

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

محمد رضا ملاخیلی میبدی، محمدرضا میبدی

شده، وابسته کردن انتخاب عامل فعال بعدی (که می‌بایستی در محیط اقامتی 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 روش تاکنون توانسته مسائل مختلف مرتبط با گراف‌های تصادفی نظری یافتن کوتاه‌ترین مسیر تصادفی^[۷] یا حل مساله فروشنده دوره گرد و یافتن مجموعه مستقل ماکزیمال^[۸] را حل کند. علاوه بر آن در موارد دیگری مثل دسته بندي اسناد وب^[۹] محاسبه رتبه اسناد^[۱۰] پیش بینی حرکت کاربران در وب^[۱۱] و مدل‌سازی کاربران در محیط‌های فرآپیوندی نظری وب^[۱۲]، کاوش در نحوه استفاده از وب^[۱۳] نیز مورد استفاده قرار گرفته است. اما عملاً نحوه فعالیت آتاماتاهای DLA، سیستم چندعامله توزیع شده مبتنی بر آتاماتای یادگیر را با محدودیت‌هایی مواجه کرده است. در این مقاله، به معروفی eDLA که توسعه ای بر نحوه فعال شدن آتاماتاهای DLA است پرداخته ایم.

ادامه این مقاله به صورت زیر سازماندهی شده است: در بخش دوم به بررسی پیش نیازهای لازم برای ادامه مقاله پرداخته‌ایم. در بخش سوم eDLA را بررسی کرده و سپس الگوریتم پیشنهادی مبتنی بر eDLA برای حل مساله یافتن زیرگراف بهینه را ارائه کرده‌ایم. بخش چهارم به بررسی ویژگی‌های ریاضی الگوریتم پیشنهادی اختصاص یافته است. در بخش ۵ نتیجه‌گیری و پیشنهادهایی برای کارهای بعدی ارائه شده است.

۲- مبانی ریاضی و مفاهیم مقدماتی

در این بخش به بررسی برخی از مفاهیم موردنیاز برای ادامه مقاله خواهیم پرداخت

۱-۲ 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 است و برخی دیگر تعاریف عمومی مفاهیم هستند.

تعریف ۱: گراف با سه تایی $(G, E(G), I_G)$ تعریف می‌شود که در آن $V(G) \neq \emptyset$ و $E(G) \neq \emptyset$ و I_G یک

چکیده - در این مقاله، یک ساختار جدید شبکه‌ای از آتاماتاهای یادگیر موسوم به آتاماتای یادگیر توزیع شده توسعه یافته معرفی می‌شود. سپس الگوریتمی مبتنی بر این ساختار شبکه‌ای برای حل مساله زیرگراف بهینه در گراف‌های تصادفی با یال‌های وزن‌دار از طریق نمونه‌گیری ارائه می‌شود. نشان داده می‌شود که ساختار شبکه‌ای جدید پیشنهادی قادر به حل مسائل بهینه سازی 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 می‌شود.

کلید واژه - آتاماتای یادگیر، آتاماتای یادگیر توزیع شده توسعه یافته، شبکه آتاماتاهای یادگیر، زیرگراف، گراف تصادفی، نمونه گیری.

۱- مقدمه

مدلهای آتاماتایی برای سیستم‌های یادگیر در دهه ۶۰ میلادی معرفی شدند و با نام آتاماتای یادگیر در مقاله^[۱] که به بررسی آنها می‌پرداخت، به محبوبیت رسیدند^[۲]. از آن زمان تاکنون توسعه‌های مختلفی چه در زمینه تئوری و چه در حوزه کاربرد آتاماتای یادگیر رخ داده است. در^[۲]،^[۳] نویسنده‌گان به تفصیل به بررسی این تحولات پرداخته اند.

آتاماتای یادگیر، اساساً عاملی ساده برای انجام کارهای ساده است.^[۳] قابلیت کامل یک آتاماتای یادگیر، در یک شبکه از آتاماتاهای و در تعامل آنها با یکدیگر ظهرور بیدا می‌کند. یکی از ساختارهای متشكل از آتاماتاهای 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 آتاماتاهای یادگیر همکار که برای حل یک مساله خاص با یکدیگر همکاری دارند، بارها و بارها در حل مسائل مختلف، خصوصاً مسائل مربوط به گراف‌های تصادفی با یال‌های وزن‌دار یا رؤس وزن‌دار مورد استفاده قرار گرفته است^{[۴]-[۶]}

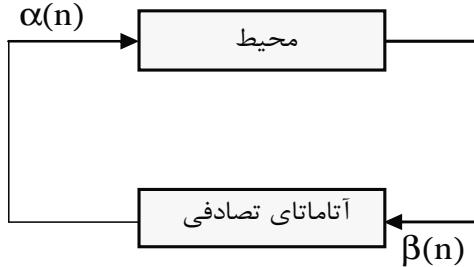
یکی از اشکالات مشهود DLA به عنوان یک سیستم چندعامله توزیع

این مقاله در تاریخ دوازدهم خرداد ۱۳۹۲ دریافت شد.

محمد رضا ملاخیلی میبدی، دانشکده مهندسی کامپیوتر، دانشگاه آزاد اسلامی، واحد علوم و تحقیقات تهران (email: m.meybodi@srbiau.ac.ir)

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

سیگنال تقویتی بازخوردی، مورد ارزیابی قرار می‌دهد. آتماتای یادگیر بر اساس سیگنال بازخوردی حاصل، بردار توزیع احتمال انتخاب اقدام‌ها را به روزرسانی می‌کند. هدف آتماتا، پیدا کردن اقدام بهینه در میان مجموعه اقدام‌های قابل انجام است؛ اقدامی که بیشترین پاداش را از محیط دریافت کند.



شکل ۱: آتماتای یادگیر تصادفی

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

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

به طوریکه :

مجموعه اقدام‌های آتماتا/مجموعه ورودی‌های محیط:

$$\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$$

مجموعه ورودی‌های آتماتا/مجموعه خروجی‌های محیط:

$$\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$$

بردار احتمال اقدام‌های آتماتا:

$$p \equiv \{p_1, p_2, \dots, p_r\}$$

الگوریتم یادگیری: $\alpha \rightarrow \varphi \rightarrow \alpha$

مجموعه احتمالات جریمه که معرف محیط می‌باشد.

$$c \equiv \{c_1, c_2, \dots, c_r\}$$

الگوریتم یادگیری یک رابطه بازگشتی است که برای انجام تغییرات و به روزرسانی در بردار احتمال اقدام‌های آتماتا در یک آتماتای یادگیر تصادفی با ساختار متغیر موردن استفاده قرار می‌گیرد. فرض کنید یک آتماتای یادگیر تصادفی ساختار متغیر در زمان k از میان k اقدام‌های α عمل $(k)_i$ را انتخاب کرده باشد. همچنین فرض کنید بردار احتمال انتخاب اقدام‌های آتماتا را با $p(k)$ نمایش داده‌یم. 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 آتماتات یادگیر باشد، بردار $p(k)$ توسط الگوریتم یادگیری خطی ارائه شده در روابط زیر به روزرسانی می‌شود. مقدار a را پارامتر پاداش و b را پارامتر جریمه می‌نامند

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

نگاشت وقوع است که به هر عضو (G) یک زوج نامرتب (یکسان یا متمایز) از V نظیر می‌کند. عناصر (G) V را رئوس گراف G و عناصر (G) V را یال‌های G می‌نامند. اگر برای یال e از G داشته باشیم $e = uv$ را پایان‌های e می‌نامیم. در ادامه این مقاله وجود I_G تصریح نمی‌شود.

تعریف ۲: گراف H را زیرگراف G می‌نامیم هرگاه $E(H) \subseteq E(G)$ و $I_H \subseteq I_G$ 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 یک سه تایی $G_R = (V, E, W)$ تعریف می‌شود که در آن مجموعه‌های $E = \{e_1, e_2, \dots, e_m\} \subseteq V \times V$ و $V = \{v_1, v_2, \dots, v_n\}$ به مجموعه $W = \{w_1, w_2, \dots, w_n\}$ وزن‌های منتبه به یال‌ها را نشان می‌دهند. ترتیب، مجموعه رئوس و مجموعه یال‌های گراف را نشان می‌دهند. نشان می‌دهد به‌گونه‌ای که متغیر تصادفی w_i نشان‌دهنده وزن یال e_i است.

تعریف ۵: دوگان خنثی یک گراف تصادفی، گرافی است هم ریخت 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 $G'_R = (V', E')$ که در آن $V' = V$ و $E' = E$ را دوگان خنثی نامیده می‌شود.

۲-۲ کاربرد گراف تصادفی و زیرگراف بهینه تصادفی

گرافهای وزن‌دار، 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 وزن‌دار مدل کرد. مثلاً یک شبکه کامپیوتری را در نظر بگیرید. وزن یال‌ها می‌تواند تاخیر لینک‌های یک شبکه را نمایش دهد. بدینهای است که در این مثال، وزن یال با همان تاخیر، یک متغیر تصادفی است که از یکتابع توزیع احتمال نامعلوم تبعیت می‌کند. یافتن مسیری با کمترین تاخیر در این شبکه می‌تواند به یافتن یک کوتاهترین مسیر تصادفی نگاشت شود. [۱۵، ۱۶]

۱-۲ آتماتای یادگیر

آتماتای یادگیر تصادفی یک واحد تصمیم‌گیرنده تطبیقی است که فرآیند یادگیری در آن، از طریق تعاملش با محیط صورت می‌گیرد. آتماتای یادگیر، مجموعه‌ای از اقدام‌های قابل انجام دارد. 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

واضح است که برقراری رابطه $E[M(n+1)/P(n)] \geq M(n)$ ایجاب میکند که $M(n)$ یک زیرمارتنگل باشد.

