

حل مساله کوتاهترین مسیر در گرافهای تصادفی در صورت همبستگی بین هزینه یالها با استفاده از بازی بین اتوماتاهای یادگیر

اصغر قربانی محمدرضا میبیدی

آزمایشگاه سیستمهای نرم افزاری
دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیرکبیر
تهران ایران
(aghorbani@aut.ac.ir , mmeybodi@aut.ac.ir)

چکیده

در اکثر مطالعاتی که در زمینه پیدا کردن کوتاهترین مسیر در گرافهای تصادفی انجام شده است فرض می‌شود که هزینه یالها مستقل از همدیگر هستند که این فرض در بسیاری از موارد فرض صحیحی نمی‌باشد چرا که ممکن است تغییر ترافیک در یک قسمت از شبکه ناشی از تغییر ترافیک در قسمتهای مجاور آن باشد. مساله پیدا کردن کوتاهترین مسیر احتمالی با یالهای همبسته در شرایطی که توزیعهای احتمالی وزن یالها از قبل مشخص است برای اولین بار توسط بارتون^۱ و پس از آن توسط والر^۲ و زیلیاسکوپولس^۳ و فن^۴ مورد بررسی قرار گرفت و الگوریتمهایی جهت حل آن پیشنهاد گردید. در این مقاله برای اولین بار الگوریتمی برای حل مساله کوتاهترین مسیر گرافهای تصادفی در شرایطی که همبستگی مابین هزینه یالها وجود دارد^۵ (SSPCL) و همچنین توزیعهای احتمالی وزن یالها از قبل شناخته شده نیست پیشنهاد میگردد. در الگوریتم پیشنهادی از بازی بین اتوماتاهای یادگیر برای پیدا کردن کوتاهترین مسیر بین یک گره و دیگر گره های گراف استفاده میشود. الگوریتم پیشنهادی سعی میکند با حداقل تعداد نمونه گیری از یالهای گراف تصادفی درخت کوتاهترین مسیر را برای یک گره ریشه مشخص پیدا نماید.

کلمات کلیدی: کوتاهترین مسیر، گرافهای تصادفی، اتوماتاهای یادگیر، بازی اتوماتاهای یادگیر، همبستگی

۱. مقدمه

برای حل مساله کوتاهترین مسیر در گرافهای قطعی، که وزن یالها در آنها مقادیر مستقل، ثابت و از پیش تعیین شده ای هستند، الگوریتم های متفاوتی با زمان چندجمله ای، نظیر دایجسترا و فلویدوارشال وجود دارد. اما این الگوریتمها در صورتی که وزن یالها به صورت پویا تغییر کنند قادر به یافتن راه حل بهینه نمی‌باشند زیرا با کوچکترین تغییری در گراف، کل گراف مجدداً بایستی بررسی شود. بدین جهت الگوریتم هایی برای مسائل کوتاهترین مسیر در گرافهای پویا^۶، گرافهایی که وزن یالها و یا ساختار گراف در طی زمان بطور پویا تغییر کنند، مطرح شده اند [21][22][13][14]. این الگوریتمها نیز برای گرافهایی تصادفی، گراف هایی که وزن یالها متغیرهای تصادفی باشند، از کارایی بالایی برخوردار نیستند. بهمین دلیل برای حل مساله کوتاهترین مسیر در گرافهایی تصادفی الگوریتمهایی طراحی شده اند. فرانک^۷ [3] نخستین کسی بود که مطالعه شبکه های تصادفی را آغاز کرد. مساله طرح شده توسط فرانک در این رابطه، تعیین توزیع احتمال طول کوتاهترین مسیر میان گره مبدأ و مقصد مشخص شده در یک شبکه بود؛ با این فرض که طول هر یک از یالها یک توزیع پیوسته دارد و طول یالهای مختلف از نظر آماری مستقل هستند. در اینگونه روشها لازم است که توزیع احتمالی وزنها یالهای گراف از پیش مشخص باشد، در صورتی که در بسیاری از محیطهای واقعی توزیع احتمال یالها از پیش مشخص نمی باشد. برای گرافهای تصادفی در شرایطی که وزن یالها از قبل شناخته شده نیست و یا اطلاعات کاملی درباره آنها در دسترس نمی‌باشد تا کنون ۴ الگوریتم مبتنی بر اتوماتاهای یادگیر برای حل مساله یافتن کوتاهترین مسیر گزارش شده است. اولین الگوریتم توسط میبیدی-بیگی [2] ارائه شده

