

الگوریتم بهینه یابی کلونی زنبورهای مصنوعی سلولی

برای محیط های پویا

ناهید شایگان پور^۱؛ محمد رضا میبدی^۲

چکیده

در سالهای اخیر علاقه رو به رشدی در حوزه هوش دسته جمعی در مسائل بهینه یابی محیط های پویا بنا بر 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 مورچه ها و زنبورها تمرکز دارد. از این مدل ها برای جستجو و حل مسائل استفاده می شود. الگوریتم کلونی زنبورهای مصنوعی یک الگوریتم بهینه یابی بر مبنای رفتار هوشمند جمعی زنبورهاست. در این مقاله، یک الگوریتم کلونی زنبورهای مصنوعی جدید به نام الگوریتم کلونی زنبورهای مصنوعی سلولی برای محیط های پویا پیشنهاد شده است. الگوریتم پیشنهاد شده بوسیله ترکیب کلونی زنبورهای مصنوعی و اتوماتای سلولی حاصل می شود. اتوماتای سلولی از یکسری سلول مثل نقاطی در یک شبکه یا مثل مربع هایی از صفحه شطرنج که از قاعده ساده ای پیروی می کنند تشکیل شده است. زنبورها به طور تصادفی در میان سلولهای اتوماتای سلولی توزیع می شوند به طوریکه هر زنبور به یک سلول، به منظور تعامل محلی با زنبورهای سلولهای مجاور تخصیص می یابد و برای یافتن راه حل ها در محیط پویا تلاش می کنند. الگوریتم پیشنهاد شده برای توابع بهینه یابی پویا مثل تابع قله های متحرک و قله های در حال نوسان که پلی برای فهم مسائل دنیای واقعی بسیار پیچیده و سخت با مسائل بسیار ساده می باشند استفاده شده است. نتایج الگوریتم پیشنهاد شده با الگوریتم های تکاملی و همچنین الگوریتم تکاملی مبتنی بر حافظه واریانس، فاصله ای، SOS, RPSO, mQSO, Adaptive mQSO مقایسه شده است. نتایج آزمایشهای بدست آمده کارایی موثر و قدرتمندی الگوریتم پیشنهاد شده را نسبت به روشهای دیگر نشان می دهد.

کلمات کلیدی

هوش دسته جمعی، کلونی زنبورهای مصنوعی، محیط پویا، اتوماتای سلولی

Cellular Artificial Bees Colony Optimization Algorithm

for Dynamic Environments

Nahid Shayeganpour; Mohammad Reza Meybodi

ABSTRACT

Problems optimization in dynamic environment has attracted a growing interest from the swarm intelligence in recent years due to its importance in real worlds optimization problems. Swarm intelligence is a new area of research concerned with studying and modeling the behavior of social insects, such as ants and bees, and using these models for problem solving and search. Artificial bee colony (ABC) algorithm is an optimization algorithm based on the intelligent behavior of honey bee swarm. In this paper, a new ABC algorithm called cellular ABC for dynamic environments is proposed. The proposed cellular ABC is obtained by combining artificial bee colony and cellular automata (CABC). Cellular automata make up of cells like points in a lattice or like squares of checker boards and it follows a simple rule. Bees are distributed randomly among the cells of the cellular automaton that each bee is allocated to one cell by local interaction with the bees in the neighboring cells, try to find the solutions in the dynamic environment. The proposed algorithm is used for optimizing dynamic functions such as moving peaks and oscillating peaks that tries to bridge the gap between very complex, hard to understand real world problems and all too simple toy problems and the result produced by CABC, evaluation algorithm, EA based on memory such as variance, mindist, SOS and RPSO, mQSO, Adaptive mQSO have been compared. The experimental result obtained showing the effective and robust performance of the proposed CABC to other methods.

KEYWORDSS

warm Intelligence, Artificial Bee Colony, Dynamic Environment, Cellular Automata

^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

1. مقدمه

یافتن بهترین راه حل برای مسائل دنیای واقعی امروزه مورد توجه بسیاری از محققین قرار گرفته از آنجا که این مسائل ماهیت پویا دارند یعنی در هدف بهینه یابی، نمونه مسئله یا در محدودیت^۱ یک مسئله بهینه یابی تغییراتی رخ می دهد، ممکن است بهینه آن مسئله تغییر کند به عبارت دیگر محیط تغییر کرده است. اگر این حالت رخ دهد وفق دادن راه حل با راه حل قدیمی ضروری است.

بنابراین، بهتر است که الگوریتم بهینه یابی داشته باشیم که قادر باشد به طور پیوسته راه حل را با محیط در حال تغییر وفق دهد و از نتایج بدست آمده گذشته استفاده مجدد کند. از آن جا که اکثر الگوریتم ها ذاتاً به طور پیوسته در حال وفق دادن هستند، یک کاندیدای مناسب به نظر می آید ولی مشکل اصلی این الگوریتم ها، این است که سرانجام به یک بهینه همگرا^۲ می شوند و بنابراین تنوع^۳ لازم را برای اکتشاف^۴ به صورت کارا و توانایی لازم برای وفق پذیری با تغییرات محیط را از دست می دهند. لذا باید الگوریتم هایی برای محیط های در حال تغییر بکار گرفته شود که قادر به تولید تنوع بعد از یک تغییر، نگهداری تنوع در طول اجرا، حفظ راه حل های قبلی و استفاده از شیوه های چندجمعیتی^۵ باشند.

الگوریتم های هوش دسته جمعی می توانند گزینه ای مناسب برای حل مسائل بهینه یابی در محیط های پویا باشند الگوریتم های زنبور عسل در میان الگوریتم های هوش دسته جمعی، بخصوص الگوریتم خوراک جویی کلونی زنبورهای مصنوعی به علت دارا بودن ویژگی هایی همچون تقسیم کار مناسب، تعاملات گوناگون با یکدیگر و خود سازماندهی در صورتیکه قادر به حفظ تنوع در طول اجرا باشد می تواند کارایی بالایی داشته باشد.

در الگوریتم های ارائه شده قبلی مثل الگوریتم های تکاملی، الگوریتم های تکاملی مبتنی بر حافظه، چند جمعیتی [۴][۵]، الگوریتم های حرکت دسته جمعی ذرات [۲][۳][۷][۱۴] سعی شده با بهبود برخی از ویژگیهای فوق نتایج بهتری حاصل شود. در این مقاله یک الگوریتم جدید پیشنهاد شده است که ترکیبی از اتوماتای سلولی و الگوریتم خوراک جویی کلونی زنبورهای مصنوعی میباشد. به طوریکه در هر سلول از اتوماتای سلولی یک زنبور قرار دارد. اتوماتای سلولی به منظور افزایش سرعت همگرایی استفاده میشود. بدین ترتیب نوعی همکاری، بین زنبور یک سلول و سلولهای همسایه شکل می گیرد که باعث عملکرد بهتر الگوریتم می گردد. اتوماتای سلولی یک الگوریتم ذاتاً موازی می باشد و این توازی ذاتی به کلونی زنبورهای مصنوعی سلولی نیز به ارث می رسد. به عبارت دیگر زنبورهای سلولی میتوانند به صورت موازی کار کنند. با توجه به این مساله، مدل ارائه شده به صورت بالقوه، قابلیت اجرا شدن در زمانی برابر با زمان مدل استاندارد الگوریتم زنبورهای مصنوعی را داراست.

ادامه این مقاله بدین صورت سازماندهی شده است. بخش ۲ به معرفی شرح رفتار زنبورها در طبیعت، الگوریتم کلونی زنبورهای مصنوعی و اتوماتای سلولی پرداخته شده است. در بخش ۳ الگوریتم پیشنهادی معرفی و در بخش ۴ به معرفی توابع پویا و آزمایش های انجام شده بروی آنها و همچنین مقایسه با روشهای دیگر انجام می شود و بخش پایانی نتیجه گیری میباشد.

