

الگوریتم ترکیبی خوراک جویی زنبور مبتنی بر حافظه و جستجو

برای بهینه‌یابی محیط‌های پویا

ناهید شایگان پور^۱، محمد رضا میبیدی^۲

^۱آموزشکده فنی و حرفه ای سما، دانشگاه آزاد اسلامی، واحد بروجرد، بروجرد، ایران
nahid_shayeganpour@yahoo.com

^۲عضو هیئت علمی دانشگاه امیر کبیر تهران، دانشکده کامپیوتر
mmeybodi@aut.ac.ir

چکیده

اخیراً در سالهای اخیر علاقه رو به رشدی در حوزه هوش دسته جمعی در مسائل بهینه‌یابی محیط‌های پویا به دلیل اهمیت آن در دنیای واقعی ایجاد شده است. هوش دسته جمعی یک حوزه جدید از تحقیقات است که بروی مطالعه و مدلسازی رفتار اجتماعی حشراتی مثل مورچه‌ها و زنبورها تمرکز دارد. کلونی زنبورها دارای این قابلیت می‌باشد که خود را به راحتی با محیط در حال تغییر وفق دهد به همین منظور می‌توان از الگوریتم‌های زنبور برای بهینه‌یابی مسائل پویا استفاده کرد. از آنجا که برخی از مسائل دنیای واقعی تکرارپذیر می‌باشند نیاز به یک حافظه جهت نگهداری راه‌حل‌های گذشته به منظور تقویت کارایی الگوریتم‌ها برای مسائل بهینه‌یابی پویا ضروری به نظر می‌رسد. در این مقاله الگوریتم خوراک جویی زنبور با یک حافظه صریح ترکیب گردیده و بر روی تابع محک قله‌های در حال نوسان آزمایش شده و نتایج با الگوریتم‌های تکاملی، الگوریتم‌های تکاملی مبتنی بر حافظه، روش‌های چندجمعیتی و مهاجرهای تصادفی مقایسه گردیده است. نتایج نشان می‌دهد وجود حافظه به دلیل به کارگیری راه‌حل‌های گذشته در بهبود الگوریتم و کاهش خطا موثر می‌باشد و از روش‌های ارائه شده قبلی بهتر است.

کلمات کلیدی

هوش دسته جمعی، حافظه صریح، محیط پویا، خوراک جویی زنبور

۱- مقدمه

در این مقاله الگوریتم پیشنهاد شده از ترکیب الگوریتم‌های رفتار خوراک‌جویی زنبور که دارای انعطاف‌پذیری، خودسازماندهی، مقیاس‌پذیری، تقسیم کار و خلاقیت در یافتن راه‌حل‌ها می‌باشند با حافظه و راه‌اندازی مجدد ایجاد گردیده است. بدین صورت که کلونی به ۲ زیرکلونی تقسیم شده و کلونی اول وظیفه استفاده از راه‌حل‌های قبلی در حافظه و کلونی دوم وظیفه اکتشاف برای یافتن راه‌حل‌های جدید را برعهده دارد و هر یک از زنبورها در کلونی سعی می‌کنند با تعامل مستقیم و به اشتراک گذاشتن اطلاعات خود، بهترین راه‌حل‌های بدست آمده را بر اساس قوانین احتمالی انتخاب کند، به عبارت دیگر زنبورها در کنار یکدیگر دارای فاکتور هوشمند می‌شوند که در غیاب یکدیگر فاقد آن هستند. در این الگوریتم هر یک از زنبورها برای اولین بار یکسری راه‌حل را به طور تصادفی انتخاب کرده و با اعمال تغییرات محلی بروی راه‌حل‌ها، آنها را بهبود می‌دهند و زنبورهای دیگر با استفاده از یک روش انتخاب حریصانه از بین راه‌حل‌های انتخابی و بر اساس احتمال خود، تعدادی را انتخاب و سپس هر گروه را بهبود و یک زنبور را از میان آنها برگزیده و دنبال می‌کنند. همچنین در این الگوریتم با ارسال دوره‌ای زنبورهای پیشاهنگ به منظور جستجوی

