

تعیین موقعیت و تعقیب چندین بهینه در محیط های پویا با استفاده از الگوریتم جهش قورباغه

سمیه رنجکش¹، محمدرضا میبدی²، بهروز معصومی³

¹ عضو هیئت علمی گروه کامپیوتر دانشگاه آزاد اسلامی واحد رودسر، s.ranjekesh2@yahoo.com

² عضو هیئت علمی دانشکده مهندسی کامپیوتر، فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، mmeybodi@aut.ac.ir

³ عضو هیئت علمی دانشکده برق-رایانه دانشگاه آزاد اسلامی واحد قزوین، masoumi_b@yahoo.com

چکیده - در بسیاری از مسائل بهینه سازی دنیای واقعی، تابع هدف، یا محدودیت ها می توانند در طول زمان تغییر یابند و بنابراین بهینه این مسائل نیز می تواند تغییر یابد. اگر هر یک از این رویدادهای نامعین در فرآیند بهینه سازی مورد توجه قرار گیرند، این مسأله دینامیک یا پویا نامیده می شود. در این گونه مسائل علاوه بر پیدا کردن بهینه سراسری می بایست آن را در طول زمان دنبال نمود. در این مقاله یک روش نوین برای تعیین موقعیت و تعقیب چندین بهینه در محیط پویا مبتنی بر الگوریتم جهش قورباغه پیشنهاد شده است. نتایج حاصل از روش پیشنهادی بر روی معیار قله های متحرک که در حال حاضر شناخته شده ترین معیار برای ارزیابی در محیط های پویا می باشد و همچنین با استفاده از خطای برون خطی ارزیابی شده است. نتایج بدست آمده نشان دهنده کارایی بالای روش پیشنهادی می باشد.

کلید واژه- الگوریتم جهش قورباغه، همگرایی، محیط پویا

1- مقدمه

یک روش جدید بر پایه الگوریتم جهش قورباغه استاندارد [1] برای بهینه سازی در محیط های پویا ارائه می گردد. در این الگوریتم، روند اجرای الگوریتم جهش قورباغه استاندارد تغییر کرده است تا الگوریتم پیشنهادی بتواند چندین قله را پیدا کند و بتواند آنها را پس از تغییر محیط دنبال کند [2]. روش پیشنهادی بر روی بنچمارک قله های متحرک (MPB) [3] که از معروفترین بنچمارک های محیط های پویا است مقایسه شده است. معیار مقایسه، خطای برون خطی است که یکی از معیارهای اصلی مقایسه الگوریتم های طراحی شده برای محیط های پویا می باشد [4]. نتایج آزمایشات نشان می دهد که روش پیشنهادی از کارایی قابل قبولی برخوردار است.

ادامه این مقاله بدین ترتیب سازماندهی شده است. در بخش دوم الگوریتم ترکیبی جهش قورباغه در محیط های پویا معرفی می شود و در بخش سوم به تشریح روش پیشنهادی

امروزه بسیاری از مسائل از نوع مسائل پویا بوده و مقدار مولفه ها و محدودیت ها با گذشت زمان تغییر می نماید. محیط های پویا دارای کاربردهای فراوانی در مسائل متفاوت می باشد. به دلیل اهمیت این نوع از مسائل در این بخش به معرفی راه کارهایی خواهیم پرداخت که تا به حال برای حل این نوع از مسائل ارائه شده اند. برای بهینه سازی در چنین محیط های پویایی الگوریتم باید به سرعت بتواند همه قله ها را بیابد، آنها تحت پوشش قرار دهد، و تمام آنها را در طول فرآیند بهینه سازی تعقیب کند. یکی از معروفترین مکانیزم ها برای رفع چالش وجود بهینه های بالقوه مکانیزم چند دستگی است. در این مکانیزم چندین الگوریتم پایه به طور موازی در حال اجرا هستند و بین آنها با توجه به ساختار کلی الگوریتم بهینه سازی پویا ارتباطات و محدودیت هایی برقرار است. وظیفه هر یک از این الگوریتم ها پوشش دادن و تعقیب کردن یک قله در فضای مسئله می باشد. در ادامه

