

یک چارچوب مبتنی بر آتاماتای یادگیر توزیع شده توسعه یافته برای حل مسأله یافتن زیرگراف بهینه تصادفی

محمد رضا ملاخلیلی میبیدی و محمد رضا میبیدی

روش تاکنون توانسته مسایل مختلف مرتبط با گراف‌های تصادفی نظیر یافتن کوتاه‌ترین مسیر تصادفی [۷] و [۶] یا حل مسأله فروشنده دوره‌گرد و یافتن مجموعه مستقل ماکزیمال [۸] و [۹] را حل کند. علاوه بر آن در موارد دیگری مثل دسته‌بندی اسناد وب [۱۰]، محاسبه رتبه اسناد [۱۱]، پیش‌بینی حرکت کاربران در وب [۱۲] و مدل‌سازی کاربران در محیط‌های فرایبندی نظیر وب [۱۳] و کاوش در نحوه استفاده از وب [۱۴] نیز مورد استفاده قرار گرفته است. اما عملاً نحوه فعالیت آتاماتاها در DLA، سیستم چندعامله توزیع شده مبتنی بر آتاماتای یادگیر را با محدودیت‌هایی مواجه کرده است. در این مقاله به معرفی eDLA که توسعه‌ای بر نحوه فعال شدن آتاماتاها در DLA است، پرداخته‌ایم.

در بخش ۲ به بررسی پیش‌نیازهای لازم برای ادامه مقاله پرداخته‌ایم و در بخش ۳ eDLA را بررسی کرده و سپس الگوریتم پیشنهادی مبتنی بر eDLA برای حل مسأله یافتن زیرگراف بهینه را ارائه کرده‌ایم. بخش ۴ به بررسی ویژگی‌های ریاضی الگوریتم پیشنهادی اختصاص یافته و در بخش ۵ نتیجه‌گیری و پیشنهادهایی برای کارهای بعدی ارائه شده است.

۲- مبانی ریاضی و مفاهیم مقدماتی

۲-۱ برخی تعاریف ریاضی

برای بررسی عملکرد الگوریتم یافتن زیرگراف بهینه پاره‌ای از مفاهیم را در این قسمت بازتعریف می‌کنیم. برخی از تعاریف برای به کارگیری در این مقاله ارائه شده و برخی دیگر تعاریف عمومی مفاهیم هستند.

تعریف ۱: گراف با سه تایی $G = (V(G), E(G), I_G)$ تعریف می‌شود که در آن $V(G) \neq \emptyset$ ، $V(G) \cap E(G) = \emptyset$ و I_G یک نگاشت وقوع است که به هر عضو $E(G)$ یک زوج نامرتب (یکسان یا متمایز) از $V(G)$ نظیر می‌کند. عناصر $V(G)$ را رئوس گراف G و عناصر $V(G)$ را یال‌های G می‌نامند. اگر برای یال e از G داشته باشیم $I_G(e) = \{u, v\}$ می‌نویسیم $I_G(e) = uv$ و u و v را پایان‌های e می‌نامیم. در ادامه این مقاله وجود I_G تصریح نمی‌شود.

تعریف ۲: گراف H را زیرگراف G می‌نامیم اگر $V(H) \subseteq V(G)$ ، $E(H) \subseteq E(G)$ و I_H برابر تحدید I_G به $E(H)$ باشد.

تعریف ۳: فرض کنید $G = (V, E)$ یک گراف باشد. اگر به هر یال $e \in E$ یک عدد نامنفی $w(e)$ که وزن یال e نامیده می‌شود نظیر کنیم، گراف حاصل را گراف با یال‌های وزن‌دار گوئیم. اگر H یک زیرگراف از G باشد، $w(H)$ وزن زیرگراف H و برابر با مجموع وزن یال‌های H است.

تعریف ۴: یک گراف با یال‌های دارای وزن تصادفی، توسط سه تایی $G_R = (V, E, W)$ تعریف می‌شود که $V = \{v_1, v_2, \dots, v_n\}$ مجموعه رئوس، $E = \{e_1, e_2, \dots, e_m\} \subset V \times V$ مجموعه یال‌های گراف و $W = \{w_1, w_2, \dots, w_n\}$ وزن‌های منتسب به یال‌ها را نشان می‌دهد به گونه‌ای که متغیر تصادفی w_i نشان‌دهنده وزن یال $e_i \in E$ است.

چکیده: در این مقاله، یک ساختار جدید شبکه‌ای از آتاماتاها یادگیر موسوم به آتاماتای یادگیر توسعه یافته معرفی شده و سپس الگوریتمی مبتنی بر این ساختار شبکه‌ای برای حل مسأله زیرگراف بهینه در گراف‌های تصادفی با یال‌های وزن‌دار از طریق نمونه‌گیری ارائه می‌شود. نشان داده شده که ساختار شبکه‌ای جدید پیشنهادی قادر به حل مسایل بهینه‌سازی روی گراف‌های تصادفی از طریق نمونه‌گیری با تعداد نمونه کمتر نسبت به روش نمونه‌گیری استاندارد است. علاوه بر این، اثباتی برای همگرایی آن به جواب بهینه ارائه شده و نشان داده می‌شود که ساختار شبکه‌ای پیشنهادی همواره با احتمال ۱ به جواب بهینه همگرا می‌گردد.

کلید واژه: آتاماتای یادگیر، آتاماتای یادگیر توزیع شده توسعه یافته، شبکه آتاماتاها یادگیر، زیرگراف، گراف تصادفی، نمونه‌گیری.

۱- مقدمه

مدل‌های آتاماتایی برای سیستم‌های یادگیر در دهه ۶۰ میلادی معرفی شدند و با نام آتاماتای یادگیر در [۱] معرفی شده و به محبوبیت رسیدند. از آن زمان تاکنون توسعه‌های مختلفی چه در زمینه تئوری و چه در حوزه کاربرد آتاماتای یادگیر رخ داده است. در [۲] و [۳] نویسندگان به تفصیل به بررسی این تحولات پرداخته‌اند.

آتاماتای یادگیر، اساساً عاملی ساده برای انجام کارهای ساده است [۳]. قابلیت کامل یک آتاماتای یادگیر، در یک شبکه از آتاماتاها و در تعامل آنها با یکدیگر ظهور پیدا می‌کند. یکی از ساختارهای متشکل از آتاماتاها یادگیر متصل به هم، آتاماتای یادگیر توزیع شده است که برای نخستین بار در [۴] معرفی و به کار گرفته شد.

DLA به عنوان شبکه‌ای متصل از آتاماتاها یادگیر همکار که برای حل یک مسأله خاص با یکدیگر همکاری دارند، بارها و بارها در حل مسایل مختلف، خصوصاً مسایل مربوط به گراف‌های تصادفی با یال‌های وزن‌دار و یا رئوس وزن‌دار مورد استفاده قرار گرفته است [۴] تا [۶].

یکی از اشکالات مشهود DLA به عنوان یک سیستم چندعامله توزیع شده، وابسته کردن انتخاب عامل فعال بعدی (که می‌بایستی در محیط اقدامی انجام دهد) به اقدام انجام شده توسط آتاماتای فعال فعلی است. به بیان ساده‌تر در DLA علاوه بر این که در هر زمان تنها یک عامل (آتاماتای یادگیر) فعال بوده و قادر به انجام کنشی در محیط است، انتخاب عامل فعال بعدی به نوع کنش عامل فعال فعلی وابسته است. این

این مقاله در تاریخ ۱۲ خرداد ماه ۱۳۹۲ دریافت و در تاریخ ۱۷ آذر ماه ۱۳۹۲ بازنگری شد.

محمد رضا ملاخلیلی میبیدی، گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه آزاد اسلامی، واحد میبد، میبد، (email: m.meybodi@srbiau.ac.ir)

محمد رضا میبیدی، آزمایشگاه محاسبات نرم، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، (email: mmeybodi@aut.ac.ir)

اقدامهای α عمل $\alpha_i(k)$ را انتخاب کرده باشد. همچنین فرض کنید بردار احتمال انتخاب اقدامهای آتاماتا را با $p(k)$ نمایش داده‌ایم. اگر a و b پارامترهایی باشند که به ترتیب میزان افزایش یا کاهش احتمالات اقدامها را مشخص می‌کنند و r تعداد اقدامهای قابل انجام توسط آتاماتای یادگیر باشد، بردار $p(k)$ توسط الگوریتم یادگیری خطی ارائه شده در روابط زیر به روز رسانی می‌شود. مقدار a را پارامتر پاداش و b را پارامتر جریمه می‌نامند

$$p_j(k+1) = \begin{cases} (\nu - a)p_j(k) + a, & j = i \\ (\nu - a)p_j(k), & \forall j \neq i \end{cases} \quad (2)$$

$$p_j(k+1) = \begin{cases} (\nu - b)p_j(k), & j = i \\ (\nu - b)p_j(k) + \frac{b}{r-1}, & \forall j \neq i \end{cases} \quad (3)$$

رابطه (۲) زمانی مورد استفاده قرار می‌گیرد که عمل $\alpha_i(k)$ منجر به دریافت پاداش از محیط شده باشد و (۳) زمانی مورد استفاده قرار می‌گیرد که این عمل به دریافت جریمه از محیط منجر شده باشد. در (۲) و (۳) اگر $a = b$ باشد روابط یادگیری خطی را الگوریتم L_{R-P} ، اگر $a \gg b$ باشد آن را L_{R-E} و اگر $b = 0$ باشد آن را L_{R-I} می‌نامند.

۲-۳-۱ معیارهای کارایی آتاماتای یادگیر

گرچه یادگیری اصولاً رهیافتی کیفی است، اما معیارهایی عددی وجود دارند که فرایند یادگیری را توصیف می‌کنند [۱۷]. برای اندازه‌گیری کارایی آتاماتای یادگیر تصادفی، معیارهای معینی تعریف شده‌اند که تأثیر آتاماتا را مشخص کرده و امکان مقایسه روش‌های مختلف یادگیری را فراهم می‌آورند. یکی از این معیارها متوسط پاداش دریافتی توسط آتاماتا است. فرض کنید آتاماتای یادگیر دارای r اقدام قابل انجام است. متوسط پاداش دریافتی توسط آتاماتای یادگیر در زمان n طبق (۴) تعریف می‌شود

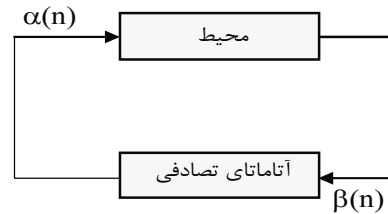
$$M(n) = E[\beta | p(n)] = \sum_{j=1}^r p_j(n) d_j = \sum_{j=1}^r p(\beta(n) = \nu | \alpha(n) = \alpha_j) p(\alpha(n) = \alpha_j) \quad (4)$$

اگر انتخاب اقدامهای آتاماتا به شکل کاملاً تصادفی انجام شود (آتاماتای شانسی محض) داریم $M(n) = M = \sum_{j=1}^r d_j / r$. زمانی که آتاماتا در حال یادگیری است، انتظار آن است که متوسط پاداشی بهتر از این مقدار داشته باشد.

تعریف ۶: آتاماتای یادگیر یا Expedient نامیده می‌شود اگر $\lim_{n \rightarrow \infty} E[M(n)] > M$. در حقیقت Expedient بودن حداقل انتظار از آتاماتای یادگیر است. رفتار بهتری که از آتاماتای یادگیر می‌توانیم انتظار داشته باشیم را به صورت زیر تعریف می‌کنیم:

تعریف ۷: آتاماتای یادگیر را بهینه یا optimal می‌گوییم اگر $\lim_{n \rightarrow \infty} E[M(n)] = d_m$ که $d_m = \max_i \{d_i\}$ می‌باشد. تعریف معادل دیگری برای بهینگی (with probability ۱) $\lim_{n \rightarrow \infty} P_m(n) = 1$ است. **تعریف ۸:** آتاماتای یادگیر یا ε -optimal نامیده می‌شود اگر $\lim_{n \rightarrow \infty} E[M(n)] > d_m - \varepsilon$ به ازای هر مقدار $\varepsilon > 0$ با انتخاب مناسب پارامترهای آتاماتای یادگیر قابل حصول باشد.

تعریف ۹: یک آتاماتای یادگیر یا absolutely - expedient نامیده می‌شود هرگاه $E[M(n+1) | P(n)] \geq M(n)$ باشد (به ازای هر مقدار $p_i(n) \in (0, 1)$ و تمام مجموعه‌های ممکن $\{d_i\} (i=1, 2, \dots, r)$ به جز حالتی که تمام d_i ها مقادیر برابر داشته باشند) [۱۸].



شکل ۱: آتاماتای یادگیر تصادفی.

تعریف ۵: دوگان خنثی یک گراف تصادفی، گرافی است هم‌ریخت با آن گراف که در آن یال‌ها وزن قطعی ۱ دارند (یا فاقد وزن هستند). به عبارت دیگر اگر $G_R = (V, E, W)$ یک گراف تصادفی باشد، گراف $G' = (V', E')$ که در آن $V' = V$ و $E' = E$ است را دوگان خنثی G_R می‌نامند.

