بهینه‌سازی دستهٔ ذرات با چندین مکانیزم تطبیقی برای بهینه سازی در محیط های پویا

بابک نصیری[۱] ؛ دانیال یزدانی[۲]؛ محمد رضا میبدی[۳]

چکیده

در بسیاری از مسائل بهینه‌سازی تابع هدف، محدودیت‌ها یا متغیرهای طراحی ممکن است در طول زمان تغییر یابند و به طبع آن بهینهٔ این مسائل نیز در طول زمان تغییر می کند. اگر هر یک از این اتفاقات غیر قابل پیشبینی در فرآیند بهینه‌سازی بوجود آید، این مسأله دینامیک یا پویا نامیده می‌شود. بسیاری از مسائل در دنیای واقعی پویا، غیرقطعی و پیچیده می‌باشند و حل آنها بصورت ایستا، به حل مسئله در دنیای واقعی کمکی نمی‌کند. در این گونه مسائل علاوه بر پیدا کردن بهینهٔ سراسری می‌بایست آن را در طول زمان دنبال نمود. در این مقاله یک الگوریتم نوین برای بهینه‌سازی در محیط پویا مبتنی بر الگوریتم بهینه‌سازی دستهٔ ذرات پیشنهاد شده است. نتایج حاصل از رهیافت پیشنهادی بر روی معیار قله‌های متحرک که در حال حاضر شناخته-شده‌ترین معیار برای ارزیابی در محیط‌های پویا می‌باشد ارزیابی شده و با نتایج حاصل از چندین الگوریتم معتبر مورد مقایسه قرار گرفته است. نتایج بدست آمده نشان‌دهندهٔ کارایی بالای الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌ها می‌باشد.

کلمات کلیدی

بهینه‌سازی، محیط‌های پویا، الگوریتم بهینه سازی دستهٔ ذرات، معیار حرکت قله‌ها.

Particle swarm optimization with multiple adaptive mechanisms for optimization in dynamic environments

Babak Nasiri; Danial Yazdani; MohammadReza Meybodi

Abstract

In many optimization problems in real world, objective function or constraints of problem can be change during time, so optima value of these problems also can be change. If each of these undefined events is considered in optimization, this problem is called dynamic. With respect to dynamic environments conditions, algorithms which were designed for optimization in these environments have some principles that distinguish them from algorithms designed in static environment. In this paper, a new algorithm for optimization in dynamic environments is proposed based on particle swarm optimization. It uses multiple mechanisms for adaptation such as adaptation of quantum cloud radios around optimum peak, self-adaptation of active swarm number and increasing local search ability around best found peak by adaptation of quantum location. The results of the proposed approach were evaluated on moving peak benchmarks, which currently are the best known criterion for evaluation in dynamic environments, and were compared with results of multiple valid algorithms. Results show high efficiency of proposed algorithm in comparison with other algorithms.

[۱] دانشگاه آزاد اسلامی قزوین، دانشکده مهندسی برق، کامپیوتر و فناوری اطلاعات، nasiri_babak@yahoo.com

[۲] دانشگاه آزاد اسلامی شیروان، دانشکده مهندسی کامپیوتر، danial.yazdani@yahoo.com

[۳] دانشگاه صنعتی امیرکبیر، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، mmeybodi@aut.ac.ir

Keywords
Optimization, Dynamic Environments, Particle Swarm Optimization Algorithm, Moving Peaks Benchmark.

۱ -مقدمه

عدم قطعیت در بسیاری از مسائل در دنیای واقعی کاملا واضح و مشهود است. یکی از روشهای برخورد با عدم قطعیت استفاده از روشهای تکاملی و هوش جمعی می‌باشد. مسائل غیرقطعی که تاکنون توسط روش‌های تکاملی مورد بررسی قرار گرفته‌اند را بطور کلی می‌توان به چهار دسته تقسیم نمود. وجود نویز در تابع ارزیاب، آشفتگی در متغیرهای طراحی، تقریبی بودن تابع ارزیاب و پویا بودن راه‌حل بهینه. در این مقاله عدم قطعیت از نوع پویا بودن راه‌حل بهینه که جزو عمومی‌ترین انواع عدم قطعیت می‌باشد، مورد توجه قرار گرفته است.

