

PSO-Great Deluge: یک روش ترکیبی جدید برای بهینه‌سازی

سجاد قطعی^۱؛ محمدرضا میبیدی^۲؛ مجتبی حسین زاده^۳؛ زهرا قطعی^۴

چکیده

در این مقاله یک روش ترکیبی جدید برای بهینه‌سازی با نام PSO-Great Deluge ارائه شده است. ایده اصلی این روش، ترکیب الگوریتم بهینه‌سازی گروه ذرات (PSO) با الگوریتم دلیوج (GDA) می‌باشد. در روش ارائه شده از خاصیت جستجوی سراسری الگوریتم بهینه‌سازی گروه ذرات و از خاصیت جستجوی محلی الگوریتم دلیوج بصورت سری بهره گرفته شده است، بطوری که در ابتدا از بهینه‌سازی گروه ذرات برای جستجوی محیط استفاده می‌شود و نتایج آن به الگوریتم دلیوج سپرده شده تا به جستجوی دقیق‌تر در پیرامون جواب بدست آمده پرداخته شود. روش پیشنهادی بر روی چند تابع استاندارد آزمایش شده و کارایی آن با چند روش همچون، الگوریتم ژنتیک و الگوریتم بهینه‌سازی گروه ذرات استاندارد مورد مقایسه قرار گرفته است. نتایج آزمایشات حاکی از این می‌باشد که روش پیشنهادی در اکثر موارد جواب قابل قبولی را ارائه داده و به طور معمول دارای پایداری لازم می‌باشد.

کلمات کلیدی

الگوریتم ژنتیک، بهینه‌سازی گروه ذرات، PSO-Great Deluge

PSO-Great Deluge: A New Hybrid Method for Optimization

Sajjad Ghatei; Mohammad Reza Meybodi; Mojtaba Hosseinzadeh; Zahra Ghatei

ABSTRACT

In this paper a new combined approach is presented known as PSO-Great Deluge; the main idea of this approach is to combine particle swarm optimization (PSO) with great deluge algorithm (GDA). In this approach, global search character of PSO and local search factor of GDA are used based on series. At the first step, PSO algorithm is used to search around environment and its results are given to GDA to search about taken results accurately. This approach is tested and its efficiency is compared with methods like genetic algorithm and standard PSO results. In this paper is shown that this approach has considerable results and usually has good stability.

KEYWORDS

Particle Swarm Optimization, Great Deluge Algorithm, Genetic Algorithm

۱. مقدمه

هدف از بهینه‌سازی جستجوی مقادیری به ازای مجموعه‌ای از پارامترهاست که تابع هدف مربوط به یک محدودیت مشخص را کمینه یا بیشینه کند. کلیه مقادیر مناسب جهت این امر را راه‌حل‌های ممکن و بهترین مقدار از این مقادیر را راه حل بهینه می‌نامند. بهینه‌سازی در زمینه‌های فراوانی نظیر تجارت، صنعت، مهندسی و علوم کامپیوتر در سطح وسیعی کاربرد دارد [۱]. یافتن راه‌حل برای مسائلی که از نوع NP می‌باشند بسیار مشکل است. روش‌هایی از جمله روش‌های هیوریستیک تا حدی این مشکل را حل نموده‌اند. توسط این روش‌ها، راه‌حل‌هایی پیدا می‌شود که تا حدی به جواب نزدیک‌ترند. بعضی از روش‌های هیوریستیک معروف عبارتند از: الگوریتم ژنتیک، الگوریتم بهینه‌سازی گروه ذرات و روش شبیه‌ساز حرارتی [۱]. کاربرد عمده این روش‌ها در حل مسائل بهینه‌سازی می‌باشد.

