

بهبود کارایی الگوریتم بهینه سازی دسته جمعی ذرات برای بهینه سازی در محیط پویا با بهره گیری از الگوریتم بهینه سازی حدی

مرضیه نخبه الفقهایی^۱، بابک نصیری^۲ و محمد رضا میبیدی^۳

^۱ دانشکده مهندسی برق، رایانه و فناوری اطلاعات، دانشگاه آزاد اسلامی، واحد قزوین، قزوین، ایران

^۲ دانشکده مهندسی برق، رایانه و فناوری اطلاعات، دانشگاه آزاد اسلامی، واحد قزوین، قزوین، ایران

^۳ دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیر کبیر (پلی تکنیک تهران)، تهران، ایران

چکیده - بسیاری از مسائل دنیای واقعی به صورت یک مسئله بهینه سازی با ماهیتی پویا هستند. در این گونه مسائل، بهینه در طول زمان تغییر می کند، بنابراین علاوه بر پیدا کردن بهینه سراسری می بایست آن را در طول زمان دنبال کرد. در این مقاله، الگوریتم ترکیبی جدیدی مبتنی بر الگوریتم دسته جمعی ذرات و الگوریتم بهینه سازی حدی برای حل مسائل بهینه سازی در محیط پویا پیشنهاد شده است. جستجوی بهینه سراسری با استفاده از الگوریتم دسته جمعی ذرات صورت می گیرد. ایده اصلی مطرح شده در این مقاله استفاده از الگوریتم بهینه سازی حدی برای جستجوهای محلی و نزدیک بهینه می باشد و می تواند بهترین جواب را از میان جواب های محلی بدست آورد. لذا با ترکیب این دو الگوریتم می توان تعاملی بین دو مفهوم اساسی مطرح در الگوریتم های تکاملی، اکتشاف و بهره برداری، ایجاد نمود و جواب های بهتری بدست آورد. نتایج حاصل از این الگوریتم ترکیبی پیشنهادی، بر روی معیار قله های متحرک که از معروف ترین معیارهای محیط های پویا است ارزیابی شده و با نتایج حاصل از چندین الگوریتم معتبر مورد مقایسه قرار گرفته است. نتایج به دست آمده بیانگر کارایی بالای الگوریتم ترکیبی پیشنهادی در مقایسه با سایر الگوریتم ها می باشد.

کلید واژه- بهینه سازی، محیط پویا، الگوریتم بهینه سازی دسته جمعی ذرات، الگوریتم بهینه سازی حدی، معیار قله های متحرک.

محیط و نیز میزان آن، بهبود جستجوی محلی این الگوریتم به کمک الگوریتم بهینه سازی حدی می باشد.

۱. مقدمه

الگوریتم بهینه سازی دسته جمعی ذرات یا PSO، در سال ۱۹۹۵ توسط Kennedy و Eberhart معرفی شد [۱]. این یک الگوریتم بهینه سازی است که از روی زندگی جمعی و گروهی پرندگان الهام می گیرد تا به راه حل بهینه برسد. تاکنون نسخه های متعددی از آن برای بهینه سازی در محیط های ایستا و پویا معرفی شده است. یک مشکل اساسی این الگوریتم از دست دادن تنوع پس از مدتی است که باعث همگرایی زودرس می شود و مشکل دیگر به خصوص برای محیط های پویا، بلا استفاده شدن حافظه پس از تغییر محیط می باشد. راه حل های متنوعی برای حل این دو مشکل پیشنهاد شده است که در بخش های بعدی به آن ها اشاره می شود.

