

بهینه‌سازی ازدحام ذرات به روش مدل مخلوط گوسی در محیط پویا

نرگس میرزائی^۱، دانشجوی کارشناسی ارشد؛ بهزاد زمانی دهکردی^۲، استادیار؛ فرشاد کیومرثی^۳، استادیار

۱- گروه مهندسی کامپیوتر- دانشکده فنی و مهندسی- واحد شهرکرد- دانشگاه آزاد اسلامی- شهرکرد- ایران - narges.mirzaeian@gmail.com

۲- گروه مهندسی کامپیوتر- دانشکده فنی و مهندسی- واحد شهرکرد- دانشگاه آزاد اسلامی- شهرکرد- ایران - bzamani@iaushk.ac.ir

۳- گروه مهندسی کامپیوتر- دانشکده فنی و مهندسی- واحد شهرکرد- دانشگاه آزاد اسلامی- شهرکرد- ایران - fkumarci@iaushk.ac.ir

چکیده: بسیاری از مسائل در دنیای واقعی به صورت یک مسئله بهینه‌سازی با ماهیتی پویا هستند به عبارتی بهینه‌های سراسری و محلی در طول زمان تغییر می‌کنند؛ از این رو در این محیط‌ها نیاز به الگوریتم‌های بهینه‌سازی است که علیرغم پیدا کردن بهینه سراسری، بتواند بهینه‌های در حال تغییر را در طول زمان دنبال کند. دو مشکل طراحی الگوریتم بهینه‌سازی ازدحام ذرات برای محیط‌های پویا، به منظور داشتن قابلیت‌های پیدا کردن راه‌حل بهینه در زمان کوتاه و پیگیری راه‌حل بعد از تغییرات محیطی عبارتند از: حافظه منسوخ شده و از دست دادن تنوع جمعیت در فضای جستجو. مشکل از دست دادن تنوع جمعیت یکی از چالش‌های اصلی در محیط‌های پویا می‌باشد، زیرا متنوع‌سازی یک گروه همگرا شده برای یافتن بهینه متحرک و سپس همگرایی مجدد آن به بهینه جدید، شدیداً کارایی الگوریتم را کاهش می‌دهد. با توجه به چالش‌های مطرح شده در این مقاله الگوریتم ترکیبی بهینه‌سازی ازدحام ذرات مبتنی بر مدل مخلوط گوسی پیشنهاد شده است. در روش پیشنهادی تغییر هر ذره براساس برآیند بهترین ذرات در هر خوشه متناسب با تعلق ذره به هر خوشه تغییر خواهد کرد. نتایج حاصل از آزمایشات روی محیط پویای قله‌های متحرک عملکرد بهتر الگوریتم پیشنهادی نسبت به سایر الگوریتم‌ها را نشان می‌دهد.

واژه‌های کلیدی: بهینه‌سازی، محیط‌های پویا، الگوریتم بهینه‌سازی ازدحام ذرات، مدل مخلوط گوسی، معیار قله‌های متحرک.

An Improved Particle Swarm Optimization Algorithm using Gaussian Mixture Model in Dynamic Environment

N. Mirzaeian¹, MSc Student; B. Zamani Dehkordi², Assistant Professor; F. Kumarci³, Assistant Professor

1- Department of Computer Engineering, Faculty of Engineering and Science, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran, Email: narges.mirzaeian@gmail.com

2- Department of Computer Engineering, Faculty of Engineering and Science, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran, Email: bzamani@iaushk.ac.ir

3- Department of Computer Engineering, Faculty of Engineering and Science, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran, Email: fkumarci@iaushk.ac.ir

Abstract: Many problems in the real world due to the local and global optimization change over time are a matter of dynamic optimization. Therefore, optimization algorithms despite global optimization and tracking of environments changed over time are needed in these environments. We are faced with two problems in designing the particle swarm optimization algorithm for dynamic environments to find the best solution in a short time and follow the solution after the environmental changes: Outdated memory and loss of population diversity in search area. The problem of loss of population diversity is one of the biggest challenges in dynamic environments because diversifying a convergent population to find dynamic optimization and then turning it into a new optimization greatly reduces the efficiency of the algorithm. Given the challenges presented in this paper, a hybrid particle swarm optimization algorithm based on a Gaussian mixture model is proposed. In the proposed method, the change of each particle is based on the result of the best particles in its cluster. The results of the Moving Peak Benchmark experiments show a better performance of the proposed algorithm than other algorithms.

Keywords: Optimization, dynamic environments, particle swarm optimization algorithm, gaussian mixture model, moving peak benchmark.

تاریخ ارسال مقاله: ۱۳۹۷/۰۷/۰۸

تاریخ اصلاح مقاله: ۱۳۹۸/۰۶/۲۸

تاریخ پذیرش مقاله: ۱۳۹۹/۰۱/۰۴

نام نویسنده مسئول: دکتر بهزاد زمانی دهکردی

نشانی نویسنده مسئول: ایران - شهرکرد - رحمتیه - دانشگاه آزاد اسلامی شهرکرد - دانشکده فنی و مهندسی - گروه مهندسی کامپیوتر

۱- مقدمه

بهینه‌سازی یکی از مهم‌ترین مسائل موجود در علوم مهندسی و ریاضیات به‌شمار می‌رود. بهینه‌سازی بدین مفهوم است که در بین پارامترهای یک تابع به دنبال مقادیری بوده که تابع را کمینه یا بیشینه کنند. کلیه مقادیر مناسب جهت این امر را، راه‌حل‌های ممکن و بهترین مقدار از این مقادیر را راه‌حل بهینه می‌نامند. بسیاری از مسائل بهینه‌سازی در دنیای واقعی پویا هستند؛ به این معنی که تابع هدف و محدودیت‌ها، می‌تواند در طول زمان تغییر کند. از آنجایی که بهینه‌این مسائل با تغییرات محیطی در طول زمان جابه‌جا می‌شوند، بنابراین در حل آن‌ها یافتن بهینه سراسری کافی نیست بلکه پیگیری بهینه در طول تغییرات اهمیت دارد [۱].

در محیط ایستا، هیچ رکورد (سابقه‌ای) برای تغییرات بهینه نگهداری نمی‌شود در عوض فقط بهینه مکان‌یابی می‌شود. ولی در محیط پویا نه تنها بهینه چندان و متعدد مکان‌یابی می‌شود بلکه باید مسیر تغییرات بهینه در فضای جستجو پیگیری گردد [۲]. در سال‌های اخیر استفاده از الگوریتم‌های تکاملی و هوش جمعی برای حل این مسائل توجه زیادی را به خود جلب کرده است. یکی از روش‌های متعلق به الگوریتم‌های تکاملی، الگوریتم بهینه‌سازی ازدحام ذرات می‌باشد که یک روش بهینه‌سازی تصادفی مبتنی بر جمعیت چندمنظوره است. این روش برای اولین بار توسط کندی و ابرهارت پیشنهاد شده است [۳].

الگوریتم بهینه‌سازی ازدحام ذرات مانند الگوریتم‌های تکاملی نه تنها باید برای پیدا کردن راه‌حل‌های بهینه در یک زمان کوتاه اصلاح شود، بلکه باید قادر به پیگیری راه‌حل، بعد از تغییراتی که در محیط رخ می‌دهد، باشد [۴]. به منظور داشتن این قابلیت‌ها چالش‌های اساسی و مهمی پیش‌روی این الگوریتم در محیط‌های پویا وجود دارد: یک مشکل اصلی این الگوریتم در محیط پویا، از دست دادن تنوع پس از مدتی است که باعث همگرایی زودرس و گیرافتادن در بهینه محلی می‌شود و مشکل دیگر به‌خصوص برای محیط‌های پویا، بلا استفاده شدن حافظه (حافظه منسوخ‌شده) بعد از تغییر محیط می‌باشد. این دو مشکل کارایی الگوریتم را هنگامی که محیط تغییر می‌کند شدیداً کاهش می‌دهد.

حافظه منسوخ شده به وضعیتی در حافظه ذرات اشاره دارد که با توجه به تغییرات در محیط ظاهر می‌شود. هنگامی که یک تغییر در محیط رخ می‌دهد، شایستگی موقعیت‌های سابق پیدا شده ممکن است دیگر معتبر نباشند و جمعیت می‌تواند اطلاعات گمراه‌کننده‌ای را ارائه دهد [۱]. مشکل حافظه منسوخ شده معمولاً به یکی از دو روش ارزیابی مجدد حافظه^۲ یا فراموشی حافظه^۴ حل می‌شود. بنابراین راه‌حل‌های به دست آمده موجود در حافظه، دیگر معتبر نمی‌باشند، یا باید آن‌ها را به‌طور کامل فراموش کرد یا باید دوباره آن‌ها را ارزیابی کرد.

بین این دو مشکل، از دست دادن تنوع بسیار جدی‌تر است و اثرات زیان‌آورتری بر روی عملکرد دارد [۱]. روش‌های زیادی برای ایجاد تنوع یا نگهداری تنوع گروه در محیط وجود دارد. در نظر گرفتن هر تغییر

به‌عنوان ورودی یک مسئله بهینه‌سازی جدید، می‌تواند ساده‌ترین راه برای واکنش در برابر تغییرات محیط باشد که باید از ابتدا حل شود. در صورتی که زمان انجام کار مهم نباشد، این روش یک گزینه مناسب است اما در بیشتر مواقع، زمان ارزش زیادی داشته و این راه‌حل مناسبی نمی‌باشد. یک تلاش طبیعی برای سرعت بخشیدن به فرایند بهینه‌سازی بعد از یک تغییر، این است که از اطلاعات قبلی برای محیط جدید استفاده کرد. سؤال اساسی که در این جا مطرح می‌شود این است که چه اطلاعاتی باید نگهداری شود و این اطلاعات چگونه باید در محیط جدید مورد استفاده قرار گیرد تا فرایند جستجو به‌درستی انجام پذیرد. مسئله دیگر این است که در بیشتر الگوریتم‌های بهینه‌سازی، پس از اینکه الگوریتم به یک نقطه همگرا شد، تنوع خود را از دست می‌دهد که این امر باعث کاهش قابلیت انطباق و سازگاری با محیط می‌شود. بنابراین در کنار انتقال اطلاعات باید به دنبال راه کارهایی برای افزایش تنوع و سازگاری الگوریتم پس از تغییر محیط بود.

تاکنون روش‌های مختلفی برای حل مسائل بهینه‌سازی با استفاده از روش‌های تکاملی و هوش جمعی ارائه شده‌اند ولی با مشکل حافظه غیرمعتبر و از دست دادن تنوع روبرو هستند. از متداول‌ترین این روش‌ها، تکنیک K-mean می‌باشد که یک خوشه‌بندی مبتنی بر فاصله است. یکی از مشکلات اصلی که در این الگوریتم وجود دارد به دام افتادن در مینیمم محلی است، وابسته به مقداردهی اولیه، وابسته به معیار فاصله و حساس به نویز است.

در این مقاله غلبه بر چالش کاهش تنوع و سازگاری الگوریتم بهینه‌سازی ازدحام ذرات پس از تغییرات محیط، الگوریتم ترکیبی جدیدی مبتنی بر الگوریتم بهینه‌سازی ازدحام ذرات و مدل مخلوط گوسی^۵ که یکی از انواع روش‌های خوشه‌بندی است (روش GMM به‌عنوان یک روش کلاس‌بندی شناخته شده است و به‌ندرت در مقالات به‌عنوان روش خوشه‌بندی به کار گرفته شده است [۵]) که از اندازه‌گیری توزیع احتمالی برای ایجاد خوشه استفاده می‌کند، پیشنهاد شده است. خوشه‌بندی kmeans به دلیل سادگی در استفاده و همگرایی سریع نسبت به سایر روش‌های EM^۶ دیگر حتی با وجود تعداد زیادی متغیر در بسیاری از کاربردها مورد استفاده قرار می‌گیرد. از طرفی kmeans به‌خوبی خوشه‌ها با اندازه، شکل و چگالی متفاوت را نمی‌یابد. GMM روی داده‌ها با اندازه و چگالی متفاوت به‌خوبی کار می‌کند. همچنین، بسیار سریع است. GMM از همان مجموعه وزن مخلوط برای همه نقاط داده یک خوشه خاص استفاده می‌کند.

