

الگوریتم ممتیکی مبتنی بر الگوریتم ازدحام ذرات برای مسئله قله‌های متحرک

مرتضی علی‌زاده^۱، محمدرضا میبیدی^۲، علیرضا رضوانیان^۳

^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

^۳ آزمایشگاه محاسبات نرم، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر تهران، ایران
a.rezvanian@aut.ac.ir

چکیده

بسیاری از مسائل دنیای واقعی ماهیتی پویا دارند، به این مفهوم که موقعیت و مقدار بهینه‌سراسری آنها در طول زمان تغییر می‌کند. یکی از این مسائل معروف در بهینه‌سازی محیط‌های پویا، مسئله تابع محک قله‌های متحرک یا ماکزیمم‌های در حال تغییر می‌باشد، که رفتاری شبیه به مسائل پویا در دنیای واقعی را دارد. در این مقاله یک رویکرد ممتیکی مبتنی بر بهینه‌سازی ازدحام ذرات برای بهینه‌سازی محیط‌های پویا ارائه شده است. در روش پیشنهادی، الگوریتم ممتیک-ازدحام ذرات فازی برای حل این مسئله مطرح شده است، که به نوعی ترکیبی از الگوریتم‌های تکاملی و هوش تجمعی می‌باشد که در این حالت الگوریتم پیشنهادی خصوصیات هر دو گروه را در خودش دارد. الگوریتم پیشنهادی برای بهینه‌یابی تابع محک قله‌های متحرک در فرکانس‌های مختلف و با تعداد قله‌های متفاوت آزمایش شده و با چند الگوریتم دیگر مقایسه شده است. نتایج آزمایشات نشان‌دهنده بهبود کارایی الگوریتم پیشنهادی نسبت به الگوریتم‌های ازدحام ذرات استاندارد و ممتیک می‌باشد.

کلمات کلیدی

بهینه‌سازی، محیط پویا، قله‌های متحرک، ممتیک، ازدحام ذرات، پارامتر فازی

مسائل زدند که توانستند از مزایای هر دو نوع استفاده کنند. یکی از این نوع الگوریتم‌ها، الگوریتم ممتیک می‌باشد که توسط بالدوین^۴ و لامارک^۵ [۱] که از نظریه‌پردازان اولیه بوده‌اند بیان شده و توسط موسکاتو^۶ در سال ۱۹۸۹ بطور کامل ارائه شده است که برای حل مسائل پویا ارائه گردیده است و از تابع محک^۷ قله‌های متحرک^۸ که رفتاری شبیه به مسائل پویا را در دنیای واقعی دارد استفاده شده است. قله‌های متحرک بیانگر یکسری اکستریم در محیطی هستند که جایگاه و شکل و ارتفاع و پهنای این اکستریم‌ها پیوسته در حال تغییر اند، حال بدنبال راهی برای پیدا کردن آنها هستیم [۲].

در مسائلی که از عدم قطعیت استفاده می‌کنند معمولاً واژه پویایی و پویا بودن راه حل بهینه همراه است. تاکنون روش‌های مختلفی برای حل مسائل پویا با استفاده از روش‌های تکاملی و هوش تجمعی ارائه شده است، ولی مشکل اصلی این الگوریتم‌ها در محیط‌های پویا، حافظه

۱- مقدمه

با توجه به پیچیدگی مسائل بهینه‌سازی و کارایی روش‌های مبتنی بر الگوریتم‌های تکاملی، در سال‌های اخیر، بکارگیری الگوریتم‌های تکاملی در حل مسائل بهینه‌یابی پویا^۱ افزایش یافته است. همچنین در کنار آن هوش دست جمعی نیز مشتاقان زیادی را به خود جلب کرده است. هوش دسته جمعی، عامل‌ها یا دسته‌های برهم کنش را که قادر به خودسازماندهی هستند مدلسازی می‌کند. کلونی مورچه^۲، دسته پرندگان و زنبورها^۳ مثال‌هایی از یک سیستم جمعی می‌باشند، ولی الگوریتم‌های تکاملی و هم الگوریتم‌های هوش دسته جمعی یکسری مزیت‌ها و معایبی دارند که ممکن است جواب بهینه را تولید نکنند، به همین منظور محققان در سال‌های اخیر دست به ترکیب این