الگوریتم بهینه سازی حدی (EO) که یک الگوریتم جستجوی محلی است برای ممانعت از راه حل های بد و یافتن راه حل های خوب می باشد. به عبارتی دیگر، این الگوریتم بر خلاف

تعداد زیادی از مسائل بهینه سازی در جهان واقعی پویا می باشند. از آن جا که بهینه این مسائل با تغییرات محیطی در طول زمان جابجا می شوند، بنابراین در حل آنها یافتن بهینه سراسری کافی نیست بلکه پیگیری بهینه در طول تغییرات اهمیت دارد. در سال های اخیر استفاده از الگوریتم های هوش جمعی برای حل این مسائل توجه زیادی را به خود جلب کرده است. یکی از این الگوریتم های هوش جمعی الگوریتم بهینه سازی دسته جمعی ذرات می باشد. ولی مشکل این الگوریتم ها این است که پیش از موعد به یک بهینه همگرا می شوند و همچنین در صورت تغییر در محیط همگرا شدن به نقطه بهینه جدید بسیار زمان گیر است. این همگرایی زودرس باعث از دست دادن تنوع و گیر افتادن در بهینه محلی می شود. یک روش مناسب برای حل این مشکل و بی اثر کردن وقوع تغییر در

$$v_{i,j}^{t+1} = wv_{i,j}^t + c_1r_1(p_{i,j}^t - x_{i,j}^t) + c_2r_2(g_j^t - x_{i,j}^t) \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

به طوری که w وزن اینرسی، c_1 و c_2 ضرایب شتاب و r_1 و r_2 اعداد تصادفی در بازه (۰ و ۱) می‌باشند. همچنین p و g به ترتیب $pbest$ و $gbest$ می‌باشند.

۲-۱- الگوریتم بهینه‌سازی دسته‌جمعی ذرات در محیط‌های پویا

تحقیقات زیادی در رابطه با بهینه‌سازی PSO در محیط‌های پویا انجام شده است.

Blackwell و Branke در مرجع [۴] mQSO را مطرح کردند که در آن 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 مقداردهی اولیه مجدد می‌کند. عملگر ضد همگرایی هنگامی که تمام گروه‌ها همگرا می‌شوند، گروه بدتر را مقداردهی اولیه مجدد می‌کند. سپس Blackwell و همکاران در مرجع [۵] Adaptive mQSO را مطرح کردند که در آن تعداد گروه‌ها از ابتدا معین نیست و با پیدا کردن قله‌های جدید افزایش می‌یابد، به این‌صورت 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.

Parrott و Li در منبع [۷] PSO مینی بر جزء (SPSO) را برای حل مسائل بهینه‌سازی پویا مطرح کردند. مرکز یک جزء که دانه جزء نامیده می‌شود ذره‌ای است که همواره بهترین برآزش را در آن جزء دارد. همه ذراتی که در شعاع از پیش تعریف شده دانه جزء واقع شوند در همان جزء قرار می‌گیرند. این الگوریتم گروه‌ها را بجای همگرایی به یک بهینه سراسری به سمت چند بهینه محلی همگرا می‌کند، از اینرو چند زیر جمعیت بطور موازی توسعه می‌یابند.

Li و Yang یک روش چند گروهی سریع را معرفی کرده‌اند که پراکندگی ذرات را در حین اجرا حفظ می‌کند [۸]. در این روش گروهی از ذرات والد میزان پراکندگی ذرات را حفظ می‌نمایند و نواحی نوید بخش فضای جستجو را با استفاده از الگوریتم برنامه‌ریزی تکاملی سریع شناسایی می‌کنند و دسته‌ای

سایر الگوریتم‌های تکاملی و هوش جمعی بدنبال همگرایی نبوده و همواره سعی در بهبود راه حل یافت شده دارد. این ویژگی الگوریتم بهینه‌سازی حدی را یک انتخاب مناسب برای محیط‌های پویا می‌سازد و سبب می‌شود الگوریتم توانایی تطبیق پذیری بسیار بالایی را با محیط پیدا کند که این امر بدون در نظر گرفتن شدت و فرکانس تغییرات می‌باشد [۲].

در این مقاله با ترکیب این دو روش به گونه‌ای که بهترین فرد بدست آمده با الگوریتم بهینه‌سازی دسته‌جمعی ذرات، توسط الگوریتم بهینه‌سازی حدی دوباره بهبود می‌یابد نتایج بهتری حاصل می‌شود.

الگوریتم پیشنهادی بر روی سناریوهای مختلف معیار قله‌های متحرک (MPB) [۳]، که از معروف‌ترین معیارهای محیط‌های پویا است به کار رفته و کارایی آن با الگوریتم‌های mQSO [۴] و Adaptive mQSO [۵] و mPSO [۶] مقایسه شده است. نتایج آزمایشات نشان می‌دهد که الگوریتم پیشنهادی از کارایی قابل قبولی برخوردار است.