برخی از مسائل پویای دنیای واقعی ماهیت تکرارپذیر دارند یعنی در هدف بهینه‌یابی، نمونه مسئله یا در محدودیت^۱ یک مسئله بهینه‌یابی تغییرات تکرار می‌شوند در این نوع مسائل انتقال اطلاعات از محیط قدیمی به محیط جدید امری اجتناب‌ناپذیر است به طوری که بتوان از راه‌حل‌های بهینه در محیط جدید استفاده نمود و این امر موجب می‌گردد که زمان کمتری را برای یافتن راه‌حل جدید اختصاص یابد. به منظور انتقال اطلاعات از محیط قدیمی به محیط جدید می‌توان از یک حافظه استفاده نمود که در زمان تغییر محیط بتوان از راه‌حل‌های موجود در حافظه برای حل مسئله جدید استفاده کرد. اضافه کردن حافظه به انواع الگوریتم‌هایی مانند الگوریتم‌های تکاملی [۱و۲] موجب جلوگیری از کشف نواحی جدید در فضای جستجو و همگرا^۲ شدن به یک بهینه را دارد بنابراین جمعیت تنوع^۳ لازم را برای اکتشاف^۴ به صورت کارا و الگوریتم توانایی لازم برای وفق‌پذیری با تغییرات محیط را از دست می‌دهد. به همین دلیل می‌توان برای رفع این مشکل از راه‌اندازی مجدد جمعیت در زمان تغییر جدید استفاده کرد.

بازیابی از حافظه تنها بعد از این که محیط تغییر کرد اتفاق می افتد. برای بازیابی افراد از حافظه، n فرد جمعیت و m فرد موجود در حافظه ترکیب شده و تنها n فرد از بهترین افراد برای جمعیت جدید انتخاب می شوند.

نتایج آزمایش های الگوریتم های مبتنی بر حافظه نشان می دهد، امکان گم کردن سیر تکاملی و جلوگیری از اکتشاف مناطق جدید در حوزه جستجو که نباید دست کم گرفته شود وجود دارد. وقتیکه محیط تغییر می کند برای جلوگیری از گم کردن سیر تکاملی، راه اندازی مجدد بهترین راه حل است و شانس خوبی برای پیدا کردن قله های جدید ایجاد می کند اگر چه مدت زمان زیادی برای رسیدن به بهینه صرف خواهد شد که با ترکیب نوعی حافظه اکتشافی با الگوریتم خوراک جویی این مشکل برطرف شده است.

۳- الگوریتم ترکیبی خوراک جویی زنبور مبتنی بر حافظه / جستجو

این الگوریتم ترکیبی از کلونی زنبور مصنوعی^۵ [۳-۵] و زنبورها^۴ [۴و۵] می باشد. هدف از ترکیب این دو الگوریتم بکارگیری نقاط مثبت و قوت دو الگوریتم برای افزایش همگرایی و یافتن بهینه سراسری است. بدین صورت که استفاده از استراتژی اصلاح کلونی زنبورهای مصنوعی، همگرایی سریع به پاسخ بهینه و استفاده از سایت بندی و رتبه بندی در الگوریتم زنبور، نزدیک شدن تا حد رسیدن به پاسخ را در جستجوی محلی تضمین می کند.

همانند الگوریتم زنبورهای مصنوعی، در این مدل نیز یک جمعیت از زنبورها وجود دارد و هر یک از زنبورها نشان دهنده یک موقعیت یا منبع غذایی می باشند در این الگوریتم ۳ دسته زنبور کارگر، تماشاگر و پیشاهنگ وجود دارد. در ابتدا تعدادی زنبور کارگر به طور تصادفی در فضای جستجو مقاداردهی شده و شایستگی آنها محاسبه می گردد، در صورتی که میزان شایستگی آنها از راه حل قدیمی بیشتر باشد، راه حل جدید انتخاب می گردد در غیر این صورت همان راه حل قدیمی باقی خواهد ماند به عبارت دیگر در پی یافتن راه حل بهتر، خود را طبق فرمول (۱) اصلاح می کنند [۴].

