

Adaptive Increasing/Decreasing PSO for Solving Dynamic Optimization Problems

Seyyed Masoud Ejabati^{1*} and Seyed Hamid Zahiri²

1*- Electrical Engineering Department, University of Birjand, Birjand, Iran.

2- Electrical Engineering Department, University of Birjand, Birjand, Iran.

^{1*}ejabati_masoud@birjand.ac.ir, and ²hzahiri@birjand.ac.ir

Corresponding author address: Seyyed Masoud Ejabati, Faculty of Electrical and Computer Engineering, University of Birjand, Birjand, Iran, Post Code: 97174 – 34765.

Abstract- Science progress has introduced new issues into the world requiring optimization algorithm with fast adaptation with uncertain environment changing with time. In these issues, location and optimized value change over time, so optimization algorithm should be capable of fast adaptation with variable conditions. This study has proposed a new algorithm based on particle optimization algorithm called Adaptive Increasing/Decreasing PSO. This algorithm, adaptively with an increase and decrease in the number of algorithm particles and effective search limit, is capable of searching and finding optimized number changed with time in non-linear and dynamic environments with undetectable changes. Also, a new definition, focused search zone, is provided for signaling hopeful areas in order to accelerate local search process and prevent premature convergence, and success index as an indicator of the behavior of centralized search area in relation to environmental conditions. Results of the proposed algorithm on the moving peaks benchmark were assessed and compared with the results of some other studies. Results show positive effects of adaptive mechanisms such as a decrease and an increase in the particles and search limit on the duration of searching and finding optimization in comparison with other multi-population based optimization algorithms.

Keywords- Adaptive Increasing and Decreasing Particles, Adaptive Search Radius, Dynamic Optimization Problems (DOPs), Local Search, Multi-population, Particle Swarm Optimization.

الگوریتم انطباقی بهینه‌سازی ذرات افزایشی کاهشی برای حل مسائل بهینه‌سازی

پویا

سیدمسعود اجابتی^{۱*}، سیدحمید ظهیری^۲

*۱- دانشکده مهندسی برق و کامپیوتر، دانشگاه بیرجند، بیرجند، ایران.

۲- دانشکده مهندسی برق و کامپیوتر، دانشگاه بیرجند، بیرجند، ایران.

^{۱*}ejabati_masoud@birjand.ac.ir, and ^۲hzahiri@birjand.ac.ir

* نشانی نویسنده مسئول: سیدمسعود اجابتی، بیرجند، دانشگاه بیرجند، دانشکده مهندسی برق و کامپیوتر، کد پستی: ۳۴۷۶۵-۹۷۱۷۴

چکیده- با پیشرفت روزافزون علم، همواره با مسائل جدیدی در دنیای واقعی روبه‌رو می‌شویم که نیاز به الگوریتم بهینه‌سازی با قابلیت انطباق سریع با محیط در حال تغییر با زمان و غیرقطعی را بیشتر نمایان می‌کند. در این گونه مسائل، شرایط همواره به گونه‌ای پیش می‌رود که مکان و مقدار بهینه در طول زمان تغییر می‌یابد؛ از این رو الگوریتم بهینه‌سازی باید توانایی انطباق سریع با شرایط متغیر را داشته باشد. در این مقاله، الگوریتم جدیدی بر مبنای الگوریتم بهینه‌سازی ذرات، به نام الگوریتم انطباقی بهینه‌سازی ذرات افزایشی کاهشی، پیشنهاد شده است. این الگوریتم همواره در روند بهینه‌سازی به‌طور انطباقی با کاهش یا افزایش تعداد ذرات الگوریتم و محدوده جستجوی مؤثر، توانایی یافتن و دنبال کردن تعداد بهینه متغیر با زمان در محیط‌های غیرخطی و پویایی که تغییرات آن قابل آشکارسازی نیست را دارد. علاوه بر این، تعاریف جدیدی به نام ناحیه جستجوی متمرکز با هدف برجسته کردن فضاهای امیدبخش برای سرعت‌بخشیدن به فرایند جستجوی محلی و جلوگیری از همگرایی زودرس و شاخص موفقیت به‌عنوان معیاری برای چگونگی رفتار ناحیه جستجوی متمرکز، نسبت به شرایط محیطی، تعریف شده است. نتایج حاصل از الگوریتم پیشنهادی، روی تابع محک قله‌های متحرک ارزیابی شده و با نتایج چندین الگوریتم معتبر مقایسه شده است. نتایج، نشان‌دهنده تأثیر مثبت مکانیزم‌های انطباقی به‌کارگرفته شده از جمله کاهش و افزایش ذرات و محدوده جستجو بر زمان یافتن و دنبال کردن چندین بهینه در مقایسه با سایر الگوریتم‌های بهینه‌سازی مبتنی بر چندجمعیتی است.

واژه‌های کلیدی: کاهش و افزایش انطباقی ذرات، شعاع جستجو انطباقی، مسائل بهینه‌سازی پویا (DOPs)، جستجوی محلی، چندجمعیتی، الگوریتم بهینه‌سازی ذرات

۱- مقدمه

در حال تغییر با زمان هستند. از این رو الگوریتم‌های بهینه‌سازی نه‌تنها باید پاسخ بهینه سراسری را در فضای جستجوی خاصی پیدا کنند، بلکه نیاز دارند که به‌طور پیوسته تغییرات بهینه سراسری و گاه چندین بهینه نزدیک به آن را در محیط‌های مختلف که تغییرات غیرخطی دارند، دنبال کنند. از این رو الگوریتم‌ها باید توانایی تطابق با تغییرات محیطی را داشته باشند. بنابراین اهداف، چالش‌ها و سنجش عملکرد و توابع محک در

مسائل دنیای واقعی مانند آنچه در طبیعت رخ می‌دهد همواره با گذشت زمان در حال تغییر است؛ از این رو امروزه برای حل مسائل بهینه‌سازی پویا، استفاده از الگوریتم‌های تکاملی و هوش جمعی که برگرفته از طبیعت است، در کانون توجه قرار دارد [۱ و ۲]. در محیط‌های پویا، برخلاف محیط‌های ایستا، مکان بهینه‌ها همواره

یکی دیگر از مشکلات پیش رو در بهینه‌سازی مسائل پویا، مشخص شدن زمان تغییر در محیط برای الگوریتم است. بسیاری از الگوریتم‌های بهینه‌سازی پویا بر مبنای آشکارسازی تغییر در محیط شکل گرفته‌اند و در این دسته از الگوریتم‌ها به محض احساس تغییر در محیط، شروع به پراکنده کردن ذرات و در واقع افزایش تنوع در محیط جدید می‌کنند. حالتی را فرض کنید که فقط قسمتی از کل فضای جستجو دستخوش تغییر شود، در این زمان پیش‌بینی آن زیرفضا و یا آشکارسازی تغییر بسیار دشوار می‌شود و همه عملکردهای الگوریتم را تحت الشعاع قرار می‌دهد. به این دلیل الگوریتم‌هایی که نیازی به آشکارسازی تغییرات در محیط نداشته باشند، در محیط‌های گوناگون عملکرد بهتری از خود نشان می‌دهند.

در این مقاله برای استفاده مؤثر از روش چندجمعیتی در محیط‌های غیرخطی و تغییرپذیر با زمان الگوریتم AidPSO^۱ پیشنهاد شده است. الگوریتم معرفی شده بر مبنای الگوریتم PSO^۲ برای محیط‌های پویا طراحی شده است و برای مشکلاتی مثل چگونگی ایجاد زیرجمعیت‌ها، جلوگیری از همگرایی زودرس، اجتماع بیش از حد افراد و تعداد آنها و چگونگی تطبیق رفتار افراد با شرایط محیطی راه کارهای نوینی ارائه شده است.

برای ارزیابی الگوریتم پیشنهادی، مسأله قله‌های متحرک که یکی از معروف‌ترین مسائل بهینه‌سازی در محیط‌های پویا است، به کار گرفته شده است [۳۲ و ۳۱]. برای ارزیابی، عملکرد AidPSO با چندین الگوریتم که از روش‌های چندجمعیتی برای حل مسائل بهینه‌سازی پویا استفاده کرده‌اند، مقایسه شده است. همچنین برای بررسی مکانیزم‌های انطباقی که رفتار الگوریتم را در شرایط محیطی مختلف کنترل می‌کند، آزمایش‌هایی انجام شده است.

سازمان‌دهی این مقاله به این صورت است که در بخش (۲) به مرور روش‌های چندجمعیتی که برای حل مسائل پویا توسعه یافته‌اند، می‌پردازیم. ساختار الگوریتم پیشنهادی AidPSO در بخش (۳) بیان می‌کنیم. بخش (۴) شامل نتایج به دست آمده از عملکرد الگوریتم در مواجهه با تابع محک قله‌های متحرک با تنظیم‌های مختلف است و در نهایت مقاله با نتیجه‌گیری و بحث در بخش (۵) به پایان می‌رسد.

۲- بهینه‌سازی با روش‌های چندجمعیتی

در این بخش، تعدادی از الگوریتم‌هایی که در این زمینه مطرح شده‌اند، مرور می‌شوند.

الگوریتم بهینه‌سازی ذرات خوشه‌ای^۴ در تحقیقات [۲۹ و ۳۰] ارائه

بهینه‌سازی محیط‌های پویا با محیط‌های ایستا کاملاً متفاوت است. در محیط‌های ایستا، همگرایی زودرس، گیرافتادن در بهینه محلی و مصالحه بین اکتشاف و استخراج، از جمله چالش‌هایی است که با آن برخورد می‌کنیم. در محیط‌های پویا علاوه بر چالش‌های قبلی آشکارسازی تغییر در محیط، تبدیل بهینه محلی به بهینه سراسری و برعکس، از دست دادن تنوع بعد از یک تغییر، حافظه منسوخ شده بعد از یک تغییر، احاطه بیش از یک بهینه توسط یک زیرجمعیت و بی‌اطلاعی از میزان و نحوه تغییرات در محیط، تغییر قسمتی از محیط و بی‌اطلاعی از زمان تغییر در محیط نیز نام برد.