تاکنون روشهای مختلفی برای حل مسائل پویا با استفاده از روش‌های پردازش تکاملی [۱] و هوش جمعی[۲][۳] ارائه شده است. دو مشکل عمده روش های پردازش تکاملی که باعث عدم توانایی استفاده مستقیم از این روش‌ها برای بهینه‌سازی در محیط‌های پویا شده است، حافظه غیرمعتبر و از دست رفتن تنوع می‌باشد. هنگامی که محیط تغییر می‌کند، راه‌حل‌های بدست آمده موجود در حافظه دیگر معتبر نمی‌باشند لذا می بایست آنها را بطور کامل فراموش نمود یا دوباره ارزیابی کرد. همچنین از آنجا که اکثر روش‌های پردازش تکاملی و هوش جمعی بدلیل ماهیتشان به یک نقطه همگرا می‌شوند لذا تنوع در جمعیتشان در محیط از بین می‌رود و در صورت تغییر در محیط، همگرا شدن به نقطه بهینه جدید در صورت امکان، بسیار زمانگیر خواهد بود. روشهای مختلفی برای ایجاد یا نگهداری تنوع دسته در محیط وجود دارد. ساده‌ترین راه برای واکنش در برابر تغییر محیط، در نظر گرفتن هر تغییر به عنوان ورود یک مسأله بهینه‌سازی جدید و حل آن از ابتدا می باشد. در صورت داشتن زمان کافی، این یک گزینه مناسب می‌باشد اما در بیشتر مواقع زمان کافی برای بهینه-سازی مجدد وجود ندارد. یک روش مناسب برای تسریع فرآیند بهینه‌سازی پس از هر تغییر، استفاده از اطلاعات مربوط به جستجوی قبلی برای تسریع در امر جستجو پس از تغییر می‌باشد. معمولاً در اکثر مسائل تغییرات عمده نمی باشند و تغییر در محیط تنها باعث جابجایی کوچکی در بهینه جدید خواهد بود. در این مواقع روش اشاره شده بسیار مناسب می باشد. میزان اهمیت استفاده از اطلاعات گذشته بمنظور تسریه امر جستجو با میزان تغییر در محیط نسبت معکوس دارد. هر چه تغییر بیشتر باشد می بایست از اطلاعات گذشته کمتر استفاده شود تا فرایند بهینه سازی به مسیری گمراه کننده نرود.

اگر تغییر به صورت گسترده باشد و فضای مسئله پس از تغییر دارای شباهت کمی با قبل باشد، شروع مجدد می‌تواند تنها گزینه مناسب باشد و هرگونه استفاده مجدد از اطلاعات جمع‌آوری شده بر مبنای مسأله قبلی گمراه‌کننده خواهد بود. در اکثر مسائل دنیای واقعی، امید می‌رود که تغییرات نسبتاً هموار باشند یعنی بین محیط مسئله پیش از تغییر و پس از تغییر ارتباطی وجود داشته باشد تا بتوان از اطلاعات قبلی برای تسریع فرآیند بهینه سازی استفاده نمود.

سؤال اساسی که در این جا مطرح می‌شود این است که چه اطلاعاتی باید نگهداری شود و این اطلاعات چگونه باید برای تسریع عمل جستجو پس از تغییر محیط استفاده شوند. مسئله دیگر این است که در بیشتر الگوریتم های بهینه سازی، پس از اینکه الگوریتم به یک نقطه همگرا شد، تنوع خود را از دست می دهد که این امر باعث کاهش انطباق پذیری و سازگاری الگوریتم در مقابل تغییر در محیط خواهد بود، بنابراین در کنار انتقال اطلاعات بین عامل های الگوریتم های بهینه سازی بین دو تغییر در محیط، می بایست به دنبال راهکارهایی برای افزایش تنوع و سازگاری الگوریتم پس از تغییر در محیط نیز بود.

تاکنون برای بهینه‌سازی در محیط‌های پویا از روش‌های تکاملی و هوش جمعی مختلفی از جمله الگوریتم بهینه‌سازی دسته ذرات(PSO)، الگوریتم ژنتیک، سیستم ایمنی مصنوعی و غیره استفاده شده است. الگوریتم بهینه سازی دسته ذرات در سال ۱۹۹۵ توسط kennedy و Eberhart معرفی شد[۴]. از آن سال تاکنون نسخه‌های مختلفی از این الگوریتم پیشنهاد شده و بهبودهای مناسبی نیز بر روی آن داده شده است. از این الگوریتم برای بهینه‌سازی در محیط‌های پویا بسیار استفاده شده است [۷-۵].

در این مقاله یک الگوریتم بهینه‌سازی مبتنی بر الگوریتم PSO پیشنهاد می‌گردد که برای عمل در محیط‌های پویا طراحی شده است و تمام الزامات محیط‌های پویا را برآورده می‌کند. الگوریتم پیشنهادی از راهکارهای خودتطبیقی نوینی برای رسیدن به نتایج بهتر بهره برده است. الگوریتم پیشنهادی بر روی سناریوهای مختلف بنچمارک قله‌های متحرک (MPB)[۸] که از معروفترین بنچمارک‌های محیط‌های پویا است به کار رفته و کارایی آن با سه الگوریتم دیگر به نام‌های mQSO [۵]، CellularPSO [۸] و Adaptive mQSO [۹] در فرکانس های مختلف با تعداد قله های مختلف مقایسه شده است. معیار مقایسه، خطای برون‌خطی است که یکی از معیارهای اصلی مقایسه الگوریتم‌های طراحی شده برای محیط‌های پویا می‌باشد[۱۰]. نتایج آزمایشات نشان می‌دهد که الگوریتم پیشنهادی از کارایی قابل قبولی برخوردار است.

ادامه این مقاله بدین ترتیب سازماندهی شده است. در بخش دوم به تشریح الگوریتم پیشنهادی می‌پردازیم. در بخش سوم نتایج آزمایشات مورد بررسی قرار می‌گیرند و بخش آخر به بیان نتیجه‌گیری می‌پردازد.

