

یک روش گروهی برای اتوماتای یادگیر

فریبا آب‌روشن
کارشناس علوم کامپیوتر دانشکده ریاضی و
علوم کامپیوتر
دانشگاه ولیعصر (عج) رفسنجان

محمدابراهیم شیری
عضو هیئت علمی دانشگاه صنعتی امیرکبیر
دانشگاه صنعتی امیرکبیر
shiri@aut.ac.ir

منصور داودی منفرد
کارشناسی ارشد علوم کامپیوتر
دانشگاه صنعتی امیرکبیر
mdmonfared@aut.ac.ir

بهینه‌سازی گروهی، (PSO) Particle Swarm Optimization، [13,14] نام برد.

از آنجایی که روشهای ابتکاری بر اساس احتمالات عمل می‌کنند، بنابراین هیچکدام از آنها تضمینی در مورد یافتن جواب بهینه ندارند، علاوه بر این کارایی تمامی این روشها با افزایش پیچیدگی مسئله کاهش می‌یابد. این پیچیدگی می‌تواند مربوط به افزایش بعد مسئله و یا شکل فضای جستجو باشد. به عنوان مثال تمام روشهای ابتکاری در فضاهای جستجوی محدب و پیوسته که در آنها جواب بهینه محلی همان جواب بهینه عمومی است، کم و بیش یکسان عمل کرده و قادر به یافتن جواب بهینه عمومی می‌باشند؛ اما فقط بعضی از آنها می‌توانند در فضاهای غیرمحدب به صورت موفق عمل نمایند، آن هم نه به صورت صد در صد، بویژه اگر تعداد جوابهای بهینه محلی زیاد باشد، کارایی چنین روشهایی نیز به شدت کاهش می‌یابد. مشکلات یافتن جواب (جوابهای) بهینه عمومی با گسستگی در فضای جستجو نیز افزایش می‌یابد. ناپیوستگی در فضای جستجو می‌تواند در یافتن جوابهای بهینه پارتو^۱ در الگوریتم‌های ابتکاری چندهدفه که باید مجموعه‌ای از جوابهای بهینه پارتو را با پراکندگی و چگالی خوبی پیدا کنند با مشکلات بسیار بیشتری مواجه سازد. الگوریتم‌های ابتکاری شیوه‌های متفاوتی را برای گریز از این مشکلات در پیش می‌گیرند که از آن جمله می‌توان به استراتژی جهش در الگوریتم‌های ژنتیکی، پذیرفتن جوابهای کم‌ارزش‌تر در الگوریتم شبیه‌سازی تبریدی و شروع مجدد تصادفی یا انتخاب اولین همسایه در الگوریتم تپه‌نوردی اشاره کرد [15-19].

اتوماتای یادگیر (LA) و بهینه‌سازی گروهی (PSO) نیز در این موارد استثنا نمی‌باشند. بهینه‌سازی گروهی روشی است که در فضاهای پیوسته بسیار سریعتر و قدرتمندتر از سایر روشها عمل می‌نماید [20]. ولی با افزایش پیچیدگیهای ذکر شده مشکلات این روش نیز دوچندان می‌شود. اتوماتای یادگیر کمی متفاوت با سایر روشهای ابتکاری عمل می‌نماید. در واقع این روش بر پایه تجربه‌ای که در تعامل با محیط به دست می‌آورد، سعی می‌کند با تصحیح میزان احتمال اعمال مختلف به سمت جواب بهینه عمومی حرکت نماید ولی همچنان مشکل گیر

چکیده: در دهه اخیر استفاده از روشهای ابتکاری و تکاملی در حل مسائل بهینه‌سازی، کنترل و طراحی که عموماً دارای فضاهای جستجوی چندبعدی و پیچیده می‌باشند، افزایش یافته است. در این مقاله یک روش جدید جستجوی گروهی که مبتنی بر اتوماتای یادگیر و بهینه‌سازی گروهی، می‌باشد، ارائه شده است. این روش که ما آن را اتوماتای یادگیر گروهی، می‌نامیم، می‌تواند بر روی هر نوع اتوماتای یادگیری پیاده‌سازی شود. ایده اصلی اتوماتای یادگیر گروهی، بکارگیری چندین جواب تصادفی به جای یک جواب تصادفی می‌باشد. زمانی که گروهی از جوابها استفاده شود، می‌توان از ویژگیهای گروهی که الگوریتم بهینه‌سازی گروهی بر مبنای آن ایجاد شده است، سود برد. برای بهره‌گیری از این منافع گروهی در اتوماتای یادگیر، بهترین جوابهای گروه نگهداری شده و از آنها در تقویت الگوریتم یادگیری اتوماتا، که مهمترین رکن یک اتوماتای یادگیر است، استفاده می‌شود.

اتوماتای یادگیر گروهی، بسیار قوی‌تر از اتوماتای یادگیر بوده و در نواحی نامحدب با احتمال بسیار بیشتری جواب بهینه عمومی را پیدا می‌کند. در پایان روش گروهی ارائه شده بر روی دو مسئله تست با اتوماتای یادگیر مقایسه شده و نتایج آن که اثباتی بر کارایی اتوماتای یادگیر گروهی می‌باشد، آورده شده است.