¹ Burton

² Waller

³ Ziliaskopoulos

⁴ Fan

⁵ Stochastic Shortest Path Problem with Correlated Link Cost

⁶ Dynamic Single-Source Shortest Path (DSSSP)

⁷ Frank

است که برای حل مسأله یافتن کوتاهترین مسیر بین دو گره در گرافهای تصادفی مورد استفاده قرار گرفته است. نسخه های اصلاح شده این الگوریتم در [1][6][13] گزارش شده است. دومین الگوریتم که توسط میسرا-اومن ارائه شده است برای یافتن کوتاهترین مسیر بین یک مبدأ و دیگرگره های گراف در یک گراف تصادفی طراحی شده است [3]. سومین الگوریتم نیز توسط میسرا-اومن ارائه شده است که میتواند برای یافتن کوتاهترین مسیر بین تمام گره های گراف در یک گراف تصادفی استفاده شود. [19]. در [6] یک الگوریتم مبتنی بر بازی بین اتوماتاهای یادگیر برای یافتن کوتاهترین مسیر بین یک مبدأ و دیگرگره ها در یک گراف تصادفی ارائه شده است.

در مدل های فوق فرض بر این است هزینه یالها در گراف مستقل از یکدیگر می باشد. در بسیاری از کاربردهای واقعی هزینه یالها کاملاً مستقل از هم نمی باشد و با تغییر ترافیک در یک قسمت از شبکه، ترافیک در قسمتهای مجاور نیز تغییر پیدا میکند. مساله کوتاهترین مسیر احتمالی با یالهای همبسته برای اولین بار توسط بارتون⁸ [15] [17] و پس از آن توسط والر⁹ و زیلیاسکوپولس¹⁰ [16] و فن¹¹ [4] مورد بررسی قرار گرفت. فن در [4] مسأله یافتن کوتاهترین مسیر در گراف های تصادفی با یالهای همبسته در شرایطی که هر یال در یکی از دو حالت ازدحام و بدون ازدحام ممکن است قرار گیرد و در عین حال توابع توزیع هزینه یالها از قبل شناخته شده باشند مطرح و الگوریتمی جهت حل آن پیشنهاد نمود. در این مقاله برای اولین بار الگوریتمی برای حل مساله کوتاهترین مسیر گرافهای تصادفی در شرایطی که همبستگی مابین هزینه یالها وجود دارد و همچنین توزیعهای احتمالی وزن یالها از قبل شناخته شده نیست پیشنهاد میگردد. در الگوریتم پیشنهادی از بازی بین اتوماتاهای یادگیر برای پیدا کردن کوتاهترین مسیر بین یک گره و دیگر گره های گراف استفاده میشود. برای این منظور شبکه های از اتوماتاهای یادگیر برای حل مساله همکاری می نمایند. در هر مرحله از الگوریتم تعدادی از اتوماتاهای یادگیر در این شبکه فعال شده و سپس هر یک از آنها یکی از اعمال خود را انتخاب می نمایند. اعمال انتخاب شده در محیط که همان گراف تصادفی می باشد اجرا و با توجه به نتیجه اعمال به آنها پاداش و یا جریمه داده می شود. الگوریتم پیشنهادی با استفاده از بازی بین اتوماتاهای یادگیر سعی میکند با حداقل تعداد نمونه گیری از یالهای گراف تصادفی درخت کوتاهترین مسیر را برای یک گره ریشه مشخص پیدا نماید.

در ادامه مقاله، ابتدا در بخش ۲ تعرف مساله، در بخش ۳ اتوماتاهای یادگیر، اتوماتای یادگیر توزیع شده و بازی بین اتوماتاهای یادگیر به اختصار معرفی میگردد. در بخش ۴ الگوریتم پیشنهادی و در بخش ۵ نتایج آزمایشها ارائه می شود. بخش پایانی مقاله نتیجه گیری میباشد.

