

بهبود بهینه سازی گروه ذرات با استفاده از اتوماتای یادگیر سلولی باز

ماندانا حمیدی^{1,3} محمد رضا میبیدی¹

¹ دانشکده مهندسی برق و کامپیوتر و فناوری اطلاعات دانشگاه آزاد اسلامی، واحد قزوین، قزوین، ایران

² دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران

³ دانشکده مهندسی برق و کامپیوتر دانشگاه آزاد اسلامی، واحد زرقان، زرقان، ایران

mmeybodi@aut.ac.ir, Mandana.hamidi@gmail.com

چکیده: در این مقاله به منظور بهبود کارایی بهینه سازی گروه ذرات (*PSO*) مدل جدیدی به نام *OCLAPSO* که ترکیبی از اتوماتای یادگیر سلولی (*OCLA*) و مدل بهینه سازی گروه ذرات (*PSO*) ارائه شده است. در این مدل به هر بعد از فضای جستجو یک اتوماتای یادگیر سلولی اختصاص داده شده است که هر سلول آن یک ذره در همان بعد خاص می باشد. اتوماتای یادگیری که در هر سلول قرار دارد وظیفه ایجاد تنوع در بین ذرات در همان بعد را عهده دارمی باشد. نتایج آزمایشها بر روی مسایل نمونه نشان میدهد که روش ارائه شده در توابع چند قله ای از عملکرد بهتری در مقایسه با مدل *PSO* استاندارد، *PSO* با وزن اینرسی و *CLA_PSO* برخوردار است.

کلمات کلیدی: بهینه سازی گروه ذرات، اتوماتاهای یادگیر، بهینه سازی، اتوماتای یادگیر سلولی

Improving Particle Swarm Optimization using Open Cellular Learning Automata based

M. Hamidi^{2,3} M. R. Meybodi¹

¹Computer Engineering and Information Technology Department, Amirkabir University of Technology
Tehran, Iran

²Computer Engineering and Information Technology Department, Azad Islamic University Qazvin, Iran

³Electronic and Computer Engineering Department, Azad Islamic University, Zarghan, Iran
(mandana.hamidi@gmail.com, mmeybodi@aut.ac.ir)

Abstract: In this paper an improvement of particle swarm optimizers (PSOs) based on open cellular learning Automata (OCLA) called OCLAPSO is proposed. In OCLAPSO each search space dimension contains an OCLA, and each cell of OCLA contains one particle of that dimension. The learning automaton in each cell of OCLA is responsible for creating diversity of particles and also creating balance between the process of global and local search. Experiments are conducted on unimodal and multimodal test functions. The results of experimentations demonstrate the superiority of OCLPSO in solving multimodal problems when compared to PSO with inertia weight (PSOw), PSO with constrict factor (PSOcf) and CLA-PSO.

Keywords: Particle Swarm Optimization, Learning Automata, Optimization, Cellular Learning Automata

مقدمه

بهینه سازی گروه ذرات¹ PSO، یک تکنیک بهینه سازی مبتنی بر جمعیت است که از زندگی جانورانی که به صورت انبوه زندگی می کنند الهام گرفته است [1]. هر عضو از جمعیت یک راه حل مساله است که در فضای جستجو قرار دارد و به سوی راه حل نهایی حرکت می کند. یکی از معایب بهینه سازی گروه ذرات بخصوص در مسایل چند قله ای، قرار گرفتن ذرات در بهینه محلی است. جهت برطرف نمودن این مشکل الگوریتم هایی ارائه شده است که می توان به الگوریتم هایی جهت تنظیم پارامترهای معادله سرعت، (تنظیم این پارامترها به صورتهای تصادفی، کاهش خطی و غیرخطی [2] و سیستم فازی [3]) الگوریتم های مبتنی بر اجتماع که ساختارهای متفاوتی از توپولوژیهای اجتماعی برای

¹ Particle Swarm Optimization

همسایگی ذرات مطرح میکنند [4] [5]، الگوریتم های چندجمعیتی که شامل مجموعه ای از گروه ها می باشند که به صورت همکاری و یا رقابتی به تبادل اطلاعات می پردازند [7] [5]، الگوریتمهایی جهت ایجاد تعادل بین جستجوی محلی و جستجوی سراسری، الگوریتمهایی جهت ایجاد تنوع در بین ذرات برای جلوگیری از همگرایی زودرس [6] و در نهایت الگوریتمهای ترکیبی می باشند که از ترکیب الگوریتم بهینه سازی گروه ذرات و گونه ای مختلف از الگوریتم ها بوجود آمده اند، مانند ترکیب PSO با الگوریتم ژنتیک [8]، برنامه نویسی تکاملی، استراتژی های تکاملی، اتوماتای یادگیر و سیستم های فازی، اشاده نمود.