آتاماتای یادگیر با مجموعه اقدام‌های متغیر:

یک آتاماتای یادگیر با مجموعه اقدام متغیر، آتاماتایی است که در آن تعداد اقدام‌های موجود هر آتاماتا در طول زمان تغییر می‌کند [۱۹]. فرض کنید مجموعه اقدام‌های آتاماتا را با $\underline{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ و مجموعه اقدام‌های قابل انجام توسط آتاماتا در زمان n را با $A(k) \subseteq \alpha$ نشان دهیم.

$$(A(k)) \in \{A_1, A_2, \dots, A_m\} \quad m = 2^{n-1}$$

انتخاب زیرمجموعه $A(k)$ از میان مجموعه اقدام‌های قابل انجام توسط آتاماتا، غالباً توسط شرایط بیرونی تحمیل می‌شود. می‌توان این گونه فرض کرد که این زیرمجموعه از اقدام‌های آتاماتا را که از این به بعد مجموعه اقدام‌های فعال آتاماتا نامیده می‌شود، توسط یک عامل بیرونی و با توزیع احتمال $\{\psi_1(k), \psi_2(k), \dots, \psi_m(k)\}$ انتخاب می‌شود.

اگر تعریف کنیم

$$\hat{p}_i(k) = \text{Prob}[\alpha(k) = \alpha_i | A(k)], \alpha_i \in A(k)$$

(احتمال انتخاب عمل α_i از میان مجموعه اقدام‌های فعال، به شرط آنکه مجموعه اقدام‌های فعال آتاماتا برابر با $A(k)$ باشد) در این صورت

$$\hat{p}_i(k) = p_i(k)/K(k) \quad (5)$$

در (۵) $K(k)$ مجموع احتمالات تمام اقدام‌های فعال (اقدام‌های عضو آنکه مجموعه اقدام‌های آتاماتا $A(k)$ است و داریم $p_i(k) = \text{Prob}[\alpha(k) = \alpha_i]$) احتمال انتخاب اقدام α_i در میان مجموعه تمام اقدام‌های آتاماتا (اعم از اینکه فعال باشد یا نباشد) است. بدین ترتیب نحوه انتخاب اقدام توسط آتاماتا یا مجموعه اقدام متغیر بدین صورت است که: فرض کنید آتاماتا مجموعه اقدام‌های فعال $A(k)$ را داشته باشد. ضریب نرمال کننده $K(k) = \sum_{\alpha_i \in A(k)} p_i(k)$ برای این مجموعه از اقدام‌های فعال محاسبه شده و بردار احتمال انتخاب اقدام‌های آتاماتا مطابق با رابطه بالا نرمال سازی می‌شود (به گونه‌ایکه مجموع آتاماتا انتخاب اقدام‌های فعال همچنان ۱ باشد). پس از این، آتاماتا بر اساس بردار جدید $\hat{p}_i(k)$ یکی از اقدام‌های (فعال) خود را انتخاب کرده و به محیط اعمال می‌کند. در مرحله بهروزرسانی نیز بردار $\hat{p}_i(k)$ مطابق با الگوریتم یادگیری مورد استفاده توسط آتاماتا بهروزرسانی می‌شود و سپس با استفاده مجدد از رابطه بالا بردار $p_i(k) = \hat{p}_i(k) \cdot K(k)$ به دست می‌آید. نشان داده شده است [۱۹] که در صورتی که الگوریتم یادگیری مورد استفاده L_{R-I} باشد، این روش ویژگی‌های ε -optimality و absolute expediency را دارد.

آتاماتای یادگیر توزیع شده:

آتاماتای یادگیر توزیع شده شبکه‌ای است از آتاماتاهای یادگیر که برای حل یک مساله خاص با یکدیگر همکاری دارند. یک DLA در قالب یک گراف ارائه می‌شود و با چندتایی (V, E, T, v_0) تعریف می‌شود که در آن V

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

رابطه (۲) زمانی مورد استفاده قرار می‌گیرد که عمل $(\alpha_i)_i$ 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} می‌نامند. اگر $b \gg a$ باشد آن را $L_{R-\varepsilon P}$ و اگر $b=0$ باشد آن را L_{R-I} می‌نامند.

معیارهای کارآیی آتاماتای یادگیر:

گرچه یادگیری اصولاً راهیافتی کیفی است، اما معیارهایی عددی وجود دارند که فرآیند یادگیری را توصیف می‌کنند [۱۷]. برای اندازه‌گیری کارآیی آتاماتای یادگیر تصادفی، معیارهای معینی تعریف شده‌اند که تاثیر آتاماتا را مشخص کرده و امکان مقایسه روش‌های مختلف یادگیری را فراهم می‌آورند. یکی از این معیارها متوسط پاداش دریافتی توسط آتاماتا پاداش دریافتی توسط آتاماتای یادگیر در زمان n به صورت (۴) تعریف می‌شود.

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

$$= \sum_{j=1}^r p(\beta(n) = 1 | \alpha(n) = \alpha_j) p(\alpha(n) = \alpha_j)$$

اگر انتخاب اقدام‌های آتاماتا به شکل کاملاً تصادفی انجام شود (آتاماتای شناسی محض) داریم $M(n) = M_0 = \sum_{j=1}^r d_j / r$. زمانی که آتاماتا در حال یادگیری است، انتظار آن است که متوسط پاداشی بهتر از این مقدار داشته باشد.

تعريف ۶: آتاماتای یادگیر Expedient نامیده می‌شود اگر $\lim_{n \rightarrow \infty} E[M(n)] > M_0$ 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:

تعريف ۷: آتاماتای یادگیر را بهینه یا optimal گوییم اگر $d_m = \max_i \{d_i\}$ باشد که $\lim_{n \rightarrow \infty} E[M(n)] = d_m$ است. تعریف معادل دیگری برای بهینگی عبارتست از $\lim_{n \rightarrow \infty} p_m(n) = 1$ (with probability 1).

تعريف ۸: آتاماتای یادگیر ε -optimal نامیده می‌شود اگر $\lim_{n \rightarrow \infty} E[M(n)] > d_m - \varepsilon$ با ازای هر مقدار $\varepsilon > 0$ با انتخاب مناسب

پارامترهای آتاماتای یادگیر قابل حصول باشد.

تعريف ۹: یک آتاماتای یادگیر 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 حالتیکه تمام d_i ها مقادیر برابر داشته باشند) [۱۸].

$S = \{s_1, s_2, \dots, s_n\}$ مجموعه‌ای از مقادیر موسوم به سطح فعالیت است که سطح فعال بودن هر آتماتای متعلق به A را مشخص می‌کند. هر یک از s_i ‌ها می‌توانند یکی از مقادیر Active و Passive باشند که آنها را به ترتیب با $s_{i,Pa}$ و $s_{i,Of}$ نمایش می‌دهیم. Fire، بالاترین سطح فعالیت و Passive و off پایین ترین سطح فعالیت هستند با این تفاوت که اگر آتماتا به سطح فعالیت off برسد، دیگر هرگز تغییر وضعیت نمی‌دهد اما در حالت Passive امکان تغییر وضعیت وجود دارد.

-۳: مجموعه‌ای محدود از قواعد حاکم بر سطح فعالیت آتماتاها است. این قواعد تابعی از سطح فعالیت فعلی هر آتماتا و سطح فعالیت آتماتاهای مجاور است و بسته به مساله‌ای که eDLA برای حل آن به کار گرفته می‌شود، می‌تواند متغیر باشد.

-۴: $S^0 = \{s_1^0, s_2^0, \dots, s_n^0\}$ وضعیت آغازین سیستم نامیده می‌شود. $F = \{S^F | S^F = \{s_1^F, s_2^F, \dots, s_n^F\}\}$ مجموعه شرایط پایانی نامیده می‌شود. مجموعه F شرایطی را برحسب سطح فعالیت آتماتاها مشخص می‌کند که در صورت تحقق حداقل یکی از آنها سیستم به وضعیت پایانی منتقل می‌شود. F حداقل یک عضو بدیهی دارد که در آن سطح فعالیت تمام آتماتاها off است.

-۵: یکتابع خاص است که بر اساس سطح فعلی فعالیت آتماتاها و مساله‌ای که eDLA برای حل آن طراحی شده است، آتماتایی را از میان مجموعه آتماتاهای موجود در eDLA انتخاب می‌کند. C را تابع آتش یا Fire Function می‌نامیم. در ادامه بحث در این خصوص بیشتر توضیح خواهیم داد

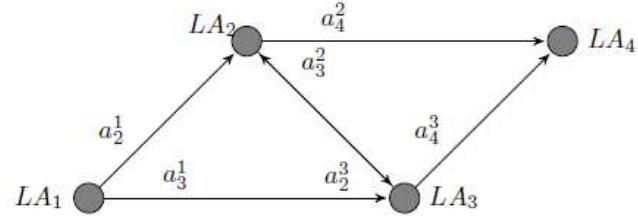
هر آتماتای یادگیر در eDLA متناسب با درجه‌اش در گراف ارتباطی (در حالتی که گراف ارتباطی G فاقد جهت باشد) و یا درجه خروجی‌اش در گراف ارتباطی (در حالتی که گراف ارتباطی G جهتدار باشد) اقدام برای انجام دادن دارد.