در سال‌های اخیر، روش‌های مختلفی برای ارتقای الگوریتم‌های تکاملی سنتی که در محیط‌های ایستا کاربرد داشتند، پیشنهاد شده‌اند. این روش‌ها را می‌توان به ۵ گروه دسته‌بندی کرد:

- افزایش تنوع بعد از تغییر در محیط [۹-۳]
- حفظ تنوع در طول اجرا [۹-۱۴]
- طرح‌های حافظه [۱۵ و ۱۶]
- روش‌های چندجمعیتی [۲۳-۱۷]
- هیبریداسیون [۲۶-۲۴]

در بسیاری از تحقیقات که برای حل مسائل بهینه‌سازی پویا صورت گرفته است، یافتن و دنبال کردن مجموعه‌ای از بهینه‌ها نسبت به یک بهینه سراسری را مؤثرتر دانسته‌اند [۲۹-۲۷]. بدین دلیل که دنبال کردن بهینه سراسری در مجموعه‌ای از بهترین بهینه‌ها، هنگامی که محیط متغیر با زمان بوده، کارآمدتر است. در همین راستا به نظر می‌رسد استفاده از روش‌های چندجمعیتی، برای یافتن و دنبال کردن چندین بهینه محلی نزدیک به بهینه سراسری در بهینه‌سازی مسائل پویا نسبت به سایر روش‌ها کارایی بهتری داشته باشد. در این روش، کل فضای جستجو به زیرفضاهایی^۱ تقسیم می‌شود. هر زیرفضا با عنوان زیرجمعیت شناخته می‌شود که یک یا تعداد کمی بهینه محلی را پوشش می‌دهد. از این پس الگوریتم به‌طور جداگانه ذرات هر زیرجمعیت را به‌روزرسانی کرده و به جستجوی بهینه بهتر می‌پردازد. موضوع چالش برانگیز در روش‌های چندجمعیتی چگونگی ایجاد تعداد مناسب زیرجمعیت و تعداد مناسب افراد برای پوشش دادن زیرفضاهای مختلف در فضای جستجو است. به‌عنوان نمونه در تحقیقات لی و یانگ روش خوشه‌بندی سلسله‌مراتبی، برای تقسیم خودکار فضای جستجو به زیرجمعیت‌ها به کار گرفته شده است [۲۹ و ۳۰].

محیط‌های پویا، ذرات به دو گروه ذرات آزاد و ذرات متمرکز تقسیم می‌شوند.

وظیفه ذرات آزاد، جستجوی همیشگی در تمام فضای جستجو است. زمانی که این ذرات به سمت همگرایی پیش بروند، تعدادی از آنها که در محدوده‌ای به نام ناحیه جستجوی متمرکز^۷ (FSZ) قرار دارند، به‌عنوان ذرات متمرکز معرفی می‌شوند. محدوده ناحیه جستجوی متمرکز حول بهترین ذره متمرکز با شعاعی با نام شعاع جستجوی انطباقی^۸ (ASR) مشخص می‌شود. با به‌روزرسانی ذرات متمرکز در صورت بهبود بهترین ذره، مرکز FSZ نیز به مکان بهترین ذره متمرکز تغییر پیدا می‌کند و سایر ذرات متمرکز این ناحیه، جستجوی متمرکز ملزم به قرارگرفتن و جستجو در این زیرفضا می‌شوند. هم‌زمان تمامی ذرات آزاد برای یافتن نقاط بهینه دیگر در فضای جستجو پراکنده می‌شوند. درواقع وظیفه ذرات آزاد جستجوی ابتدایی و یافتن زیر فضاهای محتمل و وظیفه ذرات متمرکز، یافتن و بهبود بهینه و دنبال کردن آن در حین تغییرات محیط است. تعداد ذرات متمرکز و مقدار شعاع جستجوی انطباقی با معیاری با نام شاخص موفقیت تحت کنترل قرار می‌گیرد. بدین صورت است که همواره تعداد ذرات به‌طور تطبیقی درحال تغییر است.

با توجه به اهمیت حفظ تنوع در حل مسائل بهینه‌سازی پویا و وابستگی آن به تشخیص تغییرات محیط، حائز اهمیت است که عملکرد الگوریتم‌هایی که برای محیط‌های پویا پیشنهاد می‌شود وابسته به شناسایی تغییرات در محیط نباشند. الگوریتم پیشنهادی نیازی به کشف تغییر در محیط ندارد و همواره به‌صورت تطبیقی خود را با شرایط محیطی وقف می‌دهد.

۳-۱- الگوریتم انطباقی بهینه‌سازی ذرات افزایشی کاهشی

در این بخش، الگوریتم جدیدی به نام الگوریتم انطباقی بهینه‌سازی ذرات افزایشی کاهشی برای بهینه‌سازی در محیط‌های پویا (تغییرپذیر با زمان) معرفی شده است. این الگوریتم به‌دلیل مکانیزم کوتاه و انطباق سریع با شرایط محیطی، برای حل مسائل بهینه‌سازی پویا با سرعت تغییرات بالا نیز مناسب است. در الگوریتم ۱ چهارچوب کلی الگوریتم نشان داده شده است.

حلقه اصلی الگوریتم پیشنهادی با مقداردهی اولیه برای ذرات آزاد شروع می‌شود و در مرحله بعدی مکان‌های این ذرات توسط PSO مرسوم به‌روز می‌شود. در ادامه، همگرایی ذرات آزاد بررسی شده و در صورت تشخیص امکان همگرایی، ذراتی که در ناحیه جستجوی متمرکز (FSZ) یعنی در محدوده شعاع جستجوی انطباقی (ASR) حول بهترین ذره باشند، به‌عنوان ذرات متمرکز شناخته شده و یک

شده است که در آن از یک روش خوشه‌بندی سلسله‌مراتبی برای تقسیم جمعیت اولیه به زیرجمعیت‌هایی برای پوشش‌دادن به مناطق مختلف محلی استفاده شده است.

در [۳۳] الگوریتم PSO چندجمعیتی براساس مدل جزیره‌ای معرفی شده است؛ به‌طوری‌که ذرات به‌طور منظم بین جمعیت‌های مختلف مهاجرت می‌کنند. در [۲۴] الگوریتم متفاوت عمل کرده و ارتباط بین جمعیت‌ها فقط زمانی رخ می‌دهد که تغییر در محیط مشاهده شود.

در پژوهش [۳۴] کل جمعیت به دو دسته جمعیت والدین که موظف به جستجوی فضای جستجو و جمعیت فرزندان که موظف به دنبال کردن بهینه‌ها هستند، تقسیم شده‌اند. جمعیت والدین همواره در حال بررسی شرایط برای ایجاد جمعیت کودکان هستند. قابل‌ذکر است که تعداد کل افراد همواره ثابت است. الگوریتم بهینه‌سازی چندجمعیتی سریع^۹ [۳۵] با این تفاوت که جمعیت والدین به‌عنوان جمعیت پایه بعد از تغییر در محیط به‌دنبال مناطق محتمل‌تر می‌شود و جمعیت کودکان برای جستجوی محلی در این مناطق محتمل استفاده می‌شوند، ارائه شد. ایده‌ای مشابه در [۳۶] معرفی شده است که در آن جمعیت کودکان در صورت مؤثرنبودن تا زمانی که تغییر در محیط احساس نشود، به‌اصطلاح در خواب زمستانی می‌روند.

بلکوال و برانک از خواص اتم‌ها و اصل دافعه ذرات با بار همنام برای حفظ تنوع در جمعیت‌ها استفاده کرده‌اند [۳۷]. بدین صورت که جمعیت‌های باردار برای پوشش‌دادن بهینه‌ها به‌صورت مجزا معرفی شده‌اند. در نسخه بهبودیافته [۳۸] دو قانون ابتکاری برای افزایش تنوع اضافه شده است. به‌موجب یکی از قوانین، در زمان ایجاد تغییر در محیط، تعداد ذرات کوانتوم افزایش و تعداد مسیرهای ذرات کاهش می‌یابد. در قانون دوم ذرات با عملکرد نامناسب مقداردهی اولیه و یا متوقف می‌شوند.

همچنین بسیاری از تحقیقاتی [۳۹] که تاکنون برای حل مسائل بهینه‌سازی پویا^{۱۰} (DOPs) بر مبنای الگوریتم‌های تکاملی صورت گرفته است، مبتنی بر تشخیص تغییر در محیط [۲۹]، [۳۰] و [۳۷] یا پیش‌بینی تغییرات با فرض اینکه تغییرات از یک الگوی خواص تبعیت می‌کند [۴۰]، توسعه یافته‌اند. هنگامی که تغییرات تشخیص داده و یا پیش‌بینی شد، آنگاه از استراتژی‌های گوناگونی برای افزایش تنوع بهره می‌گیرند.

۳- الگوریتم پیشنهادی

در الگوریتم پیشنهادی برای همگام‌سازی PSO مرسوم برای

در گام بعدی، موقعیت ذرات آزاد از طریق الگوریتم PSO که اولین بار از سوی کندی و ابرهاتر پیشنهاد شد [۴۱]، به روزرسانی می‌شود. روند الگوریتم PSO در الگوریتم ۲ نشان داده شده است.

الگوریتم ۲: الگوریتم بهینه‌سازی ذرات.

```

Algorithm 2 PSO
1 Evaluate the fitness of each particle
2 for each particle i do
3   Update particle i according to (1) and (2);
4   if  $f(\vec{x}_i) < f(\vec{x}_{pbest_i})$  then
5      $\vec{x}_{pbest_i} := \vec{x}_i$ ;
6   if  $f(\vec{x}_i) < f(\vec{x}_{gbest})$  then
7      $\vec{x}_{gbest} := \vec{x}_i$ ;
8   end if
9 end if
10 end for
    
```

هر ذره i با یک بردار سرعت \vec{v}_i بردار موقعیت \vec{x}_i نمایش داده می‌شود و از طریق نسخه [۴۲] با وزن انرسی^۱ به صورت زیر به روزرسانی می‌شود.

$$v_i^d = \omega v_i^d + c_1 r_1 (x_{pbest_i}^d - x_i^d) + c_2 r_2 (x_{gbest}^d - x_i^d) \quad (1)$$

$$x_i^d = x_i^d + v_i^d \quad (2)$$

در رابطه فوق x_i^d موقعیت کنونی، x_i^d موقعیت قبلی، v_i^d سرعت کنونی و v_i^d سرعت قبلی ذره i ام در بعد d ام است. $\vec{x}_{pbest_i}^d$ بهترین موقعیت ذره i ام تاکنون و \vec{x}_{gbest}^d بهترین موقعیت در بین کل ذرات است. $\omega \in (0,1)$ وزن انرسی است و میزان تأثیر سرعت قبلی در سرعت کنونی ذره را تعیین می‌کند. r_1 و r_2 اعدادی تصادفی بین صفر و یک هستند. c_1 و c_2 با نام ثابت شتاب شناخته می‌شوند و به ترتیب میزان تبعیت ذره از بهترین خود (جزء شناختی) و بهترین جمعی (جزء اجتماعی) را نشان می‌دهد.

