

# بخش بندی فضای برای بهبود کارایی الگوریتم PSO در محیط های پویا

ایمان رضازاده<sup>۱</sup>، محمد رضا میدی<sup>۲</sup>، مهدی عبدالکریم وند<sup>۳</sup>

<sup>۱</sup> دانشجوی کارشناسی ارشد هوش مصنوعی، دانشگاه آزاد اسلامی قزوین

Eiman.rezazadeh@gmail.com

<sup>۲</sup> دانشکده مهندسی کامپیوتر و فن آوری اطلاعات، دانشگاه صنعتی امیر کبیر

mmeybodi@aut.ac.ir

<sup>۳</sup> دانشجوی کارشناسی ارشد هوش مصنوعی، دانشگاه امیر کبیر

abdolkarimvand@aut.ac.ir

## چکیده

بسیاری از مسائل دنیای واقعی پویا هستند بطوری که بهینه‌های محلی و سراسری در طول زمان تغییر می‌کنند. الگوریتم PSO بر روی این مسائل به منظور پیدا کردن و دنبال کردن بهینه با موفقیت مورد استفاده قرار گرفته است. در این مقاله الگوریتم PSO برای محیط‌های پویا بهبود داده شده است. الگوریتم ارائه شده فضای را به قسمت‌های مختلفی تقسیم می‌کند و در هر قسمت به طور جداگانه گروه‌هایی ایجاد می‌شود و به جستجو می‌پردازد؛ برای بهبود سرعت همگرایی میزان اینرسی ذرات به صورت پویا تنظیم می‌شود و همچنین بهترین گروه موجود برای بهبود جواب بدست آمده یک جستجوی محلی نیز انجام می‌دهد. نتایج بدست آمده بروی بنچمارک قله‌های متحرک (MPB) نشان می‌دهد که الگوریتم ارائه شده نسبت به روش‌های مشابه عملکرد بهتری دارد.

## کلمات کلیدی

PSO, Dynamic environment, MPB

## ۱- مقدمه

پیوسته می‌رسند و باید زمانبندی شوند، پارامترهایی که بر محیط‌های پویا تأثیر می‌گذارند، فرآنس تغییرات، شدت تغییرات، تغییرات قابل پیش‌بینی و طول یا دقیق چرخه می‌باشند؛ محیط‌های پویا بر اساس تنظیمات معرفی شده به چهار دسته کلی ثابت (تغییرات یکسان در هر دوره)، دوره‌ای (بازگشت به وضعیت قبلی)، همگن (کل چشم انداز به طور مرتبط حرکت می‌کنند) و تناوبی (بهینه از یک قله چشم انداز به قله دیگر جهش می‌کند) تقسیم می‌شود.<sup>[۸]</sup>

مشکلات PSO در محیط‌های پویا عبارتند از: حافظه قدیمی و کمبود تنوع که در ذیل خواهد آمد: در صورت تغییر محیط، حافظه ذره دیگر درست نیست و اثرات بسیار بدی بر جستجو خواهد گذاشت؛ این مسئله را می‌توان با استفاده از دو روش ارزیابی مجدد حافظه و فراموش کردن حافظه حل کرد. در ارزیابی مجدد، حافظه هر ذره ارزیابی می‌شود و در

PSO یک روش جستجوی هیوریستیک نسبتاً جدید است؛ این الگوریتم به طور موفقیت آمیزی در بسیاری از زمینه‌ها از جمله بازساخت الگو و پردازش تصاویر و یادگیری ماشین و غیره به کار می‌رود. PSO الگوریتم بهینه‌ای است که از روی زندگی جمعی و گروهی مثلًا پرندگان الهام می‌گیرد تا به راه حلی بهینه برسد. در الگوریتم PSO گروهی از ذرات در فضای جستجو قرار دارند که هر ذره یک راه حل کاندید برای مسئله بهینه سازی ارائه می‌کند. موقعیت هر ذره از بهترین موقعیت ملاقات شده به وسیله خودش (به معنی تجربه خودش) و موقعیت بهترین ذره در همسایگی‌هایش (به معنی تجربه‌های همسایه) اثر می‌پذیرد. محیط‌هایی که الگوریتم های تکاملی بر روی آنها اعمال می‌شوند به دو دسته کلی ایستا (بدون تغییر) و پویا (متغیر در طول زمان) تقسیم می‌شوند؛ بیشتر مسائل دنیای واقعی پویا هستند و در طول زمان تغییر می‌کنند، به عنوان مثال کارهای جدید به طور

