

همکارانه CLA-EC

* بهرنگ مسعودی‌فر^{*} محمد رضا میبدی^{*} مجتبی هاشمی^{**}

^{*}دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران

^{**}دانشکده مهندسی کامپیوتر، دانشگاه شیراز، شیراز، ایران

چکیده

اتوماتای یادگیر سلولی^۱ (CLA) مدلی برای سیستم‌هایی است که از اجزاء ساده‌ای به نام سلول تشکیل شده‌اند و رفتار هر جزء بر اساس رفتار همسایگانش و نیز تجربیات گذشته‌اش تعیین و اصلاح می‌شود. مدل CLA-EC که اخیراً پیشنهاد شده است یک الگوریتم تکاملی است که از ترکیب CLA و مفاهیم در محاسبات تکاملی (EC) بدست آمده است. نشان داده شده است که CLA-EC همگام در حل مسائل بهینه‌سازی در مقایسه با CLA و یا الگوریتم‌های رئتیکی از کارایی بالاتری برخوردار می‌باشد. سرعت همگرایی و یا دقت این مدل در حل مسائل بهینه‌سازی بستگی زیادی به انتخاب مناسب پارامترهای این مدل دارد. انتخاب مقادیر CLA-EC نامناسب برای این پارامترها ممکن است منجر به سرعت همگرایی پایین و یا به دام افتادن در بهینه‌های محلی گردد. برای حل این مشکل در این مقاله همکارانه پیشنهاد می‌شود. در CLA-EC همکارانه، چندین CLA-EC که هر کدام دارای مقادیر متفاوت برای پارامترهای خود می‌باشند در حل مسئله همکاری می‌کنند. از این طریق می‌توان نه تنها سرعت همگرایی به راه حل بهینه را افزایش داد بلکه از به دام افتادن در بهینه‌های محلی جلوگیری نمود. برای نشان دادن برتری مدل CLA-EC همکارانه بر مدل CLA-EC مسائل بهینه‌سازی توابع استاندارد دوچون^۲، P-Peaks، کوله پشتی ۱/۰ با استفاده از هر دو مدل حل گردیده است.

کلمات کلیدی: اتماتای یادگیر سلولی، اتماتاهای یادگیر، CLA-EC، CLA-EC همکارانه، بهینه‌سازی، محاسبات تکاملی

Cooperative CLA-EC

B. Masoodifar* M. R. Meybodi* M. Hashemi**

*Computer Engineering and Information Technology Department, Amirkabir University, Tehran, Iran

**Computer Engineering Department, Shiraz University, Shiraz, Iran

Abstract

Cellular Learning Automata (CLA) which is obtained by combining cellular automata and learning automata is a mathematical model for dynamical complex systems that like cellular automata consists of a large number of simple components. These simple components, which have learning capability, act together to produce complicated behavioral patterns. CLA-EC which is introduced recently is a model which is obtained by combining CLA model and the evolutionary model. It has been shown that synchronous CLA-EC outperforms CLA and Genetic algorithms in solving optimization problems. The speed and accuracy of CLA-EC in solving optimization problems is very much dependent on the values chosen for its parameters. Inappropriate selection of values for the parameters may lead to slow convergence or trapping in local minima. In this paper, cooperative CLA-EC (CCLA-EC) as a solution to this problem has been suggested. In a CCLA-EC there are several CLA-EC cooperating to solve a particular problem. Using CCLA-EC results in higher speed of convergence and also avoiding local minima. To show the superiority of CCLA-EC over CLA-EC, optimization problems DeJong Functions, P-Peaks and zero one knapsack are solved.

Keywords: Cellular Learning Automata, Learning Automata, CLA-EC, Cooperative CLA-EC, Evolutionary Algorithm, Optimization.

¹ Cellular Learning Automata

² DeJong Functions

-۱ مقدمه

پارامترها ممکن است منجر به سرعت همگرایی پایین و یا به دام افتادن در بهینه CLA های محلی گردد. اگر نرخ یادگیری اتوماتاهای یادگیر مستقر در سلولهای بالا باشد، سرعت همگرایی افزایش می‌یابد ولی در عوض ممکن است الگوریتم در جوابهای بهینه محلی به دام می‌افتد و اگر از نرخ یادگیری پایین استفاده شود، الگوریتم از دقت خوبی برخوردار می‌شود ولی سرعت همگرایی به شدت کاهش می‌یابد. برای حل این مشکل در این مقاله مدل CLA-EC همکارانه پیشنهاد می‌شود. در CLA-EC همکارانه، چندین CLA-EC که هر کدام دارای مقادیر متفاوت برای پارامترهای خود میباشند در حل مساله همکاری میکنند. در مدل همکارانه از چندین CLA-EC با نرخهای یادگیری متفاوت استفاده می‌شود. این CLA-EC با یکدیگر در تعامل میباشند و جوابهای میانی بدست آمده را با هم به اشتراک می‌گذارند. مدل همکارانه تلاش می‌کند هم از سرعت بالای CLA-EC با نرخ یادگیری بالا، و هم از دقت بالای CLA-EC با نرخ یادگیری پایین بهره ببرد. از طریق CLA-EC همکارانه میتوان نه تنها سرعت همگرایی به راه حل بهینه را افزایش داد بلکه از به دام افتادن در بهینه های محلی جلوگیری نمود. مدل CLA-EC همکارانه بر روی مسایل بهینه‌سازی توابع استاندارد دوچونگ^۲, P-Peaks کوله پشتی ۱/۰ مورد آزمایش قرار گرفت. نتایج آزمایشها نشان داد که مدل CLA-EC همکارانه در حل مسایل در مقایسه با مدل CLA-EC از سرعت و دقت بالاتری برخوردار است.

