

یک روش بهینه‌سازی ترکیبی (CLA-EC + بهینه‌سازی حدی)

آیدین خاتم نژاد محمدرضا میبدی

ازمايشگاه محاسبات نرم

دانشکده مهندسي کامپيوتر و فناوري اطلاعات

دانشگاه صنعتی اميركبير

تهران ايران

چکیده: الگوريتم بهينه‌سازی حدی تعميم يافته برای مسائلی که بيش از یک نقطه بهينه دارند به خوبی عمل نکرده و با احتمال بالايی در نقاط بهينه محلی گرفتار می‌شود. با اين حال، اين الگوريتم اين ويزگی مهم را دارد که در صورت قرار گرفتن در اطراف نقطه بهينه، جواب را با دقت بالايی محاسبه می‌کند. از طرف ديگر مطالعات بر روی الگوريتم بهينه سازی CLA-EC نشان می‌دهند که اين الگوريتم با وجود جستجوی مناسب دامنه مساله، جواب‌هایی با دقت بالا تولید نمی‌کند. در اين مقاله يک الگوريتم بهينه سازی ترکیبی که آن را CLA-EC-EO می‌نامیم و از ترکیب الگوريتم بهينه سازی حدی تعميم يافته و الگوريتم بهينه سازی CLA-EC حاصل می‌شود پیشنهاد می‌گردد. آزمایش‌ها نشان می‌دهد که الگوريتم پیشنهادی ويزگی‌های مطلوب هر دو الگوريتم پایه خود را دارا می‌باشد.

كلمات کلیدی: بهینه‌سازی حدی تعميم يافته، CLA-EC، بهینه‌سازی

الگوريتم بهينه‌سازی حدی تعميم يافته^۱، تعميم يافته يکی از الگوريتم‌های مشتق شده از الگوريتم بهينه‌سازی حدی به نام الگوريتم بهينه‌سازی حدی با پارامتر α است. اين الگوريتم يک الگوريتم ابر اكتشافي است که نياز الگوريتم بهينه‌سازی حدی به تابع ارزیابی شایستگی محلی اجزا را از بين می‌برد. از ويزگی‌های جالب اين الگوريتم می‌توان به سرعت بالاي همگرايی، دقت بالاي جواب‌ها و تعداد پارامترهای کم آن (اين الگوريتم تنها يک پارامتر به نام α برای تنظيم دارد) اشاره کرد. هر چند اين الگوريتم جواب‌های بسیار دقیقی برای مسائل تولید می‌کند، احتمال گرفتار شدن آن در نقاط بهينه محلی بسیار بالا است. در واقع اين الگوريتم دامنه مساله را به شکل خوبی جستجو نکرده و در نتیجه در اولین نقطه بهينه‌ای که پیدا می‌کند متوقف می‌شود. اين مساله باعث شده است که استفاده از اين الگوريتم تنها به مجموعه کوچکی از مسائل (مسائلی با تنها يک نقطه بهينه) محدود شود. در اين مقاله برای حل اين مشکل پیشنهاد می‌شود که الگوريتم بهينه‌سازی حدی تعميم يافته در کنار الگوريتم بهينه‌سازی ديگری به نام CLA-EC که اخیرا معرفی شده است قرار گیرد تا در شرایطی که به تنهایی از کارایی بالايی بر خوردار نیست بتواند با کمک گرفتن از CLA-EC عملکرد خود را بهبود بخشد. علت اين پیشنهاد است که بنا بر آزمایش‌های انجام شده بر روی الگوريتم CLA-EC، مشاهده شده است که اين الگوريتم

۱. مقدمه

بهينه‌سازی حدی و الگوريتم‌های مشتق شده از آن، دسته جديدي از الگوريتم‌ها هستند که برای حل مسائل بهينه‌سازی مورد استفاده قرار می‌گيرند. اين الگوريتم‌ها بر پایه مدل تکاملی بک-اسپن شکل گرفته‌اند. اين مدل که نمونه‌ای از يک سистем دارای ويزگی بحران‌های خود سازمانده است، چگونگی تکامل گونه‌های مختلف موجودات را نمایش می‌دهد.

الگوريتم‌های مبتنی بر بهينه‌سازی حدی^۱ بر خلاف ديگر الگوريتم‌های بهينه‌سازی نظير الگوريتم ژنتيك که همواره از مجموعه‌ای از جواب‌ها (جمعيت) تشکيل شده‌اند، در هر لحظه تنها يک جواب در اختيار دارند و در هر مرحله اين جواب را بهبود می‌بخشند؛ بهبود جواب با انتخاب يکی از اجزای جواب و جايگزين کردن مقدار آن با مقداری جديد انجام می‌شود. برای انتخاب جزء تعويضی، الگوريتم از شایستگی محلی اجزا استفاده می‌کند؛ بنابراین اين الگوريتم‌ها علاوه بر تابع شایستگی کلي جواب، به تابعی برای ارزیابی شایستگی اجزای جواب (شایستگی محلی) نيز احتياج دارند. واضح است که اين تابع وابسته به نوع مساله بوده و برای هر مساله باید به صورت جداگانه تعریف و پياده‌سازی شود.

¹ Extremal Optimization

² Generalized Extremal Optimization

و وسیع که با نام بهمن^۴ شناخته می‌شود، خود را به یک حالت پایدار برساند. این سیستم‌ها تا مرز آشفتگی رفته ولی توسط بهمن به حالت عادی بر می‌گردند. تعداد و وسعت بهمن‌ها از قانون توان^۵ پیروی می‌کند:

$$y = ax^k \quad (1)$$

در این رابطه x وسعت بهمن (با توجه به نوع سیستم) می‌تواند واحدهای مختلفی داشته باشد)، a و k مقادیر ثابت که برای هر سیستم به صورت جداگانه تعریف می‌شود و α تعداد دفعات وقوع بهمن است. به عنوان مثال اگر زمین لرزه را به عنوان بهمن در نظر بگیریم، تعداد زمین لرزه‌ها با بزرگی‌های مختلف از این قانون پیروی می‌کند.

در حال حاضر روشی برای اثبات اینکه آیا سیستمی دارای ویژگی بحران خود سازمانده است یا خیر وجود ندارد و تشخیص وجود این ویژگی در یک سیستم به صورت تجربی انجام می‌شود. با این حال، تعدادی خصوصیت مشترک بین این نوع سیستم‌ها مشاهده شده است که عبارتند از: تاثیرگذاری اجزا سیستم به صورت محلی، قانون‌های ساده بین اجزاء، انتقال تغییرات بوسیله قوانین بین اجزاء، حرک خارجی با نرخ کم و پیروی حداقل یکی از اجزای سیستم از قانون توان. در صورتی که این ویژگی‌ها در سیستمی مشاهده شوند، می‌توان انتظار داشت که ویژگی بحران خود سازمانده نیز در این سیستم وجود دارد.

۲-۲. مدل بک-اسنپن^۶

مدل بک-اسنپن مدلی برای نمایش تکامل گونه‌های موجودات است که بر اساس مفهوم بحران خود سازمانده ایجاد شده است. این مدل از آرایه‌ای از سلول‌ها تشکیل شده است که در آن هر سلول نماینده یک گونه جانوری است؛ ابتدا و انتهای آرایه سلول‌ها در کنار هم قرار دارند به نحوی که آرایه صورت یک حلقه در می‌آید. در این مدل به هر گونه جانوری یک شایستگی اختصاص داده می‌شود که نشان دهنده شایستگی و مقدار تطابق آن با محیط است؛ این شایستگی به صورت عددی در $[0,1]$ در هر سلول نگهداری می‌شود.

روش کار الگوریتم مدل بک-اسنپن به این صورت است که در ابتدا همه سلول‌ها با عددی اتفاقی مقداردهی می‌شوند. سپس تا توقف الگوریتم در هر مرحله، سلوی که کمترین شایستگی را دارد (در صورتی که چندین سلول با کمترین شایستگی وجود داشته‌ند یکی از آنها به صورت اتفاقی انتخاب می‌شود) به همراه سلول‌های چپ و راست آن مجدداً به صورت اتفاقی مقداردهی

بر خلاف الگوریتم بهینه‌سازی حدی تعییم یافته دامنه مساله را به شکل مطلوبی جستجو می‌کند اما با وجود اینکه محدوده جواب را درست تشخیص می‌دهد، جواب‌های دقیقی تولید نمی‌کند. در واقع نقطه‌های قوت هر کدام از این دو الگوریتم متنطبق بر نقطه ضعف‌های الگوریتم دیگر است. بنابراین انتظار می‌رود که در صورت ترکیب این دو الگوریتم که آن را الگوریتم-CLA-EC EO می‌نامیم، بتوان ضعف‌های هر دو الگوریتم را مرتفع نمود.