[۹] دو الگوریتم جستجو، یکی در گروه والد و یکی در FMSO در گروه فرزند به کار می رود؛ عملگر جستجوی سراسری در گروه والد برای حفظ تنوع و پیدا کردن نواحی محتمل در فضای جستجو به کار رفته است و گروههای فرزند به عنوان یک جستجو با یک گروه والد که عملگر FMSO محلی به کار رفته‌اند. جستجو را در کل فضای جستجو انجام می‌دهد شروع می‌شود در هر مرحله اگر بهترین موقعیت یافت شده در گروه والد بهبود یابد یک گروه فرزند به مرکزیت بهترین موقعیت یافت شده و به شعاع از این نقطه ایجاد می‌گردد. و در [۱۱] توسط کاموسی بهبود داده شده است

ایده اصلی در رویکرد CLPSO [۴] بهره گیری از تعاملات محلی در اتوماتای سلوی و تقسیم جمعیت ذرات در داخل سلول‌های اتوماتای سلوی می‌باشد؛ هر گروه برای یافتن بهینه محلی تلاش می‌کند، که این کار باعث پیدا کردن بهینه سراسری می‌گردد. در روش‌های چند گروهی قبلی به منظور جلوگیری از قرار گیری بیش از یک گروه بر روی یک قله فاصله بین گروه‌ها محاسبه می‌شود تا دو گروه در داخل شعاع فعالیت یکدیگر قرار نگیرند؛ اما در این روش این هزینه محاسباتی با استفاده از اتوماتای سلوی حذف شده است.

در KPSO [۱] برای این که بتوان مسئله را به زیر مسئله‌ها تقسیم کرد، در هر چند تکرار متوالی، ذرات موجود در فضای مسئله خوش بندی می‌شوند و هر خوش به صورت جداگانه جستجوی خود را انجام می‌دهد. در این روش یکبار گروه بندی صورت می‌گیرد و برای چند تکرار ثابت می‌ماند بنابراین الگوریتم زمان کافی برای جستجو دارد. همچنین به دلیل استفاده از خوش بندی، موقعیت فضایی ذرات نیز در گروه‌های مختلف در نظر گرفته می‌شود. اما مشکل این روش تعیین تعداد مناسب خوش‌ها می‌باشد.

در [۱۰] لی و یانگ با استفاده از خوش بندی فازی ذرات را گروه‌بندی کرده‌اند، سپس با استفاده از نتایج خوش بندی ذرات را خوش بندی کرده‌اند و در ادامه با توجه به پیشرفت الگوریتم خوش‌ها را باهم ترکیب می‌کنند و یا اگر تعداد ذرات دو خوش کم و نزدیک هم باشند این دو خوش بهام ترکیب می‌شوند.

### ۳- روش پیشنهادی

الگوریتم پیشنهادی ابتدا فضا را با استفاده از خوش بندی فازی به قسمت‌های مختلف تقسیم می‌کند، سپس در هر یک از پارتیشن‌ها گروه‌ها مستقل‌ايجاد می‌شوند و به جستجو می‌پردازنند.

<sup>3</sup> Cellular PSO

<sup>4</sup> K-mins based PSO

فراموش کردن حافظه، موقعیت فعلی ذره بجای حافظه‌ی آن جایگزین می‌شود و بهینه سراسری به روز می‌شود. کمبود تنوع هنگامی رخ می‌دهد که گروه به یک پیک همگرا شود و محیط تغییر کند. برای حل این مشکل یک مکانیزم افزایش تنوع در هنگام تغییر، استفاده می‌شود تا تنوع کافی در نامامی زمان‌ها وجود داشته باشد. روش‌های متنوع سازی اصلی معرفی شده شامل: تصادفی کردن، دافعه، شبکه‌های دینامیکی و چند جمعیتی می‌باشد.

این مقاله شامل بخش‌هایی به این ترتیب می‌باشد: در بخش دوم مروری بر کارهای انجام شده بر روی محیط‌های پویا که در مقالات آمده، صورت گرفته است؛ در بخش سوم الگوریتم ارائه شده توصیف گردیده؛ در بخش چهارم بنچمارک مورد استفاده معرفی گردیده؛ سپس در بخش پنجم تنظیمات الگوریتم‌ها برای آزمایشات آورده شده و در بخش ششم نتایج آزمایشات بروی بنچمارک قله‌های متحرک نمایش داده شده و در آخر نتیجه‌گیری از آزمایشات صورت گرفته ارائه شده است.

### ۲- کارهای قبلی

MPSO ابتدا توسط بلک ول<sup>۱</sup> و برانک<sup>۲</sup> پیشنهاد [۵,۶] و سپس توسعه داده شد. ذرات در MPSO به  $m$  زیر گروه مستقل تقسیم می‌شوند. هر زیر گروه دارای تعداد ثابتی از ذرات است. اشتراک اطلاعات درون هر گروه به صورت سراسری است. این روش، تنوع را در دو سطح حفظ می‌نماید: گروه به تعدادی زیر گروه تقسیم می‌شود که به قسمت‌های مختلف فضای جستجو نفوذ می‌کنند (تنوع بین گروه‌ها). و هر گروه دارای تعدادی ذرات کوانتومی است که تنوع درون گروه را ایجاد می‌نمایند.