۲ رهیافت پیشنهادی

در الگوریتم پیشنهادی تعداد از پیش تعیین شده‌ای دسته وجود دارد که هر دسته وظیفه یافتن و تعقیب کردن یک قله را بر عهده دارد زیرا در محیط‌های پویا بهینه‌های محلی ممکن است پس از تغییر محیط به بهینه سراسری تبدیل شوند بنابراین لازم است تا جای ممکن تمام آنها تحت پوشش دسته‌ها قرار گیرند. ارتباط بین دسته‌ها به دو صورت محلی بر اساس r_{excl} (شعاع تعیین کننده مینیمم فاصله بین دسته ها – شعاع انحصار) و سراسری بر اساس r_{conv} (شعاع تعیین کننده فاصله ذرات معمولی و کوانتوم از یکدیگر برای ایجاد تنوع – شعاع ضد همگرایی) می‌باشد[۵]. هنگامی که تمام دسته‌ها همگرا شده باشند و تعداد قله‌ها بیش از تعداد دسته‌ها باشد، دسته‌ای که کمترین مقدار شایستگی را دارد در فضای مسئله مقداردهی اولیه می‌شود. همچنین اگر دو دسته فاصله کمتر از r_{excl} از هم داشته باشند، دسته‌ای که شایستگی کمتری دارد مقداردهی اولیه می‌شود. زیرا برای هر قله، تحت پوشش بودن توسط یک دسته کفایت می‌کند بنابراین هنگامی که دو دسته به یک قله همگرا شدند باید تنها یکی از آنها باقی بماند و دیگری در فضای مسئله مقداردهی اولیه شود تا شانس این را داشته باشد که به قله‌ای همگرا شود که هنوز تحت پوشش هیچ دسته‌ای قرار نگرفته است. همچنین هنگامی که همه دسته‌ها همگرا شده باشند، هیچ دسته‌ای نمی‌تواند از قله‌ای که تحت پوشش آن است خارج شود زیرا تنوع، گستردگی دسته و سرعت ذرات کمتر از آن است که دسته‌ها بتوانند از بهینه‌هایی که در آنها قرار گرفته‌اند به بیرون حرکت کنند. در این حالت مشکل از آنجا ناشی می‌شود که وقتی تعداد قله‌ها بیش از دسته‌ها باشد و همه دسته‌ها همگرا شده باشند، برخی از قله‌ها هیچ وقت تحت پوشش قرار نمی‌گیرند بنابراین در [۵] پیشنهاد شد در چنین شرایطی دسته‌ای که بدترین مقدار شایستگی را دارد باید در محیط مسئله مقدار دهی اولیه شود تا شانس پیدا کردن قله‌های دیگر که احتمالاً بلندتر هستند نیز به وجود آید. در الگوریتم پیشنهادی از دو تکنیک بالا استفاده شده است یعنی الگوریتم پیشنهادی از چنددستگی، انحصار و ضدهمگرایی استفاده می‌کند[۱۵].

در الگوریتم پیشنهادی همانند [۵] هر دسته از دو نوع ذره یعنی ذرات معمولی و ذرات کوانتوم تشکیل شده است اما نحوه کارکرد ذرات کوانتوم دارای تفاوت‌های اساسی با [۵] می باشد. ذرات معمولی در هر تکرار از اجرای الگوریتم بر اساس سرعت قبلی، بهترین تجربه شخصی و بهترین تجربه گروهی موقعیت خود را بهنگام می‌کنند[۱۱]. در الگوریتم پیشنهادی ابتدا ذرات معمولی موجود در تمام دسته‌ها موقعیت خود را بهنگام می‌کنند سپس موقعیت Gbest هر دسته با استفاده از تنها یک ذره کوانتوم در چند مرحله بهبود می‌یابد. در الگوریتم پیشنهادی پارامتر جدیدی به نام Try-number اضافه شده است که یک ذره کوانتوم تا Try-number بار می‌تواند موقعیت Gbest را بهبود دهد.

پس از اجرای الگوریتم Pso و بهنگام شدن موقعیت Gbest، با استفاده از رابطهٔ (۱) یک موقعیت را در شعاع r_{cloud} در اطراف Gbest در نظر می‌گیریم که در واقع همان موقعیت جدید ذرهٔ کوانتوم است.

$$Gbest = \arg\min_{Pbest_{i,j}} f\left(Pbest_{i,j}\right) \qquad (۱)$$

که در رابطهٔ ۱، $X_{i,j}$ مؤلفهٔ jام از موقعیت ذرهٔ کوانتوم دستهٔ iام است و R_j یک عدد تصادفی با توزیع یکنواخت در بازهٔ [۱, ۱-] می باشد. بنابراین موقعیت ذرهٔ کوانتوم در هر یک از ابعاد فضای جستجو می‌تواند درون شعاع r_{cloud} از ابر تشکیل شده حول و حوش Gbest باشد. پس از تعیین موقعیت ذرهٔ کوانتوم، مقدار شایستگی آن سنجیده می‌شود و با مقدار شایستگی Gbest مقایسه می‌شود. در صورتی که مقدار شایستگی موقعیت ذرهٔ کوانتوم بهتر از Gbest باشد، موقعیت Gbest به موقعیت ذرهٔ کوانتوم تغییر می‌کند و در غیر این صورت موقعیت آن بدون تغییر می‌ماند. رابطهٔ ۱ به اندازه Try-number بار اجرا می‌شود لذا موقعیت Gbest می‌تواند تا Try-number بار بهبود یابد.

