

## CLA-EC ناهمگام

بهرنگ مسعودی<sup>۱</sup> محمد رضا میبدی<sup>۱،۲</sup> رضا رستگار<sup>۱</sup>

<sup>۱</sup> آزمایشگاه محاسبات نرم، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران

و

<sup>۲</sup> مرکز تحقیقات فیزیک نظری، پژوهشکده علوم کامپیوتر، تهران، ایران

(masoodifar, meybodi, rastegar)@ ce.aut.ac.ir

### ۱- مقدمه

نیاز به جستجو در حل مسائل کاربردی امری غیر قابل اجتناب و در عین حال دشوار است و به همین جهت تعداد زیادی الگوریتمهای جستجو با فلسفه ها و دامنه های استفاده متفاوت توسط محققین پیشنهاد شده است. این الگوریتمها به دو دسته کلی جستجوهای کامل و جستجوهای مکاشفه ای تقسیم میشوند. تفاوت اساسی بین این دو دسته در این است که در جستجوهای کامل، تمام فضای جستجو به طور کامل مورد جستجو و ارزیابی قرار می گیرد تا جواب مورد نظر یافته شود، در حالیکه در جستجوهای مکاشفه ای تنها بخشی از فضا که احتمال یافتن جواب در آن بیشتر است، مورد جستجو قرار می گیرد. جستجوهای مکاشفه ای به دو دسته الگوریتمهای قطعی و غیرقطعی تقسیم میشوند. ویژگی اصلی الگوریتمهای قطعی روش تپه نوردی در این است که تحت شرایط یکسان، جوابهای یکسان می دهند. ایراد اساسی این الگوریتمها احتمال گیر افتادن در مینیمم های محلی می باشد. در مقابل، الگوریتمهای مکاشفه ای غیر قطعی از طریق جستجوهای تصادفی، در شرایط یکسان جوابهای متفاوتی تولید میکنند و در صورت گیر افتادن در مینیمم های محلی، از آنها می گریزند. الگوریتمهای مکاشفه ای غیر قطعی را بر اساس تعداد جوابهایی که در هر تکرار بررسی و ذخیره می کنند به دو دسته تقسیم می کنند. بعضی مانند تابکاری فلزات تنها یک جواب را در هر تکرار مورد بررسی قرار داده و ذخیره می کنند. بعضی دیگر در هر تکرار دسته ای از جوابها را ذخیره می کنند، که به این الگوریتمها، الگوریتمهای مبتنی بر جمعیت می گویند که الگوریتمهای تکاملی این جمله هستند.

در الگوریتمهای تکاملی<sup>۳</sup> از قوانین تکاملی موجود در طبیعت بمنظور بهینه سازی استفاده می شود. این الگوریتمها معمولا برای حل مسائل

**چکیده:** اتوماتای یادگیر سلولی<sup>۱</sup> (CLA) مدلی برای سیستمهایی است که از اجزاء ساده ای به نام سلول تشکیل شده اند و رفتار هر جزء بر اساس رفتار همسایگانش و نیز تجربیات گذشته اش تعیین و اصلاح می شود. اجزاء ساده تشکیل دهنده این مدل، از طریق تعامل با یکدیگر می توانند رفتار پیچیده ای از خود نشان دهند. CLA-EC که اخیرا پیشنهاد شده است یک الگوریتم تکاملی است که از ترکیب CLA و مفاهیم در محاسبات تکاملی (EC) بدست آمده است. CLA-EC مانند (CLA) به دو گروه همگام و ناهمگام تقسیم بندی میشود. در CLA-EC همگام، وضعیت داخلی همه سلولها بطور همزمان بروز میگردند و به همین دلیل وضعیت نهایی CLA-EC که در آن پایدار میشود بستگی به رفتاری که هر سلول از خود نشان میدهد دارد در حالیکه در CLA-EC ناهمگام وضعیت نهایی CLA-EC نه تنها بستگی به رفتار هر سلول دارد بلکه به ترتیب فعال سازی سلولها نیز وابسته میباشد. نشان داده شده است که CLA-EC همگام در حل مسایل بهینه سازی در مقایسه با CLA و یا الگوریتمهای ژنتیکی از کارایی بالاتری برخوردار میباشد. در این مقاله کارایی CLA-EC ناهمگام در حل مسایل بهینه سازی مورد مطالعه و بررسی قرار میگیرد و با نتایج حاصل از CLA-EC همگام مقایسه میشود. از طریق نتایج حاصل از بهینه سازی توابع استاندارد دوجونگ<sup>۲</sup> توسط CLA-EC همگام، CLA-EC ناهمگام و الگوریتمهای ژنتیکی نشان داده میشود که در بیشتر موارد CLA-EC ناهمگام از کارایی بالاتری برخوردار میباشد.

**کلمات کلیدی:** اتوماتاهای یادگیر سلولی، بهینه سازی، اتوماتاهای یادگیر سلولی ناهمگام، محاسبات تکاملی

<sup>1</sup> Cellular learning automata

<sup>2</sup> Dejong Functions

<sup>3</sup> Evolutionary Algorithm

کاهش یابد. اگر چه این مساله سرعت همگرایی را کاهش می دهد اما به دلیل قابلیت محاسبات موازی تاثیر این کاهش تا حد زیادی برطرف می شود. یکی از مدل‌هایی که با استفاده از آن فرایند محلی سازی عملگرهای ژنتیکی و ارتباط بین ژنومها صورت پذیرفته است، اتوماتای سلولی است [52][50][36][26]. اتوماتای سلولی علاوه بر داشتن ساختار ساده برای مدل کردن سیستمهای پیچیده، به دلیل ماهیت توزیع شدگی خود براحتی بر روی پردازنده های موازی قابل پیاده سازی است و همین امر سبب شده تا اتوماتای سلولی کاربرد خود را در این حوزه از محاسبات تکاملی پیدا کند.

اتوماتای یادگیر سلولی<sup>4</sup> (CLA) که از ترکیب اتوماهای یادگیر<sup>5</sup> (LA) و اتوماتای سلولی<sup>6</sup> (CA) حاصل شده است مدلی برای سیستم‌هایی است که از اجزاء ساده‌ای تشکیل شده‌اند و رفتار هر جزء بر اساس رفتار همسایگانش و نیز تجربیات گذشته‌اش تعیین و اصلاح می‌شود. هر اتوماتای یادگیر سلولی، از یک اتوماتای سلولی تشکیل شده است که هر سلول آن به یک یا چند اتوماتای یادگیر مجهز می‌باشد که وضعیت سلول را مشخص می‌سازند. همانند اتوماتای سلولی، قانون محلی در محیط حاکم است. در اتوماتای سلولی، قانون محلی مستقیماً وضعیت سلول را مشخص می‌کند ولی در اتوماتای یادگیر سلولی، این قانون تعیین می‌کند که آیا عمل انتخاب شده توسط یک اتوماتای یادگیر یک سلول بایستی پاداش داده شود و یا جریمه شود. عمل دادن پاداش و یا جریمه، منجر به بهنگام‌رسانی ساختار اتوماتای یادگیر سلولی به منظور نیل به یک هدف مشخص می‌گردد. پویایی اتوماتای یادگیر سلولی از به کارگیری مکرر قانون محلی بر روی اتوماتاهای یادگیر تمام سلول‌ها حاصل می‌شود. CLA به دو گروه همگام و ناهمگام تقسیم بندی میشود. در CLA همگام، همه سلولها بطور همزمان فعال میگردند و به همین دلیل وضعیت نهایی CLA که در آن پایدار میشود بستگی به رفتاری که هر سلول از خود نشان میدهد دارد. در حالیکه در CLA ناهمگام که در آن سلولها طبق یک ترتیب مشخص فعال میگردند، وضعیت نهایی CLA نه تنها بستگی به رفتار هر سلول دارد بلکه به ترتیب فعال کردن سلول-ها نیز وابسته میباشد. در CLA ناهمگام روشهای مختلفی برای به فعال کردن سلولها میتواند مورد استفاده قرار گیرد. چندین روش فعال کردن سلولها در CLA ناهمگام برای رسیدن به چنین ترتیب فعال سازی ایجاد شده‌اند. روش‌های فعال سازی سلولها در CLA ناهمگام به دو گروه تقسیم می‌شوند روش‌های مبتنی بر زمان<sup>7</sup> و روش‌های مبتنی بر گام<sup>8</sup>. در روش‌های مبتنی بر گام یک ترتیب از پیش تعریف شده، ترتیب