تغییراند. حال موضوع مهم این است، بهینه را در این تغییرات پیدا نمود. تابع قله های 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 قله ای 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 می باشد که به صورت رابطه (۱) تعریف می شود:

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

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 قله ها 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 می کند. در این تابع می توان پیچیدگی را با افزایش کاهش Δe در زمان تغییر داد. [۲]

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

بردار v را می توان وابسته به تغییر قبلی آن ایجاد کرد که در این صورت تغییر موقعیت قله ها همسو با تغییرات قبل آن می شود و یا به صورت تصادفی آنرا ایجاد نمود که موجب می شود موقعیت قله ها به صورت تصادفی تغییر کند و هیچگونه وابستگی به تغییر قبلی نداشته باشد. [۲]

۳- الگوریتم پیشنهادی

در این مقاله با ترکیب الگوریتم ازدحام ذرات^{۱۳} و تکنیک جستجوی محلی به صورت یک الگوریتم ممتیک، راه حلی برای محیط های پویا ارائه شده است. برای افزایش کارایی اکتشاف قله های جدید برای فضاهای جستجوی جدید از طرح مهاجرت ذرات تصادفی خودسازمانده بهره مند شده است. مطالعات تجربی و پیاده سازی و مقایسه بر روی قله های متحرک نشان می دهد که الگوریتم ممتیک بر پایه ازدحام ذرات در محیط های پویا بسیار قوی عمل کرده است. ازدحام ذرات دارای دو پارامتر محلی و سراسری می باشد. در اینجا از پارامتر محلی به همراه جستجوی محلی استفاده شده است. با توجه به شبهه ازدحام ذرات مبتنی بر ممتیک، چند مفهوم ضروری در ادامه توضیح داده شده است.

۳-۱- ممتیک

الگوریتم های ژنتیکی همانند سایر الگوریتم های مکاشفای^{۱۴} (مانند بهینه سازی گروه ذرات) در گام های نخست اجرای الگوریتم ناحیه هایی از فضای حالت مسئله که بهینه های سراسری و محلی در آن واقع شده

غیرمعتبر و از دست رفتن تنوع می باشد. همچنین از آنجا که اکثر روش های پردازش تکاملی و هوش جمعی به دلیل ماهیتشان به یک نقطه همگرا می شوند، لذا تنوع دسته در محیط از بین می رود و در صورت تغییر در محیط همگرا شدن به نقطه بهینه جدید در صورت امکان بسیار زمان گیر است. ساده ترین راه حل برای این مشکل (تغییرات محیط)، در نظر گرفتن هر تغییرات به عنوان ورودی یک مساله بهینه سازی جدید می باشد که باید از ابتدا حل شود.

الگوریتم ممتیک^{۱۵} گونه ای از الگوریتم های تکاملی^{۱۱} است که در آن جستجوهای ابتکاری محلی با الگوریتم ژنتیک ترکیب می شوند تا در زمان کمتر نتایج بهتری به دست آید. این الگوریتم براین ایده استوار است که هر عضو جمعیت می تواند به اندازه اطرافیان خود (همسایگان) شایستگی^{۱۱} خود را افزایش دهد. در این مقاله از الگوریتم ازدحام ذرات، که نوعی از الگوریتم های هوش جمعی محسوب می شود، به عنوان یک جستجوگر ابتکاری استفاده شده است همچنین این الگوریتم کارایی بالایی در محیط های پویا از خود نشان داده است. در این مقاله ابتدا محیط مورد نظر، سپس الگوریتم پیشنهادی و آزمایشات و نتایج مورد بحث قرار گرفته شده است.

