

الگوریتم ممیزی و خوشه‌بندی فازی در محیط‌های پویا

مرتضی علی زاده^۱، محمد رضا میبدی^۲، علیرضا رضوانیان^۲

^۱ دانشگاه آزاد اسلامی، واحد قزوین، گروه مهندسی کامپیوتر و فناوری اطلاعات، قزوین، ایران m_alizadeh@qiau.ac.ir

^۲ آزمایشگاه محاسبات نرم، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر تهران، ایران mmeybodi@aut.ac.ir

^۲ آزمایشگاه محاسبات نرم، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر تهران، ایران a.rezvanian@aut.ac.ir

چکیده - بر این اساس که اکثر الگوریتم‌های تکاملی برگرفته از طبیعت می‌باشند و با در نظر گرفتن این مسئله که در دنیای واقعی محیط، پیوسته در حال تغییر است، نیاز به روشی است که بتوان این الگوریتم‌های تکاملی را در محیط‌های پویا و در حال تغییر پیاده‌سازی و آزمایش کرد. هدف این مقاله دستیابی به جواب بهینه برای تابع قله‌های متحرک با استفاده از الگوریتم بهینه‌سازی ممیزی می‌باشد. برای دستیابی به این امر، از یک نمونه جدید چند جمعیتی در کنار الگوریتم ازدحام ذرات به همراه رویکرد ممیزی برای مسئله قله‌های متحرک استفاده شده است و جهت بهتر شدن کارایی الگوریتم، از روش خوشه‌بندی ذرات به همراه فازی سازی پارامتر تعلق به گروهها استفاده شده است. ارزیابی نتایج در شرایط استاندارد نشان می‌دهد که این مدل توانایی ردیابی توابع چند قله‌ای را داشته و عملکرد آن با در نظر گرفتن جلوگیری از همگرایی، با به کارگیری انتخاب سر خوشه‌ها با استفاده از پارامتر متعلق بودن به گروه‌ها ارتقا پیدا کرده است.

کلید واژه - محیط‌های پویا، قله‌های متحرک، بهینه‌سازی ممیزی و خوشه‌بندی فازی

۱- مقدمه

خوشه‌بندی فازی، برای محیط‌های پویا ارائه شده است. رویکرد ابتکاری در این مقاله، استفاده از خوشه‌بندی فازی ذرات و پیوند دادن و تبدیل این ذرات به جمعیت برای ترکیب با الگوریتم ممیزی مبتنی بر ازدحام ذرات می‌باشد. در این مقاله ابتدا در بخش دوم در مورد الگوریتم ممیزی و سپس خوشه بندی و در بخش چهارم ویژگی‌های محیط پویا بحث شده است، در بخش پنجم الگوریتم پیشنهادی مطرح شده و مسئله قله‌های متحرک به عنوان تابع محکی برای شبیه‌سازی محیط‌های پویا معرفی شده است. نتایج آزمایش‌ها و مقایسه با روش‌های معروف دیگر هم برای بهینه‌سازی محیط‌های پویا در بخش آخر ارائه شده است.

۲- الگوریتم ممیزی

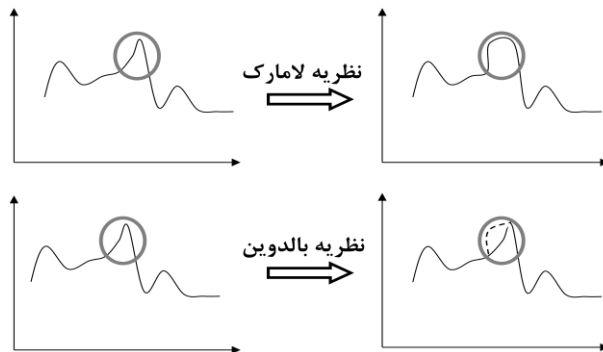
الگوریتم‌های ژنتیکی همانند سایر الگوریتم‌های مکاشفای (مانند بهینه‌سازی گروه ذرات) در گام‌های نخست اجرای الگوریتم ناحیه‌هایی از فضای حالت مسئله که بهینه‌های سراسری و محلی در آن واقع شده‌اند را به خوبی شناسایی می‌کنند اما در ادامه مسیرشان به سمت بهینه سراسری بسیار کند عمل می‌نمایند [۴].