$$v_{ij} = x_{ij} + \phi_{ij} (x_{ij} - x_{kj}) \quad (1)$$

در فرمول (۱)، ϕ_{ij} یک عدد تصادفی در بازه $[-1, 1]$ است که تولید موقعیت منابع غذایی همسایه را در اطراف x_{ij} کنترل می کند و SN تعداد زنبورها و D ابعاد مسئله می باشد. نتایج بدست آمده طبق الگوریتم زنبور رتبه بندی می شود و تعداد کمی از زنبورها با شایستگی کم از میان آنها حذف می شوند. زنبورهای تماشاگر که در کندو منتظر زنبورهای رتبه بندی شده یا همان زنبورهای کارگرها هستند با توجه به دو متغیر nep و nsp زنبورهای

تصادفی محیط، دامنه جستجو سراسری افزایش می یابد. ادامه این مقاله بدین صورت سازماندهی شده است که بخش ۲ به معرفی شرح حافظه صریح و بخش بعدی به معرفی الگوریتم پیشنهادی پرداخته است. در بخش ۳ معرفی تابع پویا و آزمایش های انجام شده بروی آن و مقایسه با روشهای دیگر [۱و۲] انجام می شود و بخش پایانی نتیجه گیری است.

۲- حافظه صریح

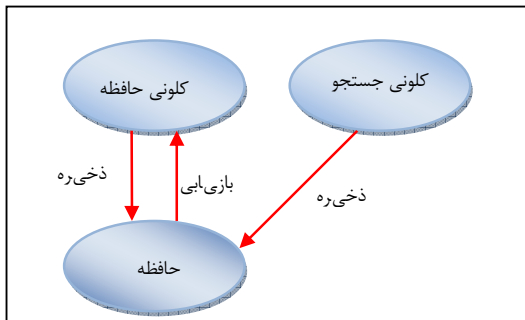
ایده حفظ کردن افراد موفق در یک پایگاه دانش [۲] در الگوریتم به نظر جذاب می آید به این ترتیب که زمانیکه بهینه دوباره در مکان قبلی ظاهر می شود یک حافظه می تواند مکان آنرا به خاطر آورد و فوراً جمعیت را به بهینه جدید حرکت دهد. حافظه می تواند در حفظ تنوع مفید باشد و تکامل را برای حوزه های امید بخش بعد از راه اندازی مجدد راهنمایی کند در صورتیکه فرض شود تغییرات در محیط کوچک هستند یک فرد (راه حل) سابقاً خوب، شبیه به فرد (راه حل) خوب بعدی خواهد بود.

در استفاده از حافظه این سوال مطرح است که از حافظه صریح یا حافظه ضمنی استفاده شود که در این مقاله حافظه صریح استفاده کردیم به این دلیل که درک تاثیر آن ساده تر بوده و بنابر آن چه که تاکنون گزارش شده است، حافظه ضمنی برای مسائلی با بیش از دو وضعیت مناسب نمی باشد. با توجه به این که از حافظه صریح استفاده خواهیم کرد سوالات زیر مطرح می شود:

- چه زمانی و کدام یک از افراد جمعیت باید در حافظه ذخیره شوند؟
- چه تعدادی از افراد باید در حافظه ذخیره شوند و بجای کدام یک باید جایگزینی صورت پذیرد تا فضا برای درج فرد جدید وجود داشته باشد؟
- کدام یک از افراد باید از حافظه بازیابی شوند و دوباره به جمعیت اضافه شوند؟

در پاسخ به سؤالها باید گفت که، افرادی که در حافظه ذخیره می شوند باید دارای شایستگی بالاتر از میانگین شایستگی کل جمعیت داشته باشند، بسیار قدیمی نباشند و در مکان های مختلف از فضای جستجو قرار گرفته باشند. پس به نظر می رسد که در هر زمان باید بهترین افراد در حافظه قرار گیرند.