^۱ دانشگاه آزاد اسلامی، واحد اهر، باشگاه پژوهشگران جوان، اهر، ایران، s.ghatei@qiau.ac.ir

^۲ دانشکده کامپیوتر و فن آوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران، mmeybodi@aut.ac.ir

^۳ دانشگاه آزاد اسلامی، واحد اهر، گروه کامپیوتر، اهر، ایران، m.hosseinzadeh@qiau.ac.ir

^۴ دانشگاه آزاد اسلامی، واحد اهر، گروه کامپیوتر، اهر، ایران، z.ghatei@yahoo.com

یک روش هیوریستیک دیگر که بسیار شبیه به روش شبیه ساز حرارتی می باشد، الگوریتم دلیوج است که در سال ۱۹۹۳ توسط دونک معرفی شده است [۲]. در این الگوریتم با تعریف یک پارامتر با عنوان سطح آب محدوده ای را برای جواب های به دست آمده تعیین می کنیم و هر بار جواب به دست آمده را با این پارامتر مورد مقایسه قرار می دهیم. در مسائل بیشینه سازی در هر مرحله مقدار این پارامتر افزایش می یابد و در مسائل کمینه سازی مقدار این پارامتر در هر مرحله کاهش می یابد.

در [۲] این الگوریتم به خوبی برای حل مسئله فروشنده دوره گرد ۴۴۲ و ۵۳۲ شهر به کار برده شده است. بورکی در [۳، ۴] از این الگوریتم برای حل مسئله زمانبندی امتحان ها استفاده کرده است. در [۵] ناهاس و همکارانش این الگوریتم را برای حل مسئله تخصیص بافر در خطوط تولید غیرقابل اعتماد به کار برده اند. در [۶] از روش ترکیبی دلیوج با دسته مورچگان برای حل مسئله تخصیص افزونگی استفاده شده است. در [۷] با بهبود الگوریتم دلیوج میزان کارایی آن بر روی مسئله لایه بندی پویا سنجیده شده است. خطاب و دیگر نویسندگان در [۸] از این الگوریتم برای بهینه سازی کارایی سیستم های سری-موازی سود برده اند. در [۹] نوعی الگوریتم دلیوج اصلاح شده غیر خطی برای حل مسئله زمانبندی امتحان ها ارائه شده است.

یک روش دیگر که در اینجا به آن پرداخته شده است، الگوریتم بهینه سازی گروه ذرات (PSO) می باشد. این الگوریتم یک روش بهینه سازی است که از رفتار اجتماعی موجود در طبیعت الهام گرفته شده است. سادگی، پیاده سازی، مقیاس پذیری در ابعاد و کارایی خوب تجربی از خواص مطلوب این الگوریتم می باشد. لذا الگوریتم PSO در برنامه ریزی غیر خطی و بهینه سازی توابع با دامنه های پیوسته گزینه قابل توجه است.

در این مقاله یک مدل جدید بنام PSO-Great Deluge پیشنهاد می گردد. در مدل پیشنهادی از هر دو خاصیت محلی و عمومی بودن الگوریتم ها به شکل خوب استفاده شده است. یعنی در ابتدا الگوریتم PSO با گام های بزرگ به جستجو پرداخته و بعد از به دست آوردن یک سری نتایج، نتایج جدید به عنوان ورودی به الگوریتم دلیوج سپرده می شود و در این مرحله این الگوریتم با گام های بسیار کوچک به جستجوی فضا می پردازد.

این مقاله به صورت زیر بخش بندی شده است: بخش دوم مختصراً به معرفی الگوریتم دلیوج پرداخته است؛ در بخش سوم الگوریتم بهینه سازی گروه ذرات توضیح داده شده است؛ در بخش چهارم مدل پیشنهادی شرح داده می شود؛ بخش پنجم شامل نتایج آزمایش ها می باشد و بخش پایانی به نتیجه گیری می پردازد.

۲. الگوریتم دلیوج

الگوریتم دلیوج یک روش جامع برای حل مسائل بهینه سازی می باشد که اولین بار توسط دونک در سال ۱۹۹۳ مطرح شد. مثل سایر روش های جستجو کننده محلی، این روش نیز به ترتیب راه حل های رایج را به شرط بهتر بودن، با بهترین جواب یافته شده تا کنون جایگزین می کند. این عمل تا زمانی که شرایط توقف ارضاء نشده ادامه می یابد. راه حل های جدید از همسایگی جواب های قبلی انتخاب می شود. مکانیسم پذیرش یا رد راه حل جدید از همسایگی در این الگوریتم، متفاوت از روش های دیگر است.