کلمات کلیدی: اتوماتای یادگیر، اتوماتای یادگیر گروهی، بهینه‌سازی گروهی، الگوریتم‌های ابتکاری، الگوریتم‌های تکاملی

۱- مقدمه

در مسائل غیرخطی و پیچیده که دارای فضاهای جستجوی چندبعدی می‌باشند، یافتن یک یا چند جواب که هدف خاصی را ارضا نمایند، کاری است پیچیده و مشکل. این نوع مسائل که عموماً در کلاس مسائل NP-complete و NP-hard طبقه‌بندی می‌شوند، دارای حل‌هایی با زمانهای نمایی می‌باشند. از اینرو استفاده از الگوریتم‌های ابتکاری یا تکاملی در حل چنین مسائلی که معمولاً مسائل بهینه‌سازی، کنترل و طراحی می‌باشند، رواج یافته است [1-8]. در این میان می‌توان از روش‌های اتوماتای یادگیر، (LA) Learning Automata، [9-12] و

تکاملی که از نحوه همکاری و هماهنگی پرنندگان در یافتن غذا الهام گرفته شده است، در فضاهای پیوسته بهتر و سریعتر از سایر نمونه‌های تکاملی دیگر مانند الگوریتم ژنتیک، کار می‌کند [20]. در الگوریتم PSO هر جواب یک ذره و جمعیت جوابها، گروه نامیده می‌شود. در PSO علاوه بر جمعیت جوابها برداری که حاوی سرعت فعلی هر ذره می‌باشد نیز وجود دارد که مقدار اولیه آن همانند خود ذره‌ها به صورت تصادفی مقداردهی می‌شود. در واقع در هر نسل بردار سرعت تغییر یافته و با جمع شدن آن با مقدار فعلی هر ذره مقدار بعدی ذره حاصل می‌شود.

به طور کلی i -امین ذره در گروه با بردار $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ که D بعد فضای جواب می‌باشد، نشان داده می‌شود. هر ذره بردار سرعتی دارد که با $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ نشان داده می‌شود. هر ذره در هر نسل با قانون به‌روزرسانی زیر تغییر می‌کند:

$$\begin{aligned} v_{id} &= w \times v_{id} \\ &+ c_1 \times \text{Rand1} \times (pbest_{id} - p_{id}) \\ &+ c_2 \times \text{Rand2} \times (Gbest_{id} - p_{id}); \end{aligned} \quad (1-2)$$

$$p_{id} = p_{id} + v_{id}.$$

رابطه بالا به ازای تمام $d = 1, 2, \dots, D$ و $i = 1, 2, \dots, N$ که N تعداد ذره‌ها در گروه است انجام می‌شود. عموماً اندازه گروه مقاداری بین ۱۰ و ۴۰ انتخاب می‌شود، البته اندازه دقیق آن با توجه به اندازه و بعد فضای جستجو تعیین می‌گردد. مقدار $pbest_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{iD})$ می‌باشد که بهترین موقعیت کشف شده تا لحظه فعلی توسط i -امین ذره را نشان می‌دهد و $Gbest$ بهترین موقعیت کشف شده توسط گروه می‌باشد. پارامتر w ، ضریب اینرسی^۴ نامیده می‌شود و میزان تاثیر سرعت قبلی را در موقعیت بعدی کنترل می‌کند. مقدار w در نسخه اصلی برابر یک بوده است و بعدها توسط Shi در [23] معرفی شد، به طور کلی مقاداری بین صفر و یک برای آن انتخاب می‌شود و وظیفه آن کنترل تعادل بین کاوش و بهره‌برداری^۵ می‌باشد. این پارامتر در بعضی از مقالات به صورت پویا و همچنین فازی همراه با افزایش تعداد نسل تغییر می‌یابد [24, 25, 26]. مقادیر c_1 و c_2 به ترتیب پارامتر فردی و اجتماعی^۶ نامیده می‌شوند. با این پارامترها می‌توان میزان همگرایی ذره‌ها را کنترل کرد. این پارامترها مقادیر ثابت مثبتی انتخاب می‌شوند. در بیشتر مقالات برای آنها مقادیری برابر ۲ انتخاب شده است. ضرایب Rand1 و Rand2 دو مقدار تصادفی بین صفر و یک می‌باشند. تلاشهای زیادی برای بهبود نسبت پارامترهای ذکر شده انجام شده است [27-31].

همانگونه که در بالا به آن اشاره شد بردار اولیه سرعت ذره‌ها به صورت تصادفی مقداردهی می‌شود، ولی باید دقت کرد که این مقادیر سرعت متناسب با فضای جستجو باشند، در واقع برای جلوگیری از گسیختگی بیش از حد ذره‌ها و پیش آمدن پدیده انفجار^۷ عموماً بردارهای سرعت

کردن در بهینه‌های محلی را دارد. هر دو روش در ادامه به طور خلاصه شرح داده می‌شوند.

هدف ما در این مقاله ارائه یک روش بهینه‌سازی گروهی بر پایه اتوماتای یادگیر است که از ویژگیها و تجربیات گروهی که در روش بهینه‌سازی گروهی از آن استفاده می‌شود، سود می‌برد. الگوریتم پیشنهادی که ما آن را اتوماتای یادگیر گروهی می‌نامیم^۲ و با SLA نشان می‌دهیم، از نظر پیچیدگی محاسباتی با اتوماتای یادگیر هم‌درجه می‌باشد ولی دو مزیت بسیار مهم نسبت به اتوماتای یادگیر دارد: سرعت جستجوی بیشتر و احتمال بیشتر در یافتن جواب بهینه عمومی. در بخش دوم از این مقاله روش بهینه‌سازی گروهی و اتوماتای یادگیر به همراه یک الگوریتم یادگیری به طور اختصار شرح داده می‌شود. در بخش سوم به تشریح دو الگوریتم پیشنهادی می‌پردازیم و در بخش پایانی به ارائه دو مثال بهینه‌سازی پرداخته و نتایج حل آنها به صورت مقایسه‌ای با الگوریتم‌های ذکر شده آورده شده است.