یکی از استراتژی های جایگزینی، کوتاهترین فاصله، است و مورد آزمایش قرار گرفته است در این استراتژی برای نگه داری تنوع، می توان دو فرد i و j با کمترین فاصله را در نظر گرفته و فردی که شایستگی کمتری دارد را جایگزین کرد. ایده مورد نظر این است که یک فرد در یک ناحیه مشخص کافی است و اگر دو فرد نزدیک به یک دیگر در یک ناحیه وجود دارند، تنها فرد شایسته تر باید باقی بماند و تعداد افراد به طور عمومی ۲۰ تا ۳۰ درصد جمعیت در نظر گرفته می شوند.



شکل (۱): الگوریتم ترکیبی زنبور بر مبنای حافظه با دو کلونی

بدین ترتیب شبه کد الگوریتم پیشنهادی در شکل (۲) نشان داده شده است.

```

1. Initialize n scout bees with random solutions for search colony and memory colony.
2. Memory=0 Chang=0 //no chang
Do
3. If (change=1) search colony=merge search colony and memory
4. For search and memory colony do
5. For each bee in colony
Calculate fitness value of the bee fi
End For
6. For each bee
Produce a modification on the Position of The bee
End For
7. Apply the greedy selection process
8. Apply the ranking evaluation bees
9. The m sites will be selected randomly (the best evaluation to m scout bee) from n
10. Recruit Bees for the selected sites and evaluate the fitness of the sites
Number of bees (nep) will be selected randomly to be sent to e
11. Sites and choosing nsp bees randomly which their number is less than nep, to be sent to m-e sites
12. For each recruit bee
Produce a modification on the Position of the bee
End For
13. Apply the greedy selection process
14. Choosing the best bee from each site (the highest fitness) to form the next bee population
15. Assign remaining bees to search randomly and evaluate their Fitnesses.
End for
16. If (change=1) memory=select best bees in search and memory colony (minimum distance replacement strategy) and Initialize n scout bees with random solutions for search colony .
While maximum iterations OR (stopping criterion not met)

```

شکل (۲): شبه کد الگوریتم پیشنهادی

۴- آزمایش‌ها

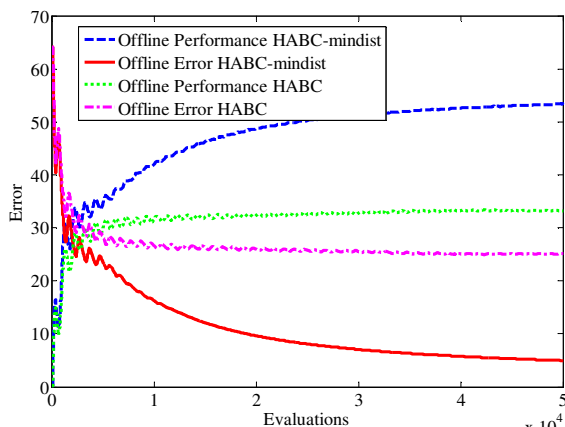
تابعی که در آزمایش‌ها مورد استفاده قرار گرفته، تابع قله‌های در حال نوسان^[۱،۲] است که در آن L منظره (معمولا L=۲) که هر کدام حاوی m قله با ارتفاع و پهنای متفاوت می‌باشند بر طبق تابع کسینوسی رابطه (۳) نوسان می‌کنند.

کارگر را انتخاب می‌کنند به عبارت دیگر به تعداد nep زنبور تماشاگر به سوی هر کارگری که شایستگی بالایی دارد فرستاده می‌شوند و به تعداد nsp زنبور تماشاگر به سوی هر کارگری که شایستگی متوسطی دارند فرستاده می‌شود (این مرحله رقص زنبورهای کارگر در کندو نام دارد و هر زنبوری که شایستگی بیشتری داشته باشد می‌تواند زنبورهای بیشتری را به سمت خود جلب کند) این عمل بوسیله یکی از روشهای انتخاب مثل انتخاب متناسب با کمک تابع احتمال فرمول (۲) بر اساس شایستگی هر یک انجام می‌شود [۵].