در مسائل بهینه‌سازی پویا، الگوریتم‌های تکاملی به سادگی قابل استفاده نیستند و کارایی لازم را مانند محیط‌های ایستا ندارند. از مشکلات اصلی این الگوریتم‌ها در محیط‌های پویا، می‌توان به حافظه غیر معتبر و از دست رفتن تنوع اشاره نمود. همچنین از آنجاکه اکثر روش‌های پردازش تکاملی به دلیل ماهیتشان به یک نقطه همگرا می‌شوند، لذا تنوع در محیط از بین می‌رود و در صورت تغییر در محیط همگرا شدن به نقطه بهینه جدید، در صورت امکان بسیار زمان گیر است. در بین کارهای انجام شده برای بهینه‌سازی محیط‌های پویا می‌توان به تلاش‌های فراوان یانگ و شاگردانش اشاره نمود که تحقیقات گسترده‌ای بر روی الگوریتم‌های ممیزی و محیط‌های پویا انجام داده‌اند [۱][۲][۳]. استفاده از ترکیبی از الگوریتم‌های ممیزی و الگوریتم‌های تپه نوردی و ازدحام ذرات از کارهای اخیرشان به حساب می‌آید. در این مقاله یک الگوریتم ممیزی مبتنی بر بهینه‌سازی ازدحام ذرات چند جمعیتی با

نیز کد می‌شود. این اعتقاد مبنای رویکرد لامارک است. در این رویکرد پس از آن که در همسایگی یک عضو جمعیت جستجوی محلی صورت گرفت، بهترین همسایه جایگزین آن عضو می‌شود، به این ترتیب بهبود در شایستگی اعضاء جمعیت هم توسط عملگرهای ژنتیک و هم توسط عملگر ممیتیک (جایگزینی بهترین همسایه) صورت می‌گیرد [۵].

۲-۲- الگوریتم‌های ممیتیک مبتنی بر نظریه بالدوین

بعضی از دیگر صاحب نظران معتقدند دلایل کافی برای این مدعا که آموخته‌های موجود در ژن او کد می‌شود وجود ندارد و آموخته‌های موجود تنها در طول حیات خود او در اختیار هستند و فقط توسط آموزش (تقلید) به موجود دیگر انتقال می‌یابند، بنابراین در رویکرد بالدوین پس از آن که در همسایگی یک عضو جمعیت جستجوی محلی صورت گرفت، شایستگی بهترین همسایه یک عضو جمعیت را جایگزین شایستگی آن عضو می‌نمایند. در واقع در این روش یک عضو جمعیت در تجربه همسایگانش سهیم می‌شود به همین علت به آن اثر بالدوین نیز می‌گویند [۵].



شکل (۱): نمای نظریه‌ها در دو بعد

در شکل ۱ قسمت بالا و نظریه لامارک جمعیت و ارزش آنها جایگزین شده‌اند ولی در نظریه بالدوین فقط ارزش جمعیت همسایه برابر بهینه محلی می‌باشد.

۳- خوشه‌بندی فازی

خوشه‌بندی یکی از شاخه‌های یادگیری بدون نظارت می‌باشد و فرآیند خودکاری است که در طی آن، نمونه‌ها به دسته‌هایی که اعضای آن مشابه یکدیگر می‌باشند تقسیم می‌شوند که به این دسته‌ها خوشه^۱ گفته می‌شود. بنابراین خوشه مجموعه‌ای از اشیاء می‌باشد که در آن اشیاء با یکدیگر مشابه بوده و با اشیاء موجود در خوشه‌های دیگر غیر مشابه می‌باشند. برای مشابه بودن می‌توان

عیب دیگری که این الگوریتم‌ها با آن مواجه می‌باشند عدم پایداری این الگوریتم‌ها است. به این معنی که کیفیت پاسخ‌هایی که از اجراهای مختلف الگوریتم به دست می‌آید ممکن است تفاوت‌های بسیاری داشته و حتی غیرقابل اعتماد باشند. از بین روش‌هایی که برای برطرف ساختن مشکلات متاهیوریستیک‌ها ارائه شده است، استراتژی ترکیبی جایگاه ویژه‌ای دارند [۵][۶].

استراتژی ترکیبی از به کارگیری تکنیک‌های مختلف در فرایند حل مسئله حاصل می‌گردد. الگوریتم‌های ممیتیک از مشهورترین اعضاء این خانواده به شمار می‌آیند که از پیوندسازی الگوریتم‌های ژنتیک با یک هیوریستیک جستجوی محلی مانند روش تپه‌نوردی حاصل می‌گردد [۶][۷]. در این الگوریتم‌ها یک عملگر جستجوی محلی پس از عملگرهای ژنتیکی شایستگی را بهبود می‌بخشد (تقلید) و پس از آن اعمال ارزیابی و جایگزینی جمعیت صورت می‌گیرد.

