

مدل جدیدی از بهینه سازی گروه ذرات مبتنی بر اتوماتای یادگیر LAPSO

ماندانا حمیدی^۱ و محمد رضا میبیدی^۲

چکیده

بهینه سازی گروه ذرات نمونه ای از الگوریتم های بهینه سازی است که از زندگی جانورانی که به صورت انبوه زندگی می کنند الهام گرفته است. مشکل این روش قرار گرفتن در بهینه های محلی مخصوصا در مسائل چند قله ای می باشد. یکی از الگوریتم هایی که جهت بهبود این مشکل ارائه شده است PSO_LA نام دارد که از یک اتوماتای یادگیر جهت تنظیم سرعت ذرات گروه استفاده نموده است. اتوماتای یادگیر دارای دو عمل «دنباله روی» و «ادامه مسیر فعلی» می باشد، که مشخص می نماید ذرات گروه به جستجوی محلی یا جستجوی عمومی بپردازند. این الگوریتم علاوه بر مزایایی که دارد دارای دو عیب عمده است اول اینکه رفتار هر ذره در گروه تابع عملی است که اتوماتای یادگیر انتخاب می نماید و یک ذره با روند حرکت مناسب نیز باید تابع گروه باشد و دوم اینکه الگوریتم توانایی بیرون آوردن ذرات را از بهینه های محلی را ندارد. جهت رفع این مشکل در این مقاله نمونه های جدیدی از الگوریتم بهینه سازی گروه ذرات مبتنی بر اتوماتاهای یادگیر پیشنهاد شده است. که از اتوماتای های یادگیر به عنوان عاملی هوشمند در تنظیم سرعت ذره استفاده شده است. در این الگوریتم ها به هر ذره از گروه یک اتوماتای یادگیر اختصاص داده می شود که هر اتوماتا به عنوان مغز ذره می باشد و حرکت ذره را در فضای جستجو رهبری می نمایند. الگوریتم های پیشنهادی بر روی هشت تابع استاندارد آزمایش شده اند و نتایج نشان می دهند که الگوریتم های پیشنهادی نسبت به الگوریتم های PSO استاندارد و PSO_LA راندمان بالاتری دارند.

کلمات کلیدی

بهینه سازی گروه ذرات، اتوماتاهای یادگیر، بهینه سازی

New Learning Automata based Particle Swarm Optimization Algorithms

Mandana Hamidi

Mohammad Reza Meybodi

Abstract

Particle swarm optimization (PSO) is a population based statistical optimization technique which is inspired by social behavior of bird flocking or fish schooling. The main weakness of PSO especially in multimodal problems is trapping in local minima. Recently a learning automata based PSO called PSO-LA to improve the performance of PSO has been reported. PSO-LA uses one learning automaton for configuring the behavior of particles and also creating a balance between the process of global and local search. Although PSO-LA produces better results than the standard PSO but like standard PSO it may trap into local minima. In this paper four improvements on PSO-LA are proposed. These improvements are proposed to reduce the probability of trapping PSO-LA into local minima. Unlike PSO-LA which uses one learning automaton to guide all particles, in the proposed PSO algorithms one learning automaton is assigned to each particle as the particle brain which controls the particle movement in the search space. The proposed algorithms are tested on 8 benchmark functions. The results have shown that the proposed PSO algorithms are superior to standard PSO, PSO with inertia weight (PSOw) and previously reported LA based PSO algorithms.

Keywords

Particle Swarm, Learning Automata, Optimization

۱. مقدمه

یکی از روشهای بهینه سازی که اخیرا توجه محققان را به خود جلب نموده است، الگوریتم بهینه سازی گروه ذرات می باشد که از رفتار اجتماعی موجوداتی که به صورت گروهی زندگی می کنند، مانند ماهی ها، زنبورها و پرندگان الهام گرفته است. این مدل اولین بار توسط کندی و

^۱ عضو هیأت علمی دانشکده مهندسی کامپیوتر دانشگاه آزاد اسلامی واحد زرقان mandana.hamidi@gmail.com

^۲ عضو هیأت علمی دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر، meybodi@ce.aut.ac.ir

ابرهارت در سال ۱۹۹۵ ارائه شد [1] سرعت همگرایی و سادگی قوانین در تعیین جهت بهینه از خصوصیات مهم این روش به شمار می آید این الگوریتم زیر مجموعه الگوریتم‌های تکاملی بوده و برپایه جمعیت بنا شده است. عیب عمده این الگوریتم قرار گرفتن ذرات در مینیمم محلی می باشد. جهت رفع این مشکل گونه های متفاوتی از این الگوریتم ارائه شده است. الگوریتم‌های بهینه‌سازی گروه ذرات را می‌توان به شش دسته بهینه‌سازی گروه ذرات تک هدفی در فضای پیوسته، [1] Niching PSO، [2]، بهینه‌سازی گروه ذرات مقید و محدود شده، Multi Objective PSO [3]، بهینه‌سازی گروه ذرات با محیط متغیر، بهینه‌سازی در فضای جستجوی پویا [4] و بهینه‌سازی گروه ذرات در فضای گسسته [6] [5] تقسیم نمود. گونه های متنوعی از بهینه‌سازی گروه ذرات تک هدفی وجود دارد که می‌توان آنها را به گروه‌های زیر تقسیم نمود: الگوریتم‌هایی جهت تنظیم پارامترهای معادله سرعت مانند وزن اینرسی، مولفه‌های شناختی و اجتماعی بهینه‌سازی گروه ذرات [7] [6]، الگوریتم‌های مبتنی بر اجتماع [11]، الگوریتم‌های چند جمعیتی [12] [12] الگوریتم‌هایی جهت ایجاد تعادل بین جستجوی محلی و سراسری [14] [15]، الگوریتم‌هایی با ایجاد تنوع در بین ذرات [13] و در نهایت الگوریتم‌های ترکیبی که از ترکیب الگوریتم بهینه‌سازی گروه ذرات و گونه ای مختلف از الگوریتم‌ها بوجود آمده اند. به عنوان مثال ترکیب PSO با الگوریتم ژنتیک [16]، برنامه نویسی تکاملی، استراتژی‌های تکاملی، سیستم های فازی [17]، اتوماتای یادگیر و اتوماتای یادگیر سلولی. [22]

یک اتوماتای یادگیر [25]، 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. 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 و بهینه‌سازی گروه ذرات اشاره نمود. در [24] از ترکیب اتوماتای یادگیر و بهینه‌سازی گروه ذرات باینری مدل جدیدی ارائه شده است که در هر بعد از ذره از یک اتوماتای یادگیر استفاده شده است. این مدل نتایج بهتری در مقایسه با مدل باینری استاندارد [5] در حل مسایل نمونه داشته است. در [21] مدل جدیدی به نام PSO_LA 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 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. در این مدل تمام ذرات یا به طور همزمان به جستجوی محلی و یا جستجوی سراسری در فضای جستجو می پردازند. مشکل این روش قرار گرفتن در مینیمم محلی است. در زیر به دلایل قرار گرفتن ذرات در مینیمم محلی می پردازیم:

۱- اگر موقعیت جدید یک ذره نسبت به موقعیت قبلی ذره بهتر باشد، عملاً حرکت آن ذره در فضای جستجو مناسب است و نیازی نیست که روند جستجوی ذره را تغییر داد. بر عکس اگر ذره‌ای در طی چند تکرار موقعیتش بهبود نیابد، بهتر است که روند جستجوی آن ذره تغییر یابد در PSO_LA اتوماتای یادگیر با انتخاب یکی از اعمال خود روند جستجوی کل گروه را تغییر می دهد بنابراین ذراتی که جستجوی خوبی داشته‌اند نیز باید تابع گروه بوده و حرکت خود را تغییر دهند و این امر باعث از بین رفتن اطلاعات مفید ذره می گردد.

