LAEDA: يك الگوريتم تكاملي جديد با استفاده از اتوماتاي يادگير

رضارستگار   محمد رضا ميبدي

rrastegar@ce.aut.ac.ir   meybodi@ce.aut.ac.ir

آزمايشگاه محاسبات نرم - دانشكده مهندسي كامپيوتر و فناوري اطلاعات - دانشگاه صنعتي اميركبير

چكيده: در سالهای اخير رويكرد جديدی به منظور حل مشكلات الگوريتمهای تكاملي[1] بويژه الگوريتمهای ژنتيكي[2] مورد توجه محققين قرار گرفته است. اين رويكرد مبتني برايجاد مدلهای احتمالاتي از ژنومها و اجزای سازنده آنها مي باشد. تا كنون الگوريتمهای متنوعي بر اين اساس ارائه شده اند كه اگر چه برخي از سادگي الگوريتمهای ژنتيكي برخوردار نيستند، اما در حل مسائل با موفقيت بيشتری روبرو بوده اند. در اين مقاله رهيافت ديگری از اين الگوريتمها را بر اساس اتوماتای يادگير[3] معرفي و مورد بررسي قرار مي دهيم. در اين رهيافت مدل احتمالاتي اجزای سازنده مساله بوسيله اتوماتای يادگير و بر اساس ژنومهای نسل توليد شده تخمين زده مي شود.الگوريتم پيشنهادی بسيار ساده و برای مسائل مورد بررسي در اين مقاله دارای كارايي خوبي مي باشد.

كلمات كليدی: الگوريتم تخمين توزيع، اتوماتای يادگير، LAEDA

۱- مقدمه

نياز به جستجو در حل مسائل كاربردي امري غير قابل اجتناب و در عين حال دشوار است. به همين جهت تعداد زيادي از الگوريتمهاي جستجو با فلسفه ها و دامنه استفاده متفاوت بوجود آمده اند. اين الگوريتمهاي جستجو را مي توان به دو دسته كلي جستجوهای كامل و جستجوهای مكاشفه ای تقسيم كرد. فرق اساسي بين اين دو دسته در اين است كه در جستجوهای كامل، تمام فضای جستجو به طور كامل مورد جستجو و ارزيابي قرار مي گيرد تا جواب مورد نظر يافته شود، در حاليكه در جستجوهای مكاشفه ای تنها بخشي از فضا كه احتمال يافتن جواب در آن بيشتر است، مورد توجه قرار مي گيرد. جستجوهای مكاشفه ای به دو دسته الگوريتمهای قطعي و غيرقطعي تقسيم پذيرند. ويژگي اصلي الگوريتمهای قطعي در اين است كه تحت شرايط يكسان، جوابهای يكسان مي دهند. از جمله اين الگوريتمها مي توان به تپه نوردی[4] اشاره كرد. ايراد اساسي اين الگوريتمها احتمال گير افتادن در مينيمم های محلي مي باشد. در مقابل، الگوريتمهای مكاشفه ای غير قطعي با استفاده از احتمالات و جستجوهای تصادفي، در شرايط يكسان جوابهای متفاوتي بدست مي آورند. همينطور در صورت افتادن در مينيمم های محلي، از آنها مي گريزند. الگوريتمهای مكاشفه ای غير قطعي را بر اساس تعداد جوابهايي كه در هر تكرار بررسي و ذخيره مي كنند به دو دسته تقسيم مي كنند. بعضي مانند تابكاری فلزات تنها يک جواب را در هر تكرار مورد بررسي قرار داده و ذخيره مي كنند. بعضي ديگر در هر تكرار دسته ای از جوابها را ذخيره مي كنند، كه به اين الگوريتمهای مبتني بر جمعيت مي گويند. دسته بزرگي از اين الگوريتمها را الگوريتمهاي ژنتيكي تشكيل مي دهد. الگوريتمهای ژنتيكي كلاسيك به دليل سادگي و عدم نياز به معادلات

---

[1] Evolutionary Algorithms
[2] Genetic Algorithms
[3] Learning Automata
[4] Hill Climbing

ديفرانسيل پيچيده در حل مسائل پيچيده با فضای جستجوی غير هموار مورد استفاده قرار مي گيرند. اين الگوريتمها بر پايه انتخاب و تركيب مجدد[5] جوابهاي مجاز مساله كار مي كنند. مجموعه جوابهاي توليد شده در هر دوره الگوريتم را يك نسل و هر كدامها از جوابها را يك ژنوم يا كروموزوم مي نامند. هر ژنوم تركيبي از متغيرهاي مساله است. در بعضي مسائل متغيرهاي مساله به هم وابسته و در بعضي ديگر از هم مستقل هستند. ولي همواره ارتباط و تبادل اطلاعات بين ژنومها از طريق انتخاب و تركيب مجدد ژنومها در يك نسل صورت مي پذيرد. اين جابجايي اطلاعات سبب مي شود تا جوابهاي جزئي با يكديگر تركيب و احتمالا جوابهايي با كيفيت بالاتر بدست آيند. اما با تمام ويژگيهاي مثبتي كه در الگوريتم ژنتيكي استاندارد وجود دارد، اين الگوريتم تنها در مواقعي كه متغيرها از هم مستقل و يا در فاصله كمي از هم در ژنوم قرار گرفته باشند، كارايي مناسبي دارد. به عبارت ديگر رفتار الگوريتم ژنتيكي وابستگي شديدی به پارامترهايی از جمله نحوه تعريف عملگرهای جهش و توليد نسل، احتمال جهش و توليد نسل، اندازه جمعيتها و تعداد نسلهای توليد شده دارد. به همين دليل گاهي در تركيب ژنومها با يكديگر نه تنها بهبودي در كيفيت در جوابها حاصل نمي شود، بلكه الگوريتم در نقاط بهينه محلي به دام مي افتد. از اين رو براي اجتناب از اين مشكل، رهيافت جديد با نام الگوريتم تخمين توزيع معرفي شد كه در آن ژنومهاي جديد بجاي استفاده از عملگرهاي مرسوم در الگوريتم ژنتيكي با استفاده از توزيع احتمالاتي تمام راه حلهاي مجاز نسل قبل ساخته مي شوند.. الگوريتم تخمين توزيع براي اولين بار در [۱] در رده الگوريتمهاي تكاملي مطرح شد و از زمان طرح آن الگوريتمهاي متنوعي از آن بوجود آمده اند.