در الگوریتم دلیوج جواب هایی مورد قبول قرار خواهند گرفت که مقدار آنها برابر یا بیشتر (برای مسائل بیشینه سازی) از مقدار سطح آب (WL) باشد. مقدار WL نیز در هر مرحله به طور یکنواخت به اندازه یک مقدار مشخص (UP) افزایش می یابد. افزایش WL تا جایی ادامه خواهد یافت که دیگر جوابی بهتر از بهترین جواب یافته شده تا کنون به دست نیاید و WL برابر با بهترین جواب یافته شده تا کنون باشد. در این مرحله، باز الگوریتم چندین بار تکرار خواهد شد و اگر نتیجه ای بهتر حاصل نشد الگوریتم به پایان می رسد. مقدار اولیه WL برابر با جواب اولیه بدست آمده خواهد بود. در شکل (۱) شبه کد مربوط به الگوریتم دلیوج آورده شده است.

```

Choose an initial configuration as Old_Config
Choose WL and Up
For n = ۰ to # of iterations
    Generate a small stochastic perturbation...
    New_Config of the solution
    If Fitness (New_Config) > WL
        Old_Config := New_Config
    End If
    WL = WL + Up
End For
    
```

شکل (۱) شبه کد الگوریتم دلیوج

۳. الگوریتم بهینه سازی گروه ذرات

بهینه سازی گروه ذرات، یک تکنیک بهینه سازی احتمالی است که بر مبنای جمعیت کار می کند. این روش در سال ۱۹۹۵ توسط ابرهات و کندی ارائه شد [۱۰] و ایده اصلی آن از رفتار دسته جمعی ماهی ها یا پرندگان به هنگام جستجوی غذا الهام گرفته شده است. گروهی از پرندگان در

فضا به صورت تصادفی به دنبال غذا می‌گردند. تنها یک تکه غذا در فضای مورد بحث وجود دارد. هیچ یک از پرندگان محل غذا را نمی‌دانند. یکی از بهترین استراتژی‌ها می‌تواند دنبال کردن پرنده‌ای باشد که کمترین فاصله را تا غذا داشته باشد. این استراتژی در واقع جانمایه الگوریتم PSO است. در الگوریتم PSO، هر راه‌حل که به آن یک ذره گفته می‌شود، معادل یک پرنده در الگوی حرکت جمعی پرندگان می‌باشد. هر ذره یک مقدار شایستگی دارد که توسط یک تابع شایستگی محاسبه می‌شود. هر چه ذره در فضای جستجو به هدف (غذا در مدل حرکت پرندگان) نزدیک‌تر باشد، شایستگی بیشتری دارد. همچنین هر ذره دارای یک سرعت است که هدایت حرکت ذره را به عهده دارد. هر ذره با دنبال کردن ذرات بهینه در حالت فعلی، به حرکت خود در فضای مسئله ادامه می‌دهد.

آغاز کار PSO به این شکل است که گروهی از ذرات (راه‌حل‌ها) به صورت تصادفی به وجود می‌آیند، با بروز شدن در طی نسل‌ها سعی در یافتن راه‌حل بهینه می‌نمایند. در هر گام، هر ذره با استفاده از دو بهترین مقدار به روز می‌شود. اولین مورد، بهترین موقعیتی است که تا کنون ذره موفق به رسیدن به آن شده است. موقعیت مذکور با نام $pbest$ شناخته و نگهداری می‌شود. بهترین مقدار دیگری که توسط الگوریتم مورد استفاده قرار می‌گیرد، بهترین موقعیتی است که توسط جمعیت ذرات به دست آمده است؛ این موقعیت با $gbest$ نمایش داده می‌شود. در برخی از ویرایش‌های PSO، ذره قسمت‌هایی از جمعیت را که همسایگان توپولوژیکی‌اش هستند انتخاب و تنها آنها را در اعمال خود دخیل می‌کند که در این حالت بهترین راه‌حل محلی که با $lbest$ نشان داده می‌شود، به جای $gbest$ مورد استفاده قرار می‌گیرد. پس از یافتن بهترین مقادیر، سرعت و مکان هر ذره با استفاده از معادلات (۱) و (۲) به روز می‌شوند.