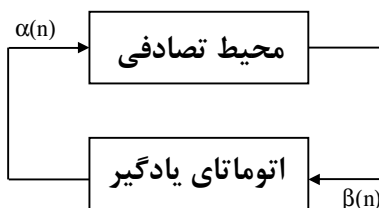
۲- یکی از مشکلات دیگر PSO_LA این است که اگر اتوماتای یادگیر عمل دنباله روی را انتخاب نماید و دائم پاداش بگیرد به جستجوی محلی خود ادامه می دهد و اگر در مینیمم محلی قرار گیرد سرعت ذره صفر می شود، در صورتیکه اتوماتای یادگیر عمل ادامه مسیر فعلی را انتخاب نماید چون سرعت صفر است ذره دیگر حرکتی ندارد و در همان نقطه ثابت می ماند

جهت بهبود مشکلات مذکور در این مقاله الگوریتم جدید بهینه‌سازی گروه ذرات مبتنی بر اتوماتای یادگیر به نامهای LAPSO1، LAPSO2، LAPSO3 و LAPSO4 ارائه شده است که در LAPSO1 همانند روش PSO_LA یک اتوماتای یادگیر برای کل گروه در نظر گرفته می شود، با این تفاوت که عمل « ادامه مسیر فعلی » تغییر داده شده است. در سه روش پیشنهادی دیگر به هر ذره از گروه یک اتوماتای یادگیر اختصاص داده می شود تا روند حرکت ذره را تنظیم نماید. اتوماتای یادگیر اختصاص داده شده به هر ذره در هر گام تعیین میکند که ذرات به مسیر فعلی ادامه دهند و یا به دنباله روی از بهترین ذرات پیدا شده تاکنون بپردازند. این روش نسبت به روش PSO_LA یک مزیت دارد و آن این است که اطلاعات مفید ذرات از بین نمی رود و تنوع ذرات افزایش یافته بنابراین امکان قرار گرفتن ذره در مینیمم محلی کمتر می باشد. جهت جلوگیری از بی حرکت ماندن ذرات، هنگامیکه سرعت ذرات از یک حد آستانه‌ای بیشتر باشد، سرعت ذره دوباره مقدار دهی می شود. 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 PSO-LA دارد.

ادامه این گزارش به صورت زیر سازماندهی شده است. در بخش دوم اتوماتای یادگیر، در بخش سوم بهینه‌سازی گروه ذرات و در بخش چهارم الگوریتم PSO_LA ارائه شده اند. در بخش پنجم چهار مدل پیشنهادی شرح داده شده است. بخش ششم نتایج شبیه سازیها بر روی هشت تابع استاندارد ارائه شده است و در نهایت در میکند و بخش پایانی نتیجه گیری میباشد.

۲. اتوماتای یادگیر^۳

اتوماتای یادگیر، 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 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 بهترین عمل را انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداکثر برساند. کارکرد اتوماتای یادگیر در تعامل با محیط، در شکل (۱) مشاهده میشود.



شکل (۱) ارتباط بین اتوماتای یادگیر و محیط

محیط را می‌توان توسط سه‌تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودیها، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه خروجیها و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمالاتی جریمه می‌باشد. هرگاه β 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 از نوع Q، $\beta(n)$ می‌تواند به طور گسسته یک مقدار از مقادیر محدود در فاصله $[0, 1]$ و در محیط از نوع S، $\beta(n)$ متغیر تصادفی در فاصله $[0, 1]$ است. c_i احتمال اینکه عمل α_i نتیجه نامطلوب داشته باشد می‌باشد. در محیط ایستا مقادیر c_i بدون تغییر می‌مانند، حال آنکه در محیط غیر ایستا این مقادیر در طی زمان تغییر می‌کنند. اتوماتای یادگیر با ساختار متغیر را میتوان توسط چهارتایی $\{\alpha, \beta, p, T\}$ نشان داد که $\alpha = \{\alpha_1, \dots, \alpha_r\}$ مجموعه عملهای اتوماتا، $\beta = \{\beta_1, \dots, \beta_m\}$ مجموعه ورودیهای اتوماتا، $p = \{p_1, \dots, p_r\}$ بردار احتمال انتخاب هریک از عملها و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می‌باشد. الگوریتم زیریک نمونه از الگوریتمهای یادگیری خطی است. فرض کنید عمل α_i در مرحله n ام انتخاب شود - پاسخ مطلوب

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1 - a)p_j(n) \quad \forall j \quad j \neq i \end{aligned} \quad (۱)$$

- پاسخ نامطلوب

$$\begin{aligned} p_i(n+1) &= (1 - b)p_i(n) \\ p_j(n+1) &= (b/r - 1) + (1 - b)p_j(n) \quad \forall j \quad j \neq i \end{aligned} \quad (۲)$$

در روابط (۱) و (۲) پارامتر پاداش و b پارامتر جریمه می‌باشند. با توجه به مقادیر a و b سه حالت زیر 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. و زمانیکه b مساوی صفر باشد الگوریتم را L_{RI} می‌نامیم [25]

۳. بهینه‌سازی گروه ذرات^۴

بهینه‌سازی 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 d ام از فضای جستجو دارد. معادله به روز نمودن سرعت و موقعیت ذره در (۳) و (۴) نمایش داده شده است.

$$\begin{aligned} V_i^d(t+1) &= wV_i^d(t) + c_1 \text{rand}_1^d(t)(pbest_i^d(t) - X_i^d(t)) \\ &\quad + c_2 \text{rand}_2^d(t)(gbest(t) - X_i^d(t)) \end{aligned} \quad (۳)$$

$$X_i^d(t+1) = X_i^d(t) + V_i^d(t+1) \quad (۴)$$

$X_i = (X_i^1, X_i^2, \dots, X_i^D)$ موقعیت ذره نام و $V_i = (V_i^1, V_i^2, \dots, V_i^D)$ سرعت ذره نام می‌باشند. بهترین موقعیت ملاقات شده توسط ذره نام $pbest_i = (pbest_i^1, pbest_i^2, \dots, pbest_i^D)$ و گروه کل گروه $gbest = (gbest^1, gbest^2, \dots, gbest^D)$ است.

³ Learning Automata

⁴Particle swarm Optimization

c_1, c_2 ضرایب مولفه های شناختی و اجتماعی می باشند. سرعت هر ذره توسط V_{max} محدود شده است. $rand_1^d$ و $rand_2^d$ دو عدد تصادفی در بازه $[0,1]$ می باشند.