۲. تعریف مساله

در این مسأله فرض می شود که یک شبکه با تعداد n گره و یک گره مقصد وجود دارد و یالها در دو حالت می توانند قرار داشته باشند؛ دارای ازدحام^{۱۲} و بدون ازدحام^{۱۳}. هدف از این مساله یافتن استراتژی بهینه (یال بهینه) در هر گره جهت رسیدن به گره مقصد می باشد. نحوه همبستگی بین یالها توسط تابع احتمال شرطی بیان می شود. در صورتی که یالی که توسط یک متحرک طی شده است تا به گره i برسد بدون ازدحام باشد تابع چگالی احتمال زمان سفر در یال e_{ij} توسط تابع $p_{ij}(t)$ نشان داده می شود و در صورتی که یالی که طی شده است تا به گره i برسد در حالت ازدحام باشد تابع چگالی احتمال زمان سفر در یال e_{ij} توسط تابع $q_{ij}(t)$ بیان می شود. متوسط زمان سفر در یال e_{ij} در حالت بدون ازدحام برابر است با t_{ij} و در حالت ازدحام برابر است با τ_{ij} ، که به ترتیب متوسط $p_{ij}(t)$ و $q_{ij}(t)$ می باشد. برای هر یال یک مقدار آستانه ازدحام τ^{0ij} وجود دارد. در صورتی که وزن یال بیشتر از τ^{0ij} باشد یال در حالت ازدحام خواهد بود و در غیر اینصورت یال در حالت بدون ازدحام می باشد. برای هر گره i چنانچه یالی که توسط یک متحرک طی شده است تا به گره i برسد بدون ازدحام باشد هزینه سفر از گره i تا گره مقصد از رابطه (۱) و چنانچه یالی که طی شده است در حالت ازدحام باشد هزینه سفر از گره i تا گره مقصد از رابطه (۲) بدست می آید. با انجام ساده سازی این دو رابطه می توان رابطه های (۳) و (۴) را بدست آورد. اثبات این رابطه ها در [4] آمده است.

$$u_i = \min_{j \neq i} \left\{ \int_0^{\tau^{0ij}} (\tau + u_j) p_{ij}(\tau) d\tau + \int_{\tau^{0ij}}^{\infty} (\tau + v_j) p_{ij}(\tau) d\tau \right\} \quad i = 1, 2, \dots, N - 1 \quad (1)$$

⁸ Burton

⁹ Waller

¹⁰ Ziliaskopoulos

¹¹ Fan

¹² Congested

¹³ Un-Congested

$$v_i = \min_{j \neq i} \left\{ \int_0^{\tau_{0ij}} (\tau + u_j) q_{ij}(\tau) d\tau + \int_{\tau_{0ij}}^{\infty} (\tau + v_j) h_{ij}(\tau) d\tau \right\}, i = 1, 2, \dots, N-1 \quad (2)$$

$$u_N, v_N = 0$$

چنانچه فرض شود:

$$\alpha_{ij} = \int_0^{\tau_{0ij}} p_{ij}(\tau) d\tau, \quad \lambda_{ij} = \int_0^{\tau_{0ij}} q_{ij}(\tau) d\tau$$

$$t_{ij} = \int_0^{\infty} \tau p_{ij}(\tau) d\tau, \quad \tau_{ij} = \int_0^{\infty} \tau q_{ij}(\tau) d\tau$$

خواهیم داشت:

$$u_i = \min_{j \neq i} \{ t_{ij} + \alpha_{ij} u_j + (1 - \alpha_{ij}) v_j \}, i = 1, 2, \dots, N-1 \quad (3)$$

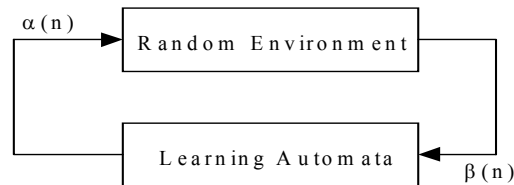
$$v_i = \min_{j \neq i} \{ \tau_{ij} + \lambda_{ij} u_j + (1 - \lambda_{ij}) v_j \}, i = 1, 2, \dots, N-1 \quad (4)$$

