$ دو عدد تصادفی در بازه $[0,1]$ می‌باشند. C_1 و C_2 ضرایب مولفه های شناختی و اجتماعی می‌باشند. الگوریتم گروه ذرات بهینه‌ساز، الگوریتمی است که از رفتار اجتماعی موجوداتی که به صورت انبوه زندگی می‌کنند، الهام گرفته است [1]. جمعیت اولیه این الگوریتم در یک بازه مشخص و به صورت تصادفی مقداردهی می‌شود. هر ذره در فضای D بعدی به عنوان یک راه حل از مساله محسوب می‌شود. ذره آم از گروه موقعیت x_i^d و سرعت v_i^d را در بعد d از فضای جستجو دارد. معادله به روز نمودن سرعت و موقعیت ذره در (۱) و (۲) نمایش داده شده است که جستجو بر اساس این فرمول‌ها برای رسیدن به یک بهینه شروع می‌شود. معادلات به روز رسانی مکان و سرعت برای i^{th} ذره به شکل زیر به (۱) نمایش در می‌آیند:

$$V_i^d(t+1) = V_i^d(t) + c_1 rand1_i^d(t)(pbest_i^d(t) - X_i^d(t)) + c_2 rand2_i^d(t)(gbest(t) - X_i^d(t))$$

$$X_i^d(t+1) = X_i^d(t) + V_i^d(t+1) \quad (2)$$

۳- تکنیک‌های Niching در الگوریتم گروه ذرات

بهینه‌ساز

تکنیک‌های Niching تلاش می‌کند تا راه حل‌های چندتایی را در مسائل بهینه‌سازی پیدا کند. زیرا زمانیکه الگوریتم‌های تکاملی بر توابع چند بهینه‌ای اعمال می‌شوند، یا به یک بهینه محلی همگرا می‌شوند و یا دچار سردرگمی شده و حتی در همگرا شدن نیز با شکست روبرو می‌شوند. تکنیک‌های Niching بر دو نوعی:

مشخص کردن Niche ها با استفاده از عملیات روی بردارها است که در الگوریتم گروه ذرات بهینه‌ساز یک امر ذاتی است. بنابراین، یک روش عالی فراهم می‌شود تا تمامی راه حل‌های بهینه را در محیط‌های چند بهینه‌ای مشخص کند.

۴-۳- گروه ذرات بهینه‌ساز مبتنی بر گونه

نسخه دیگر Niching در الگوریتم گروه ذرات بهینه‌ساز، مبتنی بر گونه‌ها است که در سال ۲۰۰۴ پیشنهاد شده است. در این مدل [5] ، الگوریتم گروه ذرات بهینه‌ساز استاندارد از پروسه‌ای شبیه به آنچه در الگوریتم زنتیک [23] انجام گرفته استفاده می‌کند تا مناطق این گونه‌ها را مشخص کند که در واقع مشخص کننده Niche ها در جمعیت هستند. زمانیکه مناطق گونه‌ها مشخص شد، تمام ذرات دیگر به Niche ای که در نزدیکترین منطقه به آنها به وجود آمده است منسوب می‌گردند و در ادامه ساختار همسایگی بر اساس مدل $nbest$ به هر یک از Niche ها اعمال می‌شود. نگرش PSO قادر Niching PSO به تعیین Niche ها به طور پویا در فضای جستجو نیز می‌باشد که این امر باعث بالا رفتن کارآیی این الگوریتم در محیط‌های پویا می‌شود. بدین منظور نیاز به پارامتر شعاع ۵ دارد تا وسعت Niche ها را مشخص کند.

$$r = \frac{\sum_{i=1}^N \min_{j \neq i} \|X_i - X_j\|}{N} \quad (3)$$

۵-۳ Niching PSO تطبیقی

آقای برد و لی یک الگوریتم جدید ارائه کردند[14] که پارامترهای اصلی Niching را بطور تطبیقی مشخص می‌کند که ANPSO نام دارد. در اولین گام از این الگوریتم، ۱ میانگین فاصله بین هر ذره و نزدیکترین همسایه‌اش را به شکل زیر با توجه به معادله (۳) محاسبه می‌کند:

سپس این فاصله برای تشکیل Niche ها استفاده می‌شود. در حقیقت ANPSO مینیمم فاصله بین ذرات در تعداد مشخصی از گامها را دنبال می‌کند. در این روش، یکبار Niche ها توسط الگوریتم مشخص می‌شوند و از توبولوژی $gbest$ برای هر Niche و از توبولوژی ون نیومن برای ذرات خارج از Niche ها استفاده می‌شود. ذراتی که تشکیل یک Niche را می‌دهند، تلاش می‌کنند تا یک جستجوی محلی را در اطراف یک بهینه