۴. الگوریتم PSO_LA

PSO_LA الگوریتم بهینه سازی گروه ذرات می باشد که بر پایه اتوماتای یادگیر بنا شده است که از یک اتوماتای یادگیر برای کنترل رفتار ذرات استفاده میکند. این اتوماتای یادگیر دارای دو عمل «دنباله روی» و «ادامه مسیر فعلی» میباشد. در ابتدا، موقعیت و سرعت ذرات و همچنین بردار احتمالات انتخاب اعمال اتوماتای یادگیر مقداردهی اولیه میشوند. سپس تا زمانیکه حداکثر تعداد گامها انجام گردد و یا هدف مورد نظر حاصل شود، مراحل زیر تکرار میشوند:

- ۱- اتوماتای یادگیر یکی از اعمالش را بر طبق بردار احتمال اعمالش، انتخاب میکند.
- ۲- با توجه به عمل انتخاب شده، نحوه بروزرسانی سرعت ذرات تعیین میشود و سپس ذرات، سرعت و موقعیت خود را بروز میکنند.
- ۳- بر اساس نتایج به روزرسانی موقعیت ذرات، عمل اتوماتای یادگیر، ارزیابی بردار احتمال انتخاب اعمال اتوماتای یادگیر اصلاح میشود. عملی که اتوماتای یادگیر در هر گام برمیزگزيند، تعیین کننده شیوه بروز کردن سرعت ذرات در آن گام میباشد. در صورت انتخاب عمل «دنباله روی»، تنها دنبال کردن بهترین تجربه شخصی و بهترین تجربه گروهی، در به روز نمودن سرعت ذرات مد نظر قرار خواهند گرفت و از سرعت فعلی ذرات صرفنظر میشود که در این صورت بروزرسانی سرعت ذرات مطابق با فرمول (۵) انجام میگردد. در صورت انتخاب عمل «ادامه مسیر فعلی»، سرعت جدید ذرات برابر با سرعت فعلی آنها خواهد بود و ذره همچنان مسیری فعلی را ادامه خواهد داد.

$$V_i^d(t+1) = c_1 rand_1^d(t)(pbest_i^d(t) - X_i^d(t)) + c_2 rand_2^d(t)(gbest(t) - X_i^d(t)) \quad (5)$$

شیوه ارزیابی عمل انتخاب شده، به این صورت است که موقعیت فعلی هر ذره با موقعیت قبلی آن مقایسه میشود. چنانچه c_{imp} درصد از جمعیت موقعیتشان بهبود یافته باشد، عمل انتخاب شده مثبت و در غیر اینصورت منفی ارزیابی میشود. c_{imp} از پارامترهای الگوریتم پیشنهادی است که بایستی با توجه به نوع مساله و اتوماتای یادگیر بکار گرفته شده، تنظیم گردد.

۵. مدل های پیشنهادی

در این قسمت چهار الگوریتم جدید که بهبود یافته الگوریتم PSOLA می باشد ارائه شده است

۱.۵.۱ LAPS01

اولین مدل پیشنهادی LAPS01 است که ترکیبی از اتوماتای یادگیر و بهینه سازی گروه ذرات می باشد. در این مدل همانند الگوریتم PSO_LA از یک اتوماتای یادگیر جهت کنترل رفتار ذرات استفاده می کند. تفاوت این روش با الگوریتم PSO_LA در این است که در صورتیکه اتوماتای یادگیر عمل «دنباله روی» را انتخاب نماید سرعت اولیه ذره در محاسبه سرعت جدید ذره نادیده گرفته می شود، در واقع وزن اینرسی ذره برابر با صفر قرار داده میشود بدین نحو ذرات به جستجوی محلی می پردازند. در صورتیکه اتوماتای یادگیر عمل «ادامه مسیر فعلی» را انتخاب نماید، جهت محاسبه سرعت جدید ذره از سرعت قبلی ذره با یک وزن اینرسی خاص نیز استفاده می شود و ذرات به جستجوی عمومی در فضای جستجو می پردازند. در PSO_LA هنگامیکه اتوماتون یادگیر عمل «ادامه مسیر فعلی» را انتخاب نماید سرعت ذرات کماکان تغییری نمی کند. بنابراین دامنه نوسانات ذره زیاد است و ذرات فقط به جستجوی عمومی می پردازند و هنگامیکه عمل «دنباله روی» را انتخاب نمایند، خیلی سریع همگرا می شوند. در LAPS01 سعی شده است که هنگامیکه اتوماتای یادگیر عمل ادامه مسیر فعلی را انتخاب نمود، سرعت ذره را به گونه ای کاهش داد که نوسانات ذره به مرور زمان کاهش یابد. الگوریتم LAPS01 در زیر ارائه شده است:

در ابتدا، موقعیت و سرعت ذرات و همچنین بردار احتمالات انتخاب اعمال اتوماتای یادگیر مقداردهی اولیه می شوند. سپس تا زمانیکه حداکثر تعداد گامها انجام گردد و یا هدف مورد نظر حاصل شود، مراحل زیر تکرار میشوند:

- ۱- اتوماتای یادگیر یکی از اعمالش را بر طبق بردار احتمال اعمالش، انتخاب می کند.
- ۲- با توجه به عمل انتخاب شده، نحوه بروزرسانی سرعت ذرات تعیین میشود و سپس ذرات، سرعت و موقعیت خود را بروز میکنند.
- ۳- بر اساس نتایج به روزرسانی موقعیت ذرات، عمل اتوماتای یادگیر، ارزیابی بردار احتمال انتخاب اعمال اتوماتای یادگیر اصلاح میشود.

۴- عملی که اتوماتای یادگیر در هر گام برمیزگزيند، تعیین کننده شیوه بروز کردن سرعت ذرات در آن گام میباشد. در صورت انتخاب عمل «دنباله روی»، بروزرسانی سرعت ذرات مطابق با فرمول (۶) انجام میگردد

$$V_i^d(t+1) = c_1 rand_1(t)(pbest_i^d(t) - X_i^d(t)) + c_2 rand_2(t)(gbest(t) - X_i^d(t)) \quad (6)$$

۵- در صورت انتخاب عمل «ادامه مسیر فعلی»، سرعت جدید ذره به صورت معادله (۷) به روز می گردد.

$$w(t) = w_0 - \frac{t * w_1}{MaxGen} \quad (7)$$

$$V_i^d(t+1) = w(t) * V_i^d(t) + c_1 * rand_{1i}^d * (pbest_i^d(t) - X_i^d(t)) + c_2 * rand_{2i}^d * (gbest^d(t) - X_i^d(t))$$

شیوه ارزیابی عمل انتخاب شده در این الگوریتم همانند الگوریتم PSO_LA می باشد. در این روش رفتار ذرات تابع رای اکثریت گروه می باشد. بنابراین ذره ای که رفتار خوبی داشته اطلاعات مفید خود را از دست می دهد. در این مقاله سعی شده است تا در مدل های پیشنهادی بعدی این مشکل رفع گردد.