ادامه مقاله بدینصورت سازماندهی شده است. در بخش‌های ۳، ۴ و ۵ به ترتیب اتوماتای سلولی، اتوماتای یادگیر و اتوماتای یادگیر سلولی مختصراً شرح داده می‌شود. در بخش ۵ یک الگوریتم بهینه‌سازی مبتنی بر CLA-EC ساده و در بخش ۶ CLA-EC همکارانه شرح داده می‌شود. نتایج آزمایشها در بخش ۷ آمده است. بخش نهایی 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 گستته‌ای هستند که رفتارشان کاملاً بر اساس ارتباط محلی استوار است. در اتوماتای سلولی یک مجموعه منظم از سلول‌ها وجود دارد که هر کدام می‌تواند 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 گسته بر طبق یک قانون محلی بهنگامرسانی می‌شوند. در تعیین وضعیت جدید برای هر سلول، مقادیر سلول‌های همسایه نیز تأثیرگذار هستند. انتخاب قوانین مختلف برای بهنگامرسانی، انواع متفاوتی از اتوماتاهای 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

اتوماتای یادگیر سلولی (CLA) که از ترکیب اتوماتاهای یادگیر^۱ (LA) و اتوماتای سلولی^۲ (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 است 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 باقیستی پاداش داده شود و یا جریمه شود. عمل دادن پاداش و یا جریمه، منجر به بروزرسانی 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.

یادگیری به عنوان یک پدیده تطبیقی در طول مدت بقای یک موجود 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 است. نظریه‌های متفاوتی در این باره ارائه شده‌اند که دو مورد از مهمترین آنها نظریه لامارکین و بالدوین می‌باشد. بر اساس نظریه 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 نظریه بالدوین یادگیری با توشیق موققتهایی که در فرایند تکامل به صورت جزئی صورت می‌پذیرند، سبب هدایت و تسريع تکامل می‌گردد[49]. اگرچه نادرستی بعضی از این نظریه‌ها در حال حاضر به طور کامل از سوی محققین علوم طبیعی مورد قبول می‌باشد، اما منشأ کارهایی در زمینه محاسبات مهم از طبیعت است[53][34]. یکی از مدل‌هایی که از ترکیب مفاهیم تکامل و یادگیری بدست آمده است مدل CLA-EC میباشد که ترکیبی از مدل LA و محاسبات تکاملی(EC) میباشد [61][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 رشتة 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 بهنگام می‌کنند. فرایند انتخاب عمل‌ها و بهنگام 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. نشان داده شده است که CLA-EC ساده در حل مسایل بهینه‌سازی در مقایسه با CLA-EC و یا الگوریتمهای ژنتیکی از کارایی بالاتری برخوردار می‌باشد.

سرعت همگرایی و دقت این مدل در حل مسایل بهینه سازی بستگی زیادی به انتخاب مناسب پارامترهای این مدل دارد. انتخاب مقادیر نامناسب برای این

³ Dejong Functions

¹ Learning Automata

² Cellular Automata

۳- اتوماتاهای یادگیر

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

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

ب- پاسخ نامطلوب

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

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

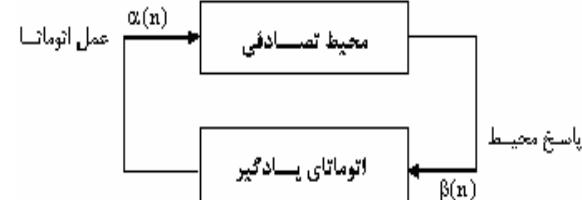
در روابط فوق، a پارامتر پاداش و b پارامتر جریمه است. با توجه به مقادیر a و b سه حالت را می‌توان در نظر گرفت. زمانیکه a و b با هم برابر باشند، الگوریتم L_{RP} ^۳ می‌نامیم. زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را L_{RI} ^۴ می‌نامیم. برای اطلاعات بیشتر درباره اتوماتاهای یادگیر و کاربردهای آن میتوان به [41-43][35][3][21]

۴- اتوماتای یادگیر سلوالی

اتوماتای یادگیر سلوالی، CIA ، 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 سلوالی، از یک اتوماتای سلوالی تشکیل شده است که هر سلول آن به یک یا بیشتر اتوماتای یادگیر مجهز می‌باشد که وضعیت این سلول را مشخص می‌سازند. همانند اتوماتای سلوالی، قانون محلی در محیط حاکم است و این قانون تعیین می‌کند که آیا عمل انتخاب شده توسط یک اتوماتا در یک سلول باستی پاداش داده شود و یا جریمه شود. عمل دادن پاداش و یا جریمه منجر به بروز درآوردن ساختار اتوماتای یادگیر سلوالی به منظور نیل به یک هدف مشخص می‌گردد. در اتوماتای یادگیر سلوالی 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 سلوالی به این صورت است که در هر لحظه هر اتوماتای یادگیر در اتوماتای یادگیر سلوالی 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 سلوالی 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 خو را بروز می‌کند. بعد از بروز درآوردن ساختار داخلی، هر اتوماتای یادگیر در اتوماتای یادگیر سلوالی 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 بهنگام سازی ساختار اتوماتاهای یادگیر موجود در اتوماتای یادگیر سلوالی توسط الگوریتم $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ انجام می‌شود. برای اطلاعات بیشتر درباره

یک اتوماتای یادگیر یک ماشین با حالات محدود^۱ 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. 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 تابی $E \equiv \{\alpha, \beta, C\}$ نشان داد که در آن $\alpha = \{\alpha_1, \dots, \alpha_r\}$ $\beta = \{\beta_1, \dots, \beta_m\}$ مجموعه خروجیها و $C = \{c_1, \dots, c_r\}$ مجموعه دو عضوی باشد. هرگاه β مجموعه جزوی باشد می‌تواند محیط از نوع P می‌باشد. در چنین محیطی $\beta_1 = 0$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته می‌شود. در محیط Q ، $\beta(n)$ می‌تواند به طور گستره‌یک 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 احتمال اینکه عمل i نتیجه نا مطلوب 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 به شرح مختصری از آن میپردازیم.



شکل ۱: ارتباط بین اتوماتای یادگیر و محیط

اتوماتاهای یادگیر با ساختار متغیر^۲: اتوماتای یادگیر با ساختار متغیر توسط α, β, p, T نشان داده می‌شود که در آن $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ $\beta = \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه عملهای اتوماتا، $p = \{p_1, p_2, \dots, p_m\}$ بردار احتمال انتخاب هریک از عملها و $T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می‌باشد. در این نوع از اتوماتاهای یادگیر، اگر عمل i در مرحله n انجام شود و پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ کاهش یافته و سایر احتمالهای افزایش می‌یابند. در هر حال، تغییرات به گونه‌ای صورت می‌گیرد تا حاصل جمع $p_i(n)$ هماواره ثابت و مساوی یک باقی بماند. الگوریتم زیر یک نمونه از الگوریتم‌های یادگیر خطی است.