۲-۲ کاربرد گراف تصادفی و زیرگراف بهینه تصادفی

گراف‌های وزن دار، ابزار بی‌نظیری برای مدل کردن بسیاری از ساختارها در دنیای علوم کامپیوتر هستند. شبکه‌های کامپیوتری، اینترنت، وب و شبکه‌های اجتماعی نمونه‌هایی هستند که می‌توان بر حسب کاربرد آنها را توسط گراف‌های وزن دار مدل کرد. مثلاً یک شبکه کامپیوتری را در نظر بگیرید. وزن یال‌ها می‌تواند تأخیر لینک‌های یک شبکه را نمایش دهد. بدیهی است که در این مثال، وزن یال یا همان تأخیر، یک متغیر تصادفی است که از یک تابع توزیع احتمال نامعلوم تبعیت می‌کند. یافتن مسیری با کمترین تأخیر در این شبکه می‌تواند به یافتن یک کوتاه‌ترین مسیر تصادفی نگاشت شود [۴]، [۱۵] و [۱۶].

۲-۳ آتاماتای یادگیر

آتاماتای یادگیر تصادفی یک واحد تصمیم‌گیرنده تطبیقی است که فرایند یادگیری در آن از طریق تعاملش با محیط صورت می‌گیرد. آتاماتای یادگیر، مجموعه‌ای از اقدامهای قابل انجام دارد که این اقدامها به تصادف و بر اساس یک بردار توزیع احتمال، انتخاب شده و به عنوان ورودی به محیط اعمال می‌شوند. محیط، اقدام انجام شده را به کمک یک سیگنال تقویتی بازخوردی، مورد ارزیابی قرار می‌دهد. آتاماتای یادگیر بر اساس سیگنال بازخوردی حاصل، بردار توزیع احتمال انتخاب اقدامها را به روز رسانی می‌کند. هدف آتاماتا، پیدا کردن اقدام بهینه در میان مجموعه اقدامهای قابل انجام است، اقدامی که بیشترین پاداش را از محیط دریافت کند.

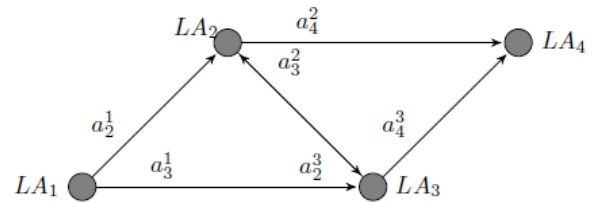
ارتباط آتاماتای تصادفی با محیط در شکل ۱ نشان داده شده است. از این مجموعه به همراه الگوریتم یادگیری تحت عنوان آتاماتای یادگیر تصادفی نام برده می‌شود و به این ترتیب آتاماتای یادگیر تصادفی را می‌توان با چهارتایی (۱) تعریف کرد

$$SLA \equiv \{\alpha, \beta, p, T, c\} \quad (1)$$

به طوری که $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه اقدامهای آتاماتا/مجموعه ورودی‌های محیط، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودی‌های آتاماتا/مجموعه خروجی‌های محیط، $p \equiv \{p_1, p_2, \dots, p_r\}$ بردار احتمال اقدامهای آتاماتا، $T \equiv \varphi \rightarrow \alpha$ الگوریتم یادگیری و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمالات جریمه که معرف محیط می‌باشند، است.

الگوریتم یادگیری یک رابطه بازگشتی است که برای انجام تغییرات و به روز رسانی در بردار احتمال اقدامهای آتاماتا در یک آتاماتای یادگیر تصادفی با ساختار متغیر مورد استفاده قرار می‌گیرد. فرض کنید یک آتاماتای یادگیر تصادفی ساختار متغیر در زمان k از میان مجموعه

در این شبکه از آتاماتاهای همکار در هر زمان تنها یک آتاماتا فعال است و تعداد اعمال قابل انجام توسط یک آتاماتا در DLA برابر است با تعداد آتاماتاهایی که به این آتاماتا متصل شده اند. انتخاب یک عمل توسط آتاماتا در این شبکه، باعث فعال شدن آتاماتای متصل شده به این آتاماتا و متناظر با این عمل می گردد. به عبارت معادل، انتخاب یک عمل توسط یک آتاماتا در این شبکه متناظر است با فعال شدن یک آتاماتای دیگر در این شبکه.



شکل ۲: آتاماتای یادگیر توزیع شده.

وجود یال (LA_i, LA_j) در این گراف به آن معناست که انتخاب عمل α_j^i توسط LA_i باعث فعال شدن LA_j می گردد. تعداد اعمال قابل انتخاب توسط LA_k برابر است با درجه خروجی این رأس. بردار احتمالات مربوط به عمل های قابل انجام توسط آتاماتای LA_k به صورت $P^k = \{p_r^k, p_v^k, \dots, p_n^k\}$ نمایش داده می شود. در این مجموعه عدد p_m^k نشان دهنده احتمال مربوط به عمل α_m^k است، انتخاب عمل α_m^k توسط LA_k باعث فعال شدن LA_m می شود و r_k تعداد اعمال قابل انجام توسط آتاماتای LA_k را نشان می دهد.

واضح است که برقراری رابطه $E[M(n+1)|P(n)] \geq M(n)$ ایجاب می کند که $M(n)$ یک زیرمارتینگل باشد.

۲-۳-۲ آتاماتای یادگیر با مجموعه اقدام های متغیر

یک آتاماتای یادگیر با مجموعه اقدام متغیر، آتاماتایی است که در آن تعداد اقدام های موجود هر آتاماتا در طول زمان تغییر می کند [۱۹]. فرض کنید مجموعه اقدام های آتاماتا را با $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ و مجموعه اقدام های قابل انجام توسط آتاماتا در زمان n را با $A(k) \subseteq \alpha$ نشان دهیم $(A(k) \in \{A_1, A_2, \dots, A_m\}, m = 2^{n-1})$.

انتخاب زیرمجموعه $A(k)$ از میان مجموعه اقدام های قابل انجام توسط آتاماتا، غالباً توسط شرایط بیرونی تحمیل می شود. می توان این گونه فرض کرد که این زیرمجموعه از اقدام های آتاماتا که از این به بعد مجموعه اقدام های فعال آتاماتا نامیده می شود، توسط یک عامل بیرونی و با توزیع احتمال $\psi_i(k) = \{\psi_1(k), \psi_2(k), \dots, \psi_m(k)\}$ انتخاب می شود. اگر تعریف کنیم $\hat{p}_i(k) = Prob[\alpha(k) = \alpha_i | A(k)]$ ، $\alpha_i \in A(k)$ احتمال انتخاب عمل α_i از میان مجموعه اقدام های فعال به شرط آن که مجموعه اقدام های فعال آتاماتا برابر با $A(k)$ باشد در این صورت

$$\hat{p}_i(k) = \frac{p_i(k)}{K(k)} \tag{5}$$

که $K(k)$ مجموع احتمالات تمام اقدام های فعال (اقدام های عضو $A(k)$) بوده و $K(k) = \sum_{\alpha_i \in A(k)} p_i(k)$ که $p_i(k) = Prob[\alpha(k) = \alpha_i]$ احتمال انتخاب اقدام α_i در میان مجموعه تمام اقدام های آتاماتا (اعم از این که فعال باشد یا نباشد) است. بدین ترتیب نحوه انتخاب اقدام توسط آتاماتا یا مجموعه اقدام متغیر بدین صورت است که فرض کنید آتاماتا مجموعه اقدام های فعال $A(k)$ را داشته باشد. ضریب نرمال کننده $K(k) = \sum_{\alpha_i \in A(k)} p_i(k)$ برای این مجموعه از اقدام های فعال محاسبه شده و بردار احتمال انتخاب اقدام های آتاماتا مطابق با رابطه بالا نرمال سازی می شود (به گونه ای که مجموع احتمال انتخاب اقدام های فعال همچنان ۱ باشد). پس از این، آتاماتا بر اساس بردار جدید $\hat{p}_i(k)$ یکی از اقدام های (فعال) خود را انتخاب کرده و به محیط اعمال می کند. در مرحله به روز رسانی نیز بردار $\hat{p}_i(k)$ مطابق با الگوریتم یادگیری مورد استفاده توسط آتاماتا به روز رسانی می شود و سپس با استفاده مجدد از رابطه بالا بردار $p_i(k) = \hat{p}_i(k) \cdot K(k)$ به دست می آید. نشان داده شده [۱۹] در صورتی که الگوریتم یادگیری مورد استفاده L_{R-1} باشد، این روش ویژگی های ϵ -optimality و absolute expediency را دارد.

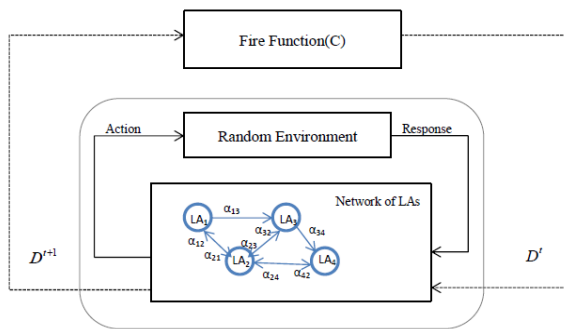
۲-۳-۳ آتاماتای یادگیر توزیع شده

آتاماتای یادگیر توزیع شده شبکه ای است از آتاماتاهای یادگیر که برای حل یک مسأله خاص با یکدیگر همکاری دارند. یک DLA در قالب یک گراف ارائه شده و با چندتایی (V, E, T, v) تعریف می شود که در آن V مجموعه رئوس گراف، E مجموعه یال ها، T مجموعه ای از الگوریتم های یادگیر مورد استفاده توسط آتاماتاها در DLA و v گره ریشه DLA است [۳].

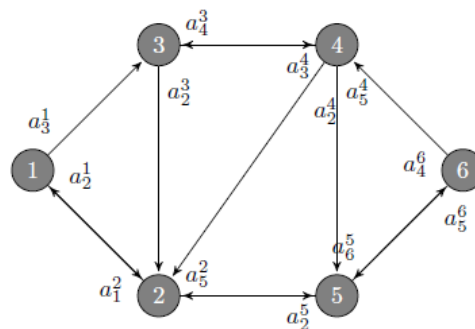
۳- ساختار و الگوریتم پیشنهادی جدید

آتاماتای یادگیر توزیع شده توسعه یافته شبکه ای است از آتاماتاهای یادگیر مرتبط با هم که تحت نظارت مجموعه ای از قواعد ارتباطی که ناظر بر ترتیب و نحوه فعالیت آنهاست، برای حل یک مسأله خاص با یکدیگر همکاری دارند. در آتاماتای یادگیر توزیع شده توسعه یافته، هر آتاماتا دارای یک سطح فعال بودن است و زمان و نحوه تغییر سطح فعالیت آتاماتاها به تبعیت از شرایط مسأله، تغییر می کند. در این آتاماتا در هر زمان تنها یک آتاماتا در سطحی قرار می گیرد که قادر به انجام اقدام و اعمال آن بر محیط است. به صورت فرمال می توان یک آتاماتای یادگیر توزیع شده توسعه یافته را به صورت زیر تعریف کرد:

- تعریف ۱۰: یک آتاماتای یادگیر توزیع شده توسعه یافته توسط چندتایی $eDLA = \{A, E, S, P, S', F, C\}$ تعریف می شود و در آن:
 - (۱) $A = \{1, 2, \dots, n\}$ مجموعه ای از آتاماتاهای یادگیر است که در آن $E \subseteq A \times A$ مجموعه ارتباطات سلسله مراتبی میان اعضای A است و $G = (A, E)$ یک گراف ارتباطی نامیده می شود.
 - (۲) $S = \{s_1, s_2, \dots, s_n\}$ مجموعه ای از مقادیر موسوم به سطح فعالیت است که سطح فعال بودن هر آتاماتای متعلق به A را مشخص می کند. هر یک از s_i ها می توانند یکی از مقادیر Passive, Active, Fire و Off باشند که آنها را به ترتیب با Pa, Ac, Fi و Off نمایش می دهیم. Fire، بالاترین سطح فعالیت و Passive و Off پایین ترین سطوح فعالیت هستند با این تفاوت که اگر آتاماتا به سطح فعالیت off برسد، دیگر هرگز تغییر وضعیت نمی دهد اما در حالت Passive امکان تغییر وضعیت وجود دارد.
 - (۳) P : مجموعه ای محدود از قواعد حاکم بر سطح فعالیت آتاماتاها است. این قواعد تابعی از سطح فعالیت فعلی هر آتاماتا و سطح فعالیت آتاماتاها مجاور است و بسته به مسأله ای که eDLA برای حل آن به کار گرفته می شود، می تواند متغیر باشد.
 - (۴) $S' = (s'_1, s'_2, \dots, s'_n)$ وضعیت آغازین سیستم نامیده می شود.
 - (۵) $F = \{S^F | S^F = (s_1^F, s_2^F, \dots, s_n^F)\}$ مجموعه شرایط پایانی نامیده می شود. مجموعه F شرایطی را بر حسب سطح فعالیت آتاماتاها مشخص می کند که در صورت تحقق حداقل یکی از آنها سیستم به وضعیت پایانی منتقل می شود. F حداقل یک عضو بدیهی دارد که در آن سطح فعالیت تمام آتاماتاها off است.



