

یافتن آلودگی های نفتی و میعانات گازی در دریاها با استفاده از شبکه های حسگر بیسیم و الگوریتم حرکت دسته جمعی ذرات (Particle Swarm Optimization)

ازمایشگاه محاسبات نرم

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صوفی رازی

زنجان ایران

روش های موجود برای یافتن آلودگی های نفتی و میعانات گازی در دریاها ؟

چکیده: در این مقاله یک روش تکاملی بر پایه شبکه های حسگر بیسیم برای یافتن لکه های نفتی و میعانات گازی موجود در دریاها و اقیانوس ها پیشنهاد می گردد. روش ارائه شده با بکارگیری شبکه های حسگر بیسیم و الگوریتم حرکت دسته جمعی ذرات می تواند در محیط های با نویز بالا و نیز در شرایطی که برخی از ند های شبکه سنسور اطلاعات نادرستی را گزارش می نمایند ، به خوبی عمل نماید و لکه های نفتی و میعانات گازی موجود در آب های آزاد را پیدا کند. در این روش هر سنسور ، میزان غلظت آلودگی اطراف خود را تعیین کرده و آن را با مقادیر سنس شده قبلی مقایسه میکند. اگر مقدار غلظت افزایش یافته باشد این نقطه به عنوان مکانی نزدیکتر به هدف در حافظه سنسور ذخیره میشود . در غیر این صورت سنسور به حرکت خود بر اساس الگوریتم پیشنهادی ادامه می دهد. همچنین در هر گام الگوریتم مکان بالاترین غلظت سنس شده توسط شبکه حسگر بین تمام حسگر های شبکه به اشتراک گذاشته میشود . حرکت بعدی هر سنسور در هر گام الگوریتم با توجه به بهترین مکانی که تا به حال سنس کرده و بهترین مکان کل شبکه حسگر توسط الگوریتم تعین میشود. در واقع الگوریتم با به کارگیری هوش جمعی روشی سریع تر به منظور کشف آلودگی ارائه میدهد. با توجه به ماهیت پویای آب دریاها به علت وجود امواج ، جریان های در یایی، باد و دیگر عوامل محیطی از نوع خاصی از الگوریتم حرکت دسته جمعی ذرات که مخصوص محیط های دینامیک است ، استفاده می کنیم و نتایج را با الگوریتم غیر دینامیک مقایسه میکنیم .

کلمات کلیدی: الگوریتم حرکت دسته جمعی ذرات ، شبکه های حسگر بیسیم ، لکه های نفتی و میعانات گازی

۱. مقدمه

استفاده از روبات های ماشینی به منظور دستیابی به هدف مد نظر در این مسئله ، روشی بهینه و سریع می باشد . یک روبات با قابلیت های بالا ممکن است بتواند به تنهایی با حرکت در محیط و تعیین میزان آلودگی در هر نقطه ، به سرعت محل آلودگی را بیابد ، ولی به چندین دلیل استفاده از چندین روبات با قابلیت های پایین تر بدین منظور راه حل مناسب تری خواهد بود . هزینه استفاده از یک روبات با قابلیت های بالا میتواند بسیار بیشتر از هزینه استفاده از چندین روبات با قابلیت های پائین تر باشد ، به علاوه در صورتی که به هر دلیل روبات خراب شود ، هزینه پرداختی در حالت استفاده از یک روبات با قابلیت های بالا بسیار بیشتر خواهد بود . دلیل دیگر وجود نویزهای احتمالی در محیط می باشد که می تواند عملکرد روبات را دچار اختلال نماید . استفاده از چندین روبات می تواند قدرت بیشتری را برای غلبه بر نویزهای احتمالی محیط فراهم آورد . در صورت استفاده از چندین روبات ، چگونگی برقراری ارتباط میان روبات ها و انتقال اطلاعات به دست آمده توسط هریک به دیگران باید مد نظر قرار گیرد . یکی از گزینه های مناسب برای این منظور

محیط زیست دریایی به سبب اینکه از عناصر بسیار گوناگونی تشکیل شده است در برابر مواد خارجی که به عمد یا غیر عمد به آن وارد میشود بسیار آسیب پذیر است. لذا هر نوع آلودگی بویژه آلودگی نفتی میتواند مشکلات بسیار جدی و خطر ناکی را بوجود آورد. ورود میلیونها لیتر نفت خام به دریاها بر اثر حوادث مختلف دریایی، تخلیه صدها هزار لیتر زایدات نفتی ناشی از تردد وسایل نقلیه زمینی، دریایی و هوایی به محیط زیست، آلودگیهای ناشی از فعالیت صدها پالایشگاه و مراکز صنعتی وابسته ، حجم عظیم پساب های صنایع و عوامل متعدد دیگر آلودگی های زیست محیطی که به گونه ای به مواد نفتی باز می گردد، همه و همه علامت هشدار است که ضرورت پرداختن به محیط زیست را بیش از پیش روشن می نماید.