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (2)$$

در فرمول (۲)، fit_i میزان شایستگی منبع غذایی متناظر با زنبور نام می‌باشد. سپس زنبورهای تماشاگر خود را طبق فرمول (۱) اصلاح می‌کنند و بوسیله یک انتخاب حریصانه، بهترین زنبور تماشاگر اصلاح شده از میان زنبورهای تماشاگر فرستاده شده به سمت هر زنبور کارگر انتخاب می‌شود. در این روش شانس بهبود هر تماشاگر بیشتر از روشهای قبل است به این علت که به هر زنبور تماشاگر این اجازه را می‌دهد تا به تعداد دو متغیر nsp و nep زنبور کارگر را دنبال کند و آنها را بهبود دهد. به عبارت دیگر یک زنبور کارگر را چندین بار تغییر داده و بهترین را انتخاب می‌کند. بعد از انتخاب بهترین تماشاگر می‌بایست بجای زنبورهایی که به علت شایستگی کم، حذف شده‌اند به طور تصادفی زنبورهایی به نام زنبور پیش‌آهنگ اضافه شود و این چرخه تا رسیدن به پاسخ بهینه تکرار می‌شود.

در شکل (۱) نشان داده شده است که چگونه از مزیت هر دو ساختار در الگوریتم استفاده شده است. این ساختار، کلونی را به دو زیر کلونی تقسیم کرده، یکی کلونی حافظه و دیگری کلونی جستجو. کلونی اول مبتنی بر حافظه و مسئول یادآوری راه حل‌های خوب قدیمی و نگهداری یک کیفیت مینیمم و اکتشاف و تصحیح کردن دانش موجود درباره فضای جستجو می‌باشد و کلونی دیگر دائماً راه حل‌های جدید را جستجو می‌کند و آنها را به حافظه ارائه می‌کند ولیکن هیچ اطلاعاتی از آن بازیابی نخواهد شد. هر کلونی طبق الگوریتم ترکیبی دارای زنبورهای تماشاگر و کارگر و عملیات گفته شده بروی آنها می‌باشد. اکتشاف بدین صورت اعمال می‌شود که کلونی جستجو بعد از هر تغییر در محیط مقدار اولیه می‌گیرد. فقط زمانیکه تنوع در جمعیت جستجو خیلی زیاد است نیازی به راه‌اندازی دوباره نمی‌باشد. در زمان تغییر، بازیابی فرد با شایستگی بالا از حافظه به کلونی حافظه به منظور استفاده از راه حل‌های قبلی ذخیره شده در حافظه انجام می‌پذیرد در این زمان برای حفظ راه حل‌های جدید بدست آمده، بهترین فرد از کلونی حافظه و کلونی جستجو به حافظه منتقل می‌شود.



شکل (۳): میانگین خطا و میانگین کارایی در فرکانس ۵۰۰۰۰ با طول چرخه ۱۰۰۰ در روش پیشنهادی بدون و با حافظه/جستجو

در شکل (۵) نتایج میانگین خطا در الگوریتم بدون حافظه و با حافظه در سه فرکانس تغییر کم، زیاد و متوسط نمایش داده شده است. نتایج نشان می‌دهد بجز در فرکانس تغییر بالا (۱۰۰۰ ارزیابی) اضافه نمودن حافظه تأثیری زیادی در کاهش میانگین خطا دارد. از نمودار کاملاً مشخص است که وقتی فرکانس تغییر افزایش می‌یابد از ۱۰۰۰۰۰ ارزیابی به سمت ۱۰۰۰ ارزیابی به علت سرعت بالای تغییرات میزان خطا افزایش چشم‌گیری دارد.

در آزمایش‌های بعدی که نتایج آن در شکل (۶ و ۷) نشان داده شده است میزان میانگین خطا الگوریتم پس از ۵۰ تغییر در فرکانس‌های مختلف با الگوریتم‌های دیگر مقایسه شده و مشخص گردیده که میزان خطا در تمام روش‌ها زمانیکه از حافظه استفاده نشده بسیار بیشتر از زمانی است که از حافظه استفاده شده (مقایسه نتایج شکل (۶ و ۷) است.