مکانیابی کند. تکیک $nbest$ از همسایگی محلی (بر اساس نزدیکی فاصله) استفاده می‌کند که در طول زمان کوچک می‌شود تا بتوانند راه حل‌های چندتایی را مشخص کند. در سال ۲۰۰۲ یک متدهای جدید دیگر به نام NichePSO پیشنهاد گردید[20]. در این متدهای زیر گروه‌ها به منظور پیدا کردن تمامی راه حل‌ها در توابع با چند راه حل بهینه استفاده شده است. روش کار به این شکل است که یک گروه اصلی که در هر لحظه می‌تواند Niche یک زیرگروه تولید کند نگهداری می‌شود و یک گروه ممکن مشخص می‌شود. گروه اصلی نیز با روش گروه ذرات بهینه‌ساز با فاکتور محدود کننده، ارزیابی می‌شود و معادلات سرعت و مکان هر ذره نیز بر اساس این مدل از الگوریتم گروه ذرات بهینه ساز به روز می‌شود. از سوی دیگر، زیرگروه‌ها نیز با الگوریتم GCPSO ارزیابی می‌شوند که همگرا شدن به یک بهینه محلی را تضمین می‌کند[21].

۳-۳- گروه ذرات مبتنی بر بردار موازی

آقای شومون و انگلیرج نوع متفاوتی از Niching را ارائه کردند و این مدل جدید را در الگوریتم ذرات بهینه ساز مبتنی بر بردار به کار گرفتند[22] . در این روش دو پارامتر برداری در معادلات به روز رسانی سرعت در نظر گرفته شده است، که به ترتیب به سمت بهترین مکان ذره و بهترین مکان در همسایگی هر ذره اشاره دارد. زمانیکه این بردارها هم جهت می‌شوند، ذره مسیر خود را به سمت بهترین همسایگی تغییر خواهد داد، در غیر نیچه به سمت بهینه دیگری حرکت می‌کند. Niching ها با توجه به این ضابطه مشخص می‌شوند. الگوریتم گروه ذرات بهینه ساز مبتنی بر بردار Niche ها را به شکل ترتیبی و با شروع از بهترین ذره عمومی و سپس تکرار این پروسه برای تمام ذرات خارج از مشخص و کشف می‌کند. در [4] نیز یک الگوریتم گروه ذرات بهینه ساز مبتنی بر بردار موازی معرفی شده است که یک نسخه کارآمدتر است. اساس این کار همان قوانین مطرح شده در بالا است با این تفاوت که یک نگرش موازی را دنبال می‌کند. در این متدهای متفاوت به شکل موازی مشخص شده و نگهداری می‌شوند. در این روش یک پروسه جدید تعریف می‌گردد و در صورتیکه دو Niche آنقدر به هم نزدیک شوند که فاصله بین آن دو کمتر از یک آستانه مشخص شود، آنها را با هم ترکیب می‌کند. نگرش مبتنی بر بردار، یک مشخصه دارد و آن

است که الگوریتم کلاس‌بندی در هر گام به الگوریتم گروه ذرات بهینه‌ساز اعمال نمی‌شود، بلکه در هر c مرحله این اتفاق می‌افتد. دلیل منطقی این انتخاب گرایش طبیعی گروه به کلاس‌بندی شدن در اطراف بهینه‌های تابع است.

استفاده از پروسه کلاس‌بندی فقط می‌تواند این گرایش طبیعی را بهبود دهد و بالاتر از همه، برای مدتی مانع ارتباط (تبدیل اطلاعاتی) بین ذرات در *Niche*‌های متفاوت می‌شود. به هر حال اگر الگوریتم کلاس‌بندی، ذرات *Niche*‌ها را در هر گام مشخص کند، ذرات فرست نخواهند داشت تا طبق الگوریتم گروه ذرات بهینه‌ساز با سایر ذرات در تبادل اطلاعاتی باشند. البته باید به این نکته اشاره کرد که این شیوه اعمال الگوریتم کلاس‌بندی باعث کم شدن بار محاسباتی الگوریتم می‌شود. پروسه کلاس‌بندی را می‌توان در هر T_c انجام داد که T تعداد نهایی تکرارها می‌باشد. بعد از اجرای الگوریتم کلاس‌بندی، پروسه کم کردن و مشخص کردن و ذرات در هر *Niche* اعمال می‌گردد. در روش پیشنهادی برای این منظور از الگوریتم سیستم ایمنی مصنوعی استفاده می‌شود.

انجام دهنده و بدین شکل کل فضای مسئله جستجو می‌شود.

