

الگوریتم بهینه یابی ترکیبی مبتنی بر رفتار خوارک جویی زنبور عسل برای محیط های پویا

ناهید شایگان پور^۱؛ محمد رضا میبدی^۲

چکیده

اخیراً علاقه به بکارگیری هوش دسته جمعی در مسائل بهینه یابی در حال تغییر افزایش یافته است. هوش دسته جمعی، عامل‌ها یا دسته‌های برهم کنش را که قادر به خودسازماندهی هستند مدلسازی می‌کند. کلونی مورچه‌ها، دسته پرنده‌گان، گله حیوانات، قالبگیری باکتریها و گروه زنبورها مثالهایی از یک سیستم جمعی می‌باشند. عمرکرد خوب بهینه یابی گروه زنبورها بروی یک سری از مسائل ایستاده است ولیکن بیشتر مسائل دنیای واقعی پویا می‌باشند به این مفهوم که موقعیت بهینه سراسری و مقدار آن ممکن است در طول زمان تغییر کند. در این مقاله یک الگوریتم بهینه یابی توسعه یافته کلونی زنبورهای مصنوعی که HABC نامیده شده برای حل مسائل پویا ارائه گردیده است. روش پیشنهاد شده برگرفته از ترکیب کلونی زنبورهای مصنوعی و الگوریتم زنبور می‌باشد. الگوریتم HABC ایده جذاب فرآیند را برای حرکت زنبورها در کلونی زنبورهای مصنوعی هدایت می‌کند و به طور موفقیت آمیزی قابلیت اکتشاف محلی را در کلونی زنبورهای مصنوعی افزایش می‌دهد و همچنین راه حل‌ها به منظور درج مهاجرهای تصادفی در الگوریتم زنبور رتبه بندی می‌شوند این موضوع به طور موثری قابلیت اکتشاف سراسری را در الگوریتم زنبور افزایش می‌دهد. دوتابع محک (توابع قله‌های متحرک و در حال نوسان) در آزمایشها به منظور مقایسه کیفیت و صحت الگوریتم تکاملی مبتنی بر حافظه واریانسی، فاصله‌ای، SOS، RPSO، mQSO، Adaptive mQSO و الگوریتم HABC شبیه سازی شده است. نتایج آزمایشات برتری دقت الگوریتم پیشنهاد شده را نسبت به روش‌های دیگر آشکار می‌کند.

کلمات کلیدی

هوش دسته جمعی، کلونی زنبورهای مصنوعی، الگوریتم زنبور، محیط پویا

Hybrid Optimization Algorithm based on Foraging Behavior of Honey Bee for Dynamic Environments
Nahid Shayeganpour; Mohammad Reza Meybodi

ABSTRACT

Recently there has been increased interest in swarm intelligence applied to changing optimization problems. Swarm intelligence models the population of interacting agents or swarms that are able to self organize. Ant colonies, bird flocking, animal herding, bacterial molding and bees swarming is a typical example of a swarm system. Bee swarm optimization (BSO) has been shown to perform well for many static problems. However, many real world problems are dynamic in the sense that the global optimum location and value may change with time. In this paper presents an enhanced artificial bee colony (ABC) optimization algorithm which is called the hybrid artificial bee colony(HABC) optimization for solving the dynamic environment problems. The proposed approach is based on combining ABC and bee algorithm. The HABC leads in the concept of universal gravitation to the movement of bees in the ABC, and it successfully increases the local exploration ability of the ABC and that ranks solutions in order to inserting random immigrants in the bee algorithm, and it effectively increases the global exploration capability of the bee algorithm. Two benchmark functions (moving peaks and oscillating peaks) are simulated in the experiments in order to compare the accuracy, quality of the EA based on

^۱ دانشجوی کارشناسی ارشد کامپیوتر گرایش نرم افزار، دانشگاه آزاد اسلامی واحد قزوین، دانشکده برق و رایانه فناوری اطلاعات
آدرس پست الکترونیکی: nahid_shayeganpour@yahoo.com

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

memory such as variance, mindist, SOS, RPSO, mQSO, Adaptive mQSO and HABC. The experimental results manifest the superiority in accuracy of the proposed HABC to other methods.

KEYWORDS

Swarm Intelligence, Artificial Bee Colony, Bee Algorithm, Dynamic Environment

۱. مقدمه

امروزه بیشتر مقالات در حیطه کلونی زنبور عسل^۱ برای بهینه یابی در محیط های ایستا بوده است، ولی مسائل دنیای واقعی اساساً پویا هستند برای مثال زمانبندی کارهای جدید که در حال اضافه شدن به صف در طی زمان هستند نمونه ای از مسئله بهینه یابی در محیط پویا می باشد. در نتیجه نیاز امروز ایجاد راهکارهایی است که پاسخگوی مسائل بهینه یابی در محیط های پویا باشد. شاید تصور این باشد که با هر تغییر، مسئله را به عنوان یک مسئله جدید شناسایی کرده و همانند مسائل محیط های ایستا حل کرد بدون اینکه از اطلاعات گذشته یا حالت قبلی مسئله استفاده کنیم. با کمی تأمل می توان دریافت که حل یک مسئله از ابتدا بدون استفاده مجدد از اطلاعات گذشته بسیار وقت گیر و حتی ممکن است یافتن یک تغییر برای مدتی غیرقابل تشخیص باشد. البته باید توجه داشت که اگر تغییرات بسیار قوی باشد حتی استفاده از اطلاعات گذشته کارا نخواهد بود زیرا شباهتی بین مسائل وجود ندارد که بتوان از اطلاعات گذشته استفاده کرد.

مطالعه کلونی زنبور و ارتباطات بین زنبورها توسط کارل فریش نکات جالبی را در مورد توانایی های ارتباطی آنها روشن کرده است. که دارای خواص جالبی از جمله تقسیم کار مناسب، تخمین زدن کیفیت منبع غذایی و تعادل بهینه از زنبورها به سوی آن منبع غذا، عدم وجود یک سیستم کنترلی مرکزی و تصمیم گیری در مورد مناسب بودن یک منبع با توجه به انرژی لازم برای جمع آوری غذا از آن منبع می باشد. در الگوریتم ارائه شده در این مقاله هر یک از زنبورها سعی می کنند با تعامل مستقیم و به اشتراک گذاشتن اطلاعات خود، بهترین راه حل های بدست آمده را بر اساس قوانین احتمالی انتخاب کند، به عبارت دیگر زنبورها در کنار یکدیگر دارای فاکتور هوشمند می شوند که در غیاب یکدیگر فاقد آن هستند. در این الگوریتم هر یک از زنبورها برای اولین بار یکسری راه حل را به طور تصادفی انتخاب کرده و با اعمال تغییرات محلی بروی راه حل ها، آنها را بهبود می دهند و زنبورهای دیگر با استفاده از یک روش انتخاب حریصانه از بین راه حل های انتخابی و بر اساس احتمالشان، تعدادی را بر اساس بیشترین و کمترین حد از پیش تعریف شده انتخاب و سپس هر گروه را بهبود و یک زنبور را از میان آنها برگزیده و دنبال می کنند. همچنین در این الگوریتم با ارسال دوره ای زنبورهای پیشاهمنگ به منظور جستجوی تصادفی محیط، دامنه جستجو سراسری افزایش می یابد. ادامه این گزارش بدین صورت سازماندهی شده است. بخش ۲ به معرفی کلونی زنبورها در طبیعت، الگوریتم کلونی زنبورهای مصنوعی، الگوریتم زنبور میپردازد. در بخش ۳ الگوریتم پیشنهادی HABC معرفی و در بخش ۴ توابع محک برای محیط های پویا و نتایج آزمایشها انجام شده بروی این توابع ارائه میگردد و بخش پایانی نتیجه گیری میباشد.