الگوریتم پیشنهادی بر روی سناریوهای مختلف محک قله‌های متحرک (MPB)^۷ که از معروف‌ترین محک‌های محیط‌های پویا است آزمایش شده و کارایی آن با الگوریتم‌های AmQSO، mPSO، HmSO، cellularDE، CDEPSO، mNAFSA، CESO، SFA، FTMPs، APSO، PSO-AQ، DPSABC، CbDE-wCA، MLDE-S، DynPopDE، Cellular PSO، FMSO، kmeansPSO، HdPSO و MQSO مقایسه شده است. معیار مقایسه خطای برون خطی (آفلاین) است که یکی از

۲-۲- مروری بر PSO بهبودیافته

از بین چالش‌های موجود در محیط‌های پویا مهم‌ترین چالش مسئله از دست‌دادن تنوع می‌باشد به همین دلیل در ادامه کارهایی که در گذشته برای حل این مشکل ارائه شده است مورد بررسی قرار می‌گیرد. روش‌های بهبودیافته PSO را می‌توان از منظر به سه دسته روش‌های تزریق تنوع به گروه پس از تشخیص تغییر در محیط، روش‌های حفظ تنوع در سراسر اجرا و روش‌های مبتنی بر حافظه تقسیم نمود.

- روش‌های تزریق تنوع به گروه پس از تشخیص تغییر در محیط یک استراتژی بسیار اساسی برای مقابله با مشکل از دست دادن تنوع، تزریق یک سطح معینی از تنوع به گروه هر زمان که یک تغییر در محیط رخ می‌دهد می‌باشد. یک مثال از این روش تصادفی‌سازی مجدد PSO (RPSO) می‌باشد که توسط هو و ابرهات [۱۰] پیشنهاد شد، با تشخیص تغییر در محیط، کل و یا قسمتی از گروه بطور تصادفی در فضای جستجو نقل مکان می‌کند.

شکرانی‌پور و همکارانش یک الگوریتم بهینه‌سازی تعاونی PSO به نام ACP^۹SO پیشنهاد داده‌اند که پارامترهای اینرسی و ضرایب شتاب را برای هر بعد در هر حلقه به صورت تطبیقی تازه‌سازی می‌کند. آن‌ها با ترکیب دو روش APSO و CPSO از مزایای ساختار تطبیقی APSO و ساختار تعاونی CPSO به صورت توأمان استفاده نمودند. روش ACP^۹SO باعث افزایش تنوع جمعیت می‌گردد [۱۱].

- روش‌های حفظ تنوع در سراسر اجرا یک گروه از مطالعات برای حفظ یک سطح پایدار از تنوع در طول کل اجرا با اجتناب از همگرایی ذرات به یک حالت مطلوب (بهینه) تلاش کرده است [۳]. به عنوان مثال یانگ الگوریتم ژنتیک مهاجران تصادفی (RIGA^{۱۰}) را پیشنهاد داد که در هر نسل، بخشی از جمعیت توسط افراد به طور تصادفی تولید شده جایگزین شده است. که این امر می‌تواند گرایش یک جمعیت را به یک ناحیه محدود از فضای جستجو کاهش دهد، در نتیجه تنوع افزایش می‌یابد [۱۲].

استفاده از دفع متقابل بین ذرات، گروه‌ها و یا بهینه‌های تصرف شده می‌تواند یک سطح مناسب از تنوع را در حین فرآیند جستجو حفظ کند. مدل اتمی [۱۳، ۱۴] در بهینه‌سازی ازدحام ذرات چندگروهی مشارکتی (mCPSO^{۱۱}) استفاده شده که یک مثال از استفاده دفع در محیط‌های پویا است. در mCPSO هر گروه شامل تعدادی از ذرات کلاسیک باردار است. ذرات کلاسیک باردار در هرگروهی یکدیگر را برای فراهم کردن یک تنوع پایدار در میان افراد دفع می‌کند.

برخی از محققان توپولوژی به اشتراک‌گذاری اطلاعات را در میان ذرات گروه به منظور بهبود عملکرد PSO در برخورد با وضعیت‌های پویا را اصلاح کرده‌اند. این می‌تواند به طور موقت تمایل ذرات را به سمت بهترین موقعیت یافت‌شده کاهش دهد. در نتیجه تنوع در میان ذرات افزایش می‌یابد. به عنوان مثال، جانسون و میدندروف [۱۵] یک PSO سلسله مراتبی (H-PSO^{۱۲}) را برای محیط‌های پویا و نویزی مورد مطالعه قرار دادند. H-PSO یک ساختار درخت‌مانند برای ایجاد یک سلسله

معیارهای اصلی مقایسه الگوریتم‌های طراحی شده برای محیط‌های پویا می‌باشد. نتایج آزمایشات نشان می‌دهد که الگوریتم پیشنهادی از کارایی قابل قبولی برخوردار است.

ساختار ادامه مقاله به این صورت تنظیم شده است. بخش دوم مروری بر الگوریتم PSO و روش‌های بهبودیافته PSO را شامل می‌شود. در بخش سوم ابتدا به معرفی محیط‌های پویا و در ادامه به مروری بر الگوریتم PSO بهبودیافته در محیط پویا پرداخته می‌شود. بخش چهارم الگوریتم پیشنهادی را تشریح می‌کند. بخش پنجم به معرفی معیارهای ارزیابی برای سنجش کارایی الگوریتم‌ها در محیط‌های پویا و آزمایشات انجام گرفته بر روی روش پیشنهادی و نتایج حاصل از آن می‌پردازد. در نهایت بخش ششم نتیجه‌گیری کلی از این مقاله را بیان می‌کند.

۲- بررسی روش‌های بهبودیافته الگوریتم بهینه‌سازی ازدحام ذرات

۲-۱- الگوریتم بهینه‌سازی ازدحام ذرات

الگوریتم‌های بهینه‌سازی ازدحام ذرات، الگوریتم‌های فراابتکاری مبتنی بر جمعیت الهام گرفته از طبیعت هستند که در اصل توسط کندی و ابرهات و شی به رسمیت شناخته شد [۶] و کاربردهای فراوانی در انتخاب ویژگی در کلاس‌بندی [۷]، پردازش ویدئو [۸]، بهبود پایداری سیستم قدرت [۹] و مسیریابی وسایل نقلیه دارد. این روش از رفتار دسته‌جمعی ماهی‌ها و پرندگان برای یافتن غذا الهام می‌گیرد. PSO با جمعیتی از ذرات تصادفی تولید شده در یک فضای جستجوی d بعدی آغاز می‌شود. سپس برای یافتن جواب بهینه در فضای مسئله با به روز کردن موقعیت و سرعت هر ذره به جستجو می‌پردازد. هر ذره i دارای سه ویژگی می‌باشد: $\vec{x}_i$ موقعیت فعلی ذره i را در فضای جستجو نشان می‌دهد. $\vec{v}_i$: سرعت ذره i است. $\vec{p}_i$ بهترین موقعیتی که تاکنون توسط ذره i یافته شده را نشان می‌دهد. هر ذره i، موقعیتش را در فضای جستجو در هر مرحله زمانی t با توجه به رابطه‌های زیر به روزرسانی می‌کند [۳].

$$\vec{v}_i(t+1) = w\vec{v}_i(t) + c_1r_1[\vec{p}_i(t) - \vec{x}_i(t)] + c_2r_2[\vec{p}_g(t) - \vec{x}_i(t)] \quad (1)$$

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \quad (2)$$

که در آن w پارامتر وزن اینرسی است که بخشی از سرعت حفظ شده در طول آخرین مرحله زمانی الگوریتم را تعیین می‌کند. c_1 و c_2 ضرایب شتاب هستند و فاکتورهای یادگیری اجتماعی و شناختی را مشخص می‌کنند و برای تنظیم درجه حرکت ذرات به ترتیب به سمت بهترین موقعیت شخصی خود و بهترین موقعیت سراسری گروه، استفاده می‌شود. r_1 و r_2 دو متغیر تصادفی مستقل با احتمال یکنواخت [0,1] هستند. $\vec{p}_g$ بهترین موقعیت سراسری که تاکنون توسط گروه یافت شده است [۳].

می‌شوند. علاوه بر این یک روش نوین مبتنی بر تغییر بردار سرعت و موقعیت ذرات برای افزایش تنوع گروه‌ها مطرح شد. الگوریتم برای جستجوی محلی از ذره کوانتوم قابل تطبیق اطراف بهترین موقعیت یافت شده استفاده می‌کند و مکانیزم خواب-بیدارنوین که برای تمرکز بر روی فشار کلی الگوریتم بر روی قله بهینه جهانی می‌باشد نیز استفاده شده است [۲۰].

هاشمی و میبدی با بهره‌گیری از تعاملات محلی در اتاماتای سلولی و تقسیم کردن جمعیت ذرات در داخل سلول‌های اتاماتای سلولی و مشخص کردن تعداد معینی ذره در هر سلول سعی بر حفظ تنوع در جمعیت نمودند [۲۱].

لی و یانگ یک روش چندگروهی سریع (FMSO)^{۱۶} را معرفی کردند که پراکندگی ذرات را در حین اجرا حفظ می‌کند. در این روش گروهی از ذرات والد میزان پراکندگی ذرات را حفظ می‌کند و نواحی نویدبخش فضای جستجو را با استفاده از الگوریتم برنامه‌ریزی تکاملی سریع شناسایی می‌کنند و دسته‌ای از گروه‌های ذرات فرزند، ناحیه محلی را با استفاده از الگوریتم بهینه‌سازی گروه ذرات سریع با هدف یافتن بهینه مسئله جستجو می‌کنند [۲۲].

شنگ یانگ و همکاران در سال ۲۰۱۰، PSO-CP را ارائه دادند. یک مدل PSO جدید که PSO با ذرات مرکب (ترکیبی) نامیده شده است تا به مشکلات بهینه‌سازی پویا رسیدگی کند [۲۳].

یک الگوریتم تکامل تفاضلی مبتنی بر خوشه جدید توسط هالدر و همکاران پیشنهاد شد. در این الگوریتم جمعیت‌های متعدد به صورت دوره‌ای توسط روش خوشه‌بندی K-mean تولید شده است و تعداد خوشه‌ها کاهش می‌یابد و یا در محدوده زمانی با توجه به عملکرد الگوریتم افزایش می‌یابد. زمانی که یک خوشه همگرا شد، آن با بهترین فرد ذخیره‌شده در یک آرشیو خارجی حذف می‌شود. زمانی که یک تغییر شناسایی شود، همه جمعیت به اندازه اولیه بازمی‌گردد و دوباره خوشه‌بندی می‌شود [۲۴].

۳- بررسی روش‌های بهبودیافته الگوریتم بهینه‌سازی ازدحام ذرات در محیط‌های پویا

در بسیاری از مسائل بهینه‌سازی دنیای واقعی، تابع هدف یا محدودیت‌ها می‌توانند در طول زمان تغییر یابند و بنابراین بهینه این مسائل نیز می‌تواند تغییر یابد. در ادبیات بهینه‌سازی در محیط‌های پویا، محققان معمولاً مسائل بهینه‌سازی که در طول زمان تغییر می‌کنند به عنوان مسائل پویا یا مسائل وابسته به زمان تعریف می‌کنند [۲۵].

۳-۱- مروری بر PSO بهبودیافته در محیط پویا

در بین روش‌های ارائه‌شده برای حل مسائل بهینه‌سازی پویا روش‌هایی که از استراتژی ترکیبی استفاده می‌کنند از جایگاه ویژه‌ای

مراتب بین ذرات را استفاده می‌کند. آن‌ها همچنین یک سلسله مراتبی جزءبندی‌شده (PH-PSO)^{۱۷} را پیشنهاد دادند. که در آن درخت به‌طور موقت به زیرگروه‌هایی پس از تغییرات شناسایی‌شده محیط تقسیم می‌شود.

• روش‌های مبتنی بر حافظه

استفاده از طرح حافظه برای بازگرداندن اطلاعات مفید در مورد موقعیت‌های سابق پیداشده در فضای جستجو می‌تواند برای مسائل بهینه‌سازی پویا مختلف مفید باشد. به عنوان مثال، وانگ و همکاران [۱۶] یک PSO مبتنی بر حافظه برای محیط‌های پویا پیشنهاد دادند. در این روش، کل جمعیت به سه بخش تقسیم می‌شود: «کشف»، جمعیتی که مسئول جمع‌آوری اطلاعات در مورد راه‌حل خوب است. «حافظه»: جمعیتی که برای ذخیره و بازیابی اطلاعات در مورد نواحی امیدوارکننده فضای جستجو است، «بهره‌برداری»: جمعیتی که برای اصلاح راه‌حل‌ها در اطراف نواحی ارائه‌شده توسط حافظه استفاده شده است.

در مقالات دیگر به بهبود PSO از دیدگاه‌های دیگر پرداخته شده است که در ادامه به مطالعه اجمالی آن‌ها پرداخته می‌شود. بیشترین تکنیک استفاده‌شده برای مقابله با مشکل از دست دادن تنوع استفاده از گروه‌های چندگانه یا تقسیم ذرات از گروه ذرات به چندین زیرگروه و اختصاص آن‌ها به مناطق مختلفی از فضای جستجو است [۳].

