



بهبود الگوریتم بهینه سازی جستجوی گروهی با مکانیزم چند تولیدکننده و همیاری گروهی

Zahede Nazemi¹, Mohammad Reza Meybodi²

1 دانشکده برق، رایانه و فناوری اطلاعات، دانشگاه آزاد اسلامی، قزوین

zahede1218@gmail.com

2 دانشکده مهندسی کامپیوتر، فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران

mmeybodi@aut.ac.ir

چکیده:

بهینه سازی جستجوی گروهی، یک روش بهینه سازی بر مبنای هوش ازدحامی می باشد، که از تحقیق در رفتار حیوانات و تئوری زندگی گروهی الهام گرفته شده است. اگر چه رقابت در میان اعضای گروه ممکن است عملکرد آن را بهبود بخشد اما پیشرفت بیشتری از طریق همیاری می توان بدست آورد. از طرف دیگر این الگوریتم بر اساس مدل تولیدکننده تکاپو کننده می باشد. تولیدکننده برای یافتن، غذا تولید می کند. و تکاپوکننده ها فرسته هایی برای ملحاق شدن به تولیدکننده را جستجو می کنند. بهینه سازی جستجوی گروهی به عنوان یک روش موثر جهت حل مسائل بهینه سازی تابعی می باشد. اصولاً یافته ها در آن به صورت تولیدکننده تکی انجام می شود. در این تحقیق ما الگوریتم GSO را با استفاده از مکانیزم همیاری گروهی و چند تولیدکننده برای حل بهتر عملکرد مسائل بهینه سازی توسعه دادیم. رویکرد پیشنهادی به منظور افزایش سرعت همگرایی الگوریتم با ادغام دو الگوریتم MPGSO و CGSO جهت بهبود کارایی الگوریتم GSO پیشنهاد شده است.

کلمات کلیدی:

الگوریتم جستجوی گروهی ، چند تولیدکننده، همیاری گروهی، توابع بهینه سازی، هوش ازدحامی

Improved group search optimization algorithm with multi-producer mechanism and group collaboration

Zaheede Nazemi¹, Mohammad Reza Meybodi²

Faculty of Electrical, Computer and Information Technology, Islamic Azad University, Qazvin 1

Faculty of Computer Engineering, Information Technology, Amir Kabir University of Technology,
Tehran

Abstract:

Group search optimization is an optimization based on intelligence-driven intelligence, inspired by animal behavior research and group life theory. Although competition among members of the group may improve its performance, further improvement can be achieved through co-operation. On the other hand, this algorithm is based on the evolving producer model. The producer produces food to find. And the tweaks look for opportunities to join the manufacturer. Optimizing group search is an effective way to solve functional optimization problems. Basically, the findings are made in a single manufacturer. In this research, we developed the GSO algorithm using a group co-operation mechanism and several manufacturers to better solve the optimization problems. The proposed approach is proposed to increase the convergence speed of the algorithm by integrating two CGSO and MPGSO algorithms to improve the performance of the GSO algorithm.

key words:

Group search algorithm, multi-producer, group collaboration, optimization functions, invocation intelligence

پارامتر تنظیم شده مناسب، درصد رنجرهای تنواع بالای از مسائل بهینه سازی را می تواند حل کند و کارایی در مقابل دیگر الگوریتم ها تکاملی ۲ با سرعت همگرایی و دقت بالا مخصوصا در مورد مسائل چند مدلی با دقت بالا، کارایی قابل ملاحظه ای دارد. و همچنین نسبت به دیگر پارامترها مثل ماکسیمم زاویه ای تعییب حساس نیست. این ویژگی حل مسائل بهینه سازی را برای برنامه های کاربردی جهان واقعی جذاب می کند. کارامدترین روش جستجوی برای متابعی است که به صورت تصادفی توزیع شده است. و همچنین از رنجرهای جنبش های مسیر رندوم را (اجرا می کنند). برای محدود کردن فضای جستجو در این الگوریتم بکار گرفته شده است.

تاکنون نسخه های بهبود یافته مختلفی از الگوریتم جستجوی گروهی برای حل مسائل بهینه سازی حوزه ایستا ارائه شده است، از جمله می توان به یک روش جدید بیان شده در رابطه با الگوریتم بهینه سازی جستجوی گروهی اشاره به رفتار بیولوژیکی حیوانات دارد [2]. در تحقیقی دیگر پژوهشگران نسخه بهبود یافته الگوریتم جستجوی گروهی جهت جستجوی سریع این الگوریتم برای حل مسائل پخش بار در سیستم های قدرت ارائه دادند [12, 3]. در تحقیق [4]، پژوهشگران با بررسی دقیق الگوریتم جستجوی گروهی و ارائه مدل بهبود یافته آن با ترکیب کردن الگوریتم GSO با یک استراتژی تکاملی توансند مسائل طراحی قطعات مکانیکی رو به خوبی حل نمایند، در تحقیقی دیگر با ارائه مدل بهبود یافته الگوریتم بهینه سازی جستجوی گروهی با استفاده از الگوریتم اجتماع پرندگان کوانتومی در یک کاربرد خاص از آن استفاده کردند [5]. در تحقیق [7] با ارائه مدل هدایت راهبر ۳ مدل بهبود یافته الگوریتم GSO ارائه شده است. در تحقیق [8] نسخه سریع الگوریتم بهینه سازی جستجوی گروهی جهت حل مسئله طراحی بهینه لایه گردید استفاده شده است. در تحقیق [14] نسخه تصادفی الگوریتم GSO جهت آموزش شبکه های عصبی ارایه شده است. در تحقیق [13] نسخه ترکیبی الگوریتم بهینه سازی جستجوی گروهی به همراه الگوریتم بهینه سازی اجتماع ذرات جهت حل مسئله طراحی سازه های فضایی ارائه شده است. در تحقیق [15] نسخه ای از الگوریتم بهینه سازی