الگوریتم‌های ممیتیکی در طی اجرا، به علت بهبود و تصحیح محلی موثری که بر روی ژن‌ها انجام می‌دهند، یک ظرفیت جستجو با قدم کوچک قوی ارائه می‌کنند، اما ممکن است ظرفیت جستجو با قدم بزرگ را از دست بدهند مانند همگرا شدن جمعیت به یک بهینه که باید در محیط‌های پویا اجتناب شود. بنابراین، این می‌تواند یک ایده پژوهشی، جالب برای امتحان کارایی الگوریتم‌های ممیتیکی که با روش‌های تنوعی مناسب، بهبود یافته‌اند در محیط‌های پویا باشد. از این رو جستجوی محلی (مم) وابسته به مسئله است. بنابراین این‌که چگونه عملگرها مم بهینه را بیابیم و از استفاده از روش‌های جستجوی محلی نامناسب جلوگیری کنیم، یکی از موضوعات بسیار مهم شده است.

هر چند نری و دیگران، یک الگوریتم ممیتیکی با مم چندگانه با یک روش غیر رقابتی پیشنهاد کردند که روش‌های جستجوی محلی متفاوت در طی دوره‌های تکاملی جمعیت‌های متفاوت می‌تواند فعال گردد. [۸]

در الگوریتم ممیتیک جستجوهای ابتکاری محلی با الگوریتم ژنتیک ترکیب می‌شوند تا در زمان کمتر نتایج بهتری به دست آید. در ادامه مفهوم‌های ممیتیک و تفاوت‌های آن با دیگر الگوریتم‌ها و کاربردش در محیط‌های پویا بحث می‌کنیم.

۲-۱- الگوریتم‌های ممیتیک مبتنی بر نظریه لامارک

برخی از متخصصان معتقدند آنچه که یک موجود در طول زندگی از محیط و جامعه خود می‌آموزد در ساختار ژنی آن موجود

منظور از $Dis_{i,x}$ فاصله ذره از مرکز خوشه و $Dep_{i,x}$ وابستگی ذره به آن خوشه می‌باشد. هر چه فاصله ذره از مرکز خوشه بیشتر باشد وابستگی ذره به آن خوشه کمتر و همین طور بلعکس.

۴- محیط پویا و قله‌های متحرک

تا حالا، دهها نوع متفاوت از مسائل پویا برای آزمایش الگوریتم‌های تکاملی مورد استفاده قرار گرفته اند. از توابع ساده ریاضی تا انواع مسائل زمانبندی برای کاربردهایی در مسائل مصنوعی شبیه به واقعیت. یک تابع محک، معروف به نام قله‌های متحرک [۱۰] برای محیط‌های پویا وجود دارد. در این تابع محک قله‌ها بطور پیوسته با زمان در حال تغییراند. در شکل ۲ نمونه ای از این تابع محک آمده است. حال موضوع مهم این است، بهینه را در این تغییرات پیدا نمود. تابع قله‌های متحرک شامل یک فضای چند بعدی، چندین قله با ارتفاع و پهنای متفاوت است، که ارتفاع، پهنای و موقعیت هر قله در طول زمان در حال تغییر است. تابع چند قله‌ای متحرک در نظر گرفته شده شامل m قله در n بعد یا پارامترهای حقیقی است و هدف، یافتن بیشینه در هر زمان در میان m قله تا تغییر بعدی می‌باشد که به صورت رابطه (۱) تعریف می‌شود:

$$F(\vec{x}, t) = \text{Max}(B(\vec{x}), \text{Max}_{i=1..m} P(\vec{X}_i, H_i(t), W_i(t), \vec{p}_i(t))) \quad (1)$$

در این تابع $B(\vec{x})$ یک شمای پایه ثابت در زمان و p تابع چند قله-ای است که در هر Δe ارزیابی، H و W ارتفاع و پهنای قله‌ها بوسیله افزودن یک عدد تصادفی گوسین با میانگین صفر و واریانس σ و مکان هر قله بوسیله افزودن بردار v یا طول ثابت s (سختی) طبق رابطه (۲) تغییر می‌کند. در این تابع می‌توان پیچیدگی را با افزایش و کاهش Δe در زمان تغییر داد [۱۰].

$$\begin{cases} \sigma \in N(0,1) \\ hi(t) = hi(t-1) + \text{height_severity} \cdot \sigma \\ Wi(t) = Wi(t-1) + \text{width_severity} \cdot \sigma \\ \vec{p}_i(t) = \vec{p}_i(t-1) + \vec{v}_i(t) \end{cases} \quad (2)$$

بردار v را می‌توان وابسته به تغییر قبلی آن ایجاد کرد که در این صورت تغییر موقعیت قله‌ها همسو با تغییرات قبل آن می‌شود و یا به صورت تصادفی آنرا ایجاد نمود که موجب می‌شود موقعیت قله‌ها به صورت تصادفی تغییر کند و هیچگونه وابستگی به تغییر قبلی نداشته باشد [۱۰].