استفاده از ایدهٔ ذرهٔ کوانتوم به صورت ذکر شده موجب افزایش سرعت همگرایی بر اساس تعداد ارزیابی شایستگی می‌شود. همچنین می‌تواند مشکل کاهش تنوع در دسته‌ها پس از تغییر محیط [۹] را به خوبی حل کند. همان‌طور که در رابطهٔ ۱ مشاهده می‌شود مقدار شعاع r_{cloud} در تعیین موقعیت ذرهٔ کوانتوم نقش اساسی دارد. در واقع کارایی ذرهٔ کوانتوم تا حد زیادی وابسته به اندازهٔ این شعاع است. در صورتی که مقدار r_{cloud} کوچک باشد توانایی جستجوی محلی الگوریتم بسیار بالا می‌رود اما باعث کاهش طول گام حرکت در هر تکرار و در نتیجه کاهش سرعت همگرایی الگوریتم می‌شود و از توانایی بازیابی تنوع دسته نیز کاسته می‌شود. در صورتی که مقدار این شعاع زیاد باشد، احتمال یافتن موقعیت‌های بهتر در آن کاهش می‌یابد و توانایی جستجوی محلی نیز کاهش می‌یابد.

در الگوریتم پیشنهادی مقدار شعاع r_{cloud} بر اساس طول گام حرکت قله‌ها تعیین می‌شود. در واقع مقدار این پارامتر باید به گونه‌ای تعیین شود که بتواند مشکل کاهش تنوع در دسته را پس از تغییر در محیط به سرعت از بین ببرد و پس از آن کوچکتر شود تا دسته بتواند با انجام یک جستجوی محلی قوی‌تر به نتایج بهتری دست یابد. در الگوریتم پیشنهادی برای دستیابی به این هدف مقدار r_{cloud} برابر با نصف میزان جابجایی قله‌ها در نظر گرفته شده است[۹] سپس با استفاده از رابطهٔ ۲ مقدار r_{cloud} در هر تکرار کاهش می‌یابد تا الگوریتم با افزایش توانایی جستجوی محلی‌اش به نتایج بهتری دست یابد.

$$r_{Cloud}(t+1) = r_{Cloud}(t) \times \left(L_{Low} + \left(Rand \times \left(L_{High} - L_{Low}\right)\right)\right) \qquad (۲)$$

در رابطهٔ (۲) مقدار r_{cloud} در هر تکرار به صورت تصادفی بر اساس مقدار این پارامتر در تکرار قبل بدست می‌آید. L_{Low} و L_{High} برابر با حد پایین و بالای درصد تغییر r_{cloud} نسبت به تکرار قبل است و Rand تابع تولید یک عدد تصادفی با توزیع یکنواخت در بازهٔ [۰،۱] است. بنابراین r_{cloud} در هر تکرار برابر با مقداری تصادفی بین L_{Low} تا L_{High} برابر نسبت به تکرار قبلی می‌باشد. برای این منظور مقدار L_{High} باید عددی کوچکتر مساوی یک در نظر گرفته شود. مقدار r_{cloud} نیز پس از هر تغییر محیط مقداردهی اولیه می‌شود. بدین ترتیب در الگوریتم پیشنهادی بین توانایی افزایش تنوع و توانایی جستجوی محلی تعادل برقرار شده است و الگوریتم می‌تواند هر دوی آنها را به خوبی انجام دهد. دلیل کاهش تصادفی r_{cloud} این است که در محیط‌های پویا الگوریتم نباید از زمان تغییر در محیط خبر داشته باشد و باید آنرا کشف کند لذا استفاده از روش‌های مبتنی بر تکرار مناسب نمی باشد. تاکنون در بسیاری از الگوریتمهای ارائه شده این نکته مهم نادیده گرفته شده است اما در رهیافت پیشنهادی بخوبی این مهم رعایت شده است.

Algorithm : Proposed PSO

```
For each swarm i
 For each Particle j
      initialize x_{i,j} , v_{i,j}
      Pbesti,j = x_{i,j}
 endfor
Gbest = arg min_{Pbest_{i,j}} f(Pbest_{i,j})
endfor
Initialize Test_point
repeat:
    for each swarm i
      for each particle j
        Update X_{i,j}(t+۱) and Pbest_{i,j}(t+۱)
      endfor
      Gbest = arg min_{Pbest_{i,j}} f(Pbest_{i,j})
    endfor
    for each swarm i
      For cnt=۱ to Try_number
        Update Q_{i,j} based on equation ۱
        If f(Q_{i,j}) > f(Gbest_i) THEN
           Gbesti = Q_{i,j}
      Endfor
    Endfor
    Update r_cloud based on equation ۲
    Evaluate Test_point
    IF new value is different from last iteration THEN
       Reinitialize r_cloud
       Reevaluate each particle attractor
       Update swarm attractor
    Execute exclusion and anti convergence[٦]
until stopping criterion is met
```

شکل۱: شبه کد رهیافت پیشنهادی.

برای شناسایی تغییر در محیط در الگوریتم پیشنهادی، یک نقطه در ابتدای اجرای الگوریتم در فضای مسئله در نظر گرفته می‌شود و در پایان هر تکرار مقدار شایستگی آن سنجیده می‌شود. در صورتی که تغییری در محیط رخ داده باشد مقدار شایستگی این نقطه تغییر کرده است. پس از کشف تغییر در محیط ابتدا مقدار r_{cloud} مقداردهی اولیه می‌شود و موقعیت بهترین تجربه شخصی و گروهی ذرات مجدداً مورد ارزیابی قرار می‌گیرد تا مقادیر حافظه معتبر شوند. شبه‌کد رهیافت پیشنهادی در شکل۱ نشان داده شده است.