۴- استفاده از الگوریتم k -means

الگوریتم گروه ذرات بهینه‌ساز :

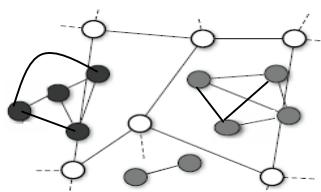
الگوریتم‌هایی که در بخش قبلی توضیح داده شده در مکانیابی راه حل‌های بهینه خیلی راضی کننده نیستند، هر چند که به عنوان نقاط شروع الگوریتم‌های قابل قبولی هستند. در این بخش روشی ارائه خواهد شد که در آن از الگوریتم‌های کلاس‌بندی به منظور تعیین *Niche*‌ها در الگوریتم گروه ذرات بهینه‌ساز استفاده می‌شود. این متد اولین بار در سال ۲۰۰۸ رأیه شد [3]. ایده اصلی این روش پذیرش ساختار همسایگی به صورت پویا است تا *Niche*‌های متفاوت در گروه تشکیل شوند. در واقع همسایگی هر ذره به همان کلاسی که ذره در آن قرار دارد، محدود می‌شود. در حالت استاندارد در الگوریتم گروه ذرات بهینه‌ساز، ذرات به سمت بهترین تجربه شخصی و بهترین تجربه در بین ذرات همسایه گرایش دارند. جایگزین کردن این گرایش به مرکز ثقل هر *k*-means *Niche* که توسط الگوریتم کلاس‌بندی تعیین می‌شود [13]، به الگوریتم گروه ذرات بهینه‌ساز این امکان را می‌دهد که در هر کلاس یک جستجوی محلی را اجرا کرده و در نتیجه در کل فضای جستجو، بهینه‌های متفاوت را مکانیابی کند. الگوریتم k -means با توجه به بهترین مکان گذشته یا بهترین تجربه خود ذرات، ذرات را کلاس‌بندی می‌کند. بعد از کلاس‌بندی ذرات ساختار همسایگی نیز تغییر خواهد کرد. بر اساس این تغییر همسایگی، هر ذره در هر کلاس فقط با ذراتی که در همان کلاس یا *Niche* هستند و بر اساس توپولوژی *gbest* قادر به تبادل اطلاعاتی است. الگوریتم کلاس‌بندی به منظور تعیین *Niche*‌ها در زمان‌های مشخص اجرا می‌شود و در فاصله هر بار اجرای آن الگوریتم گروه ذرات بهینه‌ساز در هر *Niche* اجرا می‌شود. ذرات در *Niche*‌های متفاوت در هر مرحله از اجرا می‌توانند وارد یک *Niche* مشابه شوند با توجه به اینکه به سمت یک بهینه محلی یکسان حرکت کنند و بر عکس می‌توانند به دو *Niche* متفاوت تقسیم شوند چرا که بعضی از ذرات در این *Niche* در طول اجرا به سمت بهینه دیگری در حال حرکت می‌باشند. تفاوت این روش با سایر متدهای Niching در الگوریتم گروه ذرات بهینه‌ساز این حقیقت

۵- سیستم ایمنی مصنوعی:

در ادامه از مفهوم سیستم ایمنی مصنوعی به منظور مشخص کردن ذرات هر *Niche* استفاده می‌شود. در واقع سیستم ایمنی مصنوعی از پرجمعیت و شلوغ شدن *Niche*‌ها جلوگیری می‌کند. قبل از ارائه عملکرد سیستم ایمنی مصنوعی در متد پیشنهادی، توضیحاتی درباره این الگوریتم داده می‌شود. سیستم ایمنی مصنوعی یک روش هوشمند محاسباتی جدید ارائه کرده، که از سیستم ایمنی بدن الهام می‌گیرد و دقیقاً از مدل‌های طبیعی برای حل مسائل در زمینه‌های متفاوت استفاده می‌کند [8]. علاوه بر این سیستم ایمنی مصنوعی می‌تواند در کنار سایر الگوریتم‌های محاسباتی از جمله الگوریتم‌های تکاملی به کار رود و باعث قدرتمندتر شدن این الگوریتم‌ها شود. در روش پیشنهادی از تئوری انتخاب منفی استفاده شده است [7]. این تئوری براساس این سوال که "سیستم ایمنی چگونه رفتار می‌کند زمانیکه با یک آنتی زن خودی مواجه می‌شود؟" می‌باشد.