می باشد. سپس یک موقعیت کاندید با استفاده از روابط (1) و (2) بدست می آید:

$$D = Rand.(X_b - X_w) \quad (1)$$

$$X'_w = X_w + D, \quad D_{min} \leq D \leq D_{max} \quad (2)$$

rand یک تابع تولید عدد تصادفی با توزیع یکنواخت در بازه [0,1] می باشد و برابر با طول گام جابه جایی است که مقدار آن در هر یک از ابعاد مسئله باید در محدوده $[D_{min}, D_{max}]$ باشد.

پس از بدست آوردن موقعیت، مقدار شایستگی آن سنجیده می شود و در صورتی که مقدار شایستگی آن بهتر از موقعیت قبلی باشد، X_w به موقعیت جدید خود حرکت می کند در غیر این صورت روابط (1) و (2) مجدداً اجرا می شوند با این تفاوت که به جای موقعیت X_b ، از موقعیت بهترین قورباغه در میان تمام ممپلکس ها یعنی X_g استفاده می شود.

هنگامی که یک قورباغه نتواند موقعیتش را با دنبال روی از بهترین قورباغه ممپلکس و پس از آن بهترین قورباغه دسته بهبود دهد، به طور تصادفی در فضای مسئله موقعیت دهی می شود که این مسئله برخلاف شرایط الگوریتم های بهینه سازی در محیط های پویا است. در واقع در محیط های پویا، هر الگوریتم پایه وظیفه دارد یک قله را پوشش دهد و آنرا دنبال کند اما با انجام عمل مقدار دهی در فضای مسئله ممکن است، الگوریتم پایه قله تحت پوشش خودش را رها و به دنبال یک قله بهتر حرکت کند. برای رفع این مشکل، الگوریتم را بدین ترتیب تغییر می دهیم که بردار D و موقعیت قورباغه نام به جای موقعیت دهی تصادفی در فضای مسئله، با استفاده از روابط زیر محاسبه شود:

می پردازیم. در بخش چهارم نتایج آزمایشات مورد بررسی قرار میگیرند و بخش آخر به بیان نتیجه گیری می پردازد.

2- الگوریتم ترکیبی جهش قورباغه در محیط های پویا

این الگوریتم بر اساس قوانین احتمالی، جستجوی تصادفی و جمعیت کار می کند. روند اجرای این الگوریتم برگرفته شده از رفتار قورباغه ها در تالاب ها برای یافتن غذا می باشد. هر قورباغه نشان دهنده یک راه حل شدنی در فضای مسئله می باشد. موقعیت قورباغه نام در فضای D بعدی برابر $X_i = (X_{i,1}, X_{i,2}, \dots, X_{i,D})$ می باشد و مقدار غذای موجود در موقعیت این قورباغه که همان مقدار شایستگی آن می باشد برابر $f(X_i)$ می باشد. موقعیت هایی که دارای شایستگی بیشتری هستند از مقدار غذای بیشتری برخوردارند و هدف قورباغه ها یافتن غذای بیشتر است.

بدین ترتیب قورباغه ها با اجرای روند SFLA سعی می کنند به غذای بیشتری دست یابند. ابتدا اعضای جمعیت بر اساس مقدار شایستگی شان، مرتب می شوند (بهترین قورباغه در اندیس یک قرار می گیرد). قورباغه ها به m گروه که ممپلکس نام دارند تقسیم می شوند. هر ممپلکس شامل n قورباغه خواهد بود. نحوه تقسیم بندی جمعیت به m ممپلکس بدین ترتیب است که اولین قورباغه به اولین ممپلکس، دومین قورباغه به دومین ممپلکس و m امین قورباغه به m امین ممپلکس داده خواهد شد. سپس $(m+1)$ امین قورباغه به 1 امین ممپلکس داده می شود و به همین ترتیب فرآیند دسته بندی ادامه می یابد تا در هر ممپلکس n قورباغه قرار بگیرند. پس از دسته بندی قورباغه ها در m ممپلکس، هر ممپلکس روند زیر را itr بار تکرار می کند.