یک اتوماتای یادگیر^۲ [14] LA، ماشینی است که میتواند تعدادی متناهی عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی میشود و نتیجه ارزیابی در قالب سیگنالی مثبت یا منفی به اتوماتای یادگیر داده میشود و اتوماتا از این پاسخ در انتخاب عمل بعدی استفاده میکند و بدین ترتیب به سمت انتخاب عملی که بیشترین پاداش را از محیط میگیرد، میل میکند استفاده از اتوماتای یادگیر دارای دو مزیت عمده میباشد: اولاً میتوان از دانش موجود در تعیین روند تغییرات وزن میانی استفاده نمود و ثانیاً این روند با گرفتن بازخورد از اجرای الگوریتم اصلاح گردد. اتوماتای یادگیر برای بهبود قدرت یادگیری بسیاری از الگوریتمها مورد استفاده قرار گرفته است که از آن جمله میتوان به شبکه های عصبی و الگوریتمهای ژنتیک اشاره نمود. اتوماتای یادگیر سلولی CLA، مدلی برای سیستمهایی است که از اجزاء ساده ای تشکیل شده اند که از طریق تعامل با یکدیگر میتوانند رفتار پیچیده ای از خود نشان دهند. هر اتوماتای یادگیر سلولی، از یک اتوماتای سلولی تشکیل شده است که هر سلول آن به یک یا چند اتوماتای یادگیر مجهز می باشد که وضعیت این سلول را مشخص می سازد. یک قانون محلی در محیط حاکم است که تعیین کننده پاداش و یا جریمه عمل انتخاب شده توسط یک اتوماتای یادگیر است. اخیراً چهار مدل ترکیبی از PSO و اتوماتای یادگیر و اتوماتای یادگیر سلولی ارائه شده است که راندمان بالایی نسبت به PSO استاندارد دارند. در [13] مدل PSO باینری بر پایه اتوماتای یادگیر ارائه شده است. در این روش از یک اتوماتای یادگیر در هر بعد ذره استفاده شده است. هر کدام از این اتوماتاهای یادگیر دارای دو عمل ۰ و ۱ میباشد و به عنوان مغز متفکر ذره عمل میکند و حرکت آن را در فضای حالت محدود به ۰ و ۱ کنترل و راهبری مینماید. در [12] گونه جدیدی از PSO مبتنی بر CLA ارائه شده است که مانند الگوریتم PSO باینری بر پایه اتوماتای یادگیر بوده با این تفاوت که هر ذره از گروه به عنوان یک سلول از اتوماتای یادگیر سلولی در نظر گرفته شده است. در [10] مدل جدیدی به نام PSO_LA در فضای پیوسته پیشنهاد شده است که از یک اتوماتای یادگیر برای تنظیم رفتار ذرات استفاده شده است. اتوماتای یادگیر در هر گام تعیین میکند که ذرات به مسیر فعلی ادامه دهند و یا به دنباله روی از بهترین ذرات پیدا شده تاکنون بپردازند. در این مدل تمام ذرات یا به طور همزمان به جستجوی محلی و یا جستجوی سراسری در فضای جستجو می پردازند. در [11] الگوریتم CLA-PSO پیشنهاد شده است که ترکیبی از CLA و مدل بهینه سازی گروه ذرات میباشد. در این مدل، در هر سلول از اتوماتای یادگیر سلولی یک جمعیت از ذرات قرار دارد. اتوماتای یادگیر هر سلول، وظیفه تنظیم رفتار ذرات در آن سلول را بر عهده دارد که با استفاده از اتوماتای یادگیر در هر گام تعیین میگردد که ذرات به مسیر فعلی ادامه دهند و یا به دنباله روی از بهترین ذرات پیدا شده تاکنون بپردازند. برای تنظیم رفتار ذرات، اتوماتای یادگیر در هر سلول از تجربیات خود و تجربیات اتوماتاهای یادگیر در سلولهای همسایه استفاده میکند. علاوه بر محاسنی که این روش دارد، دارای چند ایراد عمده است

در این روش اگر ذرات یک سلول نسبت به سایر ذرات موجود در سلولهای دیگر دارای بهترین تجربه گروهی باشند، الگوریتم به اتوماتای یادگیر آن گروه یا سلول به گونه ای پاداش می دهد که در ذرات عمل دنباله روی تشویق گردند. از آنجایی که در انتخاب عمل دنباله روی وزن اینرسی صفر در نظر گرفته می شود، ذرات سریعاً جذب بهترین تجربه گروهی گشته و سرعتشان رفته رفته کاهش می یابد. در صورتیکه بهترین تجربه گروهی در بهینه محلی قرار داشته باشد، ذرات دیگر توانایی بیرون آمدن از بهینه محلی را ندارند. حتی اگر سلول دیگری به عنوان بهترین سلول انتخاب گردد و اتوماتای یادگیر اختصاص داده شده به عمل ادامه مسیر فعلی را انتخاب نماید، ذرات به علت ناچیز بودن سرعتشان دیگر یارای حرکت و جستجوی مکاشفه ای را ندارند.

جهت رفع این مشکل در این مقاله مدل جدیدی به نام OCLAPSO از ترکیب اتوماتای یادگیر سلولی و بهینه سازی گروه ذرات پیشنهاد شده است که در آن سعی شده است با استفاده از اتوماتاهای یادگیری سلولی که به هر بعد از فضای جستجو اختصاص داده می شود از تجمع ذرات در یک ناحیه ممانعت کرده و تنوع بین ذرات حفظ شود. در واقع در این مدل سعی شده است که از قرار گرفتن ذرات در مینیمم محلی حتی در یک بعد خاص نیز اجتناب نماید. نتایجی که بر روی ۸ تابع استاندارد آزمایش شده است نشان می دهد که این روش راندمان بالاتری نسبت به الگوریتم های PSO با وزن اینرسی، PSO با فاکتور انقباض و CLA-PSO دارد. ادامه این مقاله بدین صورت سازماندهی شده

² Learning Automata

است. بخش ۲ به معرفی اتوماتای یادگیر سلولی، اتوماتای سلولی و اتوماتای یادگیر می پردازد. در بخش سوم مدل پیشنهادی OCLAPSO^۳ ارائه میشود. بخش چهارم اختصاص به ارایه نتایج شبیه سازیها دارد و بخش پایانی نتیجه گیری میباشد

اتوماتای یادگیر سلولی CLA

اتوماتای یادگیر سلولی ترکیبی از اتوماتای یادگیر و اتوماتای سلولی می باشد. اتوماتای سلولی [15] یک مدل ریاضی برای سیستمهایی است که در آنها چندین مؤلفه ساده (سلول) برای تولید الگوهای پیچیده با هم همکاری می کنند. این سلولها به صورت همگام و در زمانهای گسسته بر طبق یک قانون محلی بهنگام رسانی می شوند. اتوماتای یادگیر ماشینی است که میتواند تعدادی عمل منتهای را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی میشود و نتیجه ارزیابی در قالب سیگنالی مثبت یا منفی به اتوماتا داده میشود و اتوماتا از این پاسخ در انتخاب عمل بعدی تاثیر میگیرد. هدف نهایی این است که اتوماتا یاد بگیرد تا از بین اعمال خود، عملی را که احتمال دریافت پاداش از محیط را به حداکثر می رساند انتخاب نماید. کارکرد اتوماتای یادگیر در تعامل با محیط، در شکل 1 مشاهده میشود. [14] اتوماتای یادگیر با ساختار متغیر را می توان توسط چهار تایی $\{\alpha, \beta, p, T\}$ نشان داد که α مجموعه عملهای اتوماتا، β مجموعه ورودیهای اتوماتا، p بردار احتمال انتخاب هریک از عملها و $T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می باشد. ورودی محیط یکی از r عمل انتخاب شده اتوماتا است. خروجی (پاسخ) محیط به هر عمل i توسط β_i مشخص می شود. اگر β_i یک پاسخ دودویی باشد، محیط مدل P نامیده می شود.