۲. کلونی زنبور در طبیعت

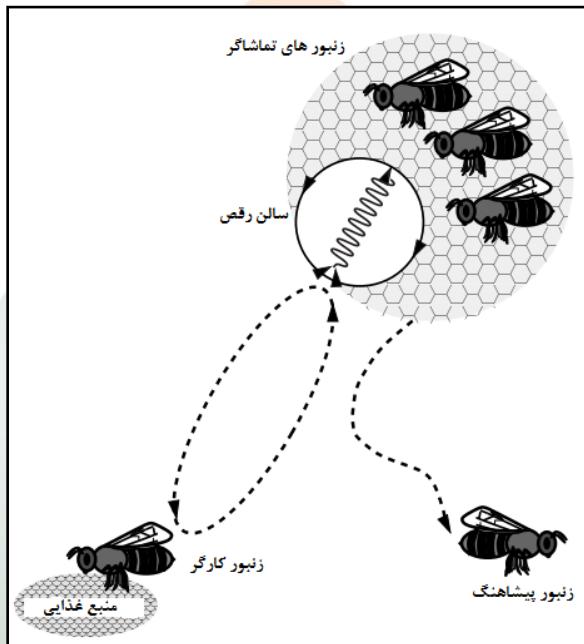
کلونی زنبورها در طبیعت [۷] شامل دو مولفه منابع غذایی^۲ و زنبوران خوراک جو^۳ می باشد. کیفیت منابع غذایی به عوامل زیادی نظر نزدیکی به کندو، میزان سهولت در استخراج شهد و دسترسی به آن بستگی دارد این مولفه ها در شکل (۱) نشان داده شده است.

زنبور خوراک جو به دو دسته زنبور تقسیم می شود که عبارتند از: زنبور بیکار^۴ و زنبور کارگر^۵. زنبور بیکار راجع به منابع غذایی (راه حل ها) هیچگونه دانشی ندارد و به نوع زنبور دیده ور^۶ و زنبور تازه نفس^۷ طبقه بندی می شود. زنبور دیده ور عمل جستجو را بدون هیچ دانشی و به صورت تصادفی شروع می کند و زنبور تازه نفس نیز زنبور بیکاری است که به سالن رقص (به محلی گفته می شود که زنبور کارگر، زنبورهای دیگر را در مورد مکان منبع غذایی مطلع می کند) رفته و جستجو خود را بوسیله استفاده از دانش رقص دم زنبورهای موجود در سالن شروع می کند.

گروه دوم زنبور کارگر می باشد، هرگاه زنبور تازه نفس منبعی غذایی را پیدا کند یک درجه ارتقا پیدا کرده و به زنبور کارگر تبدیل شده و جایگاه منبع غذایی را حفظ می کند و به سمت منبع غذایی پرواز کرده و یک بخش از شهد را برداشته و به کندو برمی گردد. در این صورت ۳ امکان برای مقدار باقی شهد در منبع وجود دارد که عبارتند:

- اگر مقدار شهد به یک سطح کم کاهش پیدا کند زنبور کارگر منبع غذایی را ترک کرده و به یک زنبور بیکار تبدیل می شود.
- اگر به مقدار کافی شهد در منبع غذایی وجود داشته باشد زنبور کارگر می تواند بدون تبادل اطلاعات به خوراک جویی خود ادامه دهد و به منبع غذایی برگردد.
- اگر به مقدار کافی شهد در منبع غذایی وجود داشته باشد زنبور کارگر می تواند به سالن رقص حرکت کند و اطلاعات منبع غذایی خود را به زنبورهای دیگر منتقل کند.

یکی از وقایع مهم در میان زنبورها [۹] تبادل اطلاعات می باشد، زنبورهای عسل از یک سیستم ارتباطی پیچیده استفاده می کنند این سیستم، آنها را قادر می سازد اطلاعاتی در مورد محل و کیفیت منابع غذایی موجود در خارج از کندو بدهست آورند. یک کندو از چندین بخش مجزا تشکیل شده است، که مهمترین بخش آن محل تبادل اطلاعات است که سالن رقص نام دارد، ارتباطات در بین زنبورها در رابطه با کیفیت منابع غذایی با رقص دم ^۸ انجام می گیرد. رقص دم شامل اطلاعاتی می باشد که عبارتند از: جهت قطعه منابع غذایی که از طریق حرکت خورشید در آسمان مشخص می شود و مسافت تا کندو توسط زمان رقص و نرخ کیفیت یا شایستگی از طریق تکرار رقص بدست می آید، البته ذکر این نکته ضروری است که انتخاب منبع توسط زنبورهای تازه نفس، به طور کامل مشخص نیست اما فرض بر این است که این انتخاب به صورت تابعی از کیفیت منبع غذایی می باشد. زنبورهای دیده ورنیز با احتمالی متناسب با کیفیت منبع غذایی اطلاعات خود را به اشتراک می گذارند. در حقیقت به اشتراک گذاشتن این اطلاعات در هنگام رقص دم در طول زمان بیشتر خواهد شد، در نتیجه تعداد زنبورهایی که به هر منبع تخصیص داده می شود متناسب با کیفیت منابع غذایی خواهد بود.



شکل (۱) مولفه های اساسی رفتار خوارک جویی زنبور

۲.۱. الگوریتم کلونی زنبورهای مصنوعی

در الگوریتم کلونی زنبورهای مصنوعی یک کلونی از زنبورهای مصنوعی شامل سه گروه از زنبورها، زنبورهای کارگر، تماشاگر و پیشاہنگ می باشد. نیمه اول کلونی شامل زنبورهای مصنوعی کارگر و نیمه دوم شامل زنبورهای تماشاگر می باشد و برای هر منبع غذایی، فقط یک زنبور کارگر وجود دارد. به عبارت دیگر، تعداد زنبورهای کارگر برابر با تعداد منابع غذایی می باشد. زنبورهای کارگر زمانیکه با یک منبع غذایی متوجه می شوند به یک زنبور پیشاہنگ تبدیل می شوند. جستجو در الگوریتم کلونی زنبورهای مصنوعی به صورت زیر دسته بندی می شود [۸]:

- زنبورهای کارگر یک منبع غذایی در مجاورت منبع غذایی حافظه خود (راه حل قبلی) جستجو می کنند به عبارت دیگر در پی یافتن راه حل بهتر، خود را طبق رابطه (۱) اصلاح می کنند .

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

$$i \neq k, k \in \{1, 2, \dots, SN\}, j \in \{1, 2, \dots, D\}, \phi_{ij} \in [-1, 1]$$

در رابطه (۱)، ϕ_{ij} یک عدد تصادفی در بازه $[0, 1]$ است که تولید موقعیت منابع غذایی همسایه را در اطراف $x_{i,j}$ کنترل می کند. در این رابطه ، BN تعداد زنبورهای کارگر میباشد. متغیر K نیز به صورت تصادفی تولید می شود و می بایست با i متفاوت باشد و هر چه

تفاوت بین $x_{i,k}$ و $x_{i,j}$ کاهش یابد، انحراف از موقعیت $x_{i,j}$ نیز کاهش خواهد یافت. در حقیقت در این رابطه سعی می شود یک بعد از ابعاد یکی از موقعیتها را انتخاب کرده و با توجه به میزان ϕ به سمت آن و یا در خلاف جهت آن حرکت انجام شود.

- زنبورهای کارگر اطلاعات خود را در مورد موقعیت های منابع غذایی در اختیار تماشاگرها در سالن رقص قرار می دهند سپس تماشاگرها یکی از آنها را بر طبق رابطه (۲) انتخاب می کنند.

زنبورهای تماشاگر نیز که در سالن رقص منتظر زنبورهای دیده ور می باشند بعد از ارزیابی، هر کدام با احتمالی متناسب با میزان کیفیت منبع غذایی هر یک از زنبوران دیده ور انتخاب می شوند که این احتمال بوسیله رابطه (۲) بدست می آید.

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad \text{رابطه (۲)}$$

در رابطه (۲)، fit_i میزان شایستگی منبع غذایی متناظر با زنbor آم میباشد.