مقدمه

بهینه سازی جستجوی گروهی (GSO)، یک الگوریتم بهینه سازی جدید، که از تحقیق در رفتار حیوانات و تئوری زندگی گروهی الهام گرفته شده، ارائه شده است. این الگوریتم مبتنی بر مدل تولید کننده_ تکاپو کننده [1] می باشد. این مدل جستجوی اعضای گروه را یا به عنوان تولید کننده و یا تکاپو کننده در نظر می گیرد. به منظور توسعه ای الگوریتم از مکانیزم اسکن و پویش حیوانات (مثلاً بینایی حیوانات) استفاده شده است. همچنین از تکاورهایی استفاده شده است تا از به دام افتادن در کمینه های محلی جلوگیری شود. تولید کننده در صفر درجه اسکن می کند و سپس در خط افقی با نمونه گیری تصادفی سه نقطه در فیلد اسکن، عملیات اسکن را انجام می دهد. یک نقطه در صفر درجه ، و یک نقطه در گوشی ای راست ابرمکعب، یک نقطه در گوشی ای چپ ابر مکعب، تولید کننده سپس بهترین نقطه را با بهترین منبع بر اساس مقدار برازنده گی پیدا می کند. اگر بهترین نقطه منبع بهتری نسبت به موقعیت فعلی داشته باشد، تولید کننده به این نقطه انتقال داده خواهد شد. یا در موقعیت فعلی باقی خواهد ماند و جهت آن را به یک زاویه ای جدید انتقال خواهد داد. اگر تولید کننده نتواند یک زاویه ای بهتری را بعد از a تکرار بدمست آورد، جهت آن را به صفر درجه باز خواهد گرداند. در طول هر تکرار جستجوی، تعدادی از اعضای گروه به عنوان تکاپو کننده انتخاب می شوند. در k امین تکرار زاویه تصادفی Φ تولید خواهد شد. و سپس به صورت فاصله تصادفی انتخاب خواهد شد. و به نقطه جدیدی انتقال پیدا می کند. برای به حداقل رساندن شناس خود برای پیدا کردن منابع از چندین استراتژی برای محدودیت جستجوی خود به یک مسیر سودآور استفاده می کنند. یک استراتژی مهم که در حال گردش است داخل یک مسیری است که لبه هایش شناسایی شده است. این استراتژی به وسیله GSO به کار گرفته شده است. به فضای جستجوی محدود رسیدگی می کند. هنگامی که یک عضو خارج از فضای جستجو باشد در فضای جستجو به وسیله تنظیماتی که برای متغیرهای مرزی انجام می شود به عقب بر خواهد گشت.

این الگوریتم از نظر ادراکی ساده است و در اکثر موارد، یک

. Evolutionary Algorithms²

⁹. Levy

.Rangers¹



1) تولید کننده در زاویه صفر درجه پویش می کند. سپس به صورت افقی با سه نقطه ای که به صورت تصادفی انتخاب می شود، به عنوان فیلد پویش در نظر گرفته می شود. یک نقطه حتماً در زاویه صفر در درجه است.

(2-1)

$$X_z = X_p^k + r_1 L_{Max} D_p^k (\varphi^k)$$

(2-2) یک نقطه در سمت راست ابر مکعب

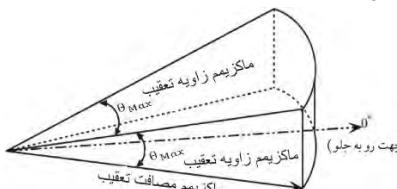
$$X_r = X_p^k + r_1 L_{Max} D_p^k (\varphi^k r_2 \theta_{Max}/2)$$

(2-3) و یک نقطه در سمت چپ ابر مکعب

$$= X_p^k + r_1 L_{Max} D_p^k (\varphi^k r_2 \theta_{Max}/2)$$

$X_i \in \mathbb{R}^{n-1}$ یک عدد تصادفی با توزیع نرمال $(0,1)$ است. r_1 که یک دنباله تصادفی با توزیع یکنواخت در محدوده $(0,1)$ است.

2) تولید کننده سپس فیلد را در بهترین نقطه و بهترین منبع (fitness value) پیدا خواهد کرد. اگر بهترین نقطه منبع ای بهتر از موقعیت فعلی خود دارد، سپس آن به این نقطه انتقال خواهد یافت.