شکل (۶) نشان می‌دهد که الگوریتم نسبت به تمامی روش‌های دیگر بدون حافظه میانگین خطای کمتری دارد و شکل (۷) نیز همین مسئله را در حالت حافظه دار نشان می‌دهد بجز در فرکانس‌های بسیار بالا که به علت تغییرات بسیار سریع، حافظه در روند پیدا کردن راه حل اختلال ایجاد می‌کند.

وقتی فرکانس تغییر به یک تغییر در هر نسل افزایش می‌یابد (یک چرخه هنوز دارای ۱۰ گام میانی می‌باشد) کارایی همه شیوه‌ها به طور معنی داری زیان می‌بیند این زیان ممکن است به سبب خصوصیت اندازه‌گیری *offline error* باشد. زمانیکه محیط در هر نسل تغییر می‌کند بهترین فرد پیدا شده در یک نسل در نسل بعدی محسوب نمی‌شود در نتیجه ارزیابی‌ها در هر نسل بسیار تأثیر گذار می‌شود. اساساً ترتیب ارزیابی‌ها درون یک نسل با اهمیت می‌باشد و در نتیجه روش جستجوی اکتشافی ویژه به خاطر اینکه اکتشاف به طور طبیعی مناطق

$$\begin{cases} f_i(t) = \omega(t)f_i(0) \\ \omega(t)f_i(0) = 0.5\cos\left(\frac{2t\pi}{\text{step}} + 2\pi\frac{i-1}{1}\right) + 0.5 \\ i = 1 \dots 1 \end{cases} \quad (3)$$

با step تعداد گام‌های میانی در یک سیکل تعریف می‌شود. در نتیجه مسئله بین L منظره نوسان می‌کند و از تابع اولی f_1 به سوی تابع f_2 و بالعکس حرکت می‌کند. در آزمایش‌ها مقادیر پارامترها طبق جدول (۱) در نظر گرفته شده است

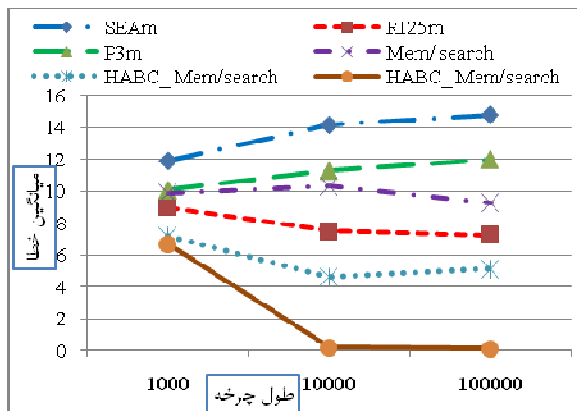
جدول (۱): پارامترهای پیش فرض در آزمایش قله‌های در حال نوسان

پارامتر	مقدار پیش فرض
تعداد مناظر	۲
P تعداد قله‌ها	۱۰
تعداد گام‌های هر چرخه	۱۰
شکل قله‌ها	ناقوسی
S طول تغییر و D تعداد ابعاد	۵ و ۱
H مقدار مینیمم و ماکزیمم ارتفاع	[۳۰ و ۷۰]
W مقدار مینیمم و ماکزیمم پهنا	[۰.۰۰۱ و ۰.۰۰۸]
A محدوده فضای جستجو	[۰ و ۱۰۰]

در تمامی روش‌ها تعداد جمعیت ۱۰۰ و در صورت استفاده از حافظه ۱۰ مکان از جمعیت به حافظه داده شده است و استراتژی جایگزینی در تمامی روش‌ها کمترین فاصله در نظر گرفته شده است. در الگوریتم‌های تکاملی به منظور تولید فرزندان و انتخاب والدین از روش بازترکیبی دوقطه‌ای با نرخ ۰.۶ و جهش گوسین با نرخ ۰.۲ و روش انتخاب تورنامنت استفاده شده است.