شکل 1: ارتباط بین اتوماتای یادگیر و محیط

الگوریتم زیر یک نمونه از الگوریتمهای یادگیری خطی است. فرض کنید عمل α_i در مرحله n ام انتخاب شود.

- پاسخ مطلوب

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

- پاسخ نامطلوب

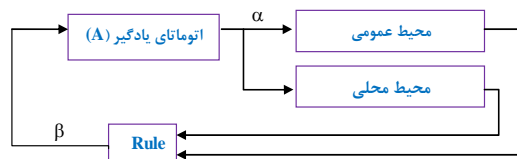
$$\begin{aligned} p_i(n+1) &= (1 - b)p_i(n) \\ p_j(n+1) &= (b/r - 1) + (1 - b)p_j(n) \quad \forall j \neq i \end{aligned} \quad (2)$$

در روابط (۱) و (۲)، a ، پارامتر پاداش و b پارامتر جریمه می باشند. با توجه به مقادیر a و b سه حالت زیر را می توان در نظر گرفت.

زمانیکه a و b با هم برابر باشند، الگوریتم را L_{RP} می نامیم، زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را L_{REP} می نامیم. و زمانیکه b مساوی صفر باشد الگوریتم را L_{RI} مینامیم [14]. قدرت اصلی اتوماتای یادگیر زمانی آشکار میشود که آنها به صورت دسته جمعی بکار روند. با توجه به این مساله و ضعفهای عنوان شده اتوماتای سلولی، در [15] با ترکیب این دو مدل، مدل اتوماتای یادگیر سلولی پیشنهاد گردید [16] [8]. در حالت کلی هر مجموعه مرتب از سلولها را می توان به عنوان همسایه در نظر گرفت. در هر لحظه هر اتوماتای یادگیر در اتوماتای یادگیر سلولی بر اساس مشاهدات قبلی و یا به صورت تصادفی یک عمل از مجموعه اعمال خود را انتخاب می کند، با توجه به اعمال انتخاب شده توسط سلولهای همسایه و قانون حاکم بر اتوماتای یادگیر سلولی، به عمل انتخاب شده پاداش داده و یا جریمه می شود. با توجه به پاداش و یا جریمه، اتوماتا رفتار خود را تصحیح کرده و ساختار داخلی اتوماتا بهنگام می گردد. بعد از بروزرسانی، هر اتوماتای یادگیر در اتوماتای یادگیر سلولی دوباره یک

³ Open Cellular Learning Automata

عمل از مجموعه اعمال خود را انتخاب کرده و انجام می‌دهد. فرآیند انتخاب عمل و دادن پاداش و یا جریمه تا زمانی که سیستم به حالت پایدار برسد و یا یک معیار از قبل تعریف شده برقرار شود، ادامه می‌یابد. اتوماتای یادگیر سلولی استاندارد، اصطلاحاً بسته خوانده می‌شود زیرا هیچگونه ارتباطی با دنیای خارج برای آن در نظر گرفته نشده است. اگر در اتوماتای سلولی یادگیر تمامی سلولها بطور همزمان بروز شود آن را اتوماتای یادگیر سلولی همگام و در غیر این صورت اتوماتای سلولی ناهمگام نامیده می‌شود [19]. نوع دیگر اتوماتای یادگیر سلولی، اتوماتای سلولی باز (OCLA) می‌باشد. در OCLA علاوه بر محیط محلی یک محیط سراسری نیز برای آن در نظر گرفته شده است (شکل ۲). در OCLA دادن جریمه و یا پاداش به عمل انتخاب شده توسط یک سلول علاوه بر اعمال انتخابی توسط همسایگانش به پاسخ محیط سراسری نیز بستگی دارد. در [16] اثبات شده است این مدل همانند CLA بسته، برای قوانین جابجایی‌پذیر، می‌تواند به نقاط بهینه محلی همگرا شود.



شکل 2: ارتباط بین یک اتوماتای یادگیر در اتوماتای سلولی باز با محیطهای محلی و سراسری

بهینه سازی گروه ذرات

بهینه سازی گروه ذرات الگوریتمی است که از رفتار اجتماعی موجوداتی که به صورت انبوه زندگی می‌کنند، الهام گرفته است. هر ذره در فضای D بعدی به عنوان یک راه حل از مساله محسوب می‌شود. ذره نام از گروه موقعیت X_i^d و سرعت V_i^d را در بعد dام از فضای جستجو دارد. معادله به روز نمودن سرعت و موقعیت ذره در (۳) و (۴) نمایش داده شده است.

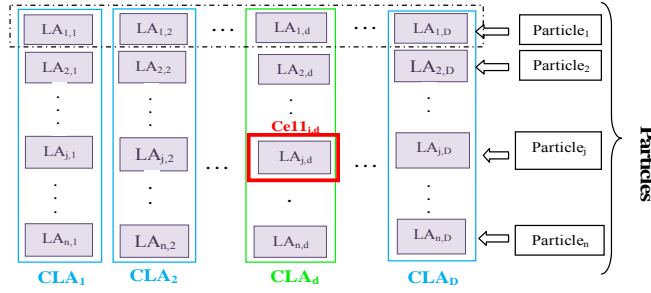
$$V_i^d(t+1) = wV_i^d(t) + c_1 rand1_i^d(t)(pbest_i^d(t) - X_i^d(t)) + c_2 rand2_i^d(t)(gbest(t) - X_i^d(t)) \quad (3)$$

$$X_i^d(t+1) = X_i^d(t) + V_i^d(t+1) \quad (4)$$

X_i موقعیت ذره نام و V_i سرعت ذره نام می‌باشند. بهترین موقعیت ملاقات شده توسط ذره نام $pbest_i$ و بهترین موقعیت ملاقات شده توسط کل گروه $gbest$ است. c_1, c_2 ضرایب مولفه های شناختی و اجتماعی می‌باشند. سرعت هر ذره توسط Vmax محدود شده است. $rand1_i^d$ و $rand2_i^d$ دو عدد تصادفی در بازه [0,1] می‌باشند.