شکل 1-2: فیلد پویش در فضای سه بعدی می باشد. و یا ماهی در موقعیت جاری می ماند و سرش به زاویه جدیدی که به صورت تصادفی تولید شده است می چرخد.

$$\varphi^{k+1} = \varphi^k + r_2 \alpha_{Max} \quad (2-4)$$

در حالی که $\alpha_{Max} \in \mathbb{R}^1$ مکریم زاویه چرخش است. (3) اگر تولید کننده نمی تواند زاویه بهتری را بعد از تکرار پیدا کند، ماهی سرش به پشت با زاویه صفر درجه می چرخد.

$$\varphi^{k+\alpha} = \varphi^k \quad (2-5)$$

در حالی که $\alpha \in \mathbb{R}^1$ و یک ثابت است. در طول هر جستجو تعدادی از اعضای گروه به عنوان تکاپو کننده انتخاب می شوند. تکاپوکننده جستجو را برای فرصت ای برای پیوستن به منابع پیدا شده به وسیله تولید کننده حفظ خواهد کرد. استراتژی تکاپوکننده در خانه گنجشک: 1) کپی ناحیه: حرکت سرتاسری ناحیه چرخش اطراف ناحیه پهلوی

جستجوی گروهی جهت خوش بندی داده های خاص استفاده شده است. در تحقیق [10] یک ساختار جدید اینترنت اشیا بر مبنای الگوریتم بهینه سازی جستجوی گروهی ارائه شده است. در [9] نوع جدید از الگوریتم بهینه سازی جستجوی گروهی بر مبنای مدل توسعه یافته تولید کننده-تکاپوکننده [1] ارائه شده است. در [11] الگوریتم GSO جهت حل مسئله مدیریت موقعیت شبکه موبایل استفاده شده است. در تحقیق [16] نسخه ای اجتماعی الگوریتم GSO در شبکه Ad Hoc استفاده شده است. در مقالات [17-19] نسخه بهینه یافته چند الگوریتم جستجوی گروهی جهت حل مسائل مختلفی همانند تشخیص ارتباط در شبکه و همچنین بهینه کردن راکتیو و ... استفاده شده است.

1- مفاهیم اولیه

بهینه سازی جستجوی گروهی GSO

جمعیت در الگوریتم GSO را گروهⁿ و هر فرد در جمعیت را عضو⁰ می نامیم. در یک فضای جستجوی n -تکرار k -ام جستجوی، یک موقعیت جاری $X_i^N \in \mathbb{R}^N$ ، یک زاویه ری راس $\varphi_i^k = (\varphi_{i1}^k, \dots, \varphi_{i(n-1)}^k) \in \mathbb{R}^{n-1}$ و یک جهت راس $\varphi_i^k = (d_{i1}^k, \dots, d_{in}^k) \in \mathbb{R}^n D_i^k$ که از φ_i^k با یک تبدیل قطبی به مختصات کارتزین بدست می آید:

$$\begin{aligned} &= \prod_{q=1}^{n-1} \cos(\varphi_{iq}^k) d_{i1}^k \\ &= \varphi_{i1}^k \sin(\varphi_{i(j-1)}^k) \prod_{q=j}^{n-1} \cos(\varphi_{iq}^k) (j = 2 \dots n-1) \\ &\varphi_{in}^k = \sin(\varphi_{i(n-1)}^k) \end{aligned}$$

در GSO، یک گروه شامل سه عضو است: تولید کننده و تکاپوکننده که رفتارشان مبتنی بر مدل PS است و رنجرها که جنبش های مسیر رندوم را اجرا می کنند. در این الگوریتم فیلد اسکن ساده است. و به فضای n بعدی تعمیم داده می شود. که ویژگی مکریم زاویه تعقیب $\theta_{Max} \in \mathbb{R}^1$ و حداقل فاصله تعقیب $L_{Max} \in \mathbb{R}^1$ به عنوان فضای سه بعدی در شکل 1 شرح داده می شود. که نوک آن موقعیت تولید کننده است. در الگوریتم GSO در k امین تکرار تولید کننده X_p رفتارش به شرح زیر می باشد.

.Group⁴
.member⁵



باشد. وهمچنین استفاده از الگوریتم MCGSO[21] که در این الگوریتم از چند تولید کننده به جای استفاده از تولید کننده تکی استفاده می شود. که برای توابع محک با اعداد بزرگ در مقایسه با GSO عملکرد بهتری داشته است. در الگوریتم ارائه شده اعضای گروه به k گروه وابسته تقسیم می شوند. در هر گروه d بعد به صورت مشترک (در حالی که $k=n$) است. هر گروه با اجرای جستجوی محلی و مقدار کمینه در خود را مشاهده می کند. در GSO استاندارد سه نوع اعضای گروه وجود دارد. G_j : گروه تولید کننده $G_j \cdot X_p$ و گروه تکاپوکننده⁸ و گروه اعضای مجزا⁹ (محدود کننده). در این رویکرد هر گروه تکاپوکننده بهترین عضو از گروه را دنبال می کند. تولید کننده $G_j \cdot X_p$ از هر گروه است. در الگوریتم پیشنهادی به جای یک تولید کننده از چند تولید کننده استفاده کردیم بنابراین ما 10 درصد از اعضای گروه را در هر گروه به عنوان تولید کننده انتخاب می کنیم و با چرخ گردان¹⁰ به صورت تصادفی به هر تولید کننده نسبت می دهیم.