در گام بعدی، حلقه اصلی الگوریتم پیشنهادی، روند همگرایی ذرات آزاد بررسی می‌شود که در الگوریتم ۳ نمایش داده شده است. شرط همگرایی اولیه، بدین صورت است که میزان تغییرات موقعیت x_{gbest} ذرات آزاد و برازندگی آن نسبت به مرحله قبل سنجیده می‌شود. در صورتی که تغییرات x_{gbest} کمتر از $r_{conv}/5$ و تغییرات برازندگی آن کمتر از r_{conv} باشد، الگوریتم امکان همگرایی ذرات آزاد را تشخیص می‌دهد. در این مرحله به بررسی موقعیت بهترین نقطه یافت شده پرداخته می‌شود. در صورتی که نقطه مذکور در هریک از FSZ ها واقع شده باشد و مقدار برازندگی آن بهتر از برازندگی مرکز FSZ واقع در آن باشد، FSZ مذکور و به تبع آن ذرات متمرکز عضو حذف و FSZ جدید با تعداد ذرات آزادی که در محدوده ASR به مرکزیت x_{gbest} باشند، تشکیل می‌شود و تمامی ذرات آزاد برای جستجوی نواحی محتمل دیگر در کل

ناحیه جستجوی متمرکز جدید ایجاد می‌شود. حال، ذرات متمرکز هر ناحیه جستجوی متمرکز به طور جداگانه توسط PSO به روزرسانی می‌شود و در صورت بهبود بهینه هر FSZ، مرکز ناحیه جستجوی متمرکز به بهینه مناسب‌تر یافت شده انتقال می‌یابد. در مرحله بعدی نواحی جستجوی متمرکز پس از طی زمان خاص به نام زمان تکامل، از جهت روی هم افتادگی بررسی می‌شوند و امکان اصلاح شعاع جستجوی انطباقی را فراهم می‌آورد. در مرحله بعدی به FSZ هایی که بهینه خود را در اثر تغییرات شدید شرایط محیطی از دست داده‌اند، زمان ارتقا تخصیص داده می‌شود تا دوباره بهینه خود را بیابند و بلافاصله حذف نشوند. اختصاص زمان ارتقا بدین دلیل است که در اکثر موارد، پس از تغییر در محیط مکان جدید بهینه به نواحی اطراف محل قبلی خود تغییر می‌یابد. در این حالت با تخصیص زمان ارتقا و افزایش شعاع انطباقی جستجو و تعداد ذرات متمرکز، می‌توان با سرعت بیشتری بهینه مدنظر را ردیابی کرد. نهایتاً در مرحله بعدی با استفاده از شاخص موفقیت، شعاع جستجوی انطباقی و تعداد ذرات متمرکز و آزاد برای شرایط کنونی محیط محاسبه می‌شود.

الگوریتم ۱: چهارچوب کلی الگوریتم پیشنهادی.

```

Algorithm 1 AidPSO
1 Initialize the free particles
2 while stop criteria is not satisfied do
3   for free particles do
4     PSO();
5   end for
6   convergenceChecking (free particles)
7   if FSA exist then
8     for each FSZ[i] do
9       PSO(FSZ[i])
10      Update FSZ centers
11      Check velocity of focus particles
12    end for
13  end if
14  ASROverlapChecking (FSZ,ASR)
15  RemoveUselessFSZ
16  CalcASR&NumOfFocusPartc
17  CalcNumOfFreePartc
18 end while
    
```

در الگوریتم پیشنهادی، ذرات به دو گروه ذرات آزاد و ذرات متمرکز تقسیم می‌شوند. ذرات آزاد در مرحله اول الگوریتم به تعداد از پیش تعیین شده، مقداردهی اولیه می‌شوند و پس از آن همواره تعداد آنها از طریق الگوریتم تعیین می‌شود. یکی از ویژگی‌های الگوریتم پیشنهادی، متغیر بودن تعداد ذرات آزاد و متمرکز در طول زمان، بسته به شرایط محیط است. بدین سان تعداد کل ذرات در طول اجرای الگوریتم، همواره تابع شرایط محیطی است و هنگامی که الگوریتم به سمت بهینه همگرا می‌شود، تعداد ذرات به تناسب کاهش می‌یابد. این ویژگی الگوریتم را از قید به روزرسانی همیشگی تعداد بالای ذرات جدا می‌سازد.

هستند، در نظر گرفته می‌شوند. دسته‌های مذکور باید به اصطلاح به پایداری نسبی رسیده باشند؛ یعنی زمان تکامل که مربوط به FSZ های جدید و زمان ارتقا که مربوط به FSZ هایی است که بهینه خود را در شرایط محیطی جدید از دست داده‌اند، طی کرده باشند. حال برای اجتناب از تحت‌تأثیر قرارگرفتن الگوریتم از بهینه‌های محلی، میانگین برانزندی بهینه‌های یافت‌شده از طریق سایر FSZ ها محاسبه می‌شود. در صورتی که برانزندی مراکز دو FSZ مذکور بهتر از ضریبی (P_{mean}) از میانگین برانزندی مراکز سایر FSZ ها بود، شعاع جستجو به نصف فاصله مراکز این دو FSZ کاهش می‌یابد تا هر دو بهینه خود را در بر گرفته و دنبال کنند.

حال اگر برانزندی یکی یا هر دو FSZ بدتر از ضریبی (P_{mean}) از میانگین برانزندی مراکز سایر FSZ ها بود، FSZ ای که برانزندی بهینه یافت‌شده آن با ذرات متمرکز مقدار بدتری نسبت به دیگری داشت، به‌عنوان بهینه محلی تلقی شده و FSZ مذکور و ذرات متمرکز مربوط به آن حذف می‌شود. بدین‌صورت هم از جستجوی چند FSZ در یک ناحیه از فضای جستجو جلوگیری شده و هم از گیرافتادن در بهینه‌های محلی اجتناب می‌شود که نتیجه آن افزایش سرعت الگوریتم در یافتن نقاط بهینه بیشتر در کل فضای جستجو در زمان کمتری است. در الگوریتم ۴ روند بررسی هم‌پوشانی FSZ ها نمایش داده شده است.

الگوریتم ۴: بررسی هم‌پوشانی نواحی جستجوی متمرکز (FSZ).

Algorithm 4 ASROverlapChecking (FSZ,ASR)

```

1  if FSZ[i] & FSZ[j] are stable and distance between FSZ[i]
   center and FSZ[j] center < 2*ASR then
2   $f_{mean}(center_{FSZ}) = \text{Calculation of mean fitness other}$ 
   centers
3  if  $f(center_{FSZ[i]})$  and  $f(center_{FSZ[j]})$  better than
    $P_{mean} * f_{mean}(center_{FSZ})$  and  $ASR > 1$  then
4   $ASR = (\text{distance between FSZ[i] center and FSZ[j]$ 
   center) / 2
5  else
6  delete FSZ with worst fitness between FSZ[i] and
   FSZ[j]
7  end if
8  end if

```

ایده دیگر به‌کارگرفته‌شده در الگوریتم پیشنهادی که در الگوریتم ۵ نمایش داده شده است، اختصاص زمان ارتقا برای FSZ هایی است که پس از تغییرات شدید در محیط دیگر شاهد نقطه بهینه در منطقه خود نیستند و یا ذرات متمرکزشان در بهینه محلی گیر افتاده‌اند. زمان ارتقا این فرصت را به FSZ های مذکور می‌دهد که بهینه خود را پیدا و دنبال کنند. همچنین درباره ذرات متمرکزی که در بهینه محلی گیر افتاده‌اند، با افزایش سرعت در زمان ارتقا فرصت دارند از بهینه محلی آزاد شوند. در اینجا نیز ضریبی (P_{mean}) از میانگین برانزندی نقاط بهینه یافت‌شده به‌عنوان معیار سنجش معرفی می‌شود و در هر مرحله با مقایسه بدترین بهینه در

فضای جستجو پراکنده می‌شوند. از سوی دیگر، زمانی که مقدار برانزندی x_{gbest} بدتر از برانزندی مرکز FSZ واقع در آن باشد، تمامی ذرات آزاد پراکنده می‌شوند.

پس از ارضای شرط اولیه همگرایی، اگر موقعیت بهترین نقطه ذرات آزاد، در هیچ‌یک از FSZ ها واقع نشده باشد، آنگاه به بررسی مجموع فواصل دو ذره آزاد که کمترین فاصله را نسبت به x_{gbest} دارند، پرداخته می‌شود. در صورتی که این مجموع نیز مقداری کمتر از r_{conv} داشته باشد، نشانه آن است که ذرات آزاد رو به همگرایی هستند و x_{gbest} به‌عنوان مرکز FSZ در نظر گرفته شده و به تعداد ذرات متمرکز از ذرات آزادی که در محدوده ASR نسبت به این مرکز باشند، به‌عنوان ذرات متمرکز آن FSZ معرفی می‌شوند و تمامی ذرات آزاد برای جستجوی مناطق محتمل دیگر در کل فضای جستجو پراکنده می‌شوند. همچنین برای آنکه در مراحل بعدی الگوریتم این دسته را به‌عنوان دسته اضافی حذف نکند، زمان تکامل در نظر گرفته می‌شود که طی آن دسته مدنظر فرصت دارد مقدار بهینه خود را بهبود بخشد.

الگوریتم ۳: بررسی همگرایی ذرات آزاد.