۲. شرح رفتار زنبورها در طبیعت

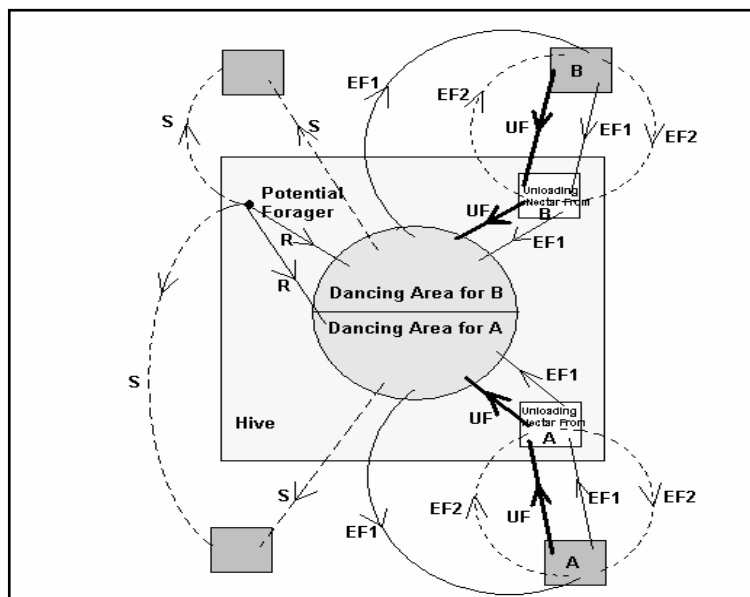
کلونی های حشرات اجتماعی به عنوان یک سیستم پویا در حال جمع آوری اطلاعات از محیط هستند و رفتار خود را بر طبق اطلاعات بدست آمده تنظیم می کنند. کلونی زنبورها در طبیعت شامل دو مولفه می باشند که عبارتند از [۸][۹]:

منابع غذایی^۶: کیفیت منابع غذایی به عوامل زیادی نظیر نزدیکی به کندو، میزان سهولت در استخراج شهد و دسترسی به آن بستگی دارد.

خوراک جو^۷: زنبوران خوراک جو به دو دسته زنبوران خوراک جو بیکار و کارگر تقسیم می شوند.

خوراک جو بیکار^۸: زنبور بیکار راجع به منابع غذایی (راه حل ها) هیچگونه دانشی ندارد و به دو نوع زنبور دیده و تازه نفس طبقه بندی می شود.

- زنبور پیشاهنگ^۹ یا دیده و زنبور دیده و عمل جستجو را بدون هیچ دانشی و به صورت تصادفی شروع می کند. درصد زنبوران پیشاهنگ از پنج تا سی درصد طبق اطلاعات درون کندو است (S در شکل (۱)).
- زنبور تازه نفس^{۱۰}: زنبور تازه نفس نیز زنبور بیکار است که به سالن رقص رفته و جستجوی خود را بوسیله استفاده از دانش رقص دم زنبورهای موجود در سالن شروع می کند (I در شکل (۱)). رقص دم شامل اطلاعات، جهت قطعه منابع غذایی که از طریق حرکت خورشید در آسمان مشخص می شود، مسافت تا کندو که توسط مدت زمان رقص و نرخ کیفیت یا شایستگی که از طریق تکرار رقص بدست می آید می باشد و به این بخش مهم از کندو که تبادل اطلاعات در آنجا انجام می شود سالن رقص گفته می شود.
- خوراک جوی کارگر^{۱۱}: هرگاه زنبور تازه نفس منبع غذایی را یافته و بهره برداری انجام دهد یک درجه ارتقاء پیدا کرده و به زنبور کارگر تبدیل می شود 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 سه امکان برای مقدار باقی مانده شهد در منبع وجود دارد که عبارتند:
- مقدار شهد به یک سطح کم کاهش پیدا کند: در اینصورت زنبور کارگر منبع غذایی را ترک کرده و به یک زنبور بیکار تبدیل می شود (UF در شکل (۱)).
- شهد به مقدار کافی در منبع غذایی وجود داشته باشد: در اینصورت زنبور کارگر می تواند بدون تبادل اطلاعات به خوراک جویی خود ادامه دهد و به منبع غذایی برگردد (EF۲ در شکل (۱)).
- شهد به مقدار کافی در منبع غذایی وجود داشته باشد: در اینصورت زنبور کارگر می تواند به سالن رقص حرکت کند و اطلاعات منبع غذایی خود را به زنبور های دیگر منتقل کند (در شکل (۱) EF۱).



شکل (۱) رفتار زنبور عسل برای عمل خوراک جویی

۲.۱. الگوریتم کلونی زنبورهای مصنوعی

الگوریتم کلونی زنبورهای مصنوعی از رفتار زنبورها در طبیعت الهام گرفته شده است. این الگوریتم شامل سه نوع زنبور دیده ور، زنبور تازه نفس و زنبور پیش آهنگ می باشد. زنبور دیده ور بوسیله راه حل های تصادفی کار خود را شروع می کند ، زنبور تازه نفس برای انتخاب یک منبع غذایی در سالن رقص منتظر مانده و رقص زنبورهای دیده ور را تماشا کرده و بر اساس کیفیت منابع غذایی، یکی از منابع غذایی را انتخاب می نماید و در انتها زنبورهای پیش آهنگ که هیچ اطلاعاتی در مورد محیط اطراف کندو ندارند به صورت تصادفی محیط اطراف کندو را به منظور یافتن یک منبع غذایی جدید جستجو می نمایند. در حقیقت این زنبورها وظیفه انجام جستجوی سراسری را بر عهده دارند [۱۰].

در ابتدا مجموعه ای از منابع غذایی به طور تصادفی انتخاب می شوند و زنبورهای دیده ور به منابع غذایی مراجعه کرده و میزان شهد آنها را محاسبه و بعد از به خاطر سپردن شایستگی خود به کندو بر می گردند و برخی دیگر به سالن رقص رفته و اطلاعات خود را با دیگران به اشتراک گذاشته و به محل منبع غذایی قبل باز می گردند و بر اساس رابطه (۱) خود را اصلاح می کنند، این اصلاح بر اساس اطلاعاتی از همسایگی همان منبع غذایی بوده و یک نوع انتخاب محلی می باشد. زنبوران تازه نفس نیز بر اساس رابطه احتمالی (۲) بدنبال زنبورهای دیده ور حرکت و به سمت منبعی حرکت می کنند که دارای کیفیت بهتری باشد، بعد از یافتن منبع، در همجواری منبع غذایی شروع به جستجو کرده و بر طبق رابطه (۱) خود را اصلاح می کنند.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad \text{رابطه (۱)}$$

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

در رابطه (۱)، ϕ_{ij} یک عدد تصادفی در بازه $[-1, 1]$ است که تولید موقعیت منابع غذایی همسایه را در اطراف x_{ij} کنترل می کند. در این رابطه، SN تعداد زنبورهای کارگر میباشد و متغیر K نیز به صورت تصادفی تولید و می بایست با \bar{a} متفاوت باشد. بر اساس رابطه (۱)، هر چه تفاوت بین x_{ij} و x_{kj} کاهش یابد، انحراف از موقعیت x_{ij} نیز کاهش خواهد یافت. در حقیقت در این رابطه سعی می کنیم یک بعد از ابعاد یکی از موقعیتهای انتخاب کرده و با توجه به میزان ϕ_{ij} به سمت آن و یا در خلاف جهت آن حرکت میکنیم.

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

در رابطه (۲) fit_i میزان شایستگی منبع غذایی متناظر با زنبور i ام میباشد.