در الگوریتم پیشنهادی مقادیر *nep* و *nsp* به ترتیب ۳ و ۲ در نظر گرفته شده و در پایان الگوریتم ۵ زنبور به طور تصادفی اضافه می‌شوند و در صورت استفاده از حافظه هر کلونی ۴۵ زنبور دارد و در هر تغییر یک زنبور نخبه به جمعیت بعدی انتقال می‌یابد.

در آزمایش‌های انجام شده هدف بررسی و مقایسه الگوریتم‌ها در محیط‌های پویا به کمک قله‌های در حال نوسان می‌باشد از آنجا که در محیط‌های پویا بهترین راه حل بدست آمده معیار مفیدی برای ارزیابی و گزارش نمی‌باشد، از *current_error* و *offline_error* جهت مقایسه با روش‌های همچون الگوریتم تکاملی، الگوریتم‌های تکاملی مبتنی بر حافظه، چندجمعیتی و مهاجرهای تصادفی استفاده شده است. در شکل (۴ و ۳) میانگین خطا و خطای لحظه‌ای روش پیشنهادی بدون حافظه و با حافظه در ۵۰۰۰۰ ارزیابی اولیه با فرکانس تغییر ۱۰۰۰۰ نشان داده شده است که سرعت کاهش میانگین خطا در شکل (۴) که با حافظه کار می‌کند نشان می‌دهد که الگوریتم با حافظه بهتر عمل می‌کند این مسئله با توجه به نحوه کاهش سریع خطای لحظه‌ای نیز قابل توجیه است.



شکل (۷): مقایسه میانگین خطا با توجه به فرکانس تغییرات بین الگوریتم پیشنهادی H_ABC_MS با الگوریتم‌های دیگر با حافظه

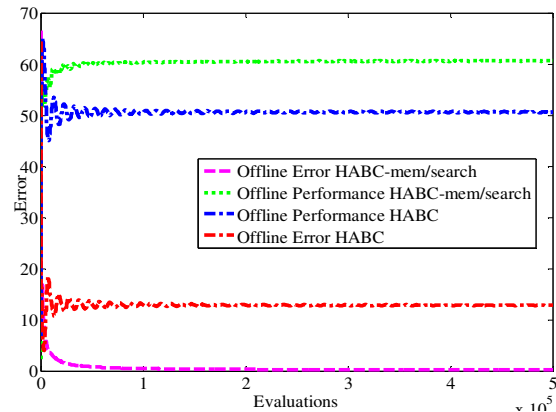
۵- نتیجه گیری

الگوریتم پیشنهادی که بر مبنای حافظه و جستجو عمل می‌کند با معیار میانگین خطا با الگوریتم‌های موجود برای حل مسئله پویایی همچون کوله پشتی پویا انجام و مقایسه شده و نتایج نشان می‌دهد این الگوریتم نسبت به آنها میانگین خطای کمتری داشته و نتایج را بهبود می‌بخشد بجز زمانیکه فرکانس تغییرات بالا مثل هر ۱ نسل که میانگین خطا افزایش نیز داشته و این ناشی از نداشتن فرصت برای جستجوی راه حل و استفاده از راه‌حل نامتناسب در تغییر بعدی می‌باشد که در زمان فرکانس بالا پیشنهاد می‌شود جمعیت فقط به یک زیرکلونی، زیرکلونی جستجو تقسیم شده و این زیرکلونی در مدت زمانهای طولانی و در صورت عدم تنوع راه‌اندازی مجدد شود. همچنین الگوریتم ترکیبی ارائه شده به تنهایی توازن مناسبی بین فرایند اکتشاف و بهره‌برداری برقرار می‌کند که ناشی از رفتار زنبورهای کارگر، تماشاگر و پیشاهنگ است که قدرت جستجوی سراسری بالایی دارد و باعث جلوگیری از همگرایی به بهینه محلی می‌گردد. به عبارتی الگوریتم توانایی یافتن راه‌حل بهینه را داراست و ضمناً با ایجاد تغییرات در شرایط مسئله بسیار سریع خود را تطبیق می‌دهد.