کنفرانس داده کاوی ایران

۳ ارزیابی

۳ ۱ – معیار قله های متحرک[۱۰]

برای ارزیابی رهیافت پیشنهادی در این مقاله از یک معیار شناخته شده و استاندارد بنام معیار قله‌های متحرک استفاده شده است. در اکثر مقالات برای ارزیابی روش‌های بهینه‌سازی در محیط‌های پویا از این معیار استفاده شده است. بدلیل پارامتریک بودن این معیار، توانایی برای ارزیابی یک دسته از مسائل پویا به آسانی قابل دسترس می‌باشد. این معیار یک فضای n بعدی را بصورت پیش‌فرض در نظر می‌گیرد که مجموعه‌ای از قله‌ها در این فضا وجود دارند.

هر قله دارای موقعیت مکانی، عرض و ارتفاع می‌باشد. زمانی که تغییر در محیط اتفاق می‌افتد (به ازای هر Δe بار ارزیابی، یک بار تغییر در محیط اتفاق می‌افتد) کلیه ویژگی‌های قله‌ها شامل مکان، عرض و ارتفاع تغییر می‌کنند. این معیار را می‌توان بصورت رابطه (۳) فرموله نمود:

$$F(\vec{x},t) = \max(B(\vec{x}), \max_{i=1\ldots m} P(\vec{x}, h_i(t), w_i(t), \vec{p}_i(t))) \quad (۳)$$

در این فرمول، m تعداد قله، $h_i(t)$ ارتفاع قله iام در لحظه t، $w_i(t)$ عرض قله iام در لحظه t، بردار x موقعیت قله در فضا، $B(x)$، مقدار اولیه غیر وابسته به زمان می‌باشد. موقعیت قله‌ها پس از تغییر بر اساس پارامتر s که معرف میزان تغییر است مشخص می‌شود. فرکانس تغییر برابر با Δe می‌باشد که تعداد ارزیابی‌ها بین دو تغییر را مشخص می‌کند. پارامتر λ میزان تغییرات را نسبت به تعییرات دفعه قبل مشخص می‌کند. اگر $\lambda = ۰$ باشد، حرکت‌ها بصورت کاملا تصادفی انجام می‌گیرد. اگر $\lambda = ۱$ باشد، حرکت در جهت قبلی انجام می‌شود . این کار تا رسیدن به مرز فضای مورد نظر ادامه می‌یابد. بصورت ریاضی می‌توان یک تغییر را در یک قله بصورت روابط (۴-۶) نمایش داد:

$$\sigma \in N(0,1)$$

$$h_i(t) = h_i(t-1) + height_severity.\sigma \quad (۴)$$

$$w_i(t) = w_i(t-1) + width_severity.\sigma \quad (۵)$$

$$\vec{p}_i(t) = \vec{p}_i(t-1) + \vec{v}_i(t) \quad (۶)$$

که در این فرمول $v_i(t)$ بصورت رابطه (۷) محاسبه می شود.

$$\vec{v}_i(t) = \frac{s}{\left|\vec{r} + \vec{v}_i(t-1)\right|}((1-\lambda)\vec{r} + \lambda \vec{v}_i(t-1)) \quad (۷)$$

همچنین شکل تابع P، بصورت مخروطی بوده و طبق رابطه (۸) بدست می آید.

$$P(\vec{x}, h_i(t), w_i(t), \vec{p}_i(t))) = h_i(t) - w_i(t) * \left\|\vec{x} - \vec{p}_i(t)\right\| \quad (۸)$$

این معیار دارای سه سناریوی مختلف از مقادیر پارامترها می‌باشد. در این مقاله برای ارزیابی از سناریوی (۲) با پارامتر تنظیم شده بصورت جدول (۱) استفاده شده است[۱۰]. این کار برای فراهم آوردن بستری مناسب برای مقایسه بین الگوریتمهای کار شده قبلی با رهیافت پیشنهادی انجام شده است. برای ارزیابی نتایج از معیار خطای برون‌خطی که با رابطه (۹) محاسبه می‌شود استفاده شده است.

$$offline_{error} = \frac{1}{T}\sum_{t=1}^{T}(Fitness(swarm_{best}(t)) \quad (۹)$$

در این رابطه، T ماکزیمم تکرار، $swarm_{best}(t)$ بهترین راه حل بدست آمده بوسیله کلیه دسته ها در تکرار t ام می باشد.

۳ ۲ –نتایج

برای ارزیابی، رهیافت پیشنهادی با سه الگوریتم شناخته شده به نامهای mQSO۱۰(۵،۵[۹]) [۵] و Adaptive mQSO [۹] و Cellular PSO[۸] مورد مقایسه قرار گرفته است. نتایج با توجه به پارامترهای جدول (۱) که با نام سناریو ۲ در مسئله MPB معروف است آورده شده است. تنها پارامتر متفاوت، تعداد قله‌ها و فرکانس تغییر می‌باشد که برای ارزیابی بهتر بین روش‌ها از ۱ قله تا ۲۰۰ قله در فرکانس‌های تغییر ۵۰۰، ۱۰۰۰، ۲۵۰۰ و ۵۰۰۰ در نظر گرفته شده است.