SPSO [۶] ذرات را به صورت پویا بین گونه‌ها توزیع می‌کند. SPSO بر پایه نظریه گونه‌ها توسعه یافته است، محدوده گونه‌ها وابسته به یک پارامتر  $s$  است که شعاع اندازه گیری شده در فاصله اقلیدسی را از مرکز گونه تا مرزهای آن مخصوص می‌کند. مرکز یک گونه که «هسته گونه» نامیده می‌شود، معمولاً ذرهای با بهترین شایستگی در گونه است. تمامی ذراتی که درون فاصله  $r$  از هسته گونه قرار می‌گیرند، به عنوان یک گونه مشابه طبقه بندی می‌شوند.

UPSO [۷] الگوریتمی است که متغیرهای سراسری و محلی PSO را کنترل کرده و قابلیت‌های کاوش و استخراج آنها را بدون تحمیل الزامات اضافی بر حسب ارزیابی‌های تابع، ترکیب و متحدد می‌سازد.

<sup>1</sup> Blackwell

<sup>2</sup> Branke

از آنجا که جستجوی یک ناحیه با بیش از یک گروه مفید نمی‌باشد، در پایان هر مرحله فاصله بین هر دو گروه مورد بررسی قرار می‌گیرد که در داخل شاعع یکدیگر قرار دارند، یا نه در صورتی که دو گروه در داخل شاعع یکدیگر قرار داشته باشند، یعنی فاصله بین بهترین موقعیت یافت شده آن‌ها از آستانه تعیین شده (شعاع دفع مابین گروه‌ها) کمتر باشد آن گروه که  $gbest$  آن دارای برازندگی کمتری است از فضای جستجو خارج می‌گردد، این کار سبب می‌شود که بیش از یک گروه بر روی یک پیک قرار نگیرد.

در ابتدای هر تکرار تعداد گروه‌های آزاد هر پارتیشن محاسبه می‌شود، اگر این تعداد از یک حد آستانه‌ای بیشتر باشد بدترین گروه پارتیشن از این فضای جستجو حذف می‌شود و در صورتی که گروه آزادی وجود نداشته باشد یک گروه جدید با موقعیت‌های تصادفی در داخل این پارتیشن به گروه‌های موجود اضافه می‌شود. زمانی که تغییری در محیط شناسایی شود حافظه ذرات ارزیابی می‌شود و سپس موقعیت هر ذره در گروه برابر یک موقعیت تصادفی در داخل یک ابر کره به مرکزیت بهترین جدید ذرات ارزیابی و بهترین تجربه شخصی و گروهی بروز رسانی می‌شود، این الگوریتم در زیر آورده شده است:

تفاوت این روش با روش‌های دیگری که از خوش بندی فازی برای بهبود الگوریتم PSO استفاده کرداند در این است که آنها از خوش بندی فازی برای دسته بندی ذرات استفاده کرداند، اما در روش پیشنهادی از خوش بندی فازی فقط برای پارتیشن بندی فضای مسئله استفاده شده است.

```

Procedure Proposed Algorithm
begin
clustering space
Initialization swarms in each partition
Repeat
    for each partition do
        if free-swarms > threshold
            delete worse swarm
        elseif free-swarms = 0
            add a swarm
        end-if
        for each swarm in partition do
            if a change is detected in the environment then
                reset partition immigration _number
                for each particle in swarm do
                    evaluate pbesti
                    pi= a random position in a hypersphere
                    with radius r centered at gbest
                    update pbest
                end-for
                update gbest
            else
                do local search around gbest and update gbest
            end-if
        end-for
    end-if
end-while