ریزشهای نفتی و زیانهای ناشی از آن، به توسعه روش های نوینی منجر شده است که انتقال و سرنوشت لکه های نفت را پیش بینی می کند.

شبکه های سنسور می باشد. روبات های با قابلیت پائین، می توانند هریک به یک فرستنده و گیرنده بی سیم مجهز شده و از این طریق به برقراری ارتباط با یکدیگر و تبادل اطلاعات بپردازند. روش پیشنهادی این مقاله جهت حل این مشکل استفاده از شبکه های حسگر و الگوریتم های تکاملی می باشد. شبکه حسگر^۱ شبکه ای است متشکل از تعداد زیادی گره کوچک است. در هر گره تعدادی حسگر وجود دارد. شبکه حسگر بشدت با محیط فیزیکی تعامل دارد و از طریق حسگرها اطلاعات محیط را گرفته و از طریق عملگرها واکنش نشان می دهد. ارتباط بین گره ها بصورت بی سیم بوده و هرگره بطور مستقل و بدون دخالت انسان کار میکند. که این امر نیازمند پروتکلی برای مسیر یابی در شبکه سنسور می باشد. در مراجع [۱] و [۲] مروری بر پروتکل های مختلف مسیر یابی در شبکه سنسور ارائه شده است.

ادامه مقاله بدین صورت سازماندهی شده است. در بخش.....

۲. حرکت دسته جمعی ذرات (PSO)

۱-۲). حرکت دسته جمعی ذرات (استاندارد)

حرکت جمعی ذرات (PSO)، یک تکنیک بهینه سازی احتمالی است که بر مبنای جمعیت کار میکند. این روش در سال ۱۹۹۵ توسط دکتر ابرهارت (Eberhart) و دکتر کندی (Kennedy) ارائه شد [۱] و ایده اصلی آن از رفتار دسته جمعی ماهیها یا پرندگان به هنگام جستجوی غذا الهام گرفته شده است. گروهی از پرندگان در فضایی به صورت تصادفی دنبال غذا میگردند. تنها یک تکه غذا در فضای مورد بحث وجود دارد. هیچ یک از پرندگان محل غذا را نمیدانند. یکی از بهترین استراتژیها میتواند دنبال کردن پرنده ای باشد که کمترین فاصله را تا غذا داشته باشد. این استراتژی در واقع جانمایه الگوریتم PSO است. در الگوریتم PSO، هر راه حل که به آن یک ذره گفته میشود، معادل یک پرنده در الگوی حرکت جمعی پرندگان میباشد. هر ذره یک مقدار شایستگی دارد که توسط یک تابع شایستگی محاسبه میشود. هر چه ذره در فضای جستجو به هدف- غذا در مدل حرکت پرندگان- نزدیکتر باشد، شایستگی بیشتری دارد. همچنین هر ذره دارای یک سرعت است که هدایت حرکت ذره را بر عهده دارد. هر ذره با دنبال کردن ذرات بهینه در حالت فعلی، به حرکت خود در فضای مساله ادامه میدهد.

آغاز کار PSO به این شکل است که گروهی از ذرات (راه حلها) به صورت تصادفی به وجود می آیند و با به روز کردن نسلیها سعی در یافتن راه حل بهینه مینمایند. در هر گام، هر ذره با

استفاده از دو بهترین مقدار به روز میشود. اولین مورد، بهترین موقعیتی است که تا کنون ذره موفق به رسیدن به آن شده است. موقعیت مذکور با نام pbest شناسایی و نگهداری میشود. بهترین مقدار دیگری که توسط الگوریتم مورد استفاده قرار میگیرد، بهترین موقعیتی است که تا کنون توسط جمعیت ذرات بدست آمده است. این موقعیت با gbest نمایش داده میشود. در برخی ویرایشهای، ذره، قسمتهایی از جمعیت را که همسایگان توپولوژیکی اش هستند، انتخاب و تنها آنها را در اعمال خود دخیل میکند که در این حالت بهترین راه حل محلی که با lbest نشان داده میشود، به جای gbest مورد استفاده قرار میگیرد. پس از یافتن بهترین مقادیر، سرعت و مکان هر ذره با استفاده از معادلات (۱) و (۲) به روز می شود.

(!)

$$v[] = v[] + c1 * rand() * (pbest[] - position[]) + c2 * rand() * (gbest[] - position[])$$

(!)

$$position[] = position[] + v[]$$

در معادلات (۱) و (۲)، $v[]$ سرعت ذره و $position[]$ محل فعلی ذره هستند که هر دو، آرایه هایی به طول تعداد ابعاد مساله میباشند. $rand()$ یک عدد تصادفی در بازه (0,1) است. $C1$ و $C2$ نیز پارامترهای یادگیری هستند و معمولاً $C1=C2=2$ در نظر گرفته میشود. سرعت ذرات در هر بعد به یک مقدار $Vmax$ محدود میشوند. اگر مجموع شتابها باعث شوند که سرعت در یک بعد از $Vmax$ بیشتر شود، مقدار سرعت در آن بعد برابر با $Vmax$ قرار میگیرد. شبه کد الگوریتم PSO در شکل (۱) مشاهده میشود.