در رابطه های (۳) و (۴) برابر است با احتمال اینکه یال e_{ij} بدون ازدحام باشد بشرط آنکه یال طی شده توسط متحرک به گره i بدون ازدحام باشد و λ_{ij} برابر است با احتمال اینکه یال e_{ij} بدون ازدحام باشد بشرطی که یال طی شده توسط متحرک به گره i دارای ازدحام باشد. برای هر گره با توجه به وضعیت یال طی شده دو مسیر بهینه وجود دارد، یک مسیر به هنگامی که یال طی شده تا گره i در حالت ازدحام بوده و یک مسیر در حالتی که یال طی شده بدون ازدحام بوده است. برای اطلاعات بیشتر می توانید به [4] و [5] مراجعه کنید.

۳. اتوماتاهای یادگیر و اتوماتای یادگیر توزیع شده و بازی بین اتوماتاهای یادگیر

در این بخش به اختصار اتوماتاهای یادگیر، اتوماتای یادگیر توزیع شده و بازی بین اتوماتاهای یادگیر شرح داده میشود.

اتوماتاهای یادگیر: اتوماتای یادگیر یک مدل انتزاعی است که تعداد محدودی عمل را می تواند انجام دهد. هر عمل انتخاب شده توسط محیطی احتمالی ارزیابی شده و پاسخی به اتوماتای یادگیر داده می شود. اتوماتای یادگیر از این پاسخ استفاده نموده و عمل خود را برای مرحله بعد انتخاب می کند. شکل ۱ ارتباط بین اتوماتای یادگیر و محیط را نشان می دهد.



شکل ۱: ارتباط بین اتوماتای یادگیر و محیط

محیط را می توان توسط سه تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودیها، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه خروجیها و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمالهای جریمه می باشد. هر گاه β مجموعه دو عضوی باشد، محیط از نوع P می باشد. در چنین محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته می شود. در محیط از نوع Q، $\beta(n)$ می تواند به طور گسسته یک مقدار از مقادیر محدود در فاصله $[0, 1]$ و در محیط از نوع S، $\beta(n)$ متغیر تصادفی در فاصله $[0, 1]$ است. c_i احتمال اینکه عمل α_i نتیجه نامطلوب داشته باشد می باشد. در محیط ایستا^{۱۴} مقادیر c_i بدون تغییر می مانند، حال آنکه در محیط غیر ایستا^{۱۵} این مقادیر در طی زمان تغییر می کنند. اتوماتاهای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم بندی میگردند. در ادامه به شرح مختصری درباره ی اتوماتای یادگیر با ساختار متغیر که در این مقاله از آنها استفاده شده است، می پردازیم.

¹⁴ Stationary

¹⁵ Non-Stationary

اتوماتای یادگیر با ساختار متغیر^{۱۶}: یک اتوماتای یادگیر با ساختار متغیر توسط ۴ تایی $\{\alpha, \beta, p, T\}$ نشان داده می شود که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عملهای اتوماتا، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودیهای اتوماتا، $p \equiv \{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب هر یک از عملها، و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می باشد. در این نوع از اتوماتاها، اگر عمل α_i در مرحله n انتخاب شود و پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمالات کاهش می یابند. و برای پاسخ نامطلوب احتمال $p_i(n)$ کاهش یافته و سایر احتمالات افزایش می یابند. در هر حال، تغییرات به گونه ای صورت می گیرد تا حاصل جمع $p_i(n)$ ها همواره ثابت و مساوی یک باقی بماند. الگوریتم زیر یک نمونه از الگوریتمهای یادگیری خطی برای اتوماتای یادگیر با ساختار متغیر میباشد

الف- پاسخ مطلوب

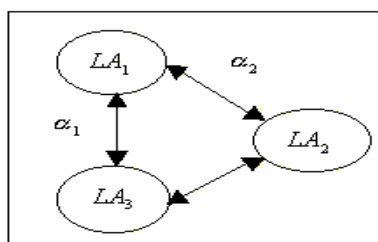
$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1-a)p_j(n) \quad j \neq i \quad \forall j \end{aligned}$$