۲- الگوریتم‌های تکاملی^۳

الگوریتم‌های تکاملی به طور کلی از شیوه‌های زندگی و سازش میان موجودات زنده الهام می‌گیرند. انواع مختلف این الگوریتم‌ها در روش‌های بهینه‌سازی جواب‌های فعلی با یکدیگر فرق دارند. به طور کلی می‌توان یک الگوریتم تکاملی مبتنی بر جمعیت را به شکل زیر نمایش داد:

۱. مقداردهی اولیه جمعیت.
۲. ارزیابی جوابهای جمعیت.
۳. تولید جمعیت بعدی با استفاده از جمعیت فعلی.
۴. اگر جوابی (یا جوابهایی) شروط پایانی را ارضا کنند الگوریتم تمام است و در غیر این صورت مجدداً از گام دوم شروع می‌شود.

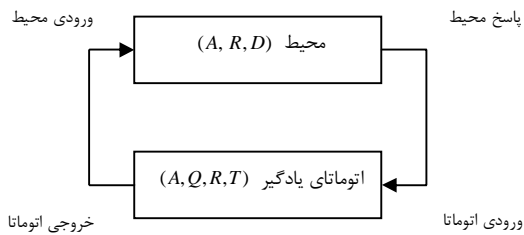
از مزایای الگوریتم‌های تکاملی می‌توان به موارد زیر که در [21] آورده شده است، اشاره کرد:

۱. پیاده‌سازی این الگوریتم‌ها ساده است.
۲. عموماً قابل بکارگیری در پردازشهای موازی می‌باشند.
۳. نیازی به پیوسته بودن توابع نمی‌باشد.
۴. معمولاً قادر به یافتن طیفی از جواب‌های نزدیک به بهینه می‌باشند.

برای بهینه‌سازی مسائل گسسته و ترکیبی بسیار مناسب هستند. در ادامه دو روش ابتکاری بهینه‌سازی گروهی و اتوماتای یادگیر که پایه‌های الگوریتم اتوماتای یادگیر گروهی می‌باشند، آورده شده‌اند.

۲-۱- الگوریتم بهینه‌سازی گروهی

الگوریتم بهینه‌سازی گروهی، PSO، توسط Kennedy و Eberhart معرفی شد [13, 14]. پس از آن محققان زیادی بر روی نسخه اصلی آن کار کرده و تغییراتی را در آن انجام داده‌اند [4, 22]. این الگوریتم



شکل (۲-۱): نمایی کلی از یک اتوماتای یادگیر

انواع مختلفی از طبقه‌بندی اتوماتاهای یادگیر انجام شده است. این طبقه‌بندی‌ها عموماً بر پایه ویژگیهای اتوماتا (قطعی، ساختار ثابت یا تصادفی)، روی ویژگیهای ورودی (دودویی، گسسته و پیوسته)، روی ویژگیهای طرح تقویتی می‌باشد [9]. در ادامه نوع خاصی از طرح تقویتی اتوماتای یادگیر را که در بخش سوم از آن استفاده شده است، می‌آوریم.

(۲-۲) طرح تقویتی خطی

if $\alpha(k) = \alpha_i$

$$\text{for } \beta(k) = 0: p_i(n+1) = \begin{cases} p_i(n) + a(1-p_i(n)), \\ (1-a)p_k(n), & k \neq i \end{cases}$$

$$\text{for } \beta(k) = 1: p_i(n+1) = \begin{cases} (1-b)p_i(n), \\ \frac{b}{r-1} + (1-b)p_k(n), & k \neq i \end{cases}$$

مقدار $\beta(k) = 0$ معادل یک پاسخ مطلوب و $\beta(k) = 1$ پاسخی نامطلوب می‌باشد. که پارامتر r تعداد اعمال اتوماتا را نشان می‌دهد. پارامترهای a و b به ترتیب میزان پاداش و جریمه برای به‌روزرسانی می‌باشند که معمولاً در کاربردهای عملی مقدار ثابتی در بازه $[0, 0.6]$ انتخاب می‌شوند. برای حالتی که مقدار b برابر صفر باشد، الگوریتم طرح تقویتی خطی reward-inaction نامیده می‌شود که در [34] با نام e -optimal شناخته شده است. حالتی که میزان جریمه و پاداش در آن برابر باشند، به نام طرح تقویتی خطی reward-penalty می‌باشد که متناسب با محیطهای ایستا می‌باشد [34,35].

۳- اتوماتای یادگیر گروهی، SLA

ایده اصلی الگوریتم گروهی ارائه شده که ما آن را اتوماتای یادگیر گروهی، (Swarm Learning Automata (SLA می‌نامیم، بکارگیری چندین جواب تصادفی به جای یک جواب تصادفی می‌باشد. زمانی که گروهی از جوابها استفاده شود، می‌توان از ویژگیهای گروهی که الگوریتم PSO بر پایه آن ایجاد شده است، سود برد. برای بهره‌گیری

در یک فاصله $[-V_{max}, V_{max}]$ مقداردهی می‌شوند و در طول اجرای الگوریتم مرتباً مقادیر سرعت بررسی می‌شوند [32,33].