۲- محیط پویا^{۱۲}

در مسائلی که از عدم قطعیت استفاده می کنند معمولاً واژه پویایی و پویا بودن همراه راه حل بهینه است. تاکنون روش های مختلفی برای حل مسائل پویا با استفاده از روش های تکاملی و هوش جمعی ارائه شده است ولی مشکل اصلی این الگوریتم ها در محیط های پویا، حافظه ی غیرمعتبر و از دست رفتن تنوع می باشد. همچنین از آنجا که اکثر روش های پردازش تکاملی و هوش جمعی به دلیل، ماهیتشان به یک نقطه همگرا می شوند، لذا تنوع دسته در محیط از بین می رود و در صورت تغییر در محیط همگرا شدن به نقطه بهینه جدید در صورت امکان بسیار زمان گیر است. ساده ترین راه حل برای این مشکل (تغییرات محیط)، در نظر گرفتن هر تغییرات به عنوان ورودی یک مساله بهینه سازی جدید می باشد که باید از ابتدا حل شود.

در صورت داشتن زمان کافی این یک گزینه مناسب می باشد. اما در بیشتر مواقع زمان موجود برای بهینه سازی مجدد نسبتاً کوتاه می باشد. یک تلاش طبیعی برای تسریع فرایند بهینه سازی پس از یک تغییر، استفاده از اطلاعات مربوط به فضای جستجوی قبلی برای پیشروی در جستجو پس از تغییر می باشد.

نکته اساسی دیگر این است که وقتی یک الگوریتم به یک نقطه همگرا می شود، تنوع خود را از دست می دهد که این امر باعث کاهش قابلیت انطباق و سازگاری با تغییر محیط می شود. بنابراین در کنار انتقال اطلاعات بین عامل های الگوریتم های بهینه سازی از قبل و بعد از تغییر محیط، باید به دنبال راهکارهایی برای افزایش تنوع و سازگاری الگوریتم پس از تغییر محیط بود.

یک تابع محک، معروف به نام قله های متحرک [۲] برای محیط های پویا وجود دارد. در این تابع محک قله ها بطور پیوسته با زمان در حال



شبه کد زیر نمای کلی از الگوریتم ممتیک می‌باشد، همانطور که می‌بینید شباهت زیادی به الگوریتم ژنتیک داشته ولی به همراه یک جستجوی محلی می‌باشد که در شبه کد شکل ۲ و ۳، این جستجوی محلی آمده است.

```
Function Mempo
Begin
  Initialize population
  Evaluation population
  While population size
    Begin
      New pop=Local search (using PSO Search Algorithm)
      Replace new pop
    Crossover
    Mutation
    Selection
  End
  Population = New pop
  Best fitness=max (Evaluation population)
End
```

شکل (۲): شبه کد رویکرد کلی برای الگوریتم پیشنهادی

```
Local Search(population)
While (size of population)
  Begin
    Best=Select some Best of population
    Generated neighbor from best
    Compare each individual with them neighbors
    If fitness(neighbor (i))>fitness(best(i))
      Lamark or Baldwin method {(Replace best by
      neighbor and fitness) or (Replace fitness of best
      neighbor)}
    End if
  Insert best in new population
  Return new population
End
```