این مقاله شامل شش بخش می‌باشد: در بخش دوم مروری بر الگوریتم PSO و کارهای انجام شده قبلی برای انطباق با محیط‌های پویا مطرح می‌شود. در بخش سوم به الگوریتم بهینه‌سازی حدی پرداخته می‌شود. در بخش چهارم الگوریتم پیشنهادی طرح می‌گردد. نتایج آزمایشات و همچنین نتیجه‌گیری این مقاله به ترتیب در بخش‌های پنجم و ششم ارائه شده است.

۲-۲- الگوریتم بهینه‌سازی دسته‌جمعی ذرات

این الگوریتم با یک گروه از جواب‌های تصادفی شروع به کار می‌کند. سپس برای یافتن جواب بهینه در فضای مساله با به روز کردن موقعیت و سرعت هر ذره به جستجو می‌پردازد. هر ذره به صورت چند بعدی با دو مقدار $x_{i,j}$ و $v_{i,j}$ که به ترتیب مکان و سرعت مربوط به بعد j از i امین ذره هستند تعریف می‌شود. در هر مرحله از حرکت جمعیت، هر ذره با توجه به دو مقدار بهترین به روز می‌شود. اولین مقدار بهترین جواب از لحاظ شایستگی است که تاکنون برای هر ذره به طور جداگانه به دست آمده است. این مقدار بهترین تجربه فردی است که $pbest$ نامیده می‌شود. مقدار بهترین دیگر که توسط PSO به دست می‌آید، بهترین مقداری است که تاکنون توسط تمام ذره‌ها در میان جمعیت به دست آمده است. این مقدار بهترین تجربه گروهی است که $gbest$ نامیده می‌شود. پس از یافتن دو مقدار $pbest$ و $gbest$ هر ذره سرعت و مکان جدید خود را با دو رابطه زیر به‌روز می‌کند:

از گروه‌های ذرات فرزند ناحیه‌ای محلی را با استفاده از الگوریتم بهینه‌سازی گروه ذرات سریع با هدف یافتن بهینه‌ی مساله جستجو می‌کنند. علاوه بر این، آنها در [۹] یک روش خوشه‌بندی را معرفی کرده‌اند که گروه ذرات را به زیر گروه‌های مختلفی افزایش می‌کند و هر یک از آنها به جستجوی یک ناحیه محلی در فضای جستجو می‌پردازند.

کاموسی و همکاران در [۶] الگوریتم mPSO را ارائه دادند که از یک گروه والد برای اکتشاف فضای جستجو و چند گروه فرزند برای بهره‌برداری از نواحی امید بخش یافته شده توسط گروه والد استفاده می‌کند.

یزدانی و همکاران در منبع [۱۰] یک الگوریتم جدید مبتنی بر روش بهینه‌سازی ذرات مطرح کردند که در آن از یک روش چند گروهی بهبود یافته استفاده شده است که در آن گروهها به دودسته یابنده و دنبال کننده تقسیم می‌شوند. علاوه بر این یک روش نوین مبتنی بر تغییر در 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-بیدار نوین که برای تمرکز بر روی فشار کلی الگوریتم بر روی قله بهینه جهانی می‌باشد نیز استفاده شده است.

۳- الگوریتم بهینه‌سازی حدی

الگوریتم بهینه‌سازی حدی یک الگوریتم تکاملی است که برای حل مسائل بهینه‌سازی مورد استفاده قرار می‌گیرد. بر خلاف بسیاری از الگوریتم‌های تکاملی که در هر لحظه از چندین جواب تشکیل شده‌اند این الگوریتم تنها از یک جواب تشکیل شده است که در هر مرحله آن را بهبود می‌دهد. این الگوریتم برای تولید جوابهای بهتر، در هر مرحله یکی از اجزای آخرین جواب ایجاد شده را انتخاب و مقدار آن را با مقداری جدید جایگزین می‌کند. انتخاب یک جزء از بین اجزای جواب با توجه به شایستگی محلی آنها انجام می‌گیرد [۲]. شبه کد این الگوریتم بصورت زیر می‌باشد:

۱. یک راه‌حل به عنوان راه‌حل اولیه تولید شده و به عنوان راه‌حل جاری و همچنین به عنوان بهترین راه‌حل در نظر گرفته می‌شود.