FOR (each group G_j)

Random generate the initial members;
Evaluate the fitness values of initial members;

END FOR

WHILE (the termination conditions are not met)

FOR (each group G_j)

Choose producer:

- Find the top $x\%$ members as producers of the group.
- Find the producer $G_j \cdot X_p$ of the Group
- Now for each of the producers do following:
- Scan at zero degree and then scan laterally by randomly sampling three points in the scanning field using (2-1) to (2-3).
- If the best point found by scanning is better than the current position, then it will fly to the new position.
- Otherwise stay at current position and turn head to a new angle using (2-4).

Perform scrounging:

- : Randomly select a percentage from the rest members to perform scrounging; All scroungers will follow its group best member $G_j \cdot X_p$.
- Create roulette wheel using (3-4-1)
- For each of the scrounger, select a producer from roulette wheel, and then perform scrounging using (2-6).

Perform dispersion:

- Select rest of the members as dispersed.

-
- . Producer⁸
 - . Scrounger⁹
 - . Ranger¹⁰
 - . Roulette Wheel¹¹

تولید کننده² به شکل زیر است: حیوانات دیگر در اطراف بدون نشانه جستجوگر مشاهده می شوند. ³ بیرون کشیدن: مصرف منابع به طور مستقیم از تولید کننده است. در الگوریتم GSO تنها منطقه کپی کردن است که شایع ترین رفتار تکاپوکننده در گنجشک اتخاذ شده است. در k امین تکرار، رفتار منطقه کپی از i امین تکاپو کننده می تواند به عنوان راه رفتن به سمت تولید کننده مدل شود.

$$X_i^{k+1} = X_i^k + r_3(X_p^k - X_i^k) \quad (2-6)$$

در حالی که $r_3 \in \mathbb{R}^1$ توالی تصادفی تکی در رنج (0,1) است.

در k امین تکرار زاویه تصادفی φ در (2-4) تولید خواهد شد. و سپس آن به صورت فاصله تصادفی انتخاب خواهد شد.

$$L_i = \alpha \cdot r L_{Max} \quad (2-7)$$

و به نقطه جدیدی انتقال پیدا می کند.

$$+ L_i D_i^k (\varphi^{k+1}) \quad (2-8)$$

$$X_i^{k+1} = X_i^k$$

ادامه این مقاله بدین ترتیب سازماندهی شده است. در بخش سوم به تشریح الگوریتم پیشنهادی می پردازیم. در بخش چهارم نتایج آزمایشات مورد بررسی قرار می گیرند و در بخش آخر به بیان نتیجه گیری می پردازیم.

به طور کلی چالش های پیش رو در الگوریتم های تکاملی شامل فرار از مینیمم محلی، سرعت بالای همگرایی الگوریتم، قابلیت جستجو⁶ و استخراج⁷ همگرایی پیش از موعد (همگرایی پیش از بلوغ) در الگوریتم های تکاملی می باشد. در این مقاله به دنبال حل چالش بالا بردن سرعت همگرایی الگوریتم هستیم.

2- رهیافت پیشنهادی

(MCGSO 1-3 چند تولید کننده، همیاری کننده
پیشنهاد شده:

در این بخش ترکیب دو الگوریتم گروه جستجوی بهینه سازی همیاری کننده [20] CGSO و گروه جستجوی چند تولید کننده ارائه شد. این توسعه براساس مدل همیاری کننده به شکل استراتژی تقسیم و غلبه، در حالی که هر گروه مسئول محدوده ای عددی از خصوصیات فضای جستجو هستند می

-
- 12. Exploration
 - 11. Exploration

همه برنامه ها در Matlab 2014 اجرا شده اند. 9 تابع محک به شکل زیر است.

جدول(1) مشخصات محدوده ی عملکرد

تابع	محدوده	f_{min}
$f_1 = \text{Sphere}$	$[-100, 100]^n$	0
$f_2 = \text{Griewank}$	$[-600, 600]^n$	0
$f_3 = \text{Rastrigin}$	$[-5.12, 5.12]^n$	0
$f_4 = \text{Ackley}$	$[-32, 32]^n$	0
$f_5 = \text{Lévi function}$	$[-10, 10]^n$	0
$f_6 = \text{Goldstein-P}$	$[-2, 2]^n$	0
$f_7 = \text{Booth}$	$[-10, 10]^n$	0
$f_8 = \text{Rosenbrock}$	$[-30, 30]^n$	0
$f_9 = \text{Matyas}$	$[-10, 10]^n$	0

جدول(2): فرمول ها

Formula
$f_1(x) = \sum_{i=1}^n x_i^2$
$f_3(x) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)^2$
$f_3(x) = \frac{1}{400} \sum_{i=1}^n (x_i - 100)^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$
$f_4(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos 2\pi x_i\right) + 20 + e$
$f_5(x) = \sin^2 3\pi x + (x - 1)^2$ $(1 + \sin^2 3\pi y) + (y - 1)^2 (1 + \sin^2 2\pi y)$
$f_6(x, y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2$ $+ (2.625 - x + xy^3)^2$
$f_7(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$
$f_8(x) = \sum_{i=1}^{n-1} [(100(x_{i+1} - x_i^2)) + (x_i - 1)^2]$
$f_9(x, y) = 0.29(x^2 + y^2) - 0.48xy$

جدول(3) پارامتر های ثابت برای همه الگوریتم ها

الگوریتم	پارامتر	مقدار
	درصد تکاپو کننده	80%
	θ_{max}	π/a^2

i. Generate a random head angle using (2-4).
 iii. Choose a random distance l_i
 3.d. Calculate the fitness value of current members:
 $f(X)$
 using (2-7) and move to the new point using (2-8).