شکل ۴: ساختار کلی eDLA.



شکل ۳: گراف ارتباطی یک eDLA نمونه.

ه) آتاماتی که Fire شده پس از انجام عمل تصادفی انتخابی، به وضعیت Off تغییر وضعیت می‌دهد.

تعریف ۱۲ (توصیف لحظه‌ای): یک اجرا یا Run را می‌توان با توصیف‌های لحظه‌ای وضعیت آتاماتا نشان داد. توصیف لحظه‌ای عبارت است از یک چهارتایی مرتب که با $D^t = (D_{Of}^t, D_{Fi}^t, D_{Ac}^t, D_{Pa}^t)$ نشان داده می‌شود که در آن $\bigcup_i D_i^t = A$ و $D_i^t \subseteq A (i = \{Of, Fi, Ac, Pa\})$ و $D_i^t \cap D_j^t = \emptyset, (i \neq j, i, j \in \{Of, Fi, Ac, Pa\})$ هر بار که آتاماتا یکی از اقدام‌های خودش را انجام می‌دهد توصیف لحظه‌ای بر اساس قواعد P تغییر می‌کند. در حالت دیگر هرگاه تابع Fire یکی از آتاماتاها را Fire می‌کند، توصیف لحظه‌ای تغییر می‌کند. علاوه بر این داریم $D^t = (D_{Of}^t, D_{Fi}^t, D_{Ac}^t, D_{Pa}^t) = (\phi, \phi, \{A\}, A - A)$ که توصیف لحظه شروع eDLA و

وضعیت $D^{final} = (D_{Of}^{final}, D_{Fi}^{final}, D_{Ac}^{final}, D_{Pa}^{final}) = (A, \phi, \phi, \phi)$ است. وضعیت پایانی بدیهی یک Run است. بدین ترتیب یک Run را به شکل زیر و به صورت دنباله‌ای از توصیف‌های لحظه‌ای تعریف می‌کنیم

$$Run = D^0 \xrightarrow{fire} D^1 \xrightarrow{action} D^2 \xrightarrow{fire} \dots D^{final} = D^0 \xrightarrow{*} D^{final}$$

برای فهم بهتر تعاریف، به نمونه‌ای از یک اجرا در eDLA شکل ۳ دقت کنید

$$\begin{aligned} & (\phi, \phi, \{\}, \{2, 3, 4, 5, 6\}) \xrightarrow{fire^1} (\phi, \{\}, \{2, 3\}, \{4, 5, 6\}) \xrightarrow{\alpha_1} \\ & (\{\}, \phi, \{2, 3\}, \{4, 5, 6\}) \xrightarrow{fire^2} (\{\}, \{3\}, \{2, 4\}, \{5, 6\}) \xrightarrow{\alpha_2} \\ & (\{1, 3\}, \phi, \{2, 4\}, \{5, 6\}) \xrightarrow{fire^3} (\{1, 3\}, \{4\}, \{2, 5, 6\}, \phi) \xrightarrow{\alpha_3} \\ & (\{1, 3, 4\}, \phi, \{2, 5, 6\}, \phi) \xrightarrow{fire^4} (\{1, 3, 4\}, \{2\}, \{5, 6\}, \phi) \xrightarrow{\alpha_4} \\ & (\{1, 3, 4, 2\}, \phi, \{5, 6\}, \phi) \xrightarrow{fire^5} (\{1, 3, 4, 2\}, \{6\}, \{5\}, \phi) \xrightarrow{\alpha_5} \\ & (\{1, 3, 4, 2, 6\}, \phi, \{5\}, \phi) \xrightarrow{fire^6} (\{1, 3, 4, 2, 6\}, \{5\}, \phi, \phi) \xrightarrow{\alpha_6} \\ & \xrightarrow{no\ action} (\{1, 3, 4, 2, 6\}, \phi, \phi, \phi) \end{aligned}$$

به این ترتیب، آتاماتای یادگیر توزیع شده توسعه یافته یک ساختار دوبخشی است، متشکل از یک تابع تصمیم‌گیرنده و یک شبکه از آتاماتای یادگیر که برای حل یک مسئله خاص به شکل گروهی با یکدیگر همکاری دارند. بسته به نوع مسئله، فرایند تصمیم‌گیری متفاوت می‌باشد اما به صورت کلی، وظیفه بخش تصمیم‌گیرنده، تعیین آتاماتایی است که می‌بایستی عملی را در محیط تصادفی انجام دهد. شکل ۴ ساختار کلی eDLA را نشان می‌دهد.

در شکل ۵ الگوریتم پیشنهادی مبتنی بر eDLA برای حل مسئله زیرگراف بهینه ارائه شده است. نحوه عملکرد این الگوریتم بسیار ساده می‌باشد، تابع fire مسئله را روی یک گراف قطعی فاقد وزن (همان گراف دوگان خنثی) حل می‌کند. بدین صورت که یک گره را مشخص می‌کند تا از میان یال‌های متصل به آن گره، آتاماتای یادگیر متناظر، یالی را انتخاب کند.

۶) C : یک تابع خاص است که بر اساس سطح فعلی فعالیت آتاماتاها و مسأله‌ای که eDLA برای حل آن طراحی شده است، آتاماتایی را از میان مجموعه آتاماتاها موجود در eDLA انتخاب می‌کند. C را تابع آتش یا Fire Function می‌نامیم. در ادامه بحث در این خصوص بیشتر توضیح خواهیم داد.

هر آتاماتای یادگیر در eDLA متناسب با درجه‌اش در گراف ارتباطی (در حالتی که گراف ارتباطی G فاقد جهت باشد) و یا درجه خروجی‌اش در گراف ارتباطی (در حالتی که گراف ارتباطی G جهت‌دار باشد) اقدام برای انجام‌دادن دارد.

نحوه انجام کار بدین صورت است که eDLA از وضعیت S^0 آغاز کرده و بر اساس مجموعه قواعد P نسبت به تغییر سطح فعال بودن آتاماتاها اقدام می‌کند. تغییر سطح فعال بودن آتاماتاها تا زمان حل کامل مسئله ادامه پیدا می‌کند. در هر زمان آتاماتایی که در سطح فعالیت Fire قرار دارد، اقدامی را انتخاب کرده و آن را به محیط اعمال می‌کند. ضمن این که بر اساس قواعد ارتباطی مجموعه‌ای از آتاماتاها مجاور با آتاماتای Fire که در پایین‌ترین سطح فعال بودن (Passive) قرار دارند، ارتقای سطح می‌یابند (به سطح Active). هر آتاماتا پس از Fire شدن و انجام اقدام به وضعیت Off تغییر وضعیت می‌دهد. انتخاب آتاماتای Fire از میان مجموعه آتاماتاها Active صورت می‌گیرد و غالباً به صورت تصادفی انجام می‌شود.

تعریف ۱۱: یک دور اجرا یا run عبارت است از off شدن وضعیت تمام آتاماتاها موجود در eDLA از طریق طی کردن زیست چرخ حیات آتاماتا به شکل $Off \rightarrow Fire \rightarrow Active \rightarrow Passive$. در آغاز هر اجرا بر اساس S^0 آتاماتایی در eDLA وجود دارد که در وضعیت Active قرار دارد. این آتاماتا را آتاماتای ریشه نیز می‌نامیم و با A نمایش می‌دهیم. هر آتاماتا در eDLA یک چرخه زیستی مختص به خود را طی می‌کند. تغییر وضعیت آتاماتاها در eDLA بر اساس قوانین P و تغییراتی که توسط تابع Fire اعمال می‌شوند، است:

الف) تمامی آتاماتاها- به غیر از آتاماتای متناظر با ریشه- در آغاز کار در وضعیت Passive قرار دارند.

ب) تابع Fire مشخص می‌کند که کدام آتاماتا از مجموعه آتاماتاها Active به وضعیت Fire تبدیل شود.

ج) با Fire شدن وضعیت یک آتاماتا، کلیه آتاماتاها مجاور آن که Passive هستند به حالت Active تغییر وضعیت می‌دهند (فعال‌سازی به شیوه همسایگی محلی بر اساس قواعد P).

د) آتاماتایی که Fire شده از میان مجموعه اقدام‌های قابل انجام خود، به تصادف اقدامی را انجام می‌دهد که توسط محیط به شکل منفرد و پس از انجام عمل، یا به صورت گروهی و در پایان یک دور اجرا مورد ارزیابی قرار می‌گیرد.

اما برای این که پیچیدگی محاسباتی این روش کاهش یابد، ناچاریم اولاً از تخمین‌های ساده نظیر تخمین‌های stochastic gradient استفاده کنیم و ثانیاً هر فرضی مبنی بر اطلاع داشتن گره‌های میانی از وزن یال نمونه‌گیری شده را کنار بگذاریم. به این ترتیب، گراف تصادفی همانند یک جعبه سیاه تصور می‌شود که الگوریتم ما تنها به وزن زیرگراف حاصل از نمونه‌گیری روی این گراف تصادفی و یک تقریب نزدیک به واقعیت از واریانس وزن زیرگراف‌های به دست آمده اطلاع دارد. به این ترتیب، هیچ فرض اضافی در خصوص شبکه تصادفی پویا اعمال نشده [۱۶] و سادگی محاسباتی آتاماتای یادگیر نیز نقض نشده است.

برای به دست آوردن معیار جدید به این صورت عمل کرده‌ایم:

(۱) تفاوت وزن زیرگراف‌های به دست آمده (W_{k+1}) در هر مرحله با میانگین وزن زیرگراف‌های به دست آمده تا قبل از آن مرحله (WTH_k) محاسبه می‌گردد و این مقدار Err نامیده می‌شود $(Err_{k+1} = W_{k+1} - WTH_k)$.

(۲) از مقدار Err برای به روز رسانی مقدار آستانه پویا استفاده می‌شود $(WTH_{k+1} = WTH_k + \alpha Err_{k+1})$.

(۳) از یک متغیر نظیر V به عنوان تخمینی از واریانس استفاده می‌کنیم. مقدار این متغیر در هر مرحله از طریق یک تخمین‌گر خطی به صورت $V_{k+1} = V_k + \beta(abs(Err_{k+1}) - V_k)$ محاسبه می‌شود. کاملاً واضح است که این تخمین‌گر، انحراف از میانگین را محاسبه می‌کند (mdev) نه انحراف معیار (sdev). اما می‌دانیم که

$$mdev^2 = (\sum |W - W_{TH}|)^2 \geq \sum |W - W_{TH}|^2 = \delta^2 = sdev^2 \quad (۶)$$

رابطه (۶) نشان می‌دهد که انحراف از میانگین بزرگ‌تر از انحراف معیار و لذا محافظه‌کارانه‌تر از آن است، ضمن این که محاسبه آن فوق‌العاده ساده‌تر از واریانس است.

(۴) برای مقایسه و تعیین پاداش یا جریمه آتاماتاها از کران بالای $WTH_k/2 + 2 \times V_k$ استفاده می‌کنیم.

به این ترتیب الگوریتم ۱ شکل ۵ قادر به حل مسأله یافتن زیرگراف بهینه تصادفی روی گراف‌های تصادفی است. کافی است که با توجه به مسأله، قواعد مناسب اتخاذ شود. نمونه‌ای از این قواعد را برای دو مسأله مورد بررسی این مقاله در جدول ۱ مشاهده می‌کنید.

۴- تحلیل رفتار الگوریتم پیشنهادی

در این قسمت به بررسی ریاضی رفتار eDLA در حل مسأله یافتن زیرگراف در گراف‌های تصادفی می‌پردازیم. روشی که در این قسمت استفاده می‌کنیم، همان روشی است که در [۱] برای بررسی همگرایی پیشنهاد شده و در [۴] برای بررسی رفتار مجانبی DLA مورد استفاده قرار گرفته است.

اساساً دو نوع مختلف از همگرایی برای الگوریتم‌های یادگیری مورد استفاده در LA وجود دارد. مورد اول برای الگوریتم‌های یادگیری expedient نظیر L_{R-P} مورد بررسی قرار می‌گیرد. در این نوع همگرایی توابع توزیع دنباله احتمالات اقدام‌ها به یک تابع توزیع همگرا می‌شود. مثلاً نشان داده شده زمانی که از الگوریتم یادگیری L_{R-P} استفاده می‌شود، $p(n)$ بردار احتمال انتخاب اقدام‌های آتاماتاها، به یک متغیر تصادفی پیوسته همگرا می‌شود که میانگین و واریانس آن قابل محاسبه بوده و مقدار واریانس را می‌توان با انتخاب مقادیر مناسب برای پارامترهای الگوریتم یادگیری، به اندازه مورد نیاز کوچک کرد. در این نوع همگرایی، مقادیر اولیه احتمالات تأثیری در رفتار مجانبی الگوریتم ندارند [۱].