$$v = v + c_1 \times rand \times (pbest - position) + c_2 \times rand \times (gbest - position) \quad (1)$$

$$position = position + v \quad (2)$$

در معادلات (۱) و (۲)، v سرعت ذره و $position$ محل فعلی ذره هستند که هر دو آرایه‌هایی به طول تعداد ابعاد مساله می‌باشند. $rand$ یک عدد تصادفی در بازه (۰،۱) است. c_1 و c_2 نیز پارامترهای یادگیری هستند و معمولاً $c_1 = c_2 = 2$ در نظر گرفته می‌شوند. سرعت ذرات در هر بعد به یک مقدار V_{max} محدود می‌شود. اگر مجموع شتاب‌ها باعث شوند که سرعت در یک بُعد از V_{max} بیشتر شود، مقدار سرعت در آن بُعد برابر V_{max} می‌شود. شبه کد الگوریتم PSO در شکل (۲) مشاهده می‌شود.

```

For each particle
  Initialize particle
End For
Do
  For each particle
    Calculate fitness value of the particle  $fp$ 
    /*updating particle's best fitness value so far*/
    If  $fp$  is better than  $pbest$ 
      set current value as the new  $pbest$ 
    End For
    /*updating population's best fitness value so far*/
    Set  $gbest$  to the best fitness value of all particles
  For each particle
    Calculate particle velocity according equation (۱)
    Update particle position according equation (۲)
  End For
While maximum iterations OR minimum error criteria is not attained
  
```

شکل (۲) شبه کد الگوریتم PSO

۴. مدل پیشنهادی PSO-Great Deluge

در این مدل جدید، از ترکیب الگوریتم دلیوج و الگوریتم بهینه‌سازی گروه ذرات استفاده شده است. هدف این مدل، این بوده که با ترکیب خاصیت محلی الگوریتم دلیوج و خاصیت سراسری الگوریتم بهینه‌سازی گروه ذرات به یک موازنه در جستجو برسیم. از هر دو خاصیت محلی و عمومی الگوریتم‌ها به شکل خوب استفاده شده است، یعنی در ابتدا الگوریتم PSO با گام‌های بزرگ به جستجو پرداخته و بعد از به دست آوردن یک سری نتایج، نتایج جدید به عنوان ورودی به الگوریتم دلیوج سپرده می‌شود و در این مرحله این الگوریتم با گام‌های بسیار کوچک به جستجوی فضا می‌پردازد. شبه کد الگوریتم پیشنهادی در شکل (۳) آمده است.

۵. نتایج شبیه‌سازی‌ها

آزمایش‌ها بر روی ۴ تابع استاندارد صورت گرفته است که معمولاً به عنوان معیار سنجش الگوریتم‌های بهینه‌سازی در فضاهای پیوسته و ایستا مورد استفاده قرار می‌گیرند. توابع استفاده شده عبارتند از: آکلی، روزنبراک، اسفیر و استپ. توابعی که آزمایش‌ها روی آنها اجرا شده است، شامل

بهینه‌سازی تعدادی از توابع N-متغیره می‌باشد. برای مقایسه دقیق‌تر روش پیشنهادی با روش‌های موجود، آزمایش‌ها در فضاها ۱۰، ۲۰ و ۳۰ بُعدی انجام شده است. لیست این توابع و مشخصه‌های آنها در جدول (۱) آمده است. برای ارزیابی روش ترکیبی پیشنهادی، آن را با سایر روش‌های متناظر مورد مقایسه قرار دادیم.

```

For each particle  $i$ 
  Randomly initialize  $v_i, x_i = p_i$ 
  Evaluate  $f(p_i)$ 
   $Pg = \arg \max \{f(p_i)\}$ 
End for
Repeat
  each particle  $i$ 
    Update particle position  $x_i$  According to equation below
     $v_i = X[v_i + c_1 e_{r_1} (Pg - x_i) + c_2 e_{r_2} (p_i - x_i)]$ ,
     $x_i = x_i + v_i$ 
    evaluate  $f(x_i)$ 
    If ( $f(x_i) > f(pg)$ )
       $p_i = x_i$ 
    End If
    If ( $f(x_i) > f(p_i)$ )
       $Pg = \arg \max \{f(p_i)\}$ 
    End If
  Until termination criterion reached
  Choose the best individual as  $Old\_Config$ 
  Choose  $WL$  and  $Up$ 
For  $n = 1$  to # of iterations
  Generate a small stochastic perturbation  $New\_Config$  of the solution
  If  $Fitness(New\_Config) > WL$ 
     $Old\_Config := New\_Config$ 
  End If
   $WL = WL + Up$ 
End For