ادامه این مقاله بدین صورت سازماندهی شده است. در بخش ۲ بحران‌های خودسازمانده و بهینه‌سازی حدی و در بخش ۳ اتماتای سلولی، اتوماتای یادگیر، اتوماتای یادگیر سلولی و مدل CLA-EC به اختصار شرح داده می‌شوند. در بخش ۴ الگوریتم پیشنهادی معرفی می‌شود و دو بخش انتهایی اختصاص به ارایه نتایج آزمایش‌ها و نتیجه گیری دارد.

۲. بحران‌های خودسازمانده و بهینه‌سازی حدی

الگوریتم بهینه‌سازی حدی تعییم یافته - همانطور که از نام آن مشخص است - تعییم یافته الگوریتم دیگری به نام بهینه‌سازی حدی می‌باشد که از اعمال تغییراتی در مدل تکاملی بک-اسنپن به وجود آمده است. مدل بک-اسنپن نیز خود نمونه‌ای از یک سیستم با ویژگی بحران خود سازمانده است.

در این بخش ابتدا بحران‌های خود سازمانده و مدل بک-اسنپن به صورت خلاصه معرفی می‌شوند. سپس بهینه‌سازی حدی و بهینه‌سازی حدی با پارامتر α معرفی و نهایتاً الگوریتم بهینه‌سازی حدی تعییم یافته شرح داده می‌شود.

۱-۱. بحران خود سازمانده^۷

بحران خود سازمانده [۱۴] یکی از ویژگی‌های سیستم‌های طبیعی است که بسیاری از رفتارهای پیچیده این سیستم‌ها را توصیف می‌کند. این مفهوم اولین بار توسط بک^۸ برای توصیف رفتار مدل توده شن معرفی شد. در حال حاضر از این مفهوم در شبیه‌سازی و پیش‌بینی زلزله و رانش‌زمین، شبیه‌سازی آتش‌سوزی در جنگل‌ها، بررسی انتشار بیماری‌ها، تغییرات آب و هوای بررسی بازار خرید و فروش کالاها استفاده می‌شود.

سیستم‌هایی که دارای ویژگی بحران خود سازمانده هستند معمولاً از یک حالت پایدار شروع به کار کرده و در طول زمان به یک نقطه بحرانی^۹ نزدیک می‌شوند. در این نقطه، کوچکترین تغییری در وضعیت، باعث می‌شود که سیستم با یک رفتار سریع

⁴ Avalanche

⁵ Power Law

⁶ Bak-Sneppen Model

⁷ Self-Organized Criticality (SOC)

⁸ Bak

⁹ Critical Point

- . جوابی به عنوان جواب اولیه تولید شده و به عنوان جواب جاری و همچنین به عنوان بهترین جواب در نظر گرفته می‌شود.
- . بدترین سلول جواب جاری انتخاب شده و مقدار آن با مقدار دیگری جایگزین می‌شود.
- . جواب ایجاد شده به عنوان جواب جاری در نظر گرفته می‌شود.
- . در صورتی که جواب جاری بهتر از بهترین جواب باشد به عنوان بهترین جواب نیز انتخاب می‌شود.
- . مرحله ۲ تا رسیدن به جواب مورد نیاز تکرار می‌شود.
- . بهترین جواب برگردانده می‌شود.

الگوریتم بهینه‌سازی حدی برای حل مسائلی نظیر تقسیم گراف به دو زیر گراف، رنگ آمیزی گراف با سه رنگ و فروشنده دوره‌گرد مورد استفاده قرار گرفته و جواب‌های مطلوبی ارائه داده است.

۴-۲. بهینه‌سازی حدی با پارامتر τ

هرچند الگوریتم بهینه‌سازی حدی جواب‌های مطلوبی برای مسائل به کار گرفته شده ارائه می‌دهد، با این حال احتمال گرفتار شدن آن در نقاط بهینه محلی و متوقف شدن آن در مکانی به جز جواب مساله زیاد است. برای حل این مشکل پارامتر جدیدی به نام τ به الگوریتم بهینه‌سازی حدی اضافه شده و الگوریتم جدیدی به نام بهینه‌سازی حدی با پارامتر τ [۱۵] معرفی شده است؛ تفاوت این الگوریتم جدید و الگوریتم بهینه‌سازی حدی به مرحله انتخاب سلول تعویضی مربوط می‌شود. در این الگوریتم برای پیدا کردن سلول تعویضی، ابتدا سلول‌ها بر اساس شایستگی و به صورت صعودی مرتب می‌شوند. سپس به هر سلول مقداری برابر P_k که از رابطه 2 قابل محاسبه است اختصاص داده می‌شود؛ در این رابطه k برابر ترتیب سلول در فهرست مرتب شده است:

$$P_k \propto k^{-\tau} \quad (2)$$

پس از نسبت دادن مقادیر P_k به سلول‌ها، سلول تعویضی به اینصورت انتخاب می‌شود که: در حلقه‌ای به طور اتفاقی یکی از سلول‌ها انتخاب می‌شود.

عددی به صورت اتفاقی بین 0 و 1 انتخاب می‌شود. در صورتی که عدد تولید شده کوچکتر از P_k سلول باشد سلول به عنوان سلول تعویضی انتخاب می‌شود و حلقه پایان می‌پذیرد، در غیر اینصورت حلقه تکرار می‌شود.

مراحل کلی الگوریتم بهینه‌سازی حدی با پارامتر τ به شرح زیر است:

می‌شوند. توقف الگوریتم می‌تواند به صورت دستی انجام شود و یا با تعریف آستانه‌ای به خود الگوریتم محول شود.

اگر در مرحله جاری، مجموعه‌ای از کوچکترین شایستگی در هر کدام از مراحل قبل را تشکیل دهیم، بزرگترین عضو این مجموعه شایستگی بحرانی نامیده می‌شود. واضح است که شایستگی بحرانی همواره مقدار ثابتی ندارد و در طول اجرا به صورت صعودی تغییر پیدا می‌کند. آزمایش‌هایی که بر روی این مدل انجام شده است نشان می‌دهد که شایستگی بحرانی دارای حدی برابر 667^* است که با گذشت زمان به سمت آن میل می‌کند.

همانطور که اشاره شد مدل بک-اسپن دارای ویژگی بحران خود سازمانده است، بنابراین باید دارای پدیده بهمن نیز باشد. بهمن در این مدل زمانی شروع می‌شود که مقدار اختصاص داده به یک سلول (سلول با شایستگی کم یا یکی از همسایه‌های آن) از شایستگی بحرانی آن مرحله کمتر شود. پایان بهمن زمانی است که با اختصاص مقادیر جدید به سلول‌ها، هیچ سلولی با شایستگی کمتر از شایستگی بحرانی مرحله وجود نداشته باشد. بزرگی بهمن نیز معادل تعداد سلول‌هایی است که در طول بهمن شایستگی کمتر از شایستگی بحرانی پیدا می‌کنند.

مدل بک-اسپن را می‌توان به اینصورت به دنیای واقعی ارتباط داد که تغییر خانه با کوچکترین شایستگی، معادل این واقعیت است که گونه‌ها با شایستگی‌های کمتر بیشتر در معرض انقراض قرار دارند و مقداردهی همسایه‌های آن نیز به این مطلب برمی‌گردد که در طبیعت تغییرات یک گونه در گونه‌های مجاور آن نیز تاثیرگذار است.

۴-۳. بهینه‌سازی حدی

بهینه‌سازی حدی [۱۵] الگوریتمی اکتشافی است که از مدل بک-اسپن الهام گرفته شده است. همانطور که از نام آن مشخص است این الگوریتم برای حل مسائل بهینه‌سازی استفاده می‌شود. در این الگوریتم سلول‌های مدل بک-اسپن برابر اجزای جواب مساله قرار داده می‌شوند. جواب در هر مرحله (تکرار حلقه اصلی الگوریتم) با جایگزینی مقدار بدترین سلول (سلول با کمترین شایستگی) با مقداری دیگر بهبود داده می‌شود. در این الگوریتم برخلاف مدل بک-اسپن، با تغییر یک سلول (گونه در مدل بک-اسپن) سلول‌های همسایه آن تغییر داده نمی‌شوند.

مشخص ترین تفاوت الگوریتم بهینه‌سازی حدی و دیگر الگوریتم‌های اکتشافی نظیر الگوریتم ژنتیک، نیاز الگوریتم به دانستن شایستگی سلول‌ها (شایستگی محلی) علاوه بر شایستگی کلی جواب است. در واقع این ویژگی الگوریتم بهینه‌سازی حدی، هسته مرکزی و عامل کار آن است. الگوریتم بهینه‌سازی حدی به صورت کلی شکل زیر را دارد:

دو بعدی باشد، ابتدا متغیرهای λ و μ را باید به صورت دودویی درآمده و سپس در کنار هم قرار داده شوند تا جواب به شکل آرایه‌ای از بیت‌ها در آید. به این ترتیب هر سلول در الگوریتم بهینه‌سازی حدی تعیین یافته معادل یک بیت از جواب خواهد بود؛ بنابراین تعداد سلول‌ها به دقت مورد نیاز و اندازه دامنه مساله بستگی دارد. واضح است که دامنه مقادیر مجاز برای سلول‌ها در الگوریتم اخیر، $0 \leq \lambda \leq 1$ می‌باشد و از این نظر در مقایسه با الگوریتم بهینه‌سازی حدی بسیار ساده‌تر است.