سمت راست معادله (۱) از سه قسمت تشکیل شده است که قسمت اول، سرعت فعلی ذره است و قسمتهای دوم و سوم تغییر سرعت ذره و چرخش آن به سمت بهترین تجربه شخصی و بهترین تجربه گروه را به عهده دارند. اگر قسمت اول را در این معادله در نظر بگیریم، آنگاه سرعت ذرات تنها با توجه به موقعیت فعلی و بهترین تجربه ذره و بهترین تجربه جمع تعیین میشود. به این ترتیب، بهترین ذره جمع، در مکان خود ثابت باقی میماند و سایرین به سمت آن ذره حرکت میکنند. در واقع حرکت دسته جمعی ذرات بدون قسمت اول معادله (۱)، پروسه ای خواهد بود که طی آن فضای جستجو به تدریج کوچک میشود و جستجوی محلی حول بهترین ذره شکل میگیرد. در مقابل اگر فقط قسمت اول معادله (۱) را در نظر بگیریم، ذرات راه عادی خود را میروند

^۱ Wireless Sensor Actor Network (WSAN)

تا به دیواره محدوده برسند و به نوعی جستجویی سراسری را انجام میدهند.

ذرات میل بیشتری به حرکات انفجاری و تجربه های تازه دارند و با گذشت زمان این میل جای خود را به دنباله روی بیشتر از بهترینها میدهد. این روش در بسیاری موارد میتواند مشکل، گیر افتادن در مینیمم های محلی را کاهش دهد. با توجه به ماهیت پویای آب دریاها به علت وجود امواج و دیگر عوامل محیطی از نوع خاصی از الگوریتم حرکت دسته جمعی ذرات که مخصوص محیط های دینامیک است استفاده می کنیم .

۳. حرکت دسته جمعی ذرات (دینامیک)

در محیط های دینامیک نمی توانیم از الگوریتم استاندارد حرکت دسته جمعی ذرات برای حل مسئله استفاده کنیم . برای حل مسائل دینامیک از الگوریتم حرکت دسته جمعی ذرات اصلاح شده ای استفاده می کنیم که با بهره گیری از مکانیزم کشف تغییر و پاسخ توانایی دنبال کردن هدف در تغییرات متوالی محیط را دارد . مشکل اصلی PSO استاندارد در محیط های دینامیک این است که سر انجام ذرات به یک هدف یا نقطه بهینه محلی هم گرا می شوند و به موجب آن تنوع مورد نیاز میان ذرات برای عملیات اکتشاف و یافتن محل بهینه کاهش پیدا میکند . با تغییر محیط به علت عدم وجود تنوع کافی میان ذرات الگوریتم PSO استاندارد توانایی دنبال کردن تغییرات محیط را ندارد .

در الگوریتم PSO دینامیک به منظور کشف تغییر ایجاد شده از یک تابع کشف تغییر (Detection Function) استفاده میکنیم . در این تابع اگر به تعداد مشخصی تکرار الگوریتم در مقدار gbest تغییری حاصل نشد تابع احتمال تغییر محیط و متعاقباً تغییر نقطه بهینه را حدس می زند . در مرحله بعد به منظور ایجاد تنوع میان ذرات از تابع (Create Diversity) استفاده میکنیم . در این تابع برای ایجاد تنوع به هر یک از ابعاد ذره به صورت رندوم عددی را اضافه می کنیم .

۳. روش پیشنهادی

مسأله یافتن آلودگی های نفتی و میعانات گازی در دریا را می توان به عنوان زیرمجموعه ای از مسائل کاوش محیط مطرح کرد . کاوش محیط دارای مقاصد مختلفی از قبیل کاوش و مانیتور کردن محیط به منظور تولد یک واقعیت مجازی از آن [۹]، کاوش مناطق ناشناخته و تولید نقشه آنها [۱۰] ، مانیتور محیط برای تشخیص وقوع پدیده هائی نظیر آتش و سیل [۱۷] و تشخیص موقعیت [۱۸] میباشد . مسأله یافتن آلودگی های نفتی و میعانات گازی در دریاها و اقیانوس ها به عنوان زیرمجموعه ای از مسائل کاوش محیط مطرح میباشد . یافتن

```

For each particle
  Initialize particle
End For
Do
  For each particle
    Calculate fitness value of the particle  $f_p$ 
    //updating particle's best fitness value
    so far)
    If  $f_p$  is better than pBest
      set current value as the new pBest
    End For
    updating population's best fitness value so
    //far)
    Set gBest to the best fitness value of
    all particles
  For each particle
    Calculate particle velocity according
    equation (1)
    Update particle position according
    equation (2)
  End For
While maximum iterations OR
  minimum error criteria is not
  attained