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 آتماتاهای اقدام می‌کند. تغییر سطح فعال بودن آتماتاهای تا زمان حل کامل Fire مساله ادامه پیدا می‌کند. در هر زمان آتماتایی که در سطح فعالیت قراردارد، اقدامی را انتخاب کرده و آن را به محیط اعمال می‌کند. ضمن این‌که، بر اساس قواعد ارتباطی مجموعه‌ای از آتماتاهای مجاور با آتماتای Fire که در پایین ترین سطح فعال بودن (Passive) قرار دارند، ارتقای سطح می‌یابند (به سطح Active). هر آتماتا پس از Fire شدن و انجام اقدام به وضعیت Off تغییر وضعیت می‌دهد. انتخاب آتماتای Fire از میان مجموعه آتماتاهای Active صورت می‌گیرد و غالباً به صورت تصادفی انجام می‌شود.

تعریف ۱۱: یک دور اجرا یا run: یک اجرا یا run عبارت است از شدن وضعیت تمام آتماتاهای موجود در eDLA از طریق طی کردن زیست چرخ حیات آتماتا به شکل

مجموعه رئوس گراف، E مجموعه یال‌ها، T مجموعه‌ای از الگوریتم‌های یادگیر مورد استفاده توسط آتماتاها در DLA و v_0 گره ریشه DLA است. [۳]

در این شبکه از آتماتاهای همکار در هر زمان تنها یک آتماتا فعال است. تعداد اعمال قابل انجام توسط یک آتماتا در DLA برابر است با تعداد آتماتاهایی که به این آتماتا متصل شده‌اند. انتخاب یک عمل توسط آتماتا در این شبکه، باعث شدن آتماتای متصل شده به این آتماتا و متناظر با این عمل می‌گردد. به عبارت معادل، انتخاب یک عمل توسط یک آتماتا در این شبکه متناظر است با فعال شدن یک آتماتای دیگر در این شبکه.



شکل ۲: آتماتای یادگیر توزیع شده

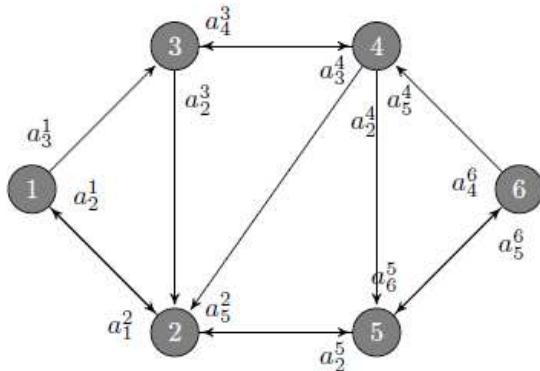
وجود یال (j) در این گراف به آن معناست که انتخاب عمل α_j توسط LA_i باعث شدن LA_j می‌گردد.

تعداد اعمال قابل انتخاب توسط LA_k برابر است با درجه خروجی این راس. بردار احتمالات مربوط به عملهای قابل انجام توسط آتماتای LA_k به صورت $p^k = \{p_1^k, p_2^k, \dots, p_{r_k}^k\}$ نمایش داده می‌شود. در این مجموعه عدد p_m^k نشانده‌نده احتمال مربوط به عمل α_m^k است. انتخاب عمل α_m^k توسط LA_k باعث شدن LA_m می‌شود. r_k تعداد اعمال قبل انجام توسط آتماتای LA_k را نشان می‌دهد.

۳- ساختار و الگوریتم پیشنهادی جدید:

آتماتای یادگیر توزیع شده توسعه یافته شبکه‌ای است از آتماتاهای یادگیر مرتبط با هم که تحت نظرارت مجموعه‌ای از قواعد ارتباطی که ناظر بر ترتیب و نحوه فعالیت آنهاست، برای حل یک مساله خاص با یکدیگر همکاری دارند. در آتماتای یادگیر توزیع شده توسعه یافته، هر آتماتا دارای یک سطح فعال بودن است و زمان و نحوه تغییر سطح فعالیت آتماتاهای به تبعیت از شرایط مساله، تغییر می‌کند. در آتماتای یادگیر توزیع شده توسعه یافته، در هر زمان تنها یک آتماتا در سطحی قرار می‌گیرد که قادر به انجام اقدام و اعمال آن بر محیط است. به صورت فرمال می‌توان یک آتماتای یادگیر توزیع شده توسعه یافته را به صورت زیر تعریف کرد

تعریف ۱۰: یک آتماتای یادگیر توزیع شده توسعه یافته توسط چندتایی $eDLA = \{A, E, S, P, S^0, F, C\}$ تعریف می‌شود و در آن:
-۱: $A = \{1, 2, \dots, n\}$: مجموعه‌ای از آتماتاهای یادگیر است که در آن $E \subseteq A \times A$ مجموعه ارتباطات سلسله مرتبی میان اعضای A است.
-۲: یک گراف ارتباطی نامیده می‌شود.



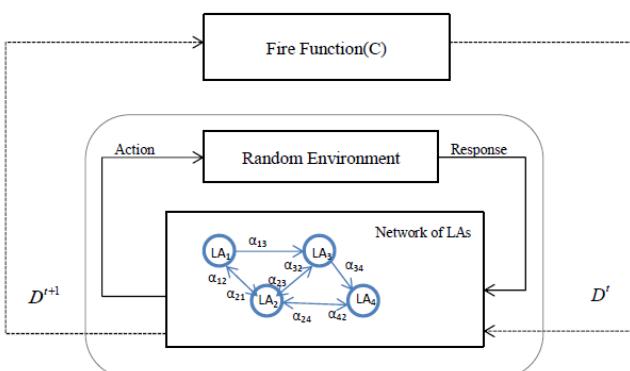
شکل ۳: گراف ارتباطی یک eDLA نمونه

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

$$\begin{aligned}
 & (\emptyset, \emptyset, \{1\}, \{2, 3, 4, 5, 6\}) \vdash_{\text{fire} 1} (\emptyset, \{1\}, \{2, 3\}, \{4, 5, 6\}) \vdash^{\alpha_{12}} \\
 & (\{1\}, \emptyset, \{2, 3\}, \{4, 5, 6\}) \vdash_{\text{fire} 3} (\{1\}, \{3\}, \{2, 4\}, \{5, 6\}) \vdash^{\alpha_{32}} \\
 & (\{1, 3\}, \emptyset, \{2, 4\}, \{5, 6\}) \vdash_{\text{fire} 4} (\{1, 3\}, \{4\}, \{2, 5, 6\}, \emptyset) \vdash^{\alpha_{46}} \\
 & (\{1, 3, 4\}, \emptyset, \{2, 5, 6\}, \emptyset) \vdash_{\text{fire} 2} (\{1, 3, 4\}, \{2\}, \{5, 6\}, \emptyset) \vdash^{\alpha_{25}} \\
 & (\{1, 3, 4, 2\}, \emptyset, \{5, 6\}, \emptyset) \vdash_{\text{fire} 6} (\{1, 3, 4, 2\}, \{6\}, \{5\}, \emptyset) \vdash^{\alpha_{65}} \\
 & (\{1, 3, 4, 2, 6\}, \emptyset, \{5\}, \emptyset) \vdash_{\text{fire} 5} (\{1, 3, 4, 2, 6\}, \{5\}, \emptyset, \emptyset) \\
 & \vdash^{\text{no action}} (\{1, 3, 4, 2, 6, 5\}, \emptyset, \emptyset, \emptyset)
 \end{aligned}$$

به این ترتیب، آتاماتای یادگیر توزیع شده توسعه یافته یک ساختار دوبخشی است، متشکل از یک تابع تصمیم‌گیرنده و یک شبکه از آتاماتاهای یادگیر که برای حل یک مساله خاص به شکل گروهی با یکدیگر همکاری دارند. بسته به نوع مساله، فرآیند تصمیم‌گیری متفاوت است. 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



شکل ۴: ساختار کلی eDLA

در شکل ۵ الگوریتم پیشنهادی متى بر eDLA برای حل مساله زیرگراف بهینه ارائه شده است. نحوه عملکرد این الگوریتم بسیار ساده است؛ تابع fire مساله را روی یک گراف قطبی 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

$\text{Passive} \rightarrow \text{Active} \rightarrow \text{Fire} \rightarrow \text{Off}$

در آغاز هر اجرا، بر اساس S^0 آتاماتایی در eDLA وجود دارد که در وضعیت Active قرارداد. این آتاماتا را آتاماتایی ریشه نیز می‌نامیم و با A_0 نمایش می‌دهیم. هر آتاماتا در eDLA یک چرخه زیستی مختص به خود را طی می‌کند. تغییر وضعیت آتاماتاهای در eDLA بر اساس قوانین P و تغییراتی که توسط تابع Fire اعمال می‌شوند، است:

الف- تمامی آتاماتاهای به غیر از آتاماتایی متناظر را ریشه- در آغاز کار در وضعیت Passive قراردارند.

ب- تابع Fire مشخص می‌کند که کدام آتاماتا از مجموعه آتاماتاهای Active به وضعیت Fire تبدیل شوند.