الف- پاسخ مطلوب

³ Linear reward penalty

⁴ Linear reward epsilon penalty

⁵ Linear reward inaction

¹ Finite State Machine

² Variable structure

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 تجارت قبلي 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، اتماتاهای یادگیر سلولی و الگوریتم ژنتیکی نشان میدهد.

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

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

یک قانون برای CLA-EC از دو بخش تشکیل شده است. بخش اول استراتژی انتخاب و بخش دوم استراتژی تولید سیگنال تقویتی می باشد. منظور از استراتژی انتخاب سلول t ام، نحوه انتخاب تعداد مشخص عملگرهای ژنتیکی P^i را دارند. مقدار P^i از میان m می باشد. در استراتژی مورد استفاده، Se ژنوم از بهترین ژنومهای P^i را انتخاب می کنیم و آن 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 $(\beta_t^{i,1}, \dots, \beta_t^{i,n}) = (\bar{\beta}_t^i)^T$ بردار مورد نظر باشد که $\bar{\beta}_t^i \in \{0,1\}^n$. ۱ به عنوان جرمیه و ۰ به عنوان پاداش درنظر گرفته می شود. اگر $(X_t^{i,1}, \dots, X_t^{i,n}) = (X_t^i)^T$ رشته ژنومی سلول t در 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 $\delta_l(X_t^{i,j} = k | P_{Se}^i) = 0$ مقدار $\beta_t^{i,j}$ به $N_{i,j}(k)$ می باشد. سپس با استفاده از صورت زیر بدست می آید:

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

اتماتاهای یادگیر سلولی و کاربردهای آن میتوان به [3][22-24][4-13] مراجعه نمود.

۵- مدل CLA-EC

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

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

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 مشکل از تعدادی اتماتاهای یادگیر می باشد که بر اساس تجارت 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 اتماتاهای یادگیر مدل ژنومی به بیتها 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 اتماتای یادگیر در نظر بگیریم. به این ترتیب با فرض دودویی بودن فضای جستجو برای رشته ژنومی به طول n (با فرض زوج بودن)، تعداد $n/2$ اتماتای یادگیر، هر کدام با چهار عمل خواهیم داشت. برای مثال برای این مورد اگر یک اتماتای یادگیر عمل ۳ را انتخاب کنند به معنای آن است که بیت ۱ و ۲ دسته متناظر آن مقادیر ۱ و ۰ 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 اتماتای یادگیر در مدل ژنومی خواهیم داشت در نظر می گیریم. هر اتماتای یادگیر مقدار بیت متناظر با خود را تعیین میکند.

حال برای ادامه بحث یک اتماتای یادگیر سلولی، $(\underline{L}_1, \dots, \underline{L}_k)$ با k سلول که هر سلول مجهز به n اتماتای یادگیر است در نظر بگیرید. حالت هر سلول توسط یک رشته ژنومی به طول n نشان داده میشود و نتیجتا هر سلول میتواند 2^n حالت مختلف اختیار کند. با فرض همگام بودن اتماتای یادگیر سلولی، در زمان t هر سلول، \bar{l} ، 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 مقابله وجود نخواهد داشت(البته این روش معادلی در بین سیستمهای طبیعی ندارد). ژنوم بر اساس ژنومهایی که انتخاب نموده است، یک بردار سیگنال تقویتی را ساخته و به اتماتاهای یادگیر خود می دهد(این مکانیسم در معادل جفت گیری ژنوم مورد نظر می باشد).

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 اتماتاهای یادگیر مقدار بیت متناظر خود در ژنوم را اعلام می کنند. در صورتی که ژنوم جدید، new_{t+1}^i ، ارزش

الگوریتم تکاملی مبتنی بر اتوماتای یادگیر سلولی همکارانه در شکل ۲ نشان داده شده است.

در الگوریتم شکل ۲ از دو CLA-EC یکی با نرخ یادگیری بالا

$CLA-EC_H$ و دیگری با نرخ یادگیری پایین $(CLA-EC_L)$ استفاده می‌شود. در هر تکرار، $CLA-EC_L$ تلاش می‌کند به طور گسترده فضای جواب را جستجو کند و به موازات آن، $CLA-EC_H$ به سرعت به سمت جواب بهینه $CLA-EC$ حرکت می‌کند. در انتهای هر تکرار، جوابهای سلولهای متناظر در دو $CLA-EC$ با یکدیگر تعویض می‌شوند. به این ترتیب، $CLA-EC_H$ جوابهای همکار خود را دریافت کرده و به سرعت به سمت جواب بهینه در محدوده این جوابها حرکت می‌کند. $CLA-EC_L$ نیز جوابهای همکار خود را دریافت کرده و جستجو را با استفاده از آن جوابها ادامه میدهد. نتایج شبیه سازیها نشان داده است که از این طریق از به دام افتادن الگوریتم در جوابهای بهینه محلی جلوگیری می‌شود.

۷- نتایج آزمایشها

برای آزمایشها از $CLA-EC$ خطی با شعاع همسایگی ۱ استفاده شده است. اتوماتای یادگیر هر سلول از نوع L_{RI} می‌باشد. $CLA-EC$ همکارانه شامل دو $CLA-EC$ خطی ساده $CLA-EC_H$ و $CLA-EC_L$ هر کدام با شعاع همسایگی ۱ می‌باشد. $CLA-EC_H$ شامل اتوماتاهای یادگیر L_{RI} با نرخ یادگیری بالا و $CLA-EC_L$ شامل اتوماتاهای یادگیر L_{RI} با نرخ یادگیری پایین می‌باشد. طول هر دوره، ۱۰ تکرار در نظر گرفته شده است. برای بررسی تأثیر تعداد سلول‌ها در کارایی الگوریتم، آزمایشها برای تعداد سلول‌های مختلف از ۱۰ تا ۵۰ سلول در $CLA-EC$ انجام گرفته است. برای سهولت در ارائه از $CLA-EC$ (automata(a,b),r,se,q) برای اشاره به الگوریتم سلول، شعاع همسایگی a ، تعداد سلول انتخابی b ، اتوماتای یادگیر با پارامترهای a ، b و q از $CLA-EC$ استفاده می‌شود.