بهینه سازی پارامتری<sup>1</sup> که سایر روشهای رسمی از حل آنها عاجزند مورد استفاده قرار می گیرند. دسته بزرگی از این الگوریتمها را الگوریتمهای ژنتیکی تشکیل می دهد که در حل مسائل پیچیده با فضای جستجوی غیر هموار مورد استفاده قرار می گیرند. این الگوریتمها بر پایه انتخاب و ترکیب مجدد<sup>2</sup> بر روی جوابهای مجاز مساله کار می کنند. مجموعه جوابهای تولید شده در هر دوره الگوریتم را یک نسل و هر کدامها از جوابها را یک ژنوم یا کروموزوم می نامند. هر ژنوم ترکیبی از متغیرهای مساله است. در بعضی از مسائل متغیرهای مساله به یکدیگر وابسته و در بعضی دیگر از هم مستقل هستند. ولی همواره ارتباط و تبادل اطلاعات بین ژنومها از طریق انتخاب و ترکیب مجدد ژنومها در یک نسل صورت می پذیرد. این جابجایی اطلاعات سبب می شود تا جوابهای جزئی<sup>3</sup> با یکدیگر ترکیب و احتمالاً جوابهایی با کیفیت بالاتر بدست آیند. اما با تمام ویژگیهای مثبتی که در الگوریتم ژنتیکی استاندارد وجود دارد، این الگوریتم تنها در مواقعی که متغیرها از هم مستقل و یا در فاصله کمی از هم در ژنوم قرار گرفته باشند، کارایی مناسبی دارد [49]. به عبارت دیگر رفتار الگوریتم ژنتیکی وابستگی شدیدی به پارامترهایی از جمله نحوه تعریف عملگرهای جهش و تولید نسل، احتمال جهش و تولید نسل، اندازه جمعیتها و تعداد نسلهای تولید شده دارد. به همین دلیل گاهی در ترکیب ژنومها با یکدیگر نه تنها بهبودی در کیفیت جوابها حاصل نمی شود، بلکه الگوریتم در نقاط بهینه محلی به دام می افتد. به منظور رفع این مشکل نسخه های متعددی از الگوریتمهای ژنتیکی بوجود آمده اند، که می توانیم آنها را به سه دسته تقسیم کنیم. دسته اول الگوریتمهایی هستند که در آنها عملگرهای ژنتیکی همزمان با فرایند تکامل به صورت خودکار بهبود می یابند [51]. در دسته دوم به طور مشابه نحوه بازنمایی مساله همزمان با تکامل متحول می شود [55]. دسته سوم الگوریتمها نیز با ساخت مدل‌های احتمالاتی از متغیرها سعی در افزایش سرعت همگرایی به سمت جواب مناسب میکنند. [51][37-40][27-33][44-49].

ویژگی دیگر الگوریتم ژنتیکی استاندارد این است که هر ژنوم می تواند با هر ژنوم دیگری در جمعیت ژنومها ارتباط داشته و ساختار ارتباطی جمعیت به صورت یک گراف کامل می باشد. در سالهای اخیر تلاش محققان برای موازی سازی محاسبات در این حوزه سبب شده تا مساله ارتباطات محلی بین ژنومها به صورت جدی مطرح شود. پژوهشهای صورت گرفته نشان می دهد مساله محلی کردن ارتباط بین ژنومها علاوه بر ساده سازی محاسبات موازی می تواند سبب افزایش کیفیت جواب الگوریتم گردد [50]. یکی از دلایل این بهبود در انتقال تدریجی اثر یک ژنوم بر سایر ژنومها است که باعث می شود تا تعصب گرایی در الگوریتم

<sup>4</sup> Cellular Learning Automata

<sup>5</sup> Learning Automata

<sup>6</sup> Cellular Automata

<sup>7</sup> Time-driven

<sup>8</sup> Step-driven

<sup>1</sup> Parametric Optimization Problems

<sup>2</sup> Recombination

<sup>3</sup> Partial Solution

فعال سازی سلول‌ها را تعیین می‌کند و زمان فعال شدن صریحاً تعریف نشده است. در روش‌های مبتنی بر زمان، الگوریتم یک زمان مشخص به هر سلول منسوب می‌کند که زمان فعال شدن آن سلول می‌باشد.

یادگیری به عنوان یک پدیده تطبیقی در طول مدت بقای یک موجودیت زنده و تکامل یک فرایند بلند مدت در طی چندین نسل می‌باشد. پاسخ به این سوال که یادگیری و تکامل تا چه حد بر یکدیگر اثر گذارند، هنوز به عنوان یک مساله باز تحقیقاتی می‌باشد. تئوریهای متفاوتی در این باره ارائه شده اند که دو مورد از مهمترین آنها تئوری لامارکین و بالدوین می‌باشد. بر اساس تئوری لامارکین، محیط در طول زندگی جاندار بر آن تاثیر می‌گذارد و این تاثیرات بروی ساختار ژنتیکی جاندار منعکس شده و به فرزندان منتقل می‌شود. در مقابل از دیدگاه تئوری بالدوین یادگیری با تشویق موفقیت‌هایی که در فرایند تکامل به صورت جزئی صورت می‌پذیرند، سبب هدایت و تسریع تکامل می‌گردد [49]. اگرچه نادرستی بعضی از این تئوریها در حال حاضر به طور کامل از سوی محققین علوم طبیعی مورد قبول می‌باشد، اما منشا کارهایی در زمینه محاسبات ملهم از طبیعت می‌باشد [53][34]. یکی از مدلهایی که از ترکیب مفاهیم تکامل و یادگیری بدست آمده است مدل CLA-EC می‌باشد که ترکیبی از مدل CLA و محاسبات تکاملی (EC) می‌باشد [56]. در مدل CLA-EC هر ژنوم به یکی از سلول‌های اتوماتای یادگیر سلولی اختصاص می‌یابد. مجموعه اعمال انتخاب شده توسط اتوماتاهای هر سلول، به رشته ژنومی آن سلول اختصاص می‌یابد. بر اساس قانون محلی، یک بردار سیگنال تقویتی ایجاد می‌گردد و به مجموعه اتوماتاهای یادگیر مستقر در آن سلول داده می‌شود. هر یک از اتوماتاهای یادگیر، بر اساس سیگنال دریافت شده و الگوریتم یادگیری، ساختار داخلی خود را به‌هنگام می‌کنند. فرایند انتخاب عمل‌ها و به‌هنگام درآوردن ساختار داخلی، تا زمانی که یک معیار از پیش تعیین شده ارضاء نشود تکرار می‌گردد. این مدل می‌تواند جهت حل مسائل بهینه‌سازی استفاده گردد. مدل CLA-EC مانند مدل CLA به دو گروه همگام و ناهمگام تقسیم بندی می‌شود. در CLA-EC همگام، وضعیت داخلی همه سلولها بطور همزمان بروز می‌گردند و به همین دلیل وضعیت نهایی CLA-EC که در آن پایدار می‌شود بستگی به رفتاری که هر سلول از خود نشان می‌دهد دارد در حالیکه در CLA-EC ناهمگام وضعیت نهایی CLA-EC نه تنها بستگی به رفتار هر سلول دارد بلکه به ترتیب فعال‌سازی سلول‌ها نیز وابسته می‌باشد. نشان داده شده است که CLA-EC همگام در حل مسایل بهینه‌سازی در مقایسه با CLA و یا الگوریتمهای ژنتیکی از کارایی بالاتری برخوردار می‌باشد. در این مقاله برای اولین بار CLA-EC ناهمگام معرفی و کارایی آن در حل مسایل بهینه‌سازی مورد مطالعه و بررسی قرار می‌گیرد و سپس با نتایج حاصل از CLA-EC همگام مقایسه می‌شود. از طریق نتایج حاصل از بهینه‌سازی توابع

استاندارد دوجونگ<sup>۱</sup> توسط CLA-EC همگام، CLA-EC ناهمگام و الگوریتمهای ژنتیکی نشان داده می‌شود که در بیشتر موارد CLA-EC ناهمگام از کارایی بالاتری برخوردار می‌باشد.