Algorithm1: eDLA-based algorithm for finding optimal stochastic sub-graph

Input: Stochastic Graph $G=(V,E,W)$; $v \in V$ Special node (root); K_T ; P_T
Output: Induced Graph
 1: $K=0$;
 2: Let S denotes the constructed Graph and W_S denotes its weight
 3: **Begin**
 4: Assign an automaton to each node of graph and construct eDLA
 5: Initialize all LA in eDLA (set activity state of v to active and others to passive);
 6: **Repeat**
 7: $S = \emptyset$; $W_S = 0$; $P = 1$;
 8: **Repeat**
 9: $A_i = \text{Fire (eDLA)}$;
 10: $\alpha = \text{selectaction}(A_i)$;
 11: $S = S + \text{edge}(A_i, \alpha)$;
 12: $W_S = W_S + \text{weight}(A_i, \alpha)$;
 13: **Until** ($A_i \neq 0$)
 14: $K = K + 1$;
 15: evaluate (eDLA, Threshold, K);
 16: Threshold = Update (Threshold, K);
 17: $\prod p(A_m, \alpha_n) = \prod p(m, n) \quad \forall \text{edge}(m, n) \in S$
 18: **Until** $K < K_T$ or $P < P_T$
 19: **End**

شکل ۵: الگوریتم پیشنهادی مبتنی بر eDLA برای یافتن زیرگراف تصادفی بهینه.

آتاماتا با انتخاب یک یال از میان یال‌های موجود گره‌ای که تابع fire انتخاب کرده، سعی می‌کند یالی را انتخاب کند که احتمال بهینه‌شدن جواب نهایی مسأله توسط آن بیشتر است (از طریق اقدام متناظری با احتمال انتخاب بیشتر). مجموعه یال‌های انتخاب‌شده توسط آتاماتاها در eDLA یک زیرگراف را تشکیل می‌دهد که توسط محیط ارزیابی می‌شود. ارزیابی محیط توسط سیگنال‌های تقویتی پاداش یا جریمه به همه آتاماتاها دخیل در ساخت زیرگراف مزبور بازخورد می‌شود. آتاماتاها از الگوریتم یادگیری L_{R-I} برای ارزیابی اقدام خود استفاده می‌کنند و در بخش ۴ و در آنالیز ریاضی eDLA خاصیت absolutely-expediency این رهیافت اثبات می‌شود.

بدین ترتیب ترکیب fire و selectaction که یکی از آنها روی گراف دوگان خنثی و دیگری روی گراف تصادفی انجام می‌شود، منجر به حل مسأله می‌شود.

در الگوریتم ۱ شکل ۵ از مقداری موسوم به مقدار آستانه^۱ پویا برای ارزیابی اقدام‌های آتاماتاها در eDLA استفاده می‌شود. (خط ۱۵) بر اساس قضیه حد مرکزی در [۴] روشی برای به روز رسانی این مقدار ارائه داده است. در روش پیشنهادی، این مقدار به صورت میانگین تمام مقادیر وزن زیرگراف‌های قبلی محاسبه شده است. این روش محاسبه تقریباً در تمام الگوریتم‌هایی که بعداً مبتنی بر DLA یا شبکه‌ای از آتاماتاها ارائه شده دیده می‌شود [۶]، [۷] و [۲۰]. بررسی‌ها نشان می‌دهد که دلیل نزدیک‌شدن مقدار آستانه پویا در الگوریتم به وزن زیرگراف بهینه، از مرحله‌ای به بعد قدرت یادگیری eDLA کاهش می‌یابد. برای حل این مشکل به جای مقدار آستانه پویا که یک میانگین‌گیری روی وزن زیرگراف‌های قبلی است، از معیار دیگری استفاده می‌کنیم که علاوه بر وزن، واریانس وزن زیرگراف‌های قبلی به دست آمده را نیز در محاسبات لحاظ کند.

جدول ۱: نحوه به کارگیری EDLA برای حل برخی مسایل بهینه‌سازی روی گراف تصادفی.

مسأله	درخت پوشای کمینه تصادفی	کوتاه‌ترین مسیر تصادفی بین s و d
انتخاب ROOT	تصادفی: از میان آتاماتاها بر اساس توزیع یکنواخت قطعی: آتاماتایی با کم‌ترین مقدار آنتروپی بردار احتمال اقدام‌ها	قطعی: مبدأ s
Fire شدن	تصادفی: از میان آتاماتاها Active بر اساس توزیع یکنواخت قطعی: آتاماتایی با کمترین مقدار آنتروپی بردار احتمال اقدام‌ها در میان آتاماتاها Active	قطعی: اقدام انجام شده توسط آتاماتای Fire فعلی تعیین کننده Fire بعدی است
Active شدن	بر اساس مجاورت با آتاماتای Fire فعلی	بر اساس مجاورت با آتاماتای Fire فعلی
خاتمه	Off شدن تمام آتاماتاها	Fire شدن آتاماتای d

تمام زیرگراف‌های مورد نظر مسأله r باشد و در مرحله k ، زیرگراف s_i که یکی از جواب‌ها است k_i بار انتخاب شده باشد و $i = 1, 2, \dots, r$ و علاوه بر آن $k = \sum_{i=1}^r k_i$. حال فرض کنید زیرگراف s_i در مرحله k انتخاب شده و نیز W_{s_i} وزن زیرگراف s_i باشد. مقدار وزن زیرگراف به دست آمده در j امین اجرای الگوریتم را با $W_{s_i}(j)$ نشان می‌دهیم ($j = 1, 2, \dots, k_i$) و نیز فرض کنید \bar{W}_{s_i} مقدار میانگین وزن زیرگراف s_i است ($i = 1, 2, \dots, r$). مقادیر d و c را به صورت زیر تعریف می‌کنیم

$$d_i(k) = Prob(W_{s_i}(k+1) \leq T(k))$$

$$c_i(k) = Prob(W_{s_i}(k+1) > T(k)) = 1 - d_i(k)$$

مقادیر d و c به ترتیب احتمال دریافت پاداش یا جریمه توسط محیط زمانی که زیرگراف s_i در مرحله k انتخاب شده را نشان می‌دهند. مقدار آستانه $T(k)$ برای $k > 1$ به صورت زیر تعریف می‌شود

$$T(k) = \frac{1}{k} \sum_{i=1}^r T_i(k_i)$$

در این رابطه k_i تعداد دفعاتی که زیرگراف s_i انتخاب شده را نشان می‌دهد و مقدار $T_i(k_i)$ برابر است با

$$T_i(k_i) = \frac{1}{k_i} \sum_{j=1}^{k_i} W_{s_i}(j)$$

شرط توقف الگوریتم آن است که حاصل ضرب احتمال انتخاب یال‌های زیرگراف s_i که آن را احتمال زیرگراف s_i نامیده و با q_i نشان می‌دهیم از یک مقدار آستانه از پیش تعیین شده بیشتر بوده و یا تعداد تکرارهای الگوریتم از مقدار آستانه‌ای بیشتر باشد.

قضیه ۱: فرض کنید $q_i(k)$ احتمال متناظر با زیرگراف بهینه s_i در گراف تصادفی مفروض در مرحله k ام را نشان دهد. اگر بردار $q(k) = [q_i(k)]_{i=1, \dots, r}$ مطابق با الگوریتم شکل ۵ تغییر کند، در این صورت $\lim_{n \rightarrow \infty} q_i(n) = 1$.

اثبات: اثبات این قضیه در چند مرحله صورت می‌گیرد.

لم ۱: اگر در الگوریتم ارائه شده شکل ۵، زیرگراف s_i با احتمال $c_i(k)$ در مرحله k جریمه شود و $\lim_{k \rightarrow \infty} c_i(k) = c^*$ باشد، در این صورت برای هر $\epsilon \in (0, 1)$ و $k > K(\epsilon)$ داریم

$$Prob(|c^* - c_i(k)| > \epsilon) < \epsilon$$

اثبات: مطابق با نمادگذاری قبلی، $c_i(k)$ احتمال جریمه شدن زیرگراف s_i را در گام k ام نشان می‌دهد. مطابق با قانون قوی اعداد بزرگ اگر $\lim_{k \rightarrow \infty} c_i(k) = c^*$ باشد آن گاه

$$\lim_{k \rightarrow \infty} Prob(|c^* - c_i(k)| > \epsilon) \rightarrow 0$$

لم ۲: اگر $d_i(k)$ و $c_i(k)$ احتمالات مربوط به جریمه و تنبیه شدن

حالت دوم همگرایی برای الگوریتم‌های یادگیری ϵ -optimal مثل L_{R-1} اتفاق می‌افتد. در این حالت از تئوری همگرایی مارتینگل استفاده می‌شود که به موجب آن دنباله‌ای از احتمالات اقدام‌ها به یک متغیر تصادفی محدود با احتمال ۱ همگرا می‌شود.

تفاوت این دو نوع همگرایی در آن است که در حالت اول زنجیره مارکوفی تولید می‌شود که ارگودیک است اما حالت جاذب ندارد ولی در روش دوم، زنجیره مارکوف تولید شده بیش از یک حالت جاذب دارد. بر خلاف روش اول، مقادیر اولیه بردار احتمالات نقش مؤثری در همگرایی روش دوم دارند.

به این ترتیب در مورد الگوریتم‌های یادگیری ϵ -optimal مثل L_{R-1} تنها چیزی که قابل بیان می‌باشد این است که $p(n)$ با یک احتمال مثبت به یک حالت دلخواه همگرا می‌شود و آنچه که اهمیت دارد، میزان عددی این احتمال است. برای حصول اطمینان از همگرایی، احتمال همگرایی به حالت مد نظر به صورت تابعی از احتمال اولیه تعیین می‌شود اما نکته اصلی یافتن چنین تابعی است.

اگر فرض کنیم $p_i(n)$ احتمال انتخاب اقدام i در یک آتاماتای یادگیر باشد که تحت الگوریتم L_{R-1} فرایند یادگیری را انجام می‌دهد، بایستی تابعی مثل $\Gamma_i(p) = pr[p_i(\infty) = 1 | p_i(0) = p]$ را برای کمی‌سازی احتمال همگرایی حالت i تعریف کرد. در [۲۱] نشان داده است تابع Γ بایستی معادله تابعی $U\Gamma_i(p) = \Gamma_i(p)$ را ارضا کند (با شرایط مرزی مناسب برای $p = 1$ و $p = 0$) که در آن U عملگری است که به صورت $U\Gamma_i(p) = E[\Gamma_i(p_i(n+1)) | p_i(n) = p]$ تعریف می‌شود. حل این معادله تابعی کار ساده‌ای نیست.

به جای حل این معادله از دو تابع با شرایط مرزی مناسب برای یافتن حدود بالا و پایین تابع Γ استفاده می‌شود. این دو تابع که آنها را $\varphi_1(\cdot)$ و $\varphi_r(\cdot)$ می‌نامیم، بایستی برای تمام مقادیر $p \in [0, 1]$ دارای شرایط (۷) باشند

$$U\varphi_1(p) \geq \varphi_1(p)$$

$$U\varphi_r(p) \leq \varphi_r(p) \tag{۷}$$

برقراری نابرابری (۷) ایجاب می‌کند که $\varphi_r(p) \leq \Gamma_i(p) \leq \varphi_1(p)$. توابع $\varphi_1(\cdot)$ و $\varphi_r(\cdot)$ به ترتیب ابرمنظم^۱ و زیرمنظم^۲ نامیده می‌شوند و یافتن آنها کار ساده‌تری است. در [۱] و [۲۱] نشان داده شده که این توابع به شکل (۸) هستند

$$\varphi_i(p) = \frac{\exp(x_i p) - 1}{\exp(x_i) - 1}, \quad i = 1, 2 \tag{۸}$$

در ادامه بحث، این نمادگذاری‌ها را در نظر بگیرید: فرض کنید تعداد

1. Super-Regular
2. Sub-Regular

جدول ۲: بررسی تغییرات در احتمال انتخاب زیرگراف بهینه در یک گراف نمونه.