LAPSO2. ۲.۵

در این روش به هر یک از ذرات موجود در گروه ذرات یک اتوماتای یادگیر اختصاص می دهد. هر اتوماتای یادگیر در واقع به عنوان مغز ذره می باشد که حرکت ذره در فضای جستجو را رهبری می نمایند. هر اتوماتای یادگیر دارای دو عمل «دنباله روی» و «دامه مسیر فعلی» می باشد. هر گاه اتوماتای تخصیص داده شده به ذره عمل «دنباله روی» را انتخاب نماید بدین مفهوم است که ذره برای حرکت در فضای جستجو باینرسی حرکتی اولیه صفر به سمت بهترین موقعیت ملاقات شده توسط گروه gbest و بهترین موقعیتی که تا بحال ملاقات کرده pbest می رود این عمل جستجوی محلی را در پی خواهد داشت. در صورتیکه اتوماتای یادگیر هر ذره عمل «دامه مسیر فعلی» را انتخاب نماید بدین مفهوم است که ذره با یک شتاب خاص در فضا حرکت می نماید و ذره همچنان مسیری فعلی را ادامه خواهد داد. این عمل باعث جستجوی سراسری و کشف قسمتهایی ناشناخته از فضای جستجو می گردد. الگوریتم این روش به صورت زیر می باشد:

۱. ذرات با یک موقعیت و سرعت اولیه به صورت تصادفی در فضای جستجو قرار داده می شوند.
۲. به هر ذره یک اتوماتای یادگیر اختصاص داده می شود.
۳. بردار احتمالات انتخاب اعمال اتوماتای یادگیر هر ذره مقداردهی اولیه میشوند.
۴. تا زمانیکه 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 ذره i ام عمل «دنباله روی» را انتخاب کرده باشد سرعت ذره به صورت فرمول (۶) به روز می شود (اینرسی سرعت ذره را صفر در نظر می گیریم).
۷. اگر اتوماتای یادگیر تخصیص داده شده به ذره «دامه مسیر فعلی» را انتخاب کرده باشد سرعت فعلی ذره برابر با سرعت قبلی ذره است بنابراین برای به روز نمودن سرعت از فرمول (۷) استفاده می شود.
۸. با توجه به عمل انتخاب شده، نحوه بروزرسانی سرعت ذرات تعیین می شود و سپس ذرات، سرعت و موقعیت خود را بروز می کنند.
۹. چنانچه موقعیت جدید ذره نسبت به موقعیت قبلی ذره بهبود یافته باشد به عمل انتخاب شده توسط ذره پاداش داده می شود و در غیر این صورت به عمل انتخاب شده جریمه تعلق می گیرد.

$$\beta = \begin{cases} 0 & \text{if } fitness(X_i(t+1)) > fitness(X_i(t)) \\ 1 & \text{otherwise} \end{cases} \quad (8)$$

۱۰. بردار احتمال انتخاب اعمال اتوماتای یادگیر اصلاح میشود.

LAPSO3. ۳.۵

در مدل پیشنهادی دوم در هر تکرار از الگوریتم ذرات رفته رفته جذب بهترین موقعیت ملاقات شده گروه می شوند. در صورتیکه قسمت دوم و سوم معادله سرعت PSO استاندارد مقدار کوچکی باشد هر چند سرعت مقدار زیادی داشته باشد به مرور زمان مقدار سرعت کاهش یافته و ذره خاصیت جستجوی عمومی خود را برای آینده از دست می دهد. بخصوص در مراحل ابتدایی الگوریتم اگر چنین اتفاقی رخ دهد دیگر ذره توانایی جستجو در نواحی جدید را ندارد. اگر قسمت دوم و سوم معادله سرعت PSO استاندارد صفر باشد سرعت ذره با نرخ (inertia weight) w شروع به کاهش می نماید تا صفر شده و ذره متوقف گردد. به عبارت دیگر اگر موقعیت ذره به بهترین موقعیت ملاقات شده توسط ذره و یا به بهترین موقعیت ملاقات شده توسط گروه منطبق شود، تنها عاملی که ذره را از بهترین موقعیت ملاقات شده کل گروه gbest دور نگه می دارد سرعت و w می باشند. در صورتیکه کلیه ذرات سرعتشان به صفر نزدیک باشد در gbest متوقف شده که باعث همگرایی زودرس در gbest می شود کما اینکه همگرایی در مینیمم محلی نیز نبوده و در بهترین موقعیت ملاقات شده توسط ذرات می باشد. جهت رفع این مشکل هنگامیکه سرعت ذرات از یک حد آستانه ای کمتر شود، سرعت ذره را دوباره مقدار دهی می نمایند تا از قرار گرفتن ذره در مینیمم محلی جلوگیری شود. معادله تنظیم دوباره سرعت ذره به صورت زیر آورده شده است.

$$V_{i+1}^d = \begin{cases} V_{i+1}^d & \text{if } |V_{i+1}^d| \geq V_c \\ u(-1,1)V_{max}/\rho & \text{if } |V_{i+1}^d| < V_c \end{cases} \quad (9)$$

نحوه انجام این الگوریتم کاملاً شبیه مدل پیشنهادی اول است با این تفاوت که در مرحله ۸ از الگوریتم بعد از به روز نمودن موقعیت و سرعت ذره، در صورتیکه سرعت ذره از حداستانه V_c کمتر بود طبق معادله (۹) دوباره مقدار دهی می‌گردد.

LAPSO4.۴.۵

پاداش دادن و جریمه نمودن یک ذره تأثیر بسیار زیادی در عملکرد ذره دارد. می‌توان به صورت دیگری نیز برای اعمال انجام شده توسط اتوماتای یادگیر ذرات پاداش و یا جریمه در نظر گرفت بدین صورت که چنانچه موقعیت جدید ذره از بهترین موقعیت ذره ملاقات شده ذره بهتر باشد به ذره پاداش داده شود. الگوریتم این روش کاملاً شبیه الگوریتم مدل پیشنهادی دوم است با این تفاوت که در مرحله ۹ نحوه پاداش دادن الگوریتم بدین صورت است که:

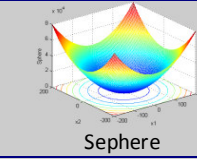
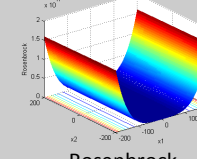
$$\beta = \begin{cases} 0 & \text{if } fitness(X_i(t+1)) > pbest_i \\ 1 & \text{otherwise} \end{cases} \quad (10)$$

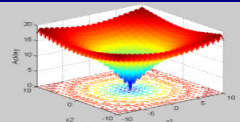
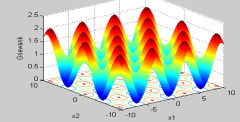
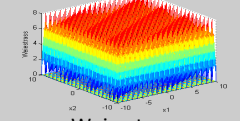
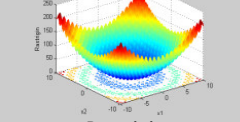
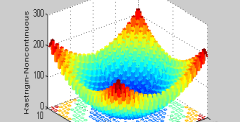
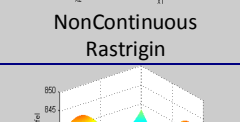
۶. نتایج شبیه سازیها