برای اشاره به $CLA-EC(automata_L(a_i,b_i),automata_H(a_h,b_h),r,se,q)$ الگوریتم $CLA-EC$ همکارانه با جمعاً q سلول (مجموع سلولهای در هر دو $CLA-EC$)، شعاع همسایگی a ، تعداد سلول انتخابی b ، اتوماتای یادگیر با پارامترهای a و b در $CLA-EC_L$ و $CLA-EC_H$ با یادگیر می‌باشد.

شکل ۳ معماری $CLA-EC$ مورد استفاده در آزمایشها می‌باشد در این شکل X_i وضعیت سلول i را مشخص می‌کند. $X_{i,j}$ محیط اتوماتای j ام سلول i ام است. ورودی محیط شامل $x_{i,j}$ و $x_{i-1,j}$ است که اعمال انتخاب شده توسط اتوماتاهای یادگیر متناظر در سلول‌های همسایه می‌باشند. این محیط، پاداش یا جریمه اتوماتای j ام سلول i را تعیین می‌کند. LA ها بیهوده از $CLA-EC$ متناظر در هر زنوم را تعیین می‌کنند و $comparator$ نتیجه بدست آمده جدید را با نتیجه قابل مقایسه کرده و در صورت بهبود جواب، وضعیت سلول را تغییر می‌دهد. شکل ۴ معماری $CLA-EC$ همکارانه را نشان می‌دهد. در این شکل هر $CLA-EC$ همکارانه با q سلول، شامل دو $CLA-EC$ ساده با $q/2$ سلول است و پس از گذشت هر دوره زمانی از پیش تعیین شده، هر $CLA-EC_H$ ، سلول i خود را با سلول i متناظر در $CLA-EC_L$ معاوضه می‌کند.

برای انجام آزمایشها از سه مساله بهینه‌سازی استفاده شده که در ادامه به

شرح آنها می‌پردازیم.

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

که $(.)u$ تابع پله است.

۶- همکارانه CLA-EC

کارایی $CLA-EC$ بستگی زیادی به نرخهای یادگیری اتوماتاهای یادگیر در سلولهای $CLA-EC$ دارد. اگر نرخ یادگیری اتوماتاهای یادگیر بالا انتخاب شود، معمولاً الگوریتم به سرعت با بهره برداری^۱ از جواب‌های موجود به سمت جواب حرکت می‌کند ولی بدليل اینکه فضای جستجو بطور کامل کاوش^۲ نمی‌شود ممکن است در نقاط بهینه محلی به دام افتد. در مقابل اگر نرخ یادگیری پایین انتخاب شود، بدليل اینکه کاوش در فضای جستجو به طور کامل صورت می‌پذیرد پاسخ بدست آمده توسط الگوریتم از دقت بسیار خوبی برخوردار است ولی در عرض سرعت همگراشی پایین می‌باشد. یک روش برای حل مشکل فوق الذکر استفاده از چندین $CLA-EC$ با نرخهای یادگیری متفاوت می‌باشد. $CLA-EC$ های همکار از طریق به استراک گذاشتن جوابهای میانی هم از سرعت بالا و هم از دقت بالا بهره می‌برند.

$CLA-EC$ همکار ارایه شده در این مقاله، شامل دو $CLA-EC$ مشخصات کاملاً یکسان ولی نرخهای یادگیری متفاوت برای اتوماتاهای یادگیر مستقر در سلولها می‌باشد. هر $CLA-EC$ در $CLA-EC$ همکار به طور دوره‌ای، جواب میانی بدست آمده در هر سلول را با جواب میانی سلول متناظر در $CLA-EC$ دیگر، تعویض می‌کند. بدین ترتیب یکی از $CLA-EC$ ها عمل کاوش و دیگری عمل بهره برداری را انجام می‌دهد. همکاری $CLA-EC$ ها حرکت سریع و دقیق به سمت جواب بهینه را باعث می‌گردند [58-60].

```

Initialize.
While not done do
    t = t + 1
    For each cell i in both  $CLA-EC_L$  and  $CLA-EC_H$  do
        in parallel
            Generate new genome
            Evaluate new genome
            If  $f(\text{new genome}) > f(\text{old genome})$  then
                Accept new genome
            End if
            Select Se cells from neighbor
            Generate the reinforcement signal vector
            Update LAS
            If  $(t \bmod N) = 0$  then
                Exchange the genome of cell i in  $CLA-EC_L$ 
                And  $CLA-EC_H$ 
            End parallel for
    End while

```

شکل ۲: الگوریتم تکاملی مبتنی بر اتوماتای یادگیر سلولی همکارانه

فرض کنید X_{ij}^t وضعیت سلول i مربوط به اتوماتای یادگیر سلولی j در گام t باشد. هر یک از $CLA-EC$ ها در $CLA-EC$ همکار به طور همگام مانند آنچه در بخش توضیحات در باره $CLA-EC$ شرح داده شد شروع به فعالیت می‌کنند. بعد از هر N تکرار مقادیر X_{i1}^t و X_{i2}^t با یکدیگر تعویض می‌شوند.

¹ Exploitation

² Exploration

الف) ترکیب‌های نامعتبر حذف شوند. در این حالت باید به صورت اتفاقی و یا حریصانه جواب غیرمجاز را تغییر دهیم تا به یک جواب مجاز تبدیل شود. در چنین حالتی تابع ارزش به صورت زیر تعریف می‌شود.

$$f_{Knapsack}(X) = \sum_{i=1}^n (X_i V_i)$$

ب) ترکیب‌های نامعتبر از جمعیت حذف نشوند، ولی به آنها امتیاز منفی داده شود و جریمه شوند.

$$f_{Knapsack}(X) =$$

$$\sum_{i=1}^n (X_i V_i) + \eta(C - \sum_{i=1}^n (X_i C_i)) u(\sum_{i=1}^n (X_i C_i) - C)$$

که $(.) u$ تابع پله و η ضریب لاغرانژ می‌باشد. از آنجا که احتمال تولید جوابهای مناسب از جوابهای نامعتبر، وجود دارد و اینگونه جوابها با اینکه خود غیرقابل قبولند ولی می‌توانند راهی برای رسیدن به جوابهای بهینه باشند از روش دوم استفاده می‌کنیم، با این توضیح که پس از محاسبه مقدار شایستگی هر جواب، اگر حاصل نامعتبر بود متناسب با اضافه ظرفیت جریمه انجام می‌شود[۵۲]. در این مقاله ضریب منظور شده برای جریمه ۱ در نظر گرفته شده است. نکته مهم دیگری که باید به آن توجه شود و استگی پیچیدگی مساله کوله پشتی به توزیع آماری ارزش و حجم بسته‌ها است. در این مقاله ۱۰۰ بسته با مشخصات تصادفی از اعداد صحیح ۱ تا ۳۰ انتخاب شده‌اند و حداقل ظرفیت کوله‌پشتی، ۱۰۰۰ در نظر گرفته شده است.