// End While

// End For

شکل (3-3) شبیه کد الگوریتم MCGSO را نشان می دهد.

4- تنظیمات نتایج:

این بخش ابتدا توصیفی از توابع محک و تنظیمات نتایج ای که بر روی MCGSO آزمایش شده است نشان می دهد. یک مقایسه ای مابین الگوریتم های MPGSO، MCGSO، CGSO، برای حل توابع بهینه سازی انجام شده است.

4-2 توابع محک و تنظیمات نتایج:

در این مقاله دنباله ای از چهار تابع محک در نظر گرفته شده است. نتایج با 30 بعد در فضای جستجو ارائه شده اند. معیار های ارزیابی از یک تحلیل تجربی، یک سیستم رتبه بندی [22] و یک فرضیه آزمایش شده از آزمون t با مقدار پارامتر $\alpha = 0.05$ انجام شده است. نتایج نقاط تحلیل شده متده با بهترین ارزش برای توابع اخیر است. سیستم رتبه بندی بر طبق متده میانگین کارایی مرتب شده است. مقادیر محدوده فضای جستجو و کمترین مقدار برای هر تابع بر طبق جدول (1-4) [23] آورده شده است. در نتایج ما بعضی از پارامتر ها با مقادیری که در جدول (2-4) آمده است ثابت شده است. حداکثر فاصله تعقیب l_{max} به شکل زیر داده شده بود:

(4-2)

$$l_{max} = \|U - L\| \sqrt{\sum_{i=1}^n (U_i - L_i)^2}$$

که L_i حد پایین و U_i حد بالا برای i امین بعد هستند. مقدار حداکثر ارزیابی های تابع (شرایط خاتمه) مجاز 50000 و تعداد آزمایش شده 35 می باشد. نتایج برای همه توابع در جدول (2-4) نشان داده شده است. در جداول نتایج به ترتیب نقطه میانه، میانگین، بهترین مقدار و بدترین مقدار در 35 بار تکرار می باشد. بهترین مقادیر نشان داده شده در نتایج تحلیلی به صورت پرنگ نشان داده شده اند. تنوع در میانگین بهترین مقدار برازنده گی برای هر روش بربطیق تعداد توابع ارزیابی بیش از 35 بار تکرار است.



DE	Median	9.770×10^{-2}	3
	Best	5.558×10^{-8}	
	Worst	1.126	
MCGSO	Mean	1.756×10^{-1}	1
	Median	7.650×10^{-2}	
	Best	3.558×10^{-8}	
	Worst	1.119×10^{-2}	

GSO	α_{max}	$\theta_{max}/2$
	φ_0	$\pi/4$
CGSO	k	5
	MPGSO	درصد تولید کننده
	رویکردهای	اندازه جمعیت
	کلی	حداکثر تعداد ارزیابی
	NFE	5×10^4
	تابع	

جدول(3-3) نتایج تابع f3

روش	متريک	مقدار	رتبه بندي
GSO	Std	3.9234	4
	Median	1.094×10^1	
	Best	4.0305	
	Worst	2.127×10^1	
PSO	Std	1.456	2
	Median	4.394	
	Best	3.0305	
	Worst	1.121×10^1	
DE	Std	1.747	3
	Median	4.494	
	Best	1.992	
	Worst	1.021×10^1	
MCGSO	Std	1.356	1
	Median	3.194	
	Best	1.845	
	Worst	1.009×10^1	

جدول(1-1) نتایج تابع f1

روش	متريک	مقدار	رتبه بندي
GSO	Std	3.820×10^{-4}	4
	Median	8.475×10^{-5}	
	Best	1.510×10^{-5}	
	Worst	2.068×10^{-3}	
PSO	Std	2.128×10^{-5}	2
	Median	7.301×10^{-7}	
	Best	7.402×10^{-8}	
	Worst	8.010×10^{-4}	
DE	Std	1.773×10^{-5}	3
	Median	6.407×10^{-8}	
	Best	8.512×10^{-11}	
	Worst	9.010×10^{-5}	
MCGSO	Std	1.128×10^{-5}	1
	Median	5.231×10^{-9}	
	Best	7.632×10^{-9}	
	Worst	8.210×10^{-4}	

جدول(3-2) نتایج تابع f2

روش	متريک	مقدار	رتبه بندي
GSO	Std	4.323×10^{-2}	4
	Median	7.446×10^{-3}	
	Best	1.331×10^{-3}	
	Worst	2.354×10^{-1}	
PSO	Mean	3.801×10^{-4}	2
	Median	3.008×10^{-4}	
	Best	1.848×10^{-5}	

روش	متريک	مقدار	رتبه بندي
GSO	Std	2.865×10^{-2}	4
	Median	3.0414×10^{-2}	
	Best	1.130×10^{-4}	
	Worst	1.429×10^{-1}	
PSO	Mean	1.956×10^{-1}	2
	Median	8.650×10^{-2}	
	Best	4.558×10^{-8}	
	Worst	1.229×10^{-1}	
	Mean	1.876×10^{-1}	