- زنبورهای تماشاگر یک منبع غذایی در مجاورت منبع غذایی انتخاب شده مرحله قبلی جستجو می کنند رابطه (۱).
- در صورتیکه یک منبع پایان پذیرد زنbor کارگری منبع را ترک کرده به یک پیشاهنگ تبدیل می شود و شروع به جستجوی یک منبع غذایی به صورت تصادفی بر طبق رابطه (۳) می کند. بدین معنی که در یک بهینه محلی قرار داریم بنابراین آن نقطه حذف می شود و یک نقطه جدید به صورت تصادفی تولید میگردد.

$$x_i^j = x_{\min}^j + rand(0,1)(x_{\max}^j - x_{\min}^j) \quad \text{رابطه (۳)}$$

هر چرخه از جستجو شامل ۳ گام می باشد: حرکت زنبورهای کارگر و تماشاگر به سوی منابع غذایی و محاسبه مقدار برآزش آنها (شهد آنها) و تعیین کردن زنبورهای پیشاهنگ و حرکت آنها به طور تصادفی به سوی منابع غذایی ممکن. یک منبع غذایی یک راه حل ممکن برای مسئله بوده و مقدار شهد منبع غذایی مربوط به کیفیت راه حل ارائه شده می باشد. در این روش تماشاگرها بوسیله یک مکانیزم انتخاب منابع غذایی متناسب با انتخاب می کنند و در انتهای هر چرخه زنبورهای پیشاهنگ برای جستجوی تصادفی و بدون هیچ دانشی انتخاب می شود هر چند که این زنبورها دارای کیفیت بالایی نیستند و هزینه زیادی نیز صرف نمی کند ولی گاهی اوقات می توانند موقعیت های غنی منابع غذایی که ناشناخته می باشند را پیدا کنند. البته ممکن است در هر چرخه زنbor پیشاهنگ نداشته باشیم زیرا می بایست منابع غذایی (کارگرها) که در چندین چرخه در الگوریتم بهبود نیافرته اند را تعیین کرده و بوسیله تولید منابع غذایی جدید تصادفی با توزیع یکتواخت جایگزین کنیم. تعداد چرخه ای که راه حل در آن بهبود نیافرته است بوسیله متغیر $limit$ مشخص می شود. این مقدار بوسیله رابطه (۴) محاسبه می شود.

$$limit = SN * D \quad \text{رابطه (۴)}$$

با توجه به توضیحات فوق پارامترهای کنترلی استفاده شده در الگوریتم کلونی زنبورهای مصنوعی شامل تعداد منابع غذایی که برابر تعداد زنبورهای کارگر، تعداد زنبورهای دیده ور، تعداد زنبورهای پیشاهنگ و ماکریتم تعداد چرخه الگوریتم می باشد که می بایست از قبل مشخص شود. شبه کد الگوریتم کلونی زنبورهای مصنوعی در شکل (۲) ارائه گردیده است [۸][۹].

کنفرانس داده کاوی ایران

۱. **Initialize** population of solution x_{ij} $i=1\dots SN$, $j=1\dots D$
۲. **Evaluate** the population
۳. **cycle=۱**
- Repeat**
۴. For the employed bees by $v_{i,j}$ using (۱) and evaluate them
۵. Produce new solutions
۶. Apply the greedy selection process
۷. Calculate probability values $P_{i,j}$ for the solution $x_{i,j}$ by (۲)

۸. Produce new solutions $v_{i,j}$ for the onlooker from the solution $x_{i,j}$ selected depending on and evaluate $P_{i,j}$ them
 ۹. Apply the greedy selection process
 ۱۰. Determine the abandoned solution for scot, if exists replace it with a new randomly produced solution $x_{i,j}$ by (۳)
 ۱۱. Memorize the best solution achieved so far
 ۱۲. cycle=cycle+1;
Until cycle=MCN

شکل (۲) شیوه کد الگوریتم کلونی زنبورهای مصنوعی

۲.۲. الگوریتم زنبور^۹

الگوریتم زنبور [۱۰][۱۱] یک الگوریتم بهینه یابی می باشد که از رفتار خوارک جویی زنبورهای عسل برای یافتن راه حل بهینه الهام گرفته شده است. این الگوریتم نیاز به تنظیم تعدادی از پارامترها قبل از آغاز دارد که در جدول (۱) نشان داده شده است.

جدول (۱) پارامترهای الگوریتم زنبور

نماد	توصیف
n	تعداد زنبورهای پیشاہنگ
m	تعداد زنبورهای انتخاب شده (سایت ها) از میان n سایت
e	تعداد زنبورهای انتخاب شده (سایت ها) از میان m سایت
nep	تعداد زنبورهای انتخاب شده (فرستاده شده) برای بهترین e سایت
m-e	تعداد زنبورهای انتخاب شده (سایت ها) باقی مانده
nsp	تعداد زنبورهای استخدام شده (فرستاده شده) برای m-e سایت
ngh	سایز اولیه قطعات که شامل سایت ها و همسایه هایشان می باشد.

این الگوریتم با n زنبور پیشاہنگ که به طور تصادفی در فضای جستجو توزیع شده اند شروع می شود. سپس برازش هر یک از زنبورها بوسیله تابع هدف معین و براین اساس از بزرگ به کوچک مرتب می شوند.

در مرحله سوم m زنبور برتر با توجه به مقدار برازش از لیست رتبه بندی شده انتخاب می شود و e سایت انتخاب می شود این e سایت بهترین سایت ها می باشدند به عبارت دیگر e و سایت (منابع غذایی)، مکان هایی می باشند که قرار است زنبورهای تازه نفس به سمت آنها حرکت یا به عبارت دیگر آنها را دنبال کنند. البته در زندگی طبیعی زنبور تعداد بیشتر زنبورها به سمت بهترین مکانها حرکت می کنند که در این الگوریتم e بهترین مکان می باشد.

در مرحله بعدی مقدار ngh انتخاب می شود این مقدار اجازه می دهد که هر کدام از این سایت ها با توجه به مقدار ngh بتوانند محدوده مجاور خود را جستجو و از آن فراتر نروند به عبارت دیگر اگر هر کدام از سایت هایک نقطه در فضای جستجو باشند این سایت تا شعاع ngh می تواند توسعه پیدا کند.

در فاز بعدی زنبورهای تازه نفس یا تماشاگر به سمت e و m-e سایت حرکت می کنند که تعداد این زنبورها با متغیرهای nep و nsp مشخص می شود که نشان دهنده این مسئله است که تعداد زنبورهایی که به سمت e حرکت می کنند بیشتر از تعداد زنبورهایی است که به سمت m-e سایت حرکت می کنند به عبارت دیگر این سایت غنی تر از دیگری است. این زنبورها در هر سایت توزیع می شوند به طوری که از شعاع ngh هر سایت تجاوز نمی کنند.

در آخرین گام بهترین زنبور در هر سایت انتخاب و وارد نسل بعدی می شود و به تعداد $n-m$ زنبور نیز به طور تصادفی تولید و به همراه زنبورهای انتخاب شده وارد نسل بعدی می شود. این مراحل تا زمانی در چرخه الگوریتم تکرار می شود تا به نتایج ارضاء کننده دست یابیم شبه کد این الگوریتم در شکل (۳) ارائه گردیده است [۱۱].

۱. Initialize population with random solutions.
۲. Evaluate fitness of the population.
۳. While (stopping criterion not met)
 - //Forming new population.
 ۴. Select sites for neighbourhood search.
 ۵. Recruit bees for selected sites (more bees for best e sites) and evaluate fitnesses.
 ۶. Select the fittest bee from each patch.
 ۷. Assign remaining bees to search randomly and evaluate their fitnesses.
۸. End While.

شکل (۳) شبه کد الگوریتم زنبور

۴. الگوریتم پیشنهادی HABC

این الگوریتم ترکیبی از کلونی زنبور مصنوعی و زنبورها 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 نشاندهنده یک موقعیت یا منبع غذایی میباشدند. در ابتدا تعدادی زنبور کارگر به طور تصادفی در فضای جستجو مقداردهی شده و شایستگی آن را محاسبه کرده، در صورتی که میزان شایستگی آن از راه حل قدیمی بیشتر باشد، راه حل جدید انتخاب میگردد و راه حل قدیمی فراموش میشود در غیر این صورت همان راه حل قدیمی باقی خواهد ماند. این تغییرات با انتخاب تصادفی یک زنبور از میان تمامی زنبورها و حرکت به سمت آن انجام میگیرد. نتایج بدست آمده طبق الگوریتم زنبور رتبه بندی می شود این الگوریتم تعداد جمعیت کارگر نزدیک به تعداد زنبورها می باشد به این علت که در رتبه بندی زنبورهای زیادی حذف نشوند. زنبورهای تماشاگر که در کندو منتظر زنبورهای رتبه بندی شده یا همان زنبورهای کارگرها هستند با توجه به دو متغیر nep و $m-e$ زنبورهای کارگر را انتخاب می کنند به عبارت دیگر به تعداد nep زنبور تماشاگر به هر یک از e سایت و به تعداد nsp زنبور تماشاگر به سایت فرستاده می شوند در صورتیکه سایت (زنبور کارگر موجود در سایت) انتخاب شده برازندگی زیادی داشته باشد به تعداد nep و در صورت برازندگی کم تعداد کمتری یا به تعداد nsp زنبور تماشاگر، آن را انتخاب می کند و این زنبورها بوسیله رابطه اصلاحی در الگوریتم زنبورهای مصنوعی نسبت به همه زنبورهای تماشاگر موجود در آن سایت مرتبط با خود بروز می شوند سپس بوسیله یک انتخاب حریصانه بهترین زنبورهای تماشاگر در هر سایت انتخاب می شوند. در این روش شناس بهبود هر تماشاگر بیشتر از روشهای قبل است به این علت که هر زنبور تماشاگر این اجازه را می دهد تا به تعداد nsp و nep زنبور کارگر را دنبال کند و آنها را بهبود دهد. به عبارت دیگر بتواند یک زنبور کارگر را چندین بار تغییر داده و بهترین را بردارد. بعد از انتخاب بهترین تماشاگر در هر سایت به تعداد m زنبور به اضافه $n-m$ زنبور با مقادیر تصادفی به چرخه بعدی وارد می شود. شبه کد این الگوریتم در شکل (۴) نشان داده شده است.