شکل (۳): شبه کد جستجوی محلی ممتیک

۲-۲- ازدحام ذرات

منبع الهام این الگوریتم، رفتار اجتماعی حیوانات، همانند حرکت دسته جمعی پرندگان و ماهی‌ها بوده است. هر عنصر جمعیت، یک ذره نامیده می‌شود. [۶] در واقع الگوریتم ازدحام ذرات از تعداد مشخصی از ذرات تشکیل می‌شود که به طور تصادفی، مقدار اولیه می‌گیرند. برای هر ذره دو مقدار وضعیت و سرعت، تعریف می‌شود که به ترتیب با یک بردار مکان و یک بردار سرعت، مدل می‌شوند. این ذرات، بصورت تکرارشونده‌ای در فضای n بعدی مسئله حرکت می‌کنند تا با محاسبه مقدار بهینگی به عنوان یک ملاک سنجش، گزینه‌های ممکن جدید را جستجو کنند. بُعد فضای مسئله، برابر تعداد پارامترهای موجود در تابع مورد نظر برای بهینه‌سازی می‌باشد. یک حافظه به ذخیره بهترین موقعیت هر ذره در گذشته و یک حافظه به ذخیره بهترین موقعیت پیش آمده در میان همه ذرات، اختصاص می‌یابد. با تجربه حاصل از این حافظه‌ها، ذرات تصمیم می‌گیرند که در نوبت بعدی، چگونه حرکت کنند. در هر بار تکرار، همه ذرات در فضای n بعدی مسئله حرکت می-

اند را به خوبی شناسایی می‌کنند اما در ادامه مسیرشان به سمت بهینه سراسری بسیار کند عمل می‌نمایند [۱].

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 متاهوریستیک‌ها ارائه شده است، استراتژی ترکیبی جایگاه ویژه‌ای دارند [۳ و ۱].

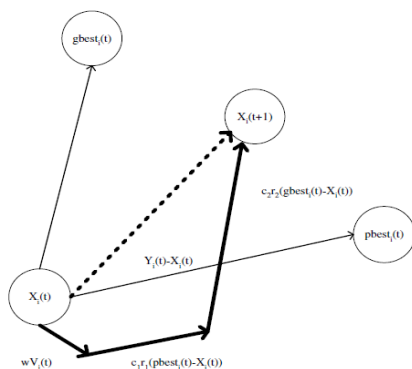
استراتژی ترکیبی از به کارگیری تکنیک‌های مختلف در فرایند حل مسئله حاصل می‌گردد. الگوریتم‌های ممتیک از مشهورترین اعضاء این خانواده به شمار می‌آیند که از پیوندسازی الگوریتم‌های ژنتیک با یک هیوریستیک جستجوی محلی مانند روش تپهنوردی حاصل می‌گردد [۳ و ۴]. در این الگوریتم‌ها یک عملگر جستجوی محلی پس از عملگرهای ژنتیکی شایستگی را بهبود می‌بخشد (تقلید) و پس از آن اعمال ارزیابی و جایگزینی جمعیت صورت می‌گیرد. این که برای پیاده سازی یک الگوریتم ممتیک برای جستجوی محلی از کدام روش استفاده شود، نتایج اجرای بسیار متفاوتی خواهد داشت و وابسته به نوع مسأله است، اما در بسیاری از مسائل به خصوص مسائلی با محیط - های گسسته نیازمند تعریف یک روش جستجوی محلی جدید است. انواع مختلفی برای الگوریتم ممتیک معرفی شده است.

۳-۱-۱- الگوریتم‌های ممتیک 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 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 ممتیک (جایگزینی بهترین همسایه) صورت می‌گیرد [۵].

۳-۱-۲- الگوریتم‌های ممتیک 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 سهم می‌شود به همین علت به آن اثر بالدوین نیز می‌گویند [۳]. لازم به ذکر است در برخی از روش‌ها از هر دو رویکرد به صورت همزمان نیز بهره برده شده است که به آنها متا لامارکی گفته می‌شود [۵].



شکل (۴): چگونگی حرکت ذره و بروز رسانی موقعیتش

تعیین پارامتر الگوریتم ازدحام ذرات، با سیستم فازی، سبب بهبود عملکرد الگوریتم ازدحام ذرات گشته است. در این مقاله پارامتر اینرسی، در طی دوره اجرای الگوریتم، به واسطه ی کنترل کننده ی فازی، تعیین و تطبیق داده می شود. کنترل کننده ی فازی دارای یک ورودی و یک خروجی می باشد که تعیین پارامتر وزن اینرسی به عنوان خروجی سیستم فازی در نظر گرفته شده است. پیاده سازی انجام شده بهبود سیستم ارائه شده، نسبت به الگوریتم استاندارد ازدحام ذرات را نشان می دهد. این الگوریتم از طریق تغییر یکی از سه پارامتر زیر قابل بهبود است :