مدل پیشنهادی OCLAPSO

این مدل ترکیبی از اتوماتای یادگیر سلولی باز و بهینه سازی گروه ذرات است. همانند مدل بهینه سازی گروه ذرات، دارای n ذره است که هر یک از ذرات دارای یک بردار موقعیت و یک بردار سرعت هستند. فضای جستجویی که ذرات در آن قرار دارند D بعدی است. به تعداد ابعاد فضای جستجو یک اتوماتای یادگیر سلولی وجود دارد. نحوه قرار گرفتن اتوماتاهای یادگیر سلولی در شکل ۳ نمایش داده شده است. در این مدل هر سلول اتوماتای یادگیر سلولی دارای یک بعد از یک ذره می‌باشد. به هر بعد از ذره یک اتوماتای یادگیر اختصاص داده شده است. بنابراین در هر سلول یک اتوماتای یادگیر وجود دارد. همسایگی که در بین سلولها در نظر گرفته شده است همسایگی مور می‌باشد. هر یک از اتوماتاهای یادگیر در هر سلول دارای دو عمل «دنباله روی» و «ادامه مسیر فعلی» می‌باشند. وظیفه تنظیم رفتار بعد خاصی از ذره به این معنی است که با استفاده از اتوماتای یادگیر در هر گام تعیین می‌گردد که ذرات به مسیر فعلی ادامه دهند و یا به دنباله روی از بهترین ذرات پیدا شده تا کنون بپردازند.



شکل ۳: ساختار مدل OCLAPSO

در صورت انتخاب عمل «دنباله روی»، تنها دنبال کردن بهترین تجربه شخصی و بهترین تجربه گروهی، در به روز نمودن سرعت یک بعد خاص از ذره مد نظر قرار خواهد گرفت. و از سرعت فعلی ذرات صرفنظر میشود. در صورت انتخاب عمل «ادامه مسیر فعلی»، در محاسبه سرعت جدید ذره، سرعت قبلی آن نیز در نظر گرفته می شود. اتوماتای یادگیر سلولی که در این مدل استفاده شده است از نوع اتوماتای یادگیر سلولی با متغیر سراسری میباشد. پاسخ سراسری که به تمام اتوماتاهای یادگیر داده می شود بدین نحو محاسبه می گردد که در صورتیکه موقعیت ذره در یک گام نسبت به موقعیت قبلی ذره بهبود یابد، عمل انتخاب شده توسط تمام اتوماتاهای یادگیر اختصاص داده شده به ابعاد ذره مثبت ارزیابی میشود و در غیر اینصورت عمل انتخاب شده منفی ارزیابی خواهد شد. این پاسخ سراسری βg نامیده می شود. پاسخ محلی یا βl که یک اتوماتای یادگیر از انتخاب عمل خود دریافت می نماید به صورت زیر محاسبه می گردد. جهت ارزیابی عمل انتخابی، واریانس موقعیت سلولهای همسایه محاسبه می شود. هر گاه این واریانس از حد خاصی کمتر باشد به این مفهوم است که ذرات در آن بعد خاص به یکدیگر نزدیکند و به سمت همگرایی پیش میروند، برای جلوگیری از همگرایی زود رس ذرات باید ذره به عمل دنباله روی ادامه ندهد. و ذرات باید به ادامه مسیر فعلی بپردازند. در چنین شرایطی اگر عمل انتخابی توسط اتوماتای یادگیر «دنباله روی» باشد، منفی ارزیابی می گردد و اگر عمل انتخابی «ادامه مسیر فعلی» باشد، مثبت ارزیابی میگردد. هر گاه واریانس بین ذرات از حد خاصی بیشتر باشد بدین مفهوم است که تنوع بین ذرات زیاد است و عمل انتخابی باعث همگرایی زودرس بین ذرات نمی گردد بدین جهت «عمل دنباله روی» مثبت ارزیابی می گردد و عمل «ادامه مسیر فعلی» منفی ارزیابی میگردد. یکی از مشکلات عمده در الگوریتم بهینه سازی گروه ذرات همگرایی زودرس است. در صورتیکه تنوع بین ذرات حتی در یک بعد خاص کاهش

الگوریتم این مدل پیشنهادی به صورت زیر است:

۱. ذرات با یک موقعیت و سرعت اولیه به صورت تصادفی در فضای جستجو قرار داده می شوند.
۲. به هر بعد از ذره یک اتوماتای یادگیر اختصاص داده می شود.
۳. بردار احتمالات انتخاب اعمال اتوماتاهای یادگیر هر ذره مقداردهی اولیه میشوند.
۴. تا زمانی که حداکثر تعداد گامها انجام گردد و یا هدف مورد نظر حاصل شود، مراحل ۵ تا ۱۰ تکرار میشوند:
۵. اتوماتای یادگیر مربوط به هر ذره یکی از اعمالش را بر طبق بردار احتمال اعمالش، انتخاب میکند.
۶. اگر اتوماتای یادگیر تخصیص داده شده به بعد d از ذره d ام عمل «دنباله روی» را انتخاب کرده باشد سرعت ذره به صورت فرمول (۵) به روز می شود (اینرسی سرعت ذره را صفر در نظر می گیریم).

$$\begin{aligned} V_i^d(t+1) &= c_1 \text{rand}_1(t)(pbest_i^d(t) - X_i^d(t)) + \\ &c_2 \text{rand}_2(t)(gbest_i^d(t) - X_i^d(t)) \quad d \in 1, \dots, N_d \end{aligned} \quad (5)$$

در صورت انتخاب عمل «ادامه مسیر فعلی»، سرعت جدید ذره به صورت معادله (۶) به روز می گردد.

$$\begin{aligned} w(t) &= w_0 - \frac{t * w_1}{MaxGen} \\ V_i^d(t+1) &= w(t) * V_i^d(t) + \\ &c_1 * \text{rand}_{i_1}^d * (pbest_i^d(t) - X_i^d(t)) + \\ &c_2 * \text{rand}_{i_2}^d * (gbest_i^d(t) - X_i^d(t)) \end{aligned} \quad (6)$$

جهت جلوگیری از هر گونه همگرایی زود رس در بین ذرات هنگامیکه سرعت ذرات در یک بعد خاص از یک حد آستانه ای کمتر شد و ذره شروع به ایستادن نمود دوباره سرعت ذره طبق رابطه (۷) مقدار دهی اولیه گردد.