Initialize n scout bees with random solutions.

Do

For each bee in colony

Calculate fitness value of the bee f_b

End For

For each bee

Produce a **modification** on the Position of The bee

End For

Apply the greedy selection process

Apply the ranking evaluation bees(Then the array will be reordered based on the evaluation from the higher to the lower Value)

The m sites will be **selected randomly** (the best evaluation to m scout bee) from n

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

Sites and choosing **nsp** bees randomly which their number is less than nep,to be sent to m-e sites

For each recruit bee

Produce a **modification** on the Position of the bee

End For

Apply the greedy selection process

Choosing the best bee from each site (the highest fitness) to form the next bee population

Assign remaining bees to search randomly and evaluate their Fitnesses.

While maximum iterations OR(stopping criterion not met)

شکل (۴) شبیه کد الگوریتم ترکیبی

۴. آزمایش ها

در آزمایشهای انجام شده هدف بررسی و مقایسه الگوریتم های تکاملی مبتنی بر حافظه، SOS, RPSO, mQSO, Adaptive mQSO و HABC در محیط های پویا به کمک توابع قله های متحرک و توابع قله های در حال نوسان می باشد از آنجا که در محیط های پویا بهترین راه حل بدست آمده معیار مفیدی برای ارزیابی و گزارش نمی باشد، از معیارهای Current _ Error , Offline _ Error جهت مقایسه و نتیجه گیری استفاده شده است [۴].

۴.۱. تابع چند قله ای متحرک^{۱۰}

جهت پدید آوردن یک محیط پویا از یک تابع محک مناسب، چند قله ای متحرک [۳][۵] استفاده شده است که شامل یک فضای چند بعدی ، چندین قله با ارتفاع و پهنای متفاوت است، که ارتفاع، پهنا و موقعیت هر قله در طول زمان در حال تغییر است. تابع چند قله ای متحرک در نظر گرفته شده شامل m قله در n بعد با پارامترهای حقیقی است و هدف، یافتن بیشینه در هر زمان در میان m قله تا تغییر بعدی می باشد که به صورت رابطه (۵) تعریف می شود:

$$F(\vec{x}, t) = \text{Max}(B(\vec{x}), \text{Max}_{i=1..m} P(\vec{X}, H_i(t), W_i(t), \vec{p}(t))) \quad (5)$$

در این تابع $B(\vec{x})$ یک شمای پایه ثابت در زمان و p تابع چند قله ای است که در هر Δe ارزیابی، ارتفاع و پهنای قله ها بوسیله افزودن یک عدد تصادفی گوسین با میانگین صفر و واریانس σ و مکان هر قله بوسیله افزودن بردار v با طول ثابت s (سختی) طبق رابطه (۶) تغییر می کند. در این تابع می توان پیچیدگی را با افزایش و کاهش Δe در زمان تغییر داد.

$$\begin{cases} \sigma \in N(0, 1) \\ h_i(t) = h_i(t-1) + height_severity \cdot \sigma \\ w_i(t) = w_i(t-1) + width_severity \cdot \sigma \\ \vec{p}_i(t) = \vec{p}_i(t-1) + \vec{v}_i(t) \end{cases} \quad (6)$$

بردار v را می توان وابسته به تغییر قبلی آن ایجاد کرد که در این صورت تغییر موقعیت قله ها همسو با تغییرات قبل آن می شود و یا به صورت تصادفی آنرا ایجاد نمود که موجب می شود موقعیت قله ها به صورت تصادفی تغییر کند و هیچگونه وابستگی به تغییر قبلی نداشته باشد. مقدادیر پارامترهای برای انجام آزمایش ها طبق جدول (۲) در نظر گرفته شده است.

جدول (۲) تنظیم پارامترهای تابع قله های متحرک

پارامتر	مقدار پیش فرض
λ فاکتور وابستگی	.
P تعداد قله ها	۱۰
f فرکانس تغییرات	هر ۵۰۰۰ ارزیابی
سختی ارتفاع	۷.۰
سختی پهنا	۱.۰
شکل قله ها	مخروطی ^{۱۱}
S طول تغییر	۱.۰
D تعداد ابعاد	۵
H مقدار مینیمم و ماکزیمم ارتفاع	[۳۰.۰, ۷۰.۰]
W مقدار مینیمم و ماکزیمم پهنا	[۱, ۱۲]
مقدار اولیه ارتفاع	۵۰.۰
A محدوده فضای جستجو	[۰, ۱۰۰]

۴.۲. تابع قله های در حال نوسان^{۱۲}

تابع پیچیده دیگری که در آزمایش ها مورد استفاده قرار می گیرد تابع قله های در حال نوسان [۳][۵] می باشد که در آن L منظره (معمولاً L=۲ که هر کدام حاوی m قله منتخب تصادفی که در تابع قله های متحرک توصیف شده، می باشد هر منظره بر طبق تابع کسینوسی رابطه (۷) نوسان می کند.

$$\begin{cases} f_i(t) = \omega(t)f_i(0) \\ \omega(t)f_i(0) = 0.5 \cos\left(\frac{2t\pi}{step} + 2\pi \frac{i-1}{l}\right) + 0.5 \\ i = 1 \dots l \end{cases} \quad (7)$$

با step تعداد گام های میانی در یک سیکل تعریف می شود. در نتیجه مسئله بین L منظره نوسان می کند. در آزمایشها مقادیر پارامترها طبق جدول (۳) در نظر گرفته شده است. با توجه با اینکه شکل قله ها بصورت ناقوسی در نظر گرفته شده اساساً "آزمایش رفتار مسئله کوله پشتی در حال تغییر را توصیف می کند.

جدول (۳) تنظیمات پیش فرض استفاده شده در آزمایشها برای محک قله های در حال نوسان

پارامتر	مقدار پیش فرض
تعداد مناظر	۲
P تعداد قله ها	۱۰
f فرکانس تغییرات	هر ۱۰۰۰ ارزیابی
تعداد گامهای هر چرخه	۱۰
شکل قله ها	ناقوسی ^{۱۳}
S طول تغییر	۱.۰
D تعداد ابعاد	۵
H مقدار مینیمم و ماکزیمم ارتفاع	[۳۰.۰, ۷۰.۰]
W مقدار مینیمم و ماکزیمم پهنا	[۰.۰۰۰۸, ۰.۰۰۱]
A محدوده فضای جستجو	[۰, ۱۰۰]

۴.۳. پارامترهای تنظیم شده الگوریتم ها در آزمایش ها

در این الگوریتم تعداد زن ها در هر کروموزوم (۱۰۰ کروموزوم) ۵ در نظر گرفته شده و به منظور تولید فرزندان و انتخاب والدین از روش بازترکیبی دو نقطه ای با نرخ ۰.۶ و جهش گوسین با نرخ ۰.۲ و روش انتخاب چرخ گردان استفاده شده است. این الگوریتم در صورت استفاده از حافظه، تعداد ۱۰ فرد را در حافظه ذخیره می کند.

در الگوریتم پیشنهادی نیز همانند پارامترهایی که در جدول (۱) توصیف شده مقادیری می باشد قبل از انجام الگوریتم تعریف شود که این مقادیر در آزمایش الگوریتم پیشنهادی به این صورت در نظر گرفته شده است تعداد راه حل های (افراد) ۱۰۰، تعداد زبوروهای انتخاب شده از میان ۱۰۰ سایت ۹۸، تعداد زبوروهای انتخاب شده از میان ۹۸ زبورو بوسیله یک روش انتخاب و مقادیر nsp و nep به ترتیب ۳ و ۲ در نظر گرفته شده و در پایان الگوریتم ۱۰ زبورو به طور تصادفی و در صورت مساوی بودن برخی زبوروهای موجود اضافه می شوند.