- ۱) ضریب وزن اینرسی
- ۲) ضریب پارامتر اجتماعی
- ۳) ضریب پارامتر ادراکی

در این مقاله سیستم فازی به عنوان یک سیستم جدا به الگوریتم اضافه شده است. پارامتر ورودی این سیستم فازی عدد تکرار^{۱۵} الگوریتم و خروجی سیستم فازی پارامتر ضریب وزنی اینرسی می باشد. ایده ی اصلی روش پیشنهادی مبتنی بر بهبود حرکات ذرات در فضای مسئله با تطبیق آن با دفعات تکرار می باشد. تعیین هدفمند پارامترهای الگوریتم ازدحام ذرات سبب کارایی بهتر الگوریتم شده است. در سال ۲۰۰۱، ابرهات سیستمی مبتنی بر فازی ارائه داد [۸] اما در این مقاله سعی شده تا با استفاده از سیستم فازی با توجه به تعداد دفعات تکرار مقدار سیستم فازی تعیین گردد [۶]

ورودی سیستم فازی شماره تکرار در جستجوی محلی بکار رفته می باشد و با تکرار کل الگوریتم رابطه ای ندارد. مقدار خروجی سیستم فازی برای الگوریتم ازدحام ذرات ضریب پارامتر اینرسی می باشد. توابع سیستم فازی برای الگوریتم ازدحام ذرات در شکل ۵ آمده است، تعریف می شود.

کنند تا در نهایت نقطه بهینه عام، پیدا شود. حال ذرات، سرعت هایشان و موقعیت شان را برحسب بهترین جواب های مطلق و محلی به روز می کنند. معادله ۳ نشان دهنده تغییر وضعیت و به روز رسانی سرعت ذره می باشد [۷].

۳-۳- ممتیک - ازدحام ذرات فازی

انواع مختلفی از الگوریتم های تکاملی و هوش جمعی وجود دارند که برای جستجو در محیط ها و بهینه یابی استفاده می شوند، که در اینجا از الگوریتم ازدحام ذرات فازی به عنوان جستجوگر بین همسایه های یک مم در نظر گرفته شده است، استفاده شده است.

هر عنصر جمعیت، یک ذره نامیده می شود (کروموزوم در ژنتیک و یا کشور در الگوریتم رقابت استعماری) است. در واقع الگوریتم ازدحام ذرات از تعداد مشخصی از ذرات تشکیل می شود که به طور تصادفی، مقدار اولیه می گیرند. برای هر ذره دو مقدار وضعیت و سرعت، تعریف می شود، همه ذرات در فضای n بعدی مسئله (همسایگی) حرکت می کنند تا در نهایت نقطه بهینه عام، پیدا شود. حال ذرات، سرعت هایشان و موقعیت شان را برحسب بهترین جواب های مطلق و محلی به روز می کنند [۶]. در واقع به سمت بهترین همسایه پیش می روند.

الگوریتم ازدحام ذرات استاندارد از سه مرحله ی زیر تشکیل شده است :

- مقدار دهی اولیه موقعیت و سرعت ذرات
- به روز رسانی بردار سرعت
- به روز رسانی موقعیت

$$\begin{aligned} v_{id}(t+1) &= \omega v_{id}(t) + C_1 r_1 (p_{id} - x_{id}(t)) + C_2 r_2 (p_{gd} - x_{id}(t)) \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1) \end{aligned} \quad (3)$$

که در آن ω پارامتر اینرسی، v_{id} سرعت i امین ذره، x_{id} موقعیت i امین ذره، r_1, r_2 اعداد تصادفی مستقل با توزیع یکنواخت، C_1, C_2 فاکتورهای یادگیری (ثابت های C_1, C_2 به ترتیب، پارامتر ادراکی و پارامتر اجتماعی نامیده می شوند)، p_{id} بهترین جواب محلی و p_{gd} بهترین جواب مطلق می باشند. الگوریتم ازدحام ذرات، بردار سرعت هر ذره را به روز کرده و سپس مقدار سرعت جدید را به موقعیت و یا مقدار سرعت ذره می افزاید که شکل ۴ نشان دهنده این گفته است [۶].