تغییرات	احتمال انتخاب
$c_i(k)q_i(k)$	$s_i, q_i(k)$
$d_i(k)p_i^{\downarrow}(k \uparrow)p_i^{\downarrow}(k \downarrow)p_i^{\downarrow}(k)p_i^{\downarrow}(k)$	$s_i, q_i(k)$
$c_i(k)q_i(k)$	$s_r, q_r(k)$
$d_r(k)p_r^{\downarrow}(k \uparrow)p_r^{\downarrow}(k \downarrow)p_r^{\downarrow}(k)p_r^{\downarrow}(k)$	$s_r, q_r(k)$
$c_i(k)q_i(k)$	$s_r, q_r(k)$
$d_r(k)p_r^{\downarrow}(k \uparrow)p_r^{\downarrow}(k \uparrow)p_r^{\downarrow}(k \downarrow)p_r^{\downarrow}(k)$	$s_r, q_r(k)$
$c_i(k)q_i(k)$	$s_r, q_r(k)$
$d_r(k)p_r^{\downarrow}(k \uparrow)p_r^{\downarrow}(k \uparrow)p_r^{\downarrow}(k \uparrow)p_r^{\downarrow}(k)$	$s_r, q_r(k)$
$c_i(k)q_i(k)$	$s_s, q_s(k)$
$d_s(k)p_s^{\downarrow}(k \downarrow)p_s^{\downarrow}(k)p_s^{\downarrow}(k)p_s^{\downarrow}(k)$	$s_s, q_s(k)$
$c_i(k)q_i(k)$	$s_s, q_s(k)$
$d_s(k)p_s^{\downarrow}(k \downarrow)p_s^{\downarrow}(k)p_s^{\downarrow}(k)p_s^{\downarrow}(k)$	$s_s, q_s(k)$
$c_i(k)q_i(k)$	$s_v, q_v(k)$
$d_v(k)p_v^{\downarrow}(k \downarrow)p_v^{\downarrow}(k)p_v^{\downarrow}(k)p_v^{\downarrow}(k)$	$s_v, q_v(k)$
$c_i(k)q_i(k)$	$s_\lambda, q_\lambda(k)$
$d_\lambda(k)p_\lambda^{\downarrow}(k \downarrow)p_\lambda^{\downarrow}(k)p_\lambda^{\downarrow}(k)p_\lambda^{\downarrow}(k)$	$s_\lambda, q_\lambda(k)$

$$E[q_r(k+1)|q(k)] = q_i(k) \times [c_i(k)q_i(k) + d_i(k)p_i^{\downarrow}(k \uparrow)p_i^{\downarrow}(k \downarrow)p_i^{\downarrow}(k)p_i^{\downarrow}(k)] + q_r(k) \times [c_r(k)q_r(k) + d_r(k)p_r^{\downarrow}(k \uparrow)p_r^{\downarrow}(k \downarrow)p_r^{\downarrow}(k)p_r^{\downarrow}(k)] + q_r(k) \times [c_r(k)q_r(k) + d_r(k)p_r^{\downarrow}(k \uparrow)p_r^{\downarrow}(k \uparrow)p_r^{\downarrow}(k \downarrow)p_r^{\downarrow}(k)] + q_r(k) \times [c_r(k)q_r(k) + d_r(k)p_r^{\downarrow}(k \uparrow)p_r^{\downarrow}(k \uparrow)p_r^{\downarrow}(k \uparrow)p_r^{\downarrow}(k)] + q_s(k) \times [c_s(k)q_s(k) + d_s(k)p_s^{\downarrow}(k \downarrow)p_s^{\downarrow}(k)p_s^{\downarrow}(k)p_s^{\downarrow}(k)] + q_s(k) \times [c_s(k)q_s(k) + d_s(k)p_s^{\downarrow}(k \downarrow)p_s^{\downarrow}(k)p_s^{\downarrow}(k)p_s^{\downarrow}(k)] + q_v(k) \times [c_v(k)q_v(k) + d_v(k)p_v^{\downarrow}(k \downarrow)p_v^{\downarrow}(k)p_v^{\downarrow}(k)p_v^{\downarrow}(k)] + q_\lambda(k) \times [c_\lambda(k)q_\lambda(k) + d_\lambda(k)p_\lambda^{\downarrow}(k \downarrow)p_\lambda^{\downarrow}(k)p_\lambda^{\downarrow}(k)p_\lambda^{\downarrow}(k)] = \sum_{j=1}^r q_j(k) [c_j(k)q_j(k) + d_j(k) \prod_{e(m,n) \in s_j} \delta_n^m(k)]$$

که در آن

$$\delta_n^m(k) = \begin{cases} p_n^m(k \uparrow) = p_n^m(k)(1-a) + a, & e(m,n) \in s_j \\ p_n^m(k \downarrow) = p_n^m(k)(1-a), & e(m,n) \notin s_j \end{cases}$$

با توجه به این مثال، به صورت کلی می‌توانیم چنین بنویسیم

$$E[q_i(k+1)|q(k)] = \sum_{j=1}^r E[q_i(k+1)|q(k), \alpha_j] \cdot p(\alpha_j | q(k)) \quad (10)$$

در (۱۰) منظور از α_j مجموعه اقدام‌هایی از eDLA است که منجر به انتخاب زیرگراف s_j گردیده است

$$E[q_i(k+1)|q(k)] = \sum_{j=1}^r E[q_i(k+1)|q(k), \alpha_j] \cdot q_j(k)$$

اما

$$E[q_i(k+1)|q(k), \alpha_j] = (1-d_j)q_i(k) + d_j \prod_{e(m,n) \in s_j \cap s_i} (p_n^m(k) \uparrow) \prod_{e(m,n) \in s_j - s_i} (p_n^m(k) \downarrow)$$

زیرگراف s_i حاصل شده در k امین مرحله را نشان دهند (روابط بالا) و بردار $\underline{q}(k)$ بر اساس الگوریتم شکل ۵ تغییر کند، در این صورت امید ریاضی شرطی $E[q_i(k+1)|q(k)]$ مطابق با روابط زیر تعریف می‌شود

$$E[q_i(k+1)|q(k)] = \sum_{j=1}^r q_j(k) [c_j(k)q_i(k) + d_j(k) \prod_{e(m,n) \in s_j} \delta_n^m(k)] \quad (9)$$

که در (۹)

$$\delta_n^m(k) = \begin{cases} p_n^m(k+1) = p_n^m(k)(1-a) + a, & e(m,n) \in s_j \\ p_n^m(k+1) = p_n^m(k)(1-a), & e(m,n) \notin s_j \end{cases}$$

اثبات: الگوریتم پیشنهادی (الگوریتم ۱ شکل ۵) از روش یادگیری L_{R-1} برای پاداش یا جریمه آتاماتاها استفاده می‌کند، لذا در مرحله k ام اجرای الگوریتم، احتمال انتخاب زیرگراف s_i (یعنی $q_i(k)$) با احتمال $c_i(k)$ بدون تغییر باقی می‌ماند، اگر زیرگراف s_i توسط محیط ارزیابی شده و جریمه شود. اما اگر زیرگراف s_i توسط محیط پاداش داده شود، در این صورت احتمال انتخاب یال‌های تشکیل‌دهنده زیرگراف s_i افزایش می‌یابد و در این حالت برای سایر زیرگراف‌ها نظیر s_j دو حالت قابل تصور است:

- الف) s_j هیچ یال مشترکی با s_i ندارد که در این صورت تمام یال‌های تشکیل‌دهنده s_j در مرحله k ام جریمه شده‌اند و در این صورت $q_j(k+1) < q_j(k)$.

- ب) s_j تعدادی یال مشترک با s_i دارد و این بدان معناست که برخی یال‌ها جریمه و برخی دیگر پاداش گرفته‌اند.

برای فهم ادامه اثبات فرض کنید گرافی داریم که در آن ۸ زیرگراف به شرح زیر جواب‌های ممکن مسأله هستند

$$\begin{aligned} s_1 &= \{(1,2), (2,5)\}, & s_2 &= \{(1,2), (2,3), (3,5)\} \\ s_3 &= \{(1,2), (2,3), (3,5)\}, & s_4 &= \{(1,2), (2,3), (3,4), (4,5)\} \\ s_5 &= \{(1,5)\}, & s_6 &= \{(1,4), (4,5)\} \\ s_7 &= \{(1,3), (3,5)\}, & s_8 &= \{(1,3), (3,4), (4,5)\} \end{aligned}$$

ضمناً فرض کنید s_4 ارض‌کننده محدودیت‌های مورد نیاز مسأله روی گراف تصادفی باشد. برای هر یک از این زیرگراف‌ها، مقدار $q(k)$ متناظر برابر است با

$$\begin{aligned} q_1(k) &= p_1^{\downarrow}(k)p_5^{\downarrow}(k), & q_2(k) &= p_1^{\downarrow}(k)p_2^{\downarrow}(k)p_3^{\downarrow}(k)p_5^{\downarrow}(k) \\ q_3(k) &= p_1^{\downarrow}(k)p_2^{\downarrow}(k)p_3^{\downarrow}(k)p_5^{\downarrow}(k) \\ q_4(k) &= p_1^{\downarrow}(k)p_2^{\downarrow}(k)p_3^{\downarrow}(k)p_4^{\downarrow}(k)p_5^{\downarrow}(k) \\ q_5(k) &= p_1^{\downarrow}(k), & q_6(k) &= p_1^{\downarrow}(k)p_4^{\downarrow}(k) \\ q_7(k) &= p_1^{\downarrow}(k)p_3^{\downarrow}(k), & q_8(k) &= p_1^{\downarrow}(k)p_3^{\downarrow}(k)p_4^{\downarrow}(k)p_5^{\downarrow}(k) \end{aligned}$$

در این روابط احتمال انتخاب یال (i, j) یا همان انتخاب اقدام α_j توسط آتاماتای fire شده A_j در مرحله k ام است. از آنجا که ۸ جواب بیشتر متصور نیست، بسته به این که در مرحله $k+1$ کدام یک از زیرگراف‌های s_1 تا s_8 انتخاب شده باشند، تغییرات حاصل در $q_r(k)$ مطابق با مجموع مقادیر در جدول ۲ است. منظور از $p_j^i(k \downarrow)$ کاهش در مقدار p_j^i و منظور از $p_j^i(k \uparrow)$ افزایش در مقدار p_j^i از طریق روابط L_{R-1} است

$$\begin{aligned} p_j^i(k \downarrow) &= p_j^i(k) \times (1-a) \\ p_j^i(k \uparrow) &= p_j^i(k) \times (1-a) + a \end{aligned}$$

با جمع مقادیر حاصل در جدول ۲ خواهیم داشت

جاذب فرایند $\{q(k)\}$ باشد و $q^* \in V_r$ حالتی را نشان دهد که $q(k)$ به آن همگرا می‌شود. تعریف می‌کنیم $\Gamma_i[q] = \Pr[q^* = e_i | q(0) = q]$. فرض کنید $C(S_r): S_r \rightarrow R$ فضای حالت تمام توابع مشتق‌پذیر پیوسته حقیقی مقدار با مشتق محدود تعریف‌شده روی S_r باشد که R یک خط حقیقی است. اگر $g[q] \in C(S_r)$ باشد، یک الگوریتم یادگیری نظیر یک عملگر U با میانگین $Ug[q] = E[g[q(k+1)] | q(k) = q]$ عمل می‌کند که در این رابطه $E[\cdot]$ نشان‌دهنده امید ریاضی است. نشان داده شده که U عملگری خطی است و توابع نامنفی را نامنفی نگه می‌دارد یعنی $Ug[q] \geq 0 \quad \forall q \in S_r$ if $g[q] \geq 0$.

نسبت به عملگر U می‌توان دو گروه از توابع را تعریف کرد. اگر $Ug[q] \geq g[q]$ تابع $g[q]$ را زیرمنظم و اگر $Ug[q] \leq g[q]$ تابع $g[q]$ را ابرمنظم گویند.

نشان داده شده که $\Gamma_i[q]$ تنها جواب پیوسته معادله $U\Gamma_i[q] = \Gamma_i[q]$ با شرایط مرزی زیر است

$$\Gamma_i[e_j] = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (13)$$

اما حل معادله تابعی $U\Gamma_i[q] = \Gamma_i[q]$ برای تعیین احتمال انتخاب زیرگراف بهینه بر حسب احتمال اولیه، کار سختی است. برای این کار از یک کران پایین برای این تابع استفاده می‌کنیم. از گزاره قبلی می‌دانیم که احتمال انتخاب زیرگراف بهینه $q_i^* \in \{0, 1\}$ است. تعریف می‌کنیم

$$\Phi_i[x, q] = \frac{\exp(-xq_i/a) - 1}{\exp(-x/a) - 1} \quad (14)$$

که در آن $x > 0$ بایستی انتخاب شود. واضح است که تابع تعریف‌شده در (۱۴) شرایط مرزی تعریف‌شده در (۱۳) را ارضا می‌کند. نشان خواهیم داد که $\Phi_i[x, q]$ یک تابع زیرمنظم بوده و لذا قادر است که یک کران پایین برای $\Gamma_i[q]$ تعریف کند. می‌توان نشان داد که $\Phi_i[x, q]$ وقتی زیرمنظم است که $\theta_i[x, q] = \exp(-xq_i/a)$ ابرمنظم باشد.