Algorithm 3 convergenceChecking ()

```

1  if  $(f(x_{gbest}^{t-1}) - f(x_{gbest}^t) < r_{conv})$  and  $(\|x_{gbest}^{t-1} - x_{gbest}^t\| <$ 
    $r_{conv}/5)$  then
2  if  $x_{gbest}$  was within FSZ[i] then
3  if  $f(x_{gbest})$  better than  $f(\text{center}_{FSZ[i]})$  then
4  Replace selected free particles with focus
   particles in FSZ[i]
5  re-initialize the free particles
6  else
7  re-initialize the free particles
8  end if
9  else
10 create a new FSZ with selected free particles as the
   focus particles
11 re-initialize the free particles
12 end if
13 end if

```

در گام بعدی موقعیت ذرات متمرکز هر FSZ به‌طور مجزا با الگوریتم PSO به‌روزرسانی می‌شوند و در صورت بهبود بهترین ذره هر FSZ، مرکز آن FSZ به موقعیت بهترین ذره تغییر مکان می‌دهد. بدین‌صورت موقعیت مکانی FSZ ها نیز در فضای جستجو به‌روزرسانی می‌شود. الگوریتم همچنین سرعت ذرات متمرکز را رصد می‌کند تا هر زمان نزدیک به صفر شدند، با پراکنده‌کردن ذرات متمرکز مذکور در محدوده FSZ خودشان از گیرافتادن در بهینه محلی جلوگیری کند.

در الگوریتم پیشنهادی مکانیسمی برای جلوگیری از هم‌پوشانی نواحی جستجوی متمرکز و موازی‌کاری ذرات متمرکز به‌کار گرفته شده است. در این روش، فواصل بین مراکز FSZ ها محاسبه شده و دو FSZ همسایه که فاصله مراکزشان کمتر از دو برابر ASR از هم

جستجوی متمرکزتری حاصل شود.

$$ASR^{t+1} = ASR^t - \frac{Success_Index}{1 + Success_Index} * ASR^t \quad (۶)$$

نهایتاً در حالتی که هیچ FSZ ای در دوره ارتقا نباشد، برای جستجوی متمرکزتر و افزایش سرعت الگوریتم، ARS و تعداد ذرات متمرکز (Num_FP) با توجه به بهبود شاخص موفقیت طبق روابط زیر کاهش می‌یابند.

$$\Delta Success_Index = Success_Index^{t-1} - Success_Index^t \quad (۷)$$

$$ASR^{t+1} = ASR^t + \Delta Success_Index * ASR^t$$

$$Num_FP^{t+1} = Num_FP^t - Success_Index * Num_FP^t \quad (۸)$$

برای افزایش سرعت الگوریتم، همواره تعداد ذرات آزاد براساس تعداد نقاط بهینه واقعی یافت‌شده تعیین می‌شود. تعداد نقاط بهینه زمانی سنجیده می‌شود که تمامی دسته‌ها دوره ارتقای خود را سپری کرده‌اند. در این حالت معیار، تعداد FSZ هایی هستند که در دوره تکامل نباشند و به اصطلاح به پایداری نسبی رسیده‌اند. طبق رابطه (۹) تعداد ذرات آزاد^{۱۱} (Num_FreeP) برای یافتن نواحی محتمل جدید براساس تعداد نقاط بهینه^{۱۲} (Num_optm) یافت شده محاسبه می‌شود.

$$Num_FreeP = 15 + 2\log(Num_optm) \quad (۹)$$

۴- نتایج شبیه‌سازی

در این بخش الگوریتم پیشنهادی برای حل مسأله MPB [۳۱] به کار گرفته شده است. ابتدا به معرفی مسأله قله‌های متحرک و نحوه اندازه‌گیری عملکرد الگوریتم‌های پویا پرداخته می‌شود. پس از آن آزمایش‌های صورت گرفته به دو بخش تقسیم می‌شوند. هدف از قسمت اول، بررسی مکانیزم کار پارامترهای انطباقی AidPSO در مواجهه با مسأله MPB است و در قسمت دوم از آزمایش‌ها، عملکرد AidPSO با تعدادی از الگوریتم‌هایی که در این زمینه معرفی شده‌اند، مقایسه شده است. نتایج تمام الگوریتم‌هایی که در این مقاله نشان داده شده‌اند، برگرفته از نتایج و پیشنهادها مطالعات پیشین است.

در الگوریتم PSO ثابت‌های شتاب c_1 و c_2 به ترتیب میزان تبعیت ذره از بهترین خود (جزء شناختی) و بهترین جمعی (جزء اجتماعی) را نشان می‌دهد و وزن انرسی سرعت همگرایی را تغییر می‌دهد. با توجه به وظایف مختلفی که برای ذرات آزاد و ذرات متمرکز در AidPSO در نظر گرفته شده است، مقادیر ثابت‌های شتاب و وزن انرسی برای به‌روزرسانی ذرات آزاد که وظیفه یافتن

FSZ ها با این معیار در صورتی که مقداری نامناسب‌تر داشته باشند، FSZ مربوطه همراه با ذرات متمرکز آن پس از طی زمان ارتقا، حذف می‌شوند.

الگوریتم ۵: حذف نواحی جستجوی متمرکز (FSZ) غیرضروری.

Algorithm 5 RemoveUselessFSZ

```

1  $f_{mean}(center_{FSZ}) =$  Calculation of mean all fitness centers
2 if FSZ[i] is stable and worst  $f(center_{FSZ[i]}) > f_{mean}(center_{FSZ})$ 
   then
3   delete FSZ with worst fitness FSZ[i]
4 end if

```

یکی از ویژگی‌های بارز این الگوریتم که سرعت آن را در یافتن بهینه افزایش داده است، رصد شرایط محیطی و به تبع آن تنظیم رفتار الگوریتم با مدیریت تعداد ذرات و محدوده جستجو است. بدین‌سان شاخص موفقیت^{۱۱} به‌عنوان معیاری برای میزان بهینه‌سازی الگوریتم تعریف شده است. در این شاخص با توجه به میانگین مقدار بهینه دسته‌ها و مقایسه آن با بدترین مقدار بهینه یافت‌شده، معیاری از کلیت رفتار دسته‌ها در مواجهه با شرایط محیطی به دست می‌آید (رابطه ۳).

$$Success_Index = \frac{mean_fit - worst_fit}{mean_fit} \quad (۳)$$

حال که معیاری از رفتار الگوریتم و شرایط محیطی به دست آمد، برای بهینه‌کردن هرچه بیشتر عملکرد الگوریتم باید میزان شعاع جستجو و تعداد ذرات متمرکز دسته‌ها را با شرایط محیطی انطباق داد. در صورتی که در محیط تغییراتی ایجاد شود، برخی از FSZ ها بهینه خود را از دست می‌دهند و در نتیجه مقدار شاخص موفقیت کاهش می‌یابد. همان‌گونه که قبلاً اشاره شد در این حالت زمان ارتقاء برای FSZ های مذکور در نظر گرفته می‌شود تا قبل از حذف شدن از سوی الگوریتم به‌عنوان دسته غیرضروری، فرصت وقف با شرایط محیطی جدید و در نتیجه یافتن بهینه تحت پوشش خود را داشته باشند. از آنجایی که الگوریتم اطلاعی از میزان تغییرات در محیط ندارد، برای سرعت‌بخشیدن به روند یافتن بهینه با توجه به مقدار شاخص موفقیت اقدام به افزایش شعاع جستجوی انطباقی (ASR) و تعداد ذرات متمرکز^{۱۱} (Num_FP) در FSZ طبق روابط زیر می‌کند.

$$ASR^{t+1} = ASR^t + Success_Index * ASR^t \quad (۴)$$

$$Num_FP^{t+1} = Num_FP^t + Success_Index * Num_FP^t \quad (۵)$$

در حین زمان ارتقا FSZ ها، بهینه یافت‌شده بهبود می‌یابد و نمود این بهبود در افزایش شاخص موفقیت پدیدار می‌شود. در این حال، شعاع جستجوی انطباقی (ASR) طبق رابطه (۴) کاهش می‌یابد تا

است. ارتفاع قله به صورت تصادفی در بازه [30,70] و پهنای آن در بازه [1,12] تغییر می‌کند.

جدول ۱: تنظیمات پیش فرض مسأله قله‌های متحرک.

Parameter	Value
Number of peaks, p	[1,200]
Change frequency, U	5000
Height severity	7.0
Width severity	1.0
Peak shape	Cone
Basic function	No
Shift length, s	1.0
Number of dimensions, D	5
Correlation coefficient, λ	0
Peaks location range	[0,100]
Peak height, H	[30,70]
Peak width, W	[1,12]
Initial value of peaks	50.0

۲-۴- اندازه گیری عملکرد الگوریتم

برای اندازه‌گیری عملکرد الگوریتم‌ها در محیط‌های پویا چندین معیار سنجش معرفی شده است [۴۳]. به منظور ایجاد نتایج قابل‌قیاس با سایر الگوریتم‌ها در این زمینه، از معیار خطای آفلاین^{۱۶} که به صورت میانگین اختلاف مقدار بهینه یافت شده توسط الگوریتم با مقدار بهینه سراسری در هر محیط تعریف می‌شود، استفاده شده است.

$$OE = \frac{1}{K} \sum_{k=1}^K (h_k - f_k) \quad (15)$$

در رابطه فوق f_k بهترین جواب یافت شده توسط الگوریتم تا قبل از تغییر k ام در محیط و h_k مقدار بهینه محیط k ام است. OE میانگین اختلاف f_k و h_k در کل K تغییر در محیط است. همه نتایج گزارش شده برای بیش از ۵۰ بار اجرای برنامه برای ۱۰۰ تغییر در محیط است.

۳-۴- بررسی پارامترهای انطباقی الگوریتم

برای نمایش روند تصمیم‌گیری الگوریتم برای مقادیر شعاع جستجوی انطباقی و تعداد ذرات آزاد و متمرکز در شرایط مختلف محیطی آزمایشاتی روی MPB با مقادیر پیش فرض صورت گرفته است.

همان‌گونه که قبلاً اشاره شد، درصد بیشتری از بهینه‌ها در زمان تغییر در محیط، جای خود را در حوالی مکان قبلی تغییر می‌دهند. از این رو می‌توان با افزایش هم‌زمان محدوده جستجو و تعداد ذرات هر ناحیه جستجوی متمرکز با سرعت و دقت بیشتری بهینه‌ها را پس از تغییر در محیط پیدا کرد. در شکل ۱ نحوه عملکرد

نواحی محتمل در کل فضای جستجو و ذرات متمرکز که موظف به یافتن و دنبال کردن بهینه در FSZ خودشان هستند، متفاوت در نظر گرفته شده است. مقادیر c_1 ، c_2 و ω ذرات آزاد برای جستجوی کلی به ترتیب ۲، ۲ و ۰/۴ و برای ذرات متمرکز با جستجوی دقیق ۳، ۱ و ۰/۵ در نظر گرفته شده است.