۲-۲- اتوماتای یادگیر

یک اتوماتای یادگیر یک ماشین تصادفی است که با یک محیط^۸ در ارتباط است و از بازخوردهایی^۹ که از محیط دریافت می‌کند به تجربه‌اش افزوده شده و اعمالی^{۱۱} متناسب با آن انجام می‌دهد. در واقع خروجی (اعمال) بر روی محیط و ورودی آن از محیط (پاسخ محیط) می‌باشد. به طور کلی یک اتوماتای یادگیر به شکل چندتایی (A, Q, R, T) و محیط به شکل (A, R, D) می‌باشد، که $A = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه اعمال اتوماتا می‌باشد (شکل ۲-۱). عمل کنونی k -ام از اتوماتا با $\alpha(k)$ نمایش داده می‌شود که در آن $\alpha(k) \in A$ به ازای $k = 0, 1, 2, \dots$ می‌باشد. واضح است که A مجموعه خروجی اتوماتا و همان مجموعه ورودی محیط می‌باشد. R دامنه پاسخ محیط می‌باشد. اگر $b(k)$ پاسخ دریافت شده از محیط توسط اتوماتا برای عمل کنونی k -ام باشد که $\forall k \beta(k) \in R$ ، آنگاه $\beta(k)$ خروجی محیط برای عمل کنونی k -ام خواهد بود. مجموعه $D = \{d_1, d_2, \dots, d_r\}$ مجموعه احتمالی پاداش^{۱۱} می‌باشد به گونه‌ای که $d_i = E[\beta(k) | \alpha(k) = \alpha_i]$ می‌باشد. اگر d_i مستقل از k باشد، اتوماتا را اتوماتای ایستا^{۱۲} می‌نامند و در غیر این صورت اتوماتای غیرایستا نامیده می‌شود. مجموعه Q حالت اتوماتا را تعریف می‌کند به گونه‌ای که $Q(k) = [P(k), D(k)]$ و $P(k) = [P_1(k), P_2(k), \dots, P_r(k)]$ می‌باشند. البته باید توجه کرد که برای هر k موارد زیر رعایت شوند.

$$0 \leq p_i(k) \leq 1, \quad \sum_{i=1}^r p_i(k) = 1.$$

بردار $D(k) = [d_1, d_2, \dots, d_r]$ بردار ارزیابی احتمالات پاداش عمل کنونی k -ام می‌باشد. الگوریتم یادگیری یا طرح تقویتی^{۱۳} نامیده می‌شود که توسط اتوماتا برای به‌روزرسانی حالتها به کار می‌رود. برای عمل کنونی k -ام رابطه

$$T: Q(k+1) = T(Q(k), \alpha(k), \beta(k))$$

یک عمل از مجموعه اعمال A انتخاب می‌شود. این انتخاب به عنوان ورودی وارد محیط شده و محیط $\beta(k)$ را با مقدار مورد انتظار d_i ، اگر رابطه $\alpha(k) = \alpha_i$ برقرار باشد، به عنوان پاسخ برمی‌گرداند. این روال تکرار می‌شود تا بهترین عمل متناسب با محیط پیدا شود.

گام ۶-۱ به ازای هر ذره G_j^t که $j=1,2,\dots,N$ مقدارهای پاداش و جریمه هر کدام از اعمال به شکل زیر تعیین می‌گردد:

$$a_i^j = \text{reward} \times w + \text{reward} \times c_1 \times \text{rand1} \times \text{sign}[\chi(\alpha_i, G_{\text{best}_j}, G_j^t)] + \text{reward} \times c_2 \times \text{rand2} \times \text{sign}[\chi(\alpha_i, G_{\text{best}_j}, G_j^t)].$$

$$b_i^j = \text{reward} \times w + \text{reward} \times c_1 \times \text{rand1} \times \text{sign}[\chi(\alpha_i, G_j^t, G_{\text{best}_j})] + \text{reward} \times c_2 \times \text{rand2} \times \text{sign}[\chi(\alpha_i, G_j^t, G_{\text{best}_j})].$$

تابع $\text{sign}[\bullet]$ تابع نشانه است که اگر آرگومان ورودی آن صفر بود، صفر را و در غیر این صورت علامت آن را برمی‌گرداند. تابع $\chi(\alpha, s_1, s_2)$ بررسی می‌کند که عمل a بر روی s_2 مقدار اختلاف $\Delta = s_1 - s_2$ را کاهش می‌دهد یا خیر. اگر Δ با اعمال α بر روی s_2 کاهش یابد، مقداری مثبت و گرنه مقداری منفی را بر می‌گرداند، البته در مواقعی که Δ برابر صفر است، بدون توجه به نوع عمل a مقدار صفر را برمی‌گرداند. سایر ضرایب و متغیرها در بخش دوم در بهینه‌سازی گروهی شرح داده شده‌اند.

گام ۶-۲ به ازای تمام $j=1,2,\dots,N$ و $i=1,2,\dots,r$ مقدار احتمالی اعمال هر ذره G_j^t به شکل زیر تغییر می‌یابد.

for $\beta(t) = 0$:

$$p_i^j(t+1) = \begin{cases} p_i^j(t) + a_i^j(1 - p_i^j(t)), \\ (1 - a_i^j)p_k^j(t) \end{cases}, k \neq i$$

for $\beta(t) = 1$:

$$p_i^j(t+1) = \begin{cases} (1 - b_i^j)p_i^j(t), \\ \frac{b_i^j}{r-1} + (1 - b_i^j)p_k^j(t+1) \end{cases}, k \neq i$$

گام ۶-۳ با استفاده از بردار احتمالی هر ذره که در گام ۵-۲ تغییر یافت یکی از اعمال مجموعه $A = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ انتخاب شده و مقدار هر ذره از G_j^t به G_j^{t+1} تغییر می‌یابد.