الگوریتم MQSO^{۱۸} که به الگوریتم ازدحام ذرات مبتنی بر ذرات کوانتوم معروف است توسط برانک و بلیکول پیشنهاد شد. در این الگوریتم کل جمعیت به چند گروه تقسیم می‌شود و شامل سه عملگر تنوع با نام‌های ذرات کوانتوم، دفع و ضد همگرایی می‌باشد. ذرات کوانتوم در موقعیت‌های تصادفی قرار می‌گیرند تا تنوع گروه‌ها را حفظ کنند. عملگر دفع هنگامی که دو گروه همپوشانی پیدا می‌کنند، گروه بدتر را مقداره‌ای اولیه مجدد می‌کند. عملگر ضد همگرایی هنگامی که تمام گروه‌ها همگرا می‌شوند، گروه بدتر را مقداره‌ای اولیه مجدد می‌کند [۱۷].

سپس بلیک و همکاران AMQSO^{۱۵} را مطرح کردند که در آن تعداد گروه‌ها از ابتدا معین نیست و با پیدا کردن قله‌های جدید افزایش می‌یابد، به این صورت که در آن عملگر ضد همگرایی هرگاه که همه گروه‌ها همگرا شدند یک گروه آزاد جدید ایجاد کرده که به پیدا کردن بهینه محلی جدید کمک می‌کند [۱۸].

کاموسی و همکاران یک الگوریتم چندگروهی را برای محیط‌های پویا پیشنهاد دادند که مسئله از دست دادن تنوع با معرفی دو نوع گروه بررسی شد. یک گروه والد، که فضای جستجو را برای پیدا کردن نقاط امیدوارکننده حاوی بهینه محلی و چندین گروه فرزند غیرهمپوشان که هرکدام مسئول بهره‌برداری از نقاط امیدوارکننده یافت شده توسط گروه والد است، بررسی می‌کند [۱۹].

یزدانی و همکاران یک الگوریتم جدید مبتنی بر روش بهینه‌سازی ذرات مطرح کردند که در آن از یک روش چندگروهی بهبود یافته استفاده شده که در آن گروه‌ها به دو دسته یابنده و دنبال کننده تقسیم

محلی محتمل است. برای دستیابی به این هدف مکانیزم یادگیری در PSO استاندارد به‌صورتی تغییر می‌کند که هر ذره اطلاعات را از بهترین موقعیت گذشته‌اش و بهترین موقعیت گذشته نزدیک‌ترین همسایه‌هایشان بیشتر از بهترین سراسری کسب می‌کند. این استراتژی ذرات را قادر می‌سازد ناحیه جستجوی محلی پیرامون خود را تشخیص داده و آن‌ها را در همسایگان مختلف اختصاص دهند. با استفاده از روش خوشه‌بندی، کل ازدحام را می‌توان به‌صورت تطبیقی به تعدادی زیرازدحام تبدیل کرده که نواحی مختلف محلی را در برمی‌گیرد.

کندی [۲۷] پیشنهاد کرد که در PSO از الگوریتم خوشه‌بندی Kmeans برای شناسایی مراکز خوشه‌های مختلف ذرات در جمعیت استفاده کند، سپس از مراکز خوشه‌ها برای جایگزینی بهترین‌های هر ذره یا بهترین‌های محلی استفاده می‌کند. بریتز و همکارانش [۲۸] الگوریتم nest PSO را پیشنهاد کرد که همسایه یک ذره را به‌عنوان نزدیکترین ذرات آن در جمعیت تعریف می‌کند. بهترین همسایه برای هر ذره به‌عنوان میانگین موقعیت نزدیکترین ذرات آن تعریف می‌شود. در خوشه‌بندی K-mean نقاط اولیه به‌صورت تصادفی انتخاب می‌شود پس اجراهای مختلف با مجموعه داده‌های یکسان، نتایج متفاوتی را حاصل می‌کند. همین مسئله گاهی سبب می‌شود این خوشه‌بندی به سمت مینیمم محلی همگرا شود که از معایب اصلی این خوشه‌بندی است و باید آن را به سمت مینیمم سراسری برد. همچنین از دیگر معایب این الگوریتم، نیاز داشتن به تعیین تعداد خوشه‌ها از ابتدا، حساس بودن به داده‌های نویزی و پرت، وابستگی نتایج نهایی به مقداردهی مراکز اولیه و تعداد خوشه‌ها، گیرافتادن الگوریتم در بهینه محلی و همگرایی زودرس را می‌توان نام برد. بنابراین این الگوریتم علیرغم موفقیت در حل بسیاری از مسائل خوشه‌بندی به‌دلیل مشکلات فوق از حل بسیاری از مسائل ناتوان خواهد بود و از قدرت عملکرد بالایی برخوردار نمی‌باشد.

۴- الگوریتم پیشنهادی

در این مقاله به‌منظور بهبود بهینه‌سازی ازدحام ذرات و افزایش تنوع و سازگاری الگوریتم پس از تغییرات محیط، الگوریتم ترکیبی جدیدی مبتنی بر الگوریتم بهینه‌سازی ازدحام ذرات و مدل مخلوط گوسی (GMPSO) که یکی از انواع روش‌های خوشه‌بندی است پیشنهاد شده است.

۴-۱- مدل مخلوط گوسی

یک تابع چگالی احتمالاتی پارامتری است که مجموع وزن‌دار چگالی مولفه‌های گوسی، را نمایش می‌دهد این مدل جهت بیشینه‌کردن شاخص درست‌نمایی یا احتمال، از تثبیت برخی پارامترها در هر بار تکرار استفاده می‌کند. پارامترهای مربوط به آن با استفاده از داده‌های آموزشی و به کارگیری الگوریتم تکراری EM^{۱۳} (بیشینه‌سازی امیدریاضی) تخمین زده می‌شوند که تخمینی از مدل ابتدایی به‌وسیله

برخوردارند. از جمله این روش‌ها می‌توان به روش ترکیبی مبتنی بر الگوریتم بهینه‌سازی ازدحام ذرات و الگوریتم فرهنگی اشاره کرد. ایده اصلی مطرح شده در این روش جستجوی بهینه سراسری با استفاده از الگوریتم بهینه‌سازی ذرات است و برای ایجاد تنوع و عدم همگرایی زودرس دسته ذرات عملگر جهش هدایت‌شده‌ای مبتنی بر دانش ذخیره شده در فضای باور الگوریتم فرهنگی معرفی شده است. علاوه بر آن با به‌کاربردن دانش تاریخچه فضای باور الگوریتم فرهنگی، سعی در پیش‌بینی حرکت قله‌ها در طی فرآیند بهینه‌سازی می‌کند. الگوریتم فرهنگی توسط رینلدس در سال ۱۹۹۴ مطرح شد. این الگوریتم از تکامل فرهنگ انسان‌ها و تأثیرپذیری افراد یک جامعه از آن و اثر آن در ایجاد نسل‌های آینده الهام گرفته شده است. همچنین از حوزه دانش برای فرآیند جستجو استفاده می‌کند. اضافه‌شدن حوزه دانش در بهبود کارایی الگوریتم‌های تکاملی مؤثر است و فرآیند جستجو را هوشمندانه‌تر می‌کند [۲۶].

روش ترکیبی دیگری که می‌توان اشاره کرد الگوریتم ترکیبی مبتنی بر الگوریتم بهینه‌سازی ازدحام ذرات و روش Kmeans که یکی از انواع روش‌های خوشه‌بندی مبتنی بر فاصله است. یکی از چالش‌های اصلی در محیط‌های پویا، حفظ تنوع در زمان پیاده‌سازی الگوریتم می‌باشد که تنوع جمعیت با استفاده از خوشه‌بندی حافظه و جمعیت حفظ گردید. در این جا حافظه و جمعیت در مرحله اول، مقداردهی می‌شوند. پس از مقداردهی، خوشه‌بندی می‌شوند. هر عضو جمعیت، برای درج و بازیابی باید در خوشه خود قرار داده شود. خوشه‌بندی جاری، یکی از تکنیک‌های یادگیری غیرنظارتی می‌باشد که در آن، نمونه‌ها، به مجموعه‌ای از خوشه‌ها تقسیم می‌شوند، به‌طوری که نمونه‌های یک خوشه، مشابه و نمونه‌های خوشه‌های مختلف، با هم متفاوت باشند. معیارهای زیادی برای اندازه‌گیری شباهت وجود دارد که یکی از آن‌ها، فاصله اقلیدسی بین دو نمونه می‌باشد. نمونه‌های مشابه دارای فاصله اقلیدسی کمتری بوده و در یک خوشه قرار می‌گیرند. این خوشه‌بندی، خوشه‌بندی مبتنی بر فاصله، نامیده می‌شود. یکی از خوشه‌بندی‌های مبتنی بر فاصله، خوشه‌بندی Kmeans می‌باشد. در این روش، مرکز یک خوشه، برابر است با میانگین مقدار داده‌های درون خوشه. سپس، داده‌ها براساس فاصله اقلیدسی با مراکز خوشه، خوشه‌بندی می‌شوند. این روش، در هر تکرار، تغییرات خوشه داخلی مدل را با تخمین مرکز خوشه جدید در مرحله تکرار، بهبود می‌دهد. به این‌صورت، داده‌ها برپایه میانگین به‌روزرسانی شده، به خوشه‌های مختلف تخصیص داده می‌شوند. این کار، تا زمانی ادامه می‌یابد که مرکز خوشه، ثابت شده و مقدار آن در تکرارهای متوالی، بدون تغییر باقی بماند و خوشه‌ها تثبیت و پایدار شوند [۴].

خوشه‌بندی در PSO می‌تواند نواحی محلی را به‌صورت انطباقی تشخیص داده و ذرات را در نواحی مختلف اختصاص دهد. در PSO مبتنی بر خوشه دو استراتژی جستجو وجود دارد: جستجوی سراسری و جستجوی محلی. برای جستجوی سراسری، هدف پوشش بهینه‌سازی

$$\bar{w}_i = \frac{1}{T} \sum_{t=1}^T p(i|x_t, \lambda) \quad (6)$$

$$\bar{\mu}_i = \frac{\sum_{t=1}^T p(i|x_t, \lambda) x_t}{\sum_{t=1}^T p(i|x_t, \lambda)} \quad (7)$$

$$\bar{\Sigma}_i = \frac{\sum_{t=1}^T p(i|x_t, \lambda) \|x_t - \bar{\mu}_i\|^2}{\sum_{t=1}^T p(i|x_t, \lambda)} \quad (8)$$

درواقع E و M مکرراً و پشت‌سرهم انجام می‌پذیرد تا تغییر پارامترهای GMM به حداقل برسد و اصطلاحاً الگوریتم همگرا شود.

۴-۲- روش پیشنهادی

در روش پیشنهادی ابتدا با استفاده از الگوریتم GMM دادگان را دسته‌بندی نموده و سپس روی هر دسته الگوریتم PSO زده می‌شود. این عمل باعث می‌شود که ذرات تعریف شده در PSO در تمامی فضای مسئله پراکنده شوند و جمعیت تنوع لازم را دارا خواهد بود. سرعت هر ذره i ، در فضای جستجو در هر مرحله زمانی t با توجه به رابطه (۹) به‌روزرسانی می‌شود. همچنین موقعیت مکانی براساس رابطه (۲) به‌دست می‌آید.

$$\begin{aligned} \vec{v}_i(t+1) &= w\vec{v}_i(t) + c_1 r_1 [\vec{p}_i(t) - \vec{x}_i(t)] \\ &\quad + c_2 r_2 \sum_{k=1}^K p(C_k|\vec{x}_i(t)) [\vec{p}_k(t) - \vec{x}_i(t)] \end{aligned} \quad (9)$$

که K تعداد خوشه‌ها، $\vec{p}_k$ بهترین موقعیت سراسری که تاکنون توسط خوشه‌کام یافت شده است، $p(C_k|\vec{x}_i(t))$ احتمال تعلق ذره i به خوشه k ام می‌باشد. این خوشه‌بندی توسط GMM انجام گرفته است. c_1 و c_2 ضرایب شتاب برای تنظیم درجه حرکت ذرات به سمت بهترین موقعیت خود و بهترین موقعیت سراسری خوشه، استفاده می‌شود. r_1 و r_2 دو متغیر تصادفی مستقل با احتمال یکنواخت هستند.

Algorithm: The GMMPSO Algorithm
 Input: The set of data items $D = \{x_1, x_2, \dots, x_n\}$, the number of clusters k .
 Ensure: a partition $C = \{C_1, C_2, \dots, C_k\}$, $\bigcup_{i=1}^k C_i = D$.
 Initialization:
 Initialize the PSO parameters.
for each particle i do
 Randomly initialize position of each particle X_i in the search range (X_{min}, X_{max}).
 Randomly initialize velocity of each particle V_i .
end
 Calculate the fitness values for each particle.
 Find the personal best position of each particle.