PSO	Best	1.572×10^{-10}							
	Worst	1.292×10^{-4}							
DE	Mean	3.195×10^{-7}	3						
	Median	4.154×10^{-10}							
	Best	1.572×10^{-11}							
	Worst	1.892×10^{-5}							
MCGSO	Mean	3.088×10^{-7}	1						
	Median	3.345×10^{-8}							
	Best	1.172×10^{-10}							
	Worst	1.192×10^{-4}							

جدول(5-3) نتایج تابع f5

جدول(7-3) نتایج تابع f7

روش	متريک	مقدار	رتبه ه بندی				
				Std	Median	Best	Worst
GSO	Std	7.515×10^{-1}	4				
	Median	3.405					
	Best	1.682					
	Worst	4.692					
PSO	Mean	5.658×10^{-1}	2				
	Median	2.436					
	Best	7.678×10^{-1}					
	Worst	3.108					
DE	Mean	4.594×10^{-1}	3				
	Median	1.645					
	Best	6.176×10^{-1}					
	Worst	2.508					
MCGSO	Mean	4.128×10^{-1}	1				
	Median	1.436					
	Best	1.555					
	Worst	1.302					

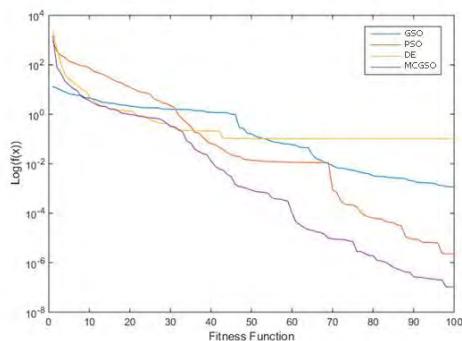
جدول(6-3) نتایج تابع f6

جدول(8-3) نتایج تابع f8

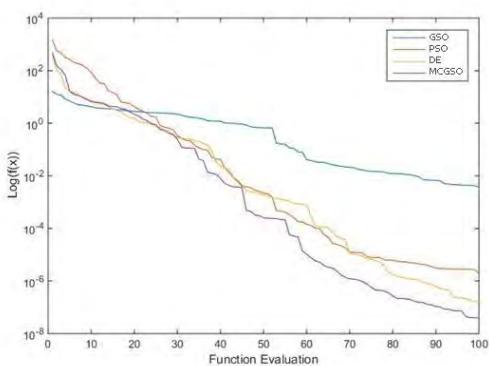
روش	متريک	مقدار	رتبه ه بندی				
				Std	Median	Best	Worst
GSO	Std	5.176×10^{-3}	4				
	Median	1.819×10^{-5}					
	Best	2.240×10^{-6}					
	Worst	1.107×10^{-2}					
PSO	Mean	4.901×10^{-3}	2				
	Median	1.176×10^{-8}					
	Best	4.160×10^{-11}					
	Worst	1.079×10^{-2}					
DE	Mean	3.901×10^{-3}	3				
	Median	1.376×10^{-8}					
	Best	4.360×10^{-11}					
	Worst	1.099×10^{-2}					
MCGSO	Mean	3.121×10^{-3}	1				
	Median	1.045×10^{-8}					
	Best	2.160×10^{-11}					
	Worst	1.002×10^{-2}					

روش	متريک	مقدار	رتبه ه بندی				
				Std	Median	Best	Worst
GSO	Std	3.528×10^1	4				
	Median	8.738×10^1					
	Best	2.017×10^1					
	Worst	1.462×10^2					
	Mean	2.398×10^2					

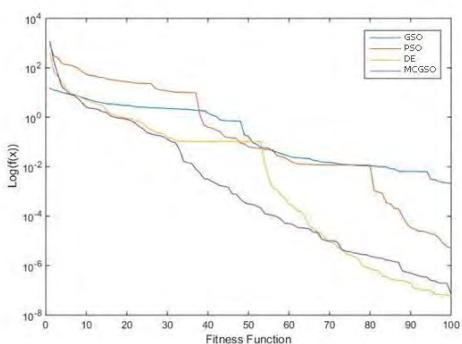
روش	متريک	مقدار	رتبه ه بندی				
				Std	Median	Best	Worst
GSO	Std	2.344×10^{-5}	4				
	Median	1.217×10^{-6}					
	Best	1.176×10^{-7}					
	Worst	1.349×10^{-4}					
	Mean	4.324×10^{-7}					
	Median	4.154×10^{-8}					



شکل(3): میانگین بهترین مقدار تابع f3



شکل(4): میانگین بهترین مقدار تابع f4



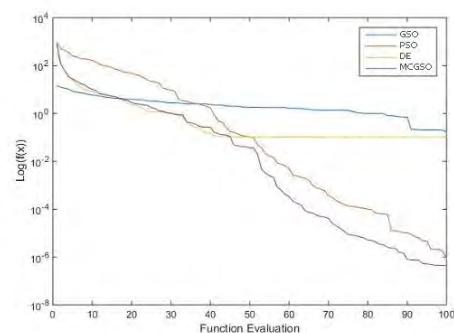
شکل(5): میانگین بهترین مقدار تابع f5

PSO	Median	7.993×10^{-1}	2
	Best	2.128×10^3	
	Worst	1.128×10^3	
DE	Mean	3.398×10^2	3
	Median	7.993×10^1	
	Best	3.228	
	Worst	1.728×10^3	
MCGSO	Mean	2.198×10^2	1
	Median	7.293×10^{-1}	
	Best	2.328×10^3	
	Worst	1.228×10^3	