آزمایشات بر روی هشت تابع استاندارد صورت گرفته است که معمولاً به عنوان معیار سنجش الگوریتمهای بهینه سازی مورد استفاده قرار میگیرند. توابع استفاده شده در جدول ۱ نمایش داده شده اند. مینیمم عمومی هر یک از توابع، محدوده مقداردهی اولیه ذرات و محدوده فضای جستجوی هر یک از توابع در جدول ۲ آورده شده است. الگوریتم های پیشنهادی با الگوریتم PSO با وزن اینرسی PSOw و PSO با فاکتور انقباض PSO-CF و PSO-LA مقایسه شده است. هر کدام از این الگوریتمها پارامترهایی دارند که باید قبل از اجرای الگوریتم به صورت دستی تنظیم شوند این تنظیمات در جدول ۳ آورده شده است. دو نمونه آزمایش بر روی ۸ تابع مذکور صورت گرفته است.

آزمایش اول بر روی ۸ تابع در ۱۰ بعد با ۵ ذره و ۳۰۰۰۰ تکرار صورت گرفته است که در جدول ۴ نتیجه آن آورده شده است. (نتایج مندرج در جداول میانگین و واریانس ۳۰ بار اجرای الگوریتم می باشد). همانطور که از نتایج مشخص است. در توابع تک قله ای مانند تابع ۱ و ۲ الگوریتم PSOw و PSOcf سریعتر همگرا می‌شوند و دارای پاسخ بهتری نسبت به الگوریتمهای پیشنهادی می‌باشند. زیرا الگوریتمهای پیشنهادی محدوده بیشتری از فضای جستجو را مورد بررسی قرار می‌دهند. اما در تابع ۳ الگوریتم LAPSO2LRP دارای راندمان بالاتری نسبت به سایر الگوریتم ها می‌باشند. در تابع ۴ الگوریتم LAPSO3LRP راندمان بالاتری دارند. در تابع ۵، ۶ و ۷ الگوریتم های پیشنهادی تفاوت قابل ملاحظه ای نسبت به سایر الگوریتم های اولیه PSOw، PSOcf و PSO-LA دارند. اما در مورد تابع ۸ کماکان هیچ کدام از الگوریتم ها پاسخ مناسبی برای این تابع نیافته اند. در کل الگوریتم های پیشنهادی راندمان بالاتری در توابع چند قله ای دارند. بهترین موقعیت ملاقات شده بر حسب تکرار در الگوریتم-های مختلف برای ۸ تابع در شکل ۲ نمایش داده شده است. با توجه به این نمودارها منحنی الگوریتم PSOw و PSOcf در توابع ۱ و ۲ بسیار سریع همگرا می‌شود. اما در توابع چند قله ای منحنی این الگوریتم ها 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 نمی‌مانند که نشان دهنده قرار نگرفتن این ذرات در مینیمم محلی است. آزمایش دوم بر روی ۸ تابع در ۳۰ بعد با ۵ ذره و ۳۰۰۰۰ تکرار صورت گرفته است که در جدول ۵ نتیجه آن آورده شده است. (نتایج مندرج در جداول میانگین و واریانس ۳۰ بار اجرای الگوریتم می باشد). با توجه به جدول می توان نتیجه گرفت که نتایج همانند پاسخهای 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 می‌باشد. در توابع چند قله ای راندمان الگوریتم های پیشنهادی نسبت به سایر الگوریتم های اولیه PSOw، PSOcf و PSO-LA بالاتر می‌باشد. بهترین موقعیت ملاقات شده بر حسب تکرار در الگوریتمهای مختلف برای ۸ تابع در شکل ۳ نمایش داده شده است. با توجه به این نمودارها در توابع چند قله ای منحنی این الگوریتم ها سیر نزولی دارند و ثابت نمی ماند که نشان دهنده قرار نگرفتن این ذرات در مینیمم محلی است.

جدول ۱: توابع مورد تست

فرمول	تابع دو بعدی
$f_1(x) = \sum_{i=1}^D x_i^2$	 Sphere
$f_2(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1}^2)^2 + (x_i^2 - 1)^2)$	 Rosenbrock

$f_3(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	 <p>Ackley</p>
$f_4(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	 <p>Greiwank</p>
$f_5(x) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k 0.5)]$ <p>$a = 0.5, b = 3, k_{\max} = 20$</p>	 <p>Weierstrass</p>
$f_6(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	 <p>Rastrigin</p>
$f_7(x) = \sum_{i=1}^D (y_i^2 - 10 \cos(2\pi y_i) + 10)$ $y_i = \begin{cases} x_i & x_i < 1/2 \\ \frac{\text{round}(2x_i)}{2} & x_i \geq 1/2 \end{cases}$ <p>for $i = 1, 2, \dots, D$</p>	 <p>NonContinuous Rastrigin</p>
$f_8(x) = 418.9829 * D - \sum_{i=1}^D x_i \sin\left(x_i ^{\frac{1}{2}}\right)$	 <p>Schwefel</p>