۲. برای این راه حل:

الف) برازش را برای هر متغیر راه‌حل محاسبه می‌کنیم.

ب) همه برازش‌ها را مرتب کرده و متغیری که دارای

کمترین برازش است را انتخاب می‌کنیم.

پ) مقدار متغیر انتخابی با مقدار دیگری در همسایگی آن جایگزین می‌شود.

ت) راه‌حل ایجاد شده به عنوان راه‌حل جاری در نظر گرفته می‌شود.

ث) در صورتی که راه‌حل جاری بهتر از بهترین جواب باشد به عنوان بهترین راه‌حل نیز انتخاب می‌شود.

۳. مرحله ۲ را تا رسیدن به راه‌حل مطلوب تکرار می‌کنیم.

۴. بهترین راه‌حل برگردانده می‌شود.

طبق نتایج موجود در [۱۱] الگوریتم بهینه‌سازی حدی به تنهایی برای مساله قله‌های متحرک خوب کار نمی‌کند. بنابراین در [۱۲] یک الگوریتم بهینه‌سازی حدی چند فردی چند مرحله‌ای ارائه شد که در حل مساله قله‌های متحرک خوب کار کرد و عمده دلیل موفقیت آن مربوط به بخش جستجوی محلی در آن است. این الگوریتم متشکل از چندین مرحله EO، مرحله تپه نوردی، مرحله چک کردن جوابهای تکراری، مرحله بهینه‌سازی مجدد و مرحله میزان‌سازی دقیق می‌باشد. سپس در [۱۳] بخش جستجوی محلی، به الگوی جستجوی HJ که یک جستجوی محلی خیلی ساده است و نیز جستجوی محلی بهبود یافته ارتقا داده شد.

۴- الگوریتم پیشنهادی

با توجه به مزایا و معایب گفته شده در مورد الگوریتم بهینه‌سازی ازدحام ذرات و برای رفع معایب آن، الگوریتم مذکور را با الگوریتم بهینه‌سازی حدی ترکیب می‌کنیم. برای اینکار بهترین فرد بدست آمده با الگوریتم بهینه‌سازی دسته جمعی ذرات، توسط الگوریتم بهینه‌سازی حدی دوباره بهبود می‌یابد.

در این الگوریتم پیشنهادی به دلیل چالش‌های بیان شده در الگوریتم PSO استاندارد، از الگوریتم PSO چند گروهی مطرح شده در [۵] تحت نام Adaptive mQSO به جای حالت استاندارد آن استفاده می‌شود. در این الگوریتم در ابتدای کار تنها یک دسته وجود دارد. یک دسته جدید هنگامی ایجاد می‌شود که تمام دسته‌های موجود در فضای مسئله همگرا شده باشند. در این الگوریتم تنها یک دسته می‌تواند در فضای مسئله به صورت آزاد حرکت کند. در صورتی که بیش از یک دسته آزاد باشد تنها بهترین آنها در فضای مسئله باقی می‌ماند و بقیه آنها از بین می‌روند. اجزای اصلی و تنظیمات پارامترهای این الگوریتم نیز مطابق با [۵] می‌باشد.

در این الگوریتم متغیر طول گام اولیه در ابتدای برنامه مقداردی اولیه می‌شود. این مقدار طی اجرای برنامه برای gbest هر گروه و برای هر بعد آن با توجه به الگوی حرکت پیشنهادی تغییر می‌کند. پس از کشف هر تغییر در محیط، طول گام‌های حرکت در الگوریتم پیشنهادی به اندازه شدت تغییر جهش می‌یابند. اینکار در یافتن سریعتر پاسخ با حرکت دادن راه‌حل در مسیر تغییر نقش بسزایی ایفا می‌کند. همچنین طی اجرای برنامه، طول گام حرکت برای gbest هر گروهی که تصادفی‌سازی بر روی آن انجام شود به مقدار اولیه تغییر می‌یابد.