کنفرانس داده کاوی ایران

جدول (۱): پارامترهای مسئله MPB.

Mتعداد قله ها	بین ۱ تا ۲۰۰
فرکانس تغییر	۵۰۰ ، ۱۰۰۰، ۲۵۰۰ و ۵۰۰۰
میزان تغییر ارتفاع	۷,۰
میزان تغییر عرض	۱,۰
شکل قله	cone
تابع اولیه	ندارد
s طول جابجایی	۱,۰
N تعداد ابعاد	۵
محدوده مکانی قله ها	[۰,۰،۱۰۰,۰]
محدوده پارامتر ارتفاع	[۳۰,۰،۷۰,۰]
محدوده پارامتر عرض	[۱،۱۲]
مقدار ارتفاع اولیه قله ها	۵۰

در آزمایشات تعداد دسته ها برابر با ۱۰ و تعداد ذرات در هر دسته برابر با ۵ و مقدار Try_number نیز برابر با ۵ در نظر گرفته شده است. مقدار اولیه r_{cloud} برابر با نصف مقدار طول جابه‌جایی قله‌ها (مقدار ۰,۵) در نظر گرفته می‌شود. تنظیمات PSO ، انحصار و ضدهمگرایی مطابق با [۵] در نظر گرفته شده است. مقدار L_{Low} و L_{High} نیز به ترتیب برابر ۰,۶ و ۱ در نظر گرفته شده‌اند. مقدار این پارامترها بر اساس آزمایشات بسیار زیادی که در این زمینه انجام داده-ایم تنظیم شده‌اند. آزمایشات ۳۰ بار تکرار شده‌اند و متوسط مقدار خطای برون‌خطی و خطای استاندارد[۱۰] الگوریتم پیشنهادی به همراه سه الگوریتم دیگر در جداول ۲ تا ۵ در فرکانس های ۵۰۰، ۱۰۰۰، ۲۵۰۰ و ۵۰۰۰ با تعداد قله های مختلف نشان داده شده است. در این جداول خطای استاندارد در کنار خطای برون‌خطی درون پرانتز نشان داده شده و دو نتیجه بهتر در هر ردیف برای تسهیل مقایسه بصورت پررنگ نوشته شده است.

جدول(۲): مقایسه خطای برون‌خطی و خطای استاندارد الگوریتم‌ها در فرکانس تغییرات ۵۰۰.

M	Cellular PSO	mQSO ۱۰(۵+۵^{q})	Adaptive mQSO	Proposed PSO
۱	۱۳,۴۶(۰,۷)	۱۳,۶۷(۰,۷۵)	**۳,۰۸(۰,۲۲)**	**۹,۵۹(۰.۶۷)**
۵	۹,۶۳(۰,۴۹)	۸,۹۸(۰,۳۵)	**۴,۹۷(۰,۲۷)**	**۷,۸۷(۰,۳۷)**
۱۰	۹,۴۲(۰,۲۱)	۸,۳(۰,۳۱)	**۶,۶۶(۰,۲۹)**	**۶,۲۹(۰,۳۵)**
۲۰	۸,۸۴(۰,۲۸)	۸,۷۹(۰,۲۳)	**۷,۰۰(۰,۲)**	**۶,۷۹(۰,۲۹)**
۳۰	۸,۸۱(۰,۲۴)	۸,۷۷(۰,۲۹)	**۷,۱۹(۰,۲۳)**	**۷,۶۸(۰,۳)**
۴۰	۸,۹۴(۰,۲۴)	۸,۲۵(۰,۲)	**۷,۳۷(۰,۲۳)**	**۷,۴۴(۰,۲۷)**
۵۰	۸,۶۲(۰,۲۳)	۸,۵۶(۰,۲)	**۷,۷۸(۰,۲۹)**	**۸,۱۶(۰,۳)**
۱۰۰	۸,۵۴(۰,۲۱)	۸,۲۴(۰,۲)	**۷,۸(۰,۱۷)**	**۸,۱۳(۰,۲۱)**
۲۰۰	۸,۲۸(۰,۱۸)	**۷,۵۴(۰,۱۸)**	۷,۵۹(۰,۱۸)	**۷,۰۵(۰,۲)**

جدول(۳): مقایسه خطای برون‌خطی و خطای استاندارد الگوریتم‌ها در فرکانس تغییرات ۱۰۰۰.