```

گروه‌ها از ذرات PSO برای جستجو استفاده می‌کنند و از آنجاکه همسایگی‌های کوچک به دلیل کاهش سرعت همگرایی و افزایش تنوع در محیط‌های پیچیده بهتر عمل می‌کنند، در این الگوریتم از همسایگی‌های کوچک استفاده شده است از طرفی به ازای تعداد گروه‌های ثابت تعداد ارزیابی‌ها کاهش می‌یابد که خود باعث می‌شود تا فضا تعداد تکرارهای بیشتری ثابت بماند و جستجوی کار آمدتری در فضا صورت بگیرد. برای بهبود سرعت همگرایی الگوریتم سعی شده تا مقدار ضریب اینرسی حرکتی ( $W$ ) به صورت پویا تنظیم شود؛ این کار با استفاده از روش مبتنی بر رتبه صورت گرفته است، به طوری که ذراتی که برازش بدتری دارند باستی در تکرار بعدی سرعت لحظه قبل تاثیر کمتری در محاسبه سرعت داشته باشند. گروه‌های یک خوش ممکن است پس از مدتی از یک خوش خارج شده و به خوش همچوار بروند در این صورت برازندگی این گروه با مقدار برازندگی بدترین گروه خوش مقصد مقایسه می‌شود، در صورتی که برازش بدترین گروه خوش مقصد از برازش گروه مهاجر بدتر باشد این گروه به جای بدترین گروه خوش مقصد کپی می‌شود در غیر این صورت گروه مهاجر حذف می‌شود. همچنان اگر تعداد مهاجرت‌ها از یک خوش زیاد باشد نشان دهنده آنست که در این خوش قله‌ای وجود ندارد، بنابراین جستجو در این خوش مفید نمی‌باشد و فقط باعث هدر رفتن تعداد ارزیابی‌ها می‌شود بنابراین این خوش به عنوان خوش خاموش در نظر گرفته می‌شود و دیگر در روند جستجو شرکت نمی‌شود، ممکن است برای تعیین خوش خاموش استفاده از میزان مهاجرت به تنها یک مناسب نباشد، به همین دلیل از ترکیب میزان برازش پارتیشن و میزان مهاجرت استفاده شده است. همچنان برای بهبود جواب بدست آمده یک جستجوی محلی در اطراف بهترین ذره بهترین گروه صورت می‌گیرد. الگوریتم روش پیشنهادی بصورت زیر می‌باشد.

ابتدا تعدادی نقاط تصادفی در فضا ایجاد می‌شود و فضا با خوش بندی نقاط ایجاد شده تقسیم بندی می‌شود، سپس در هر پارتیشن گروه‌هایی به طور تصادفی ایجاد می‌شود، بهترین تجربه هر ذره با موقعیت فعلی ذره مقدار دهی می‌شود و سپس در هر مرحله موقعیت هر ذره با استفاده از معادلات (۱) و (۲) به روز رسانی می‌شود و مقدار اینرسی نیز با استفاده از رتبه برازش هر ذره در گروه با استفاده از معادله (۳) بروز رسانی می‌شود.

$$= wv_i(t) + c_1r_1(pbest_i - p_i) + c_2r_2(gbest_i - p_i) \quad (1)$$

$$p_i(t+1) = p_i(t) + v_i(t+1) \quad (2)$$

$$w_i = w_{min} + (w_{max} - w_{min}) \frac{Rank_i}{Total\_population} \quad (3)$$

## ۵- تنظیمات اولیه برای آزمایشات

برای الگوریتم پیشنهادی ثابت شتاب  $c_1 = 0.496180$  و  $c_2 = 0.073000$  در  $w = [0.073000, 0.8758]$  تعیین شده است. تعداد پارتبیشن‌ها برابر ۳ انتخاب شده و تعداد گروه‌های موجود در هر پارتبیشن در لحظه اول برابر ۱ می‌باشد، تعداد ذرات هر گروه برابر ۳ ذره تعیین شده است. همچنین شعاع دفع ما بین گروه و شعاع ذرات کوانتوم به ترتیب برابر  $31.5$  و  $5$  تعیین شده و الگوریتم Cellular PSO، FMSO، MQSO پیشنهادی با الگوریتم‌های Kamosi و MQSO مقایسه شده است. در اینجا برای Kamosi پیکربندی  $(q_5+5) \times 10$  استفاده شده که در آن ده گروه ایجاد می‌شود و هر یک از گروه‌ها دارای ۵ ذره خنثی و دارای ۵ ذره کوانتوم می‌باشند؛ همچنین برای این الگوریتم شعاع کوانتوم برابر  $0.5$  و شعاع دفع و شعاع همگرایی برابر  $31.5$  تعیین شده است. برای الگوریتم FMSO ماکریتم تعداد گروه‌های فرزند برابر  $10$  و شعاع دفع مابین گروه‌های فرزند برابر  $25$  و تعداد ذرات در گروه والد و گروه‌های فرزند به ترتیب برابر  $100$  و  $10$  در نظر گرفته شده است. برای PSO Cellular یک اتوماتی سلولی  $5$  بعدی با  $10^5$  سلول و همسایگی مور با شعاع دو سلول در فضای جستجو به کار رفته است و ماکریتم سرعت ذرات برای شعاع همسایگی و ماکریتم تعداد ذرات برای هر سلول برابر  $10$  و شعاع جستجوی محلی را محلی برابر  $5$  تعیین شده است، همچنین جستجوی محلی را همه‌ی ذرات پس از مشاهده تغییر در محیط برای یک مرحله اجرا می‌کنند. برای روش Kamosi تعداد ذرات برای گروه والد و تعداد ذرات برای گروه‌های فرزند به ترتیب برابر  $50$  و  $10$  تعیین شده است، شعاع دفع ما بین گروه‌های فرزند و شعاع ذرات کوانتوم به ترتیب برابر  $30$  و  $5$  تعیین شده است.