در هر ممپلکس موقعیت بهترین و بدترین قورباغه از نظر مقدار شایستگی پیدا می شود که به ترتیب برابر با X_b و X_w

با پویا چندین قله وجود دارد که هر یک از آنها می توانند پس از تغییر محیط تبدیل به بهینه سراسری شود، در نتیجه هر یک از قله ها یک بهینه بالقوه محسوب می شوند. بنابراین الگوریتم باید تمامی قله ها را تحت نظر داشته باشد تا قله بهینه را پس از تغییر محیط به سرعت شناسایی کند. در amSFLA از مکانیزم چنددستگی برای پوشش قله ها استفاده می شود. بدین ترتیب چندین دسته قورباغه در فضای مسئله وجود خواهد داشت که به طور موازی و مستقل از یکدیگر فرآیند بهینه سازی را در محیط انجام می دهند. هر یک از دسته های قورباغه ها روند الگوریتم SFLA را تحت نظارت مرکزی amSFLA انجام می دهند [5] [6].

در amSFLA هر یک از دسته ها وظیفه دارد یکی از قله های موجود در فضای مسئله را تحت نظر داشته باشد. مکانیزم استفاده شده در این الگوریتم برای کنترل دسته ها به صورت تطبیقی می باشد. در amSFLA ابتدا یک دسته در فضای مسئله مقداره می شود و شروع به انجام فرآیند بهینه سازی می کند. پس از اینکه این دسته همگرا شد، یک دسته دیگر در فضای مسئله ایجاد می شود. در واقع یک دسته جدید هنگامی ایجاد می شود که تمام دسته های موجود در فضای مسئله همگرا شده باشند. هنگامی یک دسته یک قله را یافته و به آن همگرا شده که فاصله اقلیدسی موقعیت بهترین قورباغه آن در تکرار lm با موقعیت آن در تکرار $lm+n$ کمتر از یک حد آستانه به نام Γ_{conv} باشد. یک دسته پس از همگرایی به یک قله، وظیفه پوشش دادن این قله و تعقیب آن پس از تغییر محیط را بر عهده دارد. بدین ترتیب دسته های جدید در محیط به جستجوی قله هایی می پردازند که قبلاً کشف نشده باشند. بدین ترتیب تعداد دسته های موجود در محیط منطبق بر تعداد قله ای است که الگوریتم آنها را کشف کرده باشد و انتظار می رود پس از مدتی تمام قله ها تحت پوشش دسته

$$D_d = |X_{Gbest,d} - X_{worst,d}| \times rand(0,2) \quad (3)$$

$$X_{worst,d} = X_{Gbest,d} + D_d \quad (4)$$

که در آن ابتدا برای محاسبه مؤلفه d ام، فاصله بهترین قورباغه دسته و بدترین قورباغه ممپلکس در بُعد d ام محاسبه و در یک عدد تصادفی در بازه $[0,2]$ ضرب می شود. سپس موقعیت جدید بدترین قورباغه ممپلکس با استفاده از رابطه دوم بدست می آید. بدین ترتیب، موقعیت جدید این قورباغه به صورت تصادفی در محدوده ای با شعاع فاصله قبلی آن از بهترین قورباغه دسته، در اطراف بهترین قورباغه دسته موقعیت دهی می شود.