جدول(3-9) نتایج تابع f9

روش	متريک	مقدار	رتبه
			بندی
GSO	Std	3.266×10^{-3}	4
	Median	2.176×10^{-3}	
	Best	5.226×10^{-4}	
	Worst	1.322×10^{-2}	
PSO	Mean	3.477×10^{-4}	2
	Median	4.114×10^{-5}	
	Best	3.770×10^{-6}	
	Worst	1.343×10^{-3}	
DE	Mean	2.485×10^{-4}	3
	Median	3.012×10^{-5}	
	Best	3.873×10^{-6}	
	Worst	1.443×10^{-3}	
MCGSO	Mean	2.123×10^{-4}	1
	Median	3.423×10^{-5}	
	Best	3.373×10^{-7}	
	Worst	1.222×10^{-3}	

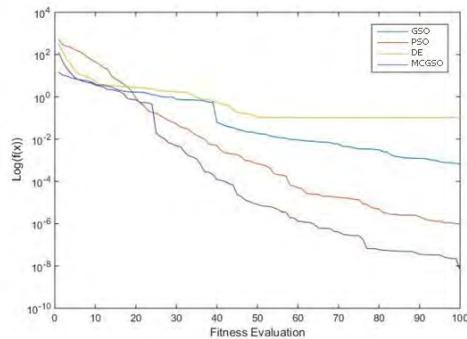
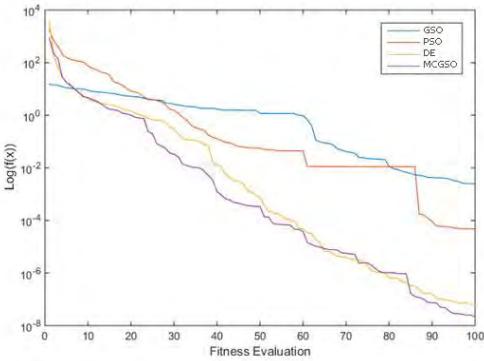
شکل(1): میانگین بهترین مقدار تابع f1



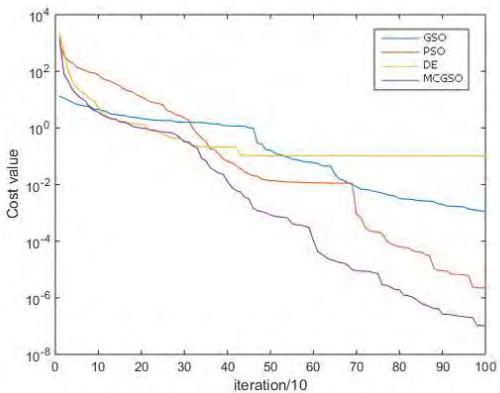
شکل(2): میانگین بهترین مقدار تابع f2



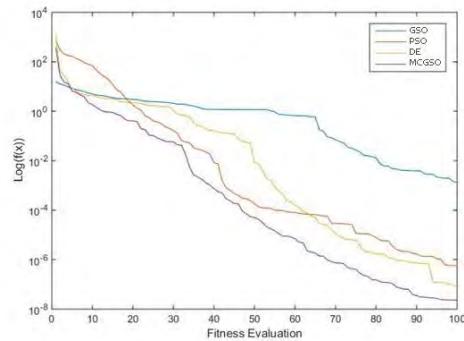
شکل(8): میانگین بهترین مقدار تابع f8



شکل(9): میانگین بهترین مقدار تابع f9



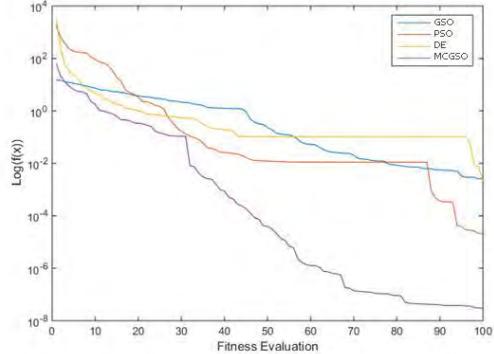
شکل(6): میانگین بهترین مقدار تابع 6



5- نتیجه گیری:

در این مقاله به ایجاد تغییراتی در بهینه سازی جستجوی گروهی پرداخته شد. در تغییر اول، استفاده از روش چند تولیدکننده بجای استفاده از تولیدکننده تکی و تغییر بعدی استفاده از روش همیاری گروهی نتایج آزمایش ها بر روی مسائل محک معروف در بهینه سازی ایستانا نشان می دهد که الگوریتم جستجوی گروهی تولیدکننده همیاری گروهی می تواند عملکرد بهتری در برخی از توابع محک نسبت به الگوریتم بهینه سازی جستجوی گروهی تولیدکننده و جستجوی گروهی همیاری گروهی داشته باشد. آزمایشهای ما نشان داد که با ترکیب بهترین راه حل های پیدا شده در هر گروه و روش چند