جدول ۱ تنظیمات استاندارد برای تابع قله‌های متحرک

مقادیر	پارامتر
۱۰	تعداد قله‌ها
۵۰۰۰	فرکانس تغییرات
۷	شدت تغییرات ارتفاع قله‌ها
۱	شدت تغییرات عرض قله‌ها
Cone	شکل قله‌ها
۱	میزان حرکت مکان قله‌ها
۵	ابعاد فضای جستجو
[۳۰۰۰، ۷۰۰]	محدوده ارتفاع قله‌ها
[۱۰۰۰، ۱۲۰۰]	محدوده عرض قله‌ها
۵۰	ارتفاع استاندارد قله‌ها
[۰، ۱۰۰]	محدوده‌ی فضای جستجو

```

for each particle in swarm do
    update particle position according to eq. 1
and eq. 2 and eq. 3.
    update pbest and gbest
end-for
end-if
end-for
for each swarm in partitions do
    if swarm partition changed from m to n
        if swarm is better than worst swarm of partition
            remove worst swarm
            move swarm to partition n
    else
        delete swarm
    end-if
    update partition m and n immigration _number
end-if
end-for
for each pair of swarms m and n, m ≠ n do
    if distance between gbestm and gbestn < rexcl
        if f(gbestm) > f(gbestn)
            Delete swarm n
        else
            Delete swarm m
        end-if
    end-if
end-for
until a maximum number of fitness evaluations is
reached
end.

```

## ۴- تابع آزمایشی مورد استفاده برای سنجش

### کارایی الگوریتم‌های پیشنهادی

برانک در [۲] یک مسأله پویا را برای آزمایش الگوریتم‌های بهینه‌سازی در محیط‌های پویا پیشنهاد داده است که تابع قله‌های متحرک<sup>۱</sup> نامیده می‌شود و معمولاً به عنوان معیاری برای سنجش الگوریتم‌های بهینه‌سازی در محیط‌های پویا مورد استفاده قرار می‌گیرد. این تابع شامل یک محیط چند بعدی است که دارای چندین قله می‌باشد، هنگامی که یک تغییر در محیط رخ دهد ارتفاع عرض و محل قله‌ها تغییر می‌نماید؛ از این تابع پویا برای تست کارایی الگوریتم پیشنهادی استفاده شده است؛ این تابع این قابلیت را دارد که ابعاد محیط، تعداد قله‌ها، شکل قله‌ها، میزان شدت تغییرات و فرکانس تغییرات را تنظیم کند. جزئیات بیشتر در رابطه با تابع MPB در وب سایت: <http://www.aifb.unikarlsruhe.de/~jbr/MovPeaks/>. یافت می‌شود.

تنظیمات پیش فرض برای آزمایشات در **Error! Reference** source not found.

<sup>۱</sup> Moving peaks benchmark

## ۶- نتایج آزمایشات

نتایج آزمایشات برای همهی الگوریتم‌ها میانگین خطای آفلاین با فاصله اطمینان ۹۵ درصد در ۱۰۰ مرتبه اجرا می‌باشد، در اینجا الگوریتم پیشنهادی با الگوریتم‌های Cellular، FMSO، MQSO و PSO مقایسه شده است. خطای آفلاین بدست آمده از آزمایشات برای محیط‌های با پویایی‌های متفاوت در جدول‌های ۲ تا ۶ نشان داده شده است. نتایج بهتر به صورت بولد می‌باشد، همان طور که مشاهده می‌شود الگوریتم پیشنهادی در فضای با تعداد قله‌های بیش از ۵ قله نتایج بهتری را نسبت به الگوریتم‌های MQSO و Cellular PSO دارد. FMSO با اختلاف کمی بهتر می‌باشد و در فضای ۵۰۰۰ روشن Kamosi با فرکانس ۲۵۰۰ و ۵۰۰۰ روشن Kamosi با اختلاف کمی بهتر می‌باشد و در فضای ۵ قله‌ای در فرکانس ۲۵۰۰ و ۵۰۰۰ روشن Kamosi با اختلاف کمی بهتر می‌باشد. اخلاف خطای آفلاین موارد روشن پیشنهادی بهتر می‌باشد، اخلاقیت خطای آفلاین الگوریتم پیشنهادی نسبت به الگوریتم دیگر با افزایش فرکانس محیط و همچنین با افزایش پیچیدگی فضا (افزایش تعداد قله‌ها) افزایش پیدا می‌کند، علت این است که الگوریتم پیشنهادی سریع‌تر می‌تواند راه حل‌های بهتر را پس از مشاهده تغییر در محیط بدست بیاورد. علاوه بر این تعداد گروه‌ها در الگوریتم پیشنهادی به علت این که پارتیشن‌ها مستقل از هم عمل می‌کنند و سرعت رشد گروه‌ها برابر با تعداد پارتیشن‌ها می‌باشد، تعداد گروه‌ها سریع‌تر به تعداد قله‌های موجود در محیط همگرا می‌شود، این مسئله سبب می‌شود که با توجه به وجود یک گروه بر روی هر قله الگوریتم پیشنهادی بهتر بتواند قله‌های متحرك را رد گیری نماید، از طرف دیگر استفاده از روش مبتنی بر رتبه برای تنظیم مقدار  $W$  باعث می‌شود تا ذراتی که مقدار برازش کمتری دارند تاثیر کمتری از سرعت قبلی بگیرند و تبعیت بیشتری از تجربه قبلی خود و گروه داشته باشند و همچنین ذراتی که برازش بیشتری دارند از سرعت قبلی خود بیشتر تبعیت می‌کنند، که این موضوع باعث می‌شود تا گروه‌ها با سرعت بیشتری به سمت بهینه حرکت کنند.