```

شکل (۳) شبه کد الگوریتم پیشنهادی

جدول (۱) توابع استاندارد آکلی، روزنبراک، اسفیر و استپ

نام تابع	محدوده متغیرها	معادله تابع
آکلی	± 32	$20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)}$
روزنبراک	$\pm 2/0.4$	$\sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$
اسفیر	$\pm 5/12$	$\sum_{i=1}^n x_i^2$
استپ	$\pm 5/12$	$\sum_{i=1}^n \lfloor x_i \rfloor$

در این روش ضریب‌های c_1 و c_2 به ترتیب برابر $2/8$ و $1/3$ در نظر گرفته شده و وزن اینرسی w را از رابطه (۳) به دست می‌آوریم. تعداد ذرات گروه و میزان فرآیند افزایش سطح آب (UP) در توابع مختلف متفاوت بوده است، ولی به طور میانگین آنها را به ترتیب ۳۰ و 0.0002 در نظر گرفته‌ایم. لازم به ذکر است که آزمایشات ۳۰ بار تکرار شده و متوسط نتایج، بهترین و بدترین آنها در جداول (۲) تا (۵) آمده‌اند.

$$w = \frac{2}{|2 - (c_1 + c_2) - \sqrt{(c_1 + c_2)^2 - 4 \times (c_1 + c_2)}|} \quad (3)$$

جدول (۲) نتایج الگوریتم پیشنهادی، PSO استاندارد و الگوریتم ژنتیک بر روی تابع آکلی

ابعاد	الگوریتم	بهترین	متوسط	بدترین
۱۰	الگوریتم ژنتیک	۰/۴۵۹۸	۰/۷۸۶۶	۰/۹۴۵۳
	PSO	۲/۶۶e-۱۵	۲/۶۶e-۱۵	۲/۶۶e-۱۵
	الگوریتم پیشنهادی	۲/۶۶e-۱۵	۲/۶۶e-۱۵	۲/۶۶e-۱۵
۲۰	الگوریتم ژنتیک	۱/۹۸۰۰	۳/۳۴۵۳	۳/۸۴۰۳
	PSO	۲/۶۶e-۱۵	۵/۵۴e-۱۵	۲/۳۲e-۱۳
	الگوریتم پیشنهادی	۲/۶۶e-۱۵	۳/۵۲e-۱۵	۲/۱۸e-۱۴
۳۰	الگوریتم ژنتیک	۳/۶۵۹۵	۵/۷۵۹۸	۶/۸۷۴۳

۲/۰۹۴۶	۲/۱۸e-۱۰	۲/۰۷e-۵	PSO	
۲/۱۲۸۶	۲/۳۶e-۱۳	۱/۳۹e-۵	الگوریتم پیشنهادی	

جدول (۳) نتایج الگوریتم پیشنهادی، PSO استاندارد و الگوریتم ژنتیک بر روی تابع روزنبراک