آنچه كه در اين مقاله به معرفي آن خواهيم پرداخت، الگوريتم جديدی از الگوريتمهای تخمين توزيع با استفاده از اتوماتای يادگير مي باشد. در اين الگوريتم به منظور ايجاد مدل احتمالاتي متغيرهاي تشكيل دهنده ژنوم از اتوماتاي يادگير استفاده مي كنيم. در ادامه مقاله در ابتدا در بخش ۲ به بررسي فرم كلي الگوريتمهاي تخمين توزيع خواهيم پرداخت. در بخش (۳) به صورت كوتاه مروری بر اتوماتای يادگير و به خصوص اتوماتای يادگير با ساختار متغير خواهيم پرداخت. سپس در بخش (۴) الگوريتم مورد نظر خود را با تفضيل بيشتری بيان خواهيم كرد. در ادامه در قسمت (۵) چند مسئله خطي و تركيبي را به منظور بررسي كارايي الگوريتم پيشنهاد شده، مورد بررسي و نتايج آن را با نتايج بدست آمده از الگوريتم ژنتيكي ساده و الگوريتم تخمين توزيع UMDA[6] مقايسه خواهيم كرد. در پايان نتيجه گيری و كارهای آينده را بيان مي كنيم.

۲- الگوريتم تخمين توزيع

با توجه به مطالب گفته شده، دسته ای از الگوريتمهای مكاشفه ای غير قطعي مبتني بر جمعيت با نام الگوريتمهای تخمين توزيع ارائه شده اند كه همانند الگوريتمهای ژنتيكي نيازمند به فضای جستجوی هموار و معادلات ديفرانسيل پيچيده نمي باشند و علاوه بر اين بسياری از مشكلات الگوريتمهای ژنتيكي را حل نموده اند. در الگوريتمهای تخمين توزيع با ساخت يک مدل احتمالاتي از اجزای سازنده ژنوم، سرعت پيشروی به سوی جواب بهينه مسئله افزايش مي يابد. در اين الگوريتمها جمعيتهای جديد با استفاده از عملگرهای جهش و توليد نسل بوجود نمي آيند. بلكه ژنومهای جديد، بر اساس توزيع احتمالاتي تخمين زده شده بر اساس ژنومهای انتخاب شده از نسلهای قبل، نمونه برداری و ساخته مي شوند. در واقع تخمين همين تابع توزيع مشكلترين بخش الگوريتمهای تخمين توزيع است. غالب كلي همه الگوريتمهای تخمين توزيع به صورت زير است :

قدم ۱- در ابتدا نخستين جمعيت، D_0، با تعداد N ژنوم تشكيل مي شود. معمولا توليد N ژنوم مذكور، با فرض توزيع يكنواخت بروی هريک از متغيرها صورت مي پذيرد. سپس هر يک از ژنومها ارزيابي مي شوند.

قدم ۲- يک تعداد مشخص، $Se \le N$، ژنوم از ميان جمعيت حاضر، D_l، بر اساس يک معيار تعريف شده (معمولا بهترينها) انتخاب مي شوند كه با D_l^{Se} مشخص مي شود.

قدم ۳- يک مدل احتمالاتي n بعدی كه نشان دهنده توزيع n متغير موجود راه حل مي باشد، تخمين زده مي شود.

قدم ۴- در پايان يک جمعيت جديد متشكل از N ژنوم جديد با استفاده از توزيع احتمالاتي بدست آمده در مراحل قبل، ايجاد مي شود.

---

[5] Recombination

[6] Univariate Marginal Distribution Algorithm

قدمهای ۲ تا ۴ الگوریتم فوق تا برقراری شرط خاتمه تکرار مي شوند. شرایط خاتمه متفاوتي مانند تولید تعداد معيني نسل، انجام تعداد معيني ارزیابي، یکنواختي در چندین نسل پایاني و ثابت ماندن تقریبي ارزش ژنومها در چند نسل پیاپي را مي توان در نظر گرفت. در شکل (۱) مي توانید نمایي از نحوه عملکرد الگوریتم تخمين توزیع را مشاهده کنید[۲].