مقایسه نتایج بدست آمده از آزمایشات نشان می‌دهد مدل همکارانه علاوه بر اینکه از دقت بسیار خوبی برخوردار است، دارای سرعت بالای نیز می‌باشد. شکل ۵ نشان می‌دهد در CLA-EC با زیاد شدن نرخ یادگیری، سرعت به شدت زیاد شده ولی دقت کاهش می‌باشد. دقیق‌ترین روش متعلق به CLA-EC(LRI(0.0001,0),1,2,-) است، اما این روش به شدت کند است و سرعت آن نسبت به روش‌های دیگر اختلاف زیادی دارد. سریع‌ترین روش را از دقتی معادل با CLA-EC(LRI(0.001,0),1,2,-) دارد اما این روش سریعاً در نقاط بهینه محلی به دام می‌افتد و جوابهای بدست آمده از دقت بسیار کمی نسبت به روش‌های دیگر برخوردار است. اما جالب توجه است که روش همکارانه که از ترکیب دو CLA-EC با نرخهای یادگیری ۰/۰۰۰۱ و ۰/۰۳ بودست آمده است دارای دقتی معادل با CLA-EC(LRI(0.0001,0),1,2,-) است. دیگر سرعت آن حتی از (-۰/۰۰۱,۰),۱,۲,- CLA-EC(LRI(0.001,0),1,2,-) نیز بیشتر است. با بررسی شکل ۶ در می‌یابیم که در مسئله P-Peaks، روش همکارانه دقیق معادل با دقیق‌ترین روش CLA-EC ساده‌یعنی CLA-EC با نرخ یادگیری ۰/۰۰۰۱ دارد ولی دارای سرعتی به مرتب بیشتر از آن است. شکل‌های ۷ و ۸ نیز نشان می‌دهند برای یافتن جواب بهینه تابع دوجونگ F2 و F1 روش EC ساده با نرخ یادگیری ۰/۰۱ دقیق‌تر از سایر روش‌هاست اما مانند مسائل قبل، بسیار کند عمل می‌کند، در عوض روش‌های با نرخ یادگیری ۰/۰۳ و ۰/۰۵ و ۰/۰۷ سرعت‌های بالاتری دارند ولی دقت آنها پایین است. اما روش‌های همکارانه که از ترکیب CLA-EC با نرخ یادگیری ۰/۰۱ و نرخهای یادگیری ۰/۰۳ و ۰/۰۵ و ۰/۰۷ ساخته شده‌اند دارای دقیق‌تر معادل با روش ساده با نرخ یادگیری ۰/۰۱ بوده و سرعت آنها نیز بسیار بهبود یافته است. در شکل ۹ با مقایسه روش‌های ساده با نرخ یادگیری ۰/۰۱ و ۰/۰۵ در می‌یابیم که روش ساده با نرخ یادگیری ۰/۰۱

توابع Dejong: 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 [۲]. توابع Dejong به شرح زیر هستند.

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

$$F_3(X) = 30 - \sum_{i=1}^5 \lfloor x_i \rfloor \quad -5.12 \leq x_i \leq 5.12$$

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

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

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

مسئله P-Peaks: فرض کنید p رشته به صورت تصادفی با طول n تولید شده و به عنوان نقاط بهینه یا Peak در نظر گرفته شوند. در این صورت هنگام ارزیابی هر رشته دلخواه در فضای جستجو، رشته‌هایی که فاصله همینگ آنها ریکی از این نقاط بهینه مینیمم باشد، بیشتر مورد توجه می‌باشند. به عبارت دیگر می‌توانیم تابع ارزیابی را به صورت زیر تعریف کنیم.

$$f_{P-Peaks}(X) = (1/n) \max_{i=1}^p (n - \text{HamDis}(X, \text{Peak}_i))$$

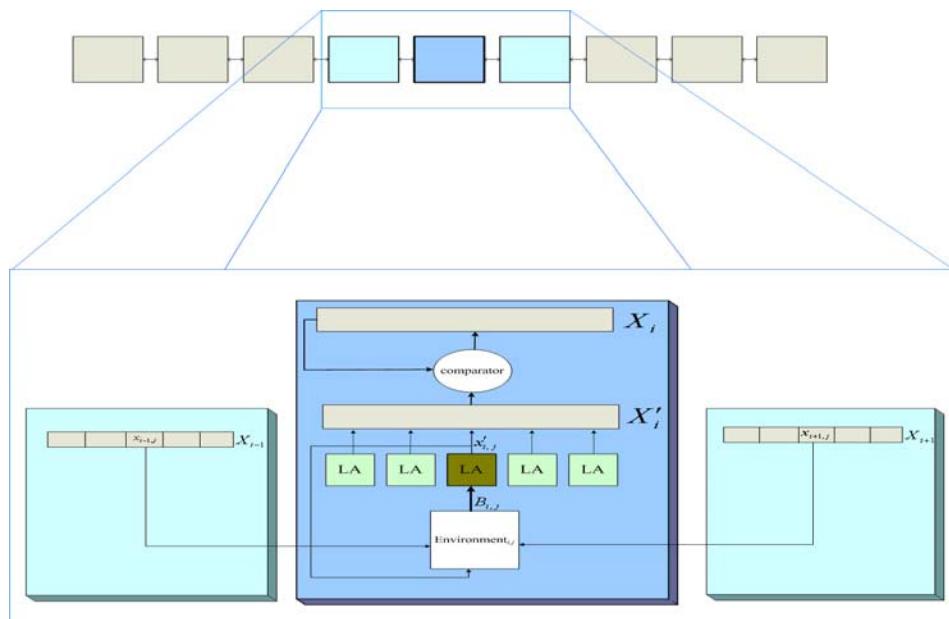
در این مساله هر چه مقدار p بیشتر باشد و استگی متغیرهای مساله نیز بیشتر خواهد بود. به عبارت دیگر می‌توانیم با تغییر مقدار p مسائلی با پیچیدگی‌های متفاوت بدست آوریم. در این مقاله مقدار p برای مساله P-Peaks به ترتیب ۱۰۰ و ۱۰۰ در نظر گرفته شده است [۵۷].