جواب این سوال خیلی پیچیده است و از مکانیزم‌های متفاوتی برای تشخیص *B-Cell*‌ها از *T-Cell*‌ها استفاده می‌کند. در این مقاله توجه و تمرکز بر روی

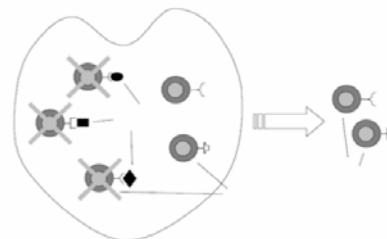
ذرات باقیمانده، که از طریق الگوریتم سیستم اینمنی مصنوعی از کلاس‌های پرجمعیت حذف شده‌اند، دوباره مقداردهی اولیه می‌شوند و به منظور گسترش جستجو در فضای مسئله مورد استفاده قرار می‌گیرد. همانطور که در شکل ۲ نشان داده شده است، بعد از اجرای الگوریتم سیستم اینمنی مصنوعی، ذراتی که از Niche شده‌اند و هیچ ارتباطی با هم ندارند، بر اساس تپولوژی ون-نیومن سازماندهی می‌شوند (ذراتی که با رنگ سفید مشخص شده‌اند) و در همسایگی یکدیگر قرار می‌گیرند. در نتیجه کارآیی جستجو در فضای مسئله برای پیدا کردن راه حل‌های جدید بالاتر می‌رود و به الگوریتم کمک می‌کند تا تمامی راه حل‌های بهینه مسئله را پیدا کند. استفاده از الگوریتم k -means مجموعه‌ای از پارامترهای جدید از جمله c ، r_k و k را اضافه می‌کند.



شکل ۲: مشخص کردن همسایگی gbest در میان ذرات غیر خودی (ذرات با رنگ تیره) و همسایگی ون-نیومن در بین ذرات خودی (ذرات با رنگ سفید).

پارامتر k که همان تعداد کلاس‌ها است معمولاً بین ۲ تا $N/2$ در حال تغییر است. بدین شکل که در ابتدای $N/2$ است. در مراحل بعدی هرگاه کلاس خالی از ذرات شود یا فاصله دو کلاس از یکدیگر (بر اساس فاصله اقلیدسی) کمتر از ϵ شود و این دو کلاس با هم ترکیب می‌شوند و تعداد کل کلاس‌ها کم می‌شود و بر عکس اگر تعدادی از ذرات یک کلاس به سمت بهینه دیگری گرایش داشته باشند آن دو کلاس از هم مجزا می‌شوند و تعداد کلاس‌ها افزایش می‌یابد. N تعداد ذرات اولیه است. پارامتر c تعداد اجرای الگوریتم گروه ذرات بهینه ساز نیز، الگوریتم گروه ذرات بهینه ساز با فاکتور محدود کننده است. همسایه‌های هر ذره شامل همه ذراتی که در کلاس مشابه هستند می‌باشد. پس k تا (به تعداد کلاس‌ها) Niche تشکیل می‌اینکه الگوریتم k -means به شدت وابسته به مقداردهی اولیه کلاس‌ها است، باید به تعداد r_k بار اجرا شود تا متغیر بودن آن را کاهش دهد. بهینه‌ترین کلاس‌بندی، کلاس‌بندی است که کمترین میزان مربع خطأ را دارد:

انتخاب منفی T-Cell از Thymic ها است که در شکل ۱ نشان داده شده است.



شکل ۱: نمایی ساده از انتخاب منفی Thymic

این پروسه مسئول حذف تمام T-Cell هایی است که گیرنده‌هایشان و توسط آنتی‌ژن‌های خودی موجود در Thymus کور می‌شود.

بعد از اجرای هر بار الگوریتم کلاس‌بندی در هر Niche میانگین شایستگی ذرات درون همان Niche محاسبه می‌شود. بر اساس این پارامتر آنتی‌ژن‌های خودی و غیر خودی از هم متمایز می‌شوند. بدین شکل که اگر شایستگی هر ذره از میانگین شایستگی ذرات همان Niche بزرگ‌تر باشد، ذره غیر خودی و در غیر اینصورت ذره خودی تشخیص داده می‌شود. همانطور که مشخص است آنتی‌ژن‌های غیر خودی اجازه ورود به گام بعدی از کلاس‌بندی را ندارند. این ذرات همان ذراتی هستند که از کلاس‌های شلوغ و پرجمعیت حذف شده‌اند. این ذرات در گام بعدی دوباره مقداردهی اولیه می‌شوند تا بتوانند مناطق دیگری از فضای مسئله را جستجو کنند. در واقع ذراتی که توسط الگوریتم سیستم اینمنی مصنوعی، جزء آنتی‌ژن‌های غیر خودی تشخیص داده می‌شوند جدا از سایر ذرات و تحت شرایطی مختلف رفتار می‌کنند. تپولوژی همسایگی در این ذرات از نوع ون-نیومن است و الگوریتم گروه ذرات بهینه‌ساز نیز، الگوریتم گروه ذرات بهینه ساز با فاکتور محدود کننده است. همسایه‌های هر ذره شامل همه ذراتی که در کلاس مشابه هستند می‌باشد. پس k تا (به تعداد کلاس‌ها) Niche تشکیل می‌شود که ذرات در آن به طور کامل متصل هستند. بدین شکل تپولوژی gbest در هر Niche فراهم می‌شود.