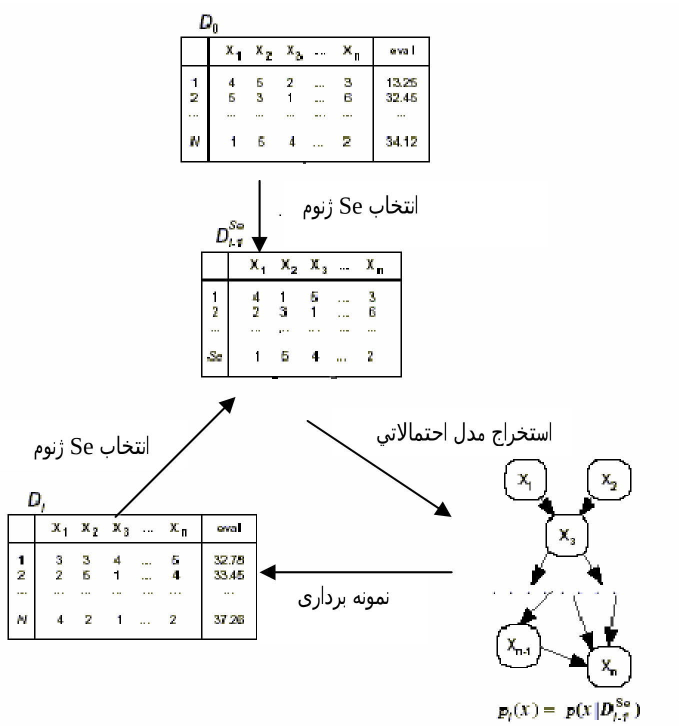
اگر فرض کنید $X_i, i=1,...,n$ یک متغیر تصادفي و x_i یک مقدار برای X_i باشد. در این صورت $X=(X_1,...,X_n)$ یک متغیر تصادفي n بعدی است که مي تواند $x=(x_1,...,x_n)$ را به عنوان مقدار خود بپذیرد. مدل گرافيکي X دارای دو جز تشکیل دهنده ساختار و یک مجموعه از چگالیهای احتمالاتي محلي است. به طوري که ساختار S بصورت یک گراف جهت دار بدون دور تعریف مي شود. این گراف نشان دهنده ساختار وابستگي بين متغيرهای X_i است. اگر در این ساختار Pa_i^S به معنای متغیرهای والد متغیر X_i باشد، آنگاه متغیر X_i از تمام متغيرهای $\{X_1,..,X_n\} \setminus Pa_i^S$ مستقل بوده و بنابراین میتوانیم تابع چگالي عمومي توام[۷] $\rho(X=x)$ را به صورت زیر بنویسیم:

$$\rho(X=x)=\rho(x_1,...x_n)=\prod_{i=1}^{n}\rho(x_i|Pa_i^S) \tag{۱}$$

از طرفي با فرض این که چگالیهای احتمالاتي محلي به مجموعه پارامترهای $\theta_S \in \Theta_S$ وابسته است بنابراین:

$$\rho(X|\theta_S)=\prod_{i=1}^{n}\rho(x_i|Pa_i^S,\theta_i) \tag{۲}$$

که $\theta_S=(\theta_1,...,\theta_n)$ است. بنابراین مدل گرافيکي احتمالاتي X به صورت $M=(S,\theta_S)$ میباشد. بر اساس نوع ارتباط بين متغيرها دسته های متفاوتي برای این مدلها تشخیص داده مي شوند. نوع ارتباطات به سه دسته تقسیم مي شوند که عبارتند از: مستقل، ارتباط دوتایي و ارتباط چندتایي.



شکل ۱- نحوه عملکرد الگوریتم تخمين توزیع

در دسته اول، متغيرها از هم مستقل هستند و بنابراین احتمال یک نمونه x، برابر حاصلضرب احتمال متغيرهای آن است. به عبارت دیگر داریم:

$$P(X=x)=\prod_{i=1}^{n}P(X_i=x_i) \tag{۳}$$

---

[7] Joint Generalized Probability Density Function

این مدل، ساده ترین مدل ممکن است و با در نظر گرفتن درصدی از خطا مي توان از آن برای حل مسائل بهينه سازی غيرخطي استفاده کرد. از الگوريتمهای اين دسته مي توان به UMDA [۳]، PBIL[8][٤و٥] وCGA[9] [٦] اشاره کرد. دو الگوريتم آخری فقط برای ژنومهای با متغيرهای دودويي (که تنها مقادير صفر و يک را مي پذيرند) بکار مي روند. از آنجايي که الگوريتم ارائه شده در اين مقاله جز دسته الگوريتمهاي مدل اول محسوب مي شود، در ادامه بخش به توضيح مختصر الگوريتمهاي مدل اول مي پردازيم. در UMDA توزيع احتمالاتي توام[10] ژنوم با استفاده از فراواني حاشيه اي[11] هر کدام از متغيرهاي ژنوم به صورت زير بدست مي آيد.

$$P_l(x_i) = \sum_{j=1}^{Se} \delta_j (X_i = x_i \mid D_{l-1}^{Se}) \Big/ Se \qquad (۴)$$

به طوريکه اگر متغير X_i ژنوم jام مجموعه D_{l-1}^{Se} مقدار x_i داشته باشد $\delta_j(X_i = x_i \mid D_{l-1}^{Se}) = 1$ مي باشد و در غير اين صورت مقدار آن صفر است. ثابت شده است که از نقطه نظر تئوري $UMDA$ رفتاري همانند الگوريتم ژنتيکي ساده خود نشان مي دهد[۷].

در $PBIL$ جمعيت ژنومها با استفاده از يك بردارِاحتمال $P_l(x) = (P_l(x_1),...,P_l(x_n))$ بازنمايي مي شود. به اين ترتيب که $P_l(x_i)$ احتمال قرار گرفتن مقدار ۱ در متغير i ام نسل l ام مي باشد. در هر نسل N ژنوم بر اساس بردار احتمال ساخته شده و Se تا از بهترين آنها انتخاب مي شوند. بردار احتمال بر اساس ژنومهاي انتخابي و با استفاده از يك قانون مشابه قانون هب بروز درآورده مي شود[٤].

$$P_{l+1}(x_i) = (1-\alpha)P_l(x_i) + \alpha\left(\sum_{k=1}^{Se} x_{i,k:N}^{l} \Big/ Se\right) \qquad (۵)$$

به صورتي که $\alpha \in (0,1]$ و $x_{i,k:N}^{l}$ برابر با مقدار x_i در k امين ژنوم انتخاب شده در نسل l ام مي باشد. در الگوريتم CGA بردار احتمال ژنومها، $P_l(x) = (P_l(x_1),...,P_l(x_n))$ ، را برابر $(0.5,...,0.5)$ قرار مي دهيم. دو ژنوم بر اساس بردار ساخته مي شود و بر اساس تابع ارزيابي مرتب مي کنيم، ژنوم با ارزش تر را winner و ژنوم ديگر را loser مي ناميم. سپس بر اساس رابطه زير بردار احتمال را بروز در مي آوريم و اين فرايند را تا همگرا شدن بردار احتمال ادامه مي دهيم[٦].

$$P_{l+1}(x_i) = \begin{cases} P_l(x_i) + 1/n & loser[i] \neq winner[i] \ \& \ winner[i] = 1 \\ p_l(x_i) - 1/n & loser[i] \neq winner[i] \ \& \ winner[i] = 0 \\ P_l(x_i) & loser[i] = winner[i] \end{cases} \qquad (۶)$$

اين الگوريتمها در مواقعي که مسائل مورد نظر داراي متغيرهاي کاملا مستقل باشند، بسيار سريع به جواب بهينه دست پيدا مي کنند. اما هر چه به وابستگي متغيره افزوده شود از کارايي اين الگوريتمها کاسته مي شود.