مسئله کوله‌پشتی صفر و یک: یک کوله‌پشتی و تعداد n بسته مفروضند. هر بسته i دارای حجم C_i و ارزش V_i می‌باشد. از طرفی کوله‌پشتی نیز ظرفیتی معین برابر با C دارد، یعنی مجموع حجم بسته‌هایی که می‌توانند در آن قرار بگیرد می‌باشد از مقدار C تجاوز نکند. مسئله از این قرار است که می‌خواهیم طوری بسته‌ها را برای قرار دادن در کوله‌پشتی انتخاب کنیم که در نهایت ارزش بسته‌های درون کوله‌پشتی حداقل شود. در پیاده‌سازی این مسئله، هر راه حل (یک ترکیب از بسته‌ها)، یک ژنوم خواهد بود. فرض کنید n بسته داشته باشیم. هر ژنوم توسط رشته‌ای بیتی، $X = (X_1, \dots, X_n)$ ، به طول n نمایش داده می‌شود. ۱ بودن بیت i ام در این رشته نشان دهنده حضور بسته i ام در ترکیب خواهد بود. نحوه تعیین مقدار تابع شایستگی، با مجموع ارزش بسته‌ها در ترکیب، نسبت مستقیم دارد. شرط ظرفیت کوله‌پشتی نیز بدین صورت است که اگر مجموع حجمها از ظرفیت کوله‌پشتی بیشتر شد، ترکیب غیرمعتبر خواهد بود. از آنجایی احتمال ایجاد هر ترکیبی از بیتها شامل حالت نامعتبر- وجود دارد، ممکن است نتیجه نامعتبر باشد. با راهلهای نامعتبر باشند. با روش زیر می‌توان به دو روش برخورد کرد:

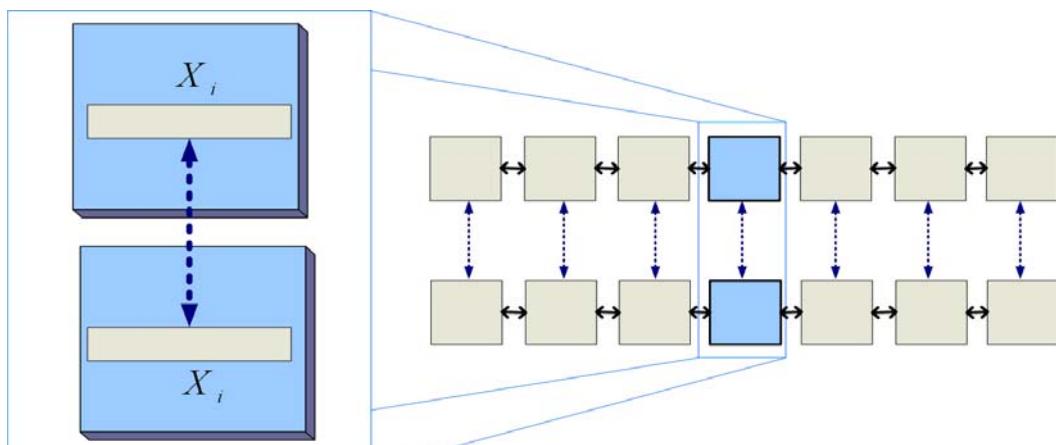
۸- نتیجه‌گیری

در این مقاله مدل CLA-EC همکارانه به منظور غلبه بر مشکلات مدل CLA-EC معرفی گردید. در CLA-EC همکارانه، چندین CLA-EC که هر کدام دارای مقادیر متفاوت برای پارامترهای خود میباشند در حل مساله همکاری میکنند. از این طریق میتوان نه تنها سرعت همگرایی به راه حل بهینه را افزایش داد بلکه از به دام افتادن در بهینه های محلی جلوگیری نمود. کارایی CLA-EC همکارانه از طریق آزمایش آن بر روی مسائل متنوعی نشان داده شد.

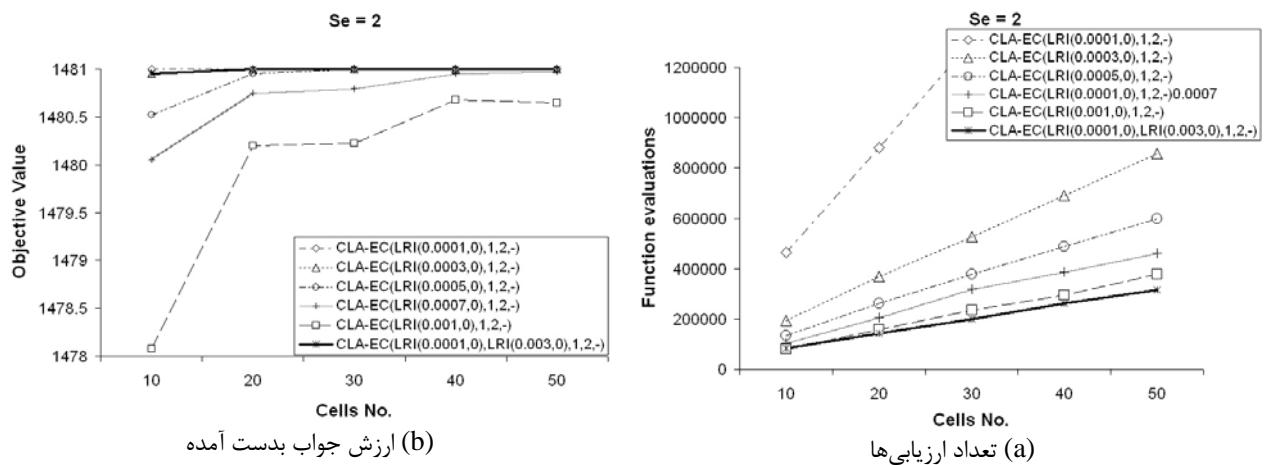
بسیار سریعتر از روش ساده با نرخ یادگیری 0.005 عمل می‌کند ولی از دقت بسیار کمتری برخوردار است. اما روش همکارانه، این دو را با هم ترکیب کرده و از مزایای هردو به نحو موثری بهره برده است به گونه‌ای که علاوه بر اینکه این روش از روش ساده با نرخ یادگیری 0.01 سریعتر عمل می‌کند، دارای دقتی بالاتر از روش ساده با نرخ یادگیری 0.05 نیز می‌باشد. در نهایت شکل ۱۰ نشان می‌دهد که همکارانه با دقتی برابر با CLA-EC با نرخ یادگیری 0.01 با سرعتی بیشتر به جواب بهینه دست پیدا می‌کنند.