ادامه مقاله بدینصورت سازماندهی شده است. در بخشهای ۲، ۳ و ۴ به ترتیب اتوماتای سلولی، اتوماتای یادگیر و اتوماتای یادگیر سلولی مختصراً شرح داده می‌شود. در بخش ۵ یک الگوریتم بهینه‌سازی مبتنی بر CLA-EC همگام و در بخش ۶ CLA-EC ناهمگام انواع روش‌های به‌هنگام‌رسانی ناهمگام شرح داده می‌شود. نتایج آزمایشها در بخش ۷ آمده است. بخش‌نهایی مقاله نتیجه‌گیری می‌باشد

## ۲- اتوماتای سلولی

اتوماتای سلولی (CA) یک مدل ریاضی برای سیستم‌هایی است که در آنها چندین مؤلفه ساده برای تولید الگوهای پیچیده با هم همکاری می‌کنند. اتوماتاهای سلولی در حقیقت سیستم‌های دینامیکی گسسته‌ای هستند که رفتارشان کاملاً بر اساس ارتباط محلی استوار است. در اتوماتای سلولی یک مجموعه منظم از سلول‌ها وجود دارد که هر کدام می‌توانند با چند وضعیت مختلف مقداردهی شوند. با توجه به تعداد مقادیری که سلولها می‌توانند اختیار کنند، اتوماتای سلولی به دو نوع دودویی و چندمقداره تقسیم می‌شود. برای هر سلول یک همسایگی از سلولها در نظر گرفته می‌شود. معمولاً همسایگی استفاده شده از نوع همسایگی نزدیک می‌باشد. این سلول‌ها در زمانهای گسسته بر طبق یک قانون محلی به‌هنگام‌رسانی می‌شوند. در تعیین وضعیت جدید برای هر سلول، مقادیر سلول‌های همسایه نیز تأثیرگذار هستند. انتخاب قوانین مختلف برای به‌هنگام‌رسانی، انواع متفاوتی از اتوماتای سلولی را بوجود آورده است. این قوانین می‌توانند به صورت قطعی و یا احتمالی بیان شوند. برای اطلاعات بیشتر در باره CA و کاربردهای آن میتوان به [54] مراجعه نمود

## ۳- اتوماتای یادگیر

اتوماتای یادگیر یک ماشین با حالات محدود<sup>۲</sup> است که میتواند تعداد محدودی عمل را انجام دهد. هر عمل انتخاب شده توسط محیطی احتمالی ارزیابی می‌گردد و پاسخی به اتوماتای یادگیر داده می‌شود. اتوماتای یادگیر از این پاسخ استفاده نموده و عمل بعدی خود را انتخاب می‌کند. در طی این فرایند، اتوماتای یادگیر یاد می‌گیرد که چگونه بهترین عمل را از بی‌ن اعمال مجاز خود انتخاب نماید. ارتباط بین اتوماتای یادگیر و محیط را نشان می‌دهد. محیط را می‌توان توسط سه تایی  $E \equiv \{\alpha, \beta, c\}$  نشان داد که در آن  $\alpha = \{\alpha_1, \dots, \alpha_r\}$  مجموعه ورودیها،  $\beta = \{\beta_1, \dots, \beta_m\}$  مجموعه خروجیها و  $c = \{c_1, \dots, c_r\}$

<sup>1</sup> Dejong Functions

<sup>2</sup> Finite State Machine

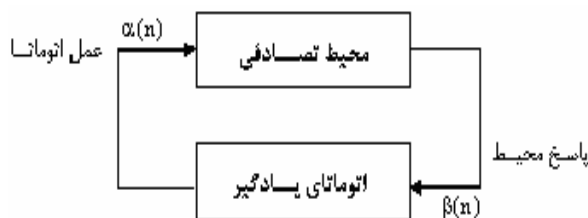
در روابط فوق،  $a$  پارامتر پاداش و  $b$  پارامتر جریمه است. با توجه به مقادیر  $a$  و  $b$  سه حالت را می‌توان در نظر گرفت. زمانیکه  $a$  و  $b$  با هم برابر باشند، الگوریتم را  $L_{RP}$  می‌نامیم. زمانیکه  $b$  از  $a$  خیلی کوچکتر باشد، الگوریتم را  $L_{Rep}$  می‌نامیم. زمانیکه  $b$  مساوی صفر باشد، الگوریتم را  $L_{RI}$  می‌نامیم. برای اطلاعات بیشتر در باره اتوماتاهای یادگیر و کاربردهای آن میتوان به [21][3][35][41-43] مراجعه نمود.

#### ۴- اتوماتای یادگیر سلولی

اتوماتای یادگیر سلولی،  $CLA$ ، مدلی برای سیستم‌هایی است که از اجزاء ساده‌ای تشکیل شده‌اند و رفتار هر جزء بر اساس رفتار همسایگانش و نیز تجربیات گذشته‌اش تعیین و اصلاح می‌شود. اجزاء ساده تشکیل دهنده این مدل، از طریق تعامل با یکدیگر می‌توانند رفتار پیچیده‌ای از خود نشان دهند. هر اتوماتای یادگیر سلولی، از یک اتوماتای سلولی تشکیل شده است که هر سلول آن به یک یا بیشتر اتوماتای یادگیر مجهز می‌باشد که وضعیت این سلول را مشخص می‌سازند. همانند اتوماتای سلولی، قانون محلی در محیط حاکم است و این قانون تعیین می‌کند که آیا عمل انتخاب شده توسط یک اتوماتا در یک سلول بایستی پاداش داده شود و یا جریمه شود. عمل دادن پاداش و یا جریمه منجر به بروز درآوردن ساختار اتوماتای یادگیر سلولی به منظور نیل به یک هدف مشخص می‌گردد. در اتوماتای یادگیر سلولی می‌توان از ساختارهای مختلفی برای همسایگی استفاده نمود. در حالت کلی هر چندتایی مرتب از سلولها را می‌توان به عنوان همسایه در نظر گرفت.

عملکرد اتوماتای یادگیر سلولی به این صورت است که در هر لحظه هر اتوماتای یادگیر در اتوماتای یادگیر سلولی یک عمل از مجموعه اعمال خود را انتخاب می‌کند. عمل انتخاب شده با توجه به اعمال انتخاب شده توسط سلولهای همسایه و قانون حاکم بر اتوماتای یادگیر سلولی پاداش داده و یا جریمه می‌شود. با توجه به اینکه عمل انتخاب شده پاداش گرفته و یا جریمه شده است، اتوماتای یادگیر رفتار خود را تصحیح کرده و از این طریق ساختار داخلی خود را بروز می‌کند. بعد از بروز درآوردن ساختار داخلی، هر اتوماتای یادگیر در اتوماتای یادگیر سلولی دوباره یک عمل از مجموعه اعمال خود را انتخاب کرده و انجام می‌دهد. فرآیند انتخاب عمل و دادن پاداش و یا جریمه تا زمانیکه سیستم به وضعیت پایدار برسد و یا یک معیار از قبل تعریف شده‌ای برقرار شود، ادامه می‌یابد. عمل بهنگام‌سازی ساختار اتوماتاهای یادگیر موجود در اتوماتای یادگیر سلولی توسط الگوریتم یادگیری انجام می‌شود. برای اطلاعات بیشتر در باره اتوماتاهای یادگیر سلولی و کاربردهای آن میتوان

مجموعه احتمالهای جریمه می‌باشد. هرگاه  $\beta$  مجموعه دو عضوی باشد محیط از نوع  $P$  می‌باشد. در چنین محیطی  $\beta_1 = 1$  به عنوان جریمه و  $\beta_2 = 0$  به عنوان پاداش در نظر گرفته می‌شود. در محیط  $Q, \beta(n)$  می‌تواند به طور گسسته یک مقدار از مقادیر محدود در فاصله  $[0,1]$  و در محیط از نوع  $S$ ،  $\beta(n)$  هر مقدار در فاصله  $[0,1]$  را اختیار کند.  $C_i$  احتمال اینکه عمل  $\alpha_i$  نتیجه نامطلوب داشته باشد، می‌باشد. در محیط ایستا مقادیر  $C_i$  در طی زمان بدون تغییر می‌مانند، حال آنکه در محیط غیر ایستا این مقادیر در طی زمان تغییر می‌کنند. اتوماتای یادگیر به دو گروه اتوماتاهای یادگیر با ساختار ثابت و اتوماتاهای یادگیر با ساختار متغیر تقسیم می‌شوند. در این مقاله از اتوماتاهای یادگیر با ساختار متغیر استفاده شده است که در ادامه این بخش توضیح داده میشود.



شکل ۱: ارتباط بین اتوماتای یادگیر و محیط

اتوماتاهای یادگیر با ساختار متغیر<sup>۱</sup>: اتوماتای یادگیر با ساختار متغیر توسط ۴ تایی  $\{\alpha, \beta, p, T\}$  نشان داده می‌شود که در آن  $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  مجموعه عملهای اتوماتا،  $\beta = \{\beta_1, \beta_2, \dots, \beta_m\}$  مجموعه ورودی‌های اتوماتا،  $p = \{p_1, p_2, \dots, p_m\}$  بردار احتمال انتخاب هریک از عمل‌ها و  $p(n+1) = T[\alpha(n), \beta(n), p(n)]$  الگوریتم یادگیری می‌باشد. در این نوع از اتوماتاهای یادگیر، اگر عمل  $\alpha_i$  در مرحله  $n$ م انجام شود و پاسخ مطلوب از محیط دریافت نماید، احتمال  $p_i(n)$  کاهش یافته و سایر احتمالات افزایش می‌یابند. در هر حال، تغییرات به گونه‌ای صورت می‌گیرد تا حاصل جمع  $p_i(n)$ ها همواره ثابت و مساوی یک باقی بماند. الگوریتم زیر یک نمونه از الگوریتم های یادگیر خطی است.