۴-۱- مسأله قله‌های متحرک (MPB)

تابع محک قله‌های متحرک [۳۱] یکی از معروف‌ترین مسائل بهینه‌سازی در محیط‌های پویا است و به طور گسترده‌ای برای ارزیابی الگوریتم‌های بهینه‌سازی پویا به کار گرفته شده است. در مسأله MPB، بهینه می‌تواند با سه ویژگی موقعیت، ارتفاع و عرض قله‌ها تغییر کند. این مسأله در D بعد به صورت زیر تعریف می‌شود.

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

در رابطه (۱۰) $H_i(t)$ و $W_i(t)$ به ترتیب ارتفاع و پهنای قله i در زمان t و $X_{ij}(t)$ j امین عنصر از مکان قله i در زمان t است. پارامتر p مستقلاً برای قله‌های مشخص، از طریق تابع max مخلوط می‌شود. مکان قله در جهتی تصادفی از طریق بردار \vec{v}_i به اندازه یک فاصله s که معرف حساسیت پویایی مسأله است، انتقال می‌یابد. حرکت یک قله تنها، به صورت رابطه زیر تعریف می‌شود.

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

بردار انتقال $\vec{v}_i(t)$ یک ترکیب خطی از بردار تصادفی \vec{r} و بردار انتقال قبلی $\vec{v}_i(t-1)$ بوده و نسبت به طول تغییر s نرمالیزه شده است. مقدار پارامتر همبستگی λ صفر در نظر گرفته شده که نشان‌دهنده ناهمبستگی حرکات قله است. روابط تغییرات یک قله به صورت زیر بیان می‌شود.

$$H_i(t) = H_i(t-1) + height_severity \times \sigma \quad (12)$$

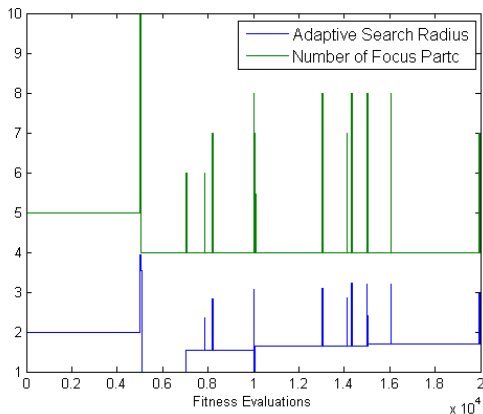
$$W_i(t) = W_i(t-1) + width_severity \times \sigma \quad (13)$$

$$\vec{X}_i(t) = \vec{X}_i(t-1) + \vec{v}_i(t) \quad (14)$$

که در آن σ یک عدد تصادفی با توزیع نرمال و میانگین صفر و واریانس یک است.

تنظیمات پیش فرض تابع محک در این مقاله مانند سایر تحقیقات، در جدول ۱ نشان داده شده است. اصطلاح فرکانس تغییر U به معنای تغییر محیط بعد از U بار ارزیابی تابع برازندگی است. بازه تغییرات مکان قله^{۱۵} در واقع محدوده فضای جستجو در هر بعد

تکرار هفتم ارزیابی برازندگی و ایجاد اولین FSZ، ۵ ذره با برازندگی بهتر به عنوان ذرات متمرکز شناخته شده و پس از آن همه ذرات



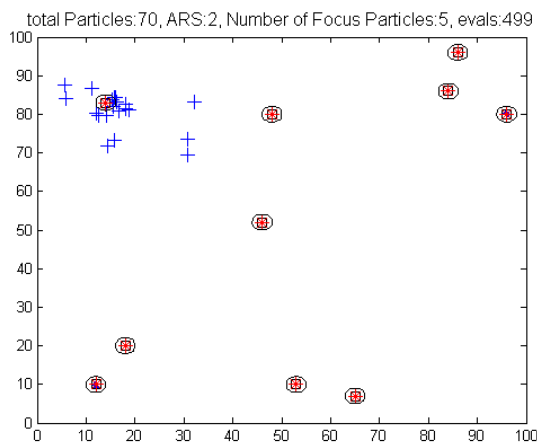
شکل ۱: عملکرد الگوریتم در تنظیم تعداد ذرات متمرکز در هر ناحیه و شعاع جستجوی انطباقی.

آزاد پراکنده می‌شوند و همین روند برای یافتن تمامی قله‌ها ادامه

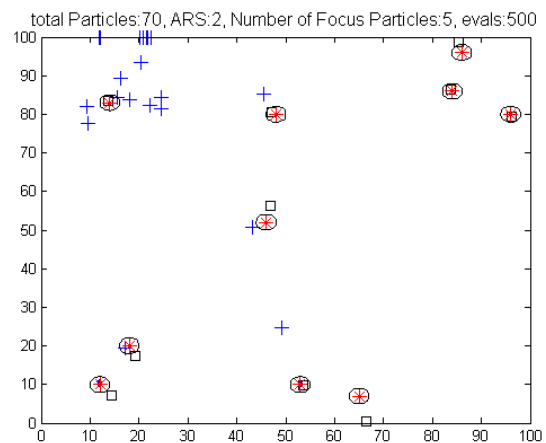
الگوریتم برای تنظیم مقادیر شعاع جستجوی انطباقی و تعداد ذرات متمرکز هر ناحیه جستجوی متمرکز در مواجهه با تغییرات شرایط محیطی نشان داده شده است. همان‌گونه که مشخص است، زمانی که محیط تغییر می‌کند، الگوریتم با توجه به میزان کاهش شاخص موفقیت، اقدام به افزایش شعاع و تعداد ذرات متمرکز هر ناحیه می‌کند. در ادامه با بهبود شاخص موفقیت برای افزایش سرعت الگوریتم و تمرکز برای یافتن بهینه‌های جدید، شعاع و تعداد ذرات متمرکز کاهش می‌یابد.

به منظور تجسم رفتار الگوریتم قبل و پس از تغییر در محیط، شکل ۲ برای محیطی دو بعدی آورده شده است. در این شکل ذرات آزاد با علامت به علاوه آبی، مکان قله‌ها با مربع مشکی، ذرات متمرکز با نقاط آبی، ناحیه جستجوی متمرکز با دایره آبی و مراکز آن با ستاره قرمز رنگ مشخص شده‌اند.

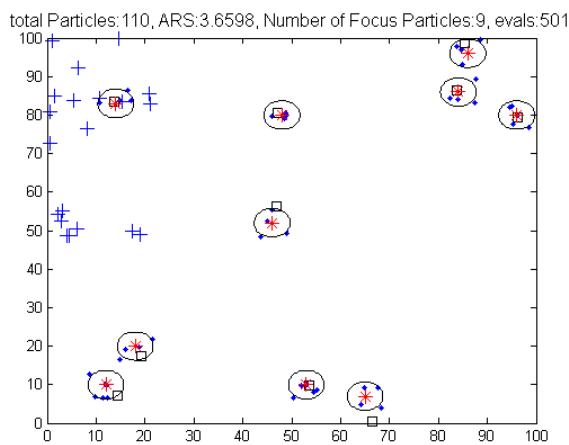
روند بهینه‌سازی از (الف) تا (د) نشان می‌دهد که الگوریتم با ذرات آزاد شروع به کار می‌کند و با پیداشدن اولین منطقه محتمل در



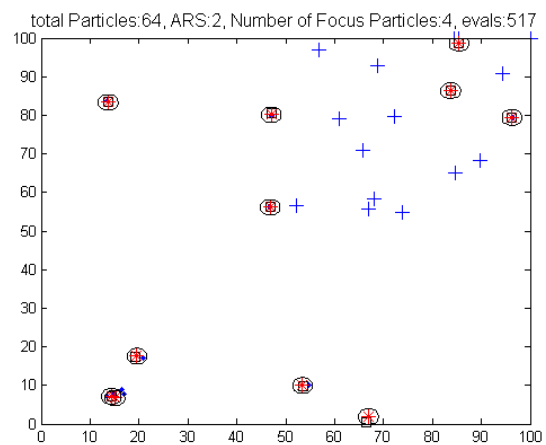
(ب)



(الف)



(د)



(ج)

شکل ۲: روند الگوریتم در یافتن قله‌ها و تغییر تعداد ذرات و شعاع جستجوی انطباقی قبل و پس از تغییر در محیط دو بعدی. (الف) یک تکرار قبل از تغییر در محیط، (ب) ایجاد تغییر در محیط، (ج) یک تکرار پس از تغییر در محیط، (د) هفده تکرار پس از تغییر در محیط.

جدول ۲: تنظیمات اولیه AidPSO برای حل مسأله MPB.

Parameter	Value
r_{conv}	0.5
P_{mean}	0.3
Evolution time	20 iteration
Upgrade time	30 iteration

در ادامه آزمایش‌های انجام‌شده برای نمایش تأثیرات تغییر تعداد قله‌ها، طول تغییر و فرکانس تغییرات محیط U بر خطای آفلاین الگوریتم پیشنهادی آمده است.

در مجموعه آزمایش‌هایی که در جدول ۳ جمع‌آوری شده است، چگونگی برخورد الگوریتم با تغییر تعداد قله‌ها برای مسأله MPB نمایش داده شده است. الگوریتم پیشنهادی بر مبنای خطای آفلاین و انحراف معیار با ۱۴ الگوریتم دیگر مقایسه شده است. مقادیر آورده‌شده برای الگوریتم‌های دیگر، برگرفته از مقالات محققان پیشین و با تنظیمات بهینه است. با توجه به جدول ۳، نتایج برای الگوریتم پیشنهادی برای تمامی تعداد قله‌ها بهتر از سایر الگوریتم‌ها در حل مسأله قله‌های متحرک است.