ج- با شدن وضعیت یک آتاماتا، کلیه آتاماتاهای مجاور آن که هستند به حالت Active تغییر وضعیت می‌دهند (فعال سازی به شیوه همسایگی محلی بر اساس قواعد P)

د- آتاماتایی که شده از میان مجموعه اقدامات 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.

ه- آتاماتایی که شده است پس از انجام عمل تصادفی انتخابی، به وضعیت Off تغییر وضعیت می‌دهد.

تعریف ۱۳: توصیف لحظه‌ای:

یک اجرا یا Run را می‌توان با توصیف‌های لحظه‌ای وضعیت آتاماتاهای نشان داد. توصیف لحظه‌ای عبارت است از یک چهارتایی مرتب که با

$$D^t = (D_{Of}^t, D_{Fi}^t, D_{Ac}^t, D_{Pa}^t)$$

$$D_i^t \subseteq A \quad (i = \{Of, Fi, Ac, Pa\}), \bigcup_i D_i^t = A$$

۹

$$D_i^t \cap D_j^t = \emptyset \quad (i \neq j, i, j \in \{Of, Fi, Ac, Pa\})$$

هربار که آتاماتا یکی از اقدامات خودش را انجام می‌دهد توصیف Fire لحظه‌ای بر اساس قواعد P تغییر می‌کند. در حالت دیگر هرگاه تابع Fire یکی از آتاماتاهای را Fire می‌کند، توصیف لحظه‌ای تغییر می‌کند. علاوه بر این داریم:

$$D^0 = (D_{Of}^0, D_{Fi}^0, D_{Ac}^0, D_{Pa}^0) = (\phi, \phi, \{A_0\}, A - A_0)$$

که در این رابطه D^0 توصیف لحظه شروع eDLA است و

$$D^{final} = (D_{Of}^{final}, D_{Fi}^{final}, D_{Ac}^{final}, D_{Pa}^{final}) = (A, \phi, \phi, \phi)$$

وضعیت پایانی بدیهی یک Run است. بدین ترتیب یک Run را به شکل زیر و به صورت دنباله‌ای از توصیف‌های لحظه‌ای تعریف می‌کنیم:

$$\begin{aligned}
 Run &= D^0 \vdash_{\text{fire}} D^1 \vdash_{\text{action}} D^2 \vdash_{\text{fire}} \dots \dots \dots D^{final} \\
 &= D^0 \vdash^* D^{final}
 \end{aligned}$$

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: $P = \prod p(A_m, \alpha_n) = \prod p(m, n) / \sum p(m, n) \in W_T$
- 18: **Until** $K < K_T$ or $P < P_T$
- 19: **End**

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

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

- 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 می‌شود. این مقدار Err نامیده می‌شود ($Err_{k+1} = W_{k+1} - WTH_k$)
- ۲- از مقدار Err برای به روزرسانی مقدار آستانه پویا در الگوریتم به وزن ($WTH_{k+1} = WTH_k + \alpha Err_{k+1}$)
- ۳- از یک متغیر نظری V به عنوان تخمینی از واریانس استفاده می‌کنیم. مقدار این متغیر در هر مرحله از طریق یک تخمین گر خطی به صورت $V_{k+1} = V_k + \beta(\text{abs}(Err_{k+1}) - V_k)$ محاسبه می‌شود. کاملاً واضح است که این تخمین گر، انحراف از میانگین را محاسبه می‌کند ($mdev$) نه انحراف معیار (sdev). اما می‌دانیم که

$$mdev^2 = (\sum |W - W_{TH}|)^2 \geq \sum |W - W_{TH}|^2 = \delta^2 = sdev^2 \quad (6)$$

رابطه (۶) نشان می‌دهد که انحراف از میانگین بزرگتر از انحراف معیار و لذا محافظه کارانه‌تر از آن است، ضمن این که محاسبه آن فوق العاده ساده‌تر از واریانس است.

۴- برای مقایسه و تعیین پاداش یا جریمه آناتماتاها از کران بالای $WTH_k / 2 + 2 \times V_k$ استفاده می‌کنیم

به این ترتیب الگوریتم ۱ شکل ۵ قادر به حل مساله یافتن زیرگراف بهینه تصادفی روی گراف‌های تصادفی است. کافی است که با توجه به

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

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

در الگوریتم ۱ شکل ۵، از مقداری موسوم به مقدار آستانه پویا (Threshold) برای ارزیابی اقدامات آناتماتا در eDLA استفاده می‌شود. (خط ۱۵) بر اساس قضیه حد مرکزی در [۴] روشی برای به روزرسانی 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 آناتماتاها ارائه شده است دیده می‌شود [۶، [۷، [۲۰]. بررسی‌ها نشان می‌دهد به دلیل نزدیک شدن 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 میانگین گیری 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.

اما، برای این که پیچیدگی محاسباتی این روش کاهش یابد، ناچاریم: اولاً از تخمین‌های ساده نظری تخمین‌های stochastic gradient استفاده کنیم. 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 اضافه‌ای در خصوص شبکه تصادفی پویا اعمال نشده است [۱۶] و سادگی محاسباتی آناتماتاها یادگیر نیز نقض نشده است.

تنها چیزی که قابل بیان است این است که $p_i(n)$ با یک احتمال مثبت به یک حالت دلخواه همگرا می‌شود. آنچه که اهمیت دارد، میزان عددی این احتمال است. برای حصول اطمینان از همگرایی، احتمال همگرایی به حالت مدنظر 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-I} فرآیند یادگیری را انجام می‌دهد، بایستی تابعی مثل $[p_i(\infty) = 1 | p_i(0) = p] = \text{pr}[p_i(\infty) = 1 | p_i(0) = p] = \Gamma_i(p)$ را برای کمی‌سازی احتمال همگرایی حالت i تعریف کرد. در [۲۱] نشان داده است تابع Γ بایستی معادله تابعی $\Gamma_i(p) = \Gamma_i(p) = \Gamma_i(p) = \Gamma_i(p)$ را ارضا کند (با شرایط مرزی مناسب برای $i=0$ و $i=p=1$) که در آن Γ عملگری است که به صورت $\Gamma = E[\Gamma_i(p_i(n+1)) | p_i(n) = p] = E[\Gamma_i(p_i(n+1)) | p_i(n) = p]$ تعریف می‌شود. حل این معادله تابعی کار ساده‌ای نیست.

به جای حل این معادله، از دوتابع، با شرایط مرزی مناسب برای یافتن حدود بالا و پایین تابع Γ استفاده می‌شود. این دو تابع که آنها را (φ_1, φ_2) مینامیم، بایستی برای تمام مقادیر $p \in [0, 1]$ دارای شرایط (۷) باشند

$$\begin{aligned} U\varphi_1(p) &\geq \varphi_1(p) \\ U\varphi_2(p) &\leq \varphi_2(p) \end{aligned} \quad (7)$$

برقراری نابرابری (۷) ایجاب می‌کند که:

$$\varphi_2(p) \leq \Gamma_i(p) \leq \varphi_1(p)$$

توابع (φ_1, φ_2) به ترتیب ابرمنظم (Super-regular) و زیرمنظم (Sub-regular) نامیده می‌شوند و یافتن آنها کار ساده‌تری است. نشان داده شده است [۲۱] که این توابع به شکل (۸) هستند

$$\varphi_i(p) = \frac{\exp(x_i p) - 1}{\exp(x_i) - 1} \quad i = 1, 2 \quad (8)$$

در ادامه این بحث، نمادگذاری‌های زیر را در نظر بگیرید:

فرض کنید تعداد تمام زیرگراف‌های موردنظر مساله r باشد. نیز فرض کنید در مرحله k ، زیرگراف s_i که یکی از جوابها است i بار انتخاب شده باشد (که $i = 1, 2, \dots, r$ است). علاوه بر آن $k = \sum_{i=1}^r k_i$. حال فرض کنید این زیرگراف s_i در مرحله k انتخاب شده باشد و نیز فرض کنید W_{s_i} وزن زیرگراف s_i باشد. مقدار وزن زیرگراف به دست آمده در ز این اجرای الگوریتم را با $(j) W_{s_i}$ نشان می‌دهیم. $(j = 1, 2, \dots, k_i)$. 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

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

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

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

مساله، قواعد مناسب اتخاذ شود. نمونه‌ای از این قواعد را برای دو مساله مورد بررسی این مقاله در جدول ۱ مشاهده می‌کنید.