مساله دیگری که در الگوریتم بهینه‌سازی حدی تعمیم یافته مطرح است، محاسبه شایستگی سلول‌ها است. همانطور که در بخش‌های اشاره شد، محاسبه شایستگی محلی مهمترین مرحله در الگوریتم بهینه‌سازی حدی است. در دو الگوریتم بهینه‌سازی حدی و بهینه‌سازی حدی با پارامتر α ، شایستگی محلی به صورت تابعی وابسته به مساله تعریف می‌شود؛ در الگوریتم بهینه‌سازی حدی تعمیم یافته روشی برای محاسبه شایستگی سلول‌ها را ارائه شده که مستقل از نوع و شکل مساله است. طبق این روش در هر مرحله بر اساس جواب جاری و به تعداد سلول‌های آن، جواب‌هایی تولید می‌شود که به عنوان فرزندان جواب جاری شناخته می‌شوند. این جواب‌ها از هم متمایز بوده و هر کدام تنها در مقدار یک سلول با جواب جاری اختلاف دارند. به عنوان مثال اگر رشته دودوبی جواب جاری برابر $100\ 100\ 100\ 100\ 100\ 1$ باشد، پنج فرزند با مقادیر $0\ 100\ 100\ 100\ 100\ 1$ و $100\ 100\ 100\ 100\ 100\ 1$ برای آن تولید می‌شود. بعد از تولید شدن جواب‌های شایستگی فرزندی از جواب جاری که در مقدار آن سلول با جواب جاری اختلاف دارد منهای شایستگی جواب جاری. به این ترتیب شایستگی محلی اجزای جواب بدون نیاز به تابعی برای محاسبه شایستگی محلی و تهها بوسیله تابع شایستگی کلی مساله محاسبه می‌شوند. لازم به ذکر است که خروجی تابع شایستگی کلی برابر مقدار تابع در نقطه (جواب) مورد نظر است. همچنین باید توجه داشت که برای توابعی که مقدار بهینه یک نقطه ماکریزم است شایستگی هر سلول باید منفی در نظر گرفته شود. مراحل الگوریتم بهینه‌سازی حدی تعمیم یافته به ترتیب در ادامه نشان داده شده‌اند:

جواب اولیه‌ای تولید شده و به عنوان جواب جاری و همچنین به عنوان بهترین جواب در نظر گرفته می‌شود.

فرزندان جواب جاری تولید شده و شایستگی آنها محسسه مم رشود.

۳. جواب‌های تولید شده (فرزندان جواب جاری) با توجه به شایستگی، آنها به صورت صعودی مرتب می‌شوند.

جواب اولیه‌ای تولید شده به عنوان جواب جاری و همچنین به عنوان بهترین جواب در نظر گرفته می‌شود.

سلول‌ها با توجه به شایستگی آنها به صورت صعودی مرتب می‌شوند.

با توجه به رابطه ۲ یک سلول به عنوان سلول تعویضی انتخاب و مقداردهی مجدد می‌شود.

جواب تولید شده به عنوان جواب جاری در نظر گرفته می‌شود.

در صورتی که جواب جاری بهتر از بهترین جواب باشد
به عنوان بهترین جواب نیز انتخاب می‌شود.

مرحله ۲ تا رسیدن به جواب مورد نیاز تکرار می شود.
بهترین جواب برگردانده می شود.

لازم به ذکر است که الگوریتم ارائه شده و نحوه انتخاب سلول تعویضی ساده ترین شکل پیاده سازی بهینه سازی حدی با پارامتر α بوده و بسته به نوع مساله قابل تغییر خواهد بود.

بهینه سازی حدی با پارامتر α مانند الگوریتم بهینه سازی حدی برای حل مسائلی نظری تقسیم گراف به دو زیر گراف، رنگ آمیزی گراف با سه رنگ و فروشنده دوره گرد به کار گرفته شده است و جواب های مطلوب تری از الگوریتم مادر خود ارائه داده است. همچنین نتایج بدست آمده توسط این الگوریتم هم از نظر دقیق و هم از نظر سرعت، کاملاً قابل رقابت با الگوریتم هایی نظری شیوه سازی تابکاری^۱ و الگوریتم ژنتیک نشان داده است.

۲-۵. بهینه‌سازی حدی تعمیم یافته

همانطور که مطرح شد استفاده از الگوریتم بهینه‌سازی حدی نیازمند تعریف تابعی برای محاسبه شایستگی محلی سلول‌های جواب است و علاوه بر آن چگونگی تعویض سلول تعویضی به شکل مساله وابسته است. بنابراین نمی‌توان از آن به راحتی و بدون نیاز به پیاده‌سازی مجدد، برای حل دامنه وسیعی از مسائل استفاده کرد. به این منظور الگوریتم بهینه‌سازی حدی تعمیم یافته بر اساس الگوریتم بهینه‌سازی حدی با پارامتر τ معرفی شده است.

الگوریتم بهینه‌سازی حدی تعمیم یافته [۱۶] یک الگوریتم ابر-اکتشافی مانند الگوریتم زنگیک و شبیه‌سازی تابکاری است. از این الگوریتم می‌توان برای حل دسته وسیعی از مسائل استفاده کرد.

ساختار الگوریتم بهینه‌سازی حدی تعمیم یافته کاملاً شبیه الگوریتم بهینه‌سازی حدی با پارامتر α است. برای استفاده از این الگوریتم، ابتدا متغیرهای مساله باید به یک رشته دودویی نگاشت شوند؛ به عنوان مثال اگر مساله پیدا کردن نقطه بهینه یک تابع

¹ Simulated Annealing

- اگر مجموعه مقادیر سلول‌ها از دو عنصر تشکیل شده باشد، اتوماتا از نوع دودویی و در صورتی که تعداد عناصر بیشتر از دو عدد باشد اتوماتا از نوع چند مقداره خواهد بود.
- در صورتی که قوانین شکلی قطعی داشته باشند اتوماتا از نوع قطعی و در صورتی که احتمالی باشند اتوماتا از نوع احتمالی می‌باشد.
- هرگاه تمامی سلول‌های اتوماتا همزمان مقداردهی شوند اتوماتا از نوع همگام و در مقابل اگر سلول‌ها با ترتیبی خاص و غیر همزمان مقداردهی شوند اتوماتا از نوع ناهمگام خواهد بود.
- اگر سلول‌های اتوماتا به شکلی منظم در کنار یکدیگر قرار گرفته باشند و تشکیل آرایه‌ای یک بعدی، دو بعدی و یا چند بعدی را بهمند، اتوماتا از نوع منظم و در صورتی که سلول‌ها به شکلی نامنظم (مثلثاً به عنوان گره‌های یک گراف) قرار گرفته باشند اتوماتا از نوع نامنظم خواهد بود.

هر چند اتوماتای سلولی امکانات زیادی را برای شبیه‌سازی و تعریف سیستم‌های گوناگون در اختیار قرار می‌دهد با این حال استفاده از آن برای شبیه‌سازی سیستم‌های طبیعی، بزرگ و ناشناخته با یک اشکال بزرگ همراه است و آن تعیین فرم قطعی قوانین است؛ زیرا در اغلب سیستم‌ها وجود نویز و عدم قطعیت مانع از تعیین قوانینی قطعی برای سیستم می‌شود. البته راهکار احتمالی کردن قوانین برای حل این مشکل پیشنهاد شده است. با این حال تعیین مقادیر احتمالات قوانین برای سیستم‌های ناشناخته خود مشکلی بزرگ است که مانع از استفاده مطلوب از این مدل می‌شود.

۲-۳ اتوماتای یادگیر^۲

اتوماتای یادگیر^[۴] [۶] مانشیتی است که می‌تواند تعداد محدودی عمل را انجام دهد؛ هرگاه این مانشین عملی را انتخاب می‌کند، عمل انتخاب شده توسط محیط ارزیابی شده و نتیجه آن به صورت یک سیگنال بازخوردی ثابت (در صورت مناسب بودن عمل) یا منفی (در صورت نامناسب بودن عمل) به اتوماتا بازگردانده می‌شود. مقدار این سیگنال در انتخاب اعمال بعدی تاثیر می‌گذارد. هدف این فرایند این است که اتوماتا بعد از گذشت مدتی به سمت مناسب‌ترین عمل خود در محیط میل کرده و یا به عبارت دیگر یاد می‌گیرد که کدام عمل بهترین عمل است. نحوه تعامل اتوماتای یادگیر و محیط در شکل ۱ نشان داده است. همانطور که در شکل مشاهده می‌شود، محیط در اتوماتای یادگیر همانند یک تابع رفتار می‌کند؛ ورودی این تابع

. با توجه به رابطه ۲ فرزندی انتخاب می‌شود.
فرزند انتخاب شده به عنوان جواب جاری در نظر گرفته می‌شود.