۳- اتوماتای یادگیر

اتوماتاي يادگير يك ماشين با حالات محدود[12] است که تعداد محدودي عمل را مي‌تواند انجام دهد. هر عمل انتخاب شده توسط محيطي احتمالي ارزيابي مي‌گردد و پاسخي به اتوماتای يادگير داده مي‌شود. اتوماتاي يادگير از اين پاسخ استفاده نموده و عمل بعدي را انتخاب مي‌کند. در طي اين فرايند، اتوماتاي يادگير ياد مي‌گيرد که چگونه بهترين عمل را انتخاب نمايد. محيط را مي توان توسط سه تايي $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha = \{\alpha_1,...,\alpha_r\}$ مجموعه ورودیها، $\beta = \{\beta_1,...,\beta_m\}$ مجموعه خروجيها و $c = \{c_1,...,c_r\}$ مجموعه احتمالهای جريمه مي باشد. هرگاه β مجموعه دو عضوی باشد محيط از نوع P مي باشد. در چنين محيطي $\beta_1 = 1$ به عنوان جريمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته مي شود. در محيط Q، $\beta(n)$ مي تواند به طور گسسته يک مقدار از مقادير محدود در فاصله $[0,1]$ و در محيط از نوع S، متغير تصادفي در فاصله $[0,1]$ باشد. c_i احتمال اينکه عمل α_i نتيجه نامطلوب داشته باشد، مي باشد. در محيط ايستا مقادير c_i بدون تغيير مي مانند، حال آنکه در محيط غير ايستا اين مقادير در طي زمان تغيير مي کنند . اتوماتای يادگير به دو

---

[8] Population-Based Incremental Learning
[9] Compact Genetic Algorithm
[10] Joint Probability Distribution
[11] Marginal Frequency
[12] Finite State Machine

دسته با ساختار ثابت و ساختار متغیر تقسیم مي شوند. از آنجا که در این تحقیق بیشتر تمرکز بروی اتوماتای با ساختار متغیر است، در ادامه این بخش به این نوع خواهیم پرداخت. اتوماتای یادگیر با ساختار متغیر توسط چهارتایي $\{\alpha, \beta, p, T\}$ نشان داده مي شود که در آن $\alpha = \{\alpha_1, ..., \alpha_r\}$ مجموعه عملهای اتوماتا، $\beta = \{\beta_1, ..., \beta_m\}$ مجموعه ورودیهای اتوماتا، $p = \{p_1, .., p_r\}$ بردار احتمال انتخاب هریک از عملها و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری مي باشد. در این نوع اتوماتاها، اگر عمل α_i در مرحله n ام انتخاب شود و پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمالها کاهش مي یابند. برای پاسخ نامطلوب احتمال $p_i(n)$ کاهش یافته و سایر احتمالات افزایش مي یابند. و در هر حال، تغییرات به گونه‌ای صورت مي پذیرد تا حاصل جمع $p_i(n)$ همواره مقدار ثابت یک باقي بماند. با توجه به مدل مارکوفي[13] اتوماتاي یادگیر، این یادگیرنده‌ها به دو دسته ارگودیك [14] و جذب شونده تقسیم مي شوند. اتوماتاي ارگودیك به یك توزیع مستقل از حالت اولیه همگرا مي شود. به همین دلیل در محیطهاي غیر ایستا این اتوماتاها کاربرد بیشتري دارند. اتوماتاي با نقاط جذب شونده بعد از چندین مرحله در یك حالت قفل شده و از آن بیرون نمي‌آید. و همین ویژگي سبب مي شود تا کاربرد آنها به محیطهاي ایستا یا نا ایستا با تغییرات بسیار کم، محدود شود. الگوریتم زیریک نمونه از الگوریتمهای یادگیری خطي است. پاسخ مطلوب:

$$
\begin{aligned}
p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\
p_j(n+1) &= (1-a)\, p_j(n) \qquad \forall j \quad j \neq i
\end{aligned}
\tag{۷}
$$

و پاسخ نامطلوب:

$$
\begin{aligned}
p_i(n+1) &= (1-b)\, p_i(n) \\
p_j(n+1) &= (b/r - 1) + (1-b)\, p_j(n) \quad \forall j \quad j \neq i
\end{aligned}
\tag{۸}
$$

در روابط (۷) و (۸)، a پارامتر پاداش و b پارامتر جریمه مي باشند. با توجه به مقادیر a و b سه حالت زیر را مي توان در نظر گرفت. حالت۱- زمانیکه a و b با هم برابر باشند، الگوریتم را L_{RP} مي نامیم. این اتوماتا از نوع اتوماتاي ارگودیك مي باشد. حالت۲-زمانیکه b از a خیلي کوچکتر باشد، الگوریتم را $L_{R\varepsilon P}$ مي نامیم.حالت۳- زمانیکه b مساوي صفر باشد. الگوریتم را L_{RI} مي نامند که از دسته اتوماتاهاي جذب شونده به یك حالت خاص مي باشد. نوع دیگری از اتوماتای یادگیر با ساختار متغیرکه از آن برای تست کارایي الگوریتم خود بهره خواهیم برد، اتوماتای *Pursuit* است. این اتوماتا یک اتوماتای غیر خطي مي باشد که با سرعت بیشتری نسبت به L_{RI} به نقاط جذب همگرا مي شود[۸]. برای مطالعه دقیقتر مي توانید به [۹] مراجعه کنید.