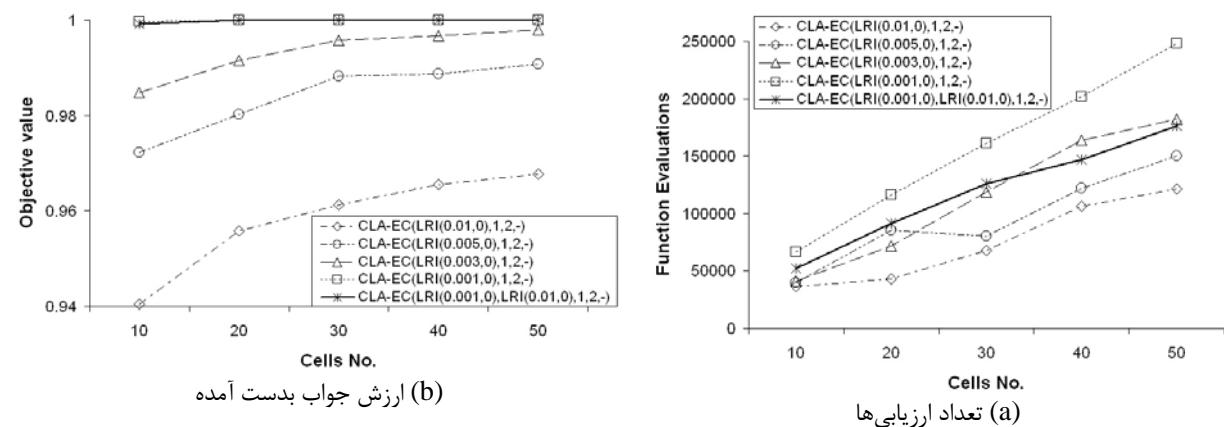
شکل ۳: ساختار CLA-EC شبیه‌سازی شده



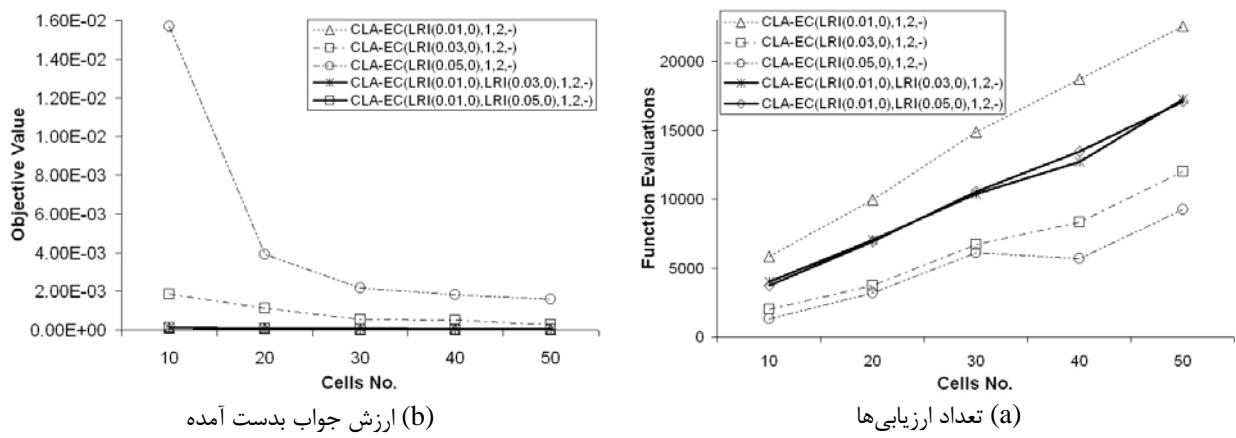
شکل ۴: ساختار CLA-EC همکارانه



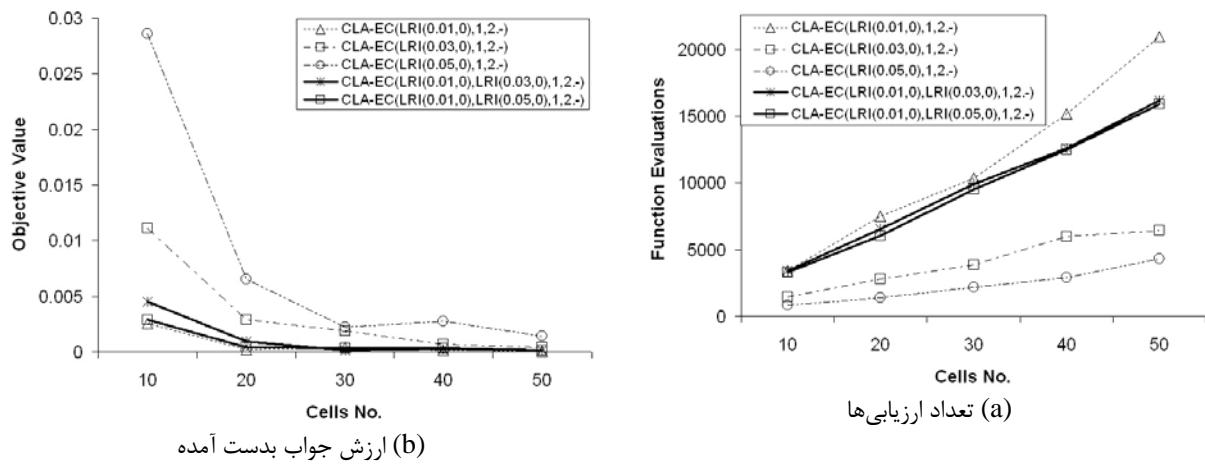
شکل 5: کوله پشتی صفر و یک



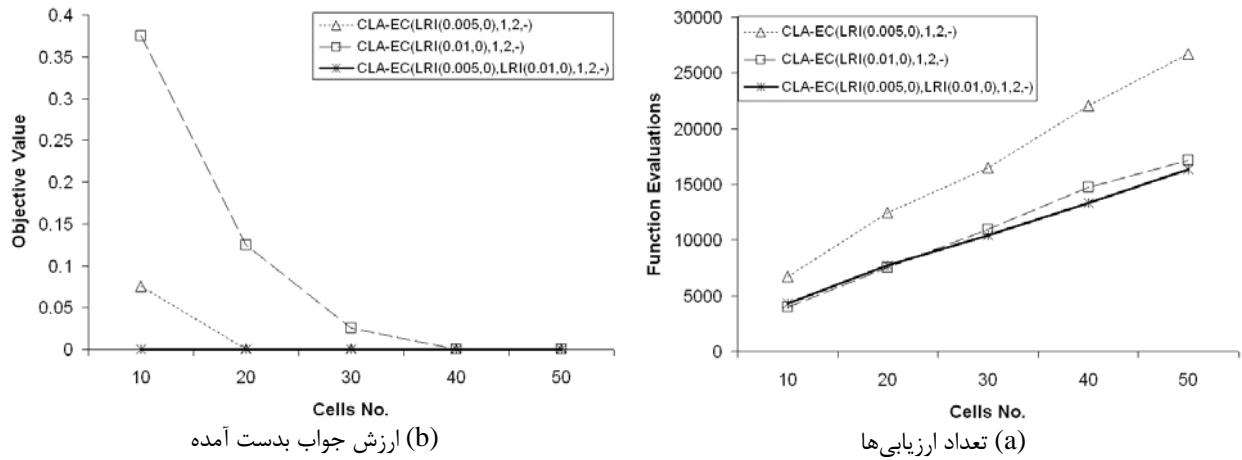
شکل 6: P-Peaks



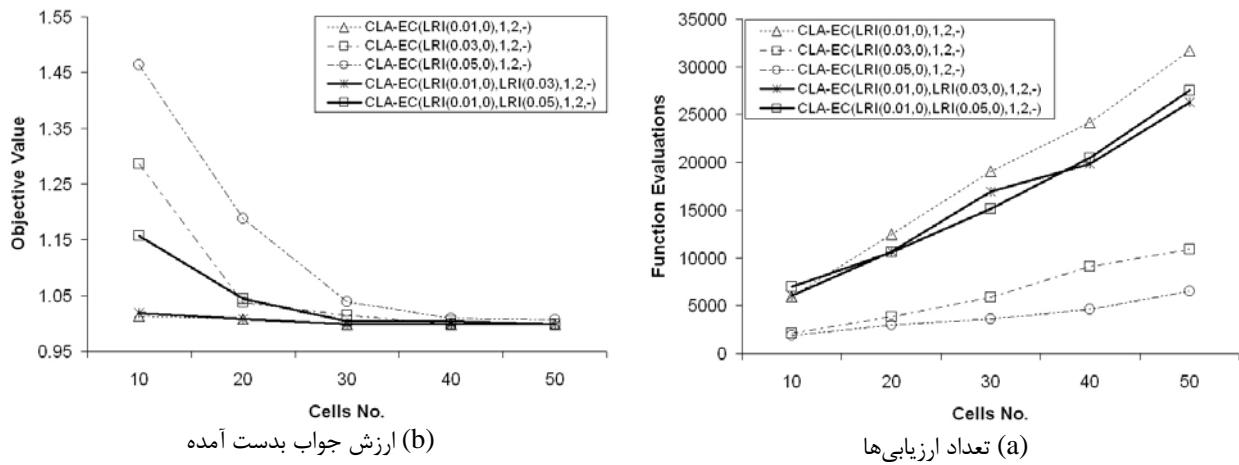
شکل 7



شکل 8



شکل 9



شکل 10

مراجع

- 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”, Proceedings of 2004 IEEE Conference on Cybernetics and Intelligent Systems (CIS), pp. 433-438, Singapore 1-3 Dec. 2004.
- [18] Rastegar, 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.
- [19] Rastegar, R., Rahmati, M. and Meybodi, M. R. “A CLA-EC based Clustering Algorithm”, Proceedings of AISTA'04 Conference, 15-18 Nov. 2004.
- [20] Saheb Zamani, M., Mehdipour, F. and Meybodi, M. R. “Implementation of Cellular Learning Automata on Reconfigurable Computing Systems”, Proceedings of IEEE CCGEI 2003 Conference, pp. 1139-1143, 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 (10th 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.
- [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 & It's 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”, , Proceedings of NASA/DOD Conference on Evolvable Hardware (EH'05), pp. 294-297, Washington DC, USA, June 29-July 1, 2005.
- [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 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

- 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.
- [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 [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", 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] Lakshmivarahan, 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

- 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.
- [57] Alba, E., Giacobini, M., Tomassini, M., and Romero, S., "Comparing Synchronous and Asynchronous Cellular Genetic Algorithms," Proceedings of the Parallel Problem Solving from Nature VII, University of Malaga, Granada, Spain, pp. 601-610, 2002.
- [58] Ursem, R. K., "Models for Evolutionary Algorithms and Their Applications in System Identification and Control Optimization" PhD Dissertation Faculty of Science of the University of Aarhus , April 2003.
- [59] Ursem, R. K., "Diversity-Guided Evolutionary Algorithms" In Merelo et al., editors, Proceedings of Parallel Problem Solving from Nature VII (PPSN-2002), pages 462–471.
- [60] Thomsen, R., Rickers, P., and Krink, T. "A Religion-Based Spatial Model For Evolutionary Algorithms". Parallel Problem Solving from Nature VI (PPSN-2000), volume 1, pages 817–826.
- [61] 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.