نمونه برداری فضا برای تولید کاندیدها با روش پیشنهادی به اینصورت است که بر روی هر بعد از راه حل جاری، با ثابت نگه داشتن بقیه ابعاد، یک جهش انجام می‌دهیم. اگر مقدار k امین بعد از راه حل جاری i را با $x_{i,k}$ و مقدار جدید حاصل از جهش آن را با $x'_{i,k}$ نشان دهیم، آنگاه جهش بصورت زیر انجام می‌شود:

$$x'_{i,k} = x_{i,k} \pm \text{steplength}_{i,k} * \text{rand} \quad (4)$$

که در آن $\text{rand} \in (0,1)$ می‌باشد و $\text{steplength}_{i,k}$ طول گام مربوط به k امین بعد از راه حل جاری i می‌باشد. آنگاه مقدار برازش کاندیدها را محاسبه کرده و بهترین کاندید را به‌عنوان راه حل جاری در نظر می‌گیریم.

بعد از این کار وارد مرحله جستجوی محلی یا به عبارتی تغییر هر یک از ابعاد راه حل جاری با استفاده از الگوی حرکت پیشنهادی می‌شویم. به اینصورت که مقادیر ابعاد راه حل جاری به‌ترتیب و به تعداد دفعات معینی با طول گام مربوط به آن بعد حرکت داده می‌شوند. پس از هر حرکت در صورتیکه برازش بهتری حاصل شد مقدار جدید را به عنوان راه حل جاری در نظر می‌گیریم و حرکت بعدی را با همان طول گام انجام می‌دهیم، در غیر اینصورت مقدار قبلی آن بعد را با طول گام جدید که منفی نصف طول گام قبلی است جمع می‌کنیم.

۵- ارزیابی

برای ارزیابی کارایی الگوریتم پیشنهادی، این الگوریتم همراه با سه الگوریتم شناخته شده به نام‌های mQSO [۴] و Adaptive mQSO [۵] و mPSO [۶] مورد مقایسه قرار گرفته است. نتایج با توجه به پارامترهای جدول ۱ که با نام سناریو ۲ در مساله MPB معروف است، آورده شده است [۳]. تنها پارامترهای متفاوت، تعداد قله‌ها و فرکانس تغییر می‌باشد که برای ارزیابی بهتر بین الگوریتم‌ها از ۱ قله تا ۲۰۰ قله در فرکانس‌های تغییر ۵۰۰،

در هر دور از اجرای الگوریتم Adaptive mQSO، چندتا از بهترین گروه‌ها انتخاب شده و gbest مربوط به این گروه‌ها به کمک الگوریتم بهینه‌سازی حدی دوباره بهبود می‌یابد. چون تعداد گروه‌ها در هر دور از اجرای الگوریتم Adaptive mQSO متغیر می‌باشد بنابراین تابع انتخاب gbest ها برای اعمال بهینه‌سازی حدی بر روی آنها در الگوریتم پیشنهادی بصورت انتخاب n تا gbest بهتر است که n بصورت زیر محاسبه می‌شود:

$$n = \text{round}((\text{swarm number} + 1)/4) \quad (3)$$

شبه کد این الگوریتم در شکل ۱ نشان داده شده است.

```
//Initialization
Begin with a single free swarm,  $M = 1$ 
FOR EACH particle  $ni$ 
    Randomly initialize  $\mathbf{v}_{ni}, \mathbf{x}_{ni}$ 
     $\text{pbest}_{ni} = \mathbf{x}_{ni}$ 
    Evaluate  $f(\text{pbest}_{ni})$ 
    FOR EACH swarm  $n$ 
         $\text{gbest}_n := \text{argmax}\{f(\text{pbest}_{ni})\}$ 
Initialize Test_point
REPEAT
    Adapt number of swarms() [5]
    Test for Change() [5]
    Select some of best swarms
    FOR EACH swarm  $n$ 
        Update particles based on particles type() [5]
    FOR gbest of each selected swarm
        Extremal Optimization()
    FOR EACH swarm  $n$ 
        Evaluate  $f(\mathbf{x}_{ni})$ 
        IF  $f(\mathbf{x}_{ni}) > f(\text{pbest}_{ni})$  THEN
             $\text{pbest}_{ni} = \mathbf{x}_{ni}$ 
        IF  $f(\mathbf{x}_{ni}) > f(\text{gbest}_n)$  THEN
             $\text{gbest}_n = \mathbf{x}_{ni}$ 
    Exclusion() [5]
UNTIL number of function evaluations performed  $> \text{max}$ 
```