جدول ۱: نحوه به کارگیری eDLA برای حل برخی مسائل بهینه سازی روی گراف تصادفی

مساله	درخت پوشای کمینه تصادفی	کوتاهترین مسیر تصادفی بین s و t
انتخاب ROOT	تصادفی: از میان آتاماتاها بر اساس توزیع یکنواخت قطعی: آتاماتایی با کمترین مقدار آنتروپی بردار	قطعی: مبدا
Fire شدن	تصادفی: از میان آتاماتاها Active بر اساس توزیع یکنواخت قطعی: آتاماتایی با کمترین مقدار آنتروپی بردار	قطعی: اقدام انجام شده توسط آتاماتای Fire فعلی تبیین کننده Fire بعدی است
Active شدن	بر اساس مجاورت با آتاماتای fire فعلی	بر اساس مجاورت با آتاماتای fire فعلی
خاتمه	شدن تمام آتاماتاها	شدن آتاماتای Fire Off

۴- تحلیل رفتار الگوریتم پیشنهادی

در این قسمت به بررسی ریاضی رفتار eDLA در حل مساله 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 است که زمانی که از الگوریتم یادگیری L_{R-P} استفاده می‌شود، (p) بردار احتمال انتخاب اقدام‌های آتاماتا، 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 موردنیاز کوچک کرد. در این نوع همگرایی، مقادیر اولیه احتمالات تاثیری در رفتار مجانبی الگوریتم ندارند. [۱]

حال دوم همگرایی برای الگوریتم‌های یادگیری ϵ -optimal می‌باشد. در این حال، 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. اما در روش دوم، زنجیره مارکوف تولید شده بیش از یک حالت جاذب دارد. برخلاف روش اول، مقادیر اولیه بردار احتمالات نقش موثری در همگرایی روش دوم دارند. به این ترتیب، در مورد الگوریتم‌های یادگیری ϵ -optimal مثل L_{R-I}

(k) c_i 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. در این حالت برای سایر زیرگراف‌ها نظیر s_j دو حالت قابل تصویر 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 ($k+1$) $q_j(k) < q_i(k)$

(ب) s_j تعدادی یال مشترک با s_i دارد. 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 مساله هستند

$$s_1 = \{(1,2), (2,5)\}$$

$$s_2 = \{(1,2), (2,4), (4,5)\}$$

$$s_3 = \{(1,2), (2,3), (3,5)\}$$

$$s_4 = \{(1,2), (2,3), (3,4), (4,5)\}$$

$$s_5 = \{(1,5)\}$$

$$s_6 = \{(1,4), (4,5)\}$$

$$s_7 = \{(1,3), (3,5)\}$$

$$s_8 = \{(1,3), (3,4), (4,5)\}$$

ضمناً فرض کنید s_4 از راضاکننده محدودیت‌های موردنیاز مساله روی گراف تصادفی باشد. برای هریک از این زیرگراف‌ها، مقدار (k) q متناظر باز است:

$$\begin{aligned} q_1(k) &= p_2^1(k)p_5^2(k) & q_2(k) &= p_2^1(k)p_4^2(k)p_5^4(k) \\ q_3(k) &= p_2^1(k)p_3^2(k)p_5^3(k) & q_4(k) &= p_2^1(k)p_3^2(k)p_4^3(k)p_5^4(k) \\ q_5(k) &= p_5^1(k) & q_6(k) &= p_4^1(k)p_5^4(k) \\ q_7(k) &= p_3^1(k)p_5^3(k) & q_8(k) &= p_3^1(k)p_4^3(k)p_5^4(k) \end{aligned}$$

در این روابط ($p_j^i(k)$) احتمال انتخاب یال (i, j) یا همان انتخاب اقدام j توسط آتاماتای fire شده s_i در مرحله k است.

از آنجا که ۸ جواب بیشتر متصوّر نیست، بسته به اینکه در مرحله $k+1$ کدام یک از زیرگراف‌های s_1 تا s_8 انتخاب شده باشد، تغییرات حاصل در (k) $q_4(k)$ مطابق با مجموع مقادیر در

است

منظور از ($p_j^i(k \downarrow)$) کاهش در مقدار $p_j^i(k)$ و منظور از ($p_j^i(k \uparrow)$) افزایش در مقدار $p_j^i(k)$ از طریق روابط L_{R-I} است

$$p_j^i(k \downarrow) = p_j^i(k) * (1-a)$$

$$p_j^i(k \uparrow) = p_j^i(k) * (1-a) + a$$

با جمع مقادیر حاصله در

خواهیم داشت:

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

در این رابطه k_i تعداد دفعاتی که زیرگراف s_i انتخاب شده است را نشان می‌دهد و مقدار (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 حاصلضرب 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 (k) احتمال متناظر با زیرگراف بهینه s_i در گراف تصادفی مفروض در مرحله k ام را نشان دهد. اگر بردار $\underline{q}(k) = [q_i(k)]_{i=1 \dots r}$ مطابق با الگوریتم شکل ۵ تغییر کند، در این

$$\text{صورت } \lim_{n \rightarrow \infty} q_i(n) = 1^{a.s.}$$

اثبات: اثبات این قضیه در چند مرحله صورت می‌گیرد

лем ۱: اگر در الگوریتم ارائه شده شکل ۵، زیرگراف s_i با احتمال (k) $c_i(k)$ در مرحله k جریمه شود، و c^* باشد در این

صورت برای هر $\epsilon \in (0, 1)$ و $k > K(\epsilon)$ داریم

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

اثبات: مطابق با نمادگذاری قبلی (k) احتمال جریمه شدن زیرگراف s_i را در گام k ام نشان می‌دهد. مطابق با قانون قوی اعدا (k) در مرحله k برای مقادیر بزرگ k باشد آنگاه

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

لم ۲: اگر (k) $d_i(k)$ و (k) $c_i(k)$ احتمالات مربوط به جریمه شدن و تنبیه شدن زیرگراف s_i حاصل شده در k امین مرحله را نشان دهدند ($d_i(k)$ و $c_i(k)$ بر اساس الگوریتم ارائه شده شکل ۵ تغییر کند، در این صورت امید ریاضی شرطی $E[q_i(k+1) | \underline{q}(k)]$ مطابق با روابط زیر تعریف می‌شود:

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

که در (۹):

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

اثبات:

الگوریتم پیشنهادی (الگوریتم ۱ شکل ۵) از روش یادگیری L_{R-I} برای پاداش یا جریمه آتاماتها استفاده می‌کند، لذا در مرحله k ام اجرای الگوریتم، احتمال انتخاب زیرگراف s_i (یعنی (k) $q_i(k)$) با احتمال

منجر به انتخاب زیرگراف s_j گردیده است.

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

اما

$$E[q_i(k+1)|\underline{q}(k), \alpha_j] = (1-d_j) q_i(k)$$

$$+ d_j \prod_{e(m,n) \in s_i \cap s_j} (p_n^m(k) \uparrow) \prod_{e(m,n) \in s_i - s_j} (p_n^m(k) \downarrow)$$

در این روابط $p_j^i(k)$ احتمال انتخاب یال (i, j) یاهمان انتخاب اقدام a_j توسط آتماتای fire شده A_i در مرحله k است. منظور از کاهش در مقدار $p_j^i(k) \uparrow$ و منظور از $p_j^i(k) \downarrow$ افزایش در

مقدار p_j^i از طریق روابط L_{R-I} است

$$p_j^i(k) \downarrow = p_j^i(k) \times (1-a)$$

$$p_j^i(k) \uparrow = p_j^i(k) \times (1-a) + a$$

به این ترتیب لم ۲ اثبات می‌گردد ■

لم ۳: اگر زیرگراف بهینه s_i جواب مساله یافتن زیرگراف بهینه تصادفی باشد، در این صورت مقدار $E[q_i(k+1) - q_i(k)|\underline{q}(k)]$ همواره ثابت است.

اثبات:

الگوریتم یافتن زیرگراف بهینه تشریح شده در شکل ۵ همواره حداقل $n-1$ یال را از میان یال‌های گراف انتخاب می‌کند تا یکی از جواب ممکن را بسازد. به این ترتیب متوسط مقدار $(q_i(k))$ برای زیرگراف s_i که از ضرب مقادیر اقدام‌های متناظر با یال‌های تشکیل دهنده s_i تشکیل می‌شود برابر است با

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

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

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

مطابق با (۱۲)، اولاً $p_s^t(k)$ و $p_u^t(k)$ مقادیری مثبت در بازه

$$\begin{aligned} E[q_4(k+1)|\underline{q}(k)] &= \\ q_1(k) \times [c_1(k)q_4(k) + d_1(k)p_2^1(k \uparrow)p_3^2(k \downarrow)p_4^3(k)p_5^4(k)] &+ \\ q_2(k) \times [c_2(k)q_4(k) + d_2(k)p_2^1(k \uparrow)p_3^2(k \downarrow)p_4^3(k)p_5^4(k)] &+ \\ q_3(k) \times [c_3(k)q_4(k) + d_3(k)p_2^1(k \uparrow)p_3^2(k \uparrow)p_4^3(k \downarrow)p_5^4(k)] &+ \\ q_4(k) \times [c_4(k)q_4(k) + d_4(k)p_2^1(k \uparrow)p_3^2(k \uparrow)p_4^3(k \uparrow)p_5^4(k)] &+ \\ q_5(k) \times [c_5(k)q_4(k) + d_5(k)p_2^1(k \downarrow)p_3^2(k)p_4^3(k)p_5^4(k)] &+ \\ q_6(k) \times [c_6(k)q_4(k) + d_6(k)p_2^1(k \downarrow)p_3^2(k)p_4^3(k)p_5^4(k)] &+ \\ q_7(k) \times [c_7(k)q_4(k) + d_7(k)p_2^1(k \downarrow)p_3^2(k)p_4^3(k)p_5^4(k)] &+ \\ q_8(k) \times [c_8(k)q_4(k) + d_8(k)p_2^1(k \downarrow)p_3^2(k)p_4^3(k)p_5^4(k)] &= \\ \sum_{j=1}^8 q_j(k) [c_j(k)q_4(k) + d_j(k) \prod_{e(m,n) \in s_4} \delta_n^m(k)] \end{aligned}$$