پس از انجام این مرحله توسط تمام ممپلکس ها (به تعداد itr بار)، شرط پایان الگوریتم سنجیده می شود و در صورتی که این شرط برقرار نشود، مجدداً تمام قورباغه ها بدون در نظر گرفتن اینکه عضو کدام ممپلکس بوده اند در یک دسته قرار می گیرند (shuffling) و این مراحل را اجرا می کنند. محیط های پویا از جمله پر چالش ترین محیط ها برای بهینه سازی محسوب می شوند. یکی از مهمترین این چالش ها تعیین موقعیت و تعقیب چندین بهینه است که در ادامه مکانیزم های به کار رفته در الگوریتم جهش قورباغه برای رفع این چالش به طور کامل تشریح می شوند.

3- روش پیشنهادی

در روش پیشنهادی از مکانیزم چنددستگی به صورت تطبیقی استفاده شده است به همین دلیل این الگوریتم را SFLA چنددسته ای تطبیقی یا amSFLA¹ می نامیم. در amSFLA با استفاده از یک مکانیزم تطبیقی هر دو چالش وجود بهینه های بالقوه و مجهول بودن تعداد قله ها رفع می شوند. همان طور که پیش از این گفته شد، در محیط هایی

¹ Adaptive Multi-swarm Shuffled Frog Leaping Algorithm

در صورتی که در تمام ارزیابی شایستگی ها مقدار خطای استاندارد برابر صفر باشد، مقدار خطای برون خطی برابر صفر خواهد شد. خطای برون خطی با استفاده از رابطه (5) محاسبه می شود.

$$offline_error = \frac{1}{Max_FE} \sum_{i=1}^{Max_FE} (f(Swarm_{best}(t)))_{(5)}$$

که در آن Max_FE برابر با تعداد نهایی ارزیابی شایستگی است و $Swarm_{best}(t)$ برابر بهترین موقعیت گروه در زمان ارزیابی شایستگی t ام است. آزمایشات 30 بار اجرا شده است و هر بار تا 100 بار تغییر محیط ادامه داشته است. آزمایشات هر بار با Random Seed های مختلف انجام شده است.

پارامتر r_{conv} جهت تعیین اینکه یک دسته همگرا شده است یا خیر به کار می رود. در صورتی که فاصله اقلیدسی تمام قورباغه های یک دسته کمتر از این پارامتر باشد دسته همگرا شده است. اگر دسته همگرا شده باشد و محیط تغییر کند دسته ها باید دوبار در محدوده مشخصی پخش شوند. مقدار این پارامتر برای تعیین همگرایی دسته ها و رفتارهایی که متقابلا باید انجام شود بسیار مهم می باشد. در جدول 2 مقادیر مختلف این پارامتر مورد بررسی قرار گرفته شده است. تصاویر حاصل از مقایسه مقادیر مختلف این پارامتر در اشکال 1 تا 7 نشان داده شده اند. همانطور که مشاهده می شود مقدار مناسب برای این پارامتر 20 می باشد.

جدول 2: تاثیر پارامتر r_{conv}

r_{conv}	خطای برون خطی (خطای استاندارد)
1	1,33(0,21)
5	1,23(0,06)
10	1,19(0,10)

ها قرار بگیرند. در نتیجه با ایجاد دسته های جدید و کشف قله ها توسط آنها، چالش مجهول بودن تعداد قله ها رفع می شود. همچنین با ساکن شدن دسته ها در قله ها، الگوریتم بر قله ها نظارت دارد و قله ای که پس از تغییر محیط تبدیل به بهینه سراسری شود را به سرعت می یابد بنابراین چالش بهینه های بالقوه نیز در amSFLA رفع شده است.

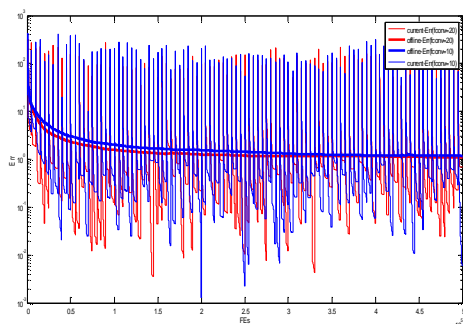
4- بررسی نتایج روش پیشنهادی

از تابع قله های متحرک برای تست کارایی روش پیشنهادی استفاده شده است این تابع این قابلیت را دارد که ابعاد محیط، تعداد قله ها، شکل قله ها، میزان شدت تغییرات و فرکانس تغییرات تنظیم شود. تنظیمات پیش فرض برای آزمایشات در جدول 1 آمده است.