معیارهای مختلفی را در نظر گرفت مثلاً می‌توان معیار فاصله را برای خوشه بندی مورد استفاده قرار داد و اشیائی را که به یکدیگر نزدیکتر هستند را بعنوان یک خوشه در نظر گرفت که به این نوع خوشه بندی، خوشه بندی مبتنی بر فاصله نیز گفته می‌شود.

خوشه بندی فازی چیست و فرق آن با مدل غیر فازی در چیست؟ جواب این سوال این است که هدف خوشه بندی یافتن خوشه‌های مشابه از اشیاء در بین نمونه‌های ورودی می‌باشد. در واقع در اینجا با مفهوم عدم قطعیت^۲ مواجه هستیم. ما خودمان نیز از عدم قطعیت در زندگی روزمره بارها استفاده کرده ایم مثلاً هوای سرد، آب داغ و غیره. در واقع تمامی مثالهای بالا مثالهایی از مجموعه‌های فازی می‌باشند. تفاوت اصلی مجموعه‌های فازی و مجموعه‌های کلاسیک در این است که تابع تعلق مجموعه‌های فازی دو مقداری نیست (۰ یا ۱) بلکه می‌تواند هر مقداری بین ۰ تا ۱ را اختیار کند. در خوشه بندی کلاسیک هر نمونه ورودی متعلق به یک و فقط یک خوشه می‌باشد و نمی‌تواند عضو دو خوشه و یا بیشتر باشد. به زبان دیگر خوشه‌ها همپوشانی ندارند. حال حالتی را در نظر بگیرید که میزان تشابه یک نمونه با دو خوشه و یا بیشتر یکسان باشد در خوشه بندی کلاسیک باید تصمیم‌گیری شود که این نمونه متعلق به کدام خوشه است. تفاوت اصلی خوشه بندی کلاسیک و خوشه بندی فازی در این است که یک نمونه می‌تواند متعلق به بیش از یک خوشه باشد. اگر نمونه‌های ورودی بصورت دقیق در دو خوشه باشد، بطوری‌که بتوان داده‌ها را به دو خوشه تقسیم کرد اما مشکلی که پیش می‌آید این است که داده مشخص شده در وسط می‌تواند عضو هر دو خوشه باشد بنابراین باید تصمیم گرفت که داده مورد نظر متعلق به کدام خوشه است، خوشه سمت راست یا خوشه سمت چپ. اما اگر از خوشه بندی فازی استفاده کنیم داده مورد نظر با تعلق ۰,۵ عضو خوشه سمت راست و با تعلق مشابه عضو خوشه سمت چپ است. و همین طور برای بقیه داده‌هایی که بطور دقیق در وسط نیستند می‌توان به عنوان مثال گفت که داده مورد نظر با تعلق ۰,۷ عضو خوشه سمت راست و با تعلق ۰,۳ عضو خوشه سمت چپ است.

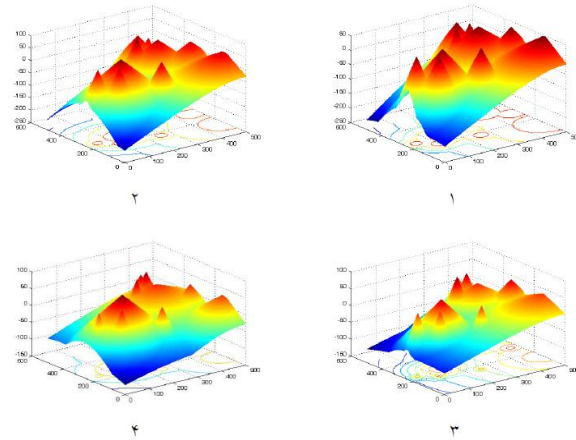
اگر بخواهیم قوانین را بیان کنیم بصورت زیر می‌توان نوشت:

IF $Dis_{i,x}$ is Low then $Dep_{i,x}$ is High

IF $Dis_{i,x}$ is Medium then $Dep_{i,x}$ is Medium

IF $Dis_{i,x}$ is High then $Dep_{i,x}$ is Low

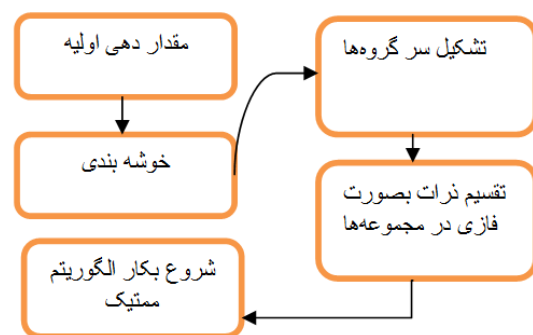
داده شده و پس از نسبت دادن مقادیر P_k به ذره، ابتدا در حلقه‌ای به طور اتفاقی یکی از ذره انتخاب می‌شود و عددی به صورت اتفاقی بین ۰ و ۱ انتخاب می‌شود سپس در صورتی که عدد تولید شده کوچکتر از P_k ذره باشد ذره به عنوان سرخوشه انتخاب می‌شود و حلقه پایان می‌پذیرد، در غیر این صورت حلقه تکرار می‌شود. لازم به ذکر است مقدار پارامتر تعلق را بین ۱ تا ۵ بوده و بسته به نوع مسئله قابل تغییر خواهد بود. این روند را ادامه می‌دهیم تا به تعداد C-max سرخوشه ایجاد شود. حال که سرخوشه‌ها انتخاب شده‌اند باید ذرات باقیمانده را با استفاده از روشی به عضویت خوشه‌های ایجاد شده دریاوریم. برای این کار از روش فازی استفاده کردیم. در این روش برای تقسیم‌بندی ابتدا باید میزان تعلق هر ذره به هر سرخوشه محاسبه شود، در اینجا این قطعیت وجود ندارد که یک ذره فقط و فقط بتواند به یک سردهسته متعلق باشد و در صورت عدم پذیرش از سوی سردهسته‌ها از بین برود و دوباره در محیط جایگذاری شود، هر ذره هر چقدر هم دور از سرخوشه‌ها باشد باز هم قطعاً به یکی از خوشه‌ها تعلق می‌یابد و به عنوان یکی از ذرات در ازدحام ذرات به کاوش محیط می‌پردازد. این امر باعث می‌شود تا پراکندگی ذرات در محیط بیشتر بوده و علاوه بر اینکه ذراتی که در ارتفاع بالاتر قرار دارند بهینه‌ی محلی را مورد جستجو قرار می‌دهند، این امکان به وجود می‌آید که این ذرات دورافتاده بهینه‌ی محلی دیگری را اکتشاف و دیگر ذرات را به سوی آن هدایت نماید. پس هر ذره به تمام سرخوشه‌ها تعلق دارد ولی این تعلق به تمام سرخوشه‌ها یکسان نیست. پس از محاسبه‌ی میزان تعلق ذرات به سرخوشه‌ها و ذخیره‌ی آنها در یک آرایه، هر ذره را به خوشه‌ای واگذار می‌کنیم. در این صورت تمام ذرات به خوشه‌ها می‌پیوندند. حال نوبت به ردیابی قله‌ها شده می‌باشد، پس از پیدا شدن قله‌ها چون محیط پیوسته تغییر می‌کند، عملگر جستجوی محلی ممیتی مبتنی بر نظریه لامارک همسایه‌ها را ردیابی کرده و در صورت جابه‌جا شدن قله ذرات اطراف آن هم جابه‌جا شده و این امر باعث عدم گم شدن قله می‌شود. چون از روش چند جمعیتی استفاده کرده‌ایم لذا دیگر نگران پیدا نکردن قله جدید دیگر در صورت بوجود آمدن نمی‌باشیم.



شکل (۲): تغییر در قله‌ها

۵- الگوریتم پیشنهادی در محیط پویا

در این مقاله با ترکیب الگوریتم ازدحام ذرات و تکنیک جستجوی محلی به صورت یک الگوریتم ممیتی، راه‌کاری برای حل مشکل محیط‌های پویا ارائه شده است. برای افزایش کارایی اکتشاف قله‌های جدید برای فضاهای جستجوی جدید الگوریتم ممیتی مبتنی بر ازدحام ذرات استفاده شده و مقایسه بر روی قله‌های متحرک نشان می‌دهد که الگوریتم ممیتی بر پایه ازدحام ذرات در محیط‌های پویا بسیار قوی عمل کرده است [۱۱]. حال به جای دسته‌بندی ذرات، آنها را خوشه‌بندی می‌کنیم [۱۲]. ذراتی را که روی صفحه‌ی محیط به صورت تصادفی پراکنده شده‌اند را چنانچه بر حسب شایستگی آنها دسته‌بندی کنیم، احتمال گرفتار شدن در نقاط بهینه محلی و متوقف شدن آن در مکانی به جز جواب مسئله زیاد است. شکل ۳ روند کار الگوریتم می‌باشد.