$$J = \sum_j \sigma_j \quad (4)$$

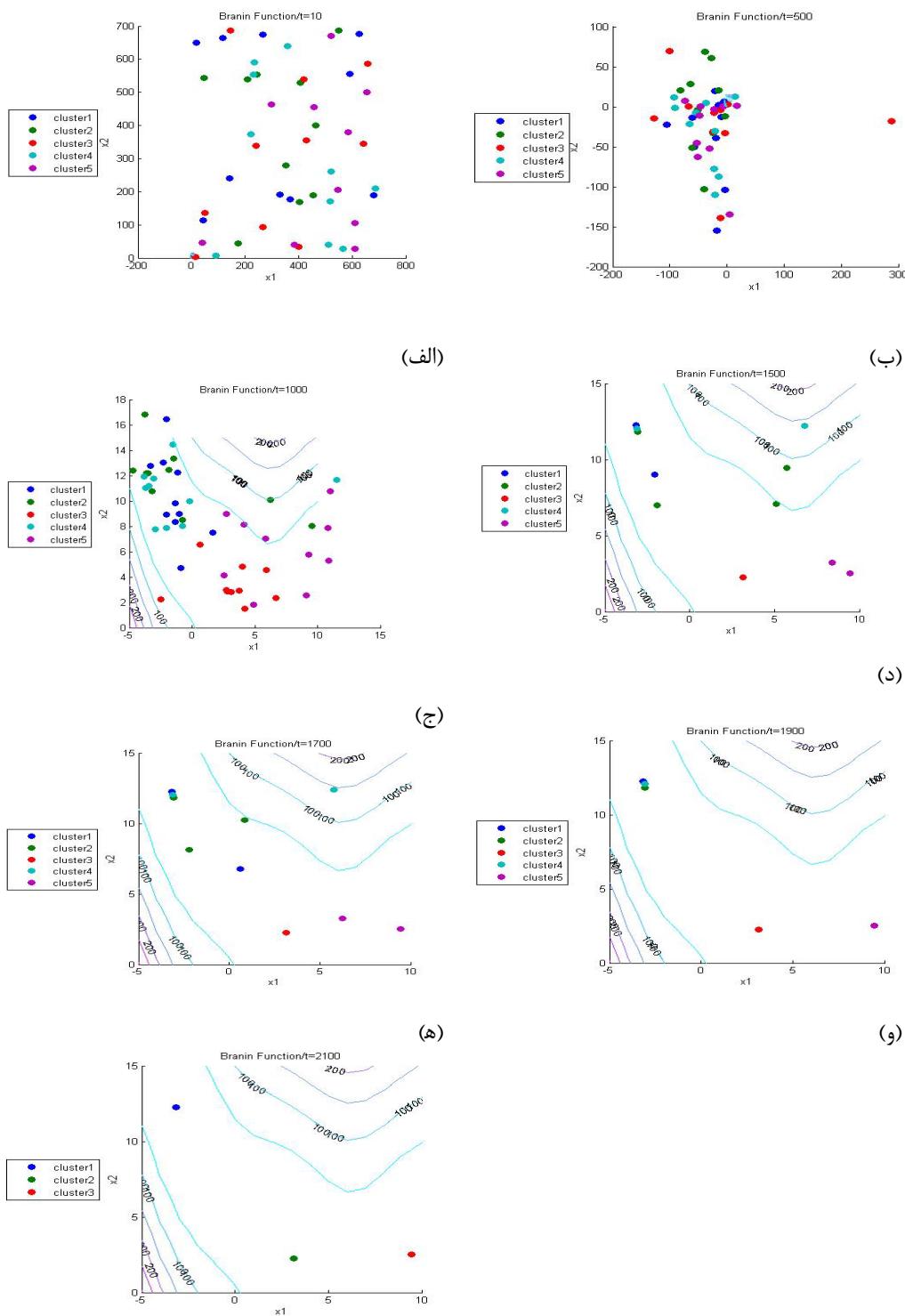
$$\sigma_j^2 = \frac{1}{N_j - 1} \sum_{p \in c_j} \|p - m_j\|^2 \quad (5)$$