الف- پاسخ مطلوب

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)]$$

$$p_j(n+1) = (1 - a)p_j(n) \quad \forall j \quad j \neq i$$

ب- پاسخ

نامطلوب

$$p_i(n+1) = (1 - b)p_i(n)$$

$$p_j(n+1) = \frac{b}{r-1} + (1 - b)p_j(n) \quad \forall j \quad j \neq i$$

<sup>2</sup> Linear reward penalty

<sup>3</sup> Linear reward epsilon penalty

<sup>4</sup> Linear reward inaction

<sup>5</sup> Cellular learning automata

<sup>1</sup> Variable structure

تقویتی را ساخته و به اتوماتاهای یادگیر خود می دهد(این مکانیسم در معادل جفت گیری ژنوم مورد نظر می باشد).

### 5- CLA-EC همگام

به منظور ساده کردن ارائه عملکرد CLA-EC به هنگام فرض می کنیم فضای جستجوی مورد نظر یک فضای متناهی دودویی باشد. بنابراین یک مساله بهینه سازی میتواند به صورت زیر طرح گردد.

$$\max \{ f(\underline{X}) \mid \underline{X} \in B^n \}$$

به طوریکه  $f(\cdot)$  یک تابع حقیقی و  $B^n = \{0,1\}^n$  فضای جستجوی دودویی مورد نظر است. هر ژنوم در این الگوریتم دارای دو مولفه رشته ژنومی و مدل ژنومی می باشد. رشته ژنومی همان راه حل های میانی مساله مورد نظر می باشند. مدل ژنومی متشکل از تعدادی اتوماتای یادگیر می باشد که بر اساس تجارب گذشته خود و ژنومهای دیگر آموزش می بینند. و بدین ترتیب فرایند تکامل به طریقی هدایت می شود تا ارزش رشته ژنومی بر اساس تابع ارزیابی بهبود یابد. نحوه انتساب اتوماتاهای یادگیر مدل ژنومی به بیتهای رشته ژنومی به گونه های مختلفی قابل طرح است. برای مثال می توانیم رشته ژنومی را به دسته های 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 ژنومی به طول  $n$  (با فرض زوج بودن)، تعداد  $n/2$  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 ۱ را می پذیرند. در این مقاله با فرض مستقل بودن متغیرهای مساله ساده ترین شکل مدل ژنومی که در آن برای هر بیت در رشته ژنومی یک اتوماتای یادگیر در مدل ژنومی خواهیم داشت در نظر می گیریم. هر اتوماتای یادگیر مقدار بیت متناظر با خود را تعیین میکند.

فرض کنید  $X_t^i \in D_f$  وضعیت سلول  $i$  در گام  $t$  باشد. برای تولید وضعیت سلول  $i$  در گام  $t+1$ ،  $X_{t+1}^i$ ، هر کدام از اتوماتاهای یادگیر مقدار بیت متناظر خود در ژنوم را اعلام می کنند. در صورتی که ژنوم جدید،  $new_{t+1}^i$ ، ارزش بیشتری نسبت به ژنوم قبلی داشته باشد، در سلول ژنوم جدید جایگزین ژنوم قبلی می شود، در غیر این صورت سلول رشته ژنومی قبلی را حفظ می کند.

$$X_{t+1}^i = \begin{cases} X_t^i & f(X_t^i) > f(new_{t+1}^i) \\ new_{t+1}^i & f(X_t^i) \leq f(new_{t+1}^i) \end{cases}$$

این بخش از الگوریتم معادل یادگیری از تجارب قبلی ژنوم و حفظ آنها در نسلهای بعدی(ژنومهای بعدی) و یا به بیان دیگر معادل تاثیر یادگیری بر تکامل از دیدگاه تئوری بالدوین می باشد. جدول ۱ مقایسه ای را بین مفاهیم در CLA-EC، اتوماتای یادگیر سلولی و الگوریتم ژنتیکی نشان میدهد.

جدول ۱: مقایسه ای بین CLA-EC، اتوماتای یادگیر سلولی و الگوریتم ژنتیکی

اتوماتای یادگیر سلولی	CLA-EC	الگوریتم ژنتیکی
سلول	سلول	فرد جمعیت
شبکه سلولها	شبکه سلولها	جمعیت یک نسل
حالت سلول	حالت سلول	ژنوم ( کروموزوم)
قوانین	عملگر ژنتیکی	عملگرهای ژنتیکی
همسایگی	همسایگی	-
اتوماتاهای یادگیر	-	-

قانون تعریف شده برای CLA-EC از دو بخش تشکیل شده است. بخش اول استراتژی انتخاب می باشد. منظور از استراتژی انتخاب سلول  $i$ ام، نحوه انتخاب تعداد مشخص  $Se$ ،  $1 \leq Se \leq m$ ، رشته ژنومی از میان  $m$  ژنوم همسایه،  $P^i$ ، می باشد. در استراتژی مورد استفاده،  $Se$  ژنوم از بهترین ژنومهای  $P^i$  را انتخاب می کنیم و آن را  $P_{Se}^i$  می نامیم. بخش دوم قانون استراتژی تولید سیگنال تقویتی با استفاده از مجموعه  $P_{Se}^i$  می باشد. بردار سیگنال تقویتی به روشهای گوناگون قابل محاسبه است. در این بخش به شرح روش مورد استفاده در این پایان نامه می پردازیم. فرض کنید  $\bar{\beta}_t^i = (\beta_t^{i,1}, \dots, \beta_t^{i,n})$  بردار مورد نظر باشد که  $\bar{\beta}_t^i \in \{0,1\}^n$  به عنوان جریمه و  $0$  به عنوان پاداش در نظر گرفته می شود. اگر  $X_t^i = (X_t^{i,1}, \dots, X_t^{i,n})$  رشته ژنومی سلول  $i$ ام در

حال برای ادامه بحث یک اتوماتای یادگیر سلولی،  $(CLA(L_1, \dots, L_k))$ ، با  $k$  سلول که هر سلول مجهز به  $n$  اتوماتای یادگیر است در نظر بگیرد. حالت هر سلول توسط یک رشته ژنومی به طول  $n$  نشان داده میشود و نتیجتاً هر سلول میتواند  $2^n$  حالت مختلف اختیار کند. با فرض همگام بودن اتوماتای یادگیر سلولی، در زمان  $t$  هر سلول،  $i$ ، رشته های ژنومی خود و همسایگان خود را مورد بررسی قرار داده و بر اساس تابع ارزیابی از میان آنها تعدادی را به عنوان ژنومهای مناسب انتخاب می کند(معادل انتخاب زوج در طبیعت). این شیوه انتخاب دو طرفه نمی باشد. به عبارت دیگر اگر یک ژنوم، ژنوم دیگری را به عنوان یکی از کاندیدهای خود انتخاب نماید، تضمینی برای انتخاب شدن این ژنوم از طرف ژنوم مقابل وجود نخواهد داشت(البته این روش معادلی در بین سیستمهای طبیعی ندارد). ژنوم بر اساس ژنومهایی که انتخاب نموده است، یک بردار سیگنال

زمان  $t$  باشد. برای هر متغیر  $X_t^{i,j}$ ،  $1 \leq j \leq n$  و به ازای مقادیر  $N_{i,j}(k)$ ،  $k = 0, 1$  به صورت زیر محاسبه می شود.

$$N_{i,j}(k) = \sum_l \delta_l(X_t^{i,j} = k | P_{Se}^i)$$

به طوریکه اگر متغیر  $X_t^{i,j}$  در  $i$  امین سلول، مقدار  $k$  را داشته باشد  $\delta_l(X_t^{i,j} = k | P_{Se}^i) = 1$  و در غیر این صورت برابر صفر است. سپس با استفاده از  $N_{i,j}(k)$  مقدار  $\beta_t^{i,j}$  را بدست می آوریم.

$$\beta_t^{i,j} = u(N_{i,j}(1) - N_{i,j}(0)) \quad \text{If } X_t^{i,j} = 0$$

$$\beta_t^{i,j} = u(N_{i,j}(0) - N_{i,j}(1)) \quad \text{If } X_t^{i,j} = 1$$

که  $u(\cdot)$  تابع پله است. الگوریتم الگوریتم تکاملی مبتنی بر اتوماتای یادگیر سلولی (CLA-EC) در شکل ۲ نشان داده شده است. نکته ای که باید به آن توجه داشته باشیم این است که نحوه انتخاب ژنومها و تولید بردار تقویتی که قانون اتوماتای یادگیر سلولی را نشان می دهد، از پارامترهای بسیار مهم الگوریتم می باشد.

Initialize.