شکل ۱. شبه کد الگوریتم پیشنهادی

شبه کد تابع بهینه‌سازی حدی فراخوانی شده در الگوریتم پیشنهادی بصورت زیر می‌باشد:

- ۱- در نظر گرفتن gbest به عنوان راه حل جاری.
- ۲- نمونه برداری فضا در هر بعد از راه حل جاری به روش پیشنهادی و تولید تعدادی کاندید. انتخاب بهترین کاندید از نظر شایستگی به عنوان راه حل جاری.
- ۳- استفاده از الگوی حرکت پیشنهادی به تعداد دفعات معین بر روی هریک از ابعاد راه حل جاری.
- ۴- ذخیره راه حل بهینه شده نهایی حاصل از مرحله قبل در ذره مربوط به آن از جمعیت.

جدول ۳: مقایسه با سایر الگوریتم‌ها در فرکانس ۱۰۰۰

M \	MQSO(5,5q)	Adaptive mQSO	mPSO	الگوریتم پیشنهادی
1	7.17(0.43)	2.33(0.31)	5.78(0.09)	0.39(0.02)
5	5.06(0.27)	2.90(0.32)	5.60(0.09)	1.25(0.09)
10	4.40(0.19)	4.56(0.40)	5.33(0.10)	1.92(0.14)
20	5.17(0.23)	5.36(0.47)	5.15(0.12)	2.91(0.11)
30	5.67(0.21)	5.20(0.38)	4.97(0.13)	3.06(0.12)
50	5.73(0.15)	6.06(0.14)	4.57(0.18)	3.54(0.09)
100	5.13(0.19)	4.77(0.45)	3.93(0.16)	3.53(0.08)
200	4.95(0.13)	5.75(0.26)	4.44(0.24)	3.73(0.10)

۱۰۰۰، ۲۵۰۰ و ۵۰۰۰ در نظر گرفته شده است. آزمایشات ۳۰

بار تکرار شده‌اند و متوسط مقدار خطای برون خطی و خطای استاندارد [۳] الگوریتم پیشنهادی به همراه سه الگوریتم دیگر در جداول ۲ تا ۵ ارائه شده‌اند. در این جداول، خطای استاندارد در کنار خطای برون خطی درون پراتنز نشان داده شده است. برای مقایسه بهتر، نتایج برتر پررنگ‌تر و بصورت مورب نمایش داده شده است.

همانطور که در جداول ۲ تا ۵ مشاهده می‌شود، نتایج الگوریتم پیشنهادی همواره از سه الگوریتم دیگر بهتر است بجز برای فرکانس ۲۵۰۰ و ۲۰۰ قله که الگوریتم mPSO دارای پاسخ بهتری است.

جدول ۱: پارامترهای MPB

پارامتر	مقدار پیش فرض
P تعداد قله ها	۱۰
فرکانس تغییرات	هر ۵۰۰۰ ارزیابی
میزان تغییر ارتفاع	7.0
میزان تغییر پهنا	1.0
شکل قله ها	Cone
S طول تغییر	1.0
λ	۰
D تعداد ابعاد	5
H مقدار مینیمم و ماکزیمم ارتفاع	[30.0, 70.0]
W مقدار مینیمم و ماکزیمم پهنا	[1, 12]
A محدوده فضای جستجو	[0, 100]