در شکل‌های ۱ تا ۴ نمودار خطای جاری و خطای آفلاین الگوریتم پیشنهادی نشان داده شده است؛ برای این که این تغییرات بهتر نشان داده شوند از ۱۰۰ مرتبه تغییر در محیط برای خاتمه اجرای الگوریتم نمودار خطای جاری و خطای آفلاین در ۲۰ مرحله اول تغییرات نشان داده شده است.

جدول ۲ نتایج بدست آمده به ازای فرکانس ۵۰۰ برای قله‌های مختلف

P	Proposed algorithm	Kamosi	mQSO10(5+5 <sup>q</sup> )	FMSO	Cellular PSO
---	--------------------	--------	---------------------------	------	--------------

1	5.69±1.94	5.46±0.30	33.67±3.42	27.58±0.94	13.46±0.7
5	5.33±1.08	5.48±0.19	11.91±0.76	19.45±0.45	9.63±0.49
10	5.81±1.01	5.95±0.09	9.62±0.34	18.26±0.32	9.42±0.21
20	5.95±0.74	6.45±0.16	9.07±0.25	17.34±0.30	8.84±0.28
30	5.88±0.58	6.60±0.14	8.80±0.21	16.39±0.48	8.81±0.24
40	6.16±0.61	6.85±0.13	8.55±0.21	15.34±0.45	8.94±0.24
50	6.09±0.71	7.04±0.10	8.72±0.20	15.54±0.26	8.62±0.23
100	6.36±0.55	7.39±0.13	8.54±0.16	12.87±0.60	8.54±0.21
200	6.29±0.53	7.52±0.12	8.19±0.17	11.52±0.61	8.28±0.18

جدول ۳ نتایج بدست آمده برای قله‌های مختلف در

فرکانس ۱۰۰۰

P	Proposed algorithm	Kamosi	mQSO10(5+5 <sup>q</sup> )	FMSO	Cellular PSO
1	2.80±1.00	2.90±0.18	18.60±1.63	14.42±0.48	6.77±0.38
5	3.04±0.64	3.35±0.18	6.56±0.38	10.59±0.24	5.30±0.32
10	3.57±0.51	3.94±0.08	5.71±0.22	10.40±0.17	5.15±0.13
20	1.12±0.67	4.33±0.12	5.85±0.15	10.33±0.13	5.23±0.18
30	4.19±0.62	4.41±0.11	5.81±0.15	10.06±0.14	5.33±0.16
40	4.13±0.38	4.52±0.09	5.70±0.14	9.85±0.11	5.61±0.16
50	4.23±0.49	4.57±0.08	5.87±0.13	9.54±0.11	5.55±0.14
100	4.12±0.35	4.77±0.08	5.83±0.13	8.77±0.09	5.57±0.12
200	4.06±0.33	4.76±0.07	5.54±0.11	8.06±0.07	5.50±0.12