While not done do

For each cell  $i$  in CLA do in parallel

Generate a new string genome

Evaluate the new string genome

If  $f(\text{new string genome}) > f(\text{old string genome})$  then

Accept the new string genome

End if

Select  $Se$  cells from neighbors of cell  $i$

Generate the reinforcement signal vector

Update LAs of cell  $i$

End parallel for

End while

شکل ۲: الگوریتم تکاملی مبتنی بر اتوماتای یادگیر سلولی (CLA-EC)

## ۶- CLA-EC ناهمگام و روشهای مختلف فعال کردن

### سلولها

در CLA-EC همگام، تمام سلولها بطور همزمان فعال می شوند و هر سلول با مقادیر سلولهای همسایه در تکرار قبلی سروکار دارد و وضعیت نهایی سلولها (ژنومها) فقط به اتوماتاهای یادگیر بستگی دارند. با استفاده از بروز رسانی ناهمگام، وضعیت نهایی، به ترتیب فعال سازی سلولها نیز وابسته خواهد بود. همچنین در روشهای بروز رسانی ناهمگام، بر خلاف روش بروز رسانی همگام، هر سلول با مقادیر سلولهای همسایه در تکرار فعلی سروکار دارد.

ادامه این بخش به شرح روشهای مختلف فعال سازی سلولها که در آزمایشها استفاده شده است می پردازد. روشهای فعال سازی به دو گروه روشهای

مبتنی بر زمان<sup>۱</sup> و روشهای مبتنی بر گام<sup>۲</sup> تقسیم بندی میشود. در روشهای مبتنی بر گام، فعال شدن یک سلول تکی "تک گام" نامیده می شود و یک سری از تک گامها یک "تکرار" را تشکیل میدهد. در روشهای مبتنی بر گام ترتیب فعال سازی با استفاده از یکی از روشهای زیر و یا هر روش دیگری تعیین میشود. در صورتیکه در روشهای مبتنی بر زمان، زمان فعال سازی بطور صریح مشخص میشود. در ادامه به شرح روشهای فعال سازی مبتنی بر گام که در آزمایشها مورد استفاده قرار گرفته است می پردازیم.

**ترتیب ثابت**<sup>۳</sup>: سلولها بر اساس یک ترتیب ثابت و از پیش تعریف شده فعال میشوند.

**ترتیب تصادفی**<sup>۴</sup>: در این روش ترتیب فعال سازی به این صورت ساخته می شود که اولین سلول به طور تصادفی از میان  $n$  سلول انتخاب می شود. سپس دومین سلول به طور تصادفی از میان  $n-1$  سلول باقی مانده انتخاب شده و به همین ترتیب ادامه میابد تا تمام سلولها انتخاب شوند. نکته ای که باید به آن توجه کرد این است که هنگام انتخاب، تمام سلولهای باقی مانده شناس یکسانی برای انتخاب شدن دارند. به عبارت دیگر، انتخاب بر اساس توزیع یکنواخت بدون جایگذاری است.

**ترتیب جدید تصادفی**<sup>۵</sup>: در این روش برای هر تکرار از یک ترتیب فعال سازی تصادفی بر مبنای توزیع یکنواخت با جایگذاری استفاده میشود. در این روش بر خلاف روش ترتیب تصادفی ترتیب فعال سازی در تمام تکرارها یکسان نمی باشد.

**ترتیب یکنواخت**<sup>۶</sup>: در این روش انتخاب سلول برای فعال سازی به طور تصادفی با توزیع یکنواخت انجام میگیرد. به عبارت دیگر از یک انتخاب تصادفی یکنواخت با جایگذاری استفاده می شود. در  $k$  تک گام (که  $k$  تعداد سلولها است) ممکن است بعضی از سلولها بیش از یک بار فعال شوند و بعضی دیگر اصلاً فعال نشوند. در این روش، برخلاف روشهای دیگر، برای انتخاب سلول بعدی به حافظه اضافی نیاز نمیشود. نیاز به حافظه اضافی در روشهای قبلی بدین دلیل است که احتمال انتخاب هر سلول برای فعال سازی به سلولهای انتخاب شده قبلی بستگی دارد.

## ۷- نتایج آزمایشها

برای آزمایشها از CLA-EC خطی با شعاع همسایگی ۱ اسنفاده شده است. اتوماتاهای یادگیر، از نوع LRP و با نرخ یادگیری  $a = b = 0.1$  استفاده گردیده است. برای بررسی تأثیر تعداد سلولها در رفتار

<sup>1</sup> time-driven

<sup>2</sup> Step driven

<sup>3</sup> Fixed directional

<sup>4</sup> Fixed random sweep

<sup>5</sup> Random new sweep

<sup>6</sup> Uniform choice

می‌یابد و سرعت آنها از الگوریتم ژنتیک استاندارد به مراتب بیشتر است. با بررسی شکل ۵ درمی‌یابیم که الگوریتم ژنتیک استاندارد و CLA-EC همگام سریع‌تر از CLA-EC ناهمگام همگرا می‌شوند ولی الگوریتم ژنتیک استاندارد به جواب بهینه دست نمی‌یابد و در بهینه‌های محلی به دام می‌افتد. شکل ۶ نشان می‌دهد که الگوریتم ژنتیک استاندارد نسبت به CLA-EC از سرعت کمتری برخوردار است. گرچه روش CLA-EC همگام سریعتر از روش‌های ناهمگام همگرا می‌شود ولی برای اینکار به تعداد بیشتری سلول نیاز دارد و در غیر این صورت در بهینه‌های محلی به دام می‌افتد. CLA-EC ناهمگام با تعداد سلول‌های کمتری به جواب بهینه دست می‌یابد. به منظور تأکید بر اختلاف روش‌های فعال سازی ناهمگام و همگام، مقایسه‌ی این روش‌ها به طور جداگانه در 6c آمده است. شکل‌های ۷ و ۸ نشان می‌دهند که رفتار روش‌های فعال سازی ناهمگام و روش همگام به هم شبیه است و الگوریتم ژنتیک استاندارد در بهینه‌های محلی به دام می‌افتد. نکته قابل توجه اینکه روش انتخاب یکنواخت علی‌رغم سادگی آن در پیاده‌سازی و عدم نیاز آن به حافظه اضافی برای انتخاب سلول بعدی، نتایج نسبتاً خوبی را در پی دارد.

#### ۸- نتیجه‌گیری

در این مقاله کارایی CLA-EC ناهمگام در حل مسایل بهینه سازی مورد مطالعه و بررسی قرار گرفت و با نتایج حاصل از CLA-EC همگام مقایسه شد. از طریق نتایج حاصل از بهینه سازی توابع استاندارد Dejong توسط CLA-EC همگام، CLA-EC ناهمگام و الگوریتم‌های ژنتیکی نشان داده شد که در بیشتر موارد CLA-EC ناهمگام از کارایی بالاتری برخوردار می‌باشد. نشان داده شد که در بین روش‌های به‌هنگام‌رسانی روش انتخاب یکنواخت علی‌رغم سادگی در پیاده‌سازی و عدم نیاز آن به حافظه اضافی برای انتخاب سلول بعدی، نتایج بهتری را تولید مینماید. نشان داده شد که در CLA-EC بخصوص CLA-EC ناهمگام چون اثرات سلول‌ها بر یکدیگر محلی است و به کندی فضای سلول‌ها را تحت الشعاع قرار دهد، برای فضای سلولی با اندازه مناسب، زمان کافی برای بررسی نواحی مختلف دامنه تابع وجود دارد، تعصب گرایی و در نتیجه توقف الگوریتم در بهینه‌های محلی کاهش می‌یابد. همچنین نشان داده شد که گرچه روش CLA-EC همگام سریعتر از روش‌های ناهمگام همگرا می‌شود ولی برای اینکار به تعداد بیشتری سلول نیاز دارد و در غیر این صورت در بهینه‌های محلی به دام می‌افتد.

الگوریتم، آزمایشها برای تعداد سلول‌های مختلف از ۳ تا ۴۹ انجام گرفته است. برای سهولت در ارائه از نماد Method-CLA-(a,b,r,se,q) برای اشاره به الگوریتم CLA-EC با q سلول، شعاع همسایگی r، تعداد سلول انتخابی se، اتوماتای یادگیر با پارامترهای پاداش a و جریمه b به روش Method (ترتیب فعال سازی) است. شکل ۳ معماری CLA-EC مورد استفاده در آزمایشها می‌باشد در این شکل  $X_i$  وضعیت سلول نام را مشخص می‌کند.  $Environment_{i,j}$  محیط اتوماتای نام سلول نام است. ورودی محیط شامل  $x_{i,j}$ ،  $x_{i+1,j}$  و  $x_{i-1,j}$  است که اعمال انتخاب شده توسط اتوماتاهای یادگیر متناظر در سلول‌های همسایه می‌باشند. این محیط، پاداش یا جریمه اتوماتای نام سلول نام را تعیین می‌کند. LAها بیت‌های متناظر در هر ژنوم را تعیین می‌کنند و comparator نتیجه بدست آمده جدید را با نتیجه قبلی مقایسه کرده و در صورت بهبود جواب، وضعیت سلول را تغییر می‌دهد. مسایل بهینه سازی، بهینه سازی توابع Dejong استفاده شده است. توابع Dejong گروهی از توابع ریاضی هستند که برای آزمایش الگوریتم‌های تکاملی طراحی شده و به عنوان معیاری برای مقایسه الگوریتم‌های مختلف مورد استفاده قرار می‌گیرد [2]. توابع Dejong به شرح زیر هستند.