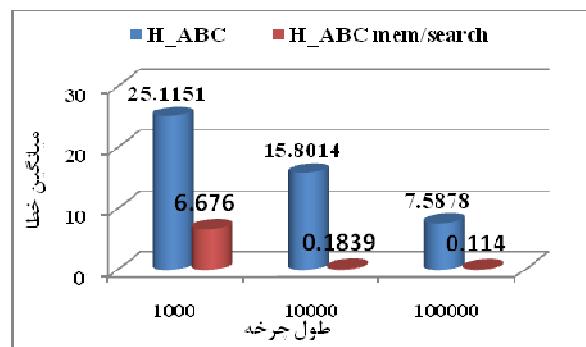
مراجع

- [1] Branke, J., *Evolutionary optimization in dynamic environments*, kluwer academic publishers, 2000.
- [2] Branke, J., "Memory Enhanced Evolutionary Algorithms for Changing Optimization Problems", in 1999 Congress on Evolutionary Computation, Washington D. C., USA, pp. 1875-1882, 1999.
- [3] Karaboga, D., Basturk, B., "A powerful and Efficient Algorithm for Numerical Function Optimization:

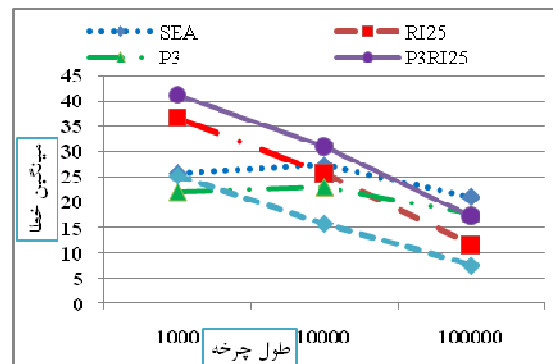
جدید از فضای جستجو را در آزمایش درگیر می‌کند، جریمه می‌گردد و قابلیت یا مزیت یافتن راه‌حل‌های خوب جدید را از دست می‌رود.



شکل (۴): میانگین خطا و خطای لحظه‌ای در فرکانس ۵۰۰۰۰ با طول چرخه ۱۰۰۰۰ در روش پیشنهادی بدون و با حافظه/جستجو



شکل (۵): میانگین خطا با توجه به فرکانس تغییرات در الگوریتم پیشنهادی با حافظه/جستجو و بدون حافظه



شکل (۶): مقایسه میانگین خطا با توجه به فرکانس تغییرات بین الگوریتم پیشنهادی H_ABC با الگوریتم‌های دیگر بدون حافظه

- Artificial Bee Colony (ABC) Algorithm*”, Journal of Global Optimization, vol. 39, pp. 459-471, 2007.
- [4] Karaboga, D., Basturk, B., “*Artificial Bee Colony (ABC) Optimization Algorithm for Solving constrained Optimization Problems*”, Department of Computer Engineering, University of Erciyes, Tech. Rep. No. 2007-02, 2007.
- [5] Karaboga, D., Basturk, B., “*On the Performance of Artificial Bee Colony (ABC) Algorithm*”, Journal of Soft computing, vol. 8, pp. 687-697, 2008.
- [6] Pham, D. T., Kog, E., Ghanbarzadeh, A., Otri, S., Rahim, S., Zaidi, M., “*The Bees Algorithm - A Novel Tool for Complex Optimisation Problems*”, Proceeding 2nd International Virtual Conference on Intelligent Production Machines and Systems, pp. 454-459, 2006.
- [7] Pham, D. T., Ghanbarzadeh, A., “*Multi-Objective Optimisation using the Bees Algorithm*”, Proceedings of IPROMS 2007 Conference, Cardiff, UK, pp. 454-461, 2007.

زیر نویس ها

-
- ¹ Restriction
² Converge
³ Diversity
⁴ Exploring
⁵ Artificial bee colony
⁶ Bee Algorithm
⁷ Oscillatory function