۴- الگوریتم تخمین توزیع با استفاده از اتوماتای یادگیر

الگوریتم ارائه شده، که آنرا از این پس تحت نام *LAEDA* [15] مي شناسیم، از الگوریتمهای قابل اعمال به مدل اول (متغیرهای مستقل) مي باشد. در این الگوریتم فرض بر مستقل بودن متغیرهای ژنومها مي باشد و به ازای هر متغیر در ژنوم یک اتوماتای یادگیر استفاده مي شود. تعداد عملهای اتوماتای یادگیرنده برابر با تعداد مقادیر مجاز برای متغیر متناظر با آن مي باشد. برای ساختن هر نمونه ژنوم در ابتدا از اتوماتای یادگیر هر متغیر در خواست مي شود تا به عمل مورد نظر خود را انتخاب کند، سپس به متغیر متناظر آن، مقدار متناظر عمل انتخاب شده را انتساب میدهیم. بنابراین احتمال ساخته شدن یک ژنوم $x = (x_1, ..., x_n)$ از رابطه زیر قابل محاسبه است.

$$
P(X = x) = \prod_{i=1}^{n} P(X_i = x_i) = \prod_{i=1}^{n} Grad_i^j \tag{۹}
$$

که $1 \leq j \leq r_i$ و $Grad_i^j$ احتمال انتخاب عمل j متناظر با مقدار x_i ، توسط اتوماتای یادگیر i ام مي باشد. در هر مرحله با استفاده از اتوماتاهای یادگیر به تعداد جمعیت، N ، ژنوم ساخته مي شود. سپس جمعیت جدید توسط تابع ارزیابي مورد ارزیابي قرار مي گیرد و از میان آنها Se ژنوم، که از بهترین ژنومها هستند انتخاب مي شوند. بعد از اعمال مکانیزمي که وابسته به مدل محیط اتوماتای یادگیرنده،(مدلهای P ، S و Q) مي باشد، بردار سیگنال تقویتي ساخته شده و عمل یادگیری را در مورد هر اتوماتای یادگیر انجام مي

---

[13] Markovian Model
[14] Ergodic Automata
[15] Learning Automata based Estimation of Distribution Algorithm

دهيم. پس از انجام عمل يادگيری نسل جديد ساخته شده و مراحل فوق تکرار مي شود. شرط پايان الگوريتم مي تواند همگرا شدن اتوماتاهای يادگير، توليد تعداد معيني نسل، انجام تعداد معيني ارزيابي و يا يکنواختي در چندين نسل پاياني باشد. به دليل ماهيت اتوماتای يادگير، ژنومهايي که در مراحل اوليه ساخته مي شوند بسيار تصادفي بوده و سمت و سوی خاصي ندارند و به همين دليل جمعيت اوليه دارای پراکندگي ژنتيکي بالايي خواهد بود. با گذشت زمان و تغيير بردار احتمال اتوماتاها، اين تنوع جمعيتي در جهت بهبود ارزش ژنومها کاهش مي يابد. برای گريز از شرايطي که الگوريتم در دام نقاط بهينه محلي گرفتار مي شود، از يک مکانيزم ثانويه توليد پاداش استفاده مي کنيم. به اين منظور در هر بار توليد نسل با يک احتمال از پيش تعريف شده به جای انتخاب Se ژنوم برتر جمعيت، N-Se ژنوم از کم ارزشترين ژنومهای جمعيت انتخاب شده و بر اساس آنها بردار سيگنال تقويتي بر اساس قانوني متفاوت ساخته مي شود.

طبق آنچه گفته شد يک الگوريتم $LAEDA$ را مي توان به صورت يک چندتايي $<N, Se, \mu, f, M, \Phi, \Psi>$ نشان داد که در آن N تعداد ژنومها در هر نسل توليد شده،Se تعداد ژنومهای انتخاب شده ، μ احتمال جريمه کردن بدترين مقادير ، f تابع ارزيابي، M مجموعه اتوماتاهای يادگير متناظر با متغيرهای سازنده ژنوم، Φ مکانيزم پاداش دهي به اتوماتاهای يادگير و بلاخره Ψ تابع تصوير کننده عملهای اتوماتاهای يادگير به مقادير متغيرها است. از مسايل مهمي که بايد به آن توجه شود، انتخاب نوع اتوماتاهای يادگير است. برای سادگي فرض مي کنيم همه اتوماتاهای يادگير مجموعه M از يک نوعند. همچنين در صورت استفاده از اتوماتاهايي که بر پايه پاداش کار مي کنند و از مکانيزم جريمه کردن در آنها استفاده نمي شود، مقدار μ را برابر صفر قرار مي دهيم. در حل مسائلي که متغيرهای تشکيل دهنده آنها کاملا مستقل باشند و با فرض آنکه تابع ارزيابي f مستقل از زمان باشد، محيطي که اتوماتای يادگير در آن قرار دارد ناايستا با تغييرات کم است. اما در حالتهايي که تابع f وابسته به زمان باشد و يا اينکه متغيرهای مساله وابسته باشند، محيط فعاليت اتوماتاهای يادگير ايستا با تغييرات بسيار شديد خواهد بود. در حالت دوم مجموعه اتوماتاهای يادگير يک بازی عمومي[16] با درجه وابستگي بالا را تشکيل خواهند داد. انتخاب نوع اتوماتاهای يادگير بکار رفته در M و نوع مکانيزم پياده سازی Φ وابستگي زيادی به مسئله مورد نظر دارد. در ادامه برای آشکار شدن بيشتر نحوه کار الگوريتم، يک نمونه از الگوريتم $LAEDA$ را با استفاده از مدل P و اتوماتای يادگير L_{RP} به صورت کد مجازي در شکل (۳) ارائه مي دهيم. همانطور که ديده مي شود با ايجاد تغيير بروی قدمهای ۳، ٤ و ٥ مي توان از اتوماتاهای يادگير متفاوت با مدل Q و S و سيستمهای پاداش دهي متفاوت استفاده کرد. البته استفاده از مدلهای Q و S باعث کندی روند همگرايي مي گردد. اما در حل بعضي از مسائل مي توان برای بالا بردن دقت از آنها استفاده کرد. در الگوريتم (۱)، مکانيزم پاداش دهي بر پايه پاداش و تنبيه در نظر گرفته شده است. در صورت استفاده از اتوماتاهای يادگيری مانند L_{RI} و يا Pursuit مي توانيم، مقدار μ را برابر صفر قرار دهيم.