ذرات به طور تصادفی در کل فضای جستجو توزیع شده‌اند. بعد از اجرای چند مرحله از الگوریتم گروه ذرات بهینه ساز، این ذرات به گروه‌های متفاوت تقسیم می‌شوند (البته این گروه‌ها همگی متعلق به فضای جستجو هستند). زمانیکه اولین کلاس‌بندی اجرا می‌شود ۵ کلاس مشخص می‌شوند که مشخص کننده Niche‌ها هستند. که ۳ تا از این کلاس‌ها نمایانگر ۳ بهینه عمومی تابع مورد بررسی است. در ادامه و همانطور که در شکل ۳ قسمت (و) دیده می‌شود، الگوریتم دقیقاً به سه کلاس همگرا شده است. در شکل ۴ نیز این روند قابل مشاهده است. چنین همگرایی هدف نهایی الگوریتم KPSO نیست: از زمانیکه که تعداد قابل قبولی از کلاس‌ها به منظور پوشش دادن تمامی بهینه‌ها شکل گرفته‌اند، حضور کلاس‌های نادرست و اضافی هیچ ضرری برای همگرایی الگوریتم نداشته است. در کنار این امر در مراحل اولیه الگوریتم، حضور کلاس‌های اضافی می‌تواند در مکانیابی مناطق جدید در فضای جستجو مفید باشد. در جدول ۱ نتایج مقایسه‌ای که روی ۳ تابع تست و الگوریتم‌های KPSO، SPSO و ANPSO انجام شده، ارائه گردیده است. در آزمایشات انجام گرفته اندازه جمعیت ۳۰ ذره، در نظر گرفته شده است. هر اجرایی، ۲۵ بار تکرار شده است. تعداد تکرار الگوریتم پیشنهادی در هر بار اجرا با توجه به همگرا شدن این متدها مشخص می‌شود. هدف از مجموعه آزمایشات، نشان دادن کارآیی بهتر و سرعت بالاتر الگوریتم پیشنهادی نسبت به روش‌های Niching PSO موجود از حمله KPSO، SPSO و ANPSO بود. اما اشاره به این نکته لازم است که در روش پیشنهادی پارامتری تحت عنوان C، تعداد گامهای اجرا بین دو الگوریتم کلاس‌بندی، اضافه شده است. در آزمایشات انجام گرفته این پارامتر برابر ۱۰ و به صورت ثابت در نظر گرفته شده است.

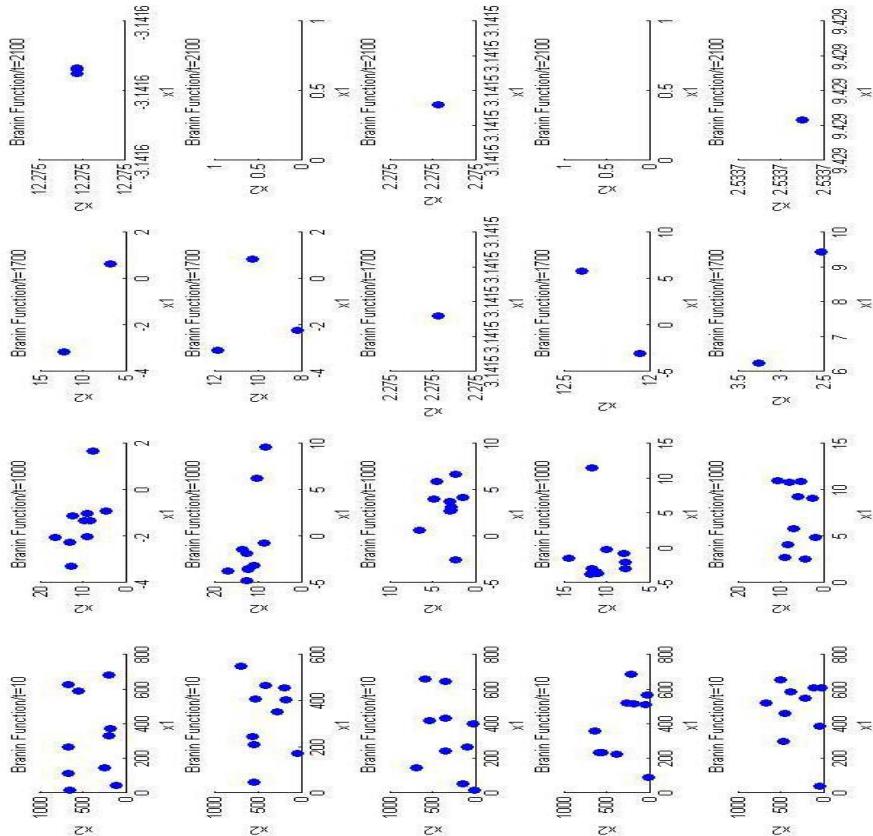
جاییکه j واریانس کلاس j و m_j ، مرکز کلاس j است و $j \in N$ سایز آن. نتایج به دست آمده نشان می‌دهد که $r_k = 10$ می‌تواند نتایج خوبی را فراهم کند. با توجه به اینکه الگوریتم پایه در این متدهای پیشنهادی الگوریتم گروه ذرات بهینه ساز با فاکتور محدود کننده است پارامترهای مشخص می‌شوند که مشخص کننده Niche‌ها هستند. $P_{incr} = g_{incr} = 0.73$ و $\chi \approx 2.05$ نیز اضافه شده است.