گام ۶-۴ برای هر ذره مقدار G_{best_j} به‌روز می‌شود. همینطور بهترین موقعیت گروه یعنی G_{best} نیز به‌روز می‌شود.

گام ۶-۵ مقدار t را یک واحد افزایش داده و به گام ۵ برمی‌گردیم. به دلیل شباهت SLA2 با SLA1، ابتدا، اتوماتای یادگیر گروهی دوم را نیز شرح داده و سپس به تحلیل دو الگوریتم می‌پردازیم.

از منافع گروهی در اتوماتای یادگیر می‌توان با استفاده از حافظه اضافی و نگهداری بهترین جواب گروه و بهترین جوابی که هر عضو گروه تا لحظه کنونی به آن دست یافته است، در الگوریتم یادگیری که در اتوماتای یادگیر به عنوان قلب سیستم یادگیری محسوب می‌شود، استفاده کرد. البته می‌توان از این ویژگیهای گروهی مستقیماً در شدت عمل استفاده شده در لحظه بعدی نیز استفاده کرد.

بر پایه بحث انجام شده در بالا در این قسمت ما دو الگوریتم SLA1 و SLA2 را ارائه می‌دهیم. هر دو الگوریتم به جای یک جواب اولیه، از گروهی از جوابهای تصادفی اولیه با اندازه N که هر کدام از آنها یک ذره نامیده می‌شوند، استفاده می‌کنند. الگوریتم SLA1 بر روی طرح تقویتی عمل می‌کند، در واقع میزان پاداش و جریمه در الگوریتم یادگیری ذکر شده در رابطه (۲-۲) که به ترتیب با پارامترهای a و b نشان داده شدند و به صورت مقدار ثابتی به کار می‌روند، در SLA1 به صورت ترکیبی از یک مقدار ثابت و یک متغیر که به صورت پویا با توجه به موقعیت بهترین جواب گروه و بهترین جواب یافت شده توسط هر ذره تغییر می‌کند، به کار می‌رود. ایده الگوریتم SLA2 این است که به جای تغییر در میزان پاداش و جریمه‌ی طرح تقویتی مستقیماً بر روی شدت میزان عمل $\alpha(k)$ که خروجی اتوماتا و ورودی محیط در لحظه کنونی بوده و متناسب با $\beta(k-1)$ انتخاب شده است، عمل می‌نماید. در ادامه الگوریتم SLA1 را به صورت گام به گام آورده‌ایم.

۳-۱- الگوریتم گام به گام اتوماتای یادگیر گروهی

اول، SLA1

گام ۱ گروهی از ذره‌ها به تعداد N به صورت تصادفی مقداردهی می‌شوند و شمارشگر تکرار نسل، t ، برابر صفر قرار داده می‌شود. (فرض کنیم گروه با G و ذره j -ام در نسل t -ام با G_j^t نشان داده می‌شود.)

گام ۲ به ازای هر ذره G_j^t که $j=1,2,\dots,N$ ، مقادیر بردار احتمالاتی به صورت $p_i^j(t) = \frac{1}{r}$ که r تعداد اعمال اتوماتا و $i=1,2,\dots,r$ می‌باشد، تعیین می‌شود.

گام ۳ مقادیر reward و penalty به عنوان پایه پاداش و جریمه در اتوماتای یادگیر تعیین می‌شوند.

گام ۴ هر ذره توسط تابع هدف یا همان تابع برازش ارزیابی می‌شود.

گام ۵ برای هر ذره j که $j=1,2,\dots,N$ بهترین جواب به دست آمده تا تکرار t -ام تعیین می‌شود، ما آن را برای ذره j با G_{best_j} نشان می‌دهیم. همینطور بهترین موقعیت گروه نیز که با G_{best} نشان داده می‌شود انتخاب می‌گردد.

گام ۶ تا زمانی که شرایط پایانی ارضا نشده است گامهای زیر انجام می‌شوند:

راحتی آن را برای مسائل بهینه‌سازی با فضای جستجوی نامحدوب نیز به کار برد.

موضوع دیگری که در اینجا مطرح می‌شود و برای تمامی الگوریتم‌های ابتکاری وجود دارد، بحث میزان و نسبت پارامترهای الگوریتم می‌باشد. در واقع مشخص کردن مقدار دقیق آنها و یا نسبت میان آنها به صورت تئوری و بدون در نظر گرفتن مشخصات و ویژگیهای دقیق فضای جستجو کار غیر ممکن است. ولی توجه به تعداد تکرارها و وسعت فضای جستجو می‌توان آنها را به شکل کارآمدی مشخص کرد.

همانگونه که در آزمایش‌های انجام شده نیز ثابت شد، SLA1 در تعداد تکرارهای به اندازه کافی بهتر از SLA2 عمل می‌کند، ولی اگر جواب نهایی را در تعداد تکرارهای کوچکی مقایسه کنیم، آنگاه SLA2 بهتر خواهد بود.

واضح است که از نظر پیچیدگی محاسباتی هر تکرار از الگوریتم‌های SLA1 و SLA2 معادل N تکرار از اتوماتای یادگیر می‌باشد. هر چند میزان N بر اساس فضای جستجو تعیین می‌شود، ولی انتخاب مقدار ثابت بین ۱۰ و ۴۰ برای اندازه گروه کافیست. علاوه بر انتخاب مقدار ثابت اندازه جمعیت، می‌توان به جای M بار تکرار در اتوماتای یادگیر از M/N تکرار در اتوماتای یادگیر گروهی استفاده کرد تا نه تنها پیچیدگی محاسباتی آنها برابر شود بلکه دقیقاً حجم عملیاتی آنها نیز از این نظر یکی شود. هر دو الگوریتم ارائه شده به N برابر حافظه جهت ذخیره ذره‌های گروه و FN برابر جهت نگهداری بردارهای احتمال مخصوص به هر ذره نیاز دارند.