به روز کردن های سرعت، تحت تأثیر هر دو مقدار بهترین جواب محلی و بهترین جواب مطلق قرار می گیرند. بهترین جواب محلی و بهترین جواب مطلق، بهترین جواب هایی هستند که تا لحظه ی جاری اجرای الگوریتم، به ترتیب توسط یک ذره و در کل جمعیت به دست آمده اند. مزیت اصلی ازدحام ذرات این است که پیاده سازی این الگوریتم ساده بوده و نیاز به تعیین پارامترهای کمی دارد.

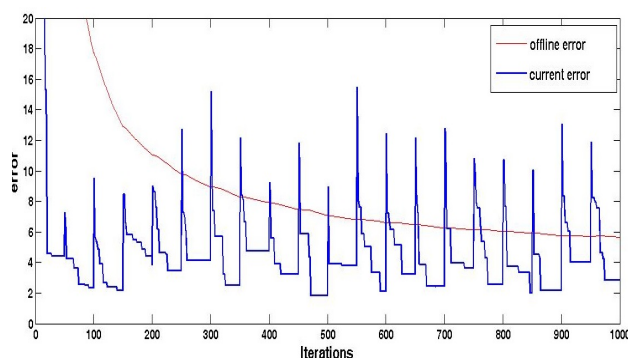
جدول (۲): تعیین پارامترهای قسمت ازدحام ذرات الگوریتم

پارامترها	مقادیر
C_1	2
C_2	2
SWARM SIZE اندازه جمعیت	80
Itration تکرار در جستجوی محلی	10
Wینرسی	[0...1] بصورت فازی

در آزمایشات انجام شده اندازه جمعیت بصورت ثابت برابر ۱۰۰ در فرکانس های مختلف جواب های جداول ۵ و ۴ 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 ۱.۴۸ بدست آمده و انحراف معیار برابر ۰.۳ می باشد و همین فرکانس، ۱۰ قله و ۲۰ بار اجرا میانگین برابر ۳.۰۱ بدست آمده و و انحراف معیار برابر ۰.۶۸ 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.

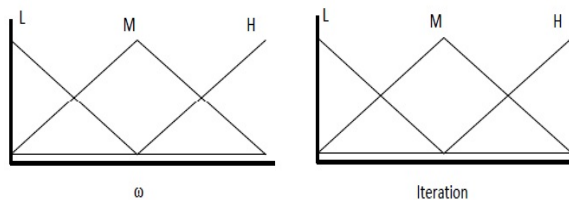
جدول (۳): میانگین و انحراف معیار در ۲۰ بار اجرا

فرکانس	تعداد قله	۱	۱۰	۲۰	۳۰
۵۰۰ و ۱۰۰۰	میانگین	۱.۴±۰.۳	۳.۰۱±۰.۶۸	۲.۹۷±۰.۶۱	۳.۱۲±۰.۷
	انحراف معیار	۱.۳±۰.۵	۲.۷±۰.۴۲	۲.۶۸±۰.۲۱	۳.۲۲±۰.۴



شکل (۶): نمودار *offline error* برای فرکانس ۵۰ و تک قله

شکل ۶ نشان دهنده تعداد تغییرات برای ۱۰۰۰ تولید نسل با فرکانس تغییر ۵۰ است، به ازای هر ۵۰ نسل تولید شده، یک تغییر دیده می شود که در کل این اجرا تغییر در قله ها ۲۰ بار می باشد. همچنین مقایسه ای هم بین الگوریتم پیشنهادی و الگوریتم های دیگر بر اساس پارامترهای استاندارد تعیین شده در جدول ۴ و ۵، نیز در جداول ۵ و ۴ با فرکانس های مختلف آمده است.