```

شکل ۱- شبه کد الگوریتم PSO

در معادله (۱) با ترکیب این دو عامل سعی شده است تا موازنه ای بین جستجوی محلی و سراسری برقرار گردد. در [۵]، یک پارامتر جدید توسط ارائه دهندگان الگوریتم اولیه PSO ، ارائه گردید. این پارامتر جدید وزن میانی نامیده و با W نشان داده شد. هدف از معرفی این پارامتر برقراری موازنه بهتر، بین جستجوی محلی و جستجوی سراسری است. به این منظور در معادله (۱)، این ضریب در سرعت اولیه ذره ضرب میشود و به عبارتی تنها قسمتی از سرعت اولیه به سرعت آینده ذره منتقل میشود و به این ترتیب معادله (۱) به شکل معادله (۳) تغییر میکند.

$$v[i] = w * v[i] + c1 * rand() * (pbest[i] - position[i]) + c2 * rand() * (gbest[i] - position[i]) \quad (1)$$

وزن میانی می تواند یک ضریب ثابت یا یک تابع خطی با زمان و یا حتی یک تابع غیرخطی با زمان نیز باشد. در بسیاری از کاربردها وزن میانی به صورت تابعی خطی با زمان در نظر گرفته میشود که تابعی نزولی است. به این ترتیب در ابتدا قسمت بیشتری از سرعت فعلی ذره در سرعت آینده اش دخیل میشود و با گذر زمان، این میزان کاهش می یابد. به عبارت بهتر، در ابتدا

آلودگی در یک محیط می تواند به سه زیربخش (! یافتن نشانه ای از آلودگی در محیط، ۲) حرکت به سمتی که میزان آلودگی سنس شده بیشتر شود و ۳) تشخیص محل آلودگی تقسیم می شود [۵].

در روش پیشنهادی گروهی از روبات ها که مجهز به سنسوری جهت اندازه گیری میزان آلودگی محیط می باشد برای یافتن آلودگی (لکه های نفتی و میعانات گازی) در محیط مستقر می شوند در واقع هر روبات به عنوان یک گره از شبکه حسگر (ماهیت فیزیکی) و همچنین به عنوان یک ذره، در الگوریتم PSO (ماهیت الگوریتمیک) است که دارای تجهیزات متناسب میباشد .

این طور فرض میکنیم که در شروع ربات ها توسط یک هواپیما به محدوده مورد نظر منتقل و در محیط به صورت تصادفی مستقر می شوند . این مرحله در واقع منطبق بر گام اولیه الگوریتم یعنی مقدار دهی اولیه ذره ها در الگوریتم PSO می باشد . در مرحله بعد روباتها شروع به کاوش در محیط و در محدوده معین شده می کنند. مکان بعدی هر روبات با توجه به الگوریتم PSO مشخص می شود. روبات ها می توانند در سه بعد حرکت کنند و مقدار آلودگی محیط اطراف خود را توسط سنسور اندازه گیری کنند . ویژگی این روش در این است که می توانیم مقدار مشخصی از آلودگی را برای روبات مشخص کنیم و به عنوان هدف برای روبات تعریف کنیم .

هر روبات، مکان بیشترین آلودگی سنس شده توسط سنسور خود را در حافظه خود ذخیره می کند. روبات ها توسط یک شبکه بی سیم با یکدیگر در ارتباط اند و مکان بیشترین آلودگی سنس شده توسط کل شبکه را به اشتراک میگذارند. ویژگی دیگر الگوریتم PSO کاهش محاسبات محلی برای هر روبات و همچنین عدم نیاز به فضای زیاد جهت ذخیره سازی اطلاعات می باشد که این ویژگی خود نشان دهنده ویژگی دیگری نیز می باشد محاسبات و ارتباط با دیگر سنسور ها مقدار زیادی از انرژی هر روبات را صرف می کند . از این رو چون در الگوریتم PSO تنها چیزی که میان روبات ها منتقل می شود gbest یا همان مکان بیشترین آلودگی می باشد و همچنین برای هر روبات تنها دو مقدار gbest و pbest ذخیره می شود از این رو روش پیشنهادی روشی بهینه، چه از نظر فضا، جهت ذخیره سازی اطلاعات و چه از نظر حجم محاسبات و ارتباطات میان سنسور ها می باشد . البته روش های دیگری جهت بالا بردن طول عمر شبکه و کاهش مصرف انرژی مورد استفاده قرار می گیرد که در مراجع [۱۱][۱۲] ذکر شده است .

مطابق گام های الگوریتم، روبات مقدار غلظت آلودگی محیط اطراف خود را اندازه میگیرد و با مقدار ذخیره شده در pbest مقایسه می کند اگر این مقدار بیشتر از pbest باشد

مکان فعلی جایگزین مقادیر قبلی می شود در غیر این صورت روبات به حرکت خود بر اساس الگوریتم ادامه می دهد. همچنین در هر گام مقادیر pbest همه روبات ها با مقدار gbest مقایسه می شود. در صورتی که مقداری بیشتر از gbest ثبت شده باشد، جایگزین مقدار فعلی gbest می شود .