لم ۴: اگر s_i زیرگراف بهینه و $\theta_i[x, q] = \exp(-xq_i/a)$ باشد که در آن $q_i = \prod_{e(m,n) \in s_i} p_n^m$ احتمال انتخاب زیرگراف s_i و a نرخ یادگیری است، در این صورت یک x وجود دارد که $\theta_i[x, q]$ ابرمنظم خواهد بود.

اثبات: برای تعیین شرایطی که تحت آن شرایط، تابع $\theta_i[x, q]$ یک تابع ابرمنظم است، از تعریف ابرمنظم و محاسبه مقدار امید ریاضی استفاده می‌کنیم. فرض کنید زیرگراف s_i جواب بهینه باشد، در این صورت

$$U\theta_i[x, q] = E[\exp(\frac{-xq_i(k+1)}{a}) | q(k) = q] \quad (15)$$

برای محاسبه مقدار امید ریاضی در (۱۵) به این ترتیب عمل می‌کنیم

$$\begin{aligned} E[\exp(\frac{-xq_i(k+1)}{a}) | q(k) = q] &= \\ \sum_{j=1}^r E[\exp(\frac{-xq_i(k+1)}{a}) | q(k) = q, s_j] q_j &= \\ \sum_{j=1}^r \exp(\frac{-xq_i(k+1)}{a}) \times q_j \times d_j + & \\ \sum_{j=1}^r \exp(\frac{-xq_i(k)}{a}) \times q_j \times (1-d_j) & \end{aligned} \quad (16)$$

با ساده‌سازی (۱۶) به (۱۷) و (۱۸) می‌رسیم. در (۱۷) داریم

در این روابط $p_j^i(k)$ احتمال انتخاب یال (i, j) یا همان انتخاب اقدام α_j توسط آتاماتای A_i شده در مرحله k ام است. منظور از $(p_j^i(k) \downarrow)$ کاهش در مقدار p_j^i و منظور از $(p_j^i(k) \uparrow)$ افزایش در مقدار p_j^i از طریق روابط L_{R-1} است

$$\begin{aligned} p_j^i(k) \downarrow &= p_j^i(k) \times (1-a) \\ p_j^i(k) \uparrow &= p_j^i(k) \times (1-a) + a \end{aligned}$$

به این ترتیب لم ۲ اثبات می‌گردد. **لم ۳:** اگر زیرگراف بهینه s_i جواب مسأله یافتن زیرگراف بهینه تصادفی باشد، در این صورت مقدار $E[q_i(k+1) - q_i(k) | q(k)]$ همواره مثبت است.

اثبات: الگوریتم یافتن زیرگراف بهینه تشریح‌شده در شکل ۵، همواره حداکثر $n-1$ یال را از میان یال‌های گراف انتخاب می‌کند تا یکی از r جواب ممکن را بسازد. به این ترتیب متوسط مقدار $q_i(k)$ برای زیرگراف s_i که از ضرب مقادیر احتمال اقدام‌های متناظر با یال‌های تشکیل‌دهنده s_i تشکیل می‌شود برابر است با

$$\begin{aligned} \Delta q_i(k) &= E[q_i(k+1) - q_i(k) | q(k)] = \\ \prod_{e(m,n) \in s_i} E[p_n^m(k+1) | p_n^m(k)] - \prod_{e(m,n) \in s_i} p_n^m(k) &\geq \\ \prod_{e(m,n) \in s_i} (E[p_n^m(k+1) | p_n^m(k)] - p_n^m(k)) &= \\ \prod_{e(m,n) \in s_i} \Delta p_n^m(k) & \end{aligned} \quad (11)$$

$p^m(k)$ بردار احتمال انتخاب اقدام‌های آتاماتای A_m متناظر با گره m در مرحله k است و برای محاسبه $\Delta q_i(k)$ به بررسی وضعیت تک‌تک آتاماتاهایی می‌پردازیم که تشکیل‌دهنده زیرگراف s_i هستند. فرض کنید که آتاماتای A_i در حین اجرای الگوریتم fire شده باشد. در این صورت میزان تغییرات احتمال انتخاب عمل u از A_i که منجر به اضافه‌شدن یال (t, u) به مجموعه یال‌های s_i می‌شود، پس از مرحله ارزیابی عبارت است از

$$\Delta p_u^i(k)_{e(t,u) \in s_i} = a p_u^i(k) \sum_{s \neq u} p_s^i(k) (c_s^i(k) - c_u^i(k)) \quad (12)$$

مطابق با (۱۲)، اولاً $p_u^i(k)$ و $p_s^i(k)$ مقادیری مثبت در بازه $(0, 1)$ هستند و ثانیاً از آنجا که $e(t, u) \in s_i$ یالی است که متعلق به زیرگراف بهینه است، توسط آتاماتای A_i انتخاب می‌شود. اما از لم ۲ می‌دانیم که برای مقادیر بزرگ k داریم $c_s^i(k) - c_u^i(k) > 0, \forall s \neq u$. بنابراین برای مقادیر بزرگ k طرف راست (۱۲) مقداری مثبت است و در نتیجه $\Delta q_i(k) \geq 0$.

گزاره: $\lim_{k \rightarrow \infty} q_i(k) = q_i^* \in \{0, 1\}$ با احتمال ۱ وجود دارد. لم ۳ بیان‌کننده آن است که $\{q(k)\}$ یک زیرمارتینگل است. با استفاده از قضایای حدی مارتینگل و در نظر گرفتن این واقعیت که $\{q(k)\}$ نامنفی و یکنوا محدود است، نتیجه می‌شود که $\lim_{k \rightarrow \infty} q_i(k) = q_i^*$ با احتمال ۱ وجود دارد. علاوه بر این اگر فرض کنیم $q_i(k) \neq 0$ و $q_i(k) \neq 1$ در این صورت با یک احتمال ناصفر به ازای تمام مقادیر k داریم $q_i(k) = q_i(k+1)$. بنابراین الزاماً $q_i^* \in \{0, 1\}$.

گزاره قبلی نشان می‌دهد که الگوریتم ممکن است به جواب بهینه همگرا نشود. سعی می‌کنیم حدودی برای تابع احتمال انتخاب زیرگراف بهینه ارائه دهیم و برای این کار از روش [۲۱] استفاده می‌کنیم. فرض کنید مجموعه بردارهای یکه $V_r = \{e_1, e_2, \dots, e_r\}$

$$\theta_i[x, \underline{q}] \sum_{j=1}^r q_j d_j = q_i d_i \theta_i[x, \underline{q}] + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j \theta_i[x, \underline{q}]$$

فلذا (۲۳) به دست می آید و تعریف می کنیم

$$V[x] = \begin{cases} \frac{\exp(x) - 1}{x}, & x \neq 0 \\ 1, & x = 0 \end{cases}$$

لذا

$$U\theta_i[x, \underline{q}] - \theta_i[x, \underline{q}] \leq -xq_i\theta_i[x, \underline{q}]((1-q_i)d_iV[-x(1-q_i)] - \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j V[xq_j]) \quad (24)$$

شرط لازم برای ابرمنظم بودن $\theta_i[x, \underline{q}]$ آن است که رابطه زیر برقرار باشد

$$(1-q_i)d_iV[-x(1-q_i)] - \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j V[xq_j] \geq 0$$

لذا

$$(1-q_i)d_iV[-x(1-q_i)] \geq \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j V[xq_j]$$

یا

$$\frac{V[-x(1-q_i)]}{V[xq_i]} \geq \frac{\sum_{j=1, j \neq i}^r q_j d_j}{(1-q_i)d_i} \quad (25)$$

در (۲۵) داریم

$$\frac{\sum_{j=1, j \neq i}^r q_j \times d_j}{(1-q_i)d_i} = \frac{\sum_{j=1, j \neq i}^r q_j \times d_j}{\sum_{j=1, j \neq i}^r q_j \times d_i} = \frac{\sum_{j=1, j \neq i}^r q_j \times \frac{d_j}{d_i}}$$

لذا

$$\sum_{j=1}^r \exp\left(\frac{-xq_i(k+1)}{a}\right) q_j d_j = q_i d_i \exp\left(\frac{-x \prod_{e(m,n) \in S_i} \{p_n^m + a(1-p_n^m)\}}{a}\right) + \sum_{j=1, j \neq i}^r q_j d_j \exp\left(\frac{-x \prod_{e(m,n) \in S_i \cap S_j} \{p_n^m + a(1-p_n^m)\}}{a} \prod_{e(m,n) \in S_i - S_j} \{p_n^m(1-a)\}}{a}\right) \quad (17)$$

$$\sum_{j=1}^r \exp\left(\frac{-xq_i(k)}{a}\right) \times q_j \times (1-d_j) = \theta_i[x, \underline{q}] \sum_{j=1}^r q_j \times (1-d_j) = \theta_i[x, \underline{q}] - \theta_i[x, \underline{q}] \sum_{j=1}^r q_j \times d_j \quad (18)$$

$$U\theta_i[x, \underline{q}] - \theta_i[x, \underline{q}] \leq q_i d_i (\exp\{-\frac{x}{a}(q_i + a(1-q_i))\} - \theta_i[x, \underline{q}]) + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j (\exp\{-\frac{x}{a}q_i(1-a)\} - \theta_i[x, \underline{q}]) = \theta_i[x, \underline{q}] (q_i d_i \exp\{-x(1-q_i)\} - 1) + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j (\exp\{xq_j\} - 1) = -xq_i\theta_i[x, \underline{q}]((1-q_i)d_i \frac{\exp\{-x(1-q_i)\} - 1}{-x(1-q_i)} - \sum_{\substack{j=1 \\ j \neq i}}^r q_j \times d_j \frac{\exp\{xq_j\} - 1}{xq_j}) \quad (23)$$

$$\prod_{e(m,n) \in S_i - S_j} \{p_n^m(1-a)\} = \frac{\prod_{e(m,n) \in S_i - S_j} \{p_n^m(1-a)\} \prod_{e(m,n) \in S_i \cap S_j} \{p_n^m(1-a)\}}{\prod_{e(m,n) \in S_i \cap S_j} \{p_n^m(1-a)\}} = \quad (19)$$

با توجه به این که $q_i = \prod_{e(m,n) \in \tau_i} p_n^m$ و (۱۶) تا (۱۹) خواهیم داشت

$$U\theta_i[x, \underline{q}] = q_i d_i \times \exp\{-\frac{x}{a}(q_i + a(1-q_i))\} + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j \exp\{-\frac{xq_i(1-a)}{a} D(p_n^m)\} + \theta_i[x, \underline{q}] - \theta_i[x, \underline{q}] \sum_{j=1}^r q_j \times d_j \quad (20)$$

اما روشن است که

$$D(p_n^m) = \frac{\prod_{e(m,n) \in S_i \cap S_j} \{p_n^m + a(1-p_n^m)\}}{\prod_{e(m,n) \in S_i \cap S_j} \{p_n^m(1-a)\}} \geq 1$$

با توجه به نزولی بودن تابع e^{-x} داریم

$$\exp\{-\frac{x}{a}q_i(1-a)\} \frac{\prod_{e(m,n) \in S_i \cap S_j} \{p_n^m + a(1-p_n^m)\}}{\prod_{e(m,n) \in S_i \cap S_j} \{p_n^m(1-a)\}} \leq \exp\{-\frac{x}{a}q_i(1-a)\} \quad (21)$$

از (۲۰) و (۲۱) به (۲۲) می رسیم

$$U\theta_i[x, \underline{q}] - \theta_i[x, \underline{q}] \leq q_i d_i \times \exp\{-\frac{x}{a}(q_i + a(1-q_i))\} + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j \times \exp\{-\frac{x}{a}q_i(1-a)\} - \theta_i[x, \underline{q}] \sum_{j=1}^r q_j d_j \quad (22)$$

اما

لازم از یال i است لذا

$$z_{\frac{\alpha}{\tau}} \times \frac{\delta_i}{\sqrt{n_i}} = t_{\frac{\alpha}{\tau}, k-1} \times \frac{S}{(|V|-1)\sqrt{k}} \quad (30)$$

و با حل (۳۰) خواهیم داشت

$$n_i = \left(\frac{z_{\frac{\alpha}{\tau}}}{t_{\frac{\alpha}{\tau}, k-1}}\right)^{\tau} \times \frac{k(|V|-1)^{\tau}}{S^{\tau}} \times \delta_i^{\tau} \quad (31)$$

و نتیجه نهایی در (۳۲) از مجموع‌گیری تعداد نمونه‌ها روی یال‌ها در (۳۱) به دست می‌آید

$$n = \sum_{i=1}^{|E|} n_i = \left(\frac{z_{\frac{\alpha}{\tau}}}{t_{\frac{\alpha}{\tau}, k-1}}\right)^{\tau} \times \frac{k(|V|-1)^{\tau}}{S^{\tau}} \times \sum_{i=1}^{|E|} \delta_i^{\tau} \quad (32)$$



قضیه ۲: اگر n تعداد نمونه‌های مورد نیاز در روش نمونه‌گیری استاندارد و m تعداد نمونه‌های مورد نیاز در روش مبتنی بر eDLA برای تضمین یک بازه اطمینان مشخص و یکسان، بالای 90% باشد، در این صورت یک کران پایین برای نسبت n/m برابر است با

$$\frac{n}{m} \geq |E| \times \frac{\min_i \delta_i^{\tau}}{\max_i \delta_i^{\tau}}$$

که در آن $|E|$ تعداد یال‌های گراف تصادفی است.