در صورتی که جواب جاری بهتر از بهترین جواب باشد به عنوان بهترین جواب نیز انتخاب می‌شود.
مرحله ۲ تا رسیدن به جواب مورد نیاز تکرار می‌شود.
بهترین جواب برگردانده می‌شود.

مهتمترین مزیت الگوریتم بهینه‌سازی حدی تعمیم یافته در مقایسه با الگوریتم‌های اکتشافی دیگر، دارای بودن تنها یک پارامتر (۲) است و از این نظر استفاده از آن بسیار ساده می‌باشد. سرعت هم‌گرا شدن این الگوریتم بسیار بالا بوده و همچنین نحوه حرکت این الگوریتم به سمت جواب به گونه‌ای است که در صورت رسیدن به جواب، جواب را با دقت بسیار بالایی تولید و محاسبه می‌کند. با این حال، علی‌رغم تمیزاتی که برای جلوگیری از گرفتار شدن الگوریتم در نقاط بهینه محلی در نظر گرفته شده است، این الگوریتم برای مسائلی که دارای نقاط بهینه محلی زیاد و پراکنده هستند، خوب عمل نکرده و جواب‌های مطلوبی ارائه نمی‌دهد.

۳. اتوماتای سلولی، اتوماتای یادگیر، اتوماتای CLA-EC یادگیر سلولی و مدل

در این بخش از مقاله اتوماتای سلولی، اتوماتاهای یادگیر و اتوماتای یادگیر سلولی به اختصار شرح داده و سپس مدل تکاملی CLA-EC که بر مبنای اتوماتای یادگیر سلولی ایجاد شده معرفی خواهد شد.

۳-۱ اتوماتای سلولی^۱ (CA)

اتوماتای سلولی [۱][۲][۳] یک مدل ریاضی برای سیستم‌هایی است که از اجزا و مولفه‌های ساده تشکیل شده، بویا بوده، با مرور زمان تغییر کرده و بر اساس ارتباط محلی بین اجزای خود فعالیت می‌کنند. اجزای سیستم در مدل اتوماتای سلولی بوسیله مجموعه‌ای منظم از سلول‌ها نمایش داده می‌شوند که در آن هر سلول معادل یکی از اجزای سیستم است. در هر قدم زمانی به هر سلول مقداری از مجموعه‌ای نامتناهی نسبت داده می‌شود که نشان دهنده وضعیت سلول است. این مقداردهی بر اساس مقدار جاری سلول، سلول‌های همسایه آن و قانون تعريف شده برای اتوماتای سلولی انجام می‌شود؛ از آنجایی که قانون در نظر گرفته شده تنها مقدار سلول و همسایه‌های آن را مورد استفاده قرار می‌دهد، قانون محلی نیز نامیده می‌شود. دسته‌بندی‌های مختلفی را می‌توان برای اتوماتای سلولی در نظر گرفت که از جمله آنها می‌توان به موارد زیر اشاره کرد:

² Learning Automata

¹ Cellular Automata

اتوماتای سلولی با یک اتوماتای یادگیر جایگزین شده است. به این ترتیب این مدل نه تنها امکان استفاده از چندین اتوماتای یادگیر را به صورت توأم فراهم می کند بلکه مشکل تعیین فرم قطعی قوانین در اتوماتای سلولی را نیز مرتفع می کند.

نحوه عملکرد و به روز رسانی سلول های اتوماتای یادگیر سلولی شباهت بسیار زیادی به اتوماتای سلولی دارد با این تفاوت که قبل از انجام اعمال به روز رسانی، ابتدا هر کدام از اتوماتاهای یادگیر عملی را انتخاب می کنند. سپس، عمل انتخاب شده توسط هر سلول در کنار اعمال انتخاب شده توسط اتوماتاهای همسایه آن به وسیله قانون محلی اتوماتای سلولی بررسی و نتیجه آن به عنوان پاسخ محیط و سیگنال بازخوردی به سلول برگردانده می شود. در نهایت، اتوماتای یادگیر هر سلول با توجه به سیگنال بازخوردی که از محیط دریافت کرده است، یا به وضعیت حدیدی می رود (در صورتی که از ساختار ثابتی داشته باشد) و یا توسط الگوریتم یادگیری بردار احتمال خود را به روز می کند (در صورتی که ساختار متغیری داشته باشد).

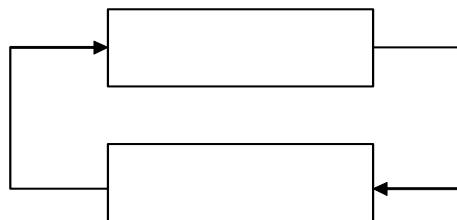
آنواع مختلفی از همسایگی ها برای اتوماتای سلولی و اتوماتای یادگیر سلولی وجود دارد که با توجه به کاربرد و شرایط مساله انتخاب و یا تعریف می شوند. پر کاربرد ترین همسایگی هایی که در این اتوماتاهای استفاده می شوند ون نیومن، مور، اسمیت و کول می باشند.

قانون های محلی برای اتوماتای یادگیر سلولی بر اساس چگونگی استفاده از وضعیت و اعمال سلول ها به سه دسته قانون عمومی^۴، قانون کلی^۵ و قانون خارجی کلی^۶ تقسیم می شوند. برای اطلاعات بیشتر در باره اتوماتای سلولی یادگیر میتوان به مراجع [۵][۲۳][۲۲] مراجعه نمود.

۴-۳. مدل CLA-EC

[۱۰] یک الگوریتم تکاملی است که از ترکیب CLA و مفاهیم مطرح در محاسبات تکاملی به وجود آمده است و از زمان معرفی تا امروز در مسائل مختلف مورد استفاده قرار گرفته است [۱۱][۱۲][۱۳][۲۰]. ساختار استفاده شده در الگوریتم CLA-EC از چندین ژنوم تشکیل شده است که هر کدام در یکی از سلول های یک اتوماتای یادگیر سلولی قرار گرفته اند. هر ژنوم از دو مولفه رشته ژنومی و مدل ژنومی تشکیل شده است رشته ژنومی که به صورت رشته ای از بیت ها می باشد معادل یکی از راه حل های میانی مساله است. مدل ژنومی نیز مجموعه ای از اتوماتاهای یادگیر می باشد که بر اساس تجارب گذشته خود و ژنوم های دیگر آموزش می بینند. به این ترتیب فرآیند تکاملی به طریقی هدایت می شود تا شایستگی رشته

عمل انتخاب شده توسط اتوماتا و خروجی آن سیگنال بازخوردی است.



شکل (۱) : تعامل اتوماتای یادگیر و محیط

دو نوع پیاده سازی برای اتوماتاهای یادگیر وجود دارد که عبارتند از اتوماتای یادگیر با ساختار ثابت^۱ و اتوماتای یادگیر با ساختار متغیر^۲. اتوماتای یادگیر با ساختار متغیر که در این مقاله استفاده شده است از برداری به نام بردار احتمال انتخاب اعمال تشکیل شده است؛ همانطور که از نام این بردار برمی آید اعضای آن نشان دهنده احتمال انتخاب عملی از بین اعمال اتوماتا هستند. بنابراین جمع احتمالات این بردار برابر ۱ و تعداد اعضای آن برابر تعداد اعمال خواهد بود. اتوماتا با توجه به این بردار عمل بعدی خود را انتخاب می کند و بعد از دریافت پاسخ محیط آن را به روز می کند. الگوریتمی که برای به روز رسانی بردار احتمال بر اساس ورودی (پاسخ محیط) مورد استفاده قرار می گیرد الگوریتم یادگیری نام دارد. یکی از متدائل ترین الگوریتم های یادگیری، الگوریتم یادگیری خطی است. در این الگوریتم هر گاه در مرحله n ام عمل α انتخاب شده باشد و این عمل از طرف محیط پاداش گرفته باشد بردار احتمال با توجه به رابطه 3 به روز می شود:

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1-a)p_j(n) \quad \forall j \neq i \end{aligned} \quad (3)$$

و در صورتی که عمل انتخاب شده از طرف محیط جریمه شود به روز رسانی طبق رابطه 4 انجام می شود:

$$\begin{aligned} p_i(n+1) &= (1-b)p_i(n) \\ p_j(n+1) &= (b/r - 1) + (1-b)p_j(n) \quad \forall j \neq i \end{aligned} \quad (4)$$

در این روابط a پارامتر پاداش و b پارامتر جریمه می باشند.