$$F_1(X) = \sum_{i=1}^3 x_i^2 \quad -5.12 \leq x_i \leq 5.12$$

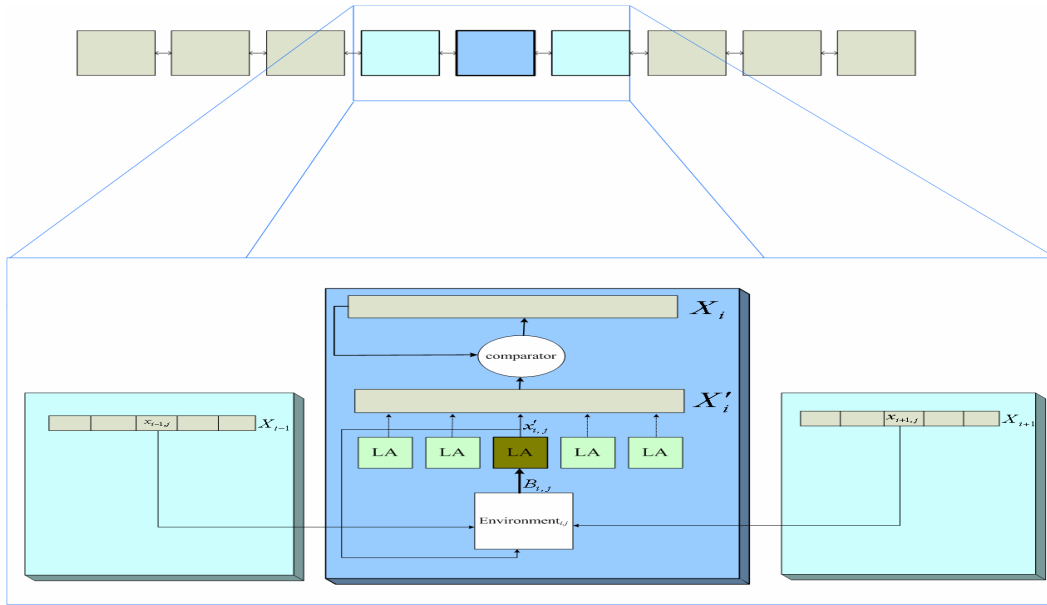
$$F_3(X) = \sum_{i=1}^5 \text{integer}(x_i) \quad -7.12 \leq x_i \leq 7.12$$

$$F_2(X) = 100(x_1^2 - x_2^2)^2 + (1 - x_1)^2 \quad -2.048 \leq x_i \leq 2.048$$

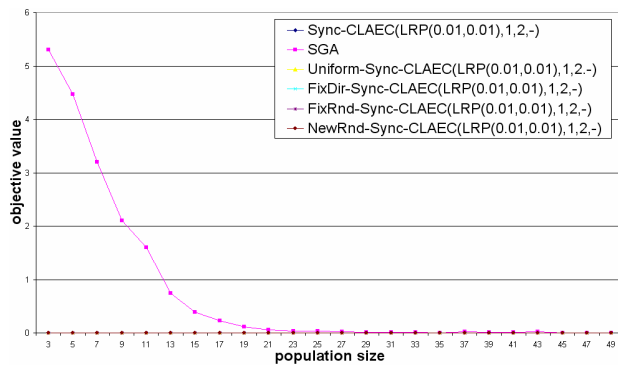
$$F_4(X) = \sum_{i=1}^{30} i x_i^4 + \text{Gaus}(0,1) \quad -1.28 \leq x_i \leq 1.28$$

$$F_5(X) = \left( 0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_j - a_{ij})^6} \right)^{-1} \quad -65.536 \leq x_i \leq 65.536$$

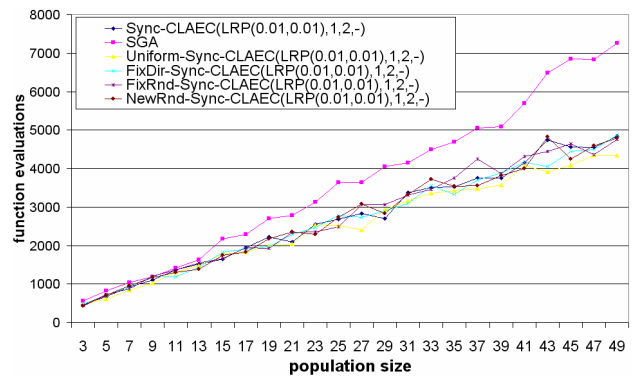
مقایسه نتایج بدست آمده از آزمایشها نشان می‌دهد که تمام روش‌های ناهمگام مورد آزمایش تقریباً عملکرد یکسانی دارند. شکل ۴ نشان می‌دهد که کارایی روش‌های ناهمگام به کارایی روش همگام نزدیک است ولی روش انتخاب یکنواخت سرعت بیشتری نسبت به روش همگام و روش‌های دیگر ناهمگام دارد. برخلاف الگوریتم ژنتیک CLA-EC ناهمگام با استفاده از تمام روش‌های فعال سازی به جواب بهینه دست



شکل ۴: ساختار CLA-EC شبیه‌سازی شده

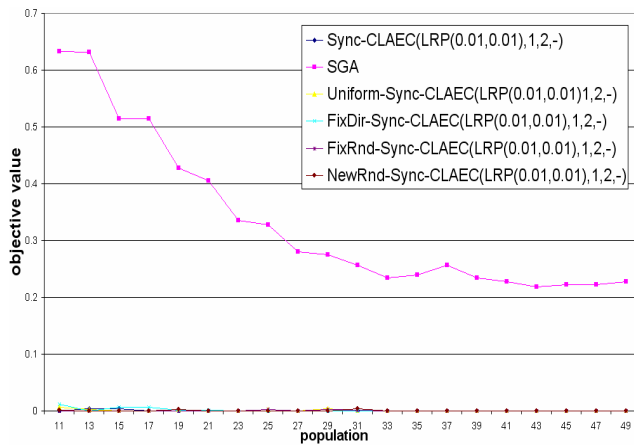


(b) ارزش جواب بدست آمده

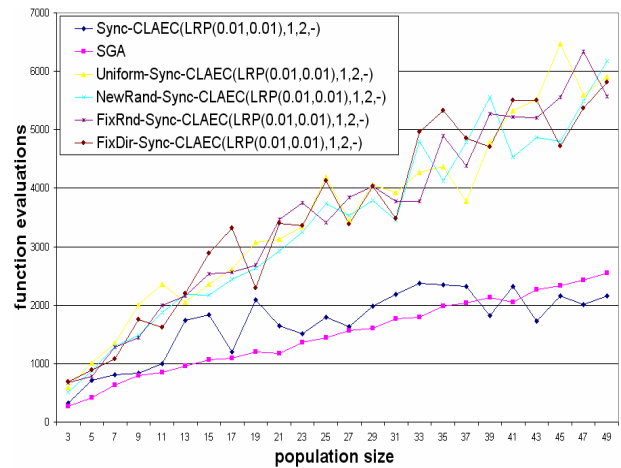


(a) تعداد ارزیابی‌ها

شکل ۴: مقایسهٔ SGAs, CLA-EC همگام و CLA-EC نا همگام با روشهای بروز رسانی مختلف برای تابع F1



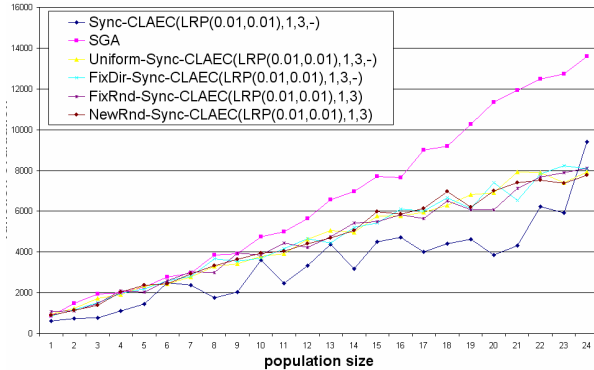
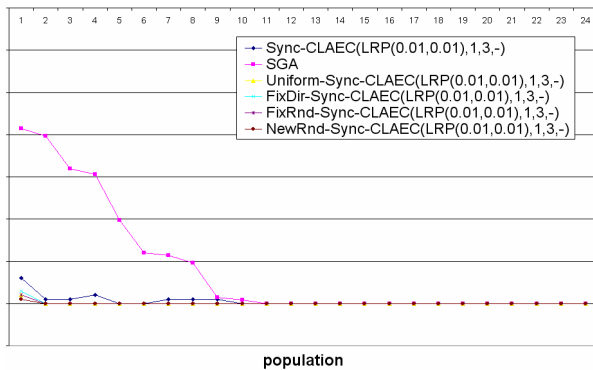
(b) ارزش جواب بدست آمده