با دقت در مقادیر خطای آفلاین AidPSO و سایر الگوریتم‌ها در جدول ۳ درمی‌یابیم که با افزایش تعداد قله‌ها، مقدار خطای آفلاین کاهش می‌یابد؛ زیرا با افزایش تعداد بهینه‌های محلی یافت‌شده، احتمال نزدیک‌تر شدن مقدار برآوردگی این بهینه‌ها به برآوردگی بهینه سراسری افزایش می‌یابد و همچنین مناطق بیشتری در فضای جستجو از طریق ذرات متمرکز تحت نظر و جستجو هستند که در نتیجه پس از تغییر در محیط، امکان حضور بهینه سراسری در این مناطق بیش از پیش افزایش می‌یابد.

در جدول ۴ مقادیر خطای آفلاین شش الگوریتم مشابه در این زمینه و الگوریتم AidPSO برای چهار طول تغییر مختلف آورده شده است. تنظیمات محیط همان مقادیر پیش فرض و مقادیر طول

می‌یابد. شکل ۲ (الف) یک گام قبل از ایجاد تغییر در محیط است که تمامی قله‌ها با FSZ ها احاطه شده‌اند. پس از تغییر در محیط مکان قله‌ها تغییر یافته و با کاهش شاخص موفقیت الگوریتم تصمیم به افزایش شعاع جستجو و تعداد ذرات متمرکز در هر FSZ می‌کند. همان‌گونه که در شکل ۲ (ب) مشخص است، با این مکانیزم اکثر قله‌ها پس از تغییر در محیط نیز در ناحیه جستجوی متمرکز قرار می‌گیرند و افزایش مقطعی تعداد ذرات متمرکز سبب افزایش سرعت پیدا کردن قله‌های مربوط به خود می‌شود. در شکل ۲ (ج) شعاع جستجوی انطباقی به مقدار $3/66$ و تعداد ذرات متمرکز به ۹ عدد برای هر FSZ افزایش می‌یابد. پس از بهبود نسبی شاخص موفقیت مقادیر ASR و تعداد ذرات متمرکز به ترتیب ۲ و ۴ کاهش پیدا می‌کند و در نهایت قله‌هایی که در هیچ FSZ ای قرار نگرفته‌اند، از طریق ذرات آزاد پیدا می‌شوند و نواحی جستجوی متمرکزی که قله‌ای را در بر ندارند توسط الگوریتم حذف می‌شود. پس از بهبود شاخص موفقیت شعاع موفقیت شعاع جستجو و تعداد ذرات متمرکز کاهش می‌یابد (شکل ۲ د).

۴-۴- مقایسه با سایر الگوریتم‌ها

در این بخش از آزمایشات، الگوریتم پیشنهادی با سایر الگوریتم‌های مطرح در این مبحث از جمله [29,30] CPSO، [37] mCPSO، [37] mQSO، [43] CESO، [44] rSPSO، [28] FTMPSO، [47] APSO، [46] mPSO، [45] AmQSO، [48]، [49] SFA، [50] PSO-AQ، [51] CDEPSO و CbDE- [52] wCA در حل مسأله MPB با تنظیمات مختلف مقایسه شده است. مقادیر بهینه پارامترهای AidPSO برای حل مسأله MPB با مقادیر پیش فرض در جدول ۲ آمده است.

جدول ۳: مقایسه خطای آفلاین الگوریتم‌ها برای تعداد قله‌های متفاوت در مسأله MPB در فرکانس تغییرات $U = 5000$.

Algorithm	Number of peaks, p							
	1	5	10	20	30	50	100	200
CPSO	0.14(0.11)	0.72(0.30)	1.06(0.24)	1.59(0.22)	1.58(0.17)	1.54(0.12)	1.41(0.08)	1.24(0.06)
mCPSO	4.93(0.17)	2.07(0.08)	2.08(0.07)	2.64(0.07)	2.63(0.08)	2.65(0.06)	2.49(0.04)	2.44(0.04)
mQSO(5,5q)	2.24(0.05)	1.82(0.08)	1.85(0.08)	2.48(0.09)	2.51(0.10)	2.53(0.08)	2.35(0.06)	2.24(0.05)
CESO	1.04(0.00)	-	1.38(0.02)	1.72(0.02)	1.24(0.01)	1.45(0.01)	1.28(0.02)	-
rSPSO	1.42(0.06)	1.04(0.03)	1.50(0.08)	2.20(0.07)	2.62(0.07)	2.72(0.08)	2.93(0.06)	2.79(0.05)
SPSO	2.64(0.10)	2.15(0.07)	2.51(0.09)	3.21(0.07)	3.64(0.07)	3.86(0.08)	4.01(0.07)	3.82(0.05)
AmQSO	2.62(0.10)	1.01(0.09)	1.51(0.10)	2.00(0.15)	2.19(0.17)	2.43(0.13)	2.68(0.12)	2.62(0.10)
mPSO	2.42(0.05)	1.82(0.08)	1.85(0.08)	2.48(0.09)	2.51(0.10)	2.53(0.08)	2.35(0.06)	2.24(0.05)
APSO	0.53(0.01)	1.05(0.06)	1.31(0.03)	1.69(0.05)	1.78(0.02)	1.95(0.02)	1.95(0.01)	1.90(0.01)
FTMPSO	0.18(0.01)	0.47(0.05)	0.67(0.04)	0.93(0.04)	1.14(0.04)	1.32(0.04)	1.61(0.03)	1.67(0.03)
SFA	0.42(0.03)	0.89(0.07)	1.05(0.04)	1.48(0.05)	1.56(0.06)	1.87(0.05)	2.01(0.04)	1.99(0.06)
PSO-AQ	0.34(0.02)	0.80(0.12)	0.89(0.03)	1.45(0.06)	1.52(0.04)	1.77(0.05)	1.95(0.05)	1.96(0.04)
CDEPSO	0.41(0.00)	0.97(0.01)	1.22(0.01)	1.54(0.01)	2.62(0.01)	2.20(0.01)	1.54(0.01)	2.11(0.01)
CbDE-wCA	0.14(0.03)	0.30(0.02)	0.86(0.08)	0.98(0.05)	1.34(0.04)	1.31(0.04)	1.35(0.03)	1.29(0.02)
AidPSO	0.15(0.02)e-10	0.02(0.01)e-1	0.32(0.05)	0.39(0.03)	0.37(0.04)	0.17(0.01)	0.15(0.02)	0.13(0.04)

تغییر ۱، ۲، ۳ و ۵ است. همان‌طور که می‌دانیم با افزایش مقدار طول تغییر، میزان جابه‌جایی مکان قله‌ها پس از تغییر در محیط افزایش می‌یابد و به‌عبارت دیگر، هرچه مقدار طول تغییر بیشتر افزایش یابد، پس از تغییر در محیط، مکان بهینه به فاصله دورتری جابه‌جا می‌شود و این قضیه کار الگوریتم را برای دنبال کردن بهینه سخت‌تر می‌کند. از این‌رو با توجه به جدول ۴ مقادیر خطای آفلاین برای الگوریتم‌ها افزایش می‌یابد. میزان افزایش محدود خطای آفلاین به ازای افزایش طول تغییر برای الگوریتم پیشنهادی، نشان از پایداری بالای الگوریتم در یافتن و دنبال کردن بهینه در هر شرایطی دارد.

جدول ۴: مقایسه خطای آفلاین الگوریتم‌ها برای مقادیر مختلف طول تغییر در مسأله MPB.

در شکل ۳ نمودار مقدار شایستگی براساس تعداد فراخوانی به‌همراه مقایسه با روش‌های دیگر ارائه شده است. برای نمایش توانایی و سرعت انطباق الگوریتم با تغییر شرایط محیطی، فرکانس تغییر محیط ۱۰۰۰ در نظر گرفته شده است. سایر تنظیمات مسأله قله‌های متحرک به‌صورت پیش‌فرض است. ذرات در الگوریتم پیشنهادی با سرعت بیشتری نسبت به سایر الگوریتم‌ها به مقادیر بهینه‌تری همگرا می‌شوند.

Algorithm	Shift severity, s			
	1	2	3	5
CPSO	1.06(0.24)	1.17(0.22)	1.36(0.28)	1.58(0.32)
mCPSO	2.05(0.07)	2.80(0.07)	3.57(0.08)	4.89(0.11)
mQSO(5.5q)	1.85(0.08)	2.40(0.06)	3.00(0.06)	4.24(0.10)
CESO	1.38(0.02)	1.78(0.02)	2.03(0.03)	2.52(0.06)
rSPSO	1.50(0.08)	1.87(0.05)	2.40(0.08)	3.25(0.09)
SPSO	2.51(0.09)	3.78(0.09)	4.96(0.12)	6.76(0.15)
AidPSO	0.32(0.05)	0.41(0.07)	0.46(0.06)	0.49(0.09)

تغییر در فرکانس تغییر محیط U در واقع زمان الگوریتم برای یافتن بهینه در هر محیط قبل از تغییر را مشخص می‌کند. واضح است که مقدار فرکانس کمتر یعنی زمان به‌مراتب کمتر برای یافتن

جدول ۵: مقایسه خطای آفلاین الگوریتم‌ها برای تعداد قله‌های متفاوت در مسأله MPB در فرکانس تغییرات $U = 500$.

Algorithm	Number of peaks, p							
	1	5	10	20	30	50	100	200
mQSO(5.5q)	33.67(3.42)	11.91(0.76)	9.62(0.34)	9.07(0.25)	8.80(0.21)	8.72(0.20)	8.54(0.16)	8.19(0.17)
AmQSO	3.02(0.32)	5.77(0.56)	5.37(0.42)	6.82(0.34)	7.10(0.39)	7.57(0.32)	7.34(0.31)	7.48(0.19)
mPSO	8.71(0.48)	6.69(0.26)	7.19(0.23)	8.01(0.19)	8.43(0.17)	8.76(0.18)	8.91(0.17)	8.88(0.14)
APSO	4.81(0.14)	4.95(0.11)	5.16(0.11)	5.81(0.08)	6.03(0.07)	5.95(0.06)	6.08(0.06)	6.20(0.04)
FTMPSO	1.76(0.09)	2.93(0.18)	3.91(0.19)	4.83(0.19)	5.05(0.21)	4.98(0.15)	5.31(0.11)	5.52(0.21)
SFA	4.72(0.12)	4.88(0.12)	5.11(0.14)	5.72(0.13)	5.97(0.12)	5.94(0.15)	6.15(0.08)	6.18(0.11)
AidPSO	0.35(0.04)	2.29(0.03)	2.53(0.04)	1.89(0.03)	1.68(0.06)	1.45(0.01)	1.11(0.02)	1.03(0.02)

جدول ۶: مقایسه خطای آفلاین الگوریتم‌ها برای تعداد قله‌های متفاوت در مسأله MPB در فرکانس تغییرات $U = 1000$.