جدول ۴: مقایسه با سایر الگوریتم‌ها در فرکانس ۲۵۰۰

M \	MQSO(5,5q)	Adaptive mQSO	mPSO	الگوریتم پیشنهادی
1	3.65(0.21)	0.87(0.11)	3.36(0.05)	0.17(0.01)
5	2.81(0.16)	2.16(0.19)	3.31(0.05)	0.63(0.09)
10	2.50(0.12)	2.49(0.10)	3.26(0.07)	1.07(0.09)
20	3.60(0.14)	2.73(0.11)	3.15(0.08)	1.67(0.12)
30	3.39(0.13)	3.24(0.18)	3.07(0.11)	1.93(0.10)
50	3.56(0.14)	3.68(0.15)	2.66(0.16)	2.17(0.10)
100	3.39(0.12)	3.53(0.14)	2.04(0.12)	2.02(0.09)
200	3.36(0.08)	3.07(0.12)	1.79(0.10)	2.07(0.07)

جدول ۵: مقایسه با سایر الگوریتم‌ها در فرکانس ۵۰۰۰

M \	MQSO(5,5q)	Adaptive mQSO	mPSO	الگوریتم پیشنهادی
1	2.05(0.10)	0.87(0.11)	0.90(0.05)	0.08(0.00)
5	1.81(0.11)	2.16(0.19)	1.21(0.12)	0.29(0.04)
10	1.63(0.09)	2.49(0.10)	1.61(0.12)	0.52(0.04)
20	2.62(0.10)	2.73(0.11)	2.05(0.08)	1.10(0.08)
30	2.60(0.09)	3.24(0.18)	2.18(0.06)	1.50(0.08)
50	2.63(0.13)	3.68(0.15)	2.34(0.06)	1.50(0.07)
100	2.54(0.08)	3.53(0.14)	2.32(0.04)	1.50(0.06)
200	2.42(0.06)	3.07(0.12)	2.34(0.03)	1.47(0.05)

جدول ۲: مقایسه با سایر الگوریتم‌ها در فرکانس ۵۰۰

M \	MQSO(5,5q)	Adaptive mQSO	mPSO	الگوریتم پیشنهادی
1	17.04(1.31)	1.72(0.11)	8.88(0.14)	0.70(0.05)
5	8.90(0.35)	4.16(0.15)	8.91(0.17)	2.40(0.14)
10	7.83(0.35)	5.11(0.20)	8.76(0.18)	3.85(0.20)
20	7.95(0.35)	5.44(0.19)	8.43(0.17)	4.68(0.18)
30	7.63(0.24)	5.96(0.16)	8.01(0.19)	5.40(0.19)
50	7.84(0.35)	6.23(0.18)	7.19(0.23)	5.52(0.19)
100	8.55(0.29)	6.20(0.16)	6.69(0.26)	5.98(0.20)
200	7.83(0.29)	6.40(0.19)	8.71(0.48)	5.71(0.14)

مشاهده می‌شود که با افزایش فرکانس تغییر، کارایی الگوریتم‌ها بهبود می‌یابد چون فرصت بیشتری برای رسیدن به اهداف تا تغییر محیطی بعدی دارند. همچنین در الگوریتم

مراجع

1. J. Kennedy and R. Eberhart, "Particle Swarm Optimization", in IEEE International Conference on Neural Networks, Vol. 4, pp. 1942-1948, Perth, November 1995.
2. S. Boettcher, and A.G. Percus, "Extremal optimization: an evolutionary local-search algorithm." In Computational Modeling and Problem Solving in the Networked World, pp. 61-77. Springer US, 2003.
3. J. Branke. The Moving Peaks Benchmark Website. <http://www.aifb.unikarlsruhe.De/jbr/movpeaks>.
4. T. Blackwell and J. Branke, "Multiswarms, exclusion, and anti-convergence in dynamic environments." Evolutionary Computation, IEEE Transactions on 10.4 (2006): 459-472.
5. T. Blackwell and J. Branke, and X. Li, "Particle swarms for dynamic optimization problems." Swarm Intelligence. Springer Berlin Heidelberg, 2008. 193-217.
6. M. Kamosi and B. Hashemi, and M. Meybodi. "A new particle swarm optimization algorithm for dynamic environments." Swarm, evolutionary, and memetic computing. Springer Berlin Heidelberg, 2010. 129-138.
7. D. Parrott and X. Li, "Locating and tracking multiple dynamic optima by a particle swarm model using speciation." Evolutionary Computation, IEEE Transactions on 10.4 (2006): 440-458.
8. Li, Changhe, and Shengxiang Yang. "Fast multi-swarm optimization for dynamic optimization problems." Natural Computation, 2008. ICNC'08. Fourth International Conference on. Vol. 7. IEEE, 2008.
9. Yang, Shengxiang, and Changhe Li. "A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments." Evolutionary Computation, IEEE Transactions on 14.6 (2010): 959-974.
10. Yazdani, Danial, et al. "A novel multi-swarm algorithm for optimization in dynamic environments based on particle swarm optimization." Applied Soft Computing 13.4 (2013): 2144-2158.
11. Moser, Irene. "Applying extremal optimisation to dynamic optimisation problems." PhD in information technology, Swinburne University of Technology. Faculty of Information and Communication Technologies (2008).
12. Moser, Irene, and Tim Hendtlass. "A simple and efficient multi-component algorithm for solving dynamic function optimisation problems." Evolutionary Computation, 2007. CEC 2007. IEEE Congress on. IEEE, 2007.
13. Moser, Irene, and Raymond Chiong. "Dynamic function optimisation with hybridised extremal dynamics." Memetic Computing 2.2 (2010): 137-148.