مکان بعدی هر روبات توسط مکان بالاترین آلودگی سنس شده توسط خودش و همچنین مکان بالاترین آلودگی سنس شده توسط کل شبکه تعیین می گردد. در واقع الگوریتم با استفاده از هوش دسته جمعی همه روبات ها را به سمت نقطه هدف هدایت می کند . در واقعیت روبات ها به سمت رو بانی میل می کنند که بالاترین مقدار آلودگی را ثبت کرده و اگر در مسیر حرکت روباتی مقدار آلودگی بیشتری را سنس کند روبات ها به سمت محل جدید میل می کنند .

مراجع

- [۱] محمد شیبانی، اتوماتای یادگیر سلولی، "انواع و کاربردهای آن"، سمینار کارشناسی ارشد، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیر کبیر، ۱۳۸۴.
- [2] Wolfram, S., "Cellular Automata, Los Alamos Science", vol. 9, pp. 2-21, Fall 1983.
- [3] Wolfram, S., "Universality and complexity in cellular automata", Physica D, no. 10, pp. 1-35, January 1984.
- [4] Narendra K. S. and Thathachar M.A.L., "Learning Automata: An Introduction", Prentice Hall, 1989.
- [5] Beigy, H. and Meybodi, M. R., "A Mathematical Framework for Cellular Learning Automata", Advances on Complex Systems, Vol. 7, Nos. 3-4, pp. 295-320, September/December 2004
- [6] Thathachar, M. A. L. and Sastry, P. S., "Varieties of Learning Automata: An Overview", IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 32, No. 6, PP. 711-722, 2002.
- [۷] محمدرضا مبینی، حمید بیگی و مسعود طاهرخانی، "اتوماتای یادگیر سلولی"، در مجموعه مقالات ششمین کنفرانس انجمن کامپیوتر ایران، ص ۱۶۳-۱۵۳، ۱۳۷۹.
- [8] Beigy, H. and Meybodi, M. R., "A Mathematical Framework for Cellular Learning Automata", Proceedings of 9th Annual CSI Computer Conference, Computer Engineering Department, Sharif University, Tehran, Iran, pp.151-159, Feb. 2004
- [۹] یاسر مهدوی فر، "اتوماتای سلولی و کاربردهای آن"، سمینار کارشناسی ارشد، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیر کبیر، ۱۳۸۳
- [10] Rastegar, R. and Meybodi, M. R., "A New Evolutionary Computing Model Based on Cellular Learning Automata", IEEE conference on Cybernetics and Intelligent Systems 2004 (CIS2004), Singapore, December 2004

Problems" Lecture Notes in Computer Science, Vol. 2723, 2003, pp.375-276

[17] Silagadze, Z. K. "Finding Two-Dimensional Peaks", <http://arxiv.org/abs/physics/0402085>

[18] <http://www.geatbx.com/docu/fcnindex-01.html>

[19] <http://optalgtoolkit.sourceforge.net/index.html>

[20] Hariri, A., Rastegar, R., Navi, K., Zamani, M. S. and Meybodi, M. R., "Cellular Learning Automata based Evolutionary Computing (CLA-EC) for Intrinsic Hardware Evolution", Proceedings of NASA/DoD Conference on Evolvable Hardware (EH'05), pp. 294-297, Washington DC, USA, June 29-July 1, 2005.

[21] Rastegar, R., Meybodi, M. R. and Hariri, A., "A New Fine Grained Evolutionary Algorithm based on Cellular Learning Automata", International Journal of Hybrid Intelligent Systems, IOS Press, Volume 3, Number 2, pp. 83-98, 2006.

[22] Beigy, H. and Meybodi, M. R., "Open Synchronous Cellular Learning Automata" Advances in Complex Systems, 2007, to appear.

[23] Beigy, H. and Meybodi, M. R., "Asynchronous Cellular Learning Automata" Automatica, Journal of International Federation of Automatic Control, 2007, to appear.

[11] Rastegar, A., Arasteh, A. R., Hariri, A. and Meybodi, M. R., "A Fuzzy Clustering Algorithm Using Cellular Learning Automata based Evolutionary Algorithm", Proceedings of Fourth International Conference on Hybrid Intelligent Systems (HIS'04), pp. 310-314, Japan, Kitakyushu, Dec. 2004

[۱۲] رضا رستگار، "تخصیص کانال در شبکه‌های سلولی مخابراتی با استفاده از اتوماتای یادگیر سلولی"، پایان نامه کارشناسی ارشد، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیر کبیر، ۱۳۸۳

[13] Masoodifar, B., Meybodi, M. R. and Rastegar, R., "Asynchronous CLA-EC", Proceedings of 11th Annual CSI Computer Conference of Iran, Fundamental Science Research Center (IPM), Computer Science Research Lab., Tehran, Iran, pp. 447-458, Jan. 24-26, 2006