در ادامه الگوریتم‌های پیشنهادی بر روی دو نسخه متفاوت از یک مسئله به کار برده شده‌اند.

۵- نتایج آزمایشها و مقایسه الگوریتم‌های LA

SLA2 و SLA1

در این قسمت ما از دو نسخه متفاوت مسئله کوچکترین دایره پوشا یا همان مسئله 1-Center برای تست الگوریتم‌های گروهی استفاده می‌کنیم. در این مسئله n نقطه در فضای d بعدی وجود دارد و هدف یافتن کوچکترین گوی d بعدی است که بتواند تمام نقاط را در بر گیرد. یک جواب کامل برای این مسئله مشخصه‌های مرکز گوی که d بعدی است و شعاع گوی که یک عدد حقیقی است، می‌باشد. در حالت ساده این مسئله دارای فضای d بعدی کاملاً محدب و پیوسته می‌باشد که با افزایش بعد فضا پیچیدگی کار نیز افزایش می‌یابد. می‌توان با اضافه کردن محدودیتهایی برای موقعیت مرکز گوی مسئله را سخت‌تر کرد. برای ایجاد فضایی پیوسته و نامحدب از این مسئله، این قید را بر روی مسئله می‌گذاریم که مرکز گوی باید خارج از یک چندوجهی d بعدی از پیش تعیین شده قرار گیرد. دو شکل مختلف مسئله برای حالتی که d برابر ۲ است به ترتیب در شکل‌های (۴-۱) و (۴-۲) نشان داده شده

۲-۳- الگوریتم اتوماتای یادگیر گروهی دوم، SLA2

در این قسمت به دلیل توضیحاتی که در بخشهای قبل داده شد از آوردن الگوریتم گام به گام اتوماتای یادگیر گروهی دوم صرف‌نظر کرده و به توضیحات زیر بسنده می‌کنیم.

در الگوریتم SLA2 مقادیر احتمالی اعمال هر ذره که در SLA1 تغییر داده می‌شد، ثابت مانده و فقط آنها را با استفاده از طرحهای تقویتی اتوماتای یادگیر که نمونه خطی آن در رابطه (۲-۲) آمده است به روز می‌کنیم. به استثنای اینکه پس از انتخاب یک عمل مانند a_j برای تغییر ذره G_j ، به تاثیر این عمل در دور شدن یا نزدیک شدن آن ذره به دو جواب G_{best} (بهترین مقدار یافت شده توسط ذره G_j تا لحظه کنونی) و همینطور G_{best} (بهترین جواب کشف شده توسط گروه تا لحظه کنونی) دقت بیشتری می‌شود. به بیان دقیقتر پس از انتخاب عمل a_j برای تغییر ذره G_j ، میزان a_j در ضریب f_j که از رابطه زیر محاسبه می‌شود، ضرب شده و سپس بر روی G_j اعمال می‌شود.

$$f_j = w + c_1 \times \text{rand1} \times \text{sign}[\chi(\alpha_j, G_{best_j}, G_j^t)] + c_2 \times \text{rand2} \times \text{sign}[\chi(\alpha_j, G_{best_j}, G_j^t)]. \quad (2-3)$$

توابع و متغیرهای به کار برده شده در رابطه بالا همان مفاهیم رابطه (۳-۱) را دارند.

پس از تعیین ضریب، عمل $f_j \alpha_j$ بر روی ذره G_j اعمال می‌شود.

۴- آنالیز الگوریتم‌های SLA1 و SLA2

واضح است که تغییرات اعمال شده در بدنه اتوماتای یادگیر، هر ذره را علاوه بر تجربیات خود آن ذره از تجربیات گروه نیز برخوردار می‌سازد، این تجربه اضافی مربوط به مشخص شدن بهترین موقعیت کشف شده توسط گروه می‌باشد. که کمک می‌کند هر ذره به سمت آن جهت‌گیری نماید. در واقع جهت‌گیری نهایی یک ذره که با انتخاب عمل خاصی در SLA1 و شدت عمل انجام شده در SLA2 صورت می‌گیرد، با توجه به سه عنصر بهترین موقعیت گروه، محتمل‌ترین عمل با توجه به بردار احتمالاتی ذره و بهترین موقعیت کشف شده توسط خود ذره انجام می‌گردد. علاوه بر این موفقیت یک اتوماتای یادگیر بسیار وابسته به موقعیت اولیه که عموماً به صورت تصادفی انتخاب می‌شود، می‌باشد و از اینرو در حالت کلی حتی اگر یک اتوماتای یادگیر در بهینه‌های محلی گیر نکند بازهم یافتن جواب بهینه عمومی برای آن کار مشکلی خواهد بود. این مشکلات در هر دو الگوریتم ارائه شده با انتخاب N موقعیت تصادفی و به اشتراک گذاری اطلاعاتشان حل شده است و می‌توان به

۶. در مواقعی که مقادیر پارامترهای SLA2 از رابطه خطی $w + c_1 \times \text{Rand1} + c_2 \times \text{Rand2} \approx 3$ پیروی می‌کنند، نتایج بهتری ظاهر می‌شود.

۷. هر چند محققان الگوریتم PSO بیشتر مقدار ۲ را برای هر دو پارامتر c_1 و c_2 پیشنهاد می‌کنند، ولی در آزمایشات ما مقادیرهای $c_1 = 0.6$ و $c_2 = 1.4$ نتایج بهتری را بویژه در حل مسئله اول، موجب می‌شوند.