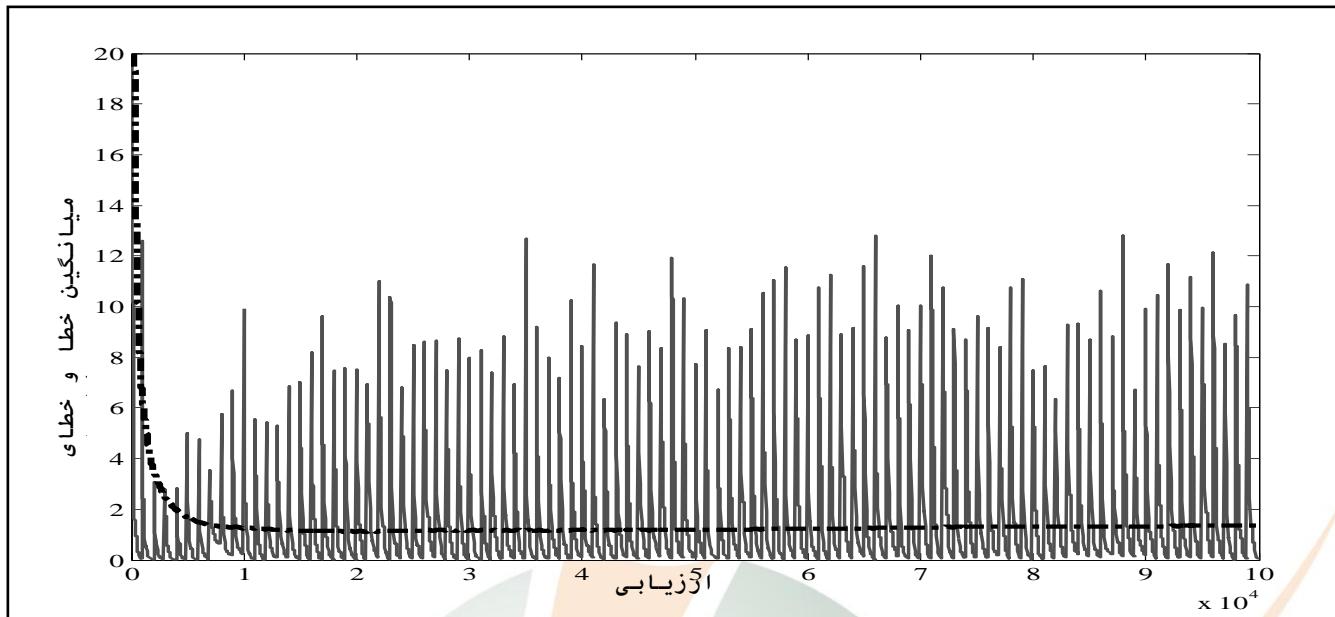
در آزمایشهای مربوط به ترکیب الگوریتم های اتوماتای سلولی و PSO مقدار فاکتورهای یادگیری c_1 و c_2 با مقدار ۱.۴۹۶۱۸۰ و وزن میانی ۰.۷۲۹۸۴۴ تنظیم شده است، ماکریم سرعت برای یک ذره و شعاع همسایگی اتوماتای سلولی و شعاع جستجوی محل تصادفی در همه آزمایشهای ۰.۵ در نظر گرفته شده است. یک اتوماتای سلولی پنج بعدی با ۱۰ سلول با همسایگی مور و شعاع ۲ سلول در فضای جستجو در نظر گرفته شده و تعداد ذرات ۱۰۰ عدد تنظیم شده است. همچنین در آزمایشهای مربوط به $mQSO_{m(n+k)}$ تعداد زیر گروه و n تعداد ذره استاندارد و k تعداد ذره های کوانتوسی می باشد.

۴.۴. آزمایش های تابع قله های متحرک

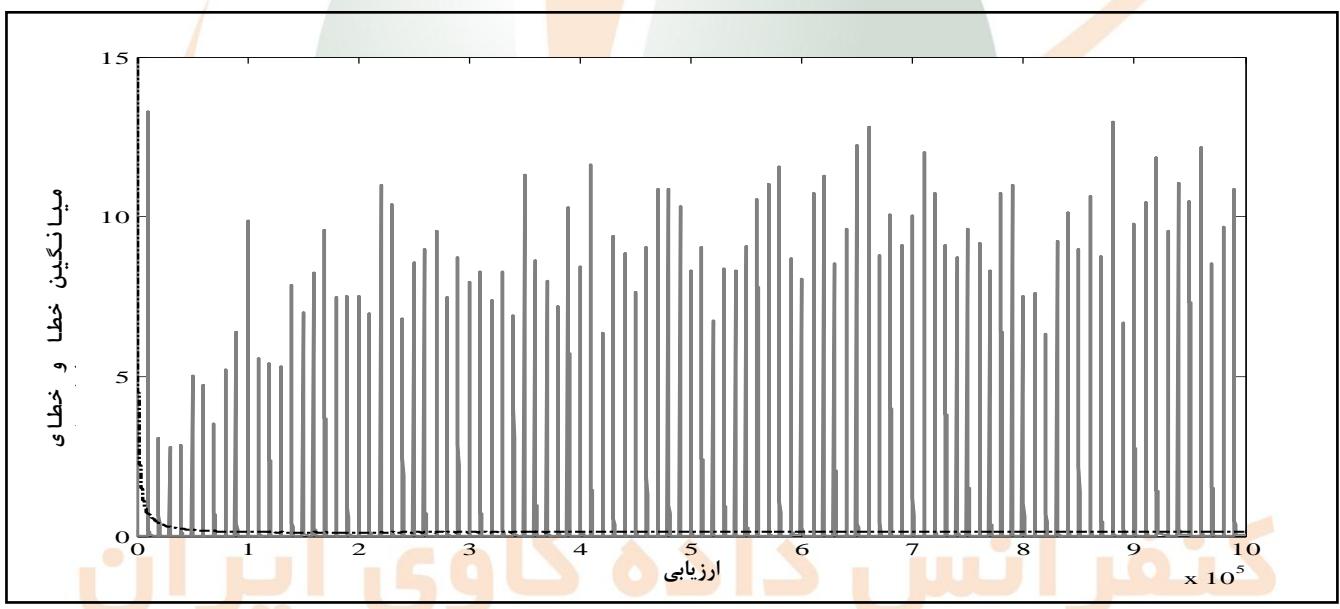
در این آزمایش ها کارایی الگوریتم CABC در محیطی پویا توسط تابع قله های متحرک با فرکانس تغییرات و تعداد قله های متفاوت ارزیابی شده است و بكمک معیار Offline error با الگوریتم های PSO [۱].RPSO [۲]، Adaptive mQSO [۳]، Adaptive cellular [۴] و الگوریتم های تکاملی مقایسه گردیده است. تعداد قله ها بین ۱ تا ۲۰۰ و فرکانس تغییرات بین هر ۵۰۰ تا ۱۰۰۰۰ ارزیابی تغییر می کند (منظور از فرکانس تغییرات پایین یعنی تعداد ارزیابی هایی که بین تغییرات وجود دارد زیاد می باشد مثل $f=500$ و منظور از فرکانس تغییرات بالا یعنی تعداد ارزیابی ها بین تغییرات کم می باشد مثل $f=5000$) از آزمایش های انجام شده نتایج زیر حاصل شده است:

- در شکل های (۵ و ۶) میانگین خطای لحظه ای روش پیشنهادی HABC در دو فرکانس تغییر مختلف ۱۰۰۰ و ۱۰۰۰۰ ارزیابی با ۱۰ قله نمایش داده شده است. با مقایسه این دو شکل مشخص می گردد که الگوریتم در فرکانس تغییر پایین نسبت به فرکانس بالا بهتر عمل می کند به این دلیل که میانگین خطای در فرکانس تغییر پایین کمتر از فرکانس تغییر بالا می باشد. تقریباً در هر دو نمودار با گذشت ۱۰۰۰۰ ارزیابی میزان میانگین خطای به یک مقدار پایدار می رسد این امر ناشی از سرعت یافتن اکثر قله ها در الگوریتم و حفظ آنها به صورت ضمنی در جمعیت می باشد.