شکل (۵): نمونه ورودی و خروجی سیستم فازی

در این سیستم از سیستم فازی ممدانی و نیز از غیر فازی ساز میانگین مراکز استفاده شده است. قوانین مورد استفاده نیز به شرح زیر است.

If itr = L then ω = H
Rules If itr = M then ω = M
If itr = H then ω = L

به همین صورت برای الگوریتم ممتیک، سیستم فازی تعریف می شود ولی با این فرق که خروجی ضریب پارامترهای جهش و تقاطع است. فرض این مقاله این است که ابتدای الگوریتم بایستی با تنوع بالا شروع به کار کند، به همین منظور نرخ تقاطع در ابتدا زیاد و نرخ جهش کم در نظر گرفته می شود و برای افزایش کارایی از فازی سازی استفاده می شود.

۴- آزمایشات و نتایج

در آزمایش های انجام شده هدف بررسی و مقایسه الگوریتم های Cpsol, mQSO, Cellular pso [9], RPSO, و الگوریتم پیشنهادی در محیط های پویا به کمک تابع قله متحرک می باشد. از آنجا که در محیط های پویا بهترین راه حل بدست آمده معیار مفیدی برای ارزیابی و گزارش نمی باشد، از معیارهای *offline Error, Current Error* جهت مقایسه و نتیجه گیری استفاده شده است.

مقادیر پارامترهای برای انجام آزمایش ها طبق جدول (۱) و جدول (۲) در نظر گرفته شده است.

جدول (۱): تعیین پارامترها

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



- [2] Branke, J. Evolutionary optimization in dynamic environments, Kluwer Academic Publishers, Dordrecht 2002.
- [3] Yew-soon ong, meng-hiot lim, ning zhu, kok-wai wong; "Classification of adaptive memetic algorithms: a comparative study", Systems, Man and Cybernetics, Part B, IEEE Transactions, vol. 36, no. 1, pp 141-152, 2006.
- [4] Guimaraes, F.G.; Campelo, F.; Igarashi, H.; Lowther, D.A.; Ramirez, J.A.;, "Optimization of Cost Functions Using Evolutionary Algorithms With Local Learning and Local Search," Magnetics, IEEE Transactions on, vol.43, no.4, pp.1641-1644, April 2007.
- [5] Krasnogor, N.; Smith, J.;, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," Evolutionary Computation, IEEE Transactions on, vol.9, no.5, pp. 474- 488, Oct. 2005.
- [6] Kumar, S.; Chaturvedi, D.K.;, "Tuning of Particle Swarm Optimization Parameter Using Fuzzy Logic," Communication Systems and Network Technologies (CSNT), 2011 International Conference on, vol., no., pp.174-179, 3-5 June 2011.
- [7] Yadmellat, P.; Salehizadeh, S.M.A.; Menhaj, M.B.;, "Fuzzy Parameter Particle Swarm Optimization," Intelligent Networks and Intelligent Systems, 2008. ICINIS '08. First International Conference on, vol., no., pp.93-98, 1-3 Nov. 2008.
- [8] Xiaohui Hu; Eberhart, R.C.;, "Adaptive particle swarm optimization: detection and response to dynamic systems," Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on, vol.2, no., pp.1666-1670, 2002.
- [9] Hashemi, A.B.; Meybodi, M.R.; "Cellular Pso: A Pso for Dynamic Environment", in to Be Appear in the 4th International Symposium on Intelligence Computation and Applications (ISICA 2009), Huangshi, China, Vol. 5821, pp 422-433, 2009.
- [10] S. Nabizadeh, A. Rezvanian, 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.
- [11] Blackwell, T.; Branke, J.; "Multiswarms, Exclusion, and Anti-Convergence in Dynamic Environments", IEEE Transactions on Evolutionary Computation, vol. 10, no. 4, pp. 459-472, 2006.