$$V_{i+1}^d = \begin{cases} V_{i+1}^d & \text{if } |V_{i+1}^d| \geq Vc \\ u(-1,1)V \max / \rho & \text{if } |V_{i+1}^d| < Vc \end{cases} \quad (۷)$$

۷. پاسخ سراسری که اتوماتای یادگیر از محیط دریافت می نماید به صورت فرمول (۸) محاسبه می گردد

$$\beta_{g_i} = \begin{cases} 0 & \text{if } fitness(X_i(t+1)) < fitness(X_i(t)) \\ 1 & \text{otherwise} \end{cases} \quad (۸)$$

۸. پاسخ محلی که اتوماتای یادگیر از محیط دریافت می نماید به صورت فرمول (۹) محاسبه می گردد

$$\begin{aligned} & \text{if } variance(neighbor(X_i^d(t+1))) < vThreshold \\ & \quad \text{if } LA_i^d.action = 1 \\ & \quad \quad \beta_{l_i}^d = 1 \\ & \quad \text{else} \\ & \quad \quad \beta_{l_i}^d = 0; \\ & \text{else} \\ & \quad \text{if } LA_i^d.action = 1 \\ & \quad \quad \beta_{l_i}^d = 0 \\ & \quad \text{else} \\ & \quad \quad \beta_{l_i}^d = 1; \\ & \text{end} \end{aligned} \quad (۹)$$

۹. پاداش و یا جریمه اتوماتای یادگیر در هر بعد بر اساس پاسخ سراسری و محلی طبق فرمول (۱۰) ارزیابی می گردد.

$$\beta_i^d = \begin{cases} \beta_{l_i}^d & \text{if } \beta_{g_i} = 1 \\ 0 & \text{if } \beta_{g_i} = 0 \end{cases} \quad (۱۰)$$

۱۰. بردار احتمال انتخاب اعمال اتوماتای یادگیر اصلاح میشود.

نتایج شبیه سازیها

آزمایشات بر روی هشت تابع استاندارد صورت گرفته است که معمولا به عنوان معیار سنجش الگوریتمهای بهینه سازی مورد استفاده قرار میگیرند. توابع استفاده شده به همراه محدوده مقدار دهی اولیه ذرات و محدوده فضای جستجوی در جدول ۲ و جدول ۱ آورده شده است. می نیمم عمومی کلیه این توابع صفر می باشد.

جدول ۱: توابع تست

	Equation
sephere	$f_1(x) = \sum_{i=1}^D x_i^2$
Rosenbrock	$f_2(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1}^2)^2 + (x_i^2 - 1)^2)$
Ackley	$f_3(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$
Greiwank	$f_4(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$

Weierstrass	$f_5(x) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k 0.5)]$ $a = 0.5, b = 3, k_{\max} = 20.$
Rastrigin	$f_6(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$
NonContinuous Rastrigin	$f_7(x) = \sum_{i=1}^D (y_i^2 - 10 \cos(2\pi y_i) + 10),$ $y_i = \begin{cases} x_i & x_i < 1/2 \\ \text{round}(2x_i)/2 & x_i \geq 1/2 \end{cases}$ $\text{for } i = 1, 2, \dots, D$
Schwefel	$f_8(x) = 418.9829 * D - \sum_{i=1}^D x_i \sin\left(x_i ^{1/2}\right)$

الگوریتم پیشنهادی با الگوریتمهای PSO با وزن اینرسی (PSOw) و PSO با فاکتور انقباض (PSO cf) و CLA-PSO مقایسه شده است. هر کدام از این الگوریتمها پارامترهایی دارند که باید قبل از اجرای الگوریتم به صورت دستی تنظیم شوند در جدول 3 آورده شده است.

جدول 2: مقادیر اولیه توابع تست

F	X*	Search Range	Initialization Range
F1	[0,0,...,0]	[-100,100] ^D	[-100,50] ^D
F2	[1,1,...,1]	[-2.048, 2.048] ^D	[-2.048, 2.048] ^D
F3	[0,0,...,0]	[-32.76, 32.76] ^D	[-32.768, 16] ^D
F4	[0,0,...,0]	[-600, 600] ^D	[-600, 200] ^D
F5	[0,0,...,0]	[-0.5, 0.5] ^D	[-0.5, 0.2] ^D
F6	[0,0,...,0]	[-5.12, 5.12] ^D	[-5.12, 2] ^D
F7	[0,0,...,0]	[-5.12, 5.12] ^D	[-5.12, 2] ^D
F8	[420.96,... 420.96]	[-500.500] ^D	[-500.500] ^D

جدول 3: تنظیمات پارامترهای الگوریتمها

	W0	W1	C1,c2	V	LA_LRP	LA_LRP
PSO w	0.9	0.4	2			
PSO cf	0.75	0.75	1.49			
CLA-PSO	0.9	0.4	1.45		$\alpha=0.01$	$\alpha=0.001$ $\beta=0.01$
OCLAPSO	0.9	0.4	2	1e-50	$\alpha=0.01$	$\alpha=0.001$ $\beta=0.01$

دو نمونه آزمایش بر روی ۸ تابع مذکور صورت گرفته است. آزمایش اول بر روی ۸ تابع در ۱۰ بعد با ۱۰ ذره و ۱۰۰۰۰ تکرار صورت گرفته است که در جدول 4 نتیجه آن آورده شده است. (نتایج مندرج در جداول میانگین و واریانس ۳۰ بار اجرای الگوریتم می باشد). همانطور که از نتایج مشخص است. همانطور که از نتایج مشخص است. در توابع تک قله ای مانند تابع ۱ و ۲ الگوریتم PSOw و PSOcf سریعتر همگرا می شوند و دارای پاسخ بهتری نسبت به الگوریتمهای پیشنهادی می باشند. زیرا الگوریتمهای پیشنهادی محدوده بیشتری از فضای جستجو را مورد بررسی قرار می دهند. اما سایر توابع چند قله ای (توابع ۳، ۴، ۵، ۶ و ۷) الگوریتم OCLAPSO دارای راندمان بالاتری نسبت به سایر الگوریتمهای مقایسه شده است. اما در مورد تابع ۸ کمترین هیچ کدام از الگوریتمها پاسخ مناسبی برای این تابع نیافته اند. در کل الگوریتمهای پیشنهادی راندمان بالاتری در توابع چند قله ای دارند. بهترین موقعیت ملاقات شده بر حسب تکرار در الگوریتمهای مختلف برای ۸ تابع در شکل ۴ نمایش داده شده است. با توجه به این نمودارها منحنی الگوریتم PSOw و PSOcf در توابع ۱ و ۲ بسیار سریع همگرا می شود. اما در توابع چند قله ای منحنی این الگوریتمها ثابت می ماند که مشخص کننده قرار گرفتن ذرات در مینیمم محلی است. در توابع چند قله ای منحنی الگوریتمهای پیشنهادی سیر نزولی دارند و ثابت نمی ماند که نشان دهنده قرار نگرفتن این ذرات در مینیمم محلی