جدول ۳: تغییرات در احتمال انتخاب زیرگراف بهینه در یک گراف نمونه

احتمال انتخاب	تغییرات
s_1	$c_1(k)q_4(k)$ $d_1(k)p_2^1(k \uparrow)p_3^2(k \downarrow)p_4^3(k)p_5^4(k)$
s_2	$c_2(k)q_4(k)$ $d_2(k)p_2^1(k \uparrow)p_3^2(k \downarrow)p_4^3(k)p_5^4(k)$
s_3	$c_3(k)q_4(k)$ $d_3(k)p_2^1(k \uparrow)p_3^2(k \uparrow)p_4^3(k \downarrow)p_5^4(k)$
s_4	$c_4(k)q_4(k)$ $d_4(k)p_2^1(k \uparrow)p_3^2(k \uparrow)p_4^3(k \uparrow)p_5^4(k)$
s_5	$c_5(k)q_4(k)$ $d_5(k)p_2^1(k \downarrow)p_3^2(k)p_4^3(k)p_5^4(k)$
s_6	$c_6(k)q_4(k)$ $d_6(k)p_2^1(k \downarrow)p_3^2(k)p_4^3(k)p_5^4(k)$
s_7	$c_7(k)q_4(k)$ $d_7(k)p_2^1(k \downarrow)p_3^2(k)p_4^3(k)p_5^4(k)$
s_8	$c_8(k)q_4(k)$ $d_8(k)p_2^1(k \downarrow)p_3^2(k)p_4^3(k)p_5^4(k)$

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:

$$\begin{aligned} E[q_i(k+1)|\underline{q}(k)] &= \\ \sum_{j=1}^r E[q_i(k+1)|\underline{q}(k), \alpha_j] \cdot p(\alpha_j | \underline{q}(k)) & \quad (14) \end{aligned}$$

در رابطه (۱۴) منظور از α_j مجموعه اقدام‌هایی از eDLA است که

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

که در آن $x > 0$ بایستی انتخاب شود. واضح است که تابع تعریف شده در رابطه (۱۴) شرایط مرزی تعریف شده در (۱۳) را ارضاء 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 $\theta_i[x, \underline{q}] = \exp(-xq_i/a)$ وقتی زیرمنظم است که ابرمنظم باشد.

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

اثبات: برای تعیین شرایطی که تحت آن شرایط، تابع $\theta_i[x, \underline{q}]$ یک ابرمنظم است، از تعریف ابرمنظم و محاسبه مقدار امید ریاضی استفاده می‌کنیم. فرض کنید زیرگراف s_i جواب بهینه باشد، در این صورت $U\theta_i[x, \underline{q}] = E[\exp(-xq_i(k+1)/a) | q(k) = \underline{q}]$ برای محاسبه مقدار امید ریاضی در رابطه (۱۵) به این ترتیب عمل می‌کنیم:

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

با ساده‌سازی رابطه (۱۶) به روابط (۱۷) و (۱۸) می‌رسیم:

$$\begin{aligned} & \sum_{j=1}^r \exp(-xq_i(k+1)/a) q_j d_j \\ &= q_i d_i \exp\left(-x \prod_{e(m,n) \in s_i} \{p_n^m + a(1-p_n^m)\} / a\right) \\ &+ \sum_{j=1, j \neq i}^r q_j d_j \exp\left(-x \prod_{e(m,n) \in s_i \cap s_j} \{p_n^m + a(1-p_n^m)\} / a\right) \end{aligned} \quad (17)$$

$$\begin{aligned} & \sum_{j=1}^r \exp(-xq_i(k)/a) \times q_j \times (1-d_j) \\ &= \theta_i[x, \underline{q}] \sum_{j=1}^r q_j \times (1-d_j) \\ &= \theta_i[x, q] - \theta_i[x, \underline{q}] \sum_{j=1}^r q_j \times d_j \end{aligned} \quad (18)$$

در رابطه (۱۷) داریم:

(۰,۱) هستند. ثانیا از آن جا که $e(t, u) \in s_i$ یالی است که متعلق به زیرگراف بهینه است، توسط آنماتاتی A_t انتخاب می‌شود. اما از لم ۲ می‌دانیم که برای مقادیر بزرگ k داریم:

$$c_s^t(k) - c_u^t(k) > 0 \quad \forall s \neq u$$

بنابراین برای مقادیر بزرگ k طرف راست معادله (۱۲) مقداری مثبت است و در نتیجه $\Delta q_i(k) \geq 0$

$$\text{گزاره: } \lim_{k \rightarrow \infty} q_i(k) = q_i^* \in \{0, 1\} \text{ با احتمال ۱ وجود دارد.}$$

لم ۳ بیان کننده 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. در این صورت 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 ناصرف به ازای تمام مقادیر k داریم $q_i(k+1) = q_i(k)$. بنابراین الزاماً

$$\blacksquare q_i^* \in \{0, 1\}$$

گزاره قبلی نشان می‌دهد که الگوریتم ممکن است به جواب بهینه همگرا نشود. سعی میکنیم، حدودی برای تابع احتمال انتخاب زیرگراف بهینه ارائه دهیم. برای این کار از روش [۲۱] استفاده می‌کنیم.

فرض کنید مجموعه بردارهای یکه $V_r = \{e_1, e_2, \dots, e_r\}$ مجموعه

حالتهای جاذب فرآیند $\{\underline{q}(k)\}$ باشد و $\underline{q}^* \in V_r$ حالتی را نشان دهد

که $\underline{q}(k)$ به آن همگرا می‌شود. تعریف می‌کنیم

$$\Gamma_i[\underline{q}] = \Pr[q^* = e_i | q(0) = \underline{q}]$$

فرض کنید $C(S_r) : S_r \rightarrow R$ فضای حالت تمام توابع مشتق‌پذیر

پیوسته حقیقی مقدار با مشتق محدود تعریف شده روی S_r باشد که R یک خط حقیقی است. اگر $g[\underline{q}] \in C(S_r)$ باشد، یک الگوریتم یادگیری

نظری یک عملگر U با میانگین $Ug[\underline{q}] = g(k+1) - g(k) = \underline{q}$ عمل می‌کند که در این رابطه U نشان‌دهنده امید ریاضی است. نشان داده شده است که U عملگری خطی است و توابع نامنفی را نامنفی نگه دارد. یعنی

$$\text{if } g[\underline{q}] \geq 0 \quad Ug[\underline{q}] \geq 0 \quad \forall \underline{q} \in S_r$$

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

نشان داده شده است که $\Gamma_i[\underline{q}] = \Gamma_i[q]$ تنها جواب پیوسته معادله $U\Gamma_i[\underline{q}] = \Gamma_i[\underline{q}]$ 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 $U\Gamma_i[\underline{q}] = \Gamma_i[\underline{q}]$ برای تعیین احتمال انتخاب زیرگراف بهینه بر حسب احتمال اولیه، کار سختی است. برای این کار از گزاره قبلی می‌دانیم که 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

$$\begin{aligned}
& U \theta_i[x, \underline{q}] - \theta_i[x, \underline{q}] \\
& \leq q_i d_i \left(\exp \left\{ -\frac{x}{a} (q_i + a(1-q_i)) \right\} - \theta_i[x, \underline{q}] \right) \\
& + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j \left(\exp \left\{ -\frac{x}{a} q_j (1-a) \right\} - \theta_i[x, \underline{q}] \right) \\
& = \theta_i[x, \underline{q}] \left(q_i d_i \exp \{-x(1-q_i)\} - 1 \right) \\
& + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j (\exp \{xq_i\} - 1) \\
& = -xq_i \theta_i[x, \underline{q}] \left((1-q_i)d_i \frac{\exp \{-x(1-q_i)\} - 1}{-x(1-q_i)} \right. \\
& \quad \left. - \sum_{\substack{j=1 \\ j \neq i}}^r q_j \times d_j \frac{\exp \{xq_i\} - 1}{xq_i} \right) \tag{۲۳}
\end{aligned}$$

تعريف می کنیم:

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

$$\begin{aligned}
& U \theta_i[x, \underline{q}] - \theta_i[x, \underline{q}] \leq \\
& -xq_i \theta_i[x, \underline{q}] \left((1-q_i)d_i V[-x(1-q_i)] \right. \\
& \quad \left. - \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j V[xq_i] \right) \tag{۲۴}
\end{aligned}$$

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

$$(1-q_i)d_i V[-x(1-q_i)] - \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j V[xq_i] \geq 0 \quad \text{لذا:}$$

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

$$\frac{V[-x(1-q_i)]}{V[xq_i]} \geq \frac{\sum_{j=1, j \neq i}^r q_j d_j}{(1-q_i)d_i} \tag{۲۵} \quad \text{در (۲۵) داریم:}$$

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

$$\min_{j \neq i} \left(\frac{d_j}{d_i} \right) \leq \frac{\sum_{j=1, j \neq i}^r q_j \times \frac{d_j}{d_i}}{\sum_{j=1, j \neq i}^r q_j} \leq \max_{j \neq i} \left(\frac{d_j}{d_i} \right) \tag{۲۶}$$

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

$$\begin{aligned}
& \prod_{e(m,n) \in s_i - s_j} \{p_n^m(1-a)\} = \\
& \frac{\prod_{e(m,n) \in s_i - s_j} \{p_n^m(1-a)\} \prod_{e(m,n) \in s_i \cap s_j} \{p_n^m(1-a)\}}{\prod_{e(m,n) \in s_i \cap s_j} \{p_n^m(1-a)\}} \tag{۱۹} \\
& = \frac{\prod_{e(m,n) \in s_i} \{p_n^m(1-a)\}}{\prod_{e(m,n) \in s_i \cap s_j} \{p_n^m(1-a)\}}
\end{aligned}$$