 The GMMPSO Loop:
while (success=0 and $t \leq \text{max iterations}$) **do**
 $C \leftarrow \{C_1, C_2, \dots, C_k\}$: the GMM partition.
 Find the best position of each cluster, $\vec{p}_k(t)$, $\forall k = 1, 2, \dots, K$.
for each particle i do
 Compute $p(C_k|\vec{x}_i(t))$, $\forall k = 1, 2, \dots, K$.
 Update the velocity of $\vec{v}_i$ using equation 10.
 Update the position of $\vec{x}_i$ using equation 2.
end
 Calculate the fitness values for each particle.
 Find the best position of each particle.
 Find the Particle with the best fitness value.
 Assign this position to global best.
End

شکل ۱: الگوریتم روش پیشنهادی GMMPSO

یک داده آموزشی ایده‌آل است [۲۹]. مدل مخلوط گوسی به‌وسیله رابطه (۳) تعیین می‌شود [۳۰].

$$p(x|\lambda) = \sum_{i=1}^M w_i g(x|\mu_i, \Sigma_i) \quad (3)$$

که x بردار داده پیوسته d بعدی (که می‌تواند ویژگی‌ها یا اندازه‌ها باشد)، λ نشان‌دهنده یک نمونه اولیه متشکل از مجموعه‌ای از پارامترهای مدل $\lambda = \{w_i, \mu_i, \Sigma_i\}$ (م $\sum_{i=1}^M w_i = 1$)، M تعداد مخلوط‌ها و $g(x|\mu_i, \Sigma_i)$ تابع چگالی احتمالاتی مؤلفه‌های گوسی می‌باشد. تابع چگالی هر مؤلفه یک تابع گوسی d متغیره به‌فرم رابطه (۴) می‌باشد.

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)\right] \quad (4)$$

که μ_i بردار میانگین مخلوط i ام و $|\Sigma_i|$ دترمینان ماتریس کواریانس می‌باشند. مدل مخلوط گوسی کامل، براساس بردارهای میانگین، ماتریس‌های کواریانس و وزن‌های مخلوط چگالی تمامی اجزاء، پارامتری می‌شود. پارامترهای GMM در نهایت به‌فرم $\lambda = \{w_i, \mu_i, \Sigma_i\}$ معرفی می‌شوند. چندین تکنیک برای تخمین پارامتر GMM وجود دارد. معروف‌ترین و تعریف‌شده‌ترین روش، روش تخمین بیشینه درست‌نمایی (ML) می‌باشد. هدف تخمین ML، یافتن پارامترهایی برای مدل می‌باشد که احتمال GMM را با توجه به داده‌های آموزشی، حداکثر سازد. تقریب‌های مرتبط با بیشترین درست‌نمایی را می‌توان با استفاده از یک حالت خاص الگوریتم بیشینه‌سازی امید ریاضی (EM) به‌دست آورد. الگوریتم EM با استفاده از متغیر پنهان z از چرخه تکرار برای برآورد پارامترها استفاده می‌کند. ایده اصلی الگوریتم EM، آن است که با یک مدل اولیه λ شروع کرده و مدل جدید $\bar{\lambda}$ را به‌گونه‌ای تخمین زده که $p(x|\bar{\lambda}) \geq p(x|\lambda)$. سپس مدل جدید، به مدل اولیه‌ای برای تکرار بعد، تبدیل شده و فرایند، تا زمانی تکرار شده که به مقدار آستانه همگرایی برسد. جهت بیشینه‌سازی درست‌نمایی مابین نمونه‌ها و توزیع گوسی براساس این معادلات، پارامترهای مدل به‌صورت مکرر تغییر می‌کنند [۳۰].

الگوریتم EM از دو گام محاسبه امیدریاضی و ماکزیمم‌سازی تشکیل شده است:

۱. گام E- امید ریاضی که در آن پارامترهای GMM برای هر نمونه از داده‌های d بعدی $x \in \{x_t\}_{t=1, \dots, T}$ با استفاده از احتمال پسین و برای تأمین مؤلفه با استفاده از رابطه (۵) به‌دست می‌آید [۳۱].

$$P(i|x_t, \lambda) = \frac{w_i g(x_t|\mu_i, \Sigma_i)}{\sum_{k=1}^M w_k g(x_t|\mu_k, \Sigma_k)} \quad (5)$$

۲. گام M- بیشینه‌سازی که در آن پارامترها مطابق با احتمال پسین محاسبه‌شده در مرحله قبل تخمین زده می‌شود. $p(i|x_t, \lambda)$ احتمال پسین تابع گوسی i ام پس از مشاهده بردار x_t می‌باشد. پارامترهای GMM نیز مطابق با روابط (۶) تا (۸) به‌روز می‌شوند [۳۱].

در یک مسیر مشخص حرکت می‌کنند. تغییرات یک قله می‌تواند به‌فرم زیر توصیف شود:

$$H_i(t) = H_i(t-1) + height_{severity}\sigma \quad (12)$$

$$W_i(t) = w_i(t-1) + width_{severity}\sigma \quad (13)$$

که در آن σ به‌طور عادی یک عدد تصادفی توزیع‌شده با میانگین ۰ و تغییرات ۱ است [۳۴]. در ادامه تنظیمات پیش‌فرض مربوط به پارامترهای تابع محک قله‌های متحرک برای ارزیابی الگوریتم پیشنهادی در محیط پویا در جدول (۱) نشان داده شده است. پارامترهای MPB براساس سناریوی ۲ این مسئله تعریف شده است [۳۵].

جدول ۱: پارامترهای استاندارد برای تابع قله‌های متحرک (MPB) [۴]

پارامتر	مقدار
تعداد قله‌ها	متغیر بین ۱ تا ۲۰۰
فرکانس تغییرات	۵۰۰، ۱۰۰۰، ۲۵۰۰، ۵۰۰۰، ۱۰۰۰۰
شدت تغییرات ارتفاع قله‌ها	۷
شدت تغییرات عرض قله‌ها	۱
شکل قله‌ها	مخروطی
تابع اولیه	ندارد
S طول جابجایی	۱، ۲، ۳، ۵
تعداد ابعاد	۵
ضریب همبستگی λ	۰
محدوده مکانی قله‌ها	[۰-۱۰۰]
محدوده پارامتر ارتفاع	[۳۰-۷۰]
محدوده پارامتر عرض	[۱-۱۲]
مقدار ارتفاع اولیه قله‌ها	۵۰

۵-۲- معیارهای ارزیابی

از آنجایی که برای مسائل بهینه‌سازی پویا راه‌حل بهینه تغییرناپذیر با زمان وجود نداشته و یک هدف ثابت نمی‌تواند پیدا شود، ولی می‌توان آن را در فضای جستجو تا حد امکان نزدیک به آن دنبال کرد. برای نمایش دادن برتری یک رویکرد نسبت به سایر رویکردها بایستی ملاک‌هایی وجود داشته باشد تا از طریق بینایی یک مقایسه بین آن‌ها صورت گیرد. بنابراین از آنجا که در محیط‌های پویا بهترین راه‌حل به‌دست آمده معیار مفیدی برای ارزیابی و گزارش نمی‌باشد، از معیارهای خطای آفلاین و خطای استاندارد جهت مقایسه استفاده می‌شود. خطای آفلاین برابر است با میانگین اختلاف بین مقادیری که تاکنون توسط الگوریتم و بهینه‌ساز سری پیدا شده است. خطای برون خطی طبق رابطه (۱۵) محاسبه می‌شود [۳۶].

$$\text{Offline error} = \frac{1}{\text{FEs}} \sum_{t=1}^{\text{FEs}} (f(\text{gbest}(t)) - f(\text{globaloptimum}(t))) \quad (15)$$

استفاده از GMM به‌جای K-mean باعث رفع نواقص موجود در روش K-mean از قبیل وابستگی به خوشه‌بندی اولیه و حساسیت به نویز را نخواهد داشت و از سویی با ایجاد تنوع لازم در جمعیت PSO عملکرد این روش نیز بهبود پیدا خواهد کرد. بنابراین انتظار می‌رود که روش پیشنهادی نسبت به روش PSO و KmeansPSO عملکرد بهتری داشته باشد. الگوریتم پیشنهادی در شکل ۱ نشان داده شده است. اثبات می‌شود که EM حتماً همگرا می‌شود. همچنین سرعت همگرایی EM برای GMM با میزان همپوشانی کامپوننت‌های مخلوط گوسی ارتباط دارد. [۳۲].

۵- نتایج آزمایشات

این بخش به ارزیابی عملکرد روش پیشنهادی در مقایسه با دیگر روش‌ها اختصاص دارد. از این‌رو در ابتدا به تعریف تابع قله‌های متحرک به‌عنوان شبیه‌ساز محیط پویا پرداخته، سپس معیارهای ارزیابی بیان‌شده و در نهایت نتایج گزارش خواهند شد.

۵-۱- مسئله محک قله‌های متحرک

برای انجام آزمایش‌ها و تست کارایی روش پیشنهادی باید از تابع محکی استفاده شود که به‌خوبی بتواند یک محیط پویا را شبیه‌سازی کند. مسئله محک قله‌های متحرک توسط برانک (۱۹۹۹) پیشنهاد شد که به‌طور گسترده به‌عنوان یک معیار در موضوع بهینه‌سازی پویا مورد استفاده قرار می‌گیرد [۳۳]. MPB یک محیط پویا حقیقی با یک فضای d بعدی که متشکل از m قله است که ارتفاع و عرض و موقعیت هر قله هر زمان تغییر اندکی در محیط رخ دهد، تغییر می‌کند [۳]. یک قله را می‌توان از سه جنبه بررسی کرد: مکان، ارتفاع و عرض. برای یک فضای d بعدی مسئله به‌صورت زیر تعریف می‌شود [۳۴]:

$$F(\vec{x}, t) = \max_{i=1, \dots, p} \frac{H_i(t)}{1 + w_i(t) \sum_{j=1}^p (x_j(t) - x_{ij}(t))^2} \quad (10)$$

$w_i(t)$ و $H_i(t)$ به‌ترتیب عرض و ارتفاع قله i در زمان t هستند و $x_{ij}(t)$ عنصر λ ام از مکان قله i در زمان t است. P به‌طور مستقل قله‌های مشخص‌شده‌ای هستند که توسط تابع max با یکدیگر ترکیب شده‌اند. موقعیت هر قله در یک مسیر تصادفی توسط بردار $\vec{v}_i$ از فاصله s انتقال می‌یابد. (s طول شیفت نامیده می‌شود که شدت پویایی مسئله را تعیین می‌کند) و حرکت یک قله تکی می‌تواند به‌صورت زیر توصیف کرد:

$$\vec{v}_i(t) = \frac{s}{|\vec{r} + \vec{v}_i(t-1)|} ((1-\lambda)\vec{r} + \lambda\vec{v}_i(t-1)) \quad (11)$$

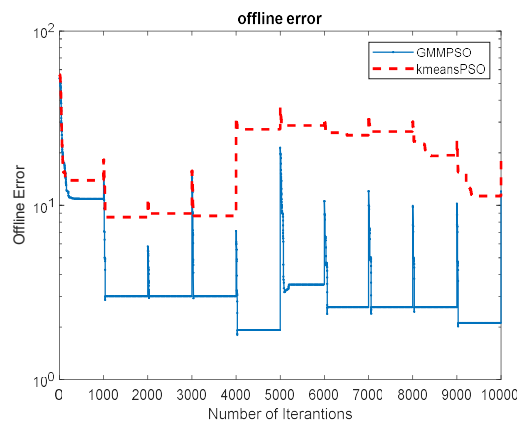
بردار انتقال $\vec{v}_i(t)$ یک ترکیب خطی از بردار تصادفی $\vec{r}$ با بردار انتقال قبلی $\vec{v}_i(t+1)$ است و به طول تغییرات s نرمال شده است. پارامتر s میزان شدت تغییرات را کنترل می‌کند. پارامتر جدید λ تعیین می‌کند که چه مقدار تغییر موقعیت یک قله بستگی به حرکت قبلی‌اش دارد. اگر $\lambda = 0$ باشد هر حرکت کاملاً تصادفی است. برای $\lambda = 1$ قله‌ها

که تعداد دفعات ارزیابی تابع برازندگی و $gbest(t)$ و $globaloptimum(t)$ به‌ترتیب بهترین موقعیت شناخته‌شده توسط الگوریتم و بهینه سراسری روی‌آمین مقدار ارزیابی است [۳۶].