در صورتی که یک منبع پایان پذیرد و یا کیفیت یک منبع غذایی مناسب نباشد، زنبور کارگر آن را رها میکند و تبدیل به یک پیش آهنگ میشود. این رفتار بدین صورت مدل میگردد که اگر شایستگی یک نقطه در چندین چرخه که توسط پارامتر $limit$ که طبق رابطه (۴) مشخص می شود بهبودی نداشته باشد، بدین معنی است که در یک بهینه محلی قرار داریم بنابراین آن نقطه حذف می شود و یک نقطه جدید به صورت تصادفی با استفاده از رابطه (۳) تولید می گردد.

$$x_{ij} = rand(L_j, U_j) \quad \text{رابطه (۳)}$$

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

متغیر L و U بیشترین و کمترین حد را برای تولید یک راه حل یا یک عدد تصادفی در رابطه (۳) مشخص می کند و همچنین متغیر D تعداد ابعاد مسئله در رابطه (۴) می باشد. اگر مسئله تعداد متغیر کمی داشته باشد این احتمال وجود دارد که این مکانیزم چندین بار به دلیل کوچک بودن تعداد سیکل رخ دهد و اگر مسئله تعداد متغیرهای زیادی داشته باشد این مکانیزم کمتر رخ می دهد بطوریکه شانس بهبود یک راه حل افزایش می یابد. مزیت این الگوریتم تعداد کم پارامترهای مورد نیاز می باشد که در جدول (۱) نشان داده شده است. سایز یا اندازه کلونی، تعداد کل زنبورهای کارگر و تماشاگر $SN * 2$ می باشد که تعداد افراد در هر گروه است [۱۱].

جدول (۱) پارامترهای الگوریتم کلونی زنبورهای مصنوعی

پارامتر	توصیف
SN	تعداد کل راه حل ها (منابع غذایی)، زنبورهای کارگر و زنبورهای تماشاگر
MCN	تعداد کل چرخه ها
limit	تعداد چرخه هایی که یک راه حل بهبود نیافته را ذخیره می کند و بعد از اتمام این تعداد چرخه می بایست راه حل در صورت عدم بهبود با مکانیزم پیش آهنگ تولید شود.

بتفضیل شبهه کد الگوریتم کلونی زنبورهای مصنوعی در شکل (۲) ارائه گردیده است:

۱ Begin
۲ Initialize the solution population $x_{ij} \quad i=1...SN, \quad j=1...D$
۳ Evaluate population
۴ cycle = ۱
۵ Repeat
۶ Generate new solutions v_i for the employed bees using (۱) and evaluate them
۷ Keep the best solution between current and candidate
۸ Select the visited solution for onlooker bees by their fitness
۹ Generate new solutions v_i for the onlooker bees using (۱) and evaluate them
۱۰ Keep the best solution between current and candidate
۱۱ Determine if exist an abandoned food source and replace it using an scout bee
۱۲ Save in memory the best solution so far
۱۳ cycle = cycle + ۱
۱۴ Until cycle = MCN
۱۵ End

شکل (۲) شبهه کد الگوریتم کلونی زنبورهای مصنوعی

کنفرانس داده کاوی ایران ۲۰۲۰ اتوماتای سلولی^{۱۲}

اتوماتای سلولی سیستم هایی گسسته، غیر مرکزی^{۱۳} و خود سازمانده^{۱۴} هستند، که قادرند با شروع از وضعیتی کاملاً نامرتب، ساختاری کاملاً مرتب تولید کنند (به سوی ساختاری مرتب پیش می روند) [۱۲]. در اتوماتای سلولی هدف مدل سازی سیستمی با معادلات پیچیده مشابه آنچه در معادلات دیفرانسیل وجود دارد، نیست. بلکه سعی در بیان سیستم با عناصری ساده و نمایش پیچیدگی آن با استفاده از تعامل این عناصر طبق قوانینی ساده است. بدین ترتیب یک فرآیند فیزیکی ممکن است به صورت فرآیندی محاسباتی ظاهر شود، از این رو می توان آن را مستقیماً روی یک کامپیوتر شبیه سازی کرد. در واقع اتوماتای سلولی، مدلی ریاضی برای سیستمهای پیچیده با اجزای تعریف شده ساده با تعاملات محلی است [۱].

در اتوماتای سلولی یک مجموعه منظم از سلولها وجود دارد که هر کدام میتوانند با چند مقدار مختلف که تعدادشان متناهی است، مقداردهی شوند. این سلولها به صورت همگام و در زمانهای گسسته بر طبق یک قانون محلی بهنگامرسانی میشوند. محلی بودن به این معناست که در تعیین مقدار جدید هر سلول، سلولهایی که در همسایگی وی هستند تاثیرگذار هستند و سلولهای دورتر، تاثیری ندارند. شبکه سلولها میتواند ابعاد متفاوتی داشته باشند.

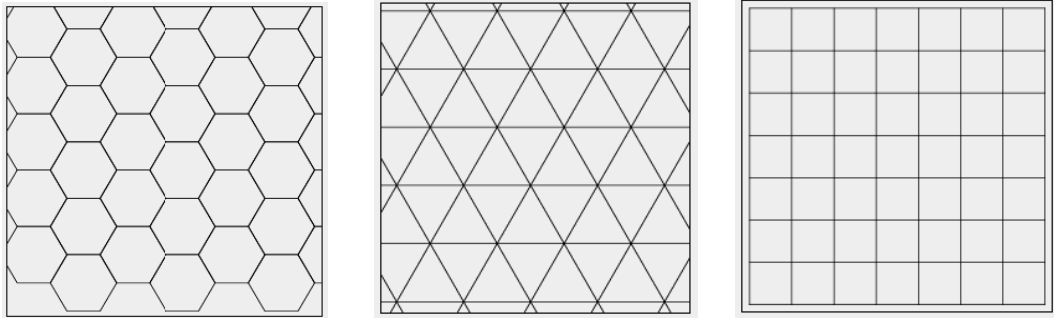
اتوماتای سلولی را می توان به صورت چهارتایی $CA = \langle L, S, N, F \rangle$ نشان داد که در آن L شبکه ای منظم و n بعدی از عناصر است. عناصر این شبکه را سلول می نامند. این عناصر می توانند در ابعاد یک، دو یا بیشتر در کنار هم چیده شده باشند. S مجموعه متناهی از حالتهاست^{۱۵} که هر سلول در هر لحظه از زمان می تواند یکی از این حالتها را بپذیرد. N مجموعه ای متناهی از همسایگی هاست که آن را با رابطه (۵) می توان نشان داد:

$$\forall c \in N, \forall r \in L \quad r + c \in L$$

رابطه (۵)

که به Γ شعاع همسایگی گفته می شود و فاصله بین سلول مورد نظر و دورترین سلول همسایگی را مشخص می کند. همچنین $S \rightarrow S^n$ ، تابع انتقال^{۱۶} شبکه، که در آن n ، بعد شبکه است. این تابع به ازای تمام سلولهای شبکه یکسان است.

اتوماتای سلولی می تواند در اشکال و انواع مختلفی پیاده سازی شود. یکی از اساسی ترین ویژگیهای اتوماتای سلولی نوع شبکه ای است که بر اساس آن محاسبات انجام می شود. ساده ترین شکل این شبکه یک خط یک بعدی است. در حالت دو بعدی، ممکن است شبکه های مربعی، مثلثی و یا شش گوشه در نظر گرفته شوند، شکل (۳) حالت های مختلف گریدهای دو بعدی ممکن را نشان می دهد.