(a) تعداد ارزیابی‌ها

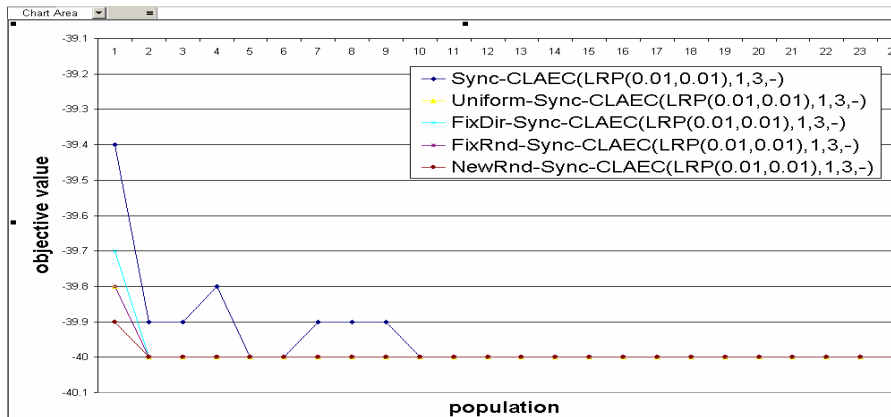
شکل 5: مقایسهٔ SGAs, CLA-EC همگام و CLA-EC نا همگام با روشهای بروز رسانی مختلف برای تابع F2





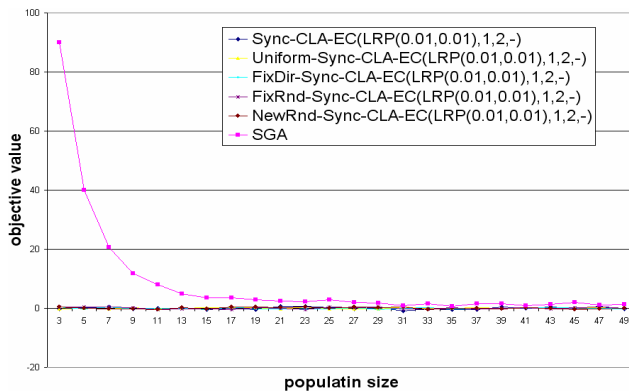
ارزش جواب بدست آمده (b)

تعداد ارزیابی‌ها (a)

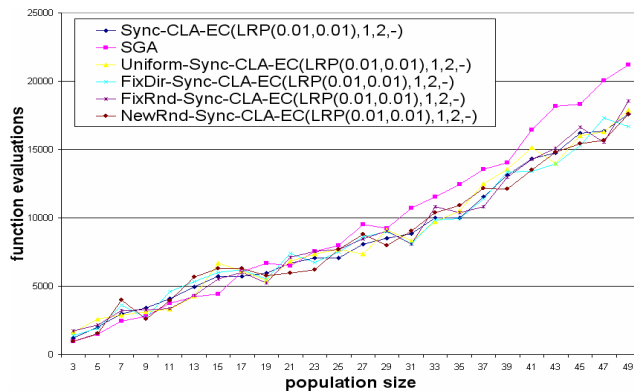


ارزش جواب بدست آمده (c)

شکل ۶: مقایسهٔ CLA-EC, SGA همگام و CLA-EC نا همگام با روشهای بروز رسانی مختلف برای تابع F3

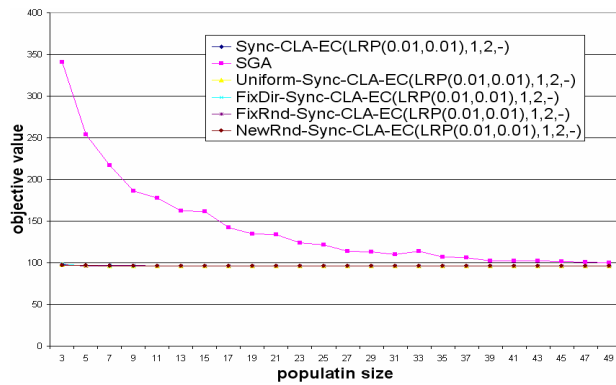


ارزش جواب بدست آمده (b)

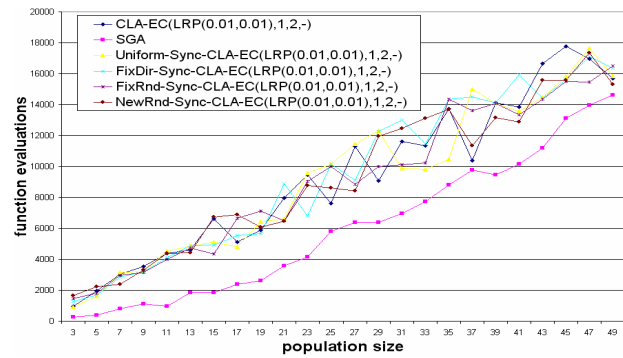


تعداد ارزیابی‌ها (a)

شکل ۷: مقایسهٔ CLA-EC, SGA همگام و CLA-EC نا همگام با روشهای بروز رسانی مختلف برای تابع F4



(b) ارزش جواب بدست آمده



(a) تعداد ارزیابی‌ها

شکل ۸: مقایسهٔ SGA, CLA-EC همگام و CLA-EC نا همگام با روشهای بروز رسانی مختلف برای تابع F5

## مراجع

- processing, control and communications, CRC Press, 1996.
- [11] Meybodi, M. R. and Kharazmi, M. R. "Application of Cellular Learning Automata to Image Processing", Journal of Amirkabir, Vol. 14, No. 56A, pp. 1101-1126, 2004.
- [12] Meybodi, M. R. and Khojaste, M. R. "Application of Cellular Learning Automata in Modeling of Commerce Networks", in Proceedings of 6th Annual International Computer Society of Iran Computer Conference CSICC2001, Isfahan, Iran, pp. 284-295, 2001.
- [13] Meybodi, M. R. and Mehdipour, F. "VLSI Placement Using Cellular Learning Automata", Journal of Modares, University of Tarbeit Modares, Vol. 16, pp. 81-95, summer 2004.
- [14] Meybodi, M. R. and Taherkhani, M. "Application of Cellular Learning Automata to Modeling of Rumor Diffusion", in Proceedings of 9th Conference on Electrical Engineering, Power and Water institute of Technology, Tehran, Iran, pp. 102-110, May 2001.
- [15] Meybodi, M. R. and Kharazmi, M. R. "Image Restoration Using Cellular Learning Automata", in Proceedings of the Second Iranian Conference on Machine Vision, Image Processing and Applications, KNU University, Tehran, Iran, pp. 261-270, 2003.
- [16] Meybodi, M. R., Beigy, H. and Taherkhani, M. "Cellular Learning Automata", Proceedings of 6th Annual International Computer Society of Iran Computer Conference CSICC2001, Isfahan, Iran, pp. 153-163, 2001.
- [17] Rastegar, R. and Meybodi, M. R. "A New Evolutionary Computing Model based on Cellular Learning Automata", to appear in proceedings of IEEE conference on Cybernetics and Intelligent Systems 2004 (CIS2004), Singapore, December 2004.
- [18] Rastegar, R., Hariri, A. and Meybodi, M. R. "A Fuzzy Clustering Algorithm using Cellular Learning Automata based Evolutionary Algorithm", accepted in
- [1] Schönfisch, B. and de Roos, A. "Synchronous and asynchronous updating in cellular automata", BioSystems, Vol. 51, pp. 123-143, 1999.
- [2] De Jong, K. A. "The Analysis of the behavior of a class of genetic adaptive systems" Ph.D. dissertation, University of Michigan, Ann Arbor, 1975.
- [3] Meybodi, M. R., Beigy, H. and Taherkhani, M. "Cellular Learning Automata and Its Applications", Journal of Science and Technology, University of Sharif, No. 25, pp.54-77, Autumn/Winter 2003-2004.
- [4] Beigy, H. and Meybodi, M. R. "A Mathematical Framework for Cellular Learning Automata", Advances in Complex Systems, Vol. 7, Nos. 3-4, pp. 295-320, September/December 2004.
- [5] Beigy, H. and Meybodi, M. R. "A Self-Organizing Channel Assignment Algorithm: A Cellular Learning Automata Approach", Vol. 2690 of Springer-Verlag Lecture Notes in Computer Science, pp. 119-126, Springer-Verlag, 2003.
- [6] Beigy, H. and Meybodi, M. R. "A Dynamic Channel Assignment Algorithm: A Cellular Learning Automata Approach", Proceedings of The 2nd Workshop on Information Technology & Its Disciplines, pp. 218-231, Kish Island, Iran, February 24-26, 2004.
- [7] Ghanbari, S. and Meybodi, M. R. "Load Balancing in Grid Computing Using Cellular Learning automata", Technical Report, Computer Engineering Department, Amirkabir University, Tehran, Iran, 2004.
- [8] A. Hariri, A., Rastegar, R., Saheb Zamani, M. R. and Meybodi, M. R., "A Parallel Hardware Implementation of CLA-EC on FPGA", Technical Report, Computer Eng. Department, Amirkabir university, 2004.
- [9] Khojasteh, M. R. and Meybodi, M. R. "Cooperation in Multi-Agent Systems Using Learning Automata", Iranian Journal of Electrical and Computer Engineering, Vol. 1, No. 2, pp.81-91, 2004.
- [10] Mars, P., Chen, J. R. and Nambiar, R. Learning algorithms theory and applications in signal