کنفرانس داده کاوی ایران

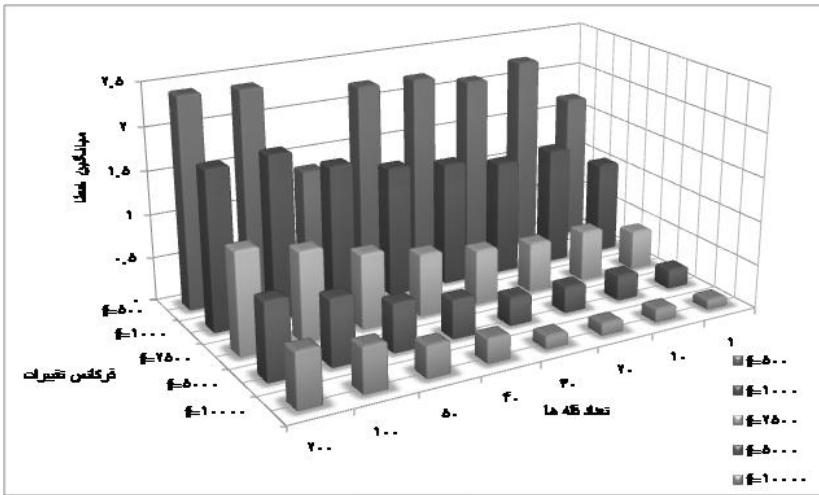


شکل (۵) میانگین خطای لحظه‌ای در فرکانس تغییر ۱۰۰۰ ارزیابی با ۱۰ قله



شکل (۶) میانگین خطای لحظه‌ای در فرکانس تغییر ۱۰۰۰۰ ارزیابی با ۱۰ قله

به منظور بررسی دقیق تر رفتار الگوریتم میانگین خطای مختلف با قله های متفاوت محاسبه و در شکل (۷) نمایش داده است. با توجه به آزمایش ها مشخص می گردد از فرکانس تغییر ۱۰۰۰ به بالا میانگین خطای افزایش تعداد قله ها افزایش پیدا می کند این امر در فرکانس تغییر ۵۰۰ متفاوت می باشد چون زمان کافی برای یافتن قله ها و همگرایی وجود ندارد. ضمناً نتایج نشان می دهد با کاهش فرکانس تغییرات از ۵۰۰ ارزیابی به سمت ۱۰۰۰۰ ارزیابی بین دو تغییر به علت ایجاد فرصت کافی برای همگرایی و یافتن قله ها میزان میانگین خطای کاهش می یابد.



شکل (۷) طیف میانگین خطای توجه به فرکانس تغییرات و تعداد قله ها در الگوریتم پیشنهادی HABC

- در محیط هایی که دارای چند قله می باشد زمانیکه قله ها هر ۵۰۰، ۱۰۰۰، ۲۵۰۰ و ۵۰۰۰ ارزیابی تغییر می کنند روش های cellular PSO و adaptive cellular PSO نسبت به بقیه روشها خوب عمل می کنند ولی روش HABC بسیار بهتر از دیگر روشها می باشد. به این دلیل که در الگوریتم پیشنهادی HABC فرایند اکتشافی و استخراجی توسط زنبورهای کارگر و تماشاگر با هم انجام می شوند و حتی بوسیله زنبورهای پیشنهادگ مناطق ناشناخته بوسیله زنبورهای کارگر و تماشاگر کشف می شود جداول (۴ تا ۸).

- روش پیشنهادی HABC در تمام فرکانس ها به خصوص در فرکانس ۱۰۰۰۰ ارزیابی نسبت به روش های PSO که ضعیف تر MQSO عمل کرده اند بسیار بهتر عمل کرده است جداول (۴ تا ۸).
- در محیط های تک قله ای همه روش ها به جز روش پیشنهادی HABC نسبت به روش RPSO ضعیف تر عمل نموده اند. عملکرد مناسب روش HABC ناشی از رفتار پیشنهادگ ها می باشد که در هر نسل جستجوی سراسری مناسبی را انجام می دهند جداول (۴ تا ۸).

جدول (۴) برای Offline error $F=500$

HABC	Adaptive Cellular PSO[۶]	Adaptive mQSO [۲]	Cellular PSO [۶]	mQSO $10(5+5^4)$ [۱]	mQSO $10(10+10^4)$ [۱]	RPSO[۱۲]	تعداد قله ها
۱.۷۰	۱۲.۱۹	۱۴.۰۷	۱۳.۴۶	۳۳.۶۷	۴۴.۴۸	۵.۲۰	۱
۲.۲۵	۹.۱۸	۹.۴۲	۹.۳۵	۹.۶۲	۱۵.۱۵	۱۸.۱۱	۱۰
۲.۱۱	۸.۷۸	۹.۵۹	۸.۸۴	۹.۰۷	۱۳.۰۴	۱۷.۸۰	۲۰
۲.۲۳	۸.۶۴	۹.۴۴	۸.۸۱	۸.۸۰	۱۲.۴۸	۱۷.۴۲	۳۰
۲.۲۴	۸.۶۸	۹.۴۴	۸.۹۴	۸.۸۵	۱۲.۲۲	۱۶.۷۹	۴۰
۱.۳۶	۸.۴۹	۹.۱۳	۸.۶۲	۸.۷۲	۱۲.۱۸	۱۶.۴۲	۵۰
۲.۴۰	۸.۲۲	۸.۸۱	۸.۵۴	۸.۵۴	۱۱.۵۴	۱۵.۰۰	۱۰۰
۲.۴۳	۷.۸۹	۸.۲۱	۸.۲۸	۸.۱۹	۱۱.۲۵	۱۴.۰۰	۲۰۰

جدول (۵) برای Offline error $F=1000$

HABC	Adaptive Cellular PSO[۶]	Adaptive mQSO [۲]	Cellular PSO [۶]	mQSO $1 \cdot (5+5^q)$ [۱]	mQSO $1 \cdot (10+10^q)$ [۱]	RPSO[۱۲]	تعداد قله ها
۱.۰۷	۵.۸۳	۶.۶۰	۶.۷۷	۱۸.۶۰	۲۱.۶۸	۲.۴۰	۱
۱.۳۵	۵.۲۹	۵.۶۴	۵.۱۹	۵.۷۱	۷.۶۶	۱۵.۷۷	۱۰
۱.۳۱	۵.۴۹	۵.۹۵	۵.۲۳	۵.۸۵	۷.۲۵	۱۵.۶۱	۲۰
۱.۴۱	۵.۴۲	۵.۹۷	۵.۳۳	۵.۸۱	۷.۱۹	۱۴.۸۸	۳۰
۱.۴۸	۵.۳۲	۶.۱۲	۵.۶۱	۵.۷۰	۷.۱۹	۱۴.۵۳	۴۰
۱.۶۲	۵.۲۵	۵.۹۸	۵.۵۵	۵.۸۷	۷.۱۴	۱۴.۱۶	۵۰
۱.۸۶	۵.۰۹	۵.۷۸	۵.۵۷	۵.۸۳	۶.۹۴	۱۲.۵۳	۱۰۰
۱.۸۲	۴.۵۹	۵.۵۴	۵.۵۰	۵.۵۴	۶.۹۸	۱۱.۵۸	۲۰۰

جدول (۶) برای Offline error F=۲۵۰۰

HABC	Adaptive Cellular PSO[۶]	Adaptive mQSO [۲]	Cellular PSO [۶]	mQSO $1 \cdot (5+5^q)$ [۱]	mQSO $1 \cdot (10+10^q)$ [۱]	RPSO[۱۲]	تعداد قله ها
۰.۴۵	۲.۰۰	۲.۴۸	۴.۱۵	۷.۶۴	۹.۸۸	۱.۰۰	۱
۰.۵۸	۳.۰۳	۲.۹۱	۲.۸۲	۳.۱۲	۴.۳۸	۱۳.۷۲	۱۰
۰.۵۷	۳.۱۷	۳.۴۰	۳.۴۱	۳.۵۸	۴.۳۴	۱۳.۸۹	۲۰
۰.۶۲	۳.۲۲	۳.۴۷	۳.۶۲	۳.۶۳	۴.۳۶	۱۳.۰۱	۳۰
۰.۷۱	۳.۲۲	۳.۵۶	۳.۸۴	۳	۴.۳۷	۱۲.۴۹	۴۰
۰.۸۵	۳.۳۰	۳.۵۶	۳.۸۶	۳.۶۳	۴.۳۶	۱۱.۹۶	۵۰
۱.۰۲	۳.۳۵	۳.۵۳	۴.۱۰	۳.۵۸	۴.۲۱	۱۰.۷۳	۱۰۰
۱.۱۶	۳.۲۹	۳.۳۷	۳.۹۷	۳.۳۰	۴.۰۴	۹.۶۳	۲۰۰