جدول 1: تنظیمات استاندارد برای تابع قله های متحرک

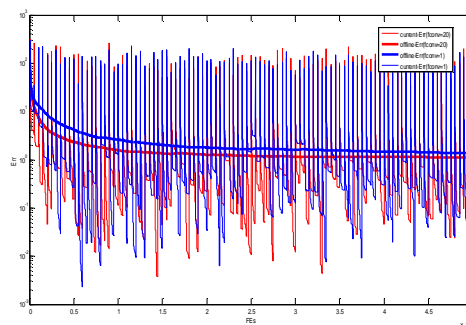
پارامتر	مقدار
تعداد قله ها m	10
فرکانس تغییرات	5000
شدت تغییرات ارتفاع قله ها	7
شدت تغییرات عرض قله ها	1
شکل قله ها	cone
میزان حرکت مکان قله ها	1
ابعاد فضای جستجو	5
محدوده ارتفاع قله ها	[30,70]
محدوده عرض قله ها	[1,12]
ارتفاع استاندارد قله ها	50
محدوده ی فضای جستجو	[0,100]

همچنین برای معیار مقایسه کارایی از خطای برون خطی استفاده می شود. خطای برون خطی برابر با متوسط شایستگی بهترین موقعیت یافت شده توسط الگوریتم در تمامی ارزیابی شایستگی ها می باشد. در واقع خطای برون خطی برابر با متوسط مقدار خطای استاندارد در طول اجرای الگوریتم است.

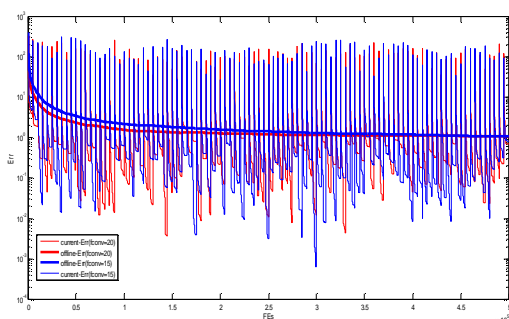


شکل 3: مقایسه خطای برون خطی و خطای استاندارد بدست آمده با r_{cove} برابر با 20 و r_{cove} برابر با 10 در 100 بار تغییر محیط با فرکانس 5000 و 10 قله

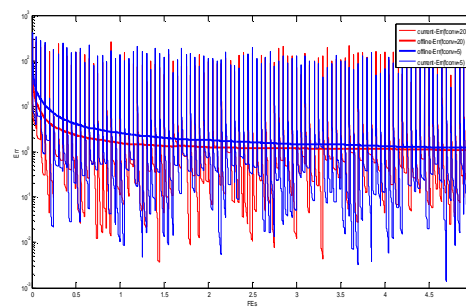
15	1,10(0,07)
20	1,09(0,03)
25	1,38(0,124)
30	1,16(0,123)
50	1,47(0,18)



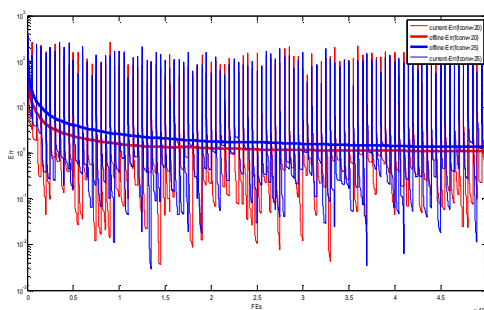
شکل 1: مقایسه خطای برون خطی و خطای استاندارد بدست آمده با r_{cove} برابر با 20 و r_{cove} برابر با 1 در 100 بار تغییر محیط با فرکانس 5000 و 10 قله