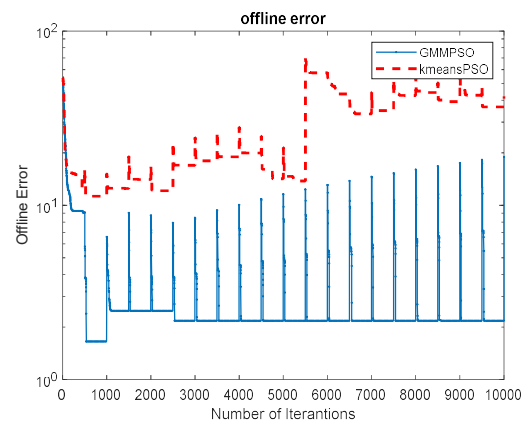
۵-۳- مقایسه روش پیشنهادی GMMPSO با دیگر روش‌ها

برای بررسی کارایی الگوریتم پیشنهادی، این الگوریتم را با الگوریتم‌های AmQSO [۱۸]، mPSO^{۱۹} [۱۹]، HmSO^{۲۰} [۳۷]، APSO^{۲۱} [۳۸]، FTMPSO^{۲۲} [۲۰]، SFA^{۲۳} [۳۹]، CESO^{۲۴} [۴۰]، CDEPSO^{۲۵} [۱]، Cellular DE^{۲۶} [۴۱]، DynPopDE^{۲۶} [۴۲]، MLDE-S^{۲۷} [۴۳]، CbDE-^{۲۸} [۴۴]، wCA^{۲۹} [۴۵]، DPSABC^{۲۹} [۴۶]، HDPSO^{۳۱} [۴۳]، KmeansPSO [۴۷]، mNAFSA [۲۱] و MQSO [۱۷، ۴۸]، مورد مقایسه قرار گرفته است. برای MQSO از پیکربندی (۵+۵q) ۱۰ استفاده شده است که در آن ده گروه ایجاد می‌شود که در آن هر یک از گروه‌ها دارای ۵ ذره خنثی و دارای ۵ ذره

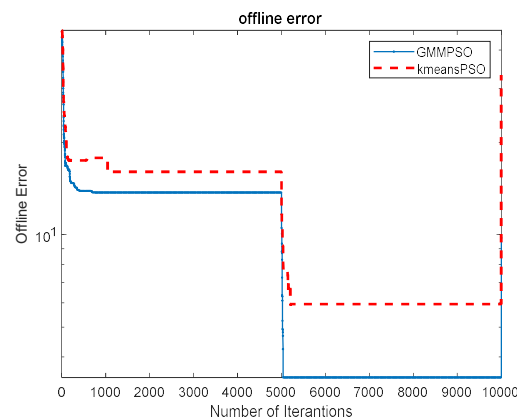
کوانتوم می‌باشند، همچنین برای این الگوریتم شعاع کوانتوم برابر ۰/۵ و شعاع دفع و شعاع همگرایی برابر ۳۱/۵ تعیین شده است. این تنظیمات پارامتر برای این الگوریتم در [۱۷، ۱۸] پیشنهاد شده است. برای الگوریتم FMSO ماکزیم تعداد گروه‌های فرزند برابر ۱۰ و شعاع دفع ما بین گروه‌های فرزند برابر ۲۵ و تعداد ذرات در گروه والد و گروه‌های فرزند به‌ترتیب برابر ۱۰۰ و ۱۰ در نظر گرفته شده است. این تنظیمات برای این الگوریتم در [۲۲] پیشنهاد شده است. در آزمایشات انجام شده، در الگوریتم پیشنهادی اندازه جمعیت به‌صورت ثابت برابر ۱۰۰، ضرایب c_1 و c_2 برابر ۱/۴۹ مقداردهی می‌شوند و وزن اینرسی w برابر ۰/۷ تعیین شده است. همچنین کران پایین و بالا به‌ترتیب ۰ و ۱۰۰ در نظر گرفته شده است. اندازه حافظه، ۱۰ و تعداد خوشه نیز ۲۰ تنظیم شده است. آزمایشات ۳۰ مرتبه با شروع تصادفی مختصات ذرات و خوشه‌بندی اولیه تکرار شده و میانگین آن‌ها گزارش شده است.



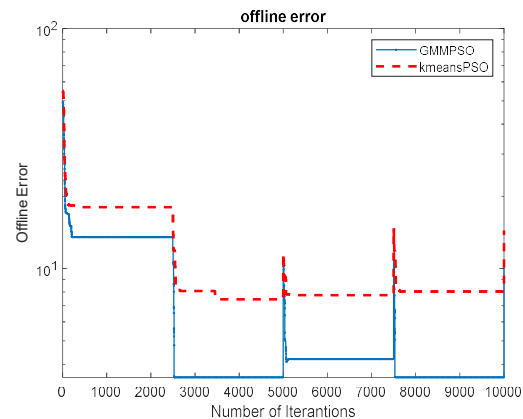
ب) فرکانس ۱۰۰۰



الف) فرکانس ۵۰۰



د) فرکانس ۵۰۰۰



ج) فرکانس ۲۵۰۰

شکل ۲: لگاریتم خطای آفلاین الگوریتم پیشنهادی (GMMPSO) و الگوریتم KmeansPSO [۴] در محیط پویا با تنظیمات جدول ۱ (دارای ۱۰ قله) در فرکانس‌های تغییر ۵۰۰، ۱۰۰۰، ۲۵۰۰ و ۵۰۰۰

در فرکانس ۵۰۰ می‌باشد. این امر نشان‌دهنده خوشه‌بندی بهتر GMM نسبت به Kmeans بوده که باعث همگرایی سریع‌تر الگوریتم و کاهش خطا می‌شود.

متوسط مقدار خطای آفلاین و خطای استاندارد [۳۵] الگوریتم پیشنهادی به همراه دیگر الگوریتم‌ها در جداول ۲ تا ۶، نشان داده شده است. برای ارزیابی بهتر بین الگوریتم‌ها از ۱ قله تا ۲۰۰ قله در فرکانس‌های تغییر ۵۰، ۱۰۰، ۲۵۰، ۵۰۰، ۱۰۰۰ در نظر گرفته شده است (بقیه پارامترها بر اساس پارامترهای جدول ۱ تحت عنوان سناریو ۲ در مسئله MPB تنظیم شده‌اند [۳۵]).

در این جداول، خطای استاندارد در کنار خطای آفلاین درون پرانتز آورده شده است. برای مقایسه بهتر، نتایج برتر به صورت ضخیم شده و خط زیر و نتایج در رتبه دوم به فرم ضخیم‌شده نشان داده شده‌اند.

تمامی آزمایش‌ها روی کامپیوتری با پردازنده Core i5-3.2GHz و حافظه 8GB در محیط نرم‌افزار matlab نسخه ۲۰۱۵ صورت گرفته است. شکل (۲) لگاریتم خطای آفلاین الگوریتم پیشنهادی GMMPSO و الگوریتم Kmeans PSO [۴] در محیط پویا در فرکانس‌های تغییر ۵۰۰، ۱۰۰۰، ۲۵۰۰ و ۵۰۰۰ با تعداد ۱۰ قله و دیگر پارامترهای استاندارد ذکر شده نشان می‌دهد. در فرکانس‌های تغییرات پایین، محیط در تعداد تکرارهای کمتری دچار تغییر می‌شود و ردیابی بهینه تغییر یافته را برای الگوریتم‌ها مشکل‌تر می‌کند. در فرکانس‌های تغییرات بالا، الگوریتم زمان بیشتری برای جستجوی بهینه تغییر یافته در اختیار دارد و در نتیجه ردیابی مورد نظر برای الگوریتم راحت‌تر است. همان‌طور که مشاهده می‌شود خطای آفلاین الگوریتم پیشنهادی، GMMPSO، نسبت به روش Kmeans PSO کمتر می‌شود و این اختلاف در فرکانس تغییر ۵۰۰ بیشتر از اختلاف

جدول ۲: مقایسه خطای آفلاین (خطای استاندارد) روی مسئله MPB با تعداد قله مختلف، فرکانس تغییر = ۵۰۰، طول جابه‌جایی = ۱

الگوریتم	تعداد قله						
	۱	۵	۱۰	۲۰	۳۰	۵۰	۱۰۰
MQSO(505q)	۳۳/۶۷(۳/۴۲)	۱۱/۹۱(۰/۷۶)	۹/۶۲(۰/۳۴)	۹/۰۵(۰/۲۵)	۸/۸۰(۰/۲۱)	۸/۷۲(۰/۲۰)	۸/۱۹(۰/۱۷)
FMSO	۲۷/۵۸(۰/۹۴)	۱۹/۴۵(۰/۴۵)	۱۸/۲۶(۰/۳۲)	۱۷/۳۴(۰/۳۰)	۱۶/۳۹(۰/۴۸)	۱۵/۳۴(۰/۴۵)	۱۲/۸۷(۰/۶۰)
AmQSO	۳/۰۲(۰/۳۲)	۵/۷۷(۰/۵۶)	۵/۳۷(۰/۴۲)	۶/۸۲(۰/۳۴)	۷/۱۰(۰/۳۹)	۷/۵۷(۰/۳۲)	۷/۴۸(۰/۱۹)
mPSO	۸/۷۱(۰/۴۸)	۶/۶۹(۰/۲۶)	۷/۱۹(۰/۲۳)	۸/۰۱(۰/۱۹)	۸/۴۳(۰/۱۷)	۸/۷۶(۰/۱۸)	۸/۹۱(۰/۱۷)
HmPSO	۸/۵۳(۰/۴۹)	۷/۴۰(۰/۳۱)	۷/۵۶(۰/۲۷)	۷/۸۱(۰/۲۰)	۸/۳۳(۰/۱۸)	۸/۸۳(۰/۱۷)	۸/۸۵(۰/۱۶)
APSO	۴/۸۱(۰/۱۴)	۴/۹۵(۰/۱۱)	۵/۱۶(۰/۱۱)	۵/۸۱(۰/۰۸)	۶/۰۳(۰/۰۷)	۵/۹۵(۰/۰۶)	۶/۰۸(۰/۰۶)
Cellular PSO	۱۳/۴۶(۰/۷)	۹/۶۳(۰/۴۹)	۹/۴۲(۰/۲۱)	۸/۸۴(۰/۲۸)	۸/۸۱(۰/۲۴)	۸/۶۲(۰/۲۳)	۸/۵۴(۰/۲۱)
FTMPSO	۱/۷۶(۰/۰۹)	۲/۹۳(۰/۱۸)	۳/۹۱(۰/۱۹)	۴/۸۳(۰/۱۹)	۵/۰۵(۰/۲۱)	۴/۹۸(۰/۱۵)	۵/۳۱(۰/۱۱)
SFA	۴/۷۲(۰/۱۲)	۴/۸۸(۰/۱۲)	۵/۱۱(۰/۱۴)	۵/۷۲(۰/۱۳)	۵/۹۷(۰/۱۲)	۵/۹۴(۰/۱۵)	۶/۱۸(۰/۱۱)
kmeansPSO	۸/۸۵(۰/۱۳)	۸/۹۳(۰/۱۷)	۸/۷۸(۰/۱۸)	۸/۴۱(۰/۱۷)	۸/۰۷(۰/۱۹)	۷/۱۷(۰/۲۱)	۷/۶۷(۰/۲۵)
GMMPSO	۷/۷۱(۰/۰۸)	۲/۸۸(۰/۲۱)	۳/۸۷(۰/۰۸)	۴/۸۶(۰/۰۹)	۵/۰۲(۰/۲۰)	۴/۹۵(۰/۱۲)	۵/۲۴(۰/۱۴)

جدول ۳: مقایسه خطای آفلاین (خطای استاندارد) روی مسئله MPB با تعداد قله مختلف، فرکانس تغییر = ۱۰۰۰، طول جابه‌جایی = ۱