شکل(7): میانگین بهترین مقدار تابع 7





- things group search optimizer." International Journal of Communication Systems, Vol. 29, No.3, pp. 535-552.
- [11] D.Wang, X.Congcong and W.Huang. (2014) "Group search optimizer for the mobile location management problem." The Scientific World Journal.
- [12] F.Chishti and A.K.Gangwar. (2015) "Group Search Optimizer for Economic Load Dispatch." pp. 92.
- [13] S.K.Zeng, and L.J.Li. (2014) "Particle swarm-group search algorithm and its application to spatial structural design with discrete variables." Int J Optim Civil Eng, Vol. 2, No. 4, pp. 443-58.
- [14] S.He. (2010) "Training Artificial Neural Networks Using Lévy Group Search Optimizer." Multiple-Valued Logic and Soft Computing, Vol. 16, No. 6, pp. 527-545.
- [15] L.D. Santos, and T.B. Ludermir. (2014) "A Group Search Optimization Method for Data Clustering." Intelligent Systems (BRACIS), Brazilian Conference on, IEEE.
- [16] X.Feng, et al. (2015) "Social Group Search Optimizer Algorithm for Ad Hoc Network." Ad Hoc & Sensor Wireless Networks, Vol. 28.
- [17] D.Arora and B.Hema. (2016) "Multi objective Group Search Optimization Approach for Community Detection in networks." International Journal of Applied Evolutionary Computation, Vol. 7, No. 3, pp.50-70.
- [18] N.Daryani, B.Ebrahim, and A. R.Shamlou. (2016) "Improved Group Search Optimization Algorithm for Multi-Objective Optimal Reactive Power Dispatch." Majlesi Journal of Electrical Engineering, Vol. 10, No. 4.
- [19] D.Magatralo, et al. (2013) "Simultaneous gene selection and cancer classification using a hybrid group search optimizer." Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation.
- [20] D.N.G. Silva, L.D.S.Pacifico and T.B.Ludermir. (2013) "Cooperative Group Search Optimization" IEEE Transactions on Evolutionary Computation.
- [21] Q.Kang, T.Lan, Y.Yan, J.and L.Wang. (2013) "Multi-Producer Group Search Optimizer for Function Optimization" IEEE Transactions on Evolutionary Computation.
- [22] T. Richer and T. Blackwell. (2006) The L'evy particle swarm. In: Proc. of the Congress on Evolutionary Computation, pp. 808– 815, IEEE Press.
- [23] He, Shan, Q.Henry Wu, and J.R.Saunders. (2009) "Group search optimizer: an optimization algorithm inspired by animal searching behavior." IEEE transactions on evolutionary computation.

تولیدکننده ترکیب بهتری را نسبت به الگوریتم GSO استاندارد و MPGSO و CGSO به دست خواهیم آورد. نتایج نشان می دهد که به طور قابل ملاحظه ای یادگیری، استراتژی الگوریتم ها همیاری گروهی اعضای جمعیت را می تواند بهبود دهد و همچنین استفاده از روش چند تولیدکننده بجای استفاده از تولیدکننده تکی در توابع محک با مقادیر زیاد می تواند مفید واقع شود.

منابع

- [1] A. Sadollah, et al. (2013) "Mine blast algorithm: A new population-based algorithm for solving constrained engineering optimization problems." Applied Soft Computing, Vol. 13, No .5, pp. 2592-2612.
- [2] S.He, Q.H.Wu, and J.R. Saunders. (2006) "A novel group search optimizer inspired by animal behavioral ecology." Evolutionary Computation, IEEE Congress on.
- [3] H.R. Abdolmohammadi, A.Kazemi, et al. (2012) "Continuous quick group search optimizer for solving non-convex economic dispatch problems." Electric Power Systems Research, Vol. 93, pp. 93-105.
- [4] H.Shen, et al. (2009) "An improved group search optimizer for mechanical design optimization problems." Progress in Natural Science, Vol .19, No .1, pp. 91-97.
- [5] D.Chen, et al. (2012) "An improved group search optimizer with operation of quantum-behaved swarm and its application." Applied Soft Computing, Vol.12, No .2, pp.712-725.
- [6] S.He, Q.H. Wu, and J.R. Saunders. (2006) "A novel group search optimizer inspired by animal behavioral ecology." Evolutionary Computation, IEEE Congress on.
- [7] S.Ghosh, K.Nandi, (2015) "Gbest-guided Group Search Optimizer Algorithm, Computer Technology & Applications", Vol 6 (3), pp. 535-542.
- [8] Q.Guang, L.Feng, and L.Lijuan. (2010) "A quick group search optimizer and its application to the optimal design of double layer grid shells." AIP Conference Proceedings. Eds. Jane WZ Lu, et al. Vol. 1233, No. 1.
- [9] J.Chen, et al. (2016) "A variant of group search optimizer for global optimization." Journal of Computational Methods in Sciences and Engineering, Vol. 6, No. 2, pp. 219-230.
- [10] X.Feng, X.Liu, and H.Yu. (2016) "A new internet of