شکل (۳) الف) شبکه مربعی

ب) شبکه مثلثی

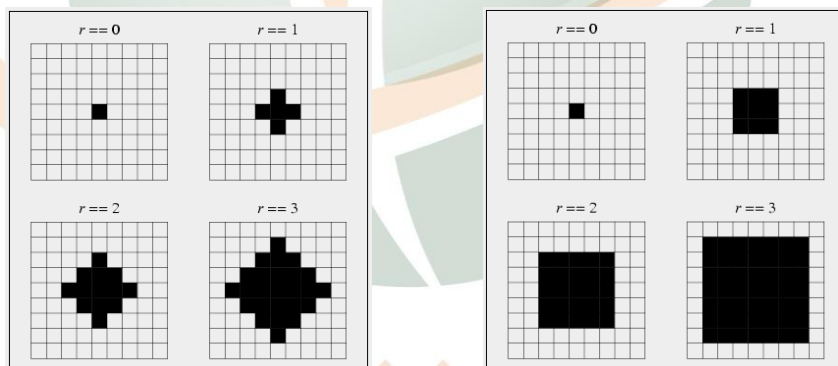
ج) شبکه شش گوشه

در اتوماتای سلولی، همسایگی سلول نیز باید مشخص باشد. منظور از همسایگی هر سلول، سلولهایی هستند که بر حالت بعدی هر سلول تاثیر می گذارند. در حالت ساده، سلولهای مجاور هر سلول به عنوان همسایه های آن در نظر گرفته شده و حالت بعدی آن را تعیین میکنند. دو همسایگی معروف که در اتوماتای سلولی دو بعدی که مبتنی بر شبکه مربعی هستند استفاده می شوند، همسایگی مور^{۱۷} و همسایگی ون نیومن^{۱۸} [۱۳] نام دارد. همسایگی مور با شعاع Γ بر طبق رابطه (۶) زیر تعریف می شود و برای چند شعاع مختلف در شکل (۴-الف) نشان داده شده است.

$$N_{(x_0, y_0)}^M = \{(x, y) : |x - x_0| \leq r, |y - y_0| \leq r\} \quad \text{رابطه (۶)}$$

همسایگی ون نیومن با شعاع Γ بر طبق رابطه (۷) تعریف شده و برای چند شعاع مختلف در شکل (۴-ب) نشان داده شده است.

$$N_{(x_0, y_0)}^V = \{(x, y) : |x - x_0| + |y - y_0| \leq r\} \quad \text{رابطه (۷)}$$



شکل (۴) الف) همسایگی مور برای شعاع $\Gamma=1$

ب) همسایگی ون نیومن برای شعاع $\Gamma=1$

شبه کد الگوریتم اتوماتای سلولی در شکل (۵) نشان داده شده است.

```

procedure Cellular Automata
begin
  repeat
    for every cell i in cellular automata do in parallel
      update state of cell i according to rule R
    endfor
  until termination condition is met
end
  
```

شکل (۵) الگوریتم کلونی زنبورهای مصنوعی

ویژگیهای اتوماتای سلولی را به اختصار میتوان به این صورت بیان نمود: فضا و زمان به صورت گسسته پیش می روند، اتوماتا همگن است، عمل به روزرسانی به صورت همگام انجام می شود و قوانین بر اساس همسایه های هر سلول تعریف می شوند. از مشکلات مهم اتوماتای سلولی تعیین فرم قطعی قوانین است. زیرا در اغلب سیستمها نویز و عدم قطعیت وجود دارد که سیستم را تحت تاثیر قرار می دهد. لذا تعیین فرم قطعی قوانین در این سیستمها کاری مشکل و در برخی موارد غیر ممکن است.

۳. الگوریتم پیشنهادی CABG

الگوریتم پیشنهاد شده بوسیله ترکیب کلونی زنبورهای مصنوعی و اتوماتای سلولی حاصل می شود که در شکل (۶) شبه کد آن نمایش داده شده است. 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، زنبورها به طور تصادفی در میان سلولهای اتوماتای سلولی توزیع می شوند به طوری که هر زنبور به یک سلول، به منظور تعامل محلی با زنبورهای سلولهای مجاور تخصیص می یابد و برای یافتن راه حل ها در محیط تلاش می کنند به عبارت دیگر هر زنبور در یک سلول قرار می گیرد و زنبورها با استفاده از موقعیت همسایگان، موقعیت بعدی خود را محاسبه می نمایند. در الگوریتم زنبورهای مصنوعی هر کدام از زنبورهای کارگر و یا تماشاگر ممکن است تغییراتی روی موقعیت منبع غذایی (راه حل) در حافظه خود ایجاد کنند و شایستگی آن را محاسبه کرده، در صورتیکه میزان شایستگی آن از راه حل قدیمی بیشتر باشد، راه حل جدید را انتخاب کنند و راه حل قدیمی فراموش شود در غیر اینصورت همان راه حل قدیمی باقی خواهد ماند. این تغییرات با انتخاب تصادفی یک یا بیش از یک زنبور از میان تمامی زنبورها و حرکت به سمت آن انجام می گیرد. در این مدل به جای انتخاب به طور تصادفی، انتخاب از میان همسایگان صورت می گیرد.

در الگوریتم زنبورهای مصنوعی، هر زنبور برای بهنگام کردن موقعیت خود از رابطه (۱) استفاده مینمود. متغیر k در این رابطه، به صورت تصادفی از بین تمامی زنبورها انتخاب میشود. در حالی که در این مدل متغیر k به صورت تصادفی از میان همسایگان زنبور مورد نظر انتخاب می شود. بنابراین رابطه به صورت رابطه (۸) تغییر خواهد نمود.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad \text{رابطه (۸)}$$

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

که متغیر ngb تعداد همسایگان آن زنبور با توجه به شبکه مورد نظر و شعاع همسایگی میباشد. بنابراین در صورتی که موقعیت جدید از موقعیت قبلی بهتر باشد. (همانند الگوریتم کلونی زنبورهای مصنوعی) هر زنبور راستای موقعیت یکی از همسایگان خود حرکت خواهد نمود.

```

Initialize the CAC
Do
  For Each cell do in parallel
    For each bee in cell
      Calculate fitness value of the bee
    End For
    For each bee
      Produce a modification on the Position of the bee (Moore or von Neumann
      Neighborhood)
    End For
  End For

  For each bee
    Calculate probability value of the bee Pb
  End For
  Use Sharing function model Roulette Wheel Selection
  For Each cell do in parallel
    For each bee
      Produce a modification on the Position of the bee (Moore or von Neumann
      Neighborhood)
    End For
  End For

/* moves Scouts */
If a position cannot be improved through Limit of cycle
  Generate a randomly uniform Position
End If
While maximum iterations OR minimum error criteria is not attained
  
```

شکل (۶) شبه کد الگوریتم پیشنهادی

۴. آزمایش‌ها

در آزمایش‌های انجام شده هدف بررسی و مقایسه الگوریتم‌های تکاملی مبتنی بر حافظه، SOS, RPSO, mQSO, Adaptive mQSO و الگوریتم CABC در محیط‌های پویا به کمک توابع قله‌های متحرک و توابع قله‌های در حال نوسان می‌باشد از آنجا که در محیط‌های پویا بهترین راه حل بدست آمده معیار مفیدی برای ارزیابی و گزارش نمی‌باشد، از معیارهای *Current _ Error*، *Offline _ Error* جهت مقایسه و نتیجه‌گیری استفاده شده است [۶].

۴.۱. تابع چند قله‌ای متحرک^{۱۹}

جهت پدید آوردن یک محیط پویا از یک تابع محک مناسب، چند قله‌ای متحرک [۴][۵] استفاده شده است که شامل یک فضای چند بعدی، چندین قله با ارتفاع و پهناهای متفاوت است، که ارتفاع، پهنا و موقعیت هر قله در طول زمان در حال تغییر است. تابع چند قله‌ای 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 می‌باشد که به صورت رابطه (۹) تعریف می‌شود:

$$F(\vec{x}, t) = \text{Max}(B(\vec{x}), \text{Max}_{i=1..m} P(\vec{X}, H_i(t), W_i(t), \vec{p}(t))) \quad \text{رابطه (۹)}$$

در این تابع $B(\vec{x})$ یک شمای پایه ثابت در زمان و p تابع چند قله‌ای است که در هر Δe ارزیابی، ارتفاع و پهناهای قله‌ها 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 (۱۰) تغییر می‌کند. در این تابع می‌توان پیچیدگی را با افزایش و کاهش Δe در زمان تغییر داد.