الگوریتم	تعداد قله						
	۱	۵	۱۰	۲۰	۳۰	۵۰	۱۰۰
MQSO(505q)	۱۸/۶۰(۱/۶۳)	۶/۵۶(۰/۳۸)	۵/۷۱(۰/۲۲)	۵/۸۵(۰/۱۵)	۵/۸۱(۰/۱۵)	۵/۸۷(۰/۱۳)	۵/۸۳(۰/۱۳)
FMSO	۱۴/۴۲(۰/۴۸)	۱۰/۵۹(۰/۲۴)	۱۰/۴۰(۰/۱۷)	۱۰/۳۳(۰/۱۳)	۱۰/۰۶(۰/۱۴)	۹/۵۴(۰/۱۱)	۸/۷۷(۰/۰۹)
AmQSO	۲/۳۳(۰/۳۱)	۲/۹۰(۰/۳۲)	۴/۵۶(۰/۴۰)	۵/۳۶(۰/۴۷)	۵/۲۰(۰/۳۸)	۶/۰۶(۰/۱۴)	۴/۷۷(۰/۴۵)
mPSO	۴/۴۴(۰/۲۴)	۳/۹۳(۰/۱۶)	۴/۵۷(۰/۱۸)	۴/۹۷(۰/۱۳)	۵/۱۵(۰/۱۲)	۵/۳۳(۰/۱۰)	۵/۶۰(۰/۰۹)
HmPSO	۴/۴۶(۰/۲۶)	۴/۲۷(۰/۰۸)	۴/۶۱(۰/۰۷)	۴/۶۶(۰/۱۲)	۴/۸۳(۰/۰۹)	۴/۹۶(۰/۰۳)	۵/۱۴(۰/۰۸)
APSO	۲/۷۲(۰/۰۴)	۲/۹۹(۰/۰۹)	۳/۸۷(۰/۰۸)	۴/۱۳(۰/۰۶)	۴/۱۲(۰/۰۴)	۴/۱۱(۰/۰۳)	۴/۲۶(۰/۰۴)
FTMPSO	۰/۸۹(۰/۰۵)	۱/۷۰(۰/۱۰)	۲/۳۶(۰/۰۹)	۳/۰۱(۰/۱۲)	۳/۰۶(۰/۱۰)	۳/۲۹(۰/۱۰)	۳/۶۳(۰/۰۹)
SFA	۲/۴۵(۰/۱۲)	۲/۷۱(۰/۰۶)	۳/۶۴(۰/۰۴)	۴/۰۱(۰/۰۷)	۴/۰۲(۰/۰۸)	۴/۱۲(۰/۰۷)	۴/۴۰(۰/۰۷)
Cellular PSO	۶/۷۷(۰/۳۸)	۵/۳۰(۰/۳۲)	۵/۱۵(۰/۱۳)	۵/۲۳(۰/۱۸)	۵/۲۳(۰/۱۶)	۵/۵۵(۰/۱۴)	۵/۵۷(۰/۱۲)
kmeansPSO	۲/۳۹(۰/۱۲)	۲/۶۵(۰/۰۷)	۳/۵۹(۰/۰۷)	۳/۹۳(۰/۰۸)	۳/۸۹(۰/۱۰)	۴/۰۵(۰/۰۸)	۴/۲۵(۰/۰۷)
GMMPSO	۰/۸۶(۰/۰۴)	۱/۶۲(۰/۱۱)	۲/۲۹(۰/۰۷)	۳/۰۵(۰/۰۹)	۳/۰۸(۰/۰۵)	۳/۲۵(۰/۰۸)	۲/۶۰(۰/۰۸)

می‌دهد. میانگین خطای آفلاین الگوریتم پیشنهادی و سایر الگوریتم‌ها با افزایش فرکانس تغییر، یعنی افزایش فاصله تغییرات از هم، کاهش

همان‌طور که در جداول ۲ تا ۶ مشاهده می‌شود، الگوریتم (GMMPSO)، نتایج بهتری را نسبت به دیگر الگوریتم‌ها از خود نشان

کند. خوشه‌بندی ذرات براساس احتمال قرارگیری هر ذره در دسته‌های متفاوت خواهد بود و در نتیجه هر ذره می‌تواند با چندین خوشه در ارتباط باشد. افزودن الگوریتم مخلوط گوسین به الگوریتم PSO باعث افزایش دقت در ازدحام ذرات و نیز خود خوشه باعث افزایش دقت در ازدحام ذرات و نیز خود خوشه‌بندی الگوریتم مخلوط گوسین می‌شود.

پیدا می‌کند. درواقع در فرکانس‌های تغییر بالا، الگوریتم فرصت بیشتری برای جستجوی بهینه داشته و سریع‌تر می‌تواند راه‌حل‌های بهتر و بهینه را پس از مشاهده تغییر در محیط به‌دست بیاورد و کارایی بالاتری از خود نشان می‌دهد.

یکی از دلایل برتری الگوریتم پیشنهادی نسبت به بقیه الگوریتم‌ها استفاده از الگوریتم مخلوط گوسین است که درواقع نوعی خوشه‌بندی است که از اندازه‌گیری توزیع احتمالاتی برای ایجاد خوشه استفاده می‌کند.

جدول ۴: مقایسه خطای آفلاین (خطای استاندارد) روی مسئله MPB با تعداد قله مختلف، فرکانس تغییر ۲۵۰۰، طول جابه‌جایی ۱

الگوریتم	تعداد قله						
	۱	۵	۱۰	۲۰	۳۰	۵۰	۱۰۰
MQSO	۷/۶۴(۰/۶۴)	۳/۲۶(۰/۲۱)	۳/۱۲(۰/۱۴)	۳/۵۸(۰/۱۳)	۳/۶۳(۰/۱۰)	۳/۶۳(۰/۱۰)	۳/۳۰(۰/۰۸)
FMSO	۶/۲۹(۰/۲۰)	۵/۰۳(۰/۱۲)	۵/۰۹(۰/۰۹)	۵/۳۲(۰/۰۸)	۵/۲۲(۰/۰۸)	۴/۹۹(۰/۰۶)	۴/۶۰(۰/۰۵)
AmQSO	۰/۸۷(۰/۱۱)	۲/۱۶(۰/۱۹)	۲/۴۹(۰/۱۰)	۲/۷۳(۰/۱۱)	۳/۲۴(۰/۱۸)	۳/۶۸(۰/۱۵)	۳/۵۳(۰/۱۴)
mPSO	۱/۷۹(۰/۱۰)	۲/۰۴(۰/۱۲)	۲/۶۶(۰/۱۶)	۳/۰۷(۰/۱۱)	۳/۱۵(۰/۰۸)	۳/۲۶(۰/۰۷)	۳/۳۱(۰/۰۵)
HmPSO	۱/۷۵(۰/۱۰)	۱/۹۲(۰/۱۱)	۲/۳۹(۰/۱۶)	۲/۴۶(۰/۰۹)	۲/۵۷(۰/۰۵)	۲/۶۵(۰/۰۵)	۲/۷۲(۰/۰۴)
APSO	۱/۰۶(۰/۰۳)	۱/۵۵(۰/۰۵)	۲/۱۷(۰/۰۷)	۲/۵۱(۰/۰۵)	۲/۶۱(۰/۰۲)	۲/۶۶(۰/۰۲)	۲/۲۶(۰/۰۲)
FTMPSO	۰/۳۹(۰/۰۲)	۰/۹۱(۰/۰۸)	۰/۲۱(۰/۰۶)	۱/۶۶(۰/۰۵)	۱/۸۷(۰/۰۵)	۲/۰۹(۰/۰۷)	۲/۲۲(۰/۰۶)
SFA	۰/۸۵(۰/۰۶)	۱/۴۹(۰/۰۷)	۱/۹۵(۰/۰۴)	۲/۲۸(۰/۰۵)	۲/۴۶(۰/۰۵)	۲/۵۷(۰/۰۵)	۲/۷۴(۰/۰۴)
Cellular PSO	۴/۱۵(۰/۲۵)	۲/۸۵(۰/۲۴)	۲/۸۲(۰/۱۴)	۳/۴۱(۰/۱۴)	۳/۶۲(۰/۱۲)	۳/۸۶(۰/۱۰)	۴/۱۰(۰/۱۱)
kmeansPSO	۰/۸۱(۰/۰۹)	۱/۴۲(۰/۰۸)	۱/۸۷(۰/۰۶)	۲/۱۹(۰/۰۷)	۲/۳۹(۰/۰۶)	۲/۵۱(۰/۰۷)	۲/۵۹(۰/۰۶)
GMMPPO	۰/۳۷(۰/۰۳)	۰/۸۵(۰/۰۹)	۱/۲۴(۰/۰۷)	۱/۶۰(۰/۰۷)	۱/۸۱(۰/۰۸)	۲/۰۵(۰/۰۹)	۲/۲۱(۰/۰۷)

جدول ۵: مقایسه خطای آفلاین (خطای استاندارد) روی مسئله MPB با تعداد قله مختلف، فرکانس تغییر ۵۰۰۰، طول جابه‌جایی ۱

الگوریتم	تعداد قله						
	۱	۵	۱۰	۲۰	۳۰	۵۰	۱۰۰
MQSO	۳/۸۲(۰/۲۵)	۱/۹۰(۰/۰۸)	۱/۹۱(۰/۰۸)	۲/۵۶(۰/۱۰)	۲/۶۸(۰/۱۰)	۳/۶۳(۰/۰۸)	۲/۵۲(۰/۰۶)
FMSO	۳/۴۴(۰/۱۱)	۲/۹۴(۰/۰۷)	۳/۱۱(۰/۰۶)	۳/۳۶(۰/۰۶)	۳/۲۸(۰/۰۵)	۳/۲۲(۰/۰۵)	۳/۰۶(۰/۰۴)
AmQSO	۰/۵۱(۰/۰۴)	۱/۰۱(۰/۰۹)	۱/۵۱(۰/۰۱)	۲/۰۰(۰/۱۵)	۲/۱۹(۰/۱۷)	۲/۴۳(۰/۱۳)	۲/۶۸(۰/۱۲)
mPSO	۰/۹۰(۰/۰۵)	۱/۲۱(۰/۱۲)	۱/۶۱(۰/۱۲)	۲/۰۵(۰/۰۸)	۲/۱۸(۰/۰۶)	۲/۳۴(۰/۰۶)	۲/۳۲(۰/۰۴)
HmPSO	۰/۸۷(۰/۰۵)	۱/۱۸(۰/۰۴)	۱/۴۲(۰/۰۴)	۱/۵۰(۰/۰۶)	۱/۵۵(۰/۰۴)	۱/۶۶(۰/۰۲)	۱/۶۸(۰/۰۳)
APSO	۰/۵۳(۰/۰۱)	۱/۰۵(۰/۰۶)	۱/۳۱(۰/۰۳)	۱/۶۹(۰/۰۵)	۱/۷۸(۰/۰۲)	۱/۹۵(۰/۰۲)	۱/۹۵(۰/۰۱)
FTMPSO	۰/۱۸(۰/۰۱)	۰/۴۷(۰/۰۵)	۰/۶۷(۰/۰۴)	۰/۹۳(۰/۰۴)	۱/۱۴(۰/۰۴)	۱/۳۲(۰/۰۴)	۱/۶۱(۰/۰۳)
SFA	۰/۴۲(۰/۰۳)	۰/۸۹(۰/۰۷)	۱/۰۵(۰/۰۴)	۱/۴۸(۰/۰۵)	۱/۵۶(۰/۰۶)	۱/۸۷(۰/۰۵)	۲/۰۱(۰/۰۴)
CESO	۱/۰۴(۰/۰۰)	-	-	۱/۳۸(۰/۰۲)	۱/۷۲(۰/۰۲)	۱/۴۵(۰/۰۱)	۱/۲۸(۰/۰۲)
mNAFSA	۰/۵۹(۰/۰۶)	۰/۶۶(۰/۰۵)	۰/۹۴(۰/۰۴)	۱/۲۹(۰/۰۵)	۱/۶۰(۰/۰۶)	۱/۸۱(۰/۰۵)	۱/۹۲(۰/۰۵)
CDEPSO	۰/۴۱(۰/۰۰)	۰/۹۷(۰/۰۱)	۱/۲۲(۰/۰۱)	۱/۵۴(۰/۰۱)	۲/۶۲(۰/۰۱)	۲/۲۰(۰/۰۱)	۱/۵۴(۰/۰۱)
cellularDE	۱/۵۳(۰/۰۷)	۱/۵۰(۰/۰۴)	۱/۶۴(۰/۰۳)	۲/۴۶(۰/۰۵)	۲/۶۲(۰/۰۵)	۲/۷۵(۰/۰۵)	۲/۷۳(۰/۰۳)
DynPopDE	-	۱/۰۳(۰/۱۳)	۱/۳۹(۰/۰۷)	-	-	۲/۱۰(۰/۰۶)	۲/۳۴(۰/۰۵)
MLDE-S	۱/۱۱(۰/۰۷)	-	۱/۳۵(۰/۰۷)	-	-	-	۱/۶۵(۰/۰۸)
CbDE-wCA	۰/۱۴(۰/۰۳)	۰/۳۰(۰/۰۲)	۰/۸۶(۰/۰۸)	۰/۹۸(۰/۰۵)	۱/۳۴(۰/۰۴)	۱/۳۱(۰/۰۴)	۱/۳۵(۰/۰۳)
DPSABC	۲/۲۵(۰/۰۰)	-	۲/۱۳(۰/۰۰)	۲/۰۷(۰/۰۰)	۱/۸۸(۰/۰۰)	۱/۹۱(۰/۰۰)	۱/۸۹(۰/۰۰)
HdPSO	۰/۱۶(۰/۰۱)	۰/۳۴(۰/۰۱)	۰/۴۹(۰/۰۱)	۰/۶۸(۰/۰۱)	۰/۸۳(۰/۰۱)	۱/۰۸(۰/۰۱)	۱/۱۵(۰/۰۱)
Cellular PSO	۲/۵۴(۰/۱۶)	۱/۷۲(۰/۱۷)	۱/۷۶(۰/۱۳)	۲/۵۹(۰/۱۰)	۲/۵۹(۰/۱۲)	۳/۲۲(۰/۱۱)	۳/۲۹(۰/۱۰)
kmeansPSO	۰/۴۳(۰/۰۳)	۰/۸۷(۰/۰۷)	۱/۰۷(۰/۰۴)	۱/۴۵(۰/۰۵)	۱/۵۴(۰/۰۶)	۱/۸۸(۰/۰۵)	۲/۰۳(۰/۰۴)
GMMPPO	۰/۱۴(۰/۰۱)	۰/۴۱(۰/۰۳)	۰/۵۵(۰/۰۳)	۰/۹۲(۰/۰۴)	۱/۰۷(۰/۰۴)	۱/۱۸(۰/۰۴)	۱/۲۳(۰/۰۳)