//الگوريتم-۱: LAEDA-L_{RP}-P

ورودی: n متغير، تابع ارزيابي f ، مجموعه مقادير مجاز برای متغيرها، محدوديتهای مسئله

خروجي: ژنومي که تابع ارزيابي f را بهينه مي کند.

قدم ۱- در ابتدا مجموعه M متشکل از n اتوماتای يادگير L_{RP} را مقدار دهي اوليه مي کنيم. سپس نخستين جمعيت، D_l که $l=0$، با تعداد N ژنوم و با استفاده از M تشکيل مي دهيم. سپس هر يک از ژنومها با استفاده از f ارزيابي مي شوند.

قدم ۲- اگر شرايط پاياني الگوريتم بدست آمده باشد، به اجرای الگوريتم خاتمه مي دهيم. والا Se ژنوم را از بهترين ژنومهای مجموعه D_l که دارای ارزش بالاتری هستند انتخاب کرده و آن را D_l^{Se} مي ناميم.

قدم ۳- با احتمال $1-\mu$ به قدم ۷ مي رويم.

قدم ۴- برای هر متغير X_i، $1 \le i \le n$، و به ازای همه مقادير k، $1 \le k \le r_i$، $N_i(x_k)$ را به صورت زير محاسبه مي کنيم.

$$N_i(x_k) = \sum_{j=1}^{N-Se} \delta_j (X_i = x_k \mid D_l - D_l^{Se})$$

به طوريکه اگر متغير X_i در j امين ژنوم مجموعه $D_l - D_l^{Se}$، مقدار k ام خود را داشته باشد $\delta_j(X_i = x_k \mid D_l - D_l^{Se}) = 1$ و در غير اين صورت برابر صفر است.

قدم ۵- برای هر متغير X_i، $1 \le i \le n$، مقدار $worst\_action_i^{\max}$ را به صورت زير محاسبه ميکنيم.

[16] General Game

$N_i^{\max} = \max_{1 \le k \le r_i} N_i(x_k)$

$worst\_action_i^{\max} = \arg \max_{1 \le k \le r_i} N_i(x_k)$

قدم ۶- به ازای تمام مقادیر $1 \le i \le n$ عمل $worst\_action_i^{\max}$ اتوماتای یادگیر i ام مجموعه M، جریمه مي شود. سپس به قدم ۱۰ مي رویم.

قدم ۷- برای هر متغیر X_i، $1 \le i \le n$، و به ازای همه مقادیر k، $1 \le k \le r_i$، $N_i(x_k)$ را به صورت زیر محاسبه مي کنیم.

$N_i(x_k) = \sum_{j=1}^{Se} \delta_j(X_i = x_k \mid D_l^{Se})$

به طوریکه اگر متغیر X_i در j امین ژنوم مجموعه D_l^{Se}، مقدار k ام خود را داشته باشد $\delta_j(X_i = x_k \mid D_l^{Se}) = 1$ و در غیر این صورت برابر صفر است.

قدم ۸- برای هر متغیر X_i، $1 \le i \le n$، مقدار $best\_action_i^{\max}$ را به صورت زیر محاسبه میکنیم.

$N_i^{\max} = \max_{1 \le k \le r_i} N_i(x_k)$

$best\_action_i^{\max} = \arg \max_{1 \le k \le r_i} N_i(x_k)$

قدم ۹- به ازای تمام مقادیر $1 \le i \le n$ به عمل $best\_action_i^{\max}$ اتوماتای یادگیر i ام مجموعه M، پاداش داده مي شود.

قدم ۱۰- مقدار l را یکي افزایش داده و یک جمعیت جدید متشکل از N ژنوم جدید با استفاده از M ایجاد مي کنیم. سپس به قدم-۲ مي رویم.

شکل ۳- کد مجازی LAEDA در حالتیکه از اتوماتای یادگیر مبتني بر پاداش-تنبیه استفاده شود.