۳-۳. اتوماتای یادگیر سلولی^۳ (CLA)

بسیاری از مسائل را نمی توان با یک اتوماتای یادگیر حل کرد؛ در این موارد مجموعه ای از اتوماتاهای یادگیر می توانند به حل مساله کمک کنند. در [۲۰][۷] مدلی متشکل از دو مفهوم اتوماتای سلولی و اتوماتای یادگیر ارائه شده است که با نام اتوماتای یادگیر سلولی شناخته می شود. در این مدل هر سلول در

⁴ General

⁵ Totalistic

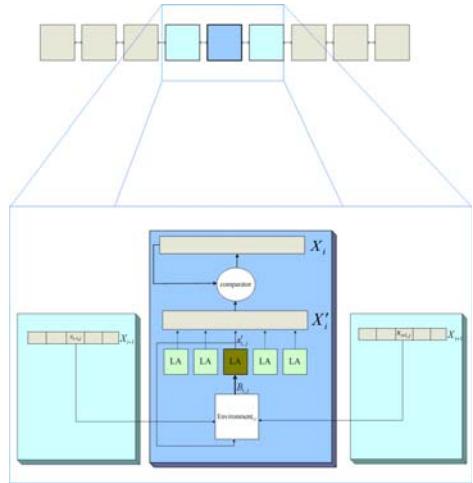
⁶ Outer totalistic

¹ Fixed Structure Learning Automata

² Variable Structure Learning automata

³ Cellular Learning Automata

که برای تولید سیگنال تقویتی استفاده می‌شوند و استراتژی تولید سیگنال تقویتی روشی است که سیگنال تقویتی بر اساس سلول‌های انتخاب شده ایجاد می‌شود. شکل ۲ معماری یکی از سلول‌ها در مدل CLA-EC را نمایش می‌دهد [۱۳].



شکل (۲) : ساختار CLA-EC

۴. الگوریتم پیشنهادی (CLA-EC-EO)

همانطور که قبلاً اشاره شد، بهینه‌سازی حدی تعمیم یافته ممکن است در یکی از نقاط بهینه محلی گرفتار شود ولی این الگوریتم این ویژگی مهم را دارد که در صورت قرار گرفتن در اطراف نقطه بهینه، جواب را با دقت بالایی پیدا می‌کند. از طرف دیگر نتایج گزارش شده برای الگوریتم CLA-EC نشان می‌دهد که این الگوریتم با وجود جستجوی مناسب دامنه مساله، جواب‌هایی با دقت بالا تولید نمی‌کند. در این بخش یک الگوریتم ترکیبی به نام CLA-EO-EC که دارای مزایای هردو الگوریتم بهینه‌سازی حدی تعمیم یافته و CLA-EC می‌باشد پیشنهاد می‌گردد. در الگوریتم CLA-EC که در این مقاله استفاده شده است، همسایه‌های یک سلول تمامی همسایه‌های آن سلول تا یک شاعع مشخص و خود سلول را شامل می‌شود. بطور مثال همسایگی با شاعع یک به انتخاب سه سلول منتهی می‌شود. برای تولید سیگنال تقویتی هر اتوماتای یادگیر سلول (زنوم)، تمامی بیت‌ها در رشته‌های زنومی سلول‌های انتخاب شده با اتوماتای می‌شوند. اگر تعداد بیت‌های هم مقدار با بیت متناظر با اتوماتای یادگیر در سلول مورد نظر، از نصف تعداد سلول‌های انتخاب شده بیشتر باشد، اتوماتای یادگیر تشویق و در غیر اینصورت جریمه می‌شود.

در الگوریتم پیشنهادی، وظیفه جستجوی دامنه مساله به عهد CLA-EC و وظیفه دقیق‌تر شدن در جواب‌ها به عهد الگوریتم بهینه‌سازی حدی است. به بیان ساده‌تر، در الگوریتم پیشنهادی ابتدا فضای مساله توسط CLA-EC جستجو می‌شود و هر جا احتمال حضور جواب وجود داشته باشد بهینه‌سازی

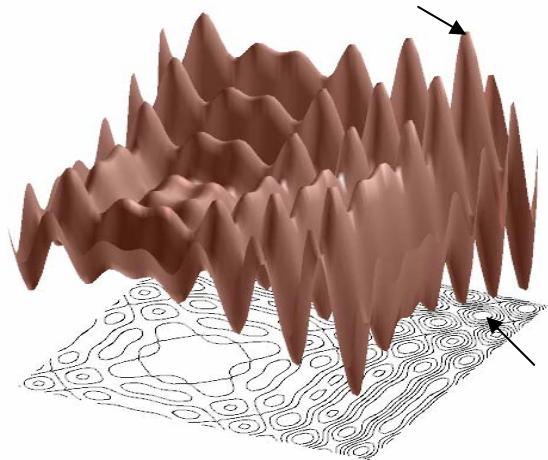
زنومی بهبود یافته و به سمت جواب مساله حرکت کند. نحوه انتساب اتوماتاهای یادگیر مدل زنومی به بیت‌های رشته زنومی به گونه‌های مختلفی قابل طرح است. برای مثال می‌توانیم رشته زنومی را به دسته‌های دو بیتی تقسیم کرده و به ازای هر دسته یک اتوماتای یادگیر در نظر بگیریم. به این ترتیب با فرض دودویی بودن فضای جستجو برای رشته زنومی به طول $n/2$ (با فرض زوج بودن)، تعداد $\frac{n}{2}$ اتوماتای یادگیر، هر کدام با چهار عمل خواهیم داشت. برای مثال برای این مورد اگر یک اتوماتای یادگیر عمل ۳ را انتخاب کند به معنای آن است که بیت ۱ و ۲ دسته متناظر آن مقادیر ۱ و ۰ را می‌پذیرند. در این مقاله با فرض مستقل بودن متغیرهای مساله، ساده‌ترین شکل مدل زنومی که در آن برای هر بیت در رشته زنومی یک اتوماتای یادگیر در مدل زنومی خواهیم داشت در نظر می‌گیریم؛ هر اتوماتای یادگیر مقدار بیت متناظر با خود را تعیین می‌کند.

همانطور که اشاره شد زنوم‌ها در الگوریتم CLA-EC در سلول‌های یک اتوماتای یادگیر سلولی در کنار هم قرار گرفته‌اند. بنابراین با فرض همگام بودن این اتوماتا، سلول‌ها به طور همزمان رشته‌های زنومی خود و همسایگان خود را مورد بررسی قرار داده و بر اساس تابع شایستگی از میان آنها تعدادی را به عنوان رشته‌های زنومی مناسب انتخاب می‌کند (معادل انتخاب زوج در طبیعت). این شیوه انتخاب دو طرفه نمی‌باشد؛ به عبارت دیگر اگر یک زنوم، زنوم دیگری را به عنوان یکی از کاندیدهای خود انتخاب نماید، تضمینی برای انتخاب شدن این زنوم از طرف زنوم مقابل وجود نخواهد داشت (البته این روش معادلی در بین سیستم‌های طبیعی ندارد). زنوم بر اساس زنومهایی که انتخاب نموده است، یک بردار سیگنال تقویتی ساخته و به اتوماتاهای یادگیر خود می‌دهد (این مکانیسم در معادل جفت گیری زنوم مورد نظر می‌باشد).

برای تولید رشته زنومی جدید هر سلول، در هر گام اتوماتای یادگیر مدل زنومی سلول عملی را انتخاب و مقدار جدیدی برای بیت‌های متناظر خود در رشته زنومی اعلام می‌کنند. در صورتی که رشته زنومی جدید شایستگی بیشتری نسبت به رشته زنومی قبلی داشته باشد، جایگزین رشته قیمی سلول می‌شود؛ در غیر اینصورت تغییری در رشته زنومی سلول ایجاد نمی‌شود. این عمل برای تمامی سلول‌های اتوماتای یادگیر سلولی انجام شده و به این ترتیب جمعیت جدیدی تولید می‌شود. این بخش از الگوریتم معادل یادگیری از تجرب قبلي زنوم و حفظ آنها در نسل‌های بعدی (زنوم‌های بعدی) و یا به بیان دیگر معادل تاثیر یادگیری بر تکامل از دیدگاه تئوری بالدوین می‌باشد. یک قانون CLA-EC از دو بخش تشکیل شده است: استراتژی انتخاب و استراتژی تولید سیگنال تقویتی. منظور از استراتژی انتخاب برای یک سلول، نحوه انتخاب تعداد مشخصی از همسایه‌های سلول است

نقاط بهینه محلی زیادی می‌باشد. ماکزیمم این تابع در شکل ۳ توسط یک پیکان نشان داده شده است

شکل‌های ۴، ۵ و ۶ نتایج اجرای الگوریتم‌های بهینه‌سازی حدی تعمیم یافته، CLA-EC-EO و CLA-EC بر روی تابع Bumps را بعد از ۳۰۰۰ ارزیابی نمایش می‌دهند. در این اشکال که تابع را از جهت محور Z (از بالا) نشان می‌دهد، نقطه بهینه مساله با یک فلش مشخص شده است. همانطور که در شکل ۴ نشان داده شده است، جواب‌های تولید شده توسط الگوریتم بهینه‌سازی حدی تعمیم یافته گستردگی بسیار کمی دارند و الگوریتم با رسیدن به اولین نقطه بهینه بر روی آن متمرکز شده و نهایتاً در همان نقطه متوقف می‌شود. این تابع یک نمونه از توابعی است که الگوریتم بهینه‌سازی حدی تعمیم یافته عملکرد الگوریتم CLA-EC را برای آن ندارد. شکل ۵ نحوه عملکرد الگوریتم Bumps نشان می‌دهد. همانطور که مشاهده می‌شود جواب‌های تولید شده توسط این الگوریتم پراکندگی بسیار زیاد و مناسبی دارند و تمامی دامنه مساله را پوشش می‌دهند و چگالی جواب‌ها در اطراف نقاط بهینه بخصوص نقطه بهینه عمومی بیشتر از سایر نقاط است.