جدول ۶: مقایسه خطای آفلاین (خطای استاندارد) روی مسئله MPB با تعداد قله مختلف، فرکانس تغییر ۱۰۰۰۰، طول جابه‌جایی ۱

الگوریتم	تعداد قله						
	۱	۵	۱۰	۲۰	۳۰	۵۰	۱۰۰
MQSO	۱/۹۰(۰/۱۸)	۱/۰۳(۰/۰۶)	۱/۱۰(۰/۰۷)	۱/۸۴(۰/۰۸)	۲/۰۰(۰/۰۹)	۱/۹۹(۰/۰۷)	۱/۷۱(۰/۰۴)
FMSO	۱/۹۰(۰/۰۶)	۱/۷۵(۰/۰۶)	۱/۹۱(۰/۰۴)	۲/۱۶(۰/۰۴)	۲/۱۸(۰/۰۴)	۲/۶۰(۰/۰۸)	۲/۰۰(۰/۰۲)
AmQSO	۰/۱۹(۰/۰۲)	۰/۴۵(۰/۰۴)	۰/۷۶(۰/۰۶)	۱/۲۸(۰/۱۲)	۱/۷۸(۰/۰۹)	۱/۵۵(۰/۰۸)	۲/۵۲(۰/۱۰)
mPSO	۰/۴۴(۰/۰۲)	۰/۷۲(۰/۰۸)	۱/۰۵(۰/۱۰)	۱/۳۷(۰/۰۶)	۱/۵۰(۰/۰۶)	۱/۶۲(۰/۰۴)	۱/۶۴(۰/۰۲)
HmPSO	۰/۴۵(۰/۰۲)	۰/۷۱(۰/۱۲)	۰/۹۴(۰/۰۹)	۱/۰۲(۰/۰۶)	۱/۱۳(۰/۰۴)	۱/۱۰(۰/۰۳)	۱/۰۷(۰/۰۲)
APSO	۰/۲۵(۰/۰۰۶)	۰/۵۷(۰/۰۳)	۰/۸۲(۰/۰۲)	۱/۲۳(۰/۰۲)	۱/۳۹(۰/۰۲)	۱/۴۶(۰/۰۱)	۱/۳۸(۰/۰۱)
FTMPSO	۰/۰۹(۰/۰۰)	۰/۳۱(۰/۰۴)	۰/۴۳(۰/۰۳)	۰/۵۶(۰/۰۱)	۰/۶۹(۰/۰۹)	۰/۸۶(۰/۰۲)	۱/۱۳(۰/۰۴)
SFA	۰/۲۶(۰/۰۳)	۰/۵۳(۰/۰۴)	۰/۷۲(۰/۰۲)	۰/۹۱(۰/۰۳)	۰/۹۹(۰/۰۴)	۱/۱۹(۰/۰۴)	۱/۴۴(۰/۰۴)
DPSABC	۲/۶۷(۰/۰۰)	-	۹/۰۰(۰/۰۱)	۶/۶۰(۰/۰۱)	۷/۷۰(۰/۰۱)	۸/۱۰(۰/۰۱)	۸/۳۴(۰/۰۱)
HdPSO	۰/۰۶(۰/۰۰)	۰/۲۵(۰/۰۱)	۰/۲۹(۰/۰۱)	<u>۰/۳۸(۰/۰۰)</u>	۰/۵۷(۰/۰۰)	۰/۷۳(۰/۰۰)	۰/۸۴(۰/۰۰)
Cellular PSO	۱/۵۳(۰/۱۲)	۰/۹۲(۰/۱۰)	۱/۲۲(۰/۰۹)	۲/۲۰(۰/۱۰)	۲/۶۰(۰/۱۳)	۲/۸۴(۰/۱۲)	۲/۹۳(۰/۰۹)
kmeansPSO	۰/۲۴(۰/۰۳)	۰/۵۱(۰/۰۴)	۰/۷۴(۰/۰۲)	۰/۹۳(۰/۰۳)	۰/۹۷(۰/۰۴)	۱/۲۱(۰/۰۴)	۱/۴۵(۰/۰۴)
GMMPSO	۰/۰۸(۰/۰۱)	<u>۰/۲۲(۰/۰۲)</u>	<u>۰/۲۶(۰/۰۲)</u>	<u>۰/۴۴(۰/۰۱)</u>	<u>۰/۵۵(۰/۰۴)</u>	<u>۰/۶۴(۰/۰۲)</u>	<u>۰/۷۷(۰/۰۳)</u>

بیش‌تر شود، کار الگوریتم برای ردیابی بهینه تغییر یافته نیز مشکل‌تر خواهد بود. همچنین عملکرد تمام الگوریتم‌ها زمانی که شدت تغییرات افزایش یابد، کاهش می‌یابد [۴] که این مطلب در جدول ۷ قابل مشاهده است.

همان‌طور که در جدول ۷ مشاهده می‌شود نتایج به‌دست آمده توسط الگوریتم پیشنهادی، GMMPSO، بسیار بهتر از نتایج حاصل از الگوریتم‌های دیگر روی مسئله محک قله متحرک با شدت تغییرات متفاوت است.

جدول ۷: مقایسه خطای آفلاین (خطای استاندارد) روی مسئله MPB با تعداد شدت تغییرات مختلف، فرکانس تغییر ۱۰۰۰۰، تعداد قله = ۱۰ و ابعاد = ۵

الگوریتم	شدت تغییرات			
	۱	۲	۳	۵
MQSO	۱/۸۵(۰/۰۸)	۲/۴۰(۰/۰۶)	۳/۰۰(۰/۰۶)	۴/۲۴(۰/۱۰)
AmQSO	۱/۵۱(۰/۱۰)	۲/۰۹(۰/۰۸)	۲/۷۲(۰/۰۹)	۳/۷۱(۰/۱۱)
mCPSO	۴/۸۹(۰/۱۱)	۳/۵۷(۰/۰۸)	۲/۸۰(۰/۰۷)	۲/۰۸(۰/۰۷)
SPSO	۲/۵۱(۰/۰۹)	۳/۷۸(۰/۰۹)	۴/۹۶(۰/۱۲)	۶/۷۶(۰/۱۵)
rSPSO	۱/۵۰(۰/۰۸)	۱/۸۷(۰/۰۵)	۲/۴۰(۰/۰۸)	۳/۲۵(۰/۰۹)
PSO-CP	۱/۳۱(۰/۰۶)	۱/۹۸(۰/۰۶)	۲/۲۱(۰/۰۶)	۳/۲۰(۰/۱۳)
SFA	۱/۰۵(۰/۰۴)	۱/۴۴(۰/۰۶)	۲/۰۶(۰/۰۷)	۲/۸۹(۰/۱۳)
MLDE-S	۱/۳۵(۰/۰۷)	۱/۹۱(۰/۰۸)	۲/۳۴(۰/۰۷)	۲/۶۸(۰/۱۱)
CbDE-wCA	۰/۸۶(۰/۰۸)	۰/۸۸(۰/۰۷)	<u>۰/۹۸(۰/۰۹)</u>	<u>۱/۵۴(۰/۰۷)</u>
HdPSO	<u>۰/۴۹(۰/۰۱)</u>	<u>۰/۷۸(۰/۰۷)</u>	۱/۴۸(۰/۰۳)	۱/۹۸(۰/۰۲)
kmeansPSO	۰/۶۸(۰/۰۳)	۱/۲۲(۰/۰۷)	۱/۴۱(۰/۰۹)	۱/۷۰(۰/۰۷)
GMMPSO	<u>۰/۲۶(۰/۰۰۵)</u>	<u>۰/۵۵(۰/۰۳)</u>	<u>۰/۸۳(۰/۰۲)</u>	<u>۱/۲۱(۰/۰۲)</u>

تکرار الگوریتم پیچیدگی محاسباتی $O(knl)$ خواهد شد. در الگوریتم GMM مرحله RM از نظر تئوری زمان اجرای بینهایت دارد اما با برخی

پیچیدگی محاسباتی هر مرحله از الگوریتم kmeans معادل $O(kn)$ می‌باشد که k تعداد خوشه و n تعداد نمونه‌ها است. بنابراین در یک