جدول 2: مقادیر اولیه توابع تست

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

جدول 3: تنظیمات پارامترهای الگوریتم ها

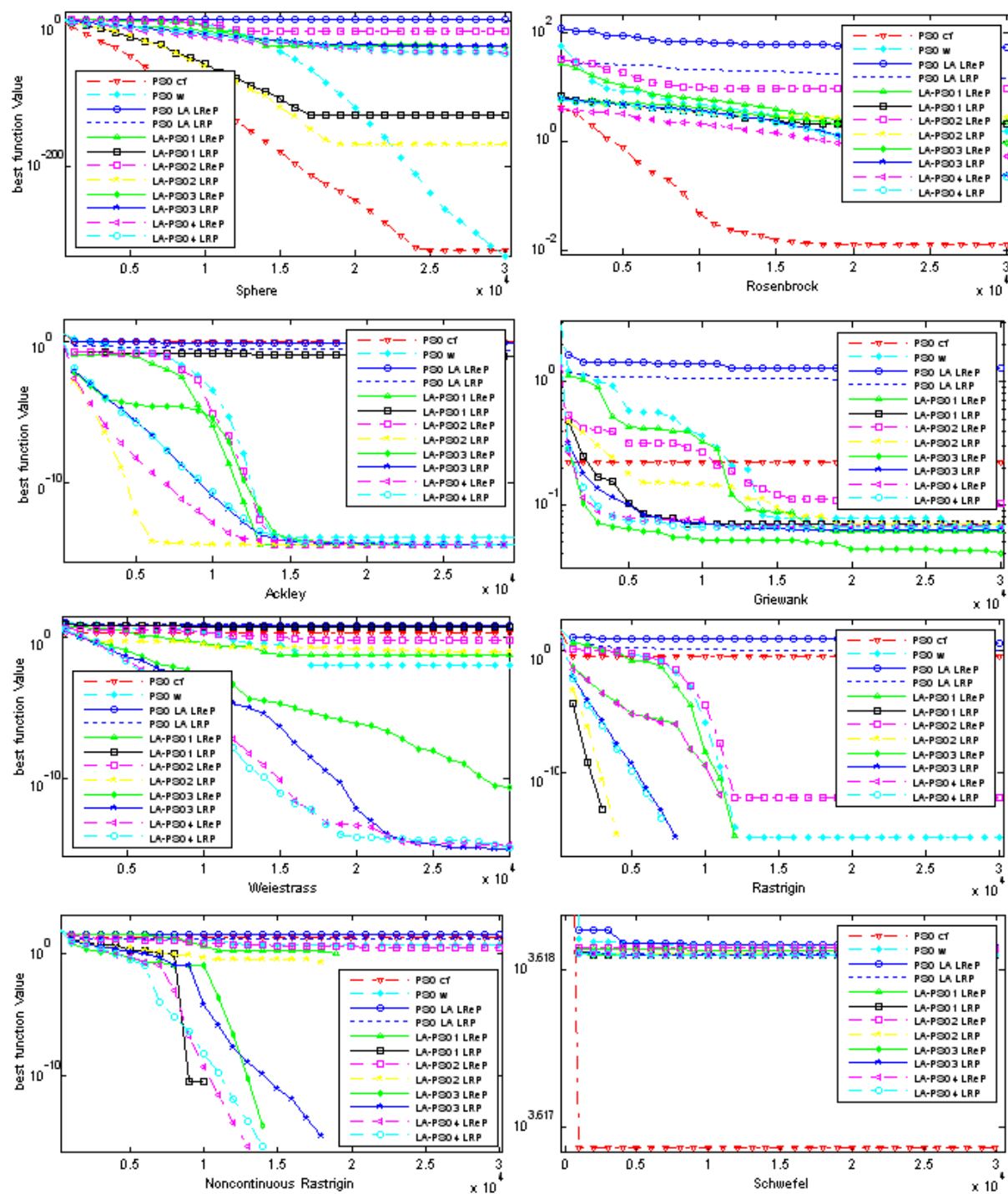
	W0	W1	C1	C2	Cimp	Vc	P	LA_LRP	LA_LReP
PSO w	0.9	0.4	1.45	1.45					
PSO cf	0.75	0.75	2	2					
PSO-LA	0.9	0.4	1.45	1.45	75			$\alpha=\beta=0.01$	$\alpha=0.001, \beta=0.01$
LAPSO1	0.9	0.4	1.45	1.45	75			$\alpha=\beta=0.01$	$\alpha=0.001, \beta=0.01$
LAPSO2	0.9	0.4	2	2				$\alpha=\beta=0.01$	$\alpha=0.001, \beta=0.1$
LAPSO3	0.9	0.4	2	2		1e-20	7	$\alpha=\beta=0.01$	$\alpha=0.001, \beta=0.01$
LAPSO4	0.9	0.4	2	2		1e-20	7	$\alpha=\beta=0.01$	$\alpha=0.001, \beta=0.01$

جدول 4: مقایسه الگوریتم پیشنهادی و سایر الگوریتم ها در ۸ تابع ۱۰ بعدی

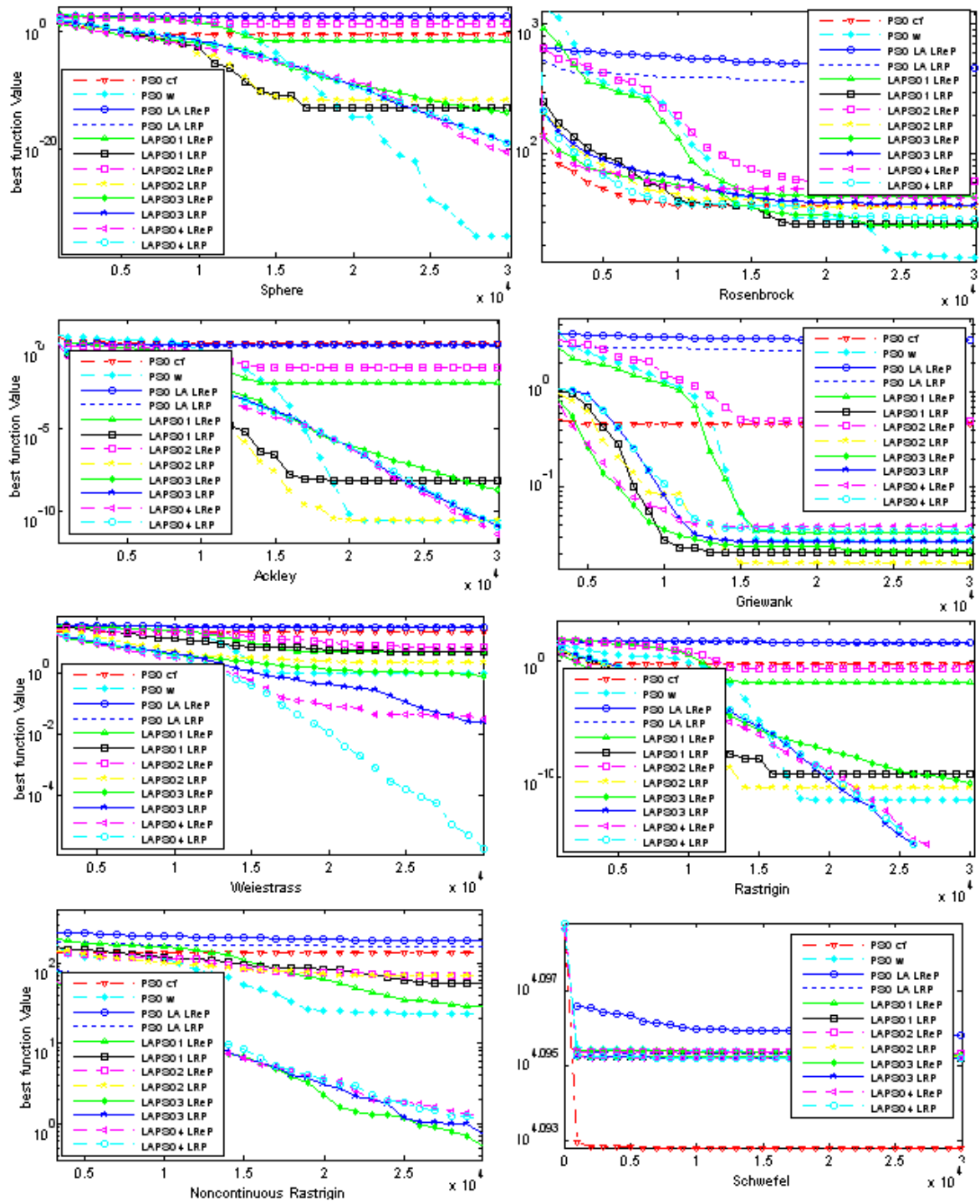
	F1	F2	F3	F4
PSO cf	1.2418e-314±4.5618e-371	0.0124±3.9667e-004	2.2737e-014±8.5323e-028	0.0799±0.0016
PSO w	2.0405e-321±4.3906e-257	1.5283±2.6112	0.6957±0.6130	0.1632±0.0085
PSO LA LRP	8.3992±7.3897	14.0788±6.4807	0.2463±0.0046	0.9643±0.0362
PSO LA LReP	56.3939±134.7245	52.9057±201.8486	0.7809±0.00978	1.2837±0.109
LAPSO1 LRP	3.0470e-129±9.1730e-257	2.0069±1.9846	0.0977±0.0197	0.0692±3.3223e-004
LAPSO1 LReP	1.6983e-034±2.8842e-067	2.1958±4.8066	3.5527e-015±0	0.0615±0.0012
LAPSO2 LRP	3.4075e-170± 3.4075e-190	2.6870±2.7885	3.1974e-015±1.2622e-030	0.0711±0.0016
LAPSO2LReP	1.4721e-014±2.1672e-027	9.0277±35.4323	3.5527e-015±0	0.1018±0.0023
LAPSO3 LRP	1.6675e-034±1.2507e-068	0.2280±0.0078	3.5527e-015±0	0.0625±9.3018e-004
LAPSO3 LReP	1.4987e-034±1.2208e-068	0.9110±0.4825	3.5527e-015±0	0.0406±5.4235e-004
LAPSO4 LRP	3.3601e-044±5.2617e-088	0.0208±0.2195	3.9080e-015±1.2622e-030	0.0671±4.4587e-004
LAPSO4 LReP	5.1131e-044±2.0892e-087	0.5263±1.4968	3.5527e-015±0	0.0674±0.0012
	F5	F6	F7	F8
PSO cf	0.0333±0.0068	3.5527e-016±1.6097e-31	0.9000±0.7667	4.1507e+003±0.0030
PSO w	2.4964±1.5691	0.8379±0.9089	21.7368±65.8713	4.138e+003±374.6959
PSO LA LRP	7.3139±0.4251	0.9815±0.0333	37.8118±0.01	4.1505e+003±0
PSO LA LReP	6.6444±0.124	4.4543±0	14±2.45	4.1510e+003±0
LAPSO1 LRP	5.2655±13.1760	0±0	0±0	4.1504e+003±0
LAPSO1 LReP	0.0532±0.0011	0±0	0±0	4.1506e+003±0
LAPSO2 LRP	0.1032±0.020	0±0	0±0	4.1504e+003±1.667e-004
LAPSO2LReP	0.6085±0.55791	7.7645e-013±5.7292e-024	3.2005±6.8477	4.1507e+003±0.0095
LAPSO3 LRP	1.0658e-015±2.9451e-030	0±0	0±0	4.1504e+003±0
LAPSO3 LReP	2.4722e-011±5.7176e-021	0±0	0±0	4.1504e+003±1.973e-004
LAPSO4 LRP	1.4211e-015±1.1780e-029	0±0	0±0	4.1504e+003±6.148e-005
LAPSO4 LReP	2.1316e-015±3.1414e-029	0±0	0±0	4.1504e+003±0