شکل (۳): نحوه عملکرد الگوریتم بهینه‌سازی حدی برای تابع Bumps

شکل ۶ جواب‌های تولید شده توسط الگوریتم CLA-EO-EC را برای تابع Bumps نشان می‌دهد. همانطور که مشاهده می‌شود پراکندگی جواب‌های تولید شده توسط این الگوریتم کمتر از الگوریتم CLA-EC است. علت این اتفاق به رفتار الگوریتم بهینه‌سازی حدی تعمیم یافته مربوط می‌شود که جواب‌هایی با فاصله‌های بسیار کم و نزدیک به هم تولید می‌کند؛ الگوریتم CLA-EC-EO در محدوده نقاطی که احتمال حضور جواب بهینه در آن وجود دارد با دقت بیشتری به جستجوی پردازد.

بررسی شکل‌های ۴ تا ۶ همچنین نشان می‌دهد که رفتار الگوریتم CLA-EO-EC شباهت بیشتری به الگوریتم

حدی وارد عمل شده و محل مورد نظر را برای یافتن جواب بهتر با دقت بیشتری جستجو می‌کند. سپس، الگوریتم CLA-EC مجدداً کنترل را بر عهده گرفته و جستجو ادامه پیدا می‌کند. لازم به ذکر است که جستجوی دامنه توسط CLA-EC قبل جایگزینی با جستجوی اتفاقی نیست، زیرا CLA-EC به شکلی هدفمند فضا را جستجو کرده و الگوریتم بهینه‌سازی حدی را به شکل بهتری به سمت نقاط بهینه هدایت می‌کند. مراحل کلی الگوریتم CLA-EO-EC به شرح زیر است.

. مقداردهی و تولید جمعیت اولیه
برای هر سلول (زنوم)، اگر شایستگی سلول انتخاب شده از همه همسایه‌هایش بیشتر بود:

۱-۱. با توجه به رشته زنومی سلول، رشته زنومی جدید توسط الگوریتم بهینه‌سازی حدی تعمیم یافته تولید می‌شود. (رشته زنومی سلول به عنوان یک جواب در نظر گرفته شده، فرزندان آن تولید و بهترین آنها انتخاب می‌شود)

۲-۲. در صورتی که شایستگی زنوم جدید بیشتر از شایستگی زنوم سلول جاری باشد، رشته جدید جایگزین زنوم سلول شده و اوتماتاهای یادگیر سلول مقداردهی مجدد می‌شوند.

در غیر این صورت:
۳-۱. سیگنال‌های تقویتی اوتماتاهای یادگیر توسط الگوریتم CLA-EC ایجاد شده و سپس بر اساس این سیگنال‌های تقویتی اوتماتاهای یادگیر جریمه شده و یا پاداش می‌گیرد.

۳-۲. یک رشته زنومی جدید توسط اوتماتاهای یادگیر در هر سلول تولید می‌شود. اگر این رشته زنومی شایسته‌تر از رشته زنومی خود سلول باشد، جایگزین رشته زنومی سلول می‌شود.

تا محقق شدن شرط توقف مرحله ۲ تکرار می‌شود.
بهترین جواب اجرا به عنوان خروجی برگردانده می‌شود.
در این الگوریتم، مرحله ۲ مربوط به بخش بهینه‌سازی حدی و مرحله ۳ مربوط به بخش CLA-EC الگوریتم می‌باشد.

۵. نتایج آزمایش‌ها

در این بخش ابتدا به بررسی رفتار الگوریتم CLA-EO-EC و مقایسه آن با رفتارهای الگوریتم‌های بهینه‌سازی حدی تعمیم یافته و CLA-EC می‌پردازیم. سپس نتایج اجرای این الگوریتم‌ها بر روی چند مساله بهینه‌سازی ارائه می‌شود.

برای بررسی رفتار الگوریتم‌ها از تابع Bumps که در جدول ۲ و شکل ۳ نمایش داده شده استفاده شده است. این تابع دارای

برای مقایسه الگوریتم‌ها از هفت تابع Bumps، Ackley's، CPF1، Langermann، Squashed Frog و Easom Path استفاده شده است. از بین این توابع چهار تابع Langermann، De Jong F2 و Easom، Ackley's Path استاندارد و پرکاربرد برای مقایسه الگوریتم‌های بهینه‌سازی، CPF1 [۱۷]، علاوه بر چهار تابع فوق از سه تابع Bumps و Squashed Frog که در [۱۹] معرفی شده‌اند نیز برای بررسی الگوریتم‌ها استفاده شده است. این هفت تابع از نظر شکل و تعداد نقاط بهینه در سه گروه زیر قرار می‌گیرند:

توابعی با چندین نقطه بهینه که یک یا گروهی از این نقاط جواب مساله‌اند مانند تابع Bumps

Langermann و Squashed Frog

توابعی با یک نقطه بهینه که شیب تابع در همه نقاط به

سمت آن می‌باشد مانند تابع Ackley's و CPF1

.Path

توابعی با یک نقطه بهینه که محل آن از طریق دنبال

کردن شیب تابع پیدا نمی‌شود مانند تابع Easom

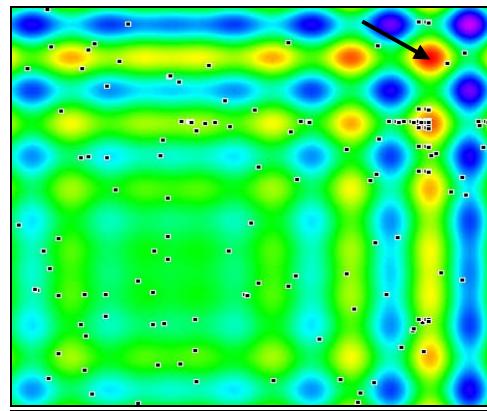
.De Jong F2

آزمایش‌های برای تعداد تکرارهای ۱۰۰۰، ۱۰۰۰۰ و ۳۰۰۰۰

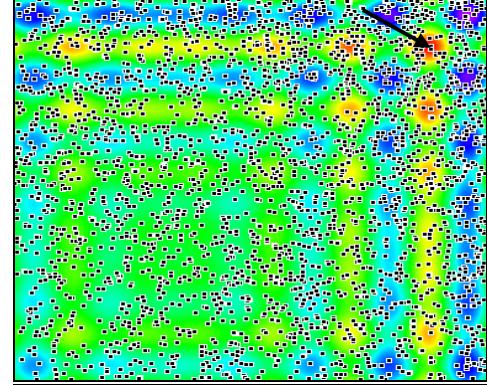
که به ترتیب با نامهای تعداد تکرار کم، متوسط و زیاد به آنها رجوع می‌شود انجام شده است. هر کدام از الگوریتم‌ها ۱۰۰ بار به ازای هر کدام از تعداد تکرارها بر روی هر یک از توابع آزمایشی اجرا شده و جواب‌های تولید شده به عنوان داده آماری برای مقایسه استفاده شده‌اند. برای این آزمایشها پارامتر τ برای الگوریتم بهینه‌سازی حدی تعیین یافته برابر ۱،۲۵، پارامترهای CLA-EC و پاداش برای الگوریتم‌های CLA-EC و CLA-EC-EO برابر ۱۰۰۱ و شاعع همسایگی برابر ۶۰ در نظر گرفته شده است. همچنین جمعیت اولیه تمامی الگوریتم‌ها برابر ۱۰۰ انتخاب شده است.

شکل‌های ۷ تا ۱۳ نمودارهای توزیع جواب‌ها در نزدیکی جواب بهینه (منتهی‌الیه سمت چپ محور افقی) را نشان می‌دهند. در این نمودارها محور افقی نشان دهنده مقدار جواب و محور عمودی نشان دهنده تعداد جواب‌ها می‌باشد. به عنوان مثال، شکل ۷-ب نشان میدهد که الگوریتم بهینه‌سازی حدی تعیین یافته ۳ بار، الگوریتم CLA-EC ۱۴ بار و الگوریتم CLA-EC-EO ۲۲ بار از ۱۰۰ اجرا به جواب بهینه همگرا شده اند. لازم به ذکر است که برای بالا بردن خوانایی نمودارها تنها ۱۰ جواب نزدیک به نقطه بهینه نشان داده شده و دقت محور عمودی نیز بر اساس بیشترین تعداد جواب‌ها در هر نمودار تنظیم شده است.