۶- نتایج شبیه‌سازیها:

در این مقاله یک نسخه از الگوریتم گروه ذرات بهینه ساز برای بهینه کردن توابعی که بیش از یک بهینه دارند، ارائه شد. در این الگوریتم از کلاس‌بندی استاندارد k-means در کنار الگوریتم سیستم ایمنی مصنوعی استفاده شد. در این الگوریتم پیشنهادی، از سیستم ایمنی مصنوعی به منظور مشخص کردن ذرات هر Niche. که خود توسط الگوریتم کلاس‌بندی مشخص شده‌اند، استفاده کردیم. به منظور بررسی کارآیی این متدهای مقایسه‌ای با متدهای دیگر Niching PSO از جمله KPSO، SPSO و ANPSO که متدهای اولیه Niching گروه ANPSO ذرات بهینه ساز بوده‌اند، انجام گرفته است. نتایج نشان دهنده کارآیی بهتر و سرعت بالاتر این متدهای نسبت به روش‌های مذکور است. این مقایسه روی سه تابع تست انجام گرفته است. لیست این توابع به همراه نتایج این مقایسات در جدول ۱ ارائه شده است. اولین تابع Branin (دو بعد با ۳ بهینه عمومی و بدون بهینه محلی)، دومین تابع RCOS، Six-Hump Camel Back (۲ بهینه Himmelblau عمومی و ۴ بهینه محلی) و تابع سوم Branin بهینه محلی (دو بعد با ۴ بهینه عمومی، و بهینه محلی ندارد) است. به منظور نشان دادن سرعت و کارآیی بالاتر الگوریتم پیشنهادی در همگرا شدن به تمامی راه حل‌های بهینه نسبت به روش‌های مشابه، در شکل ۳ چندین مرحله از اجرای این الگوریتم روی تابع Branin به نمایش درآمده است. همانطور که در شکل اول مشخص است در گام اول



شکل ۳: توزیع ذرات در فضای جستجو در تابع Branin (الف) تکرار ۱۰، (ب) تکرار ۵۰، (ج) تکرار ۱۰۰، (د) تکرار ۱۵۰، (ه) تکرار ۱۷۰، و (و) تکرار ۱۹۰.



شکل ۴: روند توزیع درات در Niche ها و خالی شدن دو عدد از Niche ها روی تابع Branin

جدول ۱: تعداد تکرار ارزیابی مورد نیاز برای پیدا کردن تمامی راه حل های عمومی

تابع	تعداد ذرات	SPSO	ANPSO	PSO K	PSO K با سیستم ایمنی مصنوعی
Branin ROC	۳۰	۳۱۶۹±۶۹۲	۳۳۲۲±۵۲۰	۲۰۱۴±۴۴۰	۱۹۱±۴۷۱
Six-Hump camel back	۳۰	۲۸۷۲±۸۱۷	۲۷۹۱±۸۰۷	۱۱۲۴±۲۱۶	۹۱۶±۲۸۴
Himmelblau	۳۰	۴۰.۹۶±۲۰.۱۸	۱۶۳۰.۱±۱۳۱۵۷	۲۲۵۹±۵۳۱	۱۷۸۹±۶۰۳

- [7] G. J. V. Nossal, "Negative selection of lymphocytes", *cell* 76, pp. 229-239, 1994.
- [8] S. A. Hofmeyr, and S. Forrest, "Immunity by design: An artificial immune system", *In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, vol. 2, pp. 1289-1296, 1999.
- [9] R. Brits, "Niching strategies for particle swarm optimization", *Master of science in the faculty of Natural & Agricultural science*, university of Pretoria, Nov. 2002.
- [10] K. E. Parsopoulos and M. N. Vrahatis, "Modification of the particle swarm optimizer for locating all the global minima", *In Artificial Neural Network and Genetic Algorithms, Computer Science*, pp. 324-327, Springer, Wien, Austria, 2001.
- [11] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas, and M. N. Vrahatis, "Stretching technique for obtaining global minimizers through particle swarm optimization", *in Proceedings of the Particle Swarm Optimization Workshop*, pp. 22-29, Indianapolis, Ind, USA, April 2001.
- [12] A. P. Engelbrecht, B. S. Masiye, and G. Pampara, "Niching ability of basic particle swarm optimization algorithms", *in Proceedings of the IEEE Swarm Intelligence Symposium (SIS '05)*, pp. 397-400, Pasadena, Calif, USA, June 2005.
- [13] R. Xu and D. Wunsch II, "Survey of clustering algorithms", *IEEE Transactions on Neural Networks*, vol. 3, no. 16, pp. 645-678, 2005.
- [14] S. Bird and X. Li, "Adaptively choosing niching parameters in a PSO," *in Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO '06)*, vol. 1, pp. 3-10, ACM Press, Seattle, Wash, USA, July 2006.
- [15] D. E. Goldberg, J. Richardson, "Genetic algorithm with sharing for multimodal function optimization", *in Proceedings of the second International Conference on Genetic Algorithms*, pp. 41-49, 1987.
- [16] J. Horn, "The nature of niching: genetic algorithms and the evolution of optimal, cooperative population", *Doctoral dissertation 95001*, Urbana, University of Illinois, Illinois Genetic algorithm Lab, 1997.
- [17] S. W. Mahfoud, "Niching methods for genetic algorithms", *IlliGAL Rep. 95001*, Urbana, University of Illinois, Illinois Genetic algorithm Lab, 1995.
- [18] G. Harik, "Finding multiple solutions using restricted tournament selection", *in L. J. Eshelman(Ed.), Proceedings of the sixth international Conference on Genetic Algorithms*, pp. 24-31, San Francisco: Morgan Kaufman, 1995.
- [19] R. Brits, A. P. Engelbercht, and F. van den Bergh, "A niching particle swarm optimization", *in Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL '02)*, Vol. 2, pp. 692-696, Singapore, Nov. 2002.
- [20] A. P. Engelbercht and F. van den Bergh, "A new locally convergent particle swarm optimizer", *in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '02)*, vol. 3, pp. 96-101, Hammamet, Tunisia, October 2002.