جدول ۴ نتایج بدست آمده برای قله‌های مختلف در

فرکانس ۲۵۰۰

m	Proposed algorithm	Kamosi	mQSO10(5+5 <sup>q</sup> )	FMSO	Cellular PSO
1	1.29±0.50	1.10±0.06	7.64±0.64	6.29±0.20	4.15±0.25
5	1.69±0.62	1.68±0.16	3.26±0.21	5.03±0.12	2.85±0.24
10	2.12±0.46	2.33±0.06	3.12±0.14	5.09±0.09	2.80±0.10
20	2.51±0.51	2.79±0.10	3.58±0.13	5.32±0.08	3.41±0.14
30	2.58±0.41	2.88±0.09	3.63±0.10	5.22±0.08	3.62±0.12
40	2.50±0.29	2.86±0.07	3.55±0.10	5.09±0.06	3.84±0.12
50	2.66±0.28	2.97±0.06	3.63±0.10	4.99±0.06	3.86±0.10
100	2.54±0.24	3.00±0.05	3.58±0.08	4.60±0.05	4.10±0.11
200	2.45±0.16	2.99±0.04	3.30±0.06	4.34±0.04	3.97±0.10

جدول ۵ نتایج بدست آمده برای قله‌های مختلف در

فرکانس ۵۰۰۰

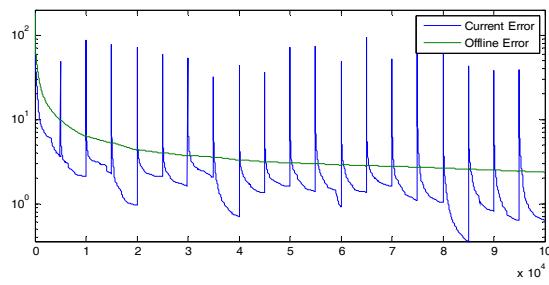
m	Proposed algorithm	Kamosi	mQSO10(5+5 <sup>q</sup> )	FMSO	Cellular PSO
1	0.58±0.22	0.56±0.04	3.82±0.35	3.44±0.11	2.55±0.12
5	1.12±0.62	1.06±0.06	1.90±0.08	2.94±0.07	1.68±0.11
10	1.31±0.29	1.51±0.04	1.91±0.08	3.11±0.06	1.78±0.05
20	1.67±0.33	1.89±0.04	2.56±0.10	3.36±0.06	2.61±0.07
30	1.82±0.33	2.03±0.06	2.68±0.10	3.28±0.05	2.93±0.08
40	1.87±0.28	2.04±0.06	2.65±0.08	3.26±0.04	3.14±0.08
50	1.92±0.22	2.08±0.02	2.63±0.08	3.22±0.05	3.26±0.08
100	1.85±0.17	2.14±0.02	2.52±0.06	3.06±0.04	3.41±0.07
200	1.73±0.11	2.11±0.03	2.36±0.05	2.84±0.03	3.40±0.06

جدول ۶ خطای آفلاین برای قله‌های مختلف در فرکانس ۱۰۰۰۰

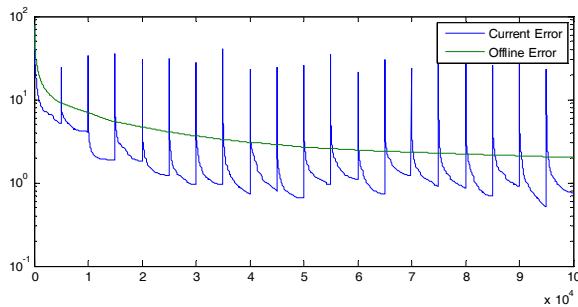
m	Proposed algorithm	Kamosi	mQSO10(5+5 <sup>q</sup> )	FMSO	Cellular PSO
1	0.32±0.10	0.27±0.02	1.90±0.18	1.90±0.06	1.53±0.12
5	0.62±0.38	0.70±0.10	1.03±0.06	1.75±0.06	0.92±0.10
10	0.85±0.36	0.97±0.04	1.10±0.07	1.91±0.04	1.19±0.07
20	1.20±0.28	1.34±0.08	1.84±0.08	2.16±0.04	2.20±0.10
30	1.28±0.26	1.43±0.05	2.00±0.09	2.18±0.04	2.60±0.13
40	1.30±0.19	1.47±0.06	1.99±0.07	2.21±0.03	2.73±0.11
50	1.32±0.18	1.47±0.04	1.99±0.07	2.60±0.08	2.84±0.12
100	1.30±0.14	1.50±0.03	1.85±0.05	2.20±0.03	2.93±0.09
200	1.23±0.11	1.48±0.02	1.71±0.04	2.00±0.02	2.88±0.07

جدول ۷ نتایج بدست آمده به ازای فرکانس ۵۰۰ برای قله‌های مختلف

P	Proposed algorithm	Kamosi	mQSO10(5+5 <sup>q</sup> )	FMSO	Cellular PSO
---	--------------------	--------	---------------------------	------	--------------