ب- پاسخ نامطلوب

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

در روابط فوق، پارامتر پاداش و a پارامتر پاداش و b پارامتر جریمه می باشد. با توجه به مقادیر a و b سه حالت را می توان در نظر گرفت. زمانی که a و b با هم برابر باشند، الگوریتم L_{RP} ^{۱۷} می نامیم. زمانی که b از a خیلی کوچکتر باشد، الگوریتم L_{REP} ^{۱۸} می نامیم. زمانی که b مساوی صفر باشد، الگوریتم L_{RI} ^{۱۹} می نامیم. برای مطالعه بیشتر در باره اتوماتاهای یادگیر می توان به [8][9][10][11] مراجعه کرد.

اتوماتای یادگیر توزیع شده (DLA): اتوماتای یادگیر توزیع شده شبکه ای از اتوماتای یادگیر است که برای حل یک مساله با یکدیگر همکاری می نمایند [6]. تعداد اقدامهای یک اتوماتا در DLA برابر تعداد اتوماتاهای متصل به این اتوماتای یادگیر می باشد. انتخاب یک اقدام توسط یک اتوماتا در شبکه، اتوماتای متناظر با این اقدام را فعال می سازد. بعنوان مثال در شکل ۲ هر اتوماتا دارای دو اقدام می باشد. انتخاب اقدام α_2 توسط LA_1 ، اتوماتا یادگیر LA_3 را فعال خواهد کرد. اتوماتای یادگیر فعال شده (LA_3) بنوبه خود یکی از اقدامهای خود را انتخاب می کند که در نتیجه آن یکی از اتوماتاهای متصل به آن اتوماتا که متناظر با اقدام انتخاب شده می باشد فعال می شود. در هر زمان فقط یک اتوماتا در شبکه فعال میباشد. بطور رسمی DLA را میتوان توسط گراف $DLA = (V, E)$ که $V = \{LA_1, LA_2, \dots, LA_n\}$ مجموعه اتوماتای یادگیر و n تعداد اتوماتاها در DLA و $E \subset V \times V$ مجموعه لبه های گراف می باشد، تعریف کرد. لبه (i, j) اقدام j اتوماتا LA_i را نشان می دهد. زمانی فعال خواهد شد که اقدام j اتوماتون LA_i انتخاب شود. تعداد اقدامهای اتوماتا LA_k ($k = 1, 2, \dots, n$) برابر درجه ی خروجی آن گره می باشد. برای اطلاعات بیشتر در باره DLA میتوان به مراجع [1] و [6] مراجعه کرد.



شکل ۲: اتوماتای یادگیر توزیع شده (DLA) با ۳ اتوماتا یادگیر

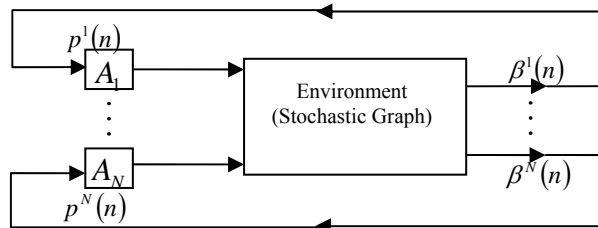
^{۱۶} Variable Learning Automata

^{۱۷} Linear Reward Penalty

^{۱۸} Linear Reward Epsilon Penalty

^{۱۹} Linear Reward Inaction

بازی بین اتوماتاهای یادگیر:²⁰ در بازی بین اتوماتاهای یادگیر که در این مقاله استفاده شده است شبکه ای از اتوماتاهای یادگیر برای حل مساله همکاری مینمایند. در این شبکه در هر مرحله تمامی اتوماتاهای یادگیر فعال شده و سپس هر یک از آنها یکی از اعمال خود را انتخاب مینمایند. اعمال انتخاب شده در محیط اجرا و با توجه به نتیجه اعمال به آنها پاداش و یا جریمه داده میشود. یالهای ورودی هر گره در این شبکه به عنوان عملهای اتوماتای یادگیر متناظر با آن گره در نظر گرفته شده اند. در شکل ۳ ارتباط بین اتوماتاهای یادگیر و محیط که در مساله ما گراف تصادفی میباشد نشان داده شده است. برای اطلاعات بیشتر در باره بازیهای بین اتوماتاهای یادگیر میتوان به [8] مراجعه کرد.