- In Proceedings of NIPS'97, pp. 424–431, MIT Press, Cambridge, MA, 1997.
- [31] Gordon, T. G., Marsh, C. and Wu, Q. H. “Stochastic Optimal Control of Active Vehicle Suspensions Using Learning Automata,” *Journal of Systems and Control Engineering*, vol. 207, pp. 143-152, 1993.
- [32] Harik, G. R., Lobo, F. G. and Goldberg, D. E. "The Compact Genetic Algorithm", *IEEE Transaction on Evolutionary Computing*, vol. 3, no. 4, pp. 287–297, 1999.
- [33] Howell, M. N., Gordon, T. J. and Brandao, F. V. “Genetic Learning Automata for Function Optimization”, *IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol 32, No. 6, pp. 804-815, 2002.
- [34] Julstrom, B. A. “Comparing Darwinian, Baldwinian, and Lamarckian Search in Genetic Algorithm for 4-Cycle Problem”, Technical Report, Department of Computer Science, St. Cloud State University, St. Cloud, USA, 1999.
- [35] Lakshminarayanan, S. *Learning algorithms: Theory and Applications*, New York: Springer-Verlag, 1981.
- [36] Lessani, M. and Meybodi, M. R. “Genetic Cellular Automata”, in proceedings of the 5th conference on Intelligent Systems, Mashhad, Iran, 2003.
- [37] Mühlenbein, H. and Mahnig, T., “Evolutionary Algorithms: From Recombination to Search Distributions”, *Theoretical Aspects of Evolutionary Computing*, Springer Publication, 2001.
- [38] Mühlenbein, H. and Mahnig, T. “The Factorized Distribution Algorithm for additively decomposed functions”, *Proceedings of the 1999 Congress on Evolutionary Computation*, IEEE press, pp. 752-759, 1999.
- [39] Mühlenbein, H. and Paab, G. “From Recombination of Genes to the Estimation of Distributions I. Binary Parameters”, In *Lecture Notes in Computer Science 1411: Parallel Problem Solving from Nature-PPSN IV*, pp. 178-187, 1996.
- [40] Mühlenbein, H. and Pelikan, M. “The Bivariate Marginal Distribution Algorithm”, *Advances in Soft Computing-Engineering Design and Manufacturing*, pp. 521-535, 1999.
- [41] Najim, K. and Poznyak, A. S., editors, *Learning automata: theory and application*, Tarrytown, New York: Elsevier Science Publishing Ltd., 1994.
- [42] Narendra, K. S. and Thathachar, M. A. L. “Learning automata a survey,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 4, no. 4, July 1974.
- [43] Narendra, K. S. and Thathachar, M. A. L. *Learning Automata: An introduction*, Prentice Hall, 1989.
- [44] Pelikan, M., Goldberg, D. E. and Cant-Paz, E. “Linkage Problem, Distribution Estimation and Bayesian Networks”, *Evolutionary Computation*, vol. 8, no. 3, pp. 311-340, 2000.
- International Conference on Hybrid Intelligent Systems (HIS 2004), Japan, 2004. (To appear in *IEEE Computer Society*)
- [19] Rastegar, R., Rahmati, M. and Meybodi, M. R. “A CLA-EC based Clustering Algorithm”, to appear in *IEEE Conference in Advances Artificial Intelligence: Theory and Application (AISTA 2004)*, Luxemburg, October 2004.
- [20] Saheb Zamani, M., Mehdipour, F. and Meybodi, M. R. "Implementation of Cellular Learning Automata on Reconfigurable Computing Systems", *IEEE CCGEI 2003 Conference*, Montreal, Canada, May 2003.
- [21] 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.
- [22] Rastegar, R. and Meybodi, M. R. “Experiments with Cellular Learning Automata”, Technical Report, Computer Engineering Department, Amirkabir University, Tehran, Iran, May 2005.
- [23] Kharazmi, M. R. and Meybodi, M. R. “Application of Cellular Learning Automata to Image Segmentation”, *Proceedings of Tenth Conference on Electrical Engineering (10<sup>th</sup> ICEE)*, University of Tabriz, Vol 1, pp. 298-306, May 2002.
- [24] Kharazmi, M. R. and Meybodi, M. R. “An Algorithm Based on Cellular Learning Automata for Image Restoration”, *Proceedings of The First Iranian Conference on Machine Vision & Image Processing*, University of Birjand, pp. 244 –254, March 2001.
- [25] Marchini, F. and Meybodi, M. R. “Application of Cellular Learning Automata to Image Processing: Finding Skeleton” ,*Proceedings of The Third Conference on Machine Vision, Image Processing and Applications (MVIP 2005)* University of Tehran, Tehran, Iran, pp. 271-280, Feb. 2005.
- [26] Alba, E. and Troya, J. M. “Analyzing Synchronous and Asynchronous Parallel Distributed Genetic Algorithms”, *Future Generation Computer Systems*, vol. 17, pp. 451-465, 2001.
- [27] Baluja, S. “Population Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning”, Technical Report CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1994.
- [28] Baluja, S. and Davies, S. “Using Optimal Dependency Trees for Combinatorial Optimization: Learning the Structure of Search Space”, Technical Report CMU-CS-97-107, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1997.
- [29] Baluja, S. and Caruana, R. "Removing The Genetics from The Standard Genetic Algorithm", In *Proceedings of ICML'95*, pp. 38–46, Morgan Kaufmann Publishers, Palo Alto, CA, 1995.
- [30] De Bonet, J. S., Isbell, C. L. and Viola, P. "MIMIC: Finding Optima by Estimating Probability Densities",

- [45] Pelikan, M., Goldberg, D. E. and Lobo, F. "A Survey of Optimization by Building and Using Probabilistic Model", Illinois Genetic Algorithm Report, no. 99018, Illinois University, Illinois, USA, September 1999.
- [46] Rastegar, R. and Meybodi, M. R. "A New Estimation of Distribution Algorithm based on Learning Automata", to appear in Iranian Journal of Electrical and Computer Engineering.
- [47] Rastegar, R. and Meybodi, M. R. "A New Estimation of Distribution Algorithm based on Learning Automata", to appear in proceedings of IEEE Conference in Advances Artificial Intelligence: Theory and Application (AISTA 2004), Luxemburg, October 2004.
- [48] Rastegar, R. and Meybodi, M. R.. "LAEDA: A New Evolutionary Algorithm using Learning Automata", accepted in 10th Annual International Computer Society of Iran Computer Conference CSICC2004, Sharif university, Tehran, Iran, 2004.
- [49] Riopka, T. P. and Bock, P. "Intelligent Recombination Using Individual Learning in a Collective Learning Genetic Algorithm", In Proc. of the Genetic and Evolutionary Computation Conf. (GECCO-2000), pp. 104-111, Morgan Kaufmann, 2000.
- [50] Rudolph, G. and Joachim, S. "A Cellular Genetic Algorithm with Self-Adjusting Acceptance Threshold", Genetic Algorithms in Engineering Systems: Innovations and Applications, Conference Publication, no. 414, pp. 365-372, 1995.
- [51] Smith, J. and Fogarty, T. C. "Self Adaptation of Mutation Rates in a Steady State Genetic Algorithm", In Proc. 3rd IEEE Conf. on Evolutionary Comp. IEEE Press, 1996.
- [52] Tomassini, M. "The Parallel Genetic Cellular Automata: Application to Global function Optimization", in proceedings of International Artificial Neural Nets and Genetic Algorithms Conference, Austria, pp. 385-391, Springer, Wein, 1993.
- [53] Whitley, D., Gordon, V. S. and K. Mathias, K. "Lamarckian Evolution, The Baldwin Effect and Function Optimization", Parallel Problem Solving from Nature III, Springer-Verlag, 1994.
- [54] Wolfram, S. Cellular Automata and Complexity, Perseus Books Group, 1994.
- [55] Harik, G. "Learning Linkage to Efficiently Solve Problems of Bounded Difficulty Using Genetic Algorithms", Illinois Genetic Algorithm Report, No. 97005, Illinois University, Illinois, USA, 1997.
- [56] Rastegar, R. and Meybodi, M. R. "A New Evolutionary Computing Model based on Cellular Learning Automata", to appear in proceedings of IEEE International Conference on Cybernetics and Intelligent Systems 2004, Singapore, 2004.