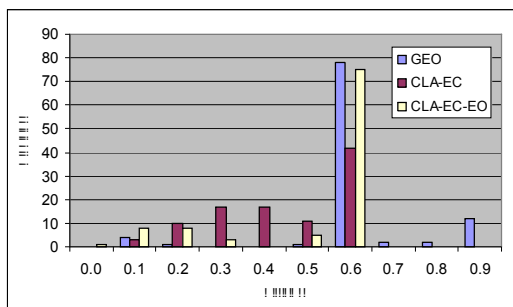
[14] Tucotte, D. L., "Self-Organized Criticality", Rep. Prog. Phys., Vol. 62, 1999, pp. 1377-1429

[15] Boettcher, S. and Percus, A. G., "Extremal Optimization: An Evolutionary Local-Search Algorithm", <http://arxiv.org/abs/cs.NE/0209030>

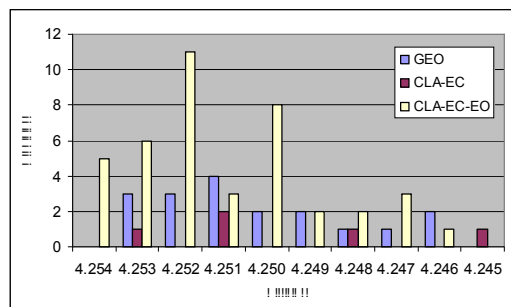
[16] Fabiano Luis de Sousa and Valeri Vlassov and Fernando Manuel Ramos, "Centralized Extremal Optimization for Solving Complex Optimal Design

جدول (۱): توابع آزمایشی استفاده شده

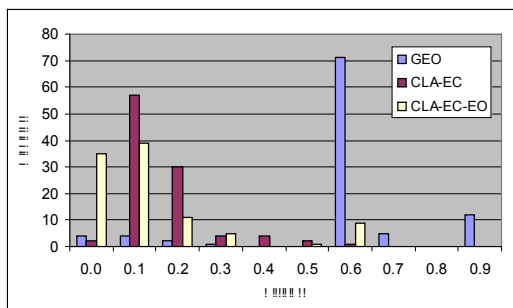
تایع	رابطه تایع	بازه مقایر ورودی
Bumps	$f(x) = x \times \sin(4 \times \Pi \times x) - y \times \sin(4 \times \Pi \times y + \Pi) + 1$	$-1 \leq x, y \leq 2$
Squashed Frog	$t = (x - 0.1)^2 + (y - 0.2)^2$ $f(x) = 1 + \sqrt[4]{t} - \cos(5 \times \Pi \times \sqrt{t})$	$-2 \leq x, y \leq 2$
Langermann	$f(x) = - \sum_{i=1}^m c_i \left(e^{-\frac{\ x - A(i)\ ^2}{\Pi}} \times \cos\left(\Pi \times \ x - A(i)\ ^2\right) \right)$	$0 \leq x, y \leq 10$ برای مشاهده مقادیر A و c به [۱۷] و [۱۹] مراجعه شود. $\bar{x}$ برداری با مولفه‌ها x و y است.
CPF1	$f(x) = x y + x + y $	$-10 \leq x, y \leq 10$
Ackley's Path	$f(x) = -a \times e^{-b \sqrt{\frac{x^2 + y^2}{2}}} - e^{\frac{\cos(cx) + \cos(cy)}{2}} + a + e$	$-32.768 \leq x, y \leq 32.768$ $a = 20, b = 0.2, c = 2\Pi$
Easom	$f(x) = -\cos(x) \times \cos(y) \times e^{-((x-\Pi)^2 + (y-\Pi)^2)}$	$-100 \leq x, y \leq 100$
De Jong F2	$f(x) = 100 \times (y - x^2)^2 + (1 - x^2)$	$-2.048 \leq x, y \leq 2.048$



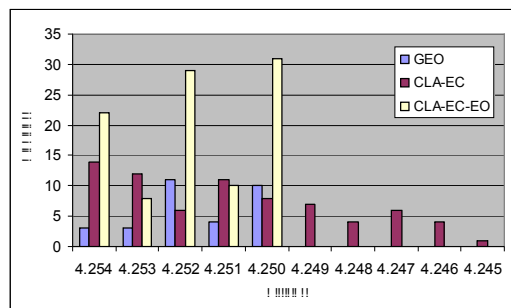
(الف)



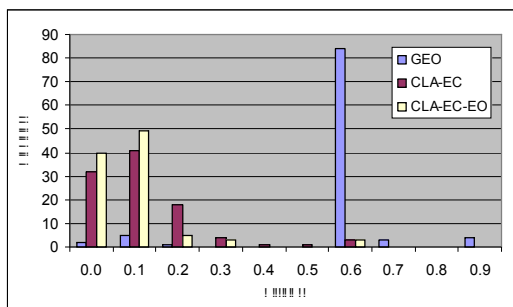
(الف)



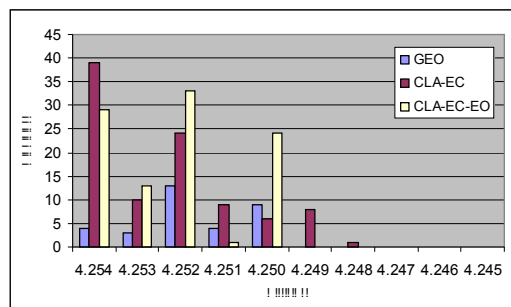
(ب)