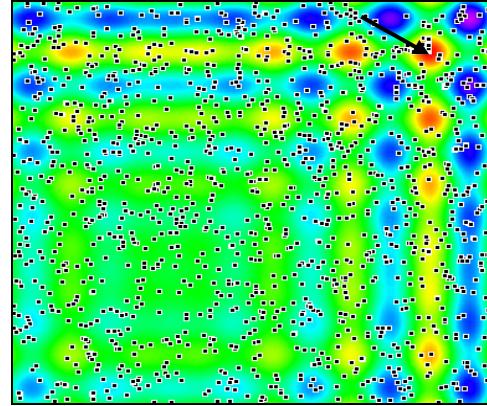
EC دارد تا به الگوریتم بهینه‌سازی حدی تعیین یافته. دلیل این امر را می‌توان در نحوه انتخاب بین مراحل ۲ و ۳ در الگوریتم CLA-EC-EO جستجو کرد. از آنجایی که از بین سلول‌های یک همسایگی تنها یک سلول بهترین است، قسمت بهینه‌سازی حدی الگوریتم فقط یکبار برای کل همسایگی اجرا می‌شود؛ به عبارت دیگر مرحله ۲ در الگوریتم CLA-EC-EO در صورتی که سلول انتخاب شده بهترین سلول همسایگی باشد اجرا می‌شود. به عنوان مثال اگر شاعع همسایگی برابر ۱۰ باشد، به ازای هر بار اجرای قسمت مربوط به بهینه‌سازی حدی (مرحله ۲)، ۲۰ بار الگوریتم CLA-EC (مرحله ۳) اجرا می‌شود. بنابراین شباهت الگوریتم CLA-EC و CLA-EC-EO توجیه‌پذیر و منطقی است.



شکل (۴): رفتار الگوریتم بهینه‌سازی حدی برای تابع Bumps



شکل (۵): رفتار الگوریتم CLA-EC برای تابع Bumps



شکل (۶): رفتار الگوریتم CLA-EO-EC برای تابع Bumps

Engineering Department, Sharif University, Tehran, Iran, pp.151-159 , Feb. 2004

[۹] یاسر مهدوی فر، "اتوماتی 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

[11] Rastegar, A., Arasteh, A. R., Harriri, 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 کاتال در شبکه‌های سلولی مخابراتی با استفاده از اتوماتی 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 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.

بررسی نمودارها نشان که الگوریتم CLA-EC-EO از هر دو الگوریتم پایه‌ای خود (CLA-EC و بهینه سازی حدی تعمیم یافته) برای توابعی که در این مقاله استفاده شده است بهتر عمل کرده است. توجه کنید که الگوریتم بهینه سازی حدی تعمیم یافته فقط برای مسایل CPF1 و Ackley's Path و الگوریتم CLA-EC فقط برای مسایل Bumps به Longermann مشاهده می‌شود که الگوریتم خوبی عمل می‌کنند. همچنین مشاهده می‌شود که الگوریتم پشنهدادی برای توابعی که دو الگوریتم پایه جواب‌های خوبی تولید نکرده‌اند، نیز دارای عملکرد خوبی می‌باشد.

۶. نتیجه‌گیری

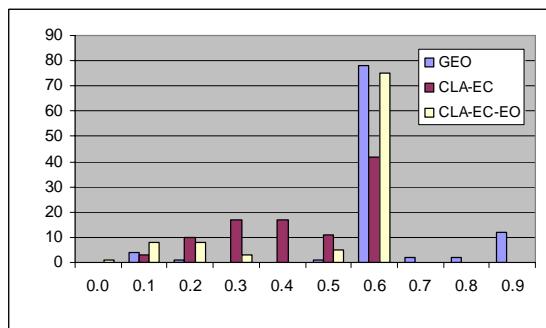
در این مقاله یک الگوریتم بهینه سازی ابر اکتشافی ترکیبی که از ترکیب الگوریتم‌های بهینه سازی حدی تعمیم یافته و CLA-EC حاصل شده است معرفی گردید. این الگوریتم ترکیبی مزایای هر دو الگوریتم بهینه سازی حدی تعمیم یافته و CLA-EC را دارد. نتایج آزمایش‌ها بر روی مسایل مختلف نشان داد که الگوریتم پیشنهادی هم از دقت الگوریتم بهینه سازی حدی تعمیم یافته و هم از قابلیت جستجوی بالای الگوریتم CLA-EC برخوردار است.

مراجع

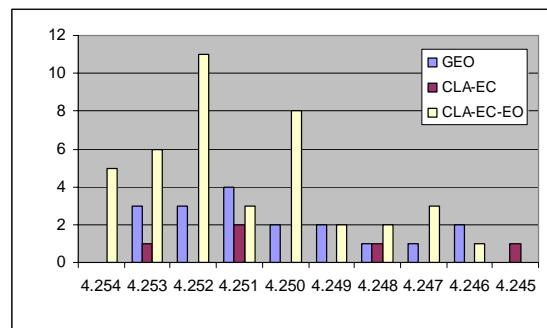
- [۱] محمد شیبانی، اتوماتی یادگیر سلولی، "انواع و کاربردهای آن" ، سمینار کارشناسی ارشد، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیر کبیر، ۱۳۸۴.
- [2] Wolfram, S., "Cellular Automata, Los Alamos Science", vol. 9, pp. 2-21, Fall 1983.
- [3] Wolfram, S., "Universality and complexity in 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 میبدی، حمید بیگی و مسعود طاهرخانی، "اتوماتی یادگیر سلولی" ، در مجموعه مقالات ششمین کنفرانس انجمن کامپیوتر ایران، ص ۱۳۷۹، ۱۵۳-۱۶۳
- [8] Beigy, H. and Meybodi, M. R., "A Mathematical Framework for Cellular Learning Automata", Proceedings of 9th Annual CSI Computer Conference, Computer

جدول (۱) : توابع آزمایشی استفاده شده

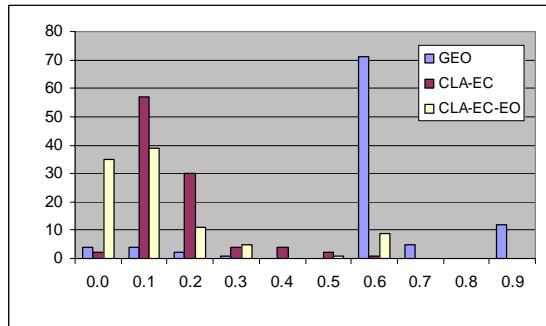
تابع	رابطه تابع	بازه مقایر ورودی
Bumps	$f(x) = x \times \sin(4 \times \Pi \times x) - y \times \sin(4 \times \Pi \times y + \Pi) + 1$	$-1 \leq x, y \leq 2$
Squashed Frog	$t = (x - 0.1)^2 + (y - 0.2)^2$ $f(x) = 1 + \sqrt[4]{t} - \cos(5 \times \Pi \times \sqrt{t})$	$-2 \leq x, y \leq 2$
Langermann	$f(x)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



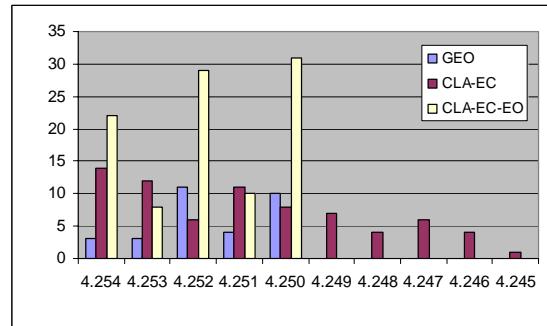
(الف)



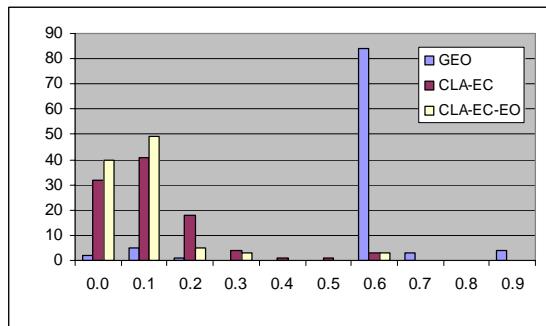
(الف)



(ب)

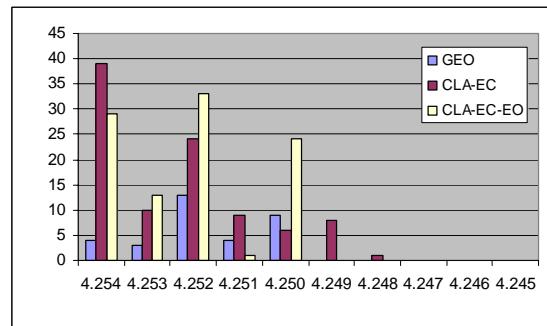


(ب)