جدول 5: مقایسه الگوریتم پیشنهادی و سایر الگوریتم ها در ۸ تابع ۳۰ بعدی با ۵ ذره

	F1	F2	F3	F4
PSOcf	0.4298±0.5723	40.0853±644.5545	1.5869±0.4308	0.4444± 0.1245
PSO w	4.217e-034±3.6705e-067	17.7492±79.5980	2.7320e-008±7.4076e-015	0.0329±0.0018
PSO LA LRP	258.8194±1.2231e+003	335.4866±1.8071e+003	1.0370±0.0052	3.4420±0.0155
PSO LA LReP	383.1832±1.3331e+003	460.5667±4.5067e+003	1.2550±0.0043	2.6987±0.1291
LAPSO1 LRP	1.401e-013±7.1681e-026	29.3942±275.0456	8.1099e-009±2.4196e-016	0.0204±0.0010
LAPSO1 LReP	0.0272± 0.0055	47.0219±666.8714	0.0061±8.7614e-005	0.0334±4.8586e-004
LAPSO2 LRP	2.167e-012±1.3394e-023	39.020±674.8228	2.6823e-011±1.9782e-021	0.016±6.2225e004
LAPSO2LReP	18.1432±3.2883e+003	61.7932±1.9677e+003	0.0511±0.0138	0.4893±1.3695
LAPSO3 LRP	8.770e-020± 2.3656e-039	40.3704±902.2664	1.1636e-011± 1.4016e-022	0.0272±7.3246e-004
LAPSO3 LReP	1.430e-014±1.9797e-027	28.0745 ±591.2671	1.9474e-009±1.9474e-009	0.0214±3.24623-004
LAPSO4 LRP	1.347e-019±2.1415e-038	31.5803±877.3246	1.2714e-011±5.4917e-023	0.0346±0.0011
LAPSO4 LReP	2.488e-021±4.6186e-041	46.1007±782.2940	3.8639e-012± 4.7452e-023	0.0386±0.0022
	F5	F6	F7	F8
PSO cf	20.7938±15.6794	0.6344±0.2649	132.0500±841.2194	1.2383e+004±5.4533e+003
PSO w	1.0371±1.0632	1.1958e-012±4.0718e-024	23.4±291.8222	1.2454e+004±0.1690
PSO LA LRP	37.0573±37.0573	33.3892±14.6191	157.3910±195.6791	1.2454e+004±0.2034
PSO LA LReP	31.5882±21.5825	44.0232±27.6941	194.8198±81.5650	1.2469e+004±17.7907
LAPSO1 LRP	4.9605±10.6045	1.7110e-10±1.3083e-019	54.9038±678.2944	1.2452e+004±0.0219
LAPSO1 LReP	4.7831±31.7103	0.0141±0.0017	28.3307±968.3732	1.2454e+004±0.1595
LAPSO2 LRP	2.3808±4.7863	1.7110e-010±1.3083e-019	68.3213±1.0758e+003	1.2452e+004±0.0079
LAPSO2LReP	7.2227±5.7324	0.2013±0.1300	65.7398±462.8724	1.2454e+004±5.2434
LAPSO3 LRP	0.0244±0.0050	1.1782e-011±1.3750e-021	0.7140±1.1523	1.2451e+004±0.0033
LAPSO3 LReP	0.8174±0.7905	2.9972e-011±7.4394e-021	0.5034±0.2814	1.2451e+004±0.4467
LAPSO4 LRP	1.278e-06±1.191e-11	0±0	1.0136±1.7770	1.2454e+004± 9.365e-004
LAPSO4 LReP	0.0314±0.0064	0±0	1.1746±2.6184	1.2454e+004±0.0021



شکل ۲: میانگین بهترین تابع شایستگی ذرات نسبت به تعداد تکرارهای الگوریتم در ۱۰ تکرار، ۵ ذره، ۳۰۰۰۰ تکرار ۸ تابع (a). Sphere (b). Rosenbrock (c). Ackley (d). Griewank (e). Weierstrass (f). Rastrigin (g). Noncontinuous Rastrigin (h) و Schwefel



شکل ۳: میانگین بهترین تابع شایستگی ذرات نسبت به تعداد تکرارهای الگوریتم در ۳۰ تکرار، ۵ ذره، ۳۰۰۰۰ تکرار و ۸ تابع (a) Sphere، (b) Rosenbrock، (c) Ackley، (d) Griewank، (e) Weierstrass، (f) Rastrigin، (g) Noncontinuous Rastrigin، (h) Schwefel.