جدول (۷) برای Offline error F=۵۰۰۰

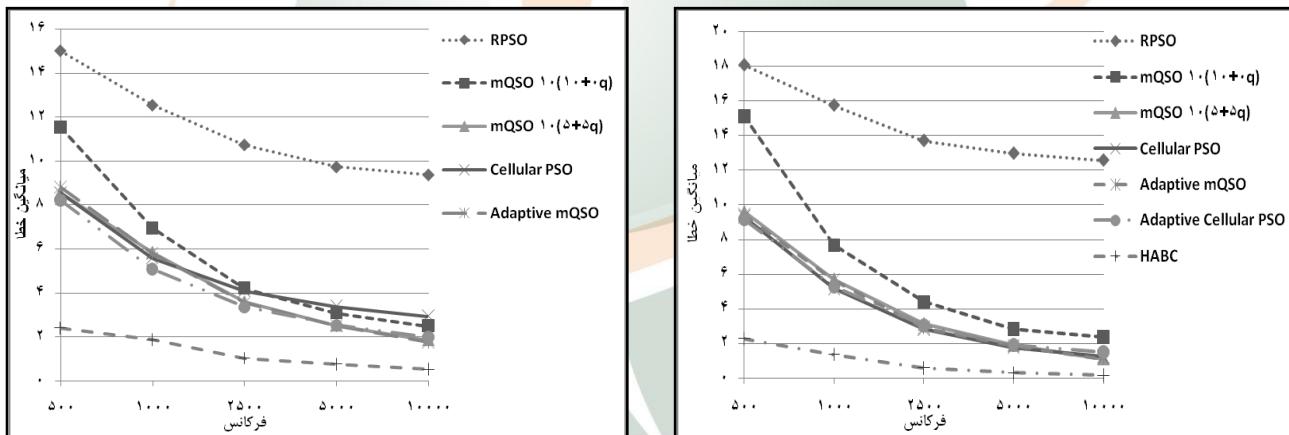
HABC	Adaptive Cellular PSO[۶]	Adaptive mQSO [۲]	Cellular PSO [۶]	mQSO $1 \cdot (5+5^q)$ [۱]	mQSO $1 \cdot (10+10^q)$ [۱]	RPSO[۱۲]	تعداد قله ها
۰.۲۲	۰.۸۷	۱.۰۹	۲.۵۴	۳.۸۲	۵.۱۷	۰.۵۶	۱
۰.۲۹	۱.۹۱	۱.۸۵	۱.۷۶	۱.۹۱	۲.۸۱	۱۲.۹۸	۱۰
۰.۳۰	۲.۲۶	۲.۱۸	۲.۵۹	۲.۵۶	۳.۲۲	۱۲.۷۹	۲۰
۰.۳۲	۲.۲۵	۲.۳۶	۲.۹۵	۲.۶۸	۳.۲۹	۱۲.۳۵	۳۰
۰.۴۴	۲.۴۱	۲.۴۲	۳.۱۱	۲.۶۵	۳.۲۴	۱۱.۳۷	۴۰
۰.۵۵	۲.۴۳	۲.۵۳	۳.۲۲	۲.۶۳	۳.۲۷	۱۱.۳۴	۵۰
۰.۷۵	۲.۵۳	۲.۵۰	۳.۳۹	۲.۵۲	۳.۰۸	۹.۷۳	۱۰۰
۰.۸۷	۲.۴۶	۲.۳۶	۳.۳۶	۲.۳۶	۲.۸۹	۸.۹۰	۲۰۰

جدول (۸) برای Offline error F=10000

HABC	Adaptive Cellular PSO[۶]	Adaptive mQSO [۲]	Cellular PSO [۶]	mQSO ۱۰(۵+۵q) [۱]	mQSO ۱۰(۱۰+۰q) [۱]	RPSO[۱۲]	تعداد قله ها
۰.۱۱	۰.۴۰	۰.۴۹	۱.۵۳	۱.۹۰	۳.۵۶	۰.۵۳	۱
۰.۱۵	۱.۴۹	۱.۱۰	۱.۲۲	۱.۱۰	۲.۳۴	۱۲.۵۸	۱۰
۰.۱۴	۱.۵۷	۱.۵۰	۲.۲۰	۱.۸۴	۲.۸۲	۱۲.۵۹	۲۰
۰.۱۵	۱.۷۷	۱.۶۴	۲.۶۰	۲.۰۰	۲.۹۱	۱۱.۵۵	۳۰
۰.۲۸	۱.۸۶	۱.۷۰	۲.۷۳	۱.۹۹	۲.۸۵	۱۱.۰۴	۴۰
۰.۳۶	۱.۹۱	۱.۷۱	۲.۸۴	۱.۹۹	۲.۷۸	۱۰.۶۲	۵۰
۰.۵۱	۱.۹۸	۱.۷۵	۲.۹۳	۱.۸۵	۲.۴۸	۹.۳۷	۱۰۰
۰.۶۲	۲.۰۵	۱.۶۶	۲.۸۸	۱.۷۱	۲.۳۲	۸.۳۰	۲۰۰

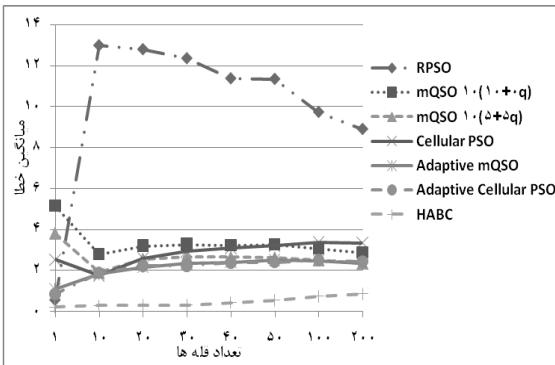
به منظور مقایسه رفتار الگوریتم ها تغییرات در فرکانس های مختلف با تعداد قله های ۱۰ و ۱۰۰ صورت گرفته است و نتایج در شکل های

(۸ و ۹) نمایش داده شده است با توجه به نمودارها مشخص گردیده که با کاهش فرکانس میانگین خطای همه روشها کاهش می یابد و با افزایش تعداد قله ها میانگین خطای همه روشها به جز روش پیشنهادی HABC کاهش می یابد به این دلیل که در روش HABC از نوعی حافظه ضمنی استفاده شده است و با افزایش تعداد قله نگهداری همه قله ها با حفظ همگرایی جمعیت امکان پذیر نیست.

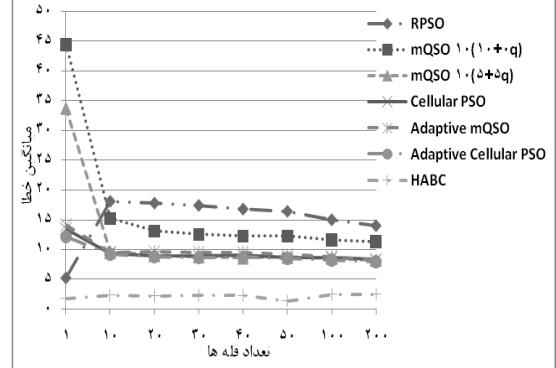


شکل (۸) نمودار میانگین خطای تغییر مختلف با تعداد ۱۰ قله
شکل (۹) نمودار میانگین خطای تغییر مختلف با تعداد ۱۰۰ قله

همانطور که در شکل (۱۰ و ۱۱) مشخص گردیده میانگین خطاهای الگوریتم ها با تعداد قله های مختلف در دو فرکانس تغییر ۵۰۰ و ۵۰۰۰ ارزیابی نمایش داده شده است این نمودارها مشخص می کنند که در اکثر روشها با افزایش تعداد قله ها میزان خطای افزایش پیدا می کند ولی زمانی که تعداد قله ها بسیار زیاد می شود به علت همپوشانی قله ها و نزدیک بودن قله ها با یکدیگر میانگین خطای کاهش می یابد. الگوریتم ها در فرکانس تغییرات ۵۰۰ و ۵۰۰۰ ارزیابی رفتار مشابهی دارند با این تفاوت که با کاهش فرکانس تغییرات میانگین خطای کاهش پیدا می کند.