(ب)



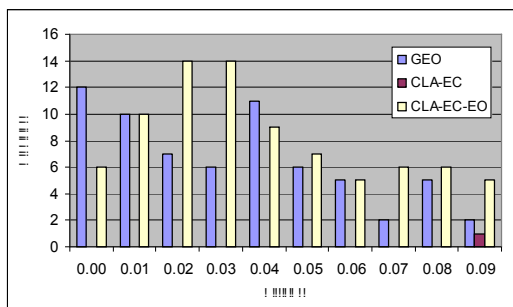
(ج)



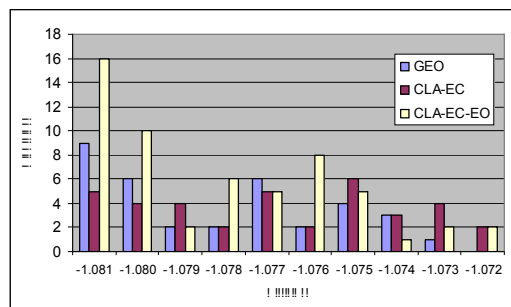
(ج)

شکل (۸): مقایسه الگوریتم‌ها برای تابع Squashed Frog
(الف) ۱۰۰۰ تکرار، (ب) ۱۰۰۰۰ تکرار و (ج) ۳۰۰۰۰ تکرار

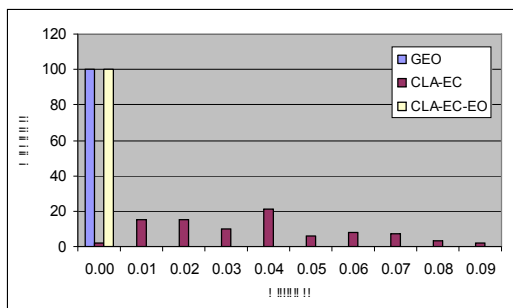
شکل (۷): مقایسه الگوریتم‌ها برای تابع Bumps
(الف) ۱۰۰۰ تکرار، (ب) ۱۰۰۰۰ تکرار و (ج) ۳۰۰۰۰ تکرار



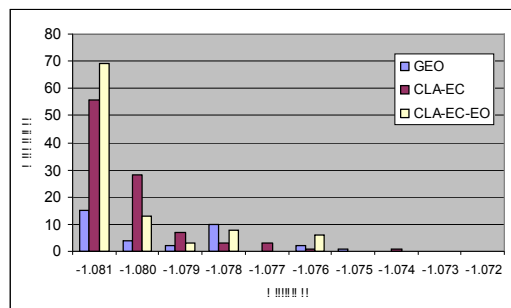
(الف)



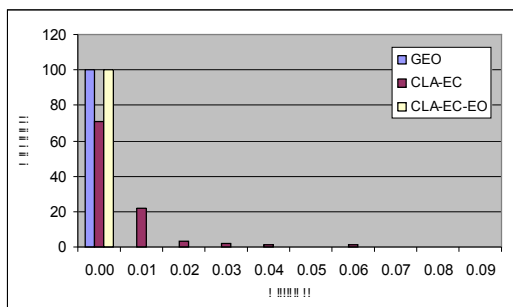
(الف)



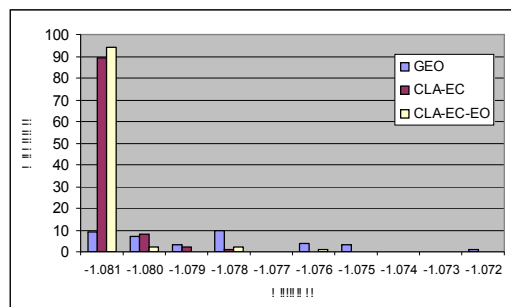
(ب)



(ب)



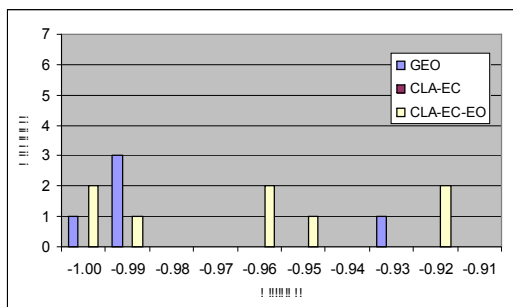
(ج)



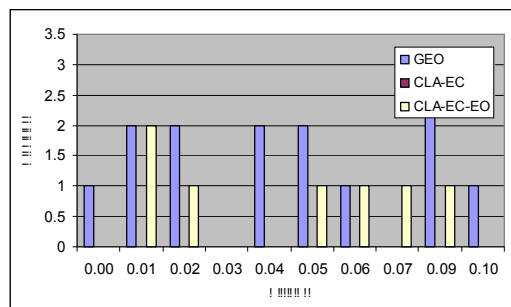
(ج)