۵ - نتایج شبیه سازی

در این قسمت نتایج الگوریتم ارائه شـده را بـرای مـسائل OneMax، EqualProducts، CheckerBoard و SubsetSum [۲] در مقابل الگوریتمهای ژنتیک و *UMDA* مقایسه خواهیم کرد. علت انتخاب *UMDA* کارایي آن و در عین حال سادگي آن در برابـر سـایر الگوریتمهای تخمین توزیع است. در همه شبیه سازی از هر سه اتوماتـای یـادگیر L_{RI}، *Pursuit* و L_{RP} اسـتفاده کـردیم. بـرای مقایـسه کارایي الگوریتمهای ذکر شده پارامترهای مشترک بین آنها مانند تعداد ژنومهای هر نسل را یکسان در نظر گرفتیم. تعداد ژنومهای هر نـسل را برای مسائل ۱، ۲، ۳ و ۴ به ترتیب برابر ۱۰، ۱۰، ۱۰،۱۰،۲۰ در نظر گرفته شد. نـرخ یـادگیری در اتوماتـای یـادگیری و μ (بـرای اتوماتـای L_{RP}) به ترتیب در همه جا برابر ۰,۰۱ و ۰,۰۵ در نظر گرفته شده است. برای توقف الگوریتمها از چند شرط استفاده شد. شرط اول که تنهـا در مورد مسائل *OneMax* و *CheckerBoard مورد استفاده قرار گرفته است، رسیدن به مقدار بهینه مي باشد. در مورد دو مساله دیگر این مقدار ناشناخته مي باشد. شرط دوم رسیدن تعداد نسل هاي تولید شده به عدد خاصي مي باشد. که این اعداد بر اساس پیچیدگي مساله متفاوت در نظر گرفته شد شرط پایاني دیگر همگن شدن جمعیت نسلها به لحاظ ارزش آنها مي باشد. بخصوص در مـورد LAEDA پـس از تولید چند نسل اتوماتاهای یادگیر تقریبا همگرا مي شوند و دیگر نیازی به تولید نسلهای دیگر نمي باشـد. در الگوریتمهـای UMDA و LAEDA، Se ژنوم با ارزش بالاتر برای بروزرساني مدل احتمالاتي ژنوم انتخاب شـدند. مقـدار Se را در همـه آزمایـشات برابـر نـصف جمعیت هر نسل قرار دادیم. و نکته آخر اینکه در الگوریتم UMDA به منظور از دست ندادن بهترین ژنومهای تولید شده فرض کردیم در فرایند جستجو همواره بهترین ژنوم نسل فعلي به نسل بعد منتقل مي شود. در جدول (۱) مشخصات مسائل بیـان شـده اسـت.* در جـدول (۲) نتایج حاصل از شبیه سازی برای مسائل مطرح شده در جدول (۱) آورده شده اند. در مورد هـر الگـوریتم دو پـارامتر در جـدول ذکـر شـده است. یکي مقدار بهینه بدست آمده مي باشد که در آن ارزش بهترین جواب بدست آمده قبـل از توقـف الگـوریتم مـي باشـد. پـارامتر دوم تعداد ارزیابي ها مي باشد که نشان دهنده تعداد ژنومهای تولید شده توسط هر الگوریتم است. نکته که باید در نظر داشـته باشـیم ایـن اسـت که برای به ازای هر مساله هر کدام از الگوریتمها ۱۰ مرتبه اجرا شده اند. و نتایج ثبت شـده در جـدول میـانگیني از نتـایج تمـام اجراهـا مـي باشند. در ادامه برای بررسي دقیقتر الگوریتم ($LAEDA$-L_{RI}-P)، وابستگي آن را نسبت بـه پارامترهـای مختلـف ماننـد *Se* و *N* را بـرای

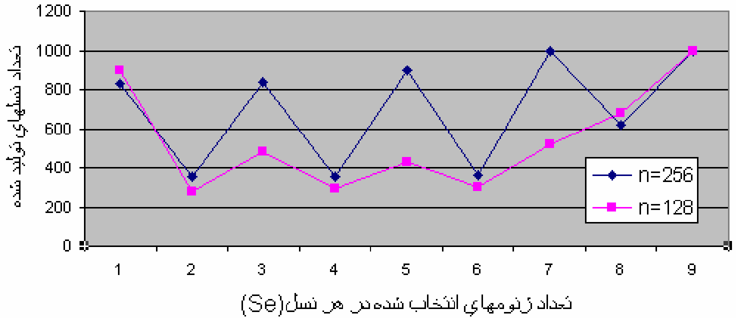
مساله *OneMax* مورد مطالعه قرار مي دهيم. در نمودار ۱ با فرض $N=10$ و $n=128,256$ مقادير متفاوت $Se=1,\ldots,9$ مـورد مطالعـه قرار گرفته است. نكته اى كه بايد به آن توجه شود، اين است كه حـداكثر تعـداد نـسلهاى توليـد شـده را برابـر ۱۰۰۰ در نظـر گرفتـه ايـم. همانطور كه در نمودار نيز ديده مي شود بهترين مقادير براي Se تقريبا نصف تعداد ژنومهاي هر نسل مي باشد. بـه عبـارت ديگـر بـا فـرض $Se \approx N/2$ احتمال توليد جوابهاي بهتر، بيشتر مي شود. در نمودار ۲ با فـرض $N \approx n/10$ و $Se \approx N/2$ تعـداد نـسلهاي توليـد شده براي رسيدن به جواب بهينه در مساله OneMax با تعداد متغيرهاي متفاوت به نمايش گذاشته شده است. براي مـشاهده نتـايج بـيشتر مي توانـد به [۱۰] مراجعه كنيد.

جدول ۱- مشخصات مسائل استفاده شده در شبيه سازى

مساله	F_{OneMax}	$F_{SubsetSum}$	$F_{CheckerBoard}$	$F_{EqualProducts}$
تعداد متغيرها	۱۲۸	۱۲۸	۱۰۰	۵۰
ماكزيمم تعداد ارزيابى	۱۰۰۰۰۰	۱۰۰۰۰۰	۱۰۰۰۰۰	۳۰۰۰۰۰
نوع مساله	ماكزيمم سازى	مينيمم سازى	ماكزيمم سازى	مينيمم سازى
مقدار بهينه	۱۲۸	نامشخص	۲۵۶	نامشخص

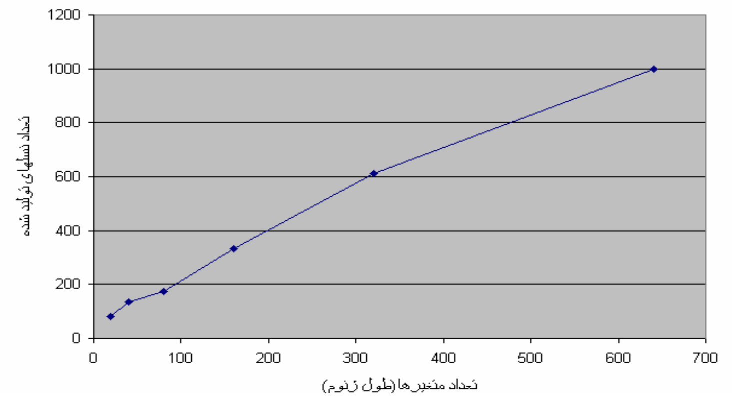
جدول ۲- نتايج بدست آمده از الگوريتمهاى *UMDA*، *SGA*[17]، *LAEDA-L_{RI}-P*، *LAEDA-Pursuit-P* و *LAEDA-L_{RP}-P* براى مسائل مطرح شده در جدول (۱)