- of Structures, Springer International Publishing, pp. 11-43, 2017.
- [7] B. Xue, M. Zhang and W. N. Browne, "Multi-objective particle swarm optimisation (PSO) for feature selection," *In: Proceedings of GECCO*, pp. 81-88, New York, NY, USA, 2012.
- [8] E. Vellasques, R. Sabourin and E. Granger, "Gaussian mixture modeling for dynamic particle swarm optimization of recurrent problems," *In: Proceedings of GECCO*, pp. 73-80, New York, NY, 2012.
- [۹] سعید اباضی، امید مرادی، «بهبود میرایی نوسانات سیستم قدرت با به‌کارگیری UPFC و تنظیم پارامترهای کنترل‌کننده بر اساس یک الگوریتم جدید PSO»، *مجله مهندسی برق دانشگاه تبریز*، دوره ۴۶، شماره ۱، صفحه ۱-۱۱، ۲۰۱۶.
- [10] X. Hu and R. C. Eberhart, "Adaptive particle swarm optimization: detection and response to dynamic systems," *In: Proceedings of Congress on Evolutionary Computation*, vol. 2, pp. 1666-1670, 2002.
- [۱۱] الهام شکرانی‌پور، امیرمسعود افتخاری‌مقدم، «ACPSO: یک الگوریتم جدید بهینه‌سازی گروه ذرات تعاونی با قابلیت به روزرسانی تطبیقی پارامترها»، *مجله مهندسی برق دانشگاه تبریز*، دوره ۴۰، شماره ۲، صفحه ۲۱-۳۶، ۲۰۱۰.
- [12] Y. Shengxiang, "Genetic algorithms with memory-and elitism-based immigrants in dynamics," *Evolutionary Computation*, vol. 16, no. 3, pp. 385-416, 2008.
- [13] T. M. Blackwell and P. J. Bentley, "Don't push me! Collision-avoiding swarms," *In: Proceedings of Congress on Evolutionary Computation*, vol. 2, pp. 1691-1696, 2002.
- [14] T. M. Blackwell and P. J. Bentley, "Dynamic search with charged swarms," *In: Proceedings of the fourth Annual Conference on Genetic and Evolutionary Computation*, pp. 19-26. Morgan Kaufmann Publishers Inc., 2002.
- [15] S. Janson and M. Middendorf, "A hierarchical particle swarm optimizer for noisy and dynamic environments," *Genetic Programming and Evolvable Machines*, vol. 7, no. 4, pp. 329-354, 2006.
- [16] H. Wang, D. Wang and S. Yang, "Triggered memory-based swarm optimization in dynamic environments," *Applications of Evolutionary Computing*, pp. 637-646, 2007.
- [17] T. Blackwell and J. Branke, "Multi- swarms, exclusion, and anti-convergence in dynamic environments," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4 pp. 459-472, 2006.
- [18] T. Blackwell, J. Branke and X. Li, "Particle swarms for dynamic optimization problems," *Book Chapter of Swarm Intelligence. Springer Berlin Heidelberg*, pp. 193-217, 2008.
- [19] M. Kamosi, A. B. Hashemi and M. R. Meybodi, "A new particle swarm optimization algorithm for dynamic environments," *In: Proceedings of International Conference on Swarm, Evolutionary and Memetic Computing*, pp. 129-138, 2010.
- [20] D. Yazdani, B. Nasiri, A. Sepas-Moghaddam and M. R. Meybodi, "A novel multi-swarm algorithm for optimization in dynamic environments based on particle swarm optimization," *Applied Soft Computing*, vol. 13, no. 4, pp. 2144-2158, 2013.
- [21] A. Hashemi and M. R. Meybodi, "Cellular PSO: A PSO for dynamic environments," *In: Proceedings of International Symposium on Intelligence Computation and Applications*, pp. 422-433, 2009.
- آستانه گذاری ها زمان آنرا محدود به $O(k^2n)$ یا $O(kmn)$ در صورتی که $k < m$ می‌کنند (m تعداد مخلوط‌های گوسی می‌باشد). این به دلیل به‌روزرسانی مخلوط‌های f_i بوده که هزینه محاسباتی $O(n_i)$ به‌ازای هر کامپوننت است اگر از EM برای این به‌روزرسانی‌ها استفاده شود. بنابراین، با جمع کردن تمام مخلوط‌ها، کل هزینه معادل $O(\sum_{i=1}^k n_i) \in O(nk^2)$ خواهد شد [۴۹]. پیچیدگی زمانی PSO در حدود $O(dnN)$ می‌باشد که N تعداد ذرات و d ابعاد ویژگی (ابعاد فضای جستجو) است [۵۰]. بنابراین در روش kmeansPSO و GMMPSO به ترتیب پیچیدگی زمانی $O(dnN + knl)$ و $O(dnN)$ خواهد بود.
- ### ۶- نتیجه‌گیری
- در این مقاله ضمن مطالعه انواع روش‌های بهبودیافته بهینه‌سازی ازدحام ذرات در محیط پویا روش جدید بهینه‌سازی ازدحام ذرات مبتنی بر الگوریتم مخلوط گوسین پیشنهاد شد. روش پیشنهادی با به‌کارگیری مخلوط‌های گوسین و نیز احتمال تعلق ذره به هر خوشه باعث همگرایی سریع‌تر به قله در فضای مسئله را به همراه داشت. تغییر ذره برآیندی از بهینه هر خوشه متناسب با درجه تعلق ذره به آن خوشه می‌باشد. این امر ضمن به‌کارگیری بهترین مکان ذره، بهینه تمام خوشه‌ها را در نظر می‌گیرد که تنوع در جمعیت که یکی از چالش‌های اصلی در محیط‌های پویا است را به نوعی تضمین می‌کند. عملکرد روش پیشنهادی روی مسئله قله‌های متحرک در کنار دیگر روش‌های شناخته‌شده بهبودیافته بهینه‌سازی ازدحام ذرات مورد ارزیابی قرار گرفت. روش پیشنهادی دارای خطای آفلاین کمتری نسبت به دیگر روش‌ها می‌باشد. همچنین سرعت همگرایی به قله در روش پیشنهادی بیشتر از روش‌های KmeansPSO می‌باشد.
- ### مراجع
- [1] J. K. Kazemi, A. Rezvanian and M. R. A. Meybodi, "CDEPSO: a bi-population hybrid approach for dynamic optimization problems," *Applied Intelligence*, vol. 40, no. 4, pp. 682-694, 2014.
- [2] I. Zahid, W. Shahzad and M. Faiza, "A diverse clustering particle swarm optimizer for dynamic environment: To locate and track multiple optima," *In: Proceedings of 10th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1755-1760, 2015.
- [3] A. Sharifi, K. Kazemi, M. J. Mahdavian and M. R. Meybodi, "A novel hybrid adaptive collaborative approach based on particle swarm optimization and local search for dynamic optimization problems," *Applied Soft Computing*, vol. 32, pp. 432-448, 2015.
- [4] S. Sadeghi, H. Parvin and F. Rad, "Particle swarm optimization algorithm for dynamic environments," *In: Proceedings of Mexican International Conference on Artificial Intelligence*, pp. 260-269, 2015.
- [5] B. Frohlich, M. Enzweiler and U. Franke, "Will this car change the lane?-Turn signal recognition in the frequency domain," *In IEEE Intelligent Vehicles Symposium Proceedings*, pp. 37-42, 2014.
- [6] A. Kaveh, *Particle swarm optimization*, Book chapter of *Advances in Metaheuristic Algorithms for Optimal Design*

- dynamic environments,” In: *Proceedings of Second World Congress on Nature and Biologically Inspired Computing (NaBIC)*, pp. 363-369, 2010.
- [38] I. Rezazadeh, M. R. Meybodi and A. Naeibi, “Adaptive particle swarm optimization algorithm for dynamic environments,” *Advances in Swarm Intelligence*, pp. 120-129, 2011.
- [39] B. Nasiri and M. R. Meybodi, “Speciation based firefly algorithm for optimization in dynamic environments,” *International Journal of Artificial Intelligence*, vol. 8, no. S12, pp. 118-132, 2012.
- [40] R. I. Lung and D. Dumitrescu, “A collaborative model for tracking optimum dynamic environments,” In: *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 564-567, 2007.
- [41] V. Noroozi, A. B. Hashemi and M. R. Meybodi, “CellularDE: a cellular based differential evolution for dynamic optimization problems,” In: *Proceedings of International Conference on Adaptive and Natural Computing Algorithms*, Springer, Berlin, Heidelberg, pp. 340-349, 2011.
- [42] M. C. Du Plessis and A. P. Engelbrecht, “Differential evolution for dynamic environments with unknown numbers of optima,” *Journal of Global Optimization*, pp. 1-27, 2013.
- [43] S. Kundu, D. Basu and S. S. Chaudhuri, “Multipopulation-based differential evolution with speciation-based response to dynamic environments,” In: *Proceedings of International Conference on Swarm, Evolutionary, and Memetic Computing*, pp. 222-235, 2013.
- [44] R. Mukherjee, G. R. Patra, R. Kundu and S. Das, “Cluster-based differential evolution with crowding archive for niching in dynamic environments,” *Information Sciences*, vol. 267, pp. 58-82, 2014.
- [45] N. Baktash and M. R. Meybodi, “A new hybrid model of PSO and ABC algorithms for optimization in dynamic environment,” *International Journal of Computer Theory and Engineering*, vol. 4, no. 3, p. 362, 2012.
- [46] D. Yazdani, B. Nasiri, R. Azizi, A. Sepas-Moghaddam and M. R. Meybodi, “Optimization in dynamic environments utilizing a novel method based on particle swarm optimization,” *International Journal of Artificial Intelligence*, vol. 11, no. A13, pp. 170-192, 2013.
- [47] D. Yazdani, B. Nasiri, A. Sepas-Moghaddam, M. R. Meybodi and M. M. Akbarzadeh-Totonchi, “mNAFSA: a novel approach for optimization in dynamic environments with global changes,” *Swarm and Evolutionary Computation*, vol. 18, pp. 38-53, 2014.
- [۴۸] طیبہ طاهری، بابک نصیری، محمدرضا میبیدی، «یک الگوریتم ترکیبی برای بهینه‌سازی در محیط پویا با استفاده از الگوریتم جستجوی هارمونی و الگوریتم گروه ذرات»، پنجمین کنفرانس ملی مهندسی برق و الکترونیک ایران، دانشگاه آزاد اسلامی واحد گناباد، ۱۳۹۲.
- [49] J. J. Verbeek, N. Vlassis and B. Kröse, “Efficient greedy learning of Gaussian mixture models,” *Neural computation*, vol. 15, no. 2, pp. 469-485, 2003.
- [50] N. Iqbal, A. Zerguine and N. Al-Dhahir, “Decision feedback equalization using particle swarm optimization,” *Signal Processing*, vol. 108, pp. 1-12, 2015.
- [22] C. Li and S. Yang, “Fast multi-swarm optimization for dynamic optimization problems,” In: *Proceedings of Fourth International Conference on Natural Computation*, vol. 7, pp. 624-628, 2008.
- [23] L. Liu, S. Yang and D. Wang, “Particle swarm optimization with composite particles in dynamic environments,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, pp. 1634-1648, 2010.
- [24] U. Halder, S. Das and D. Maity, “A cluster-based differential evolution algorithm with external archive for optimization in dynamic environments,” *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 881-897, 2013.
- [25] T. T. Nguyen, S. Yang and J. Branke, “Evolutionary dynamic optimization. A survey of the state of the art,” *Swarm and Evolutionary Computation*, vol. 6, pp. 1-24, 2012.
- [۲۶] سجاد هواسی، محمدرضا میبیدی، سمانه رحیمی، «یک الگوریتم ترکیبی جدید مبتنی بر بهینه‌سازی دسته‌جمعی و الگوریتم فرهنگی برای محیط‌های پویا»، دومین کنفرانس ملی مهندسی نرم‌افزار لاهیجان، دانشگاه آزاد اسلامی واحد لاهیجان، ۱۳۹۱.
- [27] J. Kennedy, “Stereotyping: Improving particle swarm performance with cluster analysis,” In: *Proceeding of Congress Evolutionary Computation*, pp. 1507-1512, 2000.
- [28] R. Brits, A. P. Engelbrecht and F. Van den Bergh, “Solving systems of unconstrained equations using particle swarm optimization,” In: *Proceeding of IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 102-107, 2002.
- [29] A. W. Moore, *Clustering with gaussian mixtures*, School of Computer Science, Carnegie Mellon University, 2001.
- [30] D. Reynolds, “Gaussian mixture models,” *Encyclopedia of biometrics*, pp. 827-832, 2015.
- [31] N. Shental, A. Bar-Hillel, T. Hertz and D. Weinshall, “Computing gaussian mixture models with EM using equivalence constraints,” In: *Proceedings of Advances in neural information processing systems*, pp. 465-472, 2004.
- [32] A. P. Dempster, N. M. Laird and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 39, no. 1, pp. 1-38, 1977.
- [33] C. Li, S. Yang and M. Yang, “An adaptive multi-swarm optimizer for dynamic optimization problems,” *Evolutionary computation*, vol. 22, no. 4, pp. 559-594, 2014.
- [34] C. Li, T. T. Nguyen, M. Yang, S. Yang and S. Zeng, “Multi-population methods in unconstrained continuous dynamic environments: the challenges,” *Information Sciences*, vol. 296, pp. 95-118, 2015.
- [35] J. Branke, *The moving peaks benchmark*, 1999, URL: www.aifb.uni-karlsruhe.de/~jbr/MovPeaks/movpeaks.
- [36] B. Nasiri, M. R. Meybodi and M. M. Ebadzadeh, “History - driven particle swarm optimization in dynamic and uncertain environments,” *Neurocomputing*, vol. 172, pp. 356-370, 2016.
- [37] M. Kamosi, A. B. Hashemi and M. R. Meybodi, “A hibernating multi-swarm optimization algorithm for

زیر نویس‌ها

- ¹⁷ Expectation-Maximization(EM)
- ¹⁸ Maximum Likelihood (ML)
- ¹⁹ Multi-Swarm PSO (MPSO)
- ²⁰ Hibernating Multi-Swarm Optimization (HMSO)
- ²¹ Adaptive Particle Swarm Optimization (APSO)
- ²² Finder-tracker multi-swarm PSO (FTMPSO)
- ²³ Speciation based Firefly Algorithm (SFA)
- ²⁴ Collaborative Evolutionary-Swarm Optimization(CESO)
- ²⁵ Crowding-based Differential Evolution PSO (CDEPSO)
- ²⁶ Dynamic Population Differential Evolution (DynPopDE)
- ²⁷ Multi-Population based Locally Mutated Differential Evolution with Speciation Archive
- ²⁸ Cluster-based differential evolution with Crowding Archive
- ²⁹ Particle Swarm Optimization And Artificial Bee Colony(DPSABC)
- ³⁰ PSO with adaptive quantum (PSO-AQ)
- ³¹ History-Driven Particle Swarm Optimization (HdPSO)
- ¹ Particle Swarm Optimization Algorithm (PSO)
- ² Outdated Memory
- ³ Re-evaluating the Memory
- ⁴ Forgetting the Memory
- ⁵ Gaussian Mixture Model (GMM)
- ⁶ Expectation Maximization (EM)
- ⁷ Moving Peak Benchmark (MPB)
- ⁸ Randomization PSO (RPSO)
- ⁹ Adaptive Cooperative Particle Swarm Optimizer (ACPSO)
- ¹⁰ Random Immigrants Genetic Algorithm (RIGA)
- ¹¹ Multi-Swarm Cooperative PSO (mCPSO)
- ¹² Hierarchy PSO
- ¹³ Partitioned Hierarchical PSO
- ¹⁴ Multi-quantum swarm optimization
- ¹⁵ Adaptive Multi-Quantum Swarm Optimization (AMQSO)
- ¹⁶ Fast multi-swarm optimization