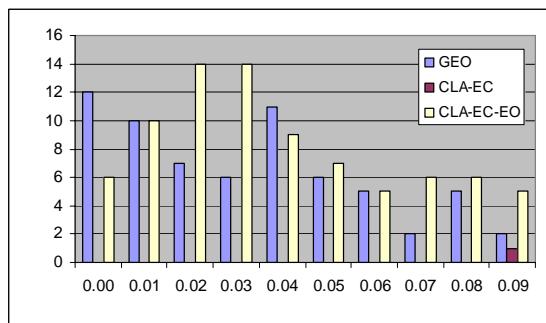
(ج)

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

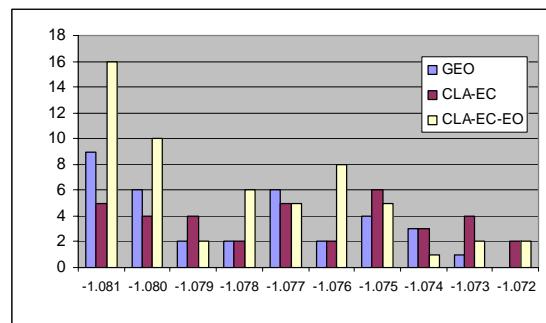


(ج)

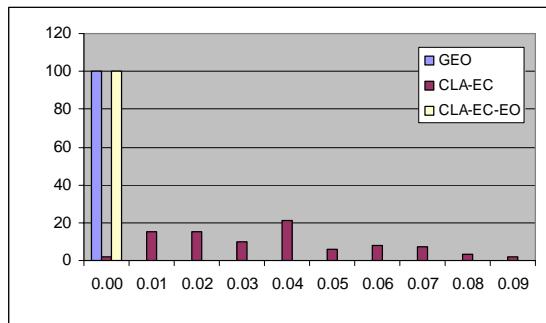
شکل (۷): مقایسه الگوریتم‌ها برای تابع Bumps
 (الف) ۱۰۰۰ تکرار، (ب) ۱۰۰۰۰ تکرار و (ج) ۳۰۰۰۰ تکرار



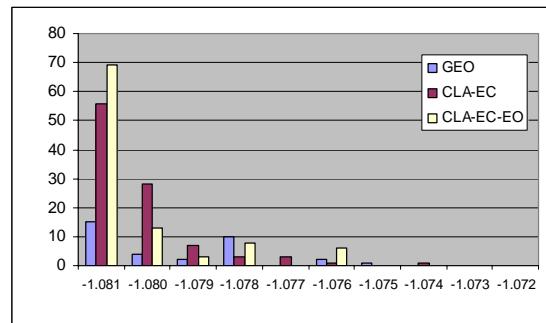
(ألف)



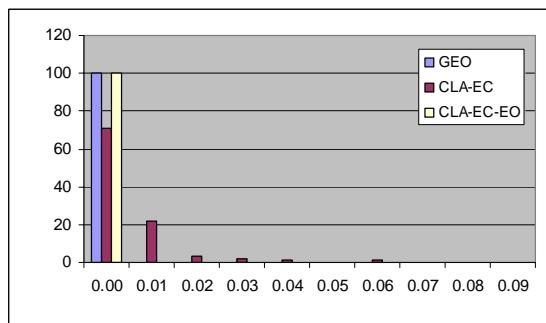
(ألف)



(ب)



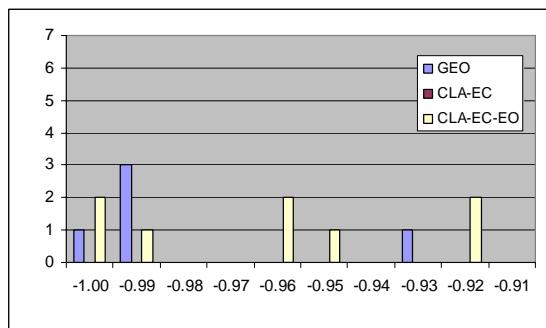
(ب)



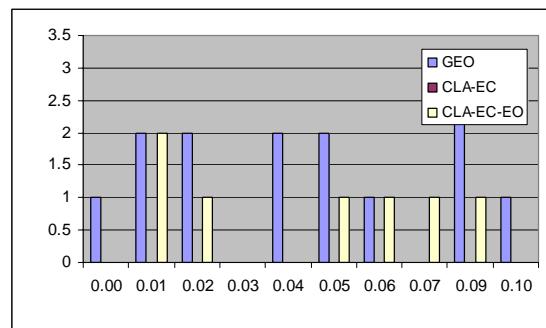
(ج)

شكل (١٠): مقایسه الگوریتم‌ها برای تابع CPF1
 (الف) ١٠٠٠ تکرار، (ب) ١٠٠٠٠ تکرار و (ج) ٣٠٠٠٠ تکرار

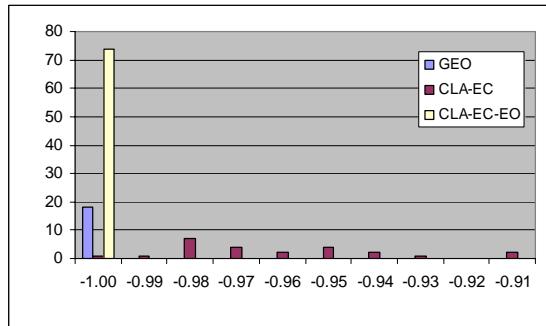
شکل (٩): مقایسه الگوریتم‌ها برای تابع Langermann
 (الف) ١٠٠٠ تکرار، (ب) ١٠٠٠٠ تکرار و (ج) ٣٠٠٠٠ تکرار



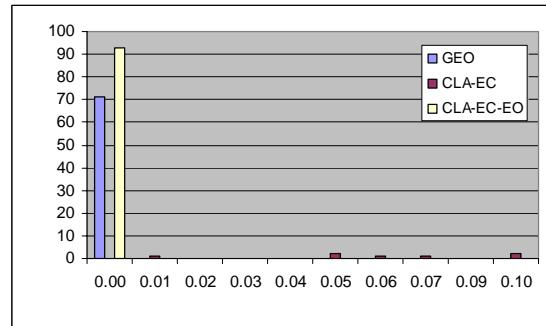
(الف)



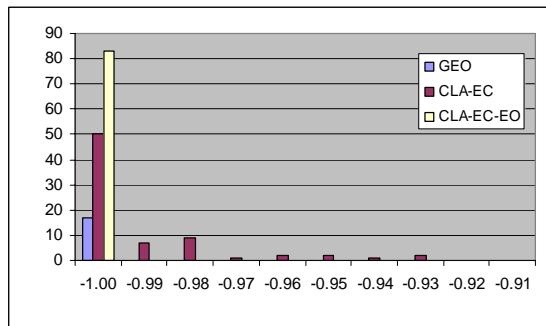
(الف)



(ب)



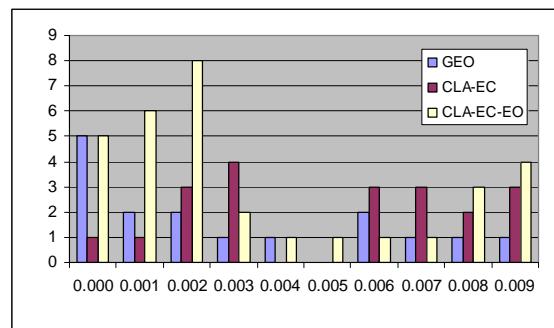
(ب)



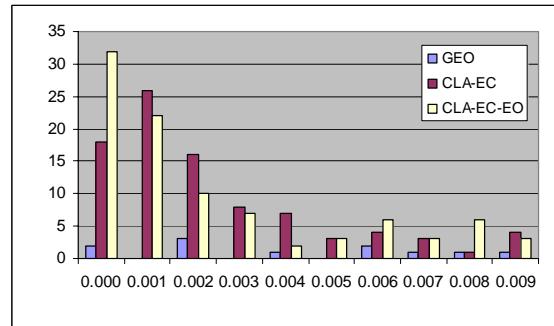
(ج)

شکل (12): مقایسه الگوریتم‌ها برای تابع Easom (الف) 1000 تکرار، (ب) 10000 تکرار و (ج) 30000 تکرار

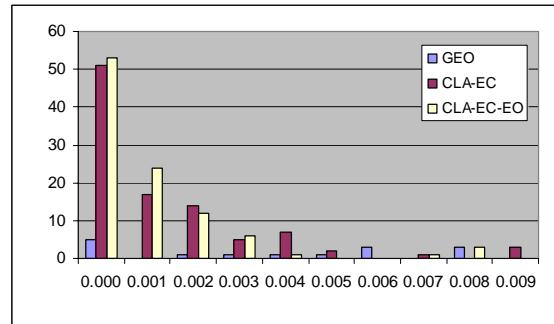
شکل (11): مقایسه الگوریتم‌ها برای تابع Ackley's Path (الف) 1000 تکرار، (ب) 10000 تکرار و (ج) 30000 تکرار



(ألف)



(ب)



(ج)

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