ابعاد	الگوریتم	بهترین	متوسط	بدترین
۱۰	الگوریتم ژنتیک	۱/۶۹۳۷	۴/۷۷۱۱	۵/۳۰۰۵
	PSO	۲/۳۵e-۰۸	۲/۸۲e-۰۵	۲/۲۲e-۰۳
	الگوریتم پیشنهادی	۴/۹۷e-۱۰	۷/۷۹e-۰۷	۳/۹۲e-۰۵
۲۰	الگوریتم ژنتیک	۵/۷۸۱۲	۷/۱۱۰۹	۸/۲۸۶۶
	PSO	۲/۷۲e-۰۵	۲/۶۵e-۰۱	۲/۵۴۳۴
	الگوریتم پیشنهادی	۴/۸۵e-۰۷	۷/۷۵e-۰۴	۳/۵۴e-۰۱
۳۰	الگوریتم ژنتیک	۱۹/۰۰۶۵	۲۱/۲۳۴۳	۲۳/۳۰۷۲
	PSO	۵/۷۳۵۰	۷/۵۱۱۱	۸/۱۴۶۰
	الگوریتم پیشنهادی	۵/۴۳۳۶	۶/۴۲۷۰	۸/۶۷۱۷

جدول (۴) نتایج الگوریتم پیشنهادی، PSO استاندارد و الگوریتم ژنتیک بر روی تابع اسفیر

ابعاد	الگوریتم	بهترین	متوسط	بدترین
۱۰	الگوریتم ژنتیک	۱/۳۳e-۵	۰/۰۴۰۹	۰/۰۵۹۸
	PSO	۱/۹۷e-۱۰۶	۱/۸۱e-۷۴	۲/۶۴e-۶۸
	الگوریتم پیشنهادی	۲/۵۲e-۱۱۳	۱/۰۸e-۸۶	۸/۷e-۷۳
۲۰	الگوریتم ژنتیک	۱/۵۴e-۳	۰۰/۳۳۳۶	۰/۵۱۳۳
	PSO	۱/۹۷e-۴۱	۲/۳۱e-۳۶	۲/۸۴e-۳۳
	الگوریتم پیشنهادی	۲/۳۳e-۴۳	۱/۲۳e-۳۸	۳/۴e-۳۵
۳۰	الگوریتم ژنتیک	۱۱۹/۱۷۷۸	۱۴۳/۶۱۵۹	۱۵۹/۲۳۸۷
	PSO	۱/۱۲e-۳۵	۸/۱۸e-۳۰	۱/۱۹e-۲۵
	الگوریتم پیشنهادی	۱/۹۸e-۴۱	۴/۹۱e-۳۳	۶/۴۰e-۳۰

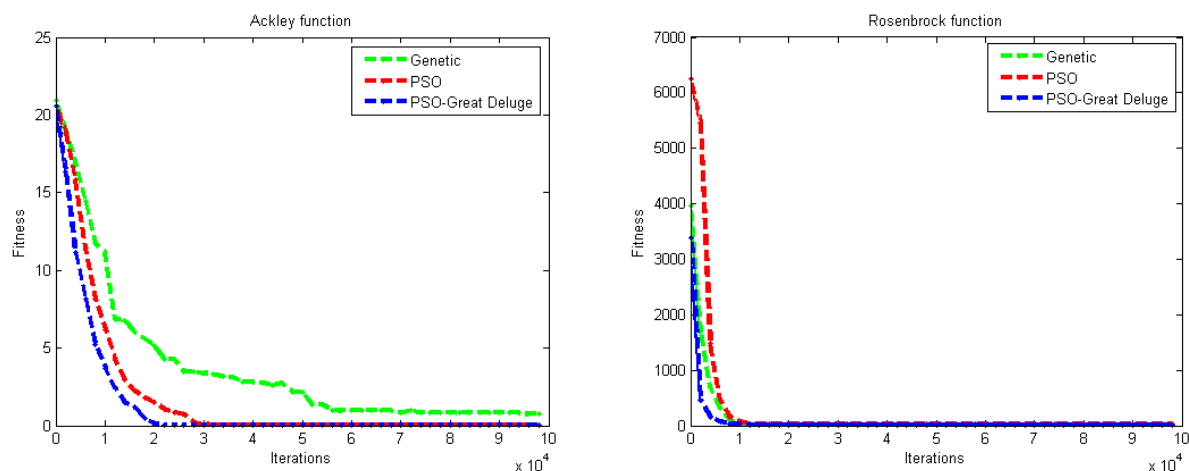
جدول (۵) نتایج الگوریتم پیشنهادی، PSO استاندارد و الگوریتم ژنتیک بر روی تابع استپ