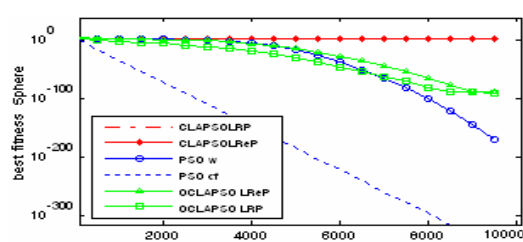
است. آزمایش دوم بر روی ۸ تابع در ۳۰ بعد با ۲۰ ذره و ۳۰۰۰۰ تکرار صورت گرفته است که در جدول ۵ نتیجه آن آورده شده است. با توجه به جدول می توان متوجه شد که نتایج همانند نتایج موجود در آزمایش با ۱۰ بعد می باشند. با این تفاوت که رسیدن به پاسخ بهینه در کلیه الگوریتم ها به علت بعد بالا کند تر می باشد. در توابع چند قله ای راندمان الگوریتم های پیشنهادی نسبت به سایر الگوریتم های اولیه PSOw, PSOcf, و CLA-PSO بالاتر می باشد. بهترین موقعیت ملاقات شده بر حسب تکرار در الگوریتم های مختلف برای ۸ تابع در شکل ۵ نمایش داده شده است. با توجه به این نمودارها در توابع چند قله ای منحنی این الگوریتم ها سیر نزولی دارند و ثابت نمی مانند که نشان دهنده فرار این ذرات در مینیمم محلی است.

نتیجه گیری

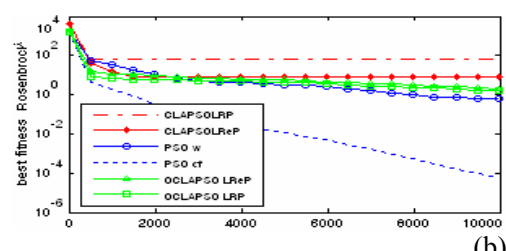
در این مقاله الگوریتم بهینه سازی جدیدی به نام OCLAPSO ارائه شده است که ترکیبی از اتوماتای یادگیر سلولی و بهینه سازی گروه ذرات است. در این مدل تلاش بر ایجاد تنوع در بین ذرات است تا ذرات در همگرایی زودرس قرار نگیرند. در این مدل به ازای هر بعد از فضای جستجو یک اتوماتای یادگیر سلولی در نظر گرفته شده که هر سلول آن یک بعد از ذره می باشد. هر سلول از اتوماتای یادگیر سلولی مجهز به یک اتوماتای یادگیر است. اتوماتای یادگیر موجود در هر بعد از ذرات به عنوان مغز آن بعد عمل می نماید و وظیفه آن جلوگیری از قرار گرفتن آن بعد خاص از ذره می باشد. بدین نحو که هر سلول به سلولهای اطراف خود نگاه میکند. اگر موقعیت ذرات در سلولهای همجوار به موقعیت ذره بسیار نزدیک باشد اتوماتای یادگیر آن سلول، به گونه ای پاداش داده می شود که از ذرات موجود در سلولهای همسایه فاصله بگیرد و به جستجوی مکاشفهای بپردازد. با توجه به نتایج OCLAPSO نسبت به الگوریتم PSO با وزن اینرسی (PSOw) و PSO با فاکتور انقباض (PSO cf) و CLA-PSO در توابع چند قله ای سریعتر همگرا می شود.

جدول ۴: مقایسه الگوریتم پیشنهادی و سایر الگوریتم ها در ۸ تابع ۱۰ بعدی

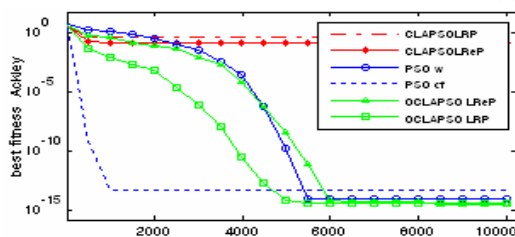
	F1	F2	F3	F4
CLA PSO LRP	10.6405±43.8358	69.7548±5.0995e+003	0.3907±0.0618	0.3437±0.2561
CLA PSO LReP	5.3374±5.3085	8.0874±15.4584	0.1371±0.0087	0.9247±0.3847
PSOw	7.2191e-196±1e-205	0.6376±0.4400	9.23715e-015±2.2719e-029	0.0964±5.4310e-004
PSO cf	0±0	5.7089e-005±3.1120e-009	4.6896e-014±4.4012e-027	0.1255±0.0021
OCLA PSO LReP	2.7695e-091±8.4184e-182	2.0036±0.1788	3.5527e-015±0	0.068±3.9992e-004
OCLA PSO LRP	5.3799e-092±2.7713e-183	1.5181±0.5399	4.737e-015±4.2073e-030	0.0607±0.0014
	F5	F6	F7	F8
PSO cf	0.0023±6.0964e-006	0.3792±0.12	8.5156±16.6647	4152.2213±0.1074
PSO w	0.017±8.2864e-004	1.8675±4.6386	14.0509±60.2519	4154.9992±29.7794
CLA PSO LRP	0.0081±2.5273e-004	0±0	0.4±0.3000	4150.6897±0.0039
CLA PSO LReP	0.7241±0.4802	0.398±0.2970	15.8±47.2000	4142.3208±121.6576
OCLA PSO LRP	2.6265e-006±2.0663e-011	0±0	0±0	4151.0751±0.0062
OCLA PSO LReP	1.4305e-005±2.9563e-010	0±0	0±0	4150.6312±0.0048