زیر نویس ها

¹ Dynamic optimization

² Ant Colony

³ Bee Colony

⁴ Baldwin

⁵ Lamarck

⁶ Muscato

⁷ Benchmark

⁸ Moves Peaks

⁹ Memetic

¹⁰ Evolutionary Algorithm

¹¹ Fitness

¹² Dynamic Environment

¹³ Particle swarm optimization

⁹ Meta heuristic

¹⁵ Iteration

¹⁶ Chaotic

جدول (۴): مقایسه *offline error* در فرکانس ۵۰۰

PROPOSAL ALGORITHM	CPSOL [10]	Cellular pso [9]	MQSO [11]	تعداد قله ها
۱.۴±۰.۳	۸.۲۹±۰.۵۵	۱۱.۴۱±۰.۶۸	۲۸.۳۸±۱.۸۹	۱
۳.۰۱±۰.۶۸	۵.۴۵±۰.۱۷	۶.۸۱±۰.۱۷	۵.۷۴±۰.۸	۱۰
۲.۹۷±۰.۶۱	۵.۴۷±۰.۱۹	۷.۷۳±۰.۰۷	۶.۸۰±۰.۰۴	۲۰
۳.۱۲±۰.۷	۵.۵۹±۰.۱۲	۸.۳۹±۰.۱۱	۷.۲۸±۰.۰۴	۳۰

جدول (۵): مقایسه *offline error* در فرکانس ۱۰۰۰

PROPOSAL ALGORITHM	CPSOL [10]	Cellular pso [9]	MQSO [11]	تعداد قله ها
۱.۳±۰.۵	۴.۷۴±۰.۳۲	۸.۷۴±۰.۴۰	۱۵.۰۲±۰.۹۹	۱
۲.۷±۰.۴۲	۳.۲۰±۰.۲۰	۴.۷۸±۰.۱۱	۴.۰۰±۰.۰۶	۱۰
۲.۶۸±۰.۲۱	۳.۵۲±۰.۱۷	۵.۹۵±۰.۱۱	۵.۱۱±۰.۰۵	۲۰
۳.۲۲±۰.۴	۳.۹۶±۰.۱۲	۶.۶۵±۰.۱۲	۵.۷۱±۰.۰۵	۳۰

نتایج نشان دهنده آن است که با افزایش فرکانس تغییر، یعنی افزایش فاصله تغییرات از هم، الگوریتم زمان بیشتری برای جستجو داشته و کارایی بالاتری از خود نشان می دهد. نتایج جداول بالا نشان دهنده کارایی بالای الگوریتم پیشنهادی نسبت به الگوریتم های دیگر بیان شده می باشد.

۵- نتیجه گیری

در این مقاله یک الگوریتم جدید برای بهینه یابی که بر پایه ترکیب دو الگوریتم ممتیک و ازدحام ذرات می باشد، پیشنهاد گردید. الگوریتم پیشنهادی دارای افزایش سرعت همگرایی و کاهش قرار گرفتن در بهینه محلی می باشد. همچنین با فازی سازی پارامتر های الگوریتم، سرعت همگرایی افزایش پیدا کرده است. در مقایسه با الگوریتم های ممتیک و ازدحام ذرات و امثال آنها بصورت منفرد، این الگوریتم کارایی قابل قبولی از خود نشان داده است. ولی هنوز امکان افزایش سرعت همگرایی وجود دارد. امکان استفاده از روش هایی از قبیل نظریه آشوب^{۱۶} برای تولید جمعیت اولیه همچنان می توانند سرعت همگرایی را افزایش دهند.

مراجع

- [1] Cotta, C., Fernandez, A.J., Gallardo, J.E., "On the Hybridization of Memetic Algorithms With Branch-and-Bound Techniques", IEEE Transactions on Systems, Man and Cybernetics, vol. 37, no. 1, pp. 77-83, Feb.2007.