شکل (۳): فلوچارت الگوریتم

برای حل این مشکل پارامتری به نام τ معرفی می‌شود [۱۲]، برای پیدا کردن سرخوشه، ابتدا ذرات بر اساس ارزش و شایستگی، به صورت صعودی مرتب می‌شوند. سپس به هر ذره مقداری برابر P_k

1. Function FMempso
2. Begin
3. Initialize Swarms
4. Evaluation Swarms
5. Find Global & Local Best

۷-۵ شهریور ۱۳۹۲، تهران، ایران

طبق مقادیر استاندارد پارامترهای مولد قله‌های متحرک، نتایج زیر در فرکانس تغییر ۵۰۰ و ۱۰۰۰ آمده است.

جدول (۱): تعیین پارامترها

مقادیر	پارامترها
1	تعداد قله ها m
500	فرکانس تغییر f
0.7	سختی ارتفاع
1.0	سختی پهنا
مخروطی	شکل قله
1.0	طول تغییر S
5	ابعاد مسئله D
[30.0 70.0]	مقدار ماکزیمم و مینیمم ارتفاع H
[1 12]	مقدار ماکزیمم و مینیمم پهنا W
50.0	مقدار اولیه ارتفاع I
[0 100]	محدوده فضای جستجو A

6. While Swarms size
7. Begin
8. While (number of cluster Leader)
9. Begin
10. Fuzzy Clustering
11. End
12. Update Swarms velocity & locations
13. Update Global & Local Best
14. New Swarms = Local search (using One Of Memetician Local Search Algorithm)
15. Replace new Swarms
16. Crossover Swarms
17. Mutation Swarms
18. Selection Swarms
19. End
20. Swarms = New Swarms
21. Best fitness = max (Evaluation Swarms)
22. End

شکل (۴): شبه کد رویکرد کلی برای الگوریتم پیشنهادی

جدول (۲): مقایسه offline error در فرکانس ۵۰۰

تعداد د قله ها	MQSO [13]	Cellular pso [13]	CPSOL [13]	Memetic+
۱	۲۸,۳۸±۱,۸۹	±۰,۶۸ ۱۱,۴۱	۸,۲۹±۰,۵۵	۱,۴±۰,۲
۱۰	۵,۷۴±۰,۸	۶,۸۱±۰,۱۷	۵,۴۵±۰,۱۷	۳,۰۱±۰,۳
۲۰	۶,۸۰±۰,۰۴	۷,۷۳±۰,۰۷	۵,۴۷±۰,۱۹	۲,۸۵±۰,۵۰
۳۰	۷,۲۸±۰,۰۴	۸,۳۹±۰,۱۱	۵,۵۹±۰,۱۲	۳,۱±۰,۳

جدول (۳): مقایسه offline error در فرکانس ۱۰۰۰

تعداد قله ها	MQSO [13]	Cellular pso [13]	CPSOL [13]	Memetic +
۱	۱۵,۰۲±۰,۹۹	۸,۷۴±۰,۴۰	۴,۷۴±۰,۳۲	۱,۳±۰,۱
۱۰	۴,۰۰±۰,۰۶	۴,۷۸±۰,۱۱	۳,۲۰±۰,۲۰	۲,۷±۰,۳۲
۲۰	۵,۱۱±۰,۰۵	۵,۹۵±۰,۱۱	۳,۵۲±۰,۱۷	۲,۶۵±۰,۲۹
۳۰	۵,۷۱±۰,۰۵	۶,۶۵±۰,۱۲	۳,۹۶±۰,۱۲	۲,۹۸±۰,۳