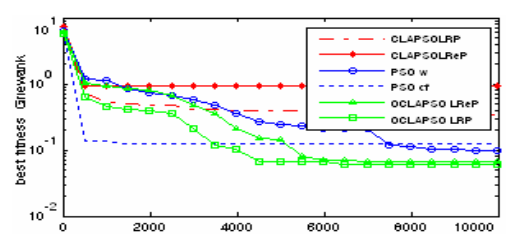
(a)



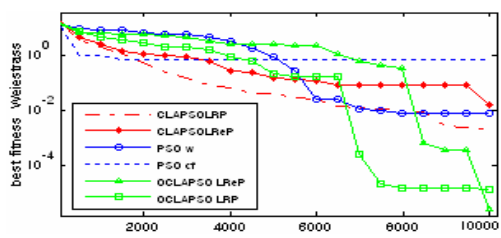
(b)



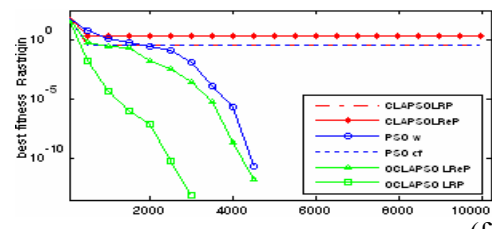
(c)



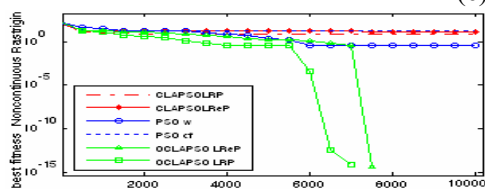
(d)



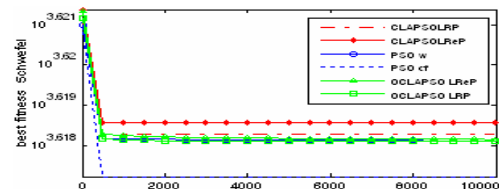
(e)



(f)



(g)

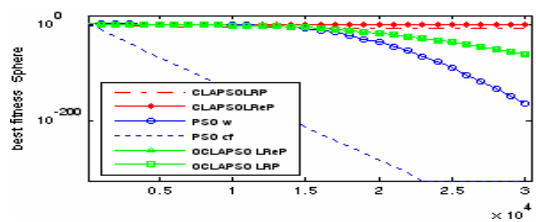


(h)

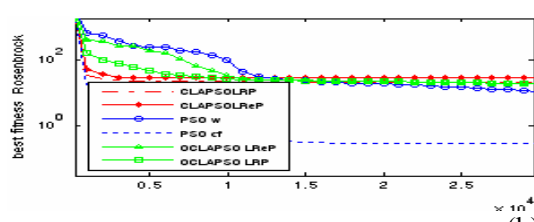
شکل ۴: میانگین بهترین تابع شایستگی ذرات نسبت به تعداد تکرارهای الگوریتم در ۱۰ بعد، ۱۰۰۰۰ تکرار ۸ تابع (a) Sphere، (b) Rosenbrock، (c) Ackley، (d) Greiwank، (e) Weierstrass، (f) Rastrigin، (g) Noncontinuous Rastrigin و (h) Schwefel

جدول ۵: مقایسه الگوریتم پیشنهادی و سایر الگوریتم ها در ۸ تابع ۳۰ بعدی

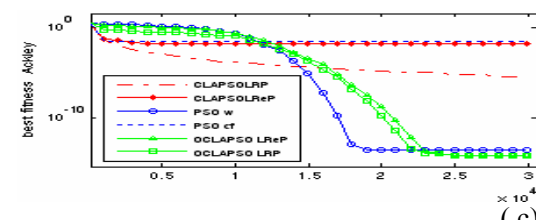
	F1	F2	F3	F4
CLA PSO LRP	10.6405±43.8358	69.7548±5.0995e+003	0.3907±0.0618	0.3437±0.2561
CLA PSO LReP	5.3374±5.3085	8.0874±15.4584	0.1371±0.0087	0.9247±0.3847
PSOw	7.2191e-196±1e-205	0.6376±0.4400	9.23715e-015±2.2719e-029	0.0964±5.4310e-004
PSO cf	0±0	5.7089e-005±3.1120e-009	4.6896e-014±4.4012e-027	0.1255±0.0021
OCLA_PSO LReP	2.7695e-091±8.4184e-182	2.0036±0.1788	3.5527e-015±0	0.068±3.9992e-004
OCLA_PSO LRP	5.3799e-092±2.7713e-183	1.5181±0.5399	4.737e-015±4.2073e-030	0.0607±0.0014
	F5	F6	F7	F8
CLA PSO LRP	0.0023±6.0964e-006	0.3792±0.12	8.5156±16.6647	4152.2213±0.1074
CLA PSO LReP	0.017±8.2864e-004	1.8675±4.6386	14.0509±60.2519	4154.9992±29.7794
PSOw	0.0081±2.5273e-004	0±0	0.4±0.3000	4150.6897±0.0039
PSO cf	0.7241±0.4802	0.398±0.2970	15.8±47.2000	4142.3208±121.6576
OCLA_PSO LReP	2.6265e-006±2.0663e-011	0±0	0±0	4151.0751±0.0062
OCLA_PSO LRP	1.4305e-005±2.9563e-010	0±0	0±0	4150.6312±0.0048



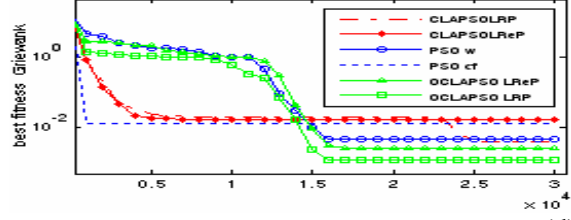
(a)