M	Cellular PSO	mQSO ۱۰(۵+۵^{q})	Adaptive mQSO	Proposed PSO
۱	۶,۷۷(۰,۳۸)	۸,۲۶(۰,۳۹)	**۱,۹۷(۰,۱۲)**	**۴,۷۱(۰,۳۵)**
۵	۵,۳۰(۰,۳۲)	۶,۰۰(۰,۲۴)	**۳,۲۵(۰,۱۴)**	**۴(۰,۲)**
۱۰	۵,۱۵(۰,۱۳)	۵,۰۴(۰,۱۷)	**۳,۸۸(۰,۱۵)**	**۳,۶۵(۰,۲)**
۲۰	۵,۲۳(۰,۱۸)	۵,۹۱(۰,۱۷)	**۴,۸۴(۰,۱۷)**	**۴,۷۵(۰,۲۳)**
۳۰	۵,۳۳(۰,۱۶)	۶,۱۲(۰,۱۶)	**۵,۰۷(۰,۱۶)**	**۴,۸(۰,۱۷)**
۴۰	۵,۶۱(۰,۱۶)	۵,۸۶(۰,۱۶)	**۵,۲۰(۰,۱۹)**	**۵,۲۹(۰,۱۴)**
۵۰	۵,۵۵(۰,۱۴)	۵,۹۰(۰,۱۷)	**۵,۱۱(۰,۱۴)**	**۵,۱۵(۰,۱۶)**
۱۰۰	۵,۵۷(۰,۱۲)	۵,۵۷(۰,۱۴)	**۵,۴۱(۰,۱۵)**	**۴,۷۹(۰,۱۶)**
۲۰۰	۵,۵۰(۰,۱۲)	۵,۴۷(۰,۱۴)	**۵,۰۷(۰,۱۴)**	**۴,۹۳(۰,۱۸)**

همانطور که مشاهده می‌شود نتایج نشان می‌دهد روش پیشنهادی نسبت به دو روش MQSO۱۰(۵+۵q) و Cellular PSO بهتر عمل می‌نماید و با الگوریتم Adaptive mQSO قابل مقایسه می‌باشد. هر چه تعداد قله‌ها بالاتر و فرکانس ارزیابی بالاتر باشد روش پیشنهادی نسبت به Adaptive mQSO جواب بهتری را ارائه می‌دهد.

جدول(۴): مقایسه خطای برون‌خطی و خطای استاندارد الگوریتم‌ها در فرکانس تغییرات ۲۵۰۰.

M	Cellular PSO	mQSO ۱۰(۵+۵^q)	Adaptive mQSO	Proposed PSO
۱	۴,۱۵(۰,۲۵)	۳,۷۳(۰,۱۸)	**۰,۸۲(۰,۰۵)**	**۲,۱۹(۰,۱۵)**
۵	۲,۸۵(۰,۲۴)	۳,۴۵(۰,۱۵)	**۱,۹۱(۰,۱۴)**	**۱,۹۶(۰,۱۲)**
۱۰	۲,۸۰(۰,۱۰)	۲,۹۶(۰,۱)	**۲,۲۵(۰,۰۸)**	**۱,۷۹(۰,۰۸)**
۲۰	۳,۴۱(۰,۱۴)	۴(۰,۱۳)	**۲,۹۴(۰,۱)**	**۳,۱۴(۰,۱۴)**
۳۰	۳,۶۲(۰,۱۲)	۳,۹۴(۰,۱۱)	**۳,۲۷(۰,۱۱)**	**۲,۹۳(۰,۱۳)**
۴۰	۳,۸۴(۰,۱۲)	۳,۹۷(۰,۱)	**۳,۳۲(۰,۰۹)**	**۳,۲۵(۰,۱۳)**
۵۰	۳,۸۶(۰,۱۰)	۳,۷۹(۰,۱)	**۳,۳۶(۰,۱)**	**۳,۲۳(۰,۱۳)**
۱۰۰	۴,۱۰(۰,۱۱)	۳,۹۳(۰,۱۱)	**۳,۲۸(۰,۰۸)**	**۳,۲۰(۰,۱۱)**
۲۰۰	**۳,۳۸(۰,۰۸)**	۳,۶۵(۰,۰۹)	۳,۹۷(۰,۱۰)	**۲,۹۳(۰.۰۸)**

جدول(۵): مقایسه خطای برون‌خطی و خطای استاندارد الگوریتم‌ها در فرکانس تغییرات ۵۰۰۰.

M	Cellular PSO	mQSO ۱۰(۵+۵^q)	Adaptive mQSO	Proposed PSO
۱	۲,۵۵(۰,۱۲)	۲,۵۱(۰,۱۲)	**۰,۴۴(۰,۰۲)**	**۱,۳۸(۰,۱)**
۵	۱,۶۸(۰,۱۱)	۲,۲۴(۰,۱)	**۱,۰۹(۰,۱۴)**	**۱,۱۸(۰,۰۸)**
۱۰	۱,۷۸(۰,۰۵)	۲(۰,۰۷)	**۱,۵۱(۰.۰۷)**	**۱,۰۱(۰,۰۶)**
۲۰	۲,۶۱(۰,۰۷)	۳,۱۷(۰,۱)	**۲,۰۴(۰,۰۶)**	**۲,۳۱(۰,۱۳)**
۳۰	۲,۹۳(۰,۰۸)	۳,۱۶(۰,۰۹)	**۲,۳۵(۰,۰۶)**	**۲,۵(۰,۱۱)**
۴۰	۳,۱۴(۰,۰۸)	۳,۲۳(۰,۱۲)	**۲,۶(۰,۰۸)**	**۲,۵۱(۰,۱)**
۵۰	۳,۲۶(۰,۰۸)	۳,۰۷(۰,۰۹)	**۲,۵۲(۰,۰۷)**	**۲,۴۷(۰,۱۱)**
۱۰۰	۳,۴۱(۰,۰۷)	۳,۰۸(۰,۰۷)	**۲,۵۷(۰,۰۷)**	**۲,۳۸(۰,۰۷)**
۲۰۰	۳,۴۰(۰,۰۶)	۳,۰۹(۰,۰۸)	**۲,۶۵(۰,۰۷)**	**۲,۳۵(۰,۰۶)**