همان‌طور که در شکل ۴ مشخص است ابتدا در بندهای ۳ و ۴ ذرات مقداردهی اولیه شده و میزان تناسب آن‌ها حساب می‌شود. سپس میزان بهترین سراسری و بهترین محلی پیدا شده، حال به تعداد مراکز خوشه‌ها بایستی ذرات را خوشه بندی کنیم. چگونگی خوشه بندی در ابتدای همین بخش توضیح داده شده است. سپس مراحل الگوریتم ازدحام ذرات شروع شده، ذرات سرعتشان و همین‌طور موقعیتشان با روند پیشرفت الگوریتم تغییر کرده و حال نوبت به بخش ممیتی می‌باشد. بند ۱۴ همسایگی ذرات را بررسی کرده و موقعیت ذرات همسایه با در نظر گرفتن نوع ممیتی (بالدوین یا لامارک) دستخوش تغییر قرار می‌گیرد. حال در بندهای ۱۵ تغییرات اعمال می‌شود. برای حفظ تنوع می‌توان از دو متد ژنتیکی هم استفاده کرد که در بندهای ۱۷ و ۱۸ آمده است. برای اطلاع بیشتر در مورد چگونگی الگوریتم ازدحام ذرات و متدهای ژنتیکی می‌توان از مرجع ۱۱ با توضیحات بیشتر بهره برد.

۶- آزمایش و نتایج

خطای آفلاین به‌عنوان معروف‌ترین معیار ارزیابی کارایی کمی برای بهینه‌سازی محیط‌های پویا به‌صورت معادله ۳ تعریف می‌شود.

Offline – Error =

$$\frac{1}{T} \sum_{t=1}^T (\text{Minimum Error After last change}) \quad (3)$$

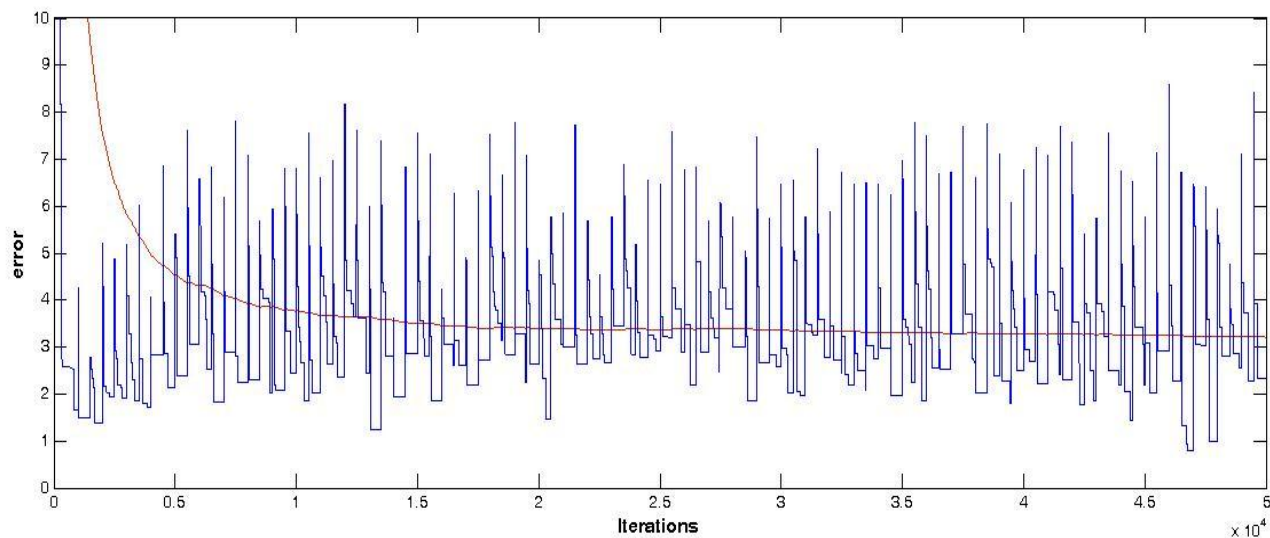
در آزمایشات انجام شده اندازه جمعیت بصورت ثابت برابر ۱۰۰ در فرکانسهای مختلف جوابهای جداول بالا را تولید کرده و مقایسه بر اساس میانگین گیری از ۲۰ بار اجرای مکرر انجام شده است. به عنوان نمونه در فرکانس ۵۰۰، ۱ قله و ۲۰ بار اجرا میانگین برابر ۱,۴ بدست آمده و انحراف معیار برابر ۰,۲ می باشد. که این انحراف معیار نشان دهنده کارایی الگوریتم ها و اثبات نتیجه آنها می باشد، هر چه میزان این شاخص پایین بوده به منزله بهتر بودن الگوریتم از لحاظ سازگاری با محیط است. در جدولهای ۳ و ۲ الگوریتم ما با سه الگوریتم دیگر مقایسه شده است. جدولهای ۴ و ۵ نشان دهنده کارایی بهتر الگوریتم نسبت به نسخه غیر فازی می باشد. در شکل ۵ نمایی از تغییرات خطای جاری و همین طور خطای آفلاین (بصورت نزولی) در آزمایش با فرکانس تغییر ۵۰۰ و ۱۰ قله آمده است.