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

رابطه (۱۰)

بردار v را می‌توان وابسته به تغییر قبلی آن ایجاد کرد که در این صورت تغییر موقعیت قله‌ها همسو با تغییرات قبل آن می‌شود و یا به صورت تصادفی آنرا ایجاد نمود که موجب می‌شود موقعیت قله‌ها به صورت تصادفی تغییر کند و هیچگونه وابستگی به تغییر قبلی نداشته باشد. مقادیر پارامترهای برای انجام آزمایش‌ها طبق جدول (۲) در نظر گرفته شده است.

جدول (۲) تنظیم پارامترهای تابع قله‌های متحرک

پارامتر	مقدار پیش فرض
λ فاکتور وابستگی	۰
P تعداد قله‌ها	۱۰
f فرکانس تغییرات	هر ۵۰۰۰ ارزیابی
سختی ارتفاع	۷.۰
سختی پهنا	۱.۰
شکل قله‌ها	مخروطی ^{۲۰}
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^{۲۱}

تابع پیچیده دیگر که در آزمایش ها مورد استفاده قرار می گیرد تابع قله های در حال نوسان [۴][۵] می باشد که در آن L منظره (معمولا $L=2$) 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 (۱۱) نوسان می کند.

$$\begin{cases} f_i(t) = \omega(t) f_i(0) \\ \omega(t) f_i(0) = 0.5 \cos\left(\frac{2t\pi}{step} + 2\pi \frac{i-1}{l}\right) + 0.5 \\ i = 1 \dots l \end{cases} \quad \text{رابطه (۱۱)}$$

با $step$ تعداد گام های میانی در یک سیکل تعریف می شود. در نتیجه مسئله بین L منظره نوسان می کند. در آزمایش ها 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 [۴][۵]."

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 ^{۲۲}
s طول تغییر	۱۰
D تعداد ابعاد	۵
H مقدار مینیمم و ماکزیمم ارتفاع	[۳۰۰, ۷۰۰]
W مقدار مینیمم و ماکزیمم پهنا	[۰.۰۰۱ و ۰.۰۸]
A محدوده فضای جستجو	[۰, ۱۰۰]

۴.۳. پارامترهای تنظیم شده الگوریتم ها در آزمایش ها

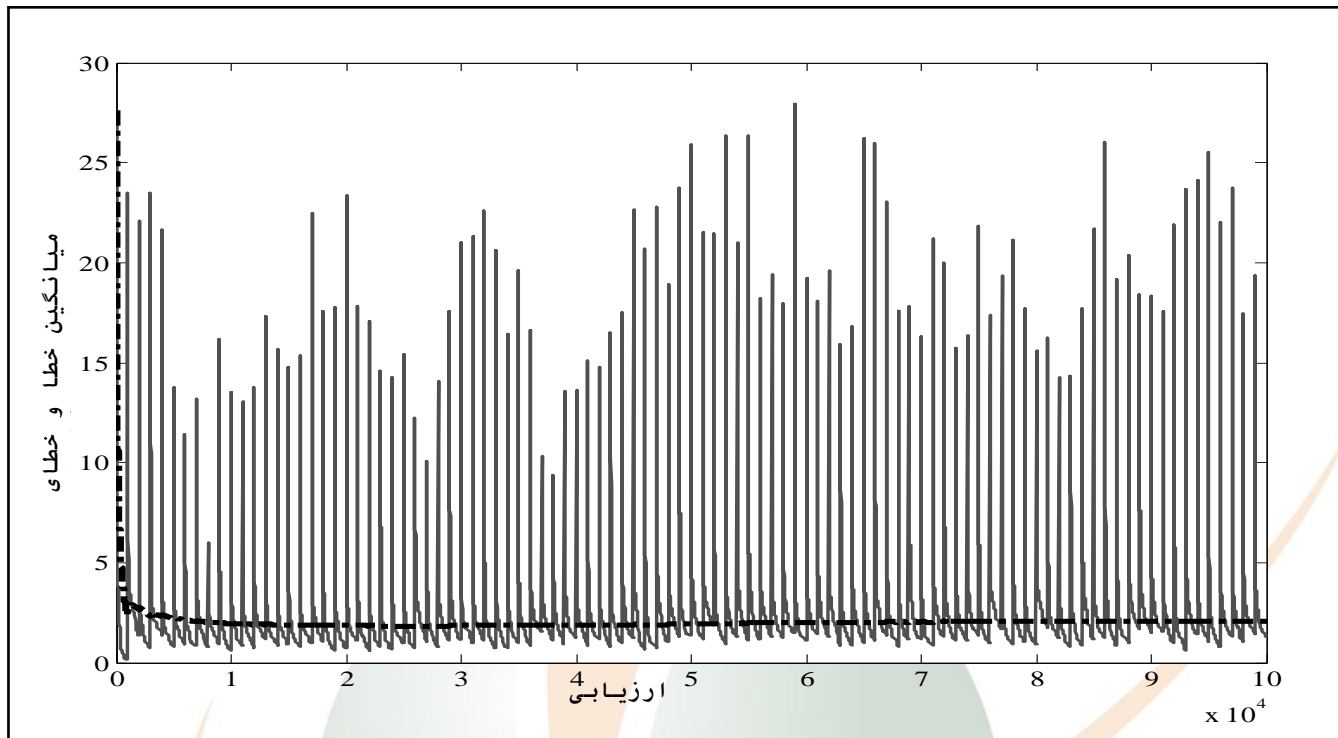
در این الگوریتم تعداد ژن ها (۱۰۰ ژن) در هر کروموزوم (تعداد ابعاد قله ها) ۵ در نظر گرفته شده و به منظور تولید فرزندان و انتخاب والدین از روش باز ترکیبی دو نقطه ای با نرخ ۰.۶ و جهش گوسین با نرخ ۰.۲ و روش انتخاب چرخ گردان استفاده شده است و در صورت استفاده از حافظه، تعداد ۱۰ فرد را در حافظه ذخیره می کند.

الگوریتم زنبورهای مصنوعی سلولی نیز دارای پارامترهایی می باشد که قبل از شروع الگوریتم می بایست تنظیم شود که عبارتند از تعداد زنبورهای کارگر ۱۰۰ راه حل، نوع همسایگی ون نیومن با شعاع همسایگی ۵ و تعداد ۱۰ زنبورهای پیش آهنگ در هر نسل در نظر گرفته شده است. 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، ماکزیمم سرعت برای یک ذره و شعاع همسایگی اتوماتای سلولی و شعاع جستجوی محل تصادفی در همه آزمایشها ۰.۵ در نظر گرفته شده است. یک اتوماتای سلولی پنج بعدی با 10^5 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 $m(n+k^q)$ ، تعداد زیر گروه و n تعداد ذره استاندارد و k تعداد ذره های کوانتومی می باشد.