شکل ۳: ارتباط بین اتوماتاهای یادگیر و محیط یک در بازی بین اتوماتاهای یادگیر

۴. الگوریتم پیشنهادی

برای حل این مسئله هر گره شبکه به یک گروه ۴ اتوماتای یادگیر (QLA^{21}) که طبق شکل ۴ با محیط تصادفی در ارتباط می باشند مجهز می باشد. در هر زمان با توجه به شرایط محیط فقط دو اتوماتای یادگیر از ۴ اتوماتای یادگیر در هر گره فعال خواهند بود: ۲ اتوماتای یادگیر برای زمانیکه محیط در حالت ازدحام باشد و ۲ اتوماتای دیگر برای زمانی که محیط در حالت بدون ازدحام باشد. یکی از ۲ اتوماتای فعال در هر زمان وظیفه یادگیری همبستگی را به عهده دارند که اتوماتای یادگیرنده همبستگی نامیده می شوند و اتوماتای یادگیر فعال دیگر وظیفه یادگیری یال بهینه را به عهده دارد که اتوماتای یادگیرنده یال بهینه نامیده می شوند. در این شبکه هر اتوماتای یادگیر با ۲ اندیس مشخص می شود. اندیس اول مشخص کننده نوع اتوماتای یادگیر (یادگیرنده یال بهینه e) و یا یادگیرنده همبستگی (c) و اندیس دوم مشخص کننده شرایط ترافیکی میباشد (cg برای حالت ازدحام و ucg برای حالت بدون ازدحام). وظیفه هر یک از این ۴ اتوماتای یادگیر در هر گره i به شرح زیر میباشد.

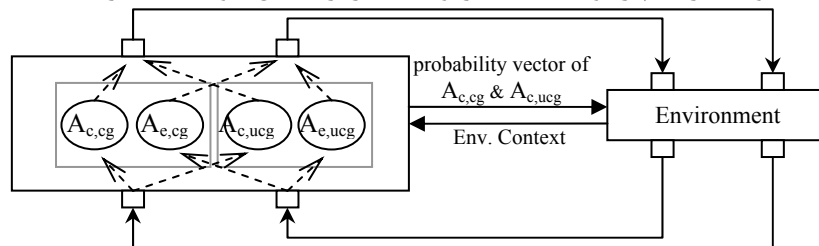
$A_{c,cg}^i$: اتوماتای یادگیر برای یادگیری همبستگی بین یالها در شرایطی که محیط دارای ازدحام باشد.

$A_{e,cg}^i$: اتوماتای یادگیر برای یادگیری انتخاب یال بهینه در شرایطی که محیط دارای ازدحام باشد.

$A_{c,ucg}^i$: اتوماتای یادگیر برای یادگیری همبستگی بین یالها در شرایطی که محیط بدون ازدحام باشد.

$A_{e,ucg}^i$: اتوماتای یادگیر برای یادگیری انتخاب یال بهینه در شرایطی که محیط بدون ازدحام باشد.

ورودی های QLA علاوه بر پاداش و جریمه هایی که توسط محیط به عملهای هر اتوماتای یادگیر داده می شود و وضعیت جاری محیط (ازدحام و یا بدون ازدحام) نیز می باشد که باعث فعال شدن اتوماتای یادگیر مربوط به هر حالت می شود. خروجی های QLA علاوه بر عملهای انتخاب شده توسط هر کدام از اتوماتاهای یادگیر، وضعیت (بردار احتمال) اتوماتای یادگیرنده همبستگی QLA نیز می باشد.