ابعاد	الگوریتم	بهترین	متوسط	بدترین
۱۰	الگوریتم ژنتیک	-۶۰	-۵۸/۴۰۰۰	-۵۵
	PSO	-۶۰	-۶۰/۰۰۰۰	-۶۰
	الگوریتم پیشنهادی	-۶۰	-۶۰/۰۰۰۰	-۶۰
۲۰	الگوریتم ژنتیک	-۱۲۰	-۱۱۶/۳۲۰۰	-۱۱۲
	PSO	-۱۲۰	-۱۱۹/۵۰۰۰	-۱۱۸
	الگوریتم پیشنهادی	-۱۲۰	-۱۲۰	-۱۲۰
۳۰	الگوریتم ژنتیک	-۱۷۲	-۱۶۸	-۱۶۵
	PSO	-۱۷۸	-۱۷۰	-۱۶۷
	الگوریتم پیشنهادی	-۱۸۰	-۱۷۵	-۱۷۲

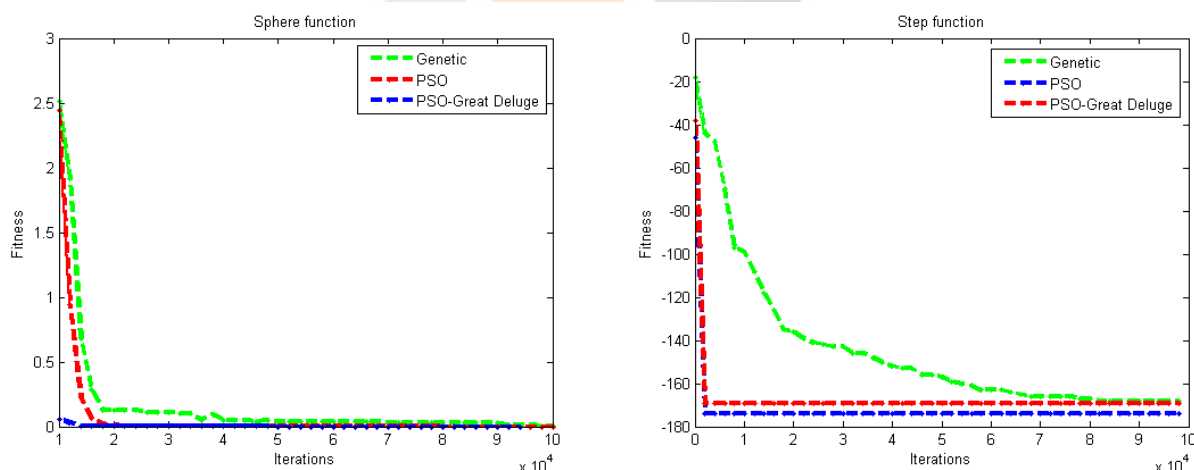
همانطور که در جداول (۲) تا (۵) مشاهده می‌کنید، عملکرد الگوریتم پیشنهادی در اکثر موارد نسبت به سایر الگوریتم‌ها بهتر می‌باشد. در الگوریتم پیشنهادی، چون راه‌حل نهایی توسط الگوریتم دلیوج با گام‌های کوچک به دست می‌آید بنابراین دقت این الگوریتم نسبت به سایر الگوریتم‌های مورد مقایسه قرار گرفته بالاتر رفته و در کل دارای کارایی بهتری نسبت به ۲ روش دیگر دارد.

در شکل‌های (۴) تا (۵)، نمودار مقدار تابع شایستگی را در طول ۱۰۰۰۰۰ تکرار از اجرای الگوریتم ژنتیک، PSO استاندارد و الگوریتم پیشنهادی برای توابع استاندارد آکلی، روزنبراک، اسفیر و استپ نشان داده شده‌اند. همانطور که مشاهده می‌کنید در تمامی موارد الگوریتم پیشنهادی هم زودتر از سایر الگوریتم‌ها همگرا شده و هم در پایان نتیجه بهتری نسبت به ۲ روش دیگر می‌دهد و این نتیجه به طور معمول دارای

پایداری لازم می‌باشد. این در حالی است که هر کدام از سایر روش‌ها در برخی مسائل خوب کار می‌کنند، اما در مسائل دیگر کارایی مطلوبی ارائه نمی‌کنند.