۷. نتیجه گیری

بهینه سازی گروه ذرات یکی از قویترین روشهای بهینه سازی است که از زندگی جانورانی که به صورت انبوه زندگی می کنند الهام می گیرد. الگوریتم PSO_LA گونه جدیدی از بهینه سازی گروه ذرات است که از یک اتوماتای یادگیر برای تنظیم رفتار گروه و برقراری موازنه بین جستجوی سراسری و جستجوی محلی استفاده می نماید. مشکل اصلی این مدل علاوه بر قرار گرفتن در بهینه های محلی، این است که یک ذره باید روند حرکت خود را به تبعیت اکثر افراد گروه، تغییر دهد. هر چند امکان دارد روند کنونی حرکت ذره مناسب باشد. این تابعیت صرف از گروه باعث می شود اطلاعات مفید ذرات گروه نادیده گرفته شود. جهت بهبود مشکلات مذکور در این مقاله الگوریتم جدید بهینه سازی گروه ذرات مبتنی بر اتوماتای یادگیر به نامهای LAPS01, LAPS02, LAPS03 و LAPS04 ارائه شده است که در LAPS01 همانند روش PSO_LA یک اتوماتای یادگیر برای کل گروه در نظر گرفته می شود، در سه روش پیشنهادی دیگر به هر ذره از گروه یک اتوماتای یادگیر اختصاص داده می شود تا روند حرکت ذره را تنظیم نماید. اتوماتای یادگیر اختصاص داده شده به هر ذره در هر گام تعیین میکند که ذرات به مسیر فعلی ادامه دهند و یا به دنباله روی از بهترین ذرات پیدا شده تاکنون بپردازند. نتایج آزمایشها نشان می دهند که روش ارائه شده از عملکرد بهتری در مقایسه با مدل PSO استاندارد و PSO_LA برخوردار است.

۸. مراجع

- [1] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization", Proc. of IEEE International Conference on Neural Networks, Piscataway, NJ, p.p. 1942-1948, 1995.
- [2] K. E. Parsopoulos, V. P. Plagianakos, G.D. Magoulas, and M. N. Vrahatis, "Objective Function "Stretching" to Alleviate Convergence to Local Minima" NonLinear Analysis, Theory , Methods and Applications, vol. 47,number 5, pp.3419-3424,2001.
- [3] C. Coello Coello and M. Lechuga, "MOPSO: A Proposal for Multiple Objective Particle Swarm Optimizations", In Congress on Evolutionary Computation, Piscatawa, New Jersey, USA, vol. 2, p.p. 1051-1056, IEEE Service Center, 2002.
- [4] X. Hu and R. C. Eberhart, "Multiobjective Optimization using Dynamic Neighborhood Particle Swarm Optimization" , in Proc. Congr. Evol. Compute, Honolulu, HI pp. 1677-1681, 2002.
- [5] J. Kennedy and R. C. Eberhart, "A Discrete Binary Version of the Particle Swarm Algorithm", Proceedings of the International Conference on Systems, Man, and Cybernetics, IEEE Service Center, Piscataway, NJ, pp. 4104-4108, 1997.
- [6] P. Yin. "A.Discrete Particle Swarm Algorithm for Optimal Polygona Approximation of Digital Curves". Journal of Visual Communication and Image Representation ,15(2):241-260, June 2004.
- [7] A. Ratnaweera, S. Halgamuge, and H. Watson, "Self-organizing hierarchical particle swarm optimizer with time varying accelerating coefficients," IEEE Trans. Evol. Computing., vol. 8, pp. 240-255, Jun. 2004.
- [8] Y. Zheng, L. Ma, L. Zhang, and Qian . "Emperical Study of Particle Swarm Optimizer with Increasing Inertia Weight". In Proceeding of IEEE Congress on Evolutionary Computation, pp.221-226. IEEE Press,2003.
- [9] A. Ratnaweera, S. Halgamuge, and H. Watson, "Particle Optimizer with Time varying Acceleration Coefficients". In Proceedings of the international Conference on Soft Computing and Intelligent Systems, pp. 240-25, 2002.
- [10] A. Ratnaweera, S. Halgamuge, and H. Watson, "Particle Swarm Optimization with Self-Adaptive Acceleration Coefficients". In Proceedings of the First International Conference on Fuzzy Systems and Knowledge Discovery, pp. 2411-2418 December 2003.
- [11] J. Kennedy , "Small Worlds and Mega-Minds : Effects of Neighborhood Topology on Particle Swarm Performance". In Proceeding s of The IEEECongress on Evolutionary Computation , vol. 3, pp. 1931-1938 , July 1999.
- [12] F. Van den Bergh and A.P. Engelbrecht. "Effects of Swarm Size on Cooperative Particle Swarm Optimizers". In Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco, USA, pp.892-899,2001.
- [13] A. Silva, A. Neves, and E. Costa, "An Empirical Comparison of Particle Swarm and Predator Prey Optimization" ,In Proceedings of the Thirteenth Irish Conference on Edificial Intelligence and Cognitive Science , In: Lecture Notes in Artificial Intelligence , vol. 2464,pp. 103-110 , Springer_Verlag,2002.
- [14] J.S. Vesterstrom , J. Giget, and T.Krink, "Division of Labor in Particle Swarm Optimizatio" . In Proceedings of the IEEE Congress on Evolutionary Computation , pp. 1570-1575 IEEE press,2002.
- [15] P. Yin. A.Discrete Particle Swarm Algorithm for Optimal Polygona Approximation of Digital Curves. Journal of Visual Communication and Image Representation , Vol. 15, No. 2 ,pp. 241-260, June 2004.
- [16] M. Løvberg, T. Rasmussen and T. Krink. "Hybrid Particle Swarm Optimizer with Breeding and Subpopulation". In Proceedings of the Third Genetic and Evolutionary Computation Conference, vol. 1, pp. 469-476, 2001.
- [17] Y. Shi and R. C. Eberhart, "A Modified Particle Swarm Optimizer", IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, USA, 1998.

- [18] M. A. L. Thathachar and P. S. Sastry, "*Varieties of Learning Automata: An Overview*", IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 32, No. 6, pp. 711-722, 2002.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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
- [23] 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 Lumpur, Malaysia, Nov. 25-28, 2007
- [24] R. Rastegar, M. R. Meybodi and K. Badie, "*A New Discrete Binary Particle Swarm Optimization based on Learning Automata*", In Proceedings of International Conference on Machine Learning and Applications(ICMLA2004), 16-18 December 2004, pp.456-462, USA, IEEE Press., 2004
- [25] K. S. Narendra and M. A. L Thathachar., "*Learning Automata: An Introduction*", Prentice-Hall Inc, 1989.