شکل 4: مقایسه خطای برون خطی و خطای استاندارد بدست آمده با r_{cove} برابر با 20 و r_{cove} برابر با 15 در 100 بار تغییر محیط با فرکانس 5000 و 10 قله



شکل 2: مقایسه خطای برون خطی و خطای استاندارد بدست آمده با r_{cove} برابر با 20 و r_{cove} برابر با 5 در 100 بار تغییر محیط با فرکانس 5000 و 10 قله



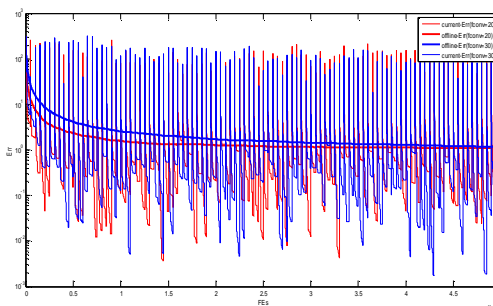
5- نتیجه گیری

در این مقاله یک روش نوین برای تعیین موقعیت و تعقیب چندین بهینه در محیط پویا مبتنی بر الگوریتم جهش قورباغه پیشنهاد شده است. نتایج حاصل از روش پیشنهادی بر روی معیار قله های متحرک که در حال حاضر شناخته شده ترین معیار برای ارزیابی در محیط های پویا می باشد و همچنین با استفاده از خطای برون خطی ارزیابی شده است. نتایج بدست آمده نشان دهنده کارایی بالای روش پیشنهادی می باشد.

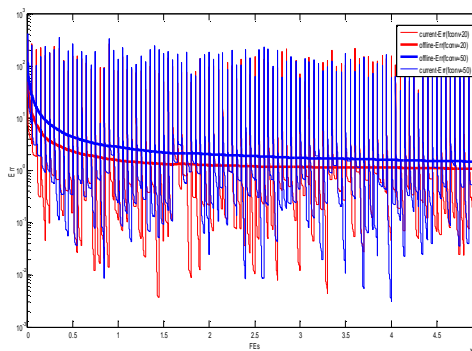
منابع

- [1] M. Eusuff, K. Lansey, F. Pasha, "Shuffled frog-leaping algorithm: amemetic meta-heuristic for discrete optimization," *Engineering Optimization*, vol. 38, No. 2, pp. 129-154, 2006.
- [2] S. Ranjesh, M. R. Meybodi, B. Masomi "A new Approach for Optimization in Dynamic Environments based on Frog Leaping Algorithm," in *The sixth Iran Data Mining Conference, IDMC, Tehran, Iran, 2012*.
- [3] <http://www.aifb.unikarlsruhe.de/~jbr/MovPeaks/>
- [4] 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. 1-16, 2010.
- [5] T. Blackwell, J. Branke, "Multiswarms, exclusion, and anti-convergence in dynamic environments," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 459-472, 2006.
- [6] T. Blackwell, J. Branke, X. Li, "Particle swarms for dynamic optimization problems," *Swarm Intelligence*, pp. 193-217, 2008.

شکل 5: مقایسه خطای برون خطی و خطای استاندارد بدست آمده با r_{cove} برابر با 20 و r_{cove} برابر با 25 در 100 بار تغییر محیط با فرکانس 5000 و 10 قله



شکل 6: مقایسه خطای برون خطی و خطای استاندارد بدست آمده با r_{cove} برابر با 20 و r_{cove} برابر با 30 در 100 بار تغییر محیط با فرکانس 5000 و 10 قله



شکل 7: مقایسه خطای برون خطی و خطای استاندارد بدست آمده با r_{cove} برابر با 20 و r_{cove} برابر با 50 در 100 بار تغییر محیط با فرکانس 5000 و 10 قله