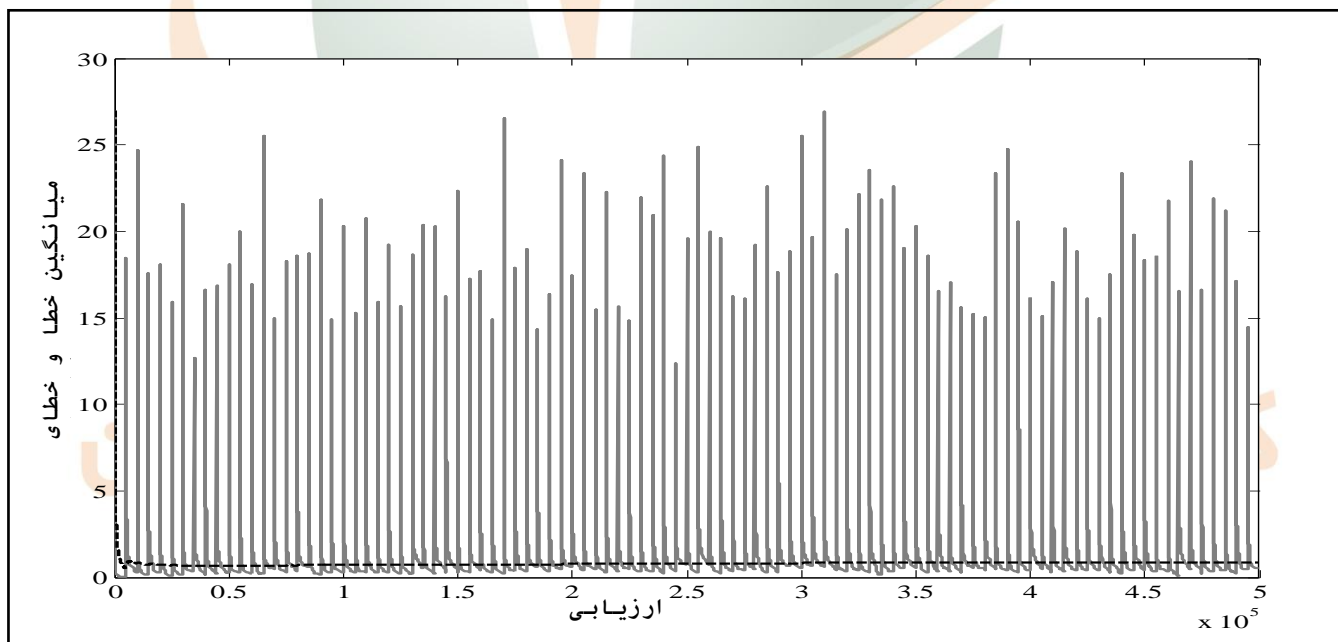
۴.۴. آزمایشهای تابع قله های متحرک

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 بکمک معیار Offline error با الگوریتم های PSO [۱۴]، RPSO [۲]، mQSO [۳]، Adaptive mQSO [۷]، Cellular PSO [۷]، Adaptive cellular [۷] 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 $f=10000$ و منظور از فرکانس تغییرات بالا یعنی تعداد ارزیابی ها بین تغییرات کم می باشد مثل $f=500$) از آزمایش های انجام شده نتایج زیر حاصل شده است:

- در شکل های (۷ و ۸) میانگین خطا و خطای لحظه ای روش پیشنهادی CABC در دو فرکانس تغییر مختلف ۱۰۰۰ و ۵۰۰۰ ارزیابی با ۱۰۰ 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 خطا در فرکانس تغییر پایین کمتر از فرکانس تغییر بالا می باشد.

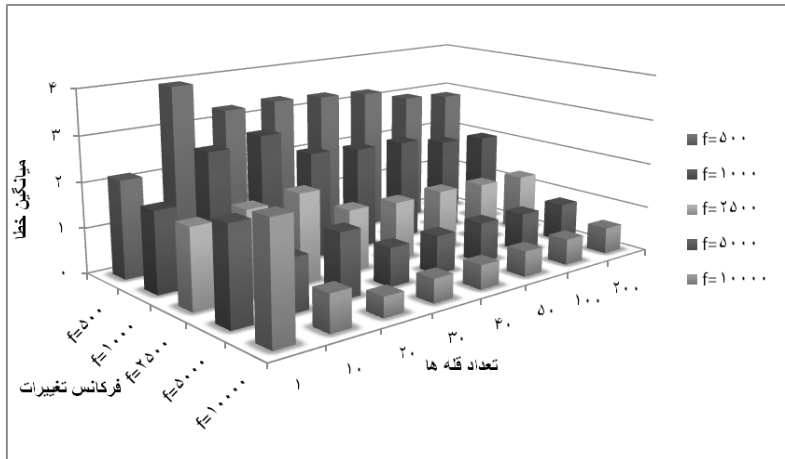


شکل (۷) میانگین خطا و خطای لحظه ای در فرکانس تغییر ۱۰۰۰ ارزیابی با ۱۰۰ قله



شکل (۸) میانگین خطا و خطای لحظه ای در فرکانس تغییر ۵۰۰۰ ارزیابی با ۱۰۰ قله

- به منظور بررسی دقیق تر رفتار الگوریتم میانگین خطا در فرکانس های تغییر مختلف با قله های متفاوت محاسبه و در شکل (۹) نمایش داده شده است. با توجه به آزمایش ها مشخص می گردد در توابع چند قله ای با کاهش فرکانس تغییر بدلیل اینکه زمان کافی برای یافتن قله ها و همگرایی وجود دارد مقدار میانگین خطا کاهش می یابد اما در حالتی که فقط یک قله وجود دارد نتیجه کاملاً معکوس می باشد به این دلیل که پس از گذشت چند نسل شبکه سلولی تبدیل به یک شبکه با مقادیر نسبتاً یکسان می شود و دیگر سلولها قادر به اصلاح خود به کمک سلولهای مجاور نیستند.



شکل (۹) طیف میانگین خطا با توجه به فرکانس تغییرات و تعداد قله ها در الگوریتم پیشنهادی CABC

- در محیط هایی که دارای چند قله می باشند روشهای CABC، adaptive cellular و cellular PSO نسبت به بقیه روشها خوب عمل می کنند ولی روش CABC عملکرد بهتری نسبت به بقیه روشها دارد به این دلیل که فرایند اصلاح توسط زنبورهای کارگر و تماشاگر بوسیله همسایگی ون نیومن با شعاع وسیعی انجام می شود و چون سلولها در این شعاع پراکندگی مناسبی دارند این امر موجب می گردد یک جستجوی مناسب برای یافتن بهینه سراسری صورت پذیرد جدول های (۴ تا ۸).
- در محیط های تک قله ای همه روش ها نسبت به روش RPSO ضعیف تر عمل نموده اند. روش پیشنهادی CABC بجز فرکانس های تغییر ۵۰۰ و ۱۰۰۰ عملکرد مناسب نداشته و هر چه فرکانس تغییر کاهش پیدا می کند نتایج بدست آمده بدتر می شود این موضوع در نمودار شکل (۸) به طور آشکار دیده می شود جدول های (۴ تا ۸).

جدول (۴) Offline error برای $F=500$

CABC	Adaptive Cellular PSO[۷]	Adaptive mQSO [۳]	Cellular PSO[۷]	mQSO $10 \cdot (5+5^9)$ [۲]	mQSO $10 \cdot (10+0^9)$ [۲]	RPSO[۱۴]	تعداد قله ها
۲.۱۵	۱۲.۱۹	۱۴.۰۷	۱۳.۴۶	۳۳.۶۷	۴۴.۴۸	۵.۲۰	۱
۳.۹۷	۹.۱۸	۹.۴۲	۹.۳۵	۹.۶۲	۱۵.۱۵	۱۸.۱۱	۱۰
۳.۳۰	۸.۷۸	۹.۵۹	۸.۸۴	۹.۰۷	۱۳.۰۴	۱۷.۸۰	۲۰
۳.۳۵	۸.۶۴	۹.۴۴	۸.۸۱	۸.۸۰	۱۲.۴۸	۱۷.۴۲	۳۰
۳.۲۹	۸.۶۸	۹.۴۴	۸.۹۴	۸.۵۵	۱۲.۲۲	۱۶.۷۹	۴۰
۳.۲۲	۸.۴۹	۹.۱۳	۸.۶۲	۸.۷۲	۱۲.۱۸	۱۶.۴۲	۵۰
۲.۹۶	۸.۲۲	۸.۸۱	۸.۵۴	۸.۵۴	۱۱.۵۴	۱۵.۰۰	۱۰۰
۲.۸۶	۷.۸۹	۸.۲۱	۸.۲۸	۸.۱۹	۱۱.۲۵	۱۴.۰۰	۲۰۰