اثبات: مطابق لم قبل اگر k تعداد نمونه‌گیری از زیرگراف تصادفی به کمک eDLA و n تعداد نمونه‌های لازم در نمونه‌گیری استاندارد باشد، یک کران پایین برای n به منظور تضمین بازه اطمینان و دقتی معادل با روش eDLA از (۳۲) به دست می‌آید. اما هر نمونه از زیرگراف ایجادشده توسط eDLA حداکثر شامل $|V|-1$ نمونه‌گیری از یال‌های گراف تصادفی است و اگر m تعداد کل نمونه‌های گرفته‌شده از یال‌ها در روش مبتنی بر eDLA باشد، داریم $m \leq k \times (|V|-1)$ و لذا

$$\frac{n}{m} \geq \left(\frac{z_{\frac{\alpha}{\tau}}}{t_{\frac{\alpha}{\tau}, k-1}}\right)^{\tau} \times \frac{|V|-1}{S^{\tau}} \times \sum_{i=1}^{|E|} \delta_i^{\tau} \quad (33)$$

با فرض $1-\alpha \geq 90\%$ به عنوان بازه اطمینان مشترک، $k \geq 120$ و $|V|-1 \geq 5$ داریم

$$t_{\frac{\alpha}{\tau}, k-1} = 1.645 \cong z_{\frac{\alpha}{\tau}}$$

با قراردادن این مقادیر در (۳۳) به (۳۴) می‌رسیم

$$\frac{n}{m} \geq (|V|-1) \times \sum_{i=1}^{|E|} \frac{\delta_i^{\tau}}{S^{\tau}} \quad (34)$$

الگوریتم مبتنی بر eDLA در هر مرحله حداکثر $|V|-1$ یال را از میان مجموعه یال‌های گراف برای ساخت زیرگراف انتخاب می‌کند. اگر طول زیرگراف را با متغیر تصادفی X و طول هر یک از یال‌های گراف نظیر یال i را با متغیر تصادفی Y_i نشان دهیم، داریم

$$X = \sum_{i \in M} Y_i, \quad |M| \leq |V|-1, \quad M \subset E \quad (35)$$

که M نشان‌دهنده زیرگراف تشکیل‌شده در هر مرحله توسط eDLA است و از (۳۵) و بر اساس قضایای محاسبه واریانس (۳۶) نتیجه می‌شود

$$\min_{j \neq i} \frac{d_j}{d_i} \leq \frac{\sum_{j=1, j \neq i}^r q_j \times \frac{d_j}{d_i}}{\sum_{j=1, j \neq i}^r q_j} \leq \max_{j \neq i} \frac{d_j}{d_i} \quad (26)$$

با جایگذاری (۲۶) در (۲۵) شرط ابرمنظم بودن به (۲۷) تقلیل می‌یابد

$$\frac{V[-x(1-q_i)]}{V[xq_i]} \geq \min_{j \neq i} \frac{d_j}{d_i} \quad (27)$$

تعریف می‌کنیم $H[x] = \ln V[x]$. با توجه به محدب‌بودن $H[x]$ می‌توان نشان داد [۲۱]

$$\frac{1}{V[x]} \leq \frac{V[-x(1-q_i)]}{V[xq_i]} \leq V[-x] \quad (28)$$

از آنجا که $V[-x]$ تابعی پیوسته و یکنوا کاهشی با $V[0]=1$ است مقدار $x = x^*$ ای وجود دارد که برای تمام مقادیر در بازه $x \in (0, x^*]$ داریم $\min_{j \neq i} (d_j/d_i) = V[-x^*] \leq V[-x]$ (۲۵) ارضا خواهد شد که در این صورت $\theta_i[x, q]$ ابرمنظم خواهد بود. ■

با توجه به لم ۴ و تعریف تابع $\Phi_i[x, q]$ مشاهده می‌شود که به ازای هر $\varepsilon > 0$ مفروض، ثابت مثبت $a^* < 1$ به گونه‌ای موجود است که نابرابری $1-\varepsilon \leq \Phi_i[x, q] \leq \Gamma_i[q]$ برقرار است و این به آن معناست که $\lim_{n \rightarrow \infty} q_i(n) = 1$ به این ترتیب اثبات قضیه ۱ کامل می‌شود. ■

لم ۵: اگر مجموعه زیرگراف‌های حاصل از به کارگیری eDLA برای حل مسأله زیرگراف بهینه تصادفی مجموعه $\{S_1, S_2, \dots, S_k\}$ با وزن‌های $\{w_1, w_2, \dots, w_k\}$ باشد و نیز با فرض

$$S^{\tau} = \frac{\sum_{i=1}^k w_i^{\tau}}{k-1}$$

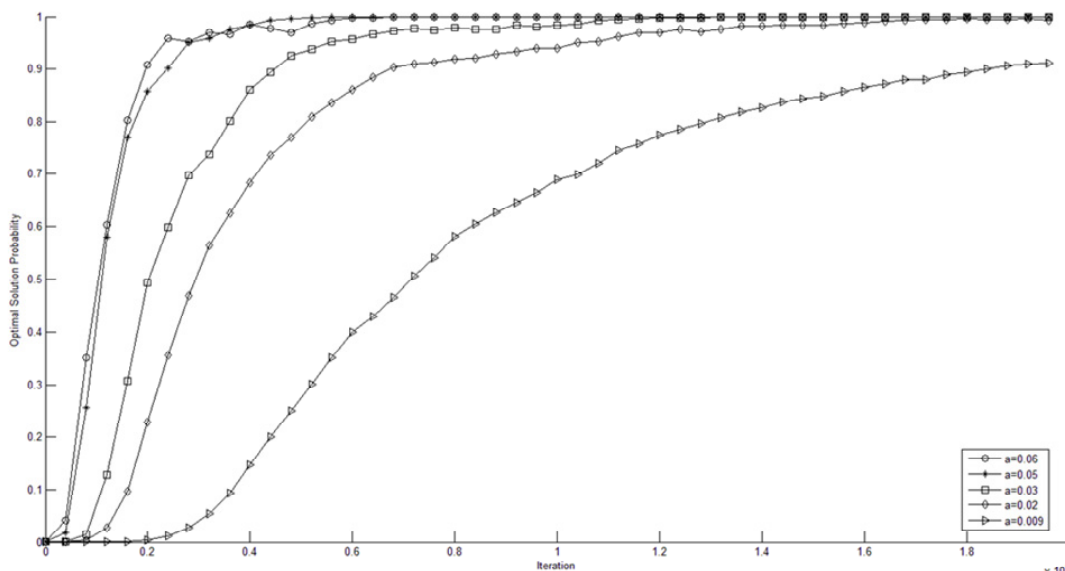
در این صورت تعداد کل نمونه‌های لازم از یال‌ها در روش نمونه‌گیری استاندارد برابر است با

$$n = \left(\frac{z_{\frac{\alpha}{\tau}}}{t_{\frac{\alpha}{\tau}, k-1}}\right)^{\tau} \times \left(\frac{k(|V|-1)^{\tau}}{S^{\tau}}\right) \times \sum_{i=1}^{|E|} \delta_i^{\tau} \quad (29)$$

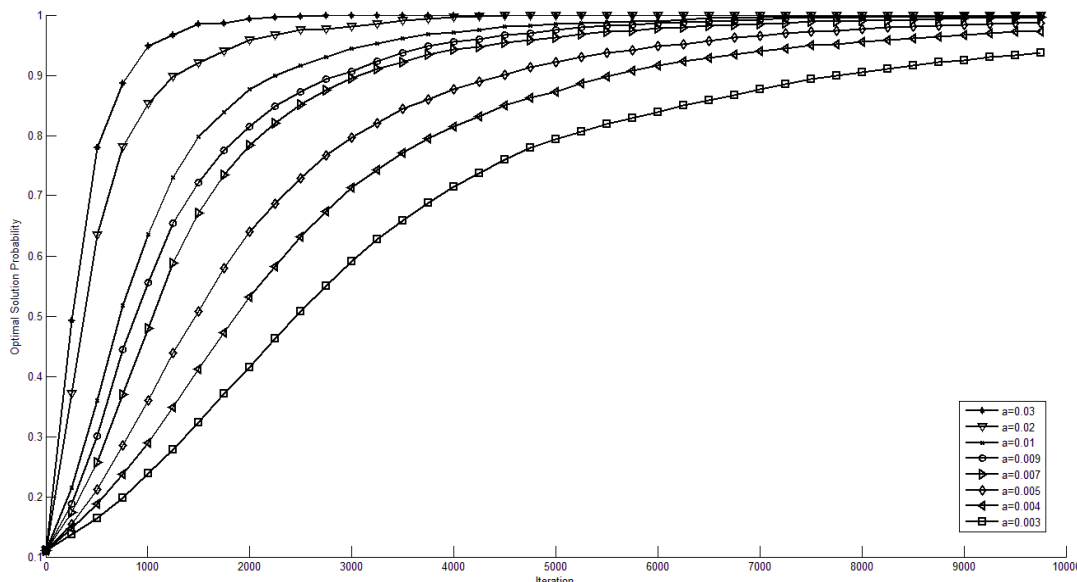
که k تعداد نمونه‌های حاصل از الگوریتم ۱ مبتنی بر eDLA، واریانس یال i و z و t توزیع‌های نرمال و t-student هستند.

اثبات: با توجه به نامشخص‌بودن واریانس جامعه اصلی (وزن زیرگراف‌های بهینه) و بر اساس قضایای مربوط به نمونه‌گیری، می‌توان متغیر تصادفی وزن مجموعه زیرگراف‌های نمونه‌گیری شده توسط eDLA را با یک توزیع t با $k-1$ درجه آزادی تخمین زد [۲۲] و [۲۳]. در این صورت حدود اطمینان بازه $100(1-\alpha)\%$ مبتنی بر این توزیع برابر است با $\pm t_{\alpha/\tau, k-1} \times S/\sqrt{k}$ اما اگر فرض کنیم با استفاده از نمونه‌گیری تصادفی از یال‌های گراف تصادفی نمونه‌گیری کرده باشیم و با استناد به قضیه تصحیح Bonferroni و با توجه به این که بیشترین طول یک زیرگراف $|V|-1$ است، اگر حدود اطمینان برای هر یک از یال‌ها $\pm t_{\alpha/\tau, k-1} \times S/((|V|-1)\sqrt{k})$ در بازه $100(1-\alpha)\%$ باشد در این صورت مطابق حدود اطمینان $\pm t_{\alpha/\tau, k-1} \times S/\sqrt{k}$ برای زیرگراف‌ها با $100(1-\alpha)\%$ تضمین شده است.

اما حدود اطمینان بازه $100(1-\alpha)\%$ برای هر یک از یال‌ها مثل یال i برابر است با $\pm z_{\alpha/\tau} \times \delta_i/\sqrt{n_i}$ که در این رابطه n_i تعداد نمونه‌های



شکل ۶: نحوه تغییر احتمال درخت پوشای کمینه بهینه (جواب مسأله) در حین تکرار الگوریتم در شبکه alex1-b به ازای مقادیر مختلف نرخ یادگیری.



شکل ۷: نحوه تغییر احتمال مسیر بهینه (جواب مسأله) در حین تکرار الگوریتم در گراف Net4 به ازای مقادیر مختلف نرخ یادگیری.

alex1-b از [۲۴] مورد بررسی قرار داده‌ایم. هدف از این آزمایش، بررسی نحوه همگرایی الگوریتم مبتنی بر eDLA به جواب بهینه بوده است. در آزمایش اول از eDLA برای یافتن درخت پوشای کمینه استفاده شده و به ازای هر نرخ یادگیری، الگوریتم دقیقاً ۲۰۰۰۰ دور اجرا شده (۲۰۰۰۰ درخت پوشا انتخاب گردیده) و الگوریتم یادگیری مورد استفاده توسط آتاماتاها الگوریتم خطی L_{R-1} بوده است. این کار ۱۰۰ بار تکرار شده و متوسط احتمال انتخاب مسیر بهینه محاسبه شده که نتایج مقایسه‌ای را در شکل ۶ مشاهده می‌کنید.