شکل (۴) نتایج اجرای PSO-Great Deluge، الگوریتم ژنتیک و PSO استاندارد در ۳۰ بار اجرا بر روی تابع آکلی و روزنبراک ۳۰ بعدی



شکل (۵) نتایج اجرای PSO-Great Deluge، الگوریتم ژنتیک و PSO استاندارد در ۳۰ بار اجرا بر روی تابع اسفیر و استپ ۳۰ بعدی

۶. نتیجه گیری

در این مقاله یک مدل ترکیبی جدید برای بهینه‌سازی ارائه شده است. هدف این مدل این بوده که با ترکیب خاصیت محلی بودن الگوریتم دلیوج و خاصیت سراسری الگوریتم بهینه‌سازی دسته ذرات به یک موازنه در جستجو برسیم. از هر دو خاصیت محلی و عمومی بودن الگوریتم‌ها به شکل خوب استفاده شده است، یعنی در ابتدا الگوریتم PSO با گام‌های بزرگ به جستجو پرداخته و بعد از به دست آوردن یک سری نتایج، نتایج جدید به عنوان ورودی به الگوریتم دلیوج سپرده می‌شود و در این مرحله این الگوریتم با گام‌های بسیار کوچک به جستجوی فضا می‌پردازد. نتایج شبیه‌سازی‌ها نشان داد که مدل ارائه شده جواب‌های بهتری در مقایسه با الگوریتم ژنتیک و PSO استاندارد تولید می‌نماید.

مراجع

- [۱] Y. Liu; K. M. Passino; "Swarm Intelligence: A Survey", In Proc. of ۴th International Conference of Swarm Intelligence, ۲۰۰۵.
- [۲] G. Dueck; "New optimization heuristics: The great deluge algorithm and the record-to-record travel", Journal of Computational Physics, vol. ۱۰۴, pp. ۸۶-۹۲, ۱۹۹۳.
- [۳] E. Burke; Y. Bykov; J. Newall; S. Petrovic; "A timepredefined approach to course timetabling", Yugoslav Journal of Operations Research, vol. ۱۳(۲), pp. ۱۳۹-۱۵۱, ۲۰۰۳.
- [۴] E. Burke; Y. Bykov; J. Newall; S. Petrovic; "A time-predefined local search approach to exam timetabling problems", IIE Trans, vol. ۳۶(۶), pp. ۵۰۹-۲۸, ۲۰۰۴.
- [۵] N. Nahas; D. Ait-Kadi; M. Nourelfath; "A new approach for buffer allocation in unreliable production lines", Int J Prod Econ, vol. ۱۰۳(۲), pp. ۷۳-۸۱, ۲۰۰۶.
- [۶] N. Nahas; M. Nourelfath; D. Ait-Kadi; "Coupling ant colony and the degraded ceiling algorithm for the redundancy allocation problem of series-parallel systems", Reliab Eng Syst Saf, vol. ۹۲(۲), pp. ۲۱۱-۲۲, ۲۰۰۷.

- [۷] N. Nahas; M. Nourelfath; D. Ait-Kadi; "A two-phase extended great deluge algorithm for the dynamic layout problem", In: International conference on industrial engineering and systems management, Beijing, China; ۲۰۰۷.
- [۸] A. Khatab; D. Ait-Kadi; A. Artiba; "Selective maintenance optimization for multi-mission series—parallel systems", Eur J Oper Res, Submitted, ۲۰۰۸.
- [۹] L. S. Dario; J. H. Obit; "Great Deluge with Non-linear Decay Rate for Solving Course Timetabling Problems", International IEEE Conference "Intelligent Systems", vol. ۴, pp. ۸-۱۸, ۲۰۰۸.
- [۱۰] J. Kennedy; Eberhart, R.C.; "Particle Swarm Optimization", Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. ۱۹۴۲-۱۹۴۸, ۱۹۹۵.



کنفرانس داده کاوی ایران