دلیل اینکه نتایج مربوط به Adaptive mQSO در برخی مواقع بهتر از الگوریتم پیشنهادی بوده است این موضوع می باشد که در این الگوریتم تعداد دسته ها به صورت تطبیقی و بر اساس تعدا قله های یافت شده تنظیم می شود. بدین ترتیب می تواند عملکرد بهتری را در تعداد قله های کمتر از ۱۰ ارائه کند. در واقع به دلیل ثابت بودن تعداد دسته ها در روش پیشنهادی برابر ۱۰ دسته، در تعداد قله های کمتر از ۱۰، تعدادی دسته که به هیچ قله ای همگرا نشده اند تنها فقط باعث هدر شدن تعدادی ارزیابی از تابع شایستگی می شوند و هیچ کاری نمی توانند انجام دهند که این مسئله برای Adaptive mQSO حل شده است. همچنین در تعداد قله های بالاتر، Adaptive mQSO تعداد دسته های بیشتری در اختیار دارد که باعث بهبود نتایج آن شده است. با این حال الگوریتم پیشنهادی با تعداد ثابتی جمعیت توانسته از دو روش دیگر که در آنها تعداد جمعیت ثابت است نتایج فوق العاده بهتری را ارائه کند و نتایج قابل مقایسه ای نیز با الگوریتم تطبیقی Adaptive mQSO ارائه دهد.

۴ -نتیجه‌گیری

در این مقاله یک روشی نوین برای بهینه‌سازی در محیط‌های پویا پیشنهاد شد و نتایج بر روی تابع معیار حرکت قله‌ها با چندین روش شناخته شده دیگر مورد مقایسه قرار گرفت. نتایج آزمایشات نشان داد که الگوریتم پیشنهادی از کارایی قابل قبولی مخصوصاً در مقابل روش‌های mQSO و Cellular PSO که دارای جمعیت ثابتی بودند برخوردار بود اما نتایج آن در مواردی پایین‌تر از روش Adaptive mQSO بود. دلیل این امر، نیز برتری Adaptive mQSO در تطبیقی بودن تعداد دسته‌ها متناسب با قله ها می باشد. برای بهبود الگوریتم پیشنهادی می‌توان تعداد دسته‌ها را در آن به صورت تطبیقی در نظر گرفت به طوری که به ازای هر قلهٔ یافت شده در فضای مسئله یک دسته ایجاد شود.

کنفرانس داده کاوی ایران

۵ -مراجع

۱- Y. Jin and J. Branke, “Evolutionary Optimization in uncertain environments –A Survey”, in IEEE Transaction on Evolutionary Computation, vol. ۹ , No. ۳ , pp. ۳۰۳-۳۱۷, ۲۰۰۵.

۲- C. Li and S. Yang, “Fast Multi-Swarm Optimization for Dynamic Optimization Problems”, In ۴th International Conference on Natural Computation, Jinan, Shandong, China, vol. ۷, pp.۶۲۴–۶۲۸, ۲۰۰۸.

۳- S. Yang and C. Li, “A Clustering Particle Swarm Optimizer for Locating and Tracking Multiple Optima in Dynamic Environments”, in IEEE Transaction on Evolutionary Computation, pp. ۱-۱۶, ۲۰۱۰.

۴- J. Kennedy and R. Eberhart, “Particle Swarm Optimization”, in IEEE International Conference on Neural Networks, Vol. ۴, pp. ۱۹۴۲-۱۹۴۸, Perth, November ۱۹۹۵.

۵- T. Blackwell and J. Branke, “Multiswarm, Exclusion, and Anti-Convergence in Dynamic Environment”, in IEEE Transaction on Evolutionary Computation, Vol. ۱۰, No. ۴, pp. ۴۵۹-۴۷۲, ۲۰۰۶.

۶- T. Blackwell and J. Branke, “Particle Swarms for Dynamic Optimization Problems”, in Swarm Intelligence, pp.۱۹۳–۲۱۷, ۲۰۰۸.

۷- W. Du and B. Li, “Multi-Strategy Ensemble Particle Swarm Optimization for Dynamic Optimization”, in Information Sciences: an International Journal Vol.۱۷۸, pp.۳۰۹۶–۳۱۰۹, ۲۰۰۸.

۸- B. Hashemi and M. R. Meybodi, “Cellular PSO: A PSO for Dynamic Environments”, in Advances in Computation and Intelligence, Lecture Notes in Computer Science, vol. ۵۸۲۱, pp. ۴۲۲-۴۳۳, ۲۰۰۹.

۹- T. Blackwell and J. Branke, “Particle Swarms for Dynamic Optimization Problems”, in Swarm Intelligence, pp.۱۹۳–۲۱۷, ۲۰۰۸.

۱۰- http://www.aifb.unikarlsruhe.de/~jbr/MovPeaks/

۱۱- Y. Shi and R. Eberhart, “A Modified Particle Swarm Optimization”, In IEEE International Conference on Evolutionary Computation Proceedings, pp. ۶۹-۷۳, Anchorage, ۱۹۹۸.

کنفرانس داده کاوی ایران