در آزمایش دوم از eDLA برای یافتن کوتاه‌ترین مسیر در گراف تصادفی میان یک مبدأ و مقصد مشخص استفاده شده است. در این آزمایش نیز به ازای هر نرخ یادگیری، الگوریتم دقیقاً ۱۰۰۰۰ دور اجرا شده است (۱۰۰۰۰ مسیر). برای محاسبه مقادیر احتمالات مربوط به مسیر بهینه این کار ۱۰۰ دور انجام شده و نتایج متوسط محاسبه شده است. مقادیر احتمال انتخاب مسیر بهینه به ازای مقادیر مختلف نرخ یادگیری را در شکل ۷ مشاهده می‌کنید. در اینجا نیز الگوریتم یادگیری مورد استفاده توسط آتاماتاها، الگوریتم خطی L_{R-1} بوده است.

$$s^r = Var(X) = \sum_i Var(Y_i) \tag{36}$$

$$i \in M, |M| \leq |V| - 1, M \subset E$$

اما داریم $\sum_{i=1}^{|E|} \delta_i^r \geq |E| \times \min_i \delta_i^r$ و $s^r \leq (|V| - 1) \times \max_i \delta_i^r$ و لذا

$$\frac{\sum_{i=1}^{|E|} \delta_i^r}{s^r} \geq \frac{|E|}{|V| - 1} \times \frac{\min_i \delta_i^r}{\max_i \delta_i^r} \tag{37}$$

به این ترتیب از (۳۴) و (۳۷) نتیجه (۳۸) به دست می‌آید

$$\frac{n}{m} \geq |E| \times \frac{\min_i \delta_i^r}{\max_i \delta_i^r} \tag{38}$$

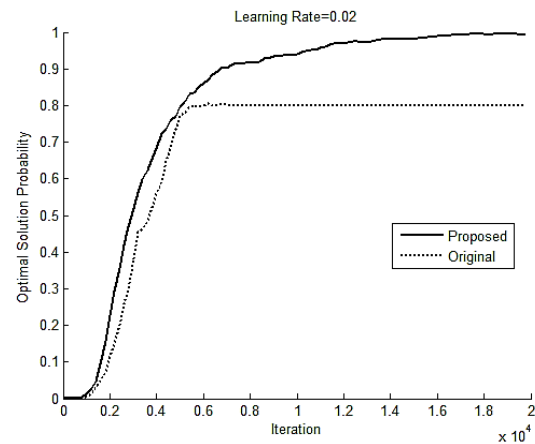
۵- بررسی تجربی

برای بررسی قضیه ۱ در روش پیشنهادی، بر اساس جدول ۱ و الگوریتم ۱ ارائه شده در این مقاله، مسایل یافتن کوتاه‌ترین مسیر تصادفی و درخت پوشای کمینه تصادفی را به ترتیب در گراف‌های Net4 از [۴] و

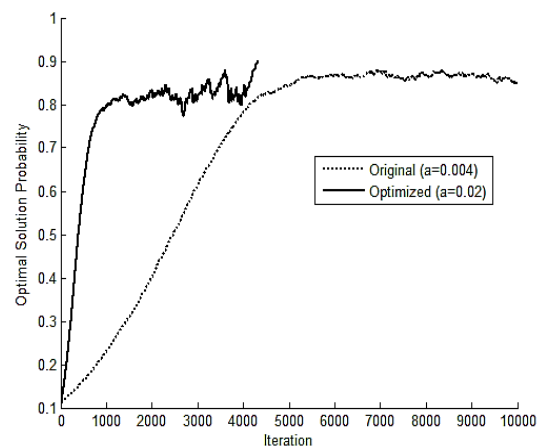
مختلف فعال بودن، محدودیت DLA در نحوه فعال سازی آتاماتاها را مرتفع کرده و از این شبکه از آتاماتاها به عنوان چارچوبی یک دست برای حل مسایل مختلف بهینه سازی روی گراف های تصادفی استفاده کند. بررسی های ریاضی نشان داد که در صورت وجود زیرگراف بهینه، الگوریتم مبتنی بر eDLA با احتمال ۱ به جواب بهینه همگرا می شود. علاوه بر این یک کران پایین برای نسبت تعداد نمونه های لازم در روش استاندارد به روش مبتنی بر eDLA برای یافتن جواب (با دقت یکسان) ارائه شد. به کارگیری eDLA برای حل مسایل بهینه سازی جایگشت تصادفی در دست بررسی است.

مراجع

- [1] K. S. Narendra and M. A. L. Thathachar, "Learning automata: a survey," *IEEE Trans. Syst. Man, Cybernetics*, vol. 14, no. 4, pp. 323-334, 1974.
- [2] M. L. Thathachar and P. S. Sastry, "Varieties of learning automata: an overview," *IEEE Trans. Syst. Man. Cybern. B. Cybern.*, vol. 32, no. 6, pp. 711-722, Jan. 2002.
- [3] H. Beigy and M. Meybodi, Intelligent Channel Assignment in Cellular Networks: A Learning Automata Approach, Amirkabir University of Technology, 2004.
- [4] H. Beigy and M. R. Meybodi, "Utilizing distributed learning automata to solve stochastic shortest path problems," *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, vol. 14, no. 5, pp. 591-615, Oct. 2006.
- [5] M. R. Meybodi and H. Beigy, "A sampling method based on distributed learning automata," in *Proc. the 10th Iranian Conf. on Electrical Engineering*, vol. 1, pp. 618-626, May 2002.
- [6] M. R. Mollakhalili Meybodi and M. R. Meybodi, "A new distributed learning automata based algorithm for solving stochastic shortest path," presented at *6th Conf. on Intelligent Systems*, 2004.
- [7] M. R. Meybodi and H. Beigy, "Solving stochastic shortest path problem using Monte Carlo sampling method: a distributed learning automata approach," *Neural Networks and Soft Computing*, Physica-Verlag HD, pp. 626-632, 2003.
- [8] A. Motevalian and M. R. Meybodi, "Solving maximal independent set problem using distributed learning automata," in *Proc. 14th Iranian Electrical Engineering Conf., ICEE2006*, Tehran, Iran, 16-18 May, 2006.
- [9] A. Alipour and M. R. Meybodi, "Solving traveling salesman problem using distributed learning automata," in *Proc. 10th Annual CSI Computer Conf.*, pp. 759-761, Tehran, Iran, Feb. 2005.
- [10] S. Saati and M. Meybodi, "A self organizing model for document structure using distributed learning automata," in *Proc. 2nd Int. Conf. on Information and Knowledge Technology, IKT2005*, Tehran, Iran, 24-26 May, 2005.
- [11] Z. Anari, M. R. Meybodi, and B. Anari, "Web page ranking based on fuzzy and learning automata," in *Proc. Int. Conf. Manag. Emergent Digit. Ecosyst., MEDES'09*, pp. 162-166, Lion, France, 27-31 Oct. 2009.
- [12] M. R. Mollakhalili Meybodi and M. R. Meybodi, "Link prediction in adaptive web sites using distributed learning automata," in *Proc. 13th Annual CSI Computer Conf. of Iran*, Kish Island, Iran, 9-11, Mar. 2008.
- [13] M. R. Mollakhalili Meybodi and M. R. Meybodi, "A distributed learning automata based approach for user modeling in adaptive hypermedia," in *Proc. Congress on Electrical, Computer, and Information Technology*, 2012.
- [14] A. Baradaran Hashemi and M. R. Meybodi, "Web usage mining using distributed learning automata," in *Proc. 12th Annual CSI Computer Conf. of Iran*, pp. 553-560, 2007.
- [15] S. Misra and B. J. Oommen, "Dynamic algorithms for the shortest path routing problem: learning automata - based solutions," *IEEE Trans. Syst. Man. Cybern. B. Cybern.*, vol. 35, no. 6, pp. 1179-1192, Dec. 2005.
- [16] K. Liu, Stochastic Online Learning for Network Optimization under Random Unknown Weights, 18 pp. 1-18.
- [17] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction*, Prentice Hall, p. 476, 1989.
- [18] S. Lakshminarayanan and M. A. L. Thathachar, "Absolutely expedient learning algorithms for stochastic automata," *IEEE Trans. Syst. Man Cybern.*, vol. 6, pp. 281-286, 1973.



شکل ۸: مقایسه تغییرات احتمال درخت پوشای بهینه در الگوریتم پیشنهادی (Proposed) و الگوریتم ارائه شده در [۲۵] (Original) در نرخ یادگیری ۰/۰۲ در گراف $\lambda=1-b$.



شکل ۹: مقایسه سرعت همگرایی الگوریتم پیشنهادی مقاله (Optimal) در برابر الگوریتم مبتنی بر DLA (Original) [۴] (در حالت حداقل های لازم برای همگرایی ۱۰۰٪).

مقادیر آستانه مقایسه برای تنبیه یا پاداش به زیرگراف انتخاب شده توسط eDLA نیز در هر دو الگوریتم بر اساس پیشنهاد ارائه شده در این مقاله و با دخالت دادن واریانس محاسبه شده است.

شکل های ۶ و ۷ صحت قضیه ۱ را نشان می دهند. علاوه بر این نشان می دهند روش فعلی ارائه شده بر خلاف روش های قبلی برای حل این مسایل در [۴] و [۲۰] تقریباً مستقل از نرخ یادگیری، سرعت همگرایی بالاتری دارد.

مقایسه الگوریتم پیشنهادی برای حل مسایل درخت پوشای کمینه تصادفی و کوتاه ترین مسیر تصادفی که به ترتیب در [۲۵] و [۴] ارائه شده اند حاکی از عملکرد بهتر روش پیشنهادی است. در آزمایش انجام شده در مورد درخت پوشای کمینه تصادفی این بهبود می تواند هم ناشی از ساختار پیشنهادی و هم ناشی از معیار جدید مقایسه ای باشد (شکل ۸) در حالی که در مورد کوتاه ترین مسیر تصادفی به دلیل این که DLA و eDLA یکسان عمل می کنند، بهبود نتایج قطعاً ناشی از معیار مقایسه ای جدید است (شکل ۹).

۶- نتیجه گیری

در این مقاله، چارچوبی جدید مبتنی بر آتاماتاها یادگیر موسوم به eDLA برای حل مسأله یافتن زیرگراف بهینه در گراف های تصادفی ارائه گردید. نشان داده شد که این ساختار می تواند از طریق تعریف سطوح

محمد رضا ملاخیلی میبدی تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد مهندسی کامپیوتر به ترتیب در سال های ۱۳۸۰ و ۱۳۸۲ از دانشگاه های شهید بهشتی و صنعتی امیرکبیر تهران و دکترای مهندسی کامپیوتر را در سال ۱۳۹۳ در واحد علوم و تحقیقات دانشگاه آزاد به پایان رسانده است. وی هم‌اکنون عضو هیأت علمی دانشگاه آزاد اسلامی واحد میبد است. زمینه های تحقیقاتی مورد علاقه ایشان عبارتند از: شبکه های کامپیوتری و وب، شبکه های مبتنی بر نرم افزار، محاسبات نرم و کاربردهای آن، یادگیری و الگوریتمها، گراف های تصادفی و شبکه های پیچیده.

محمد رضا میبدی تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد اقتصاد به ترتیب در سال های ۱۳۵۲ و ۱۳۵۶ از دانشگاه شهید بهشتی و در مقاطع کارشناسی ارشد و دکترای علوم کامپیوتر به ترتیب در سال های ۱۳۵۹ و ۱۳۶۲ از دانشگاه اوکلاه های آمریکا به پایان رسانده است و هم‌اکنون استاد دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر می باشد. نام برده قبل از پیوستنش به دانشگاه صنعتی امیرکبیر در سال های ۱۳۶۲ الی ۱۳۶۴ استادیار دانشگاه میشیگان غربی و در سال های ۱۳۶۴ الی ۱۳۷۰ دانشیار دانشگاه اوهایو در ایالات متحده آمریکا بوده است. زمینه های تحقیقاتی مورد علاقه ایشان عبارتند از: الگوریتم های موازی، پردازش موازی، محاسبات نرم و کاربردهای آن، شبکه های کامپیوتری و مهندسی نرم افزار.

- [19] M. A. L.Thathachar and B. Harita, "Learning automata with changing number of actions," *IEEE Trans. Syst. Man, Cybern.-Part A Syst. Humans*, vol. 17, no. 6, pp. 1095-1100, Dec. 1987.
- [20] J. Akbari Torkestani and M. R. Meybodi, "A learning automata-based heuristic algorithm for solving the minimum spanning tree problem in stochastic graphs," *J. Supercomput.*, vol. 59, no. 2, pp. 1035-1054, Oct. 2012.
- [21] F. Norman, "On the linear model with two absorbing," *J. Math. Psychol.*, vol. 5, pp. 225-241, Jun. 1968.
- [22] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. New York, USA: McGrawHill, 1991.
- [23] S. M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*, 3rd. Ed., Elsevier Academic Press, 2004.
- [24] K. R. Hutson and D. R. Shier, *Online Supplement to 'Bounding Distributions for the Weight of a Minimum Spanning Tree in Stochastic Networks'*, 12 pp.
- [25] J. Akbari Torkestani and M. R. Meybodi, "Learning automata -based algorithms for solving stochastic minimum spanning tree problem," *Appl. Soft Comput.*, vol. 11, no. 6, pp. 4064-4077, Sep. 2011.