مساله		*UMDA*	*SGA*	$LAEDA\text{-}L_{RI}\text{-}P$	*LAEDA-Pursuit-P*	$LAEDA\text{-}L_{RP}\text{-}P$
F_{OneMax}	مقدار بهينه بدست آمده	۱۲۸	۱۲۵,۴	۱۲۸	۱۲۸	۱۲۸
	تعداد ارزيابيها	۷۳۵۰	۳۶۲۱۰	۳۲۳۹	۲۶۷۰	۵۲۰۰
$F_{SubsetSum}$	مقدار بهينه بدست آمده	۰,۰۰۳۲۲	۰,۰۰۳۴۴	۰,۰۰۳۸۴	۰,۰۴۸۳۵	۰,۰۰۳۲۱
	تعداد ارزيابيها	۶۴۳۰	۱۲۴۳۰	۴۰۵۰	۳۸۰	۷۵۳۰
$F_{CheckerBoard}$	مقدار بهينه بدست آمده	۲۴۱	۲۴۶	۱۸۶	۱۶۶	۲۱۰
	تعداد ارزيابيها	۵۳۰۰۰	۵۲۰۰۰	۱۰۰۰۰	۵۰۰	۴۴۰۰۰
$F_{Equal\ Products}$	مقدار بهينه بدست آمده	۱,۹۵	۳,۳۵	۱,۱۶	۲,۹۷	۱,۱۰۳
	تعداد ارزيابيها	۳۶۳۲۰	۱۹۳۴۳۰	۱۷۰۰۰	۴۰۰۰	۲۲۳۴۰



نمودار ۱- در اين نمودار تاثير تعداد ژنومهاى انتخاب شده (*Se*) بروى تعداد نسلهاى لازم براى رسيدن به جواب بهينـه در مـسئله *OneMax* نـشان داده شده است. تعداد ژنومهاى هر نسل برابر ۱۰ و طول ژنوم ۱۲۸ و ۲۵۶ مى باشد.

---
[17] Simple Genetic Algorithm



نمودار ۲- این نمودار با فرض Se=N/2 میزان نسلهای لازم برای رسیدن به جواب را برای مساله OneMax با تعداد متغیرهای مختلف (طول ژنوم) نشان مي دهد.

نتيجه گيري

در اين مقاله الگوريتم تكاملي جديدي از دسته الگوريتمهاي تخمين توزيع بر اساس اتوماتای یادگیر معرفي شد. در اين الگوريتم با فرض استقلال متغيرهاي مساله بهينه سازي، از اتوماتاي يادگير براي تخمين توزيع احتمالاتي متغيرها استفاده شد. اگر چه فرض مستقل بودن متغيرها در اكثر مسائل بخصوص مسائل بهينه سازي تركيبي نقض مي شود ولي آزمايشات بروي تعدادي از مسائل مطرح شده نشان دهنده كارايي خوب الگوريتم مي باشد. آنچه كه مي تواند به عنوان کارهای آینده بروی اين الگوريتم صورت پذيرد، استفاده از انواع اتوماتاهای یادگیر با ساختارهای متغیر وثابت و بررسي نتايج بدست آمده در مورد مسائل مختلف با توجه به خطي يا غير خطي بودن آنها، مطالعه بروی تاثیر پارامترنرخ یادگیری و مدل مورد استفاده در دادن سیگنالهای تقويتي و موازي سازی محاسبات مي باشد.

مراجع

[1] Mühlenbein, H., and Paaß, G., “From Recombination of Genes to the Estimation of Distributions I. Binary Parameters”, In Lecture Notes in Computer Science 1411: Parallel Problem Solving from Nature-PPSN IV, PP. 178-187, 1996.

[2] Larranaga, P., Etxeberria, R., Lozano, J. A., and Pena, J. M., “Optimization by Learning and Simulation of Bayesian and Gaussian Networks”, Technical Report EHU-KZAA-IK-4/99, Department of Computer Science and Artificial Intelligence, University of Basque Country, December 1999.

[3] Mühlenbein, H., “The Equation for Response to Selection and Its Use for Prediction”, Evolutionary Computation, Vol. 5, No. 3, PP. 303-346, 1998.

[4] Baluja, S., Caruana, R., "Removing The Genetics from The Standard Genetic Algorithm", In Proceedings of ICML’95, PP. 38–46, Morgan Kaufmann Publishers, Palo Alto, CA, 1995.

[5] Baluja, S., “Population Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning”, Technical Report CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1994.

[6] Harik, G. R., Lobo, F. G., and Goldberg, D. E., "The Compact Genetic Algorithm", IEEE Transaction on Evolutionary Computing, Vol. 3, No. 4, PP. 287–297, 1999.

[7] Mühlenbein, H., and Mahnig, T., “Evolutionary Algorithms: From Recombination to Search Distributions”, Theoretical Aspects of Evolutionary Computing, Springer Publication, 2001.

[8] Oommen, B. J., and Agache, M., “A Comparison of Continuous and Discredited Pursuit Learning Schemes”, Technical Report, School of Computer Science, Carleton University, Canada, Ottawa, K1S 5B6, 1999.

[9] Narendra, K. S., and Thathachar, M. A. L., *Learning Automata: An Introduction*, Printice-Hall Inc, 1989.

[10] Rastegar, R., and Meybodi, M. R., “LAEDA: A New Evolutionary Algorithm using Learning Automata”, Technical Report, Computer Engineering Department, Amirkabir University, Tehran, 2003.