شکل (۱۰): مقایسه الگوریتم‌ها برای تابع $CPF1$
(الف) ۱۰۰۰ تکرار، (ب) ۱۰۰۰۰ تکرار و (ج) ۳۰۰۰۰ تکرار

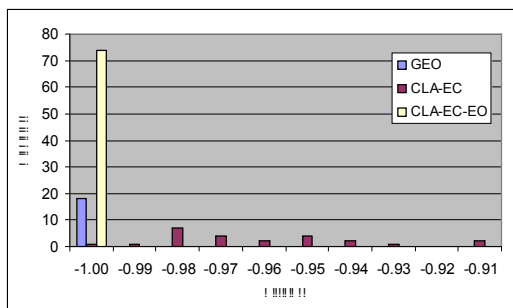
شکل (۹): مقایسه الگوریتم‌ها برای تابع $Langermann$
(الف) ۱۰۰۰ تکرار، (ب) ۱۰۰۰۰ تکرار و (ج) ۳۰۰۰۰ تکرار



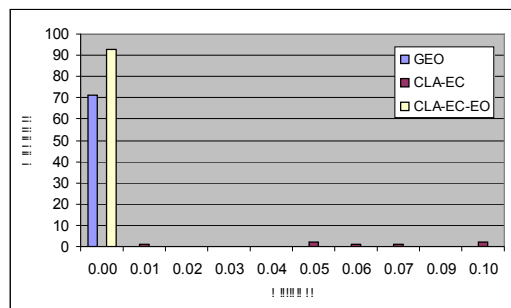
(الف)



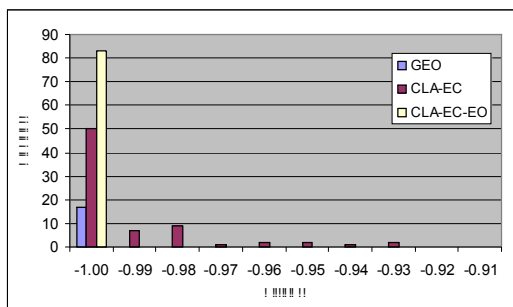
(الف)



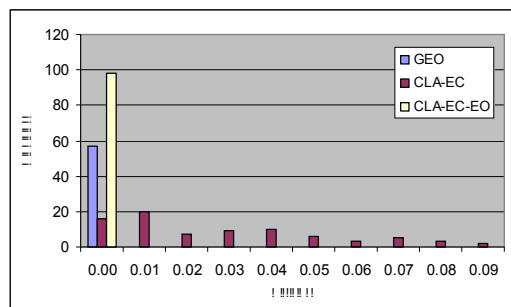
(ب)



(ب)



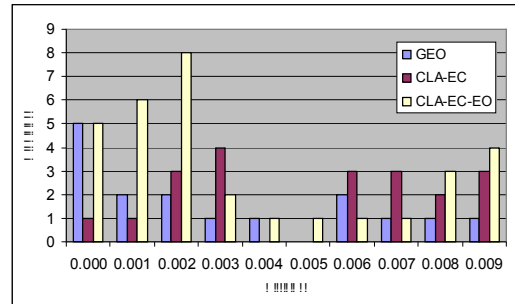
(ج)



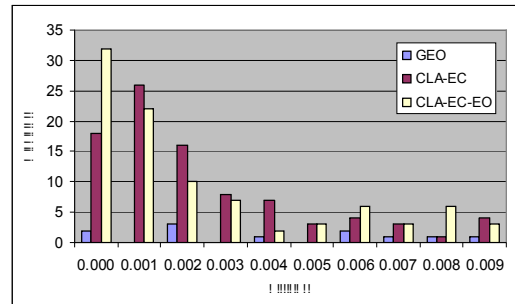
(ج)

شکل (۱۲): مقایسه الگوریتم‌ها برای تابع Easom
(الف) ۱۰۰۰ تکرار، (ب) ۱۰۰۰۰ تکرار و (ج) ۳۰۰۰۰ تکرار

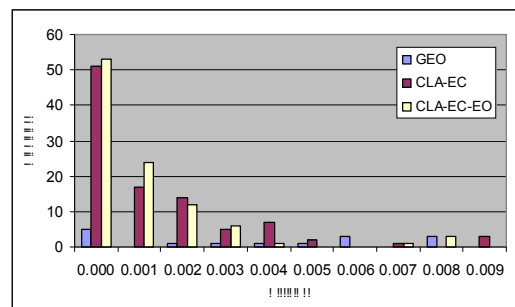
شکل (۱۱): مقایسه الگوریتم‌ها برای تابع Ackley's Path
(الف) ۱۰۰۰ تکرار، (ب) ۱۰۰۰۰ تکرار و (ج) ۳۰۰۰۰ تکرار



(الف)



(ب)



(ج)

شکل (۱۳): مقایسه الگوریتم‌ها برای تابع De Jong F2
 (الف) ۱۰۰۰ تکرار، (ب) ۱۰۰۰۰ تکرار و (ج) ۳۰۰۰۰ تکرار