۶- نتیجه‌گیری

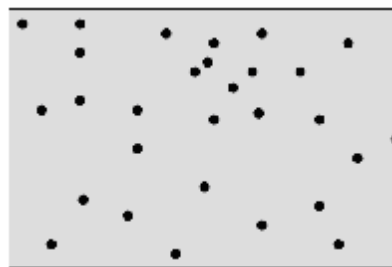
در این مقاله دو الگوریتم گروهی برای اتوماتای یادگیر با عناوین اتوماتای یادگیر گروهی اول و دوم پیشنهاد شدند. از آنجایی که دو الگوریتم پیشنهادی کم‌وبیش شبیه هم بوده و نتایج تقریباً مشابهی را نیز در آزمایش‌های انجام شده داشتند، در اینجا آنها را به صورت مشخصی از یکدیگر جدا نساخته و با عنوان اتوماتای یادگیر گروهی از هر دو آنها یاد کردیم. کارایی بهتر این دو الگوریتم از نظر قدرت و گریز از بهینه‌های محلی به دلیل به‌اشتراک گذاری اطلاعات گروهی، در مقابل اتوماتای یادگیر کاملاً روشن است.

اتوماتای یادگیر گروهی می‌تواند بر روی هر نوع طرح تقویتی پیاده‌سازی شود که ما در اینجا آن را با طرح تقویتی خطی سازگار کردیم. موضوع تحقیقی که باقی می‌ماند سازگاری این روش گروهی با سایر الگوریتم‌های یادگیری اتوماتا می‌باشد.

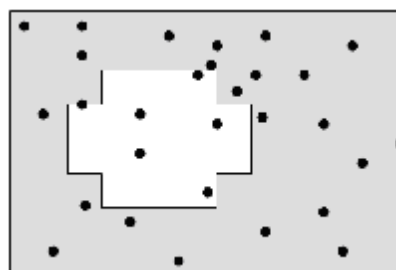
مراجع

- [1] M. Loomans, H. Visser, *Application of the genetic algorithm for optimization of large solar hot water systems*, Solar Energy 72 (5) (2002) 427–439.
- [2] S. Dehuri, R. Mall, *Predictive and comprehensible rule discovery using a multi-objective genetic algorithm*, Knowledge-Based Systems 19 (2006) 413–421.
- [3] A. Hidenori, M. Yoshibumi, *Voltage and Reactive Power Control by Particle Swarm Optimization Considering Control Process*, Electrical Engineering in Japan, Vol. 151, No. 1, 2005, Translated from Denki Gakkai Ronbunshi, Vol. 124-B, No. 1, January 2004, pp. 95–102.
- [4] Y. Fukuyama, *Comparative Studies of Particle Swarm Optimization Techniques for Reactive Power Allocation Planning in Power Systems*, Electrical Engineering in Japan, Vol. 153, No. 1, 2005, pp. 690–696.
- [5] X. Zeng, Z. Liu, *A learning automata based algorithm for optimization of continuous complex functions*, Information Sciences 174 (2005) 165–175.
- [6] Aoki, T., T. Suzuki, and S. Okuma, "Acquisition of Optimal Action Selection in Autonomous Mobile Robot Using Learning Automata (Experimental Evaluation)," Proceedings of the IEEE Conference on Fuzzy Logic and Neural Networks/Evolutionary Computation, pp.56-63, Nagoya, Japan, Nov. 1995.
- [7] Gilbert, V., J. Thibault, and K. Najim, "Learning Automata for the Control and Optimization of a Continuous Stirred Tank Fermenter," IFAC Symposium

است. ناحیه خاکستری ناحیه موجه جواب می‌باشد. توجه کنید که محیط چند وجهی دارای نواحی بهینه محلی نسبتاً زیادی، بویژه برای مقادیرهای بزرگ d می‌باشد.



شکل (۴-۱): فضای محدب و پیوسته



شکل (۴-۲): فضای پیوسته و نامحدب

در حل مسئله دوم ما از راهکار تابع جریمه^{۱۴} که روش بسیار محبوبی در الگوریتم‌های تکاملی برای ارضا محدودیتها می‌باشد استفاده کرده‌ایم. برای مقایسه دقیقتر این سه الگوریتم همانگونه که در قسمت آنالیز SLA بیان شد، در تمام اجراها اگر از اندازه جمعیت N و تعداد تکرار Rep در SLA1 و SLA2 استفاده کرده‌ایم، اتوماتای یادگیر تا $N \times \text{Rep}$ تکرار اجرا شده است. نتایج مهم حاصل از اجرای آزمایشها بدین شرح است:

۱. در تمامی اجراها هر دو الگوریتم SLA1 و SLA2 جوابهای بسیار بهتری را نسبت به LA به دست آورده‌اند.
۲. با انتخاب c_2 بزرگتر در حل مسئله اول برای SLA1 و SLA2 جواب بهتری حاصل می‌شود.
۳. اگر تعداد تکرارها کوچکتر شوند، الگوریتم SLA2 نسبت به SLA1 جوابهای بهتری را بدست می‌آورد، این نتیجه برای تکرارهای بیشتر به صورت عکس می‌باشد.
۴. اگر در حالتی که تمام پارامترهای هر سه الگوریتم ثابت است، تعداد نقطه‌های ورودی مسئله را بیشتر کنیم و یا بعد نقاط (d) را بزرگتر کنیم، کارایی LA نسبت به دو الگوریتم SLA بسیار پایین‌تر می‌آید.
۵. به طور میانگین اگر نسبت جواب LA به SLA1 را در مسئله اول برای $d = 20$ مقدار $1/8$ باشد در مسئله دوم این مقدار در حدود $5/2$ یعنی حدوداً سه برابر می‌باشد.