جدول (۵) Offline error برای $F=1000$

CABC	Adaptive Cellular PSO[۷]	Adaptive mQSO [۳]	Cellular PSO[۷]	mQSO $10 \cdot (5+5^9)$ [۲]	mQSO $10 \cdot (10+0^9)$ [۲]	RPSO[۱۴]	تعداد قله ها
۱.۷۸	۵.۸۳	۶.۶۰	۶.۷۷	۱۸.۶۰	۲۱.۶۸	۲.۴۰	۱
۲.۷۹	۵.۲۹	۵.۶۴	۵.۱۹	۵.۷۱	۷.۶۶	۱۵.۷۷	۱۰
۲.۹۵	۵.۴۹	۵.۹۵	۵.۲۳	۵.۸۵	۷.۲۵	۱۵.۶۱	۲۰
۲.۳۷	۵.۴۲	۵.۹۷	۵.۳۳	۵.۸۱	۷.۱۹	۱۴.۸۸	۳۰
۲.۲۸	۵.۳۲	۶.۱۲	۵.۶۱	۵.۷۰	۷.۱۹	۱۴.۵۳	۴۰
۲.۲۵	۵.۲۵	۵.۹۸	۵.۵۵	۵.۸۷	۷.۱۴	۱۴.۱۶	۵۰
۲.۰۸	۵.۰۹	۵.۷۸	۵.۵۷	۵.۸۳	۶.۹۴	۱۲.۵۳	۱۰۰
۲.۰۰	۴.۵۹	۵.۵۴	۵.۵۰	۵.۵۴	۶.۹۸	۱۱.۵۸	۲۰۰

جدول (۶) Offline error برای $F=2500$

CABC	Adaptive Cellular PSO[γ]	Adaptive mQSO [3]	Cellular PSO[γ]	mQSO $10(5+5^q)$ [2]	mQSO $10(10+0.9^q)$ [2]	RPSO[14]	تعداد قله ها
1.74	2.00	2.48	4.15	7.64	9.88	1.00	1
1.85	3.03	2.91	2.82	3.12	4.38	13.72	10
1.96	3.17	3.40	3.41	3.58	4.34	13.89	20
1.37	3.22	3.47	3.62	3.63	4.36	13.01	30
1.31	3.32	3.56	3.84	3	4.37	12.49	40
1.34	3.30	3.56	3.86	3.63	4.36	11.96	50
1.29	3.35	3.53	4.10	3.58	4.21	10.73	100
1.26	3.29	3.37	3.97	3.30	4.04	9.63	200

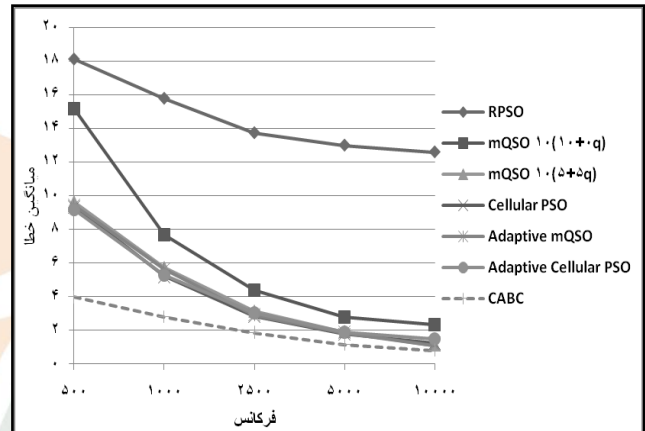
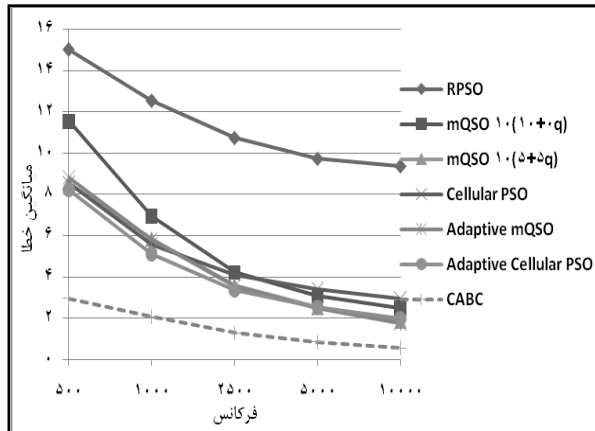
جدول (7) Offline error برای $F=5000$

CABC	Adaptive Cellular PSO[γ]	Adaptive mQSO [3]	Cellular PSO[γ]	mQSO $10(5+5^q)$ [2]	mQSO $10(10+0.9^q)$ [2]	RPSO[14]	تعداد قله ها
2.10	0.87	1.09	2.54	3.82	5.17	0.56	1
1.15	1.91	1.85	1.76	1.91	2.81	12.98	10
1.43	2.26	2.18	2.59	2.56	3.22	12.79	20
0.85	2.25	2.36	2.95	2.68	3.29	12.35	30
0.85	2.41	2.42	3.11	2.65	3.24	11.37	40
0.89	2.43	2.53	3.22	2.63	3.27	11.34	50
0.86	2.53	2.50	3.39	2.52	3.08	9.73	100
0.85	2.46	2.36	3.36	2.36	2.89	8.90	200

جدول (8) Offline error برای $F=10000$

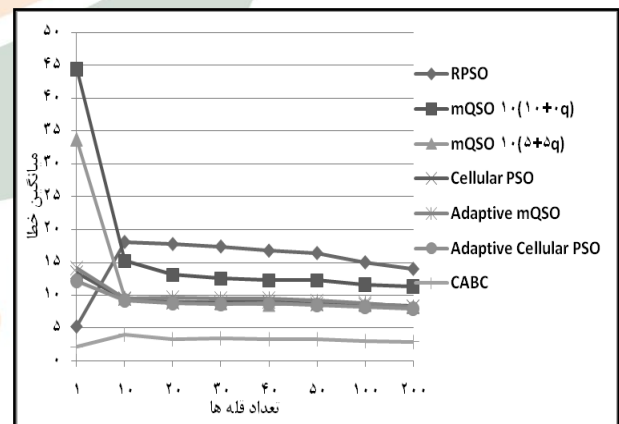
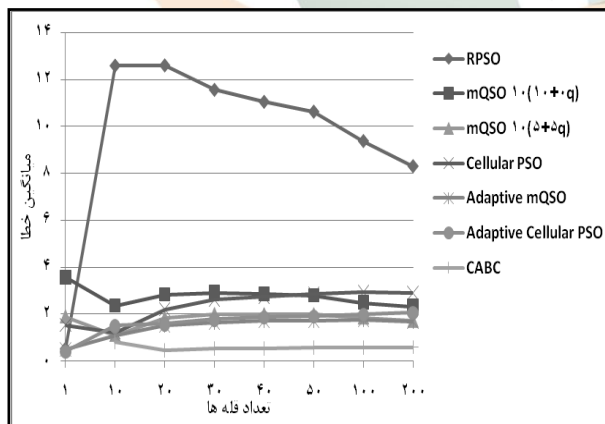
CABC	Adaptive Cellular PSO[γ]	Adaptive mQSO [3]	Cellular PSO[γ]	mQSO $10(5+5^q)$ [2]	mQSO $10(10+0.9^q)$ [2]	RPSO[14]	تعداد قله ها
2.50	0.40	0.49	1.53	1.90	3.56	0.53	1
0.80	1.49	1.10	1.22	1.10	2.34	12.58	10
0.44	1.57	1.50	2.20	1.84	2.82	12.59	20
0.52	1.77	1.64	2.60	2.00	2.91	11.55	30
0.53	1.86	1.70	2.73	1.99	2.85	11.04	40
0.56	1.91	1.71	2.84	1.99	2.78	10.62	50
0.57	1.98	1.75	2.93	1.85	2.48	9.37	100
0.58	2.05	1.66	2.88	1.71	2.32	8.30	200

- به منظور مقایسه رفتار الگوریتم‌ها تغییرات در فرکانس‌های مختلف با تعداد قله‌های ۱۰ و ۱۰۰ صورت گرفته است و نتایج در شکل‌های (۱۰ و ۱۱) نمایش داده شده است با توجه به نمودارها مشخص گردیده که با کاهش فرکانس میانگین خطای همه روشها کاهش می‌یابد و با افزایش تعداد قله‌ها میانگین خطای همه روشها کاهش می‌یابد.
- روش پیشنهادی **CABC** در تعداد قله‌های کم و فرکانس تغییر پایین تقریباً مشابه همه روشهای به جز **RPSO** عمل می‌کند ولی با افزایش تعداد قله‌ها نتایج بهتری حاصل می‌شود که اختلاف میانگین خطا بین این روش در فرکانس ۱۰۰۰۰ با تعداد ۱۰۰ قله در شکل (۱۱) نسبت به شکل (۱۰) ناشی از همین موضوع می‌باشد.