با توجه به اینکه $q_i = \prod_{e(m,n) \in \tau_i} p_n^m$ و با توجه به روابط (۱۶) و (۱۷) و (۱۸) خواهیم داشت

$$\begin{aligned}
U \theta_i[x, \underline{q}] &= q_i d_i \times \exp \left\{ -\frac{x}{a} (q_i + a(1-q_i)) \right\} \\
& + \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j \exp \left\{ -\frac{xq_j(1-a)}{a} D(p_n^m) \right\} \tag{۲۰} \\
& + \theta_i[x, \underline{q}] - \theta_i[x, \underline{q}] \sum_{j=1}^r q_j \times d_j
\end{aligned}$$

اما روشن است که:

$$D(p_n^m) = \frac{\prod_{\substack{e(m,n) \in s_i \cap s_j \\ j \neq i}} \{p_n^m + a(1-p_n^m)\}}{\prod_{e(m,n) \in s_i \cap s_j} \{p_n^m(1-a)\}} \geq 1$$

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

$$\begin{aligned}
& \exp \left\{ -\frac{x}{a} q_i (1-a) \frac{\prod_{\substack{e(m,n) \in s_i \cap s_j \\ j \neq i}} \{p_n^m + a(1-p_n^m)\}}{\prod_{e(m,n) \in s_i \cap s_j} \{p_n^m(1-a)\}} \right\} \tag{۲۱} \\
& \leq \exp \left\{ -\frac{x}{a} q_i (1-a) \right\}
\end{aligned}$$

از روابط (۲۰) و (۲۱) به رابطه (۲۲) می رسیم:

$$\begin{aligned}
U \theta_i[x, \underline{q}] - \theta_i[x, \underline{q}] &\leq \\
q_i d_i \times \exp \left\{ -\frac{x}{a} (q_i + a(1-q_i)) \right\} & \tag{۲۲} \\
+ \sum_{\substack{j=1 \\ j \neq i}}^r q_j d_j \times \exp \left\{ -\frac{x}{a} q_j (1-a) \right\} - \theta_i[x, \underline{q}] \sum_{j=1}^r q_j d_j &
\end{aligned}$$

اما

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

فلذا:

یال i برابر است با $\pm z_{\alpha/2} \times \delta_i / \sqrt{n_i}$ که در این رابطه n_i تعداد نمونه‌های لازم از یال i است لذا:

$$z_{\alpha/2} \times \delta_i / \sqrt{n_i} = t_{\alpha/2,k-1} \times S / ((|V|-1)\sqrt{k}) \quad (30)$$

و با حل معادله (۳۰) خواهیم داشت:

$$n_i = (z_{\alpha/2} / t_{\alpha/2,k-1})^2 \times (k(|V|-1)^2 / s^2) \times \delta_i^2 \quad (31)$$

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

$$n = \sum_{i=1}^{|E|} n_i = (z_{\alpha/2} / t_{\alpha/2,k-1})^2 \times (k(|V|-1)^2 / s^2) \times \sum_{i=1}^{|E|} \delta_i^2 \quad (32)$$

قضیه ۲: اگر n تعداد نمونه‌های موردنیاز در روش نمونه‌گیری استاندارد و m تعداد نمونه‌های موردنیاز 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

$$\frac{n}{m} \geq |E| \times \min_i \delta_i^2 / \max_i \delta_i^2$$

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 دقیقی معادل با روش eDLA از رابطه (۳۲) به دست می‌آید

اما هر نمونه از زیرگراف ایجاد شده توسط eDLA حداکثر شامل $|V|-1$ نمونه‌گیری از یال‌های گراف تصادفی است و اگر m تعداد کل نمونه‌های گرفته شده از یال‌ها در روش مبتنی بر eDLA باشد

$$m \leq k * (|V|-1)$$

ولذا:

$$n/m \geq (z_{\alpha/2} / t_{\alpha/2,k-1})^2 \times (|V|-1) / s^2 \times \sum_{i=1}^{|E|} \delta_i^2 \quad (33)$$

با فرض $1-\alpha \geq 90\%$ به عنوان بازه اطمینان مشترک و $k \geq 120$ و با

$$\text{فرض این که } |V|-1 \geq 5 \quad \text{داریم:}$$

$$t_{\alpha/2,k-1} = 1.645 \approx z_{\alpha/2}$$

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

$$n/m \geq (|V|-1) \times \sum_{i=1}^{|E|} \delta_i^2 / s^2 \quad (34)$$

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 نظری یال i را با متغیر تصادفی Y_i نشان دهیم داریم

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

در رابطه (۳۵) M نشان‌دهنده زیرگراف تشکیل شده در هر مرحله توسط eDLA است. از رابطه (۳۵) و بر اساس قضایای محاسبه واریانس

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

تعريف می‌کنیم $V[x] = \ln V[x]$. با توجه به محدب بودن $H[x]$ می‌توان نشان داد $H[x] \leq V[-x]$.

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

از آنجا که $V[-x]$ تابعی پیوسته و یکنوا کاهاشی با $V[0]=1$ است $x \in (0, x^*)$ ای وجود دارد که برای تمام مقادیر در بازه $[0, x^*]$ داریم $x = x^*$. با انتخاب $x = x^*$ نابرابری

(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 $\epsilon > 0$ مفروض، ثابت مثبت $a^* < a$ به گونه‌ای موجود است که نابرابری $\Phi_i[x, q] \leq \Gamma_i[q] - \epsilon$ برقرار است. این به آن معناست

که $\lim_{n \rightarrow \infty} q_i(n) = 1$ a.s. به این ترتیب اثبات قضیه ۱ کامل می‌شود. ■

лем ۵: اگر مجموعه زیرگراف‌های حاصل از به کارگیری eDLA برای حل مساله 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 اینکه

$$S^2 = \left(\sum_{i=1}^k w_i^2 - (\sum_{i=1}^k w_i)^2 / k \right) / (k-1)$$

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

$$n = (z_{\alpha/2} / t_{\alpha/2,k-1})^2 \times (k(|V|-1)^2 / S^2) \times \sum_{i=1}^{|E|} \delta_i^2 \quad (29)$$

در (۲۹) K تعداد نمونه‌های حاصله از الگوریتم ۱ مبتنی بر eDLA

$$\delta_i^2 \text{ واریانس یال } i \text{ و } t \text{ توزیع‌های نرمال و } t\text{-student}$$

اثبات: با توجه به نامشخص بودن 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][۲۳]. در این صورت حدود اطمینان بازه $100(1-\alpha)\%$ مبتنی بر این توزیع برابر است با

$$\pm t_{\alpha/2,k-1} \times S / \sqrt{k}$$

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 $\pm t_{\alpha/2,k-1} \times S / ((|V|-1)\sqrt{k})$ باشد در

این صورت مطابق حدود اطمینان $\pm t_{\alpha/2,k-1} \times S / \sqrt{k}$ برای زیرگراف‌ها با درصد $(1-\alpha) \times 100\%$ تضمین شده است.

اما حدود اطمینان بازه $100(1-\alpha)\%$ برای هر یک از یال‌ها مثل

مورد استفاده توسط آنماتاها الگوریتم خطی L_{R-I} بوده است. این کار ۱۰۰ بار تکرار شده و متوسط احتمال انتخاب مسیر بهینه محاسبه شده است. 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 (۱۰۰۰ 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 آنماتاها، الگوریتم خطی L_{R-I} بوده است.

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

شکل ۶ و شکل ۷ صحت قضیه ۱ را نشان می‌دهند. علاوه بر این نشان می‌دهند روش فعلی ارائه شده برخلاف روش‌های قبلی برای حل این مسائل در [۴] و [۲۰] تقریباً مستقل از نرخ یادگیری، سرعت همگرایی بالاتری دارد.

رابطه (۳۶) نتیجه می‌شود

$$s^2 = \text{Var}(X) = \sum_i \text{Var}(Y_i) \quad (36)$$

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

$$\text{اما، داریم: } s^2 \leq (|V| - 1) \times \max_i \delta_i^2 \text{ و } \sum_{i=1}^{|E|} \delta_i^2 \geq |E| \times \min_i \delta_i^2 \quad \text{ولذا}$$

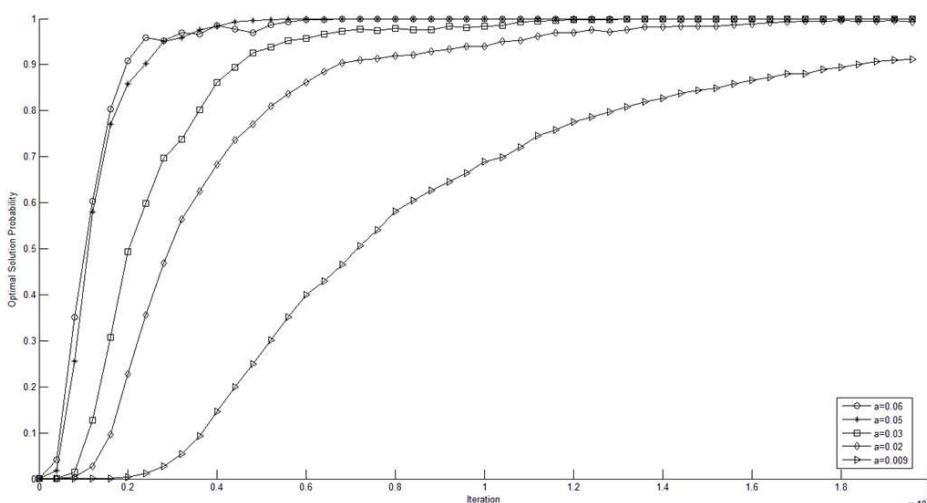
$$\frac{\sum_{i=1}^{|E|} \delta_i^2}{s^2} \geq \frac{|E|}{|V| - 1} \times \frac{\min_i \delta_i^2}{\max_i \delta_i^2} \quad (37)$$