Algorithm	Number of peaks, p							
	1	5	10	20	30	50	100	200
mQSO(5.5q)	18.60(1.63)	6.56(0.38)	5.71(0.22)	5.85(0.15)	5.81(0.15)	5.87(0.13)	5.83(0.13)	5.54(0.11)
AmQSO	2.33(0.31)	2.90(0.32)	4.56(0.40)	5.36(0.47)	5.20(0.38)	6.06(0.14)	4.77(0.45)	5.75(0.26)
mPSO	4.44(0.02)	3.93(0.16)	4.57(0.18)	4.97(0.13)	5.15(0.12)	5.33(0.10)	5.60(0.09)	5.78(0.09)
APSO	2.72(0.04)	2.99(0.09)	3.87(0.08)	4.13(0.06)	4.12(0.04)	4.11(0.03)	4.26(0.04)	4.21(0.02)
FTMPSO	0.89(0.05)	1.70(0.10)	2.36(0.09)	3.01(0.12)	3.06(0.10)	3.29(0.10)	3.63(0.09)	3.74(0.09)
SFA	2.45(0.12)	2.71(0.06)	3.64(0.04)	4.01(0.07)	4.02(0.08)	4.12(0.07)	4.40(0.07)	4.43(0.07)
AidPSO	0.01(0.00)	1.23(0.04)	1.18(0.05)	1.03(0.06)	0.81(0.03)	0.76(0.03)	0.72(0.03)	0.75(0.05)

جدول ۷: مقایسه خطای آفلاین الگوریتم‌ها برای تعداد قله‌های متفاوت در مسأله MPB در فرکانس تغییرات $U = 10000$.

Algorithm	Number of peaks, p							
	1	5	10	20	30	50	100	200
mQSO(5.5q)	1.90(0.18)	1.03(0.06)	1.10(0.07)	1.84(0.09)	2.00(0.09)	1.99(0.07)	1.85(0.05)	1.71(0.04)
AmQSO	0.19(0.02)	0.45(0.04)	0.76(0.06)	1.28(0.12)	1.78(0.09)	1.55(0.08)	1.89(0.14)	2.52(0.10)
mPSO	0.27(0.02)	0.70(0.10)	0.97(0.04)	1.34(0.08)	1.43(0.05)	1.47(0.04)	1.50(0.03)	1.48(0.02)
APSO	0.25(0.01)	0.57(0.03)	0.82(0.02)	1.23(0.02)	1.39(0.02)	1.46(0.01)	1.38(0.01)	1.36(0.01)
FTMPSO	0.09(0.00)	0.31(0.04)	0.43(0.03)	0.56(0.01)	0.69(0.09)	0.86(0.02)	1.08(0.03)	1.13(0.04)
SFA	0.26(0.03)	0.53(0.04)	0.72(0.02)	0.91(0.03)	0.99(0.04)	1.19(0.04)	1.44(0.04)	1.52(0.03)
AidPSO	0.1(0.05)e-13	0.04(0.00)	0.03(0.00)	0.04(0.00)	0.04(0.00)	0.01(0.00)	0.52(0.02)e-2	0.58(0.03)e-2

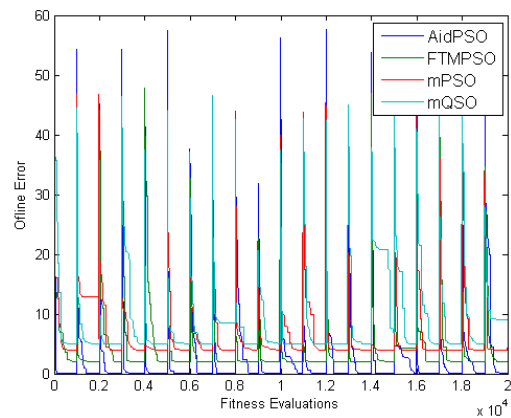
تعداد کل ذرات و اندازه نواحی محتمل که امکان یافتن بهینه در آن بیشتر است، کنترل می‌شود.

در بهینه‌سازی پویا، کاستن از زمان بهینه‌سازی حائز اهمیت است. کاهش زمان بهینه‌سازی ارتباط مستقیم با همگرایی سریع‌تر الگوریتم به بهینه سراسری دارد. در مسائل پویا این همگرایی باید با حفظ تنوع در کل فضای جستجو توأم باشد. به همین دلیل، اکثر الگوریتم‌های بهینه‌سازی در مسائل پویا با تعداد بالای ذرات شروع به کار می‌کنند و با پیشرفت روند بهینه‌سازی از تعداد ذرات کاسته می‌شود. این قضیه به معنای صرف زمان بیهوده برای محاسبه برزندگی تعداد ذرات بالا در مراحل اولیه بهینه‌سازی است. برای جلوگیری از این دست محاسبات بیهوده، الگوریتم پیشنهادی با ذرات کم شروع به کار می‌کند و همواره افزایش تعداد ذرات، تابع شرایط محیط است که عبارت‌اند از: افزایش تعداد نقاط بهینه، تغییر در شرایط محیطی و یا کاهش مقدار شاخص موفقیت که نماینده پیشرفت مطلوب الگوریتم است. در مقابل، با پایداری شدن شرایط محیط، الگوریتم اقدام به کاهش تعداد ذرات کرده و خود سبب کاهش بار محاسباتی و افزایش سرعت بهینه‌سازی می‌شود.

در محیط‌های پویا تغییرات محیطی همواره به گونه‌ای نیست که آشکارشدنی باشد. مثلاً امکان دارد فقط ناحیه‌ای از کل فضای جستجو دستخوش تغییر شود و یا تغییر در محیط‌های نویری را نمی‌توان به سادگی آشکارسازی کرد. در این صورت عملکرد الگوریتم‌های مبتنی بر آشکارسازی تغییرات محیط دچار مشکل اساسی می‌شوند. برای رفع این کاستی، الگوریتم AidPSO به گونه‌ای طراحی شده است که نیاز به آشکارسازی تغییر در محیط را ندارد و همواره خود را با شرایط محیط وفق می‌دهد.

الگوریتم پیشنهادی برای حل مسأله قله‌های متحرک (MPB) که یکی از معروف‌ترین توابع محک در محیط‌های پویا است، مورد بررسی قرار گرفته است. گستردگی آزمایش‌ها برای تنظیمات مختلف مسأله MPB اعم از تعداد قله‌های متفاوت، تغییر طول تغییر و فرکانس تغییر محیط، نشان از عملکرد مناسب الگوریتم در مقایسه با سایر الگوریتم‌های مطرح در زمینه بهینه‌سازی پویا دارد.

برای پژوهش‌های آتی، به کارگیری الگوریتم برای بهینه‌سازی مسائل واقعی دنیا، پیشنهاد می‌شود. همچنین به کارگیری الگوریتم در خوشه‌یابی^{۱۷} پویا مانند داده‌های web مدنظر است. به کارگیری مکانیزم‌های خودانطباقی برای پارامترهای ساختاری الگوریتم می‌تواند توانایی روش را برای انطباق سریع‌تر با شرایط محیطی در پی داشته باشد.



شکل ۳: مقایسه خطای آفلاین با تعداد قله ۱۰ و فرکانس ۱۰۰۰

برای نمایش سرعت اجرای الگوریتم پیشنهادی جدول ۸ آورده شده است. داده‌ها برای یک بار تغییر در محیط پنج‌بعدی و تعداد قله ۱۰ برای فرکانس ۱۰۰۰ برای الگوریتم AidPSO و ۵ الگوریتم دیگر نشان داده شده است. نتایج این جدول در کنار جدول ۶ نشان از سرعت و دقت الگوریتم در رسیدن به پاسخ بهینه‌تر است. الگوریتم با صرف زمان کمتر به مقدار خطای آفلاین بهتری نسبت سایر الگوریتم‌ها دست یافته است.

جدول ۸: مقایسه زمان اجرای الگوریتم‌ها برای مسأله MPB.

Algorithm	time, s
CPSO	79.2394
mCPSO	81.8219
mQSO(5,5q)	80.3029
CESO	81.2834
rPSO	78.3975
AidPSO	65.8952

۵- نتیجه‌گیری

امروزه الگوریتم‌های تکاملی برای حل مسائل بهینه‌سازی در محیط‌های پویا که بهینه با زمان تغییر می‌کند، بسیار به کار گرفته شده‌است. به این دلیل که مکانیزم کار این الگوریتم‌ها برگرفته از طبیعت است و شرایط طبیعی نیز همراه با زمان در حال تغییر و تحول است. در این راستا، نه تنها یافتن بهینه، بلکه دنبال کردن بهینه یافت شده حائز اهمیت است. برای دستیابی به این هدف، در بسیاری از تحقیقات، روش‌های چندجمعیتی پیشنهاد شده است.