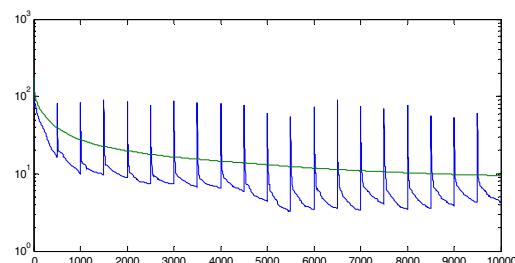
شکل ۳ نمودار خطای آفلاین و خطای جاری الگوریتم پیشنهادی در محیط پویایی با  $10^4$  قله و فرکانس  $5000$



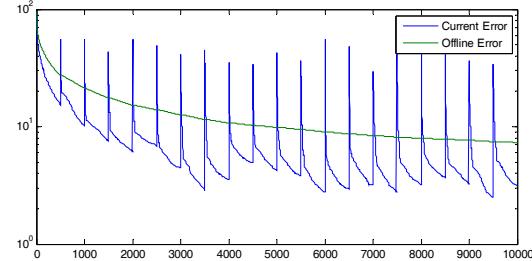
شکل ۴ نمودار خطای آفلاین و خطای جاری الگوریتم پیشنهادی در محیط پویایی با  $50$  قله و فرکانس  $5000$

## منابع

- [1] Alessandro Passaro and Antonina Starita, "Particle Swarm Optimization for Multimodal Functions: a Clustering Approach," Journal of Artificial Evolution and Applications, vol 2008, article id 482032, 2008.
- [2] J. Branke, "Memory enhanced evolutionary algorithms for changing optimization problems," Congress on Evolutionary Computation CEC99, vol 3, pp. 1875-1882, 1999.
- [3] T. Blackwell, "Particle swarm optimization in dynamic environments," in Evolutionary Computation in Dynamic and Uncertain Environments, Studies in Computational Intelligence, vol. 51, pp. 29–49, 2007.
- [4] Ali B. Hashemi and M. R. Meybodi, "Cellular PSO: A PSO for Dynamic Environments," Advances in Computation and Intelligence, vol 5821, pp. 422-433, 2009.
- [5] T. Blackwell, "Particle swarm optimization in dynamic environments," Evolutionary Computation in Dynamic and Uncertain Environments, Springer, Berlin, Germany, 2007.
- [6] T. Blackwell, J. Branke and X. Li, "Particle swarms for dynamic optimization problems," Swarm Intelligence: Introduction and Applications, Berlin, Germany, 2008.
- [7] K. E. Parsopoulos and M. N. Vrahatis, "Unified particle swarm optimization in dynamic environments," Applications on Evolutionary Computing, vol 3449, pp.590-599, 2005.
- [8] J. Branke, "Evolutionary optimization in dynamic environments," <http://www.amazon.com/Evolutionary-Optimization-Environments-Algorithms-Computation/dp/0792376315>.



شکل ۱ نمودار خطای آفلاین و خطای جاری الگوریتم پیشنهادی در محیط پویایی با  $10^4$  قله و فرکانس  $500$



شکل ۲ نمودار خطای آفلاین و خطای جاری الگوریتم پیشنهادی در محیط پویایی با  $50$  قله و فرکانس  $500$

## ۷- نتیجه گیری

در این مقاله روش جدیدی برای بهبود الگوریتم PSO به منظور بهبود کارآیی آن در محیط‌های پویا ارائه گردید. الگوریتم ارائه شده فضا را با استفاده از خوش بندی به قسمت‌های مختلف تقسیم می‌کند و در هر قسمت گروه‌ها به طور مستقل از بخش دیگر به جستجو می‌پردازند، برای بهبود جواب‌های بدست آمده توسط بهترین گروه، اطراف بهترین ذره این گروه یک جستجوی محلی صورت می‌گیرد و همچنین مقدار اینرسی به صورت پویا تنظیم می‌شود و به منظور جلوگیری از جستجوی یک فضا با بیش از یک گروه اگر دو گروه باهم همپوشانی داشته باشند، گروه بدتر از فضای جستجو حذف می‌شود، نتایج بدست آمده نشان می‌دهد که روش پیشنهادی از روش‌های دیگر بهتر عمل می‌کند و با افزایش پیچیدگی فضا کارایی الگوریتم ارائه شده افزایش می‌یابد.

- [9] Li, C., Yang, S.: Fast Multi-Swarm Optimization for Dynamic Optimization Problems. In: Fourth International Conference on Natural Computation, Jinan, Shandong, China, vol. 7, pp.624–628, 2008.
  - [10] Changhe Li and Shengxiang Yang, "A Clustering Particle Swarm Optimizer for Dynamic Optimization", IEEE, 978-1-4244-2959-2/09/\$25.00\_c 2009.
- [ ] مسعود کامویی، " بهبود بهینه سازی گروه ذرات در محیط های پویا" پایان نامه کارشناسی ارشد، دانشگاه آزاد واحد علوم و تحقیقات ، تهران، تابستان ۸۹