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

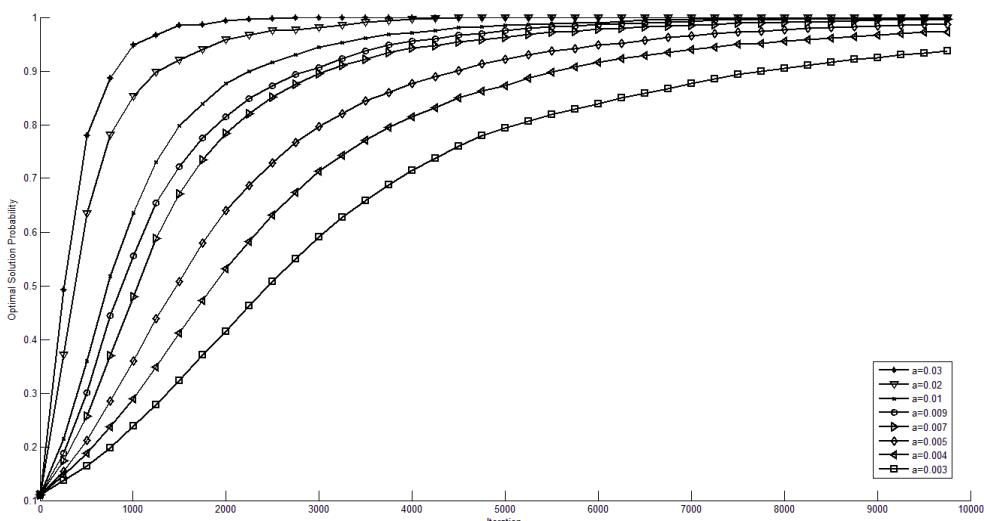
$$\frac{n}{m} \geq |E| \times \frac{\min_i \delta_i^2}{\max_i \delta_i^2} \quad (38)$$

۵- بررسی تجربی:

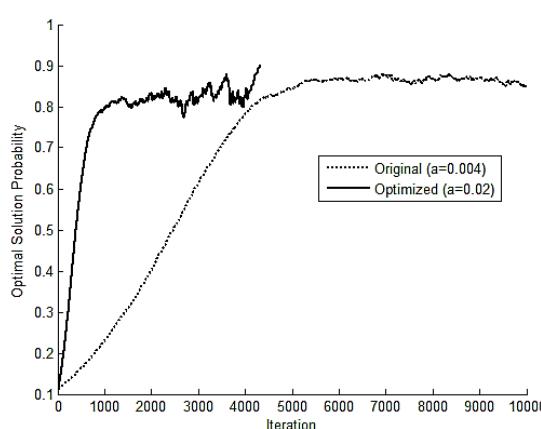
برای بررسی قضیه ۱ در روش پیشنهادی، بر اساس جدول ۱ و الگوریتم ۱ ارائه شده در این مقاله، مسائل یافتن کوتاهترین مسیر تصادفی و درخت پوشای کمینه تصادفی را به ترتیب در گراف‌های Net4 ([۴]) و alex1-b ([۲۴]) مورد بررسی قرار داده‌ایم. هدف از این آزمایش، بررسی نحوه همگرایی الگوریتم مبتنی بر eDLA به جواب بهینه بوده است. در آزمایش اول، از eDLA برای یافتن درخت پوشای کمینه استفاده شده است. به ازای هر نرخ یادگیری، الگوریتم دقیقاً ۲۰۰۰۰ دور اجرا شده است. به ازای هر نرخ یادگیری، الگوریتم دقیقاً ۲۰۰۰۰ دور اجرا شده است (۲۰۰۰۰ درخت پوشای انتخاب شده است) الگوریتم یادگیری



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



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



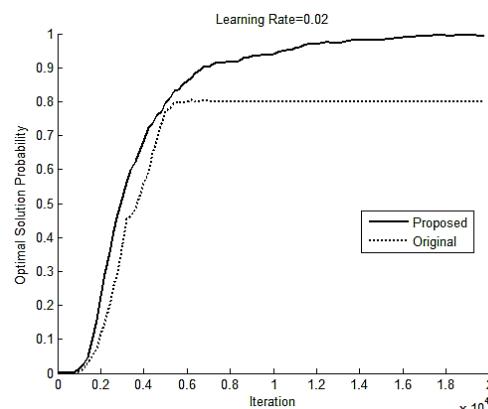
شکل ۸: مقایسه سرعت همگرایی الگوریتم پیشنهادی مقاله (Optimal) در برابر الگوریتم مبتنی بر DLA (Original) [۴] (درحال حداقل های لازم برای همگرایی (%) ۱۰۰)

۶ - نتیجه گیری:

در این مقاله، چارچوبی جدید مبتنی بر آناتماتاهاز یادگیر موسوم به eDLA برای حل مساله یافتن زیرگراف بهینه در گرافهای تصادفی ارائه شد. نشان داده شد که این ساختار می‌تواند از طریق تعريف سطوح مختلف فعال بودن، محدودیت DLA در نحوه فعال سازی آناتماتاها را مرتفع کرده و از این شبکه از آناتماتاها به عنوان چارچوبی یک دست برای حل مسائل مختلف بهینه سازی روی گرافهای 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 مسائل بهینه سازی جایگشت تصادفی در دست بررسی است.

مقایسه الگوریتم پیشنهادی برای حل مسائل درخت پوشای کمینه تصادفی و کوتاهترین 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 کوتاهترین مسیر تصادفی به دلیل اینکه DLA و eDLA یکسان عمل می کنند، بهبود نتایج قطعاً ناشی از معیار مقایسه ای جدید است (شکل ۹)



شکل ۹: مقایسه تغییرات احتمال درخت پوشای بهینه در الگوریتم پیشنهادی (Proposed) و الگوریتم ارائه شده در [۲۵] (Original) در نرخ یادگیری ۰،۰۲ در گراف alex1-b

- منابع و مراجع:

- [1] K. S. Narendra and M. A. L. Thathachar, "Learning Automata: A Survey," *IEEE Transactions Syst. man, cybernetics*, vol. SMC-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. 711–22, 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 *10th iranian conference on Electrical Engineering*, 2002, vol. I, no. May.
- [6] M. R. MollakhaliliMeybodi and M.R.Meybodi, "A New Distributed Learning Automata Based Algorithm for Solving Stochastic Shortest Path," in *6th Conference on Intelligent Systems*, 2004.
- [7] H. Beigy and M. R. Meybodi, "Solving Stochastic Shoretsi Path Problem Using Monte Carlo Sampling Method: A Distributed Learning Automata Approach," *Springer-Verlag Lect. Notes Adv. Soft Comput. Neural Networks Soft Comput.*, pp. 626–632, 2003.
- [8] A. Motevalian and M. R. Meybodi, "Solving Maximal Independent Set Problem Using Distributed Learning Automata," in *14th Iranian Electrical Engineering Conference(ICEE2006)*, 2006, vol. 1.
- [9] A.Alipour and M.R.Meybodi, "Solving Traveling Salesman Problem Using Distributed Learning Automata," in *0th Annual CSI Computer Conference*, 2005, pp. 759–761.
- [10] S. Saati and M. Meybodi, "A Self Organizing Model for Document Structure Using Distributed Learning Automata," in *Second International Conference on Information and Knowledge Technology (IKT2005)*, 2005.
- [11] Z. Anari, M. R. Meybodi, and B. Anari, "Web page ranking based on fuzzy and learning automata," *Proc. Int. Conf. Manag. Emergent Digit. Ecosyst. - MEDES '09*, p. 162, 2009.
- [12] M. R. MollakhaliliMeybodi and M. R. Meybodi, "Link Prediction in Adaptive Web Sites Using Distributed Learning Automata," in *13th Annual CSI Computer Conference of Iran*, 2008.
- [13] M. R. MollakhaliliMeybodi and M. R. Meybodi, "A Distributed Learning Automata Based Approach for User Modeling in Adaptive Hypermedia," in *Congress on Electrical, Computer and Information Technology*, 2012.
- [14] A. BaradaranHashemi and M. R. Meybodi, "Web Usage Mining Using Distributed Learning Automata," in *12th Annual CSI Computer Conference of Iran*, 2007, pp. 553–560.
- [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. 1179–92, Dec. 2005.
- [16] K. Liu, "Stochastic Online Learning for Network Optimization under Random Unknown Weights," pp. 1–18.
- [17] N. K. S. and T. M. L, *Learning Automata: an Introduction*. Prentice Hall, 1989, p. 476.
- [18] S. Lakshminarayanan and M. A. L. Thathachar, "Absolutely Expedient Learning Algorithms for Stochastic Automata," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-6, pp. 281–286, 1973.
- [19] B. Thathachar , MAL and Harita, "Learning automata with changing number of actions," *IEEE Trans. Syst. Man. Cybern. - Part A Syst. Humans*, vol. 17, no. 6, pp. 1095–1100, 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, 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*, Third. 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 '," pp. 1–12.
- [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.