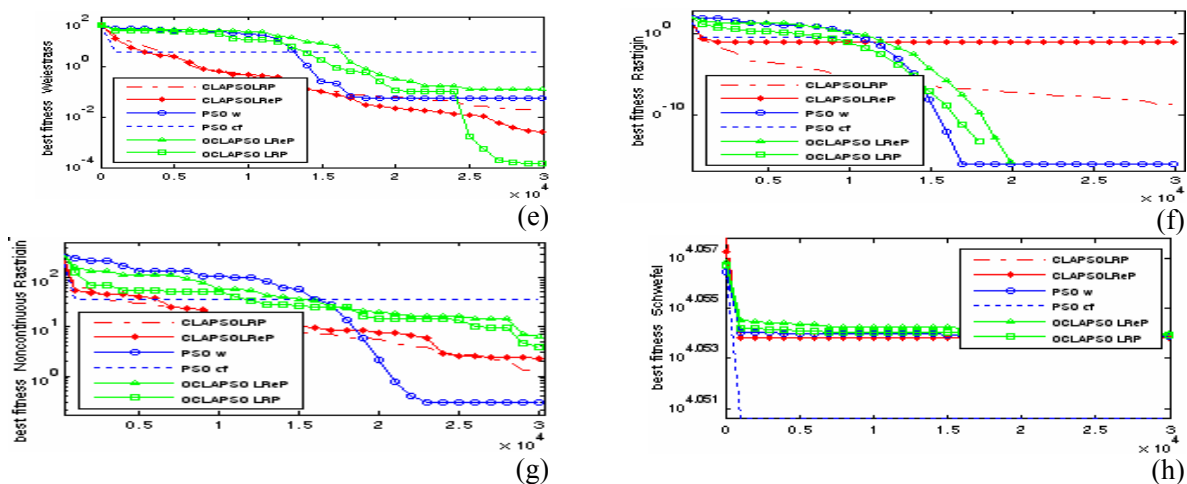
(b)



(c)



(d)



شکل ۵: میانگین بهترین تابع شایستگی ذرات نسبت به تعداد تکرارهای الگوریتم در ۱۰ تکرار، ۵ ذره، ۱۰۰۰۰ تکرار ۸ تابع (a) Sphere، (b) Rosenbrock، (c) Ackley، (d) Greiwank، (e) Weierstrass، (f) Rastrigin، (g) Noncontinuous Rastrigin، (h) Schwefel،

منابع

- [1] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization", Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942-1948, 1995.
- [2] A. Ratnaweera, S. Halgamuge, and H. Watson, "Particle Swarm Optimization with Self-Adaptive Acceleration Coefficients, Proceedings of the First International Conference on Fuzzy Systems and Knowledge Discovery, pages 2411-2418 December 2003.
- [3] Y. Zheng, L. Ma, L. Zhang, and Qian, "Emperical Study of Particle Swarm Optimizer with Increasing Inertia Weight", Proceeding of IEEE Congress on Evolutionary Computation, Pages 221-226. IEEE Press, 2003
- [4] J. Kennedy, "Small Worlds and Mega-Minds: Effects of Neighborhood Topology on Particle Swarm Performance Proceeding s of The IEEE Congress on Evolutionary Computation , vol. 3, pp. 1931-1938 , July 1999.
- [5] F. Van den Bergh and A.P. Engelbrecht, "Effects of Swarm Size on Cooperative Particle Swarm Optimizers", Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco, USA, pp. 899-893, 2001.
- [6] A. Silva, A. Neves, and E. Costa, "An Empirical Comparison of Particle Swarm and Predator Prey Optimization", Proceedings of the Thirteenth Irish Conference on Artificial Intelligence and Cognitive Science , Lecture Notes in Artificial Intelligence , volume 2464, pages 103-110 , Springer Verlag, 2002.
- [7] J.S. Vesterstrom, J. Giget, and T. Krink, "Division of Labor in Particle Swarm Optimization", Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1570-1575 IEEE press, 2002.
- [8] M. Løvberg, T. Rasmussen and T. Krink, "Hybrid Particle Swarm Optimizer with Breeding and Subpopulation", Proceedings of the Third Genetic and Evolutionary Computation Conference, vol. 1, pp. 476-469, 2001.
- [9] H. Beigy and M. R. Meybodi, "A Mathematical Framework for Cellular Learning Automata", Advances on Complex Systems, Vol. 7, Nos. 3-4, pp. 295-320, 2004.
- [10] M. Sheybani, and M.R. Meybodi, "PSO-LA: A New Model for Optimization", Proceedings of 12th Annual CSI Computer Conference of Iran, Shahid Beheshti University, Tehran, Iran, pp. 1162-1169, Feb. 20-22, 2007.
- [11] M. Sheybani and M. R. Meybodi, "CLAPSO: A New Model for Optimization", Proceedings of 15th Conference on Electrical Engineering (15th ICEE), Volume on Computer, Telecommunication Research Center, Tehran, Iran, May 17, 2007
- [12] N. Jafarpour and M.R. Meybodi, "A Hybrid Method for Optimization (Discrete PSO + CLA)", Proceedings of International Conference on Intelligence and Advance Systems (ICIAS2007), Kuala Lumpor, Malaysia, Nov. 25-28, 2007
- [13] R. Rastegar, M. R. Meybodi and K. Badie, "A New Discrete Binary Particle Swarm Optimization based on Learning Automata", Proceedings of International Conference on Machine Learning and Applications (ICMLA2004), 16-18 December 2004, pp. 456-462, USA, IEEE Press., 2004.
- [14] K. S. Narendra and M. A. L. Thathachar., Learning Automata: An Introduction, Prentice-Hall Inc, 1989
- [15] S. Wolfram, "Cellular Automata", Los Alamos Science, vol. 9, pp. 2-21, Fall 1983.
- [16] M. R. Meybodi and H. Beigy, "A Note on Learning Automata Based Schemes for Adaptation of BP Parameters", Journal of Neurocomputing, Vol. 48, No. 4, pp. 957-974, October 2005.
- [17] H. Beigy, and M. R. Meybodi, "Open Synchronous Cellular Learning Automata", Proceedings of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI2004), pp. 9-15, Orlando, Florida, USA. July 18-21, 2004.

- [18] H. Beigy and M. R. Meybodi, "*Asynchronous Cellular Learning Automata*", Automatica, Journal of International Federation of Automatic Control, 2007, *Vol. 44, No. 5, May 2008*.
- [19] H. Beigy and M. R. Meybodi, "A Learning Automata Based Algorithm for Determination of Minimum Number of Hidden Units for Three Layers Neural Networks", Journal of Amirkabir, Vol. 12, No. 46, pp. 111-136, 2001.