شکل (۱۰) نمودار میانگین خطا در فرکانس‌های تغییر مختلف با تعداد ۱۰ قله
شکل (۱۱) نمودار میانگین خطا در فرکانس‌های تغییر مختلف با تعداد ۱۰۰ قله

- همانطور که در شکل (۱۲ و ۱۳) مشخص گردیده میانگین خطاهای الگوریتم‌ها با تعداد قله‌های متفاوت در دو فرکانس تغییر ۵۰۰ و ۵۰۰۰ ارزیابی نمایش داده شده است این نمودارها مشخص می‌کنند روش **CABC** به جز در حالت تک قله‌ای در چند قله‌ای‌ها نسبت به بقیه روشها بهتر عمل می‌کند و میانگین خطا تقریباً از ۲۰ قله به بالا در فرکانس‌های متفاوت به یک حالت پایدار می‌رسد.



شکل (۱۲) نمودار میانگین خطا با تعداد قله‌های مختلف و فرکانس تغییر ۵۰۰
شکل (۱۳) نمودار میانگین خطا با تعداد قله‌های مختلف و فرکانس تغییر ۵۰۰۰

۴.۵. 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 CABC در مقایسه با روشهای حافظه دار که در این نوع مسائل که منظره مجدداً تکرار می شود کاربرد دارند، میانگین خطای بالایی داشته و فقط زمانیکه در روشهای حافظه دار از شمای حافظه واریانسی استفاده شده نتیجه راضی کننده بوده است ولی در مقایسه با روش های بدون حافظه نتایج بسیار بهتر بوده و میانگین خطای پایینی حاصل گردیده است جدول (۹ و ۱۰).

جدول (۹) میانگین خطا روشهای با حافظه به جز در روش پیشنهادی با طول چرخه ۱۰۰۰۰ ارزیابی بعد از گذشت ۵۰۰۰۰۰ ارزیابی

CABC	Mem/2search	Mem/search	P3RI25m	P3m	RI25m	SEAm	شمای حافظه
۱۶.۰۳	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	۵.۵۷	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 خطا روشهای بدون حافظه با طول چرخه ۱۰۰۰۰ ارزیابی بعد از ۵۰۰۰۰۰۰ ارزیابی

CABC	P3RI25	P3	RI25	SEA
۱۶.۰۳	۳۰.۹۶	۲۲.۹۷	۲۵.۵۸	۲۷.۲۹

- در فرکانس تغییر بسیار زیاد (طول چرخه ۱۰۰۰۰ ارزیابی) روش پیشنهاد شده CABC در مقایسه با روشهای حافظه دار میانگین خطای بالایی داشته ولی در مقایسه با روش های بدون حافظه از روش P3RI25 و RI25 بهتر عمل کرده است جدول (۱۱).
- در آزمایش قله های در حال نوسان رفتار الگوریتم در برابر مسائل تکرار پذیر بررسی می شود و چون در حل اینگونه مسائل وجود حافظه صریح بسیار موثر است نتایج بدست آمده جدول (۱۱) این را موضوع آشکار می کند و عملکرد ضعیف CABC به علت نداشتن حافظه آشکارتر می گردد.
- در روشهای بدون حافظه روش پیشنهادی CABC کارایی مناسبی داشته، علت ضعف و کارایی کم این روش در مقابل SEA و P3 ماهیت این روشها می باشد چون این روشها با داشتن مناطق جداگانه برای جستجو و داشتن چندین جمعیت به نوعی از یک حافظه صریح استفاده نموده اند جدول (۱۲).

جدول (۱۱) میانگین خطا روشهای با حافظه به جز روش پیشنهادی برای طول چرخه ۱۰۰۰ ارزیابی بعد از ۵۰۰ ارزیابی

CABC	Mem/2search	Mem/search	P3RI25m	P3m	RI25m	SEAm	شمای حافظه
۳۰.۶۵	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	۲۳.۳۷	واریانس

جدول (۱۲) میانگین خطا روشهای بدون حافظه برای طول چرخه ۱۰۰۰ ارزیابی بعد از ۵۰۰ ارزیابی

CABC	P3RI25	P3	RI25	SEA
۳۰.۶۵	۴۱.۱۴	۲۲.۰۹	۳۶.۵۶	۲۵.۶۷

5. نتیجه گیری

در این مقاله یک الگوریتم جدید برای بهینه یابی که بر پایه ترکیب دو الگوریتم کلونی زنبورهای مصنوعی و الگوریتم اتوماتای سلولی میباشد، پیشنهاد گردید. الگوریتم پیشنهادی CABO دارای قابلیت موازی سازی آسان، افزایش سرعت همگرایی و کاهش قرار گرفتن در بهینه محلی می باشد. در این الگوریتم از حافظه صریح استفاده نشده و در حل مسائل پویا که تکرارپذیر نمی باشند در مقایسه با الگوریتم های Adaptive mQSO، Cellular PSO، Adaptive cellular و الگوریتم های تکاملی مبتنی بر حافظه دارای میانگین خطای کمتری می باشد. این الگوریتم در مسائل پویای تکرار پذیر نسبت به روشهای بدون حافظه عملکرد مناسبی دارد لذا پیشنهاد می شود این الگوریتم با یک حافظه صریح در حل مسائل تکرار پذیر ترکیب شود.

6. مراجع

- [1] Anvarinejad, T.; Meybodi, M. R.; "Fuzzy Cellular Automata", Proceedings of The 5th Iranian Conference on Fuzzy Systems, Imam Hussein University, Tehran, pp.57-65, 2004.
- [2] 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.
- [3] 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.
- [4] Branke, J.; "Evolutionary optimization in dynamic environments", kluwer academic publishers, Edition 1, 2000.
- [5] Branke, J.; "Memory Enhanced Evolutionary Algorithms for Changing Optimization Problems", in 1999 Congress on Evolutionary Computation, Washington D.C., USA, pp. 1875-1882, 1999.
- [6] 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.
- [7] 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.
- [8] Karaboga, D.; "An Idea Based On Honey bee swarm numerical Optimization", Department of Computer Engineering, University of Erciyes, Tech. Rep. TR06, 2005.
- [9] Karaboga, D.; Basturk, B.; "On the Performance of Artificial Bee Colony (ABC) Algorithm", Journal of Soft computing, vol. 8, pp. 687-697, 2008.
- [10] Karaboga, D.; Basturk, B.; "A powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm", Journal of Global Optimization, vol. 39, pp. 459-471, 2007.
- [11] 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.
- [12] Martin, O.; Odluzko, A.; Wolfram, S.; "Algedric properties of Cellular Automata", Math. Phys., vol. 93, pp. 219-258, 1984.
- [13] Packard, N. H.; Wolfram, S.; "Two-Dimensional Cellular Automata", Journal of Statistical Physics, vol. 38, pp. 901-946, 1985.
- [14] 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.

¹ Restriction

² Converge

³ Diversity

⁴ Exploring

⁵ Multi population

⁶ Food Sources

⁷ Forager

⁸ Unemployed foragers

⁹ Scout Bee

¹⁰ Recruit

¹¹ Employed foragers

¹² Bee Algorithm

¹³ Decentralized

¹⁴ Self-organizing

¹⁵ State

¹⁶ Transition function

¹⁷ Moore Neighborhood

¹⁸ von Neumann Neighborhood

¹⁹ Moving peaks

²⁰ Cone

²¹ Oscillatory function

²² Bell curve