شکل (11) نمودار میانگین خطای مختلف قله های با تعداد قله های مختلف



شکل (10) نمودار میانگین خطای مختلف و فرکانس تغییر ۵۰۰۰ و فرکانس تغییر ۵۰۰۰

۴.۴. تابع قله های در حال نوسان

در قله های در حال نوسان یک محیط بسیار پویا جهت بررسی الگوریتم ها ایجاد می شود در این تابع بدون تغییر موقعیت قله ها، ارتفاع و پهنای قله ها هر ۱۰۰۰ و ۱۰۰۰۰ ارزیابی از یک منظره به سمت منظره دیگر نزدیک می گردد و با گذشت ۱۰ مرحله مجدداً به موقعیت منظره اولیه باز می گردد. مقادیر آزمایشها در جدول (۹ تا ۱۲) ارائه گردیده است و نتایج زیر حاصل شده است.

- در فرکانس تغییر کم (طول چرخه ۱۰۰۰۰ ارزیابی) روش پیشنهاد شده HABC در مقایسه با روش های حافظه دار که در این نوع مسائل که منظره مجدداً تکرار می شود کاربرد دارند میانگین خطای نسبتاً بالای داشته و در بعضی موارد نتایج بهتر نیز بوده است (مثل روش SEAM و همه روشها در شمای حافظه واریانسی) ولی در مقایسه با روش های بدون حافظه نتایج بسیار راضی کننده می باشد و میانگین خطای پایینی حاصل گردیده است جدول (۹ و ۱۰).

جدول (۹) میانگین خطای روش های با حافظه به جز روش پیشنهادی با طول چرخه ۱۰۰۰۰ ارزیابی بعد از ۵۰۰۰۰۰ ارزیابی

HABC	Mem/2search	Mem/search	P³RI²⁵m	P³m	RI²⁵m	SEAm	شمای حافظه
11.۲۴	۷.۷۴	۴.۶۳	۱۰.۳۳	۱۱.۳۰	۷.۵۱	۱۴.۱۸	کمترین فاصله
	۸.۴۷	۴.۶۲	۱۲.۰۴	۱۱.۵۹	۷.۶۷	۱۴.۱۷	کمترین فاصله ۲
	۹.۸۱	۵.۰۷	۱۱.۹۱	۱۱.۴۴	۷.۶۸	۱۴.۲۴	شبیه ترین
	۱۵.۶۷	۱۳.۶۵	۲۸.۰۱	۱۸.۱۸	۱۴.۰۱	۱۸.۹۹	واریانس

جدول (۱۰) میانگین خطای روش های بدون حافظه با طول چرخه ۱۰۰۰۰ ارزیابی بعد از ۵۰۰۰۰۰ ارزیابی

HABC	P³RI²⁵	P³	RI²⁵	SEA
11.۲۴	۳۰.۹۶	۲۲.۹۷	۲۵.۵۸	۲۷.۲۹

در فرکانس تغییر بسیار زیاد (طول چرخه ۱۰۰۰ ارزیابی) روش پیشنهاد شده HABC در مقایسه با روش های حافظه دار میانگین خطای بالای داشته ولی در مقایسه با روش های بدون حافظه از روش P³RI²⁵ و RI²⁵ بهتر عمل کرده که در مجموع نشان می دهد وجود یک حافظه صریح در این نوع محیط ها برای روش پیشنهادی ضروری است و با افزایش فرکانس حافظه ضمنی نمی تواند تمامی راه حل های بهینه را نگهداری و از آنها استفاده کند جدول (۱۱ و ۱۲).

جدول (۱۱) میانگین خطای روش های با حافظه به جز روش پیشنهادی با طول چرخه ۱۰۰۰ ارزیابی بعد از ۵۰۰ ارزیابی

HABC	Mem/2search	Mem/search	P³RI²⁵m	P³m	RI²⁵m	SEAm	شمای حافظه
۲۷.۲۴	۸.۳۸	۷.۱۹	۹.۸۷	۱۰.۱۰	۹.۰۱	۱۱.۹۳	کمترین فاصله
	۹.۲۳	۸.۷۶	۱۷.۸۹	۱۰.۰۷	۹.۵۱	۱۲.۳۳	کمترین فاصله ۲
	۹.۹۹	۸.۵۸	۱۵.۳۹	۱۰.۳۶	۱۰.۱۵	۱۲.۹۹	شبیه ترین
	۱۶.۸۴	۲۴.۹۱	۳۶.۹۴	۲۰.۳۵	۲۹.۴۱	۲۲.۳۷	واریانس

جدول (۱۲) میانگین خطای روش‌های بدون حافظه با طول چرخه ۱۰۰۰ ارزیابی بعد از ۵۰۰ ارزیابی

HABC	P ^۳ RI ^{۲۵}	P ^۳	RI ^{۲۵}	SEA
۲۷.۲۴	۴۱.۱۴	۲۲.۰۹	۳۶.۵۶	۲۵.۶۷

۵. نتیجه گیری

در این مقاله یک الگوریتم جدید برای بهینه‌یابی که بر پایه دو الگوریتم کلونی زنبورهای مصنوعی و الگوریتم زنبور میباشد، پیشنهاد گردید. این الگوریتم به علت سرعت همگرایی بالا در حل مسائل محیط‌های پویا و بلادرنگ بسیار مناسب بوده و به دلیل توازن بین فرایند اکتشاف و بهره برداری که ناشی از رفتار زنبورهای کارگر، تماشاگر و پیشاپنگ می‌باشد قدرت جستجوی سراسری بالایی دارد که باعث جلوگیری از همگرایی به بهینه محلی می‌گردد. به عبارتی الگوریتم توانایی یافتن راه حل بهینه را دارد. ضمناً با توجه به اینکه از نوعی حافظه ضمنی در الگوریتم استفاده شده این مدل با ایجاد تغییرات در شرایط مسئله بسیار سریع خود را تطبیق می‌دهد. پیشنهاد می‌شود در مسائلی که دارای تعداد بهینه‌های سراسری بسیار زیادی می‌باشد و لازم است این بهینه‌ها نگهداری شوند از یک حافظه صریح استفاده شود به این دلیل که جمعیت قادر به حفظ همه بهینه‌ها به صورت ضمنی نمی‌باشد.

۶. مراجع

- [1] Blackwell, T.; Branke, J.; "Multiswarms, Exclusion, and Anti-Convergence in Dynamic Environments", IEEE Transactions on Evolutionary Computation, vol. 10, no. 4, pp. 459-472, 2006.
- [2] Blackwell, T.; Branke, J.; Li, x.; "Particle Swarms for Dynamic Optimization Problems", in Swarm Intelligence, Natural Computing Series, vol. Part II, pp. 193-217, 2008.
- [3] Branke, J.; "Memory Enhanced Evolutionary Algorithms for Changing Optimization Problems", in 1999 Congress on Evolutionary Computation, Washington D.C., USA, pp. 1875-1882, 1999.
- [4] Branke, J.; Schmeck, H.; "Designing Evolutionary Algorithms for Dynamic Optimization Problems", in Advances in Evolutionary Computing: Theory and Applications, Springer-Verlag New York, pp. 239-262, 2003.
- [5] Branke, J.; "Evolutionary optimization in dynamic environments", kluwer academic publishers, Edition 1, 2000.
- [6] Hashemi, A.B; Meybodi, M.R.; "Cellular Pso: A Pso for Dynamic Environment", in to Be Appear in the 4th International Symposium on Intelligence Computation and Applications (ISICA 2009), Huangshi, China, 2006.
- [7] Karaboga, D. "An Idea Based On Honey bee swarm numerical Optimization", Department of Computer Engineering, University of Erciyes, Tech. Rep. TR06, 2005.
- [8] Karaboga, D.; Basturk, B.; "On the Performance of Artificial Bee Colony (ABC) Algorithm", Journal of Soft computing, vol. 8, pp. 687-697, 2008.
- [9] 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.
- [10] 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.
- [11] Pham, D.T; Ghanbarzadeh, A.; "Multi-Objective Optimisation using the Bees Algorithm", Proceedings of IPROMS 2007 Conference, Cardiff, UK, pp. 454-461 , 2007.
- [12] Hu, X.; Eberhart, R.C.; "Adaptive Particle Swarm Optimization: Detection and Response to Dynamic Systems," in IEEE Congress on Evolutionary Computation, Honolulu, HI, USA, pp. 1666-1670, 2002.

- ‘ Honey Bee Colony
- ‘ Food Sources
- ‘ Forager bee
- ‘ Unemployed forager
- ‘ Employed foragers
- ‘ Scout bee
- ‘ Recruit bee
- ‘ Waggle dance
- ‘ Bee Algorithm
- ‘ Moving peaks
- ‘ Cone
- ‘ Oscillatory function
- ‘ Bell curve