در این مقاله الگوریتم نوینی برای بهینه‌سازی در محیط‌های غیرخطی پویا پیشنهاد شده است. الگوریتم انطباقی بهینه‌سازی ذرات افزایشی کاهشی به صورت انطباقی تعداد ذرات را با شرایط محیط برای یافتن و دنبال کردن بهینه، افزایش و یا کاهش می‌دهد. در این الگوریتم ذرات به دو گروه ذرات آزاد با وظیفه کشف نواحی محتمل و ذرات متمرکز با وظیفه یافتن بهینه در نواحی محتمل و دنبال کردن آن، تقسیم شده‌اند و با مکانیزم‌هایی

- [18] B. van Veen, M. Emmerich, Z. Yang, T. Bäck, J. Kok, "Ant colony algorithms for the dynamic vehicle routing problem with time windows", In 5th International Work-Conference on the Interplay Between Natural and Artificial Computation (IWINAC), pp. 1-10, 2013.
- [19] Z. Yang, M. Emmerich, T. Bäck, "Ant based solver for dynamic vehicle routing problem with time windows and multiple priorities", In 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 2813-2819, 2015.
- [20] L. Melo, F. Pereira, E. Costa, "Multi-caste ant colony algorithm for the dynamic traveling salesperson problem", In Adaptive and Natural Computing Algorithms, pp. 179-188, 2013.
- [21] L. Melo, F. Pereira, E. Costa, "Extended experiments with ant colony optimization with heterogeneous ants for large dynamic traveling salesperson problems", In 14th International Conference on Computational Science and Its Applications (ICCSA), pp. 171-175, 2014.
- [22] U. Boryczka, L. Stráková, "Heterogeneous DPSO algorithm for DTSP", In Computational Collective Intelligence (ICCCI), pp. 119-128, 2015.
- [23] M. Okulewicz, J. Mańdziuk, "Two-phase multi-swarm PSO and the dynamic vehicle routing problem", In IEEE Symposium on Computational Intelligence for Human-like Intelligence (CIHLI), pp. 1-8, 2014.
- [24] M. Mavrovouniotis, S. Yang, "A memetic ant colony optimization algorithm for the dynamic travelling salesman problem", Soft Computing, Vol. 15, No. 7, pp. 1405-1425, 2011.
- [25] M. Mavrovouniotis, S. Yang, "Dynamic vehicle routing: A memetic ant colony optimization approach", In Automated Scheduling and Planning, pp. 283-301, 2013.
- [26] M. Mavrovouniotis, F. M. Müller and S. Yang, "Ant colony optimization with local search for the dynamic travelling salesman problems", IEEE Transactions on Cybernetics, Vol. 99, pp. 1-14, 2016.
- [27] L. Liu and S. R. Ranjithan, "An adaptive optimization technique for dynamic environments", Engineering Applications of Artificial Intelligence, Vol. 23, No. 5, pp. 772-779, 2010.
- [28] D. Parrott, X. Li, "Locating and tracking multiple dynamic optima by a particle swarm model using speciation", IEEE Transactions on Evolutionary Computation, Vol. 10, No. 4, pp. 440-458, 2006.
- [29] S. Yang and C. Li, "A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments", IEEE Transactions on Evolutionary Computation, Vol. 14, No. 6, pp. 959-974, 2010.
- [30] C. Li and S. Yang, "A clustering particle swarm optimizer for dynamic optimization", In IEEE Congress on Evolutionary Computation, pp. 439-446, 2009.
- [31] J. Branke, "Memory enhanced evolutionary algorithms for changing optimization problems", In Proceedings of the 1999 Congress on Evolutionary Computation, Vol. 3, pp. 1875-1882, 1999.
- [32] J. Yaochu and J. Branke, "Evolutionary optimization in uncertain environments-a survey", IEEE Transactions on Evolutionary Computation, Vol. 9, pp. 303-317, 2005.
- [33] M. Khouadjia, E. Alba, L. Jourdan, E.-G. Talbi, "Multi-swarm optimization for dynamic combinatorial problems: A case study on dynamic vehicle routing problem", In Swarm Intelligence, pp. 227-238, 2010.
- [34] J. Branke, T. Kaubler, C. Schmidh, and H. Schmeck, "A multipopulation approach to dynamic optimization problem", In Proceedings of the Congress on Evolutionary Computation, pp. 299-308, 2000.
- [35] C. Li and S. Yang, "Fast multi-swarm optimization for dynamic optimization problems", In 4th International Conference on Natural Computing, Vol. 7, pp. 624-628, 2008.
- [1] M. Mavrovouniotis, Ch. Li and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithm and application", Swarm and Evolutionary Computation, Vol. 33, pp. 1-17, 2017.
- [2] 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.
- [3] M. Mavrovouniotis, S. Yang, "Adapting the pheromone evaporation rate in dynamic routing problems", In European Conference on the Applications of Evolutionary Computation, pp. 606-615, 2013.
- [4] A. Baykasoğlu, F. Ozsoydan, "An improved firefly algorithm for solving dynamic multidimensional knapsack problems", Expert Systems with Applications, Vol. 41, pp. 3712-3725, 2014.
- [5] J. Euchi, A. Yassine, H. Chabchoub, "The dynamic vehicle routing problem: Solution with hybrid metaheuristic approach", Swarm and Evolutionary Computation, Vol. 21, pp. 41-53, 2015.
- [6] Y. E. Demirtas, E. Özdemir, U. Demirtas, "A particle swarm optimization for the dynamic vehicle routing problem", In Modeling, Simulation, and Applied Optimization (ICMSAO), pp. 1-5, 2015.
- [7] M. R. Khouadjia, B. Sarasola, E. Alba, L. Jourdan, E.-G. Talbi, "A comparative study between dynamic adapted PSO and VNS for the vehicle routing problem with dynamic requests", Applied Soft Computing, Vol. 12, pp. 1426-1439, 2012.
- [8] M. Okulewicz, J. Mańdziuk, "Application of particle swarm optimization algorithm to dynamic vehicle routing problem", In Artificial Intelligence and Soft Computing (ICAISC), pp. 547-558, 2013.
- [9] M. Mavrovouniotis, S. Yang, "Applying ant colony optimization to dynamic binary-encoded problem", In Applications of Evolutionary Computation, Vol. 9028, pp. 845-856, 2015.
- [10] U. Boryczka, L. Stráková, "Diversification and entropy improvement on the DPSO algorithm for DTSP", In Intelligent Information and Database Systems (ACIIDS), pp. 337-347, 2015.
- [11] M. Mavrovouniotis, S. Yang, "Ant colony optimization with self-adaptive evaporation rate in dynamic environments", In IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE), pp. 47-54, 2014.
- [12] M. Mavrovouniotis, S. Yang, "Memory-based immigrants for ant colony optimization in changing environments", In Applications of Evolutionary Computation, Vol. 6624, pp. 324-333, 2011.
- [13] M. Mavrovouniotis, S. Yang, "Interactive and non-interactive hybrid immigrants schemes for ant algorithms in dynamic environments", In IEEE Congress on Evolutionary Computation (CEC), pp. 1542-1549, 2014.
- [14] S. Gao, Y. Wang, J. Cheng, Y. Inazumi, Z. Tang, "Ant colony optimization with clustering for solving the dynamic location routing problem", Applied Mathematics and Computation, Vol. 285, pp. 149-173, 2016.
- [15] M. Mavrovouniotis, S. Yang, "Ant colony optimization with memorybased immigrants for the dynamic vehicle routing problem", In 2012 IEEE Congress on Evolutionary Computation (CEC), pp. 2645-2652, 2012.
- [16] M. Mavrovouniotis, S. Yang, "Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors", Applied Soft Computing, Vol. 13, pp. 4023-4037, 2013.
- [17] M. Mavrovouniotis, S. Yang, X. Yao, "Multi-colony ant algorithms for the dynamic travelling salesman problem", In IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE), pp. 9-16, 2014.

زیر نویس‌ها:

- ¹ Sub-spaces
- ² Adaptive Increasing/Decreasing Particle Swarm Optimization
- ³ Particle Swarm Optimization
- ⁴ Clustering PSO (CPSO)
- ⁵ Fast Multi-Swarm Optimization (FMSO)
- ⁶ Dynamic Optimization Problems
- ⁷ Focused Search Zone
- ⁸ Adaptive Search Radius
- ⁹ Inertia weight
- ¹⁰ Success Index
- ¹¹ The Number of Focus Particles
- ¹² The Number of Free Particles
- ¹³ The Number of Optimums
- ¹⁴ Shift length
- ¹⁵ Peak location range
- ¹⁶ Offline Error (OE)
- ¹⁷ Clustering

- [36] M. Kamosi, A. B. Hashemi, and M. R. Meybodi, "A hibernating multiswarm optimization algorithm for dynamic environments", In World Congress on Nature and Biologically Inspired Computing, NaBIC2010, pp. 363–369, 2010.
- [37] T. M. Blackwell and J. Branke, "Multiswarms, exclusion, and anticonvergence in dynamic environments", IEEE Transactions on Evolutionary Computation, Vol. 10, No. 4, pp. 459–472, 2006.
- [38] I. del Amo, D. Pelta, Gonz'aez, and J. Iez, "Using heuristic rules to enhance a multiswarm pso for dynamic environments", In IEEE Congress on Evolutionary Computation, pp. 1–8, 2010.
- [39] C. Li, S. Yang, "A general framework of multipopulation methods with clustering in undetectable dynamic environments", IEEE Transactions on Evolutionary Computation, Vol. 16, No. 4, pp. 556–577, 2012.
- [40] A. Simoes and E. Costa, "Evolutionary algorithms for dynamic environments: prediction using linear regression and markov chains", In Parallel Problem Solving from Nature, pp. 306–315, 2008.
- [41] J. Kennedy and R. C. Eberhart, "Particle swarm optimization", In Proc. IEEE International Conference on Neural Networks, pp. 1942–1948, 1995.
- [42] Y. Shi, R. Eberhart, "A modified particle swarm optimizer", In IEEE World Congress on Computational Intelligence, pp. 69–73, 1998.
- [43] R. I. Lung and D. Dumitrescu, "A collaborative model for tracking optima in dynamic environments", In IEEE Congress on Evolutionary Computation, pp. 564–567, 2007.
- [44] S. Bird and X. Li, "Using regression to improve local convergence", In IEEE Congress on Evolutionary Computation, pp. 592–599, 2007.
- [45] T. Blackwell, J. Branke, and X. Li, "Particle swarms for dynamic optimization problems", Swarm Intelligence, pp. 193–217, 2008.
- [46] M. Kamosi, A. Hashemi, and M. Meybodi, "A New Particle Swarm Optimization Algorithm for Dynamic Environments", In International Conference on Swarm, Evolutionary, and Memetic Computing, pp. 129–138, 2010.
- [47] I. Rezazadeh, M. Meybodi, and A. Naebi, "Adaptive particle swarm optimization algorithm for dynamic environments", Advances in Swarm Intelligence, pp. 120–129, 2011.
- [48] 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, pp. 2144–2158, 2013.
- [49] B. Nasiri and M. Meybodi, "Speciation based firefly algorithm for optimization in dynamic environments", International Journal of Artificial Intelligence, Vol. 8, pp. 118–132, 2012.
- [50] 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, pp. 170–192, 2013.
- [51] J. K. Kordestani, A. Rezvanian, and M. R. Meybodi, "CDEPSO: a bi-population hybrid approach for dynamic optimization problems", Applied intelligence, Vol. 40, pp. 682–694, 2014.
- [52] 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.