- Proceedings of IEEE International Symposium on Intelligence Control, pp. 974-979, 2003.
- [29] K. Yasuda, A. Ide and N. Iwasaki. *Adaptive Particle Swarm Optimization*. In Proceedings of IEEE International Conference on Systems, Man and Cybernetics, pp. 1554-1559, 2003.
- [30] I. Trelea. *The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection*. Information Processing Letters, vol. 85, no. 6, pp. 317-325, 2003.
- [31] A. IDE , K. YASUDA , *A Basic Study of Adaptive Particle Swarm Optimization*, Electrical Engineering in Japan, Vol. 151, No. 3, 2005, pp. 550-557.
- [32] Clerc M, Kennedy J. *The particle swarm: Explosion, stability, and convergence in a multi-dimensional complex space*. IEEE Trans Evolutionary Computation, Vol. 6, Issue 1, p 58-73, 2002.
- [33] S. Mikki, A. Kishk, *Improved Particle Swarm Optimization Technique Using Hard Boundary Constrains*, MICROWAVE AND OPTICAL TECHNOLOGY LETTERS / Vol. 46, No. 5, September 5, 2005.
- [34] Naruse, K., and Y. Kakazu, "Strategy Acquisition of Path Planning of Redundant Manipulator using Learning Automata", *IEEE International Workshop on Neuro-Fuzzy Control*, pp.154-159, 1993.
- [35] Bush, R. R., and F. Mosteller, *Stochastic Models for Learning*, New York: Wiley, 1958.
- on Adaptive systems in Control and Signal Processing, Grenoble, France, July 1992.
- [8] Marsh, C., and Gordon T. J., "The Application of Learning Automata to Controller Design in Slow-Active Automobile Suspensions," International Journal for Vehicle Mechanics and Mobility, vol. 24, no. 8, pp. 597-616, 1995.
- [9] K. Najim, A.S. Poznyak, *Learning Automata: Theory and Applications*, Pergamon Press, 1994.
- [10] A. S. Poznyak, K. Najim, *Learning Automata and Stochastic Optimization*, Springer, 1997.
- [11] K.S. Narendra, M.A.L. Thathachar, *Learning Automata: An Introduction*, Prentice Hall, 1989.
- [12] G.I. Papadimitriou, *Hierarchical discredited pursuit nonlinear learning automata with rapid convergence and high accuracy*, IEEE Transactions on Knowledge and Data Engineering 6 (4) (1994) 654-659.
- [13] R.C. Eberhart, P.K. Simpson and R.W. Dobbins (1996). *Computational Intelligence PC Tools*, Academic Press Professional, Boston, MA.
- [14] Kennedy and R.C. Eberhart (1995). *Particle Swarm Optimization*. Proc. IEEE International Conference on Neural Networks, Piscataway, NJ, IV:1942-1948.
- [15] Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*, Wiley, 2001.
- [16] Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, 1989.
- [17] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*. AI Series, Springer Verlag, New York, 1994.
- [18] P. Van Laarhoven and E. Aarts. *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers, 1987.
- [19] F. Glover. *Tabu Search – Part I*. ORSA Journal on Computing, vol. 1, no. 3, pp. 190-206, 1989
- [20] J. Kennedy, and W.M. Spears. "Matching Algorithms to Problems: An Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multimodal Problem Generator." Proceedings (f t h e IEEE Inf'l Conference on Evolutionary Computation. 1998
- [21] G. Venter and J. Sobieszczanski-Sobieski, *Particle Swarm Optimization*. In the 43rd AIAA/ASME/ASCE/AHA/ASC Structures, Structural Dynamics and Materials Conference, Denver, Colorado, USA, 2002.
- [22] I. d. F. V. Antonio, *Optimization Of Nonlinear Constrained Particle Swarm*, Úkio Technoginis IR Ekonominis VYSTYMAS Technological and Economic Development of Economy, 2006, Vol XII, No 1, 30-36
- [23] Y. Shi and R. Eberhart. *A Modified Particle Swarm Optimizer*. In Proceedings of the IEEE International Conference on Evolutionary Computation, Piscataway, New Jersey, pp. 69-73, 1998.
- [24] C. E. Russell, S. Yuhui, *Particle Swarm Optimization: Developments, Applications and Resources*. 2001 IEEE.
- [25] S. Yuhui, C. E. Russell, Empirical study of Particle Swarm Optimization. 1999 IEEE.
- [26] S. Yuhui, C. E. Russell, Fuzzy Adaptive Particle Swarm Optimization, 2001 IEEE.
- [27] F. Van den Bergh and A.P. Engelbrecht. *A New Locally Convergent Particle Swarm Optimizer*. In Proceedings of the IEEE Conference on Systems, Man, and Cybernetics, Hammamet, Tunisia, 2002.
- [28] Y. Zheng, L. Ma, L. Zhang and J. Qian. *Robust PID Controller Design using Particle Swarm Optimizer*. In

زیر نویس ها

¹ Pareto optimal

² Swarm Learning Automata

³ Evolutionary Algorithms

⁴ Inertia weight

⁵ Exploration and Exploitation

⁶ Cognitive and Social parameters

⁷ Explosion phenomenon

⁸ Environment

⁹ Feedback

¹⁰ Actions

¹¹ Reward set

¹² Stationary automata

¹³ Reinforcement scheme

¹⁴ Penalty function