پیشنهادی با افزایش تعداد قله‌ها، تعداد گروه‌ها و در نتیجه تعداد کلی ذرات نیز بیشتر می‌شود و این باعث افزایش تعداد ارزیابی‌ها برآزش شده و در نتیجه سرعت همگرایی و نیز کارایی با افزایش تعداد قله‌ها کاهش می‌یابد.

یکی از دلایل برتری این الگوریتم نسبت به سه الگوریتم دیگر، استفاده از تابع بهینه‌سازی حدی می‌باشد. افزودن این تابع به الگوریتم PSO اگرچه باعث افزایش تعداد ارزیابی‌ها برآزش در هر دور می‌شود اما با نزدیک‌تر کردن بهینه سراسری به پاسخ، سرعت همگرایی را قبل از هر تغییر افزایش می‌دهد. همچنین تابع بهینه‌سازی حدی فقط بر روی چندتا از بهترین گروه‌ها انجام می‌شود تا ارزیابی‌ها برآزش کمتری برای قله‌های غیر بهینه هدر رود.

دلیل دیگر بهتر بودن این الگوریتم، انطباق تعداد گروه‌ها با تعداد قله‌ها است که این باعث هدر رفتن تعداد ارزیابی‌ها برآزش کمتری شده و بخصوص برای تعداد قله‌های کمتر که تعداد کلی ذرات نیز در آن کمتر است پاسخ‌های بسیار خوبی مشاهده می‌شود.

در الگوریتم پیشنهادی، برای یافتن تغییرات محیطی، فقط یک ارزیابی برآزش برای Test_point در هر دور از اجرای الگوریتم انجام می‌شود که این تعداد در الگوریتم‌های دیگر خیلی بیشتر و مساوی با تعداد گروه‌های موجود در مساله می‌باشد.

۶- نتیجه‌گیری

برای رفع معایب الگوریتم بهینه‌سازی ازدحام ذرات، الگوریتم مذکور را با الگوریتم بهینه‌سازی حدی ترکیب می‌کنیم. با ترکیب این دو الگوریتم از قدرت مکاشفه کل فضای جستجوی الگوریتم بهینه‌سازی ازدحام ذرات (برای یافتن بهینه سراسری) و نیز قدرت بهره‌برداری الگوریتم بهینه‌سازی حدی در یافتن جوابهای دقیق در قسمتی از فضای جستجو استفاده می‌کنیم. در این الگوریتم پیشنهادی به دلیل چالش‌های بیان شده در الگوریتم PSO استاندارد، از الگوریتم Adaptive mQSO به جای حالت استاندارد آن استفاده شد. نتایج بر روی معیار قله‌های متحرک با چند الگوریتم شناخته شده دیگر مورد مقایسه قرار گرفت. نتایج آزمایشات بیانگر کارایی بهتر الگوریتم پیشنهادی در مقایسه با الگوریتم‌های دیگر می‌باشد. در کار بعدی با بهبود بیشتر الگوریتم PSO و همچنین جستجوی محلی در EO می‌توان به نتایج بهتری رسید.