۷- نتیجه‌گیری:

در این مقاله یک نگرش جدید از تکنیک Niching در الگوریتم گروه ذرات بهینه‌ساز و مبتنی بر کلاس‌بندی ارائه شد. در این متدهای جدید Niching ها توسط الگوریتم k-means کلاس‌بندی استاندارد شده و ذرات هر نیچ نوشیت الگوریتم سیستم اینمنی مصنوعی Niching گردید. البته لازم به ذکر است که تمامی تلاش ما بر این بوده است که ساختار اصلی الگوریتم گروه ذرات بهینه ساز حفظ شود. در واقع تکنیک Niching ساختار همسایگی ذرات را تحت تاثیر قرار می‌دهد و در رفتار شخصی هر ذره در هر کلاس بی تاثیر است. علاوه بر این بعد از مشخص شدن زیر گروه‌ها، در هر زیر گروه الگوریتم گروه ذرات بهینه ساز اجرا می‌شود و در همان منطقه از فضای مسئله، یک جستجوی محلی را انجام می‌دهد. روش پیشنهادی قابل رقابت با روش KPSO همچنین سایر روش‌های Niching در الگوریتم گروه ذرات بهینه ساز از SPSO و ANPSO استفاده از سیستم اینمنی مصنوعی در این متدها و با توجه به مشخص کردن ذرات هر کلاس و حذف ذرات با شایستگی کمتر، باعث بالاتر رفتن سرعت الگوریتم شده است.

مراجع

- [1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization", *in Proceedings of IEEE International Conference on Neural Networks (ICNN '95)*, vol. 4, pp. 1942-1948, IEEE Service Center, Perth, Western Australia, Nov. 1995.
- [2] S. W. Mahfoud, "Niching method for genetic algorithms", *PHD thesis*, University of Illinois at Urbana-Champaign, Champaign, Ill, USA, 1995.
- [3] A. Passaro and A. Starita, "Particle swarm optimization for multimodal functions: a clustering approach", *Journal of the Artificial Evolution and Applications*, vol. 2008, No. 482032, February 2008.
- [4] I. L. Schoeman and A. P. Engelbrecht, "A parallel vector-based particle swarm optimizer," *in Proceedings of the 7th International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA '05)*, Coimbra, Portugal, March 2005.
- [5] X. Li, "Adaptively choosing neighborhood bests using species in a particle swarm for multimodal function optimization", *in Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO '04)*, vol. 3102 of Lecture Notes in Computer Science, pp. 105-116, Springer, Seattle, Wash, USA, June 2004.
- [6] D. Beasley, D. R. Bull, and R. R. Martin, "A sequential niche technique for multimodal function optimization", *Evolutionary Computation*, vol. 1, no. 2, pp. 101-125, 1993.

[21] I. L. Schoeman and A. P. Engelbrecht, "Using vector operations to identify niches for particle swarm optimization," in *Proceedings of the IEEE International Conference on Cybernetics and Intelligent Systems (CCIS '04)*, vol. 1, pp. 361-366, Singapore, December 2004.

[22] J. P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization", *Evolutionary Computation*, Vol. 10, no. 3, pp. 207-234, 2002. J. R. Beveridge", segmenting images using localizing histograms and region merging", *Int.J.of Compt. Vision.* vol. 2, 1989.

[23] K. A. de Jong, "An analysis of the behavior of a class of genetic adaptive systems", *PhD thesis*, Dept. of Computer and Communication Sciences, University of Michigan, 1975.