جدول (۴): مقایسه خطای آفلاین در فرکانس ۵۰۰

Memetic+	Memetic[11]	قله
۱,۴±۰,۲	۱,۴±۰,۳	۱
۳,۱±۰,۳	۳,۱۲±۰,۷	۳۰

جدول (۵): مقایسه خطای آفلاین در فرکانس ۱۰۰۰

Memetic+	Memetic[11]	قله
۱,۳±۰,۱	۱,۳±۰,۵	۱
۲,۹۸±۰,۳	۳,۲۲±۰,۴	۳۰



شکل (۵): نمودار خطای جاری و آفلاین در فرکانس ۵۰۰ و ۱۰ قله

مراجع

- [1] H. Wang, et al., "A particle swarm optimization based memetic algorithm for dynamic optimization problems," *Natural Computing*, vol. 9, no. 3, pp. 703-725, 2010.
- [2] Wang, H., Wang, D., Yang, S., "A Memetic Algorithm with Adaptive Hill Climbing Strategy for Dynamic Optimization Problems," *Soft Computing*, vol. 13, no. 8-9, pp. 763-780, 2008.
- [3] Liu, L., Yang, S., and Wang, D., "Particle Swarm Optimization With Composite Particles in Dynamic Environments," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 40, no. 6, pp. 1634-1648, 2010.
- [4] Cotta, C., Fernandez, A. J., Gallardo, J. E., "On the Hybridization of Memetic Algorithms with Branch-and-Bound

۶- نتیجه گیری

نتایج جداول ۳ و ۲ و ۴ و ۵ بر اساس میانگین گیری پس از ۲۰ بار اجرا، حاکی از آن است که الگوریتم پیشنهادی نسبت به نسخه بدون چند جمعیتی خود و الگوریتمهای دیگر کارایی بهتری از خود نشان داده است. در مقایسه با الگوریتمهای ممیتیک و ازدحام ذرات و امثال آنها بصورت منفرد، این الگوریتم کارایی قابل قبولی از خود نشان داده است. ولی هنوز امکان افزایش سرعت همگرایی پس از اعمال تغییر در محیط وجود دارد.

- Techniques," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 37, no. 1, pp. 77-83, 2007.
- [5] Chen, X., Ong, Y., Lim, M., Tan, K., "A Multi-facet Survey on Memetic Computation", *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 591-607, 2011.
- [6] Yang, S., Li. C., "A Clustering Particle Swarm Optimizer for Locating and Tracking Multiple Optima in Dynamic Environments," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 6, pp. 959-974, 2010.
- [7] Noroozi, V., Hashemi, A.B., Meybodi, M.R., "CellularDE: A Cellular Based Differential Evolution for Dynamic Optimization Problems," *Adaptive and natural computing algorithms*, vol. 6593, pp. 340-349, 2011.
- [8] Branke, J. "Evolutionary Optimization in Dynamic Environments," *IEEE transactions on evolutionary computation*, vol. 9, no. 3, 2005.
- [9] Kumar, S., Chaturvedi. D. K., "Tuning of Particle Swarm Optimization Parameter Using Fuzzy Logic," *2011 International Conference on Communication Systems and Network Technologies (CSNT)*, vol., no., pp.174-179, 2011.
- [10] Branke, J. "Evolutionary optimization in dynamic environments," *Kluwer Academic Publishers, Dordrecht* 2002.
- [11] Alizadeh, M., Meybodi, M. R. and Rezvanian, A., "A Memetic Algorithm Based on PSO for Moving Peak Problem", *Proceeding of 18th National Conference of Computer Society of Iran*, Sharif University, Tehran, Iran, March 14-16, 2013.
- [12] Jafareih, F., Meybodi, M. R. and Golshani, Z., "Optimization Algorithms based on Particle Swarm Optimization for Dynamic Environment using Fuzzy Clustering", *Proceedings of International Conference on Electrical Engineering, Isfahan University of Technology, Isfahan, Iran, May 11-13, 2010*.
- [13] S. Nabizadeh, A. Rezvanian, and M. R. Meybodi, "Tracking Extrema in Dynamic Environment using Multi-Swarm Cellular PSO with Local Search," *International Journal of Electronics & Informatics (IJEI)*, vol. 1, no. 1, pp. 29-37, 2012.

زیر نویس ها

¹ Cluster

² Distance-based Clustering

³ Uncertainty