شکل ۴: یک QLA و نحوه رابطه آن با محیط

در ابتدای الگوریتم شبکه ای از QLA ها، متناظر با گراف مساله ایجاد میشود. در هر مرحله از الگوریتم تعدادی از اتوماتاهای یادگیر در این شبکه فعال شده و هر یک از آنها یکی از اعمال خود را انتخاب می نمایند. اعمال انتخاب شده در محیط که همان گراف تصادفی

²⁰ Game of Learning Automata

²¹ Quadra Learning Automata

می‌باشد اجرا و با توجه به نتیجه اعمال به آنها پاداش و یا جریمه داده می‌شود. الگوریتم پیشنهادی سعی میکند با حداقل تعداد نمونه گیری از یالهای گراف تصادفی در شرایطی که توزیع احتمال وزن یالها از قبل شناخته شده نیست کوتاهترین مسیر را از گره‌های گراف به یک گره مقصد مشخص پیدا نماید. همچنین این الگوریتم احتمال های همبستگی بین یالها را نیز بدست می آورد (λ_{ij} و α_{ij}). گره در حالت ازدحام قرار دارد اگر اکثر یالهای ورودی به گره در حالت ازدحام باشد. چنانچه تعداد یالهای در حالت ازدحام و بدون ازدحام مساوی باشد بصورت تصادفی یکی از حالت‌های ازدحام و یا بدون ازدحام انتخاب می شود. مراحل الگوریتم به شرح زیر میباشد.

مرحله ۱: یک شبکه از QLA متناظر با گراف مسئله ایجاد می‌شود. تمام اتوماتاهای یادگیرنده یال بهینه از الگوریتم یادگیری L_{R-I} و تمام اتوماتاهای یادگیرنده همبستگی از الگوریتم یادگیری L_{R-P} استفاده میکنند. نرخ یادگیری دو اتوماتای یادگیرنده یال بهینه یکسان در نظر گرفته می‌شود و همچنین ضرایب جریمه و پاداش در دو اتوماتای یادگیرنده همبستگی نیز یکسان می‌باشد. اقدامهای هر اتوماتای یادگیرنده یال بهینه، یالهای خروجی از گره متناظر با QLA می‌باشد؛ در گره i اقدامهای اتوماتای یادگیرنده یال بهینه در شرایط ازدحام $\alpha_{e,ug}^i = \{\alpha_{(e,ug)_1}^i, \alpha_{(e,ug)_2}^i, \dots, \alpha_{(e,ug)_{r_i}}^i\}$ و اقدامهای اتوماتای یادگیرنده یال بهینه در شرایط بدون ازدحام $\alpha_{e,ug}^i = \{\alpha_{(e,ug)_1}^i, \alpha_{(e,ug)_2}^i, \dots, \alpha_{(e,ug)_{r_i}}^i\}$ میباشد. بردارهای احتمال اتوماتاهای یادگیرنده یال بهینه برای شرایط ازدحام و عدم ازدحام به ترتیب $p_{e,ug}^i = (p_{(e,ug)_1}^i, p_{(e,ug)_2}^i, \dots, p_{(e,ug)_{r_i}}^i)$ و $p_{e,ug}^i = (p_{(e,ug)_1}^i, p_{(e,ug)_2}^i, \dots, p_{(e,ug)_{r_i}}^i)$ میباشد. مقدار اولیه هر کدام از اعضای این دوبردار برابر با $1/r_i$ می باشد که r_i تعداد یالهای خروجی از گره i است. عملهای مجاز هر یک از ۲ اتوماتای یادگیرنده همبستگی "ازدحام" (cg) و "بدون ازدحام" (ug) می باشد؛ یعنی $\alpha_{c,ug} = \{cg, ug\}$ و $\alpha_{c,ug} = \{cg, ug\}$ ، $p_{c,ug}^i = (p_{(c,ug)_{cg}}^i, p_{(c,ug)_{ug}}^i)$ و $p_{c,ug}^i = (p_{(c,ug)_{cg}}^i, p_{(c,ug)_{ug}}^i)$ برای هر یک از گره‌ها دو مقدار آستانه پویا در نظر گرفته می‌شود، t_{cg}^i و t_{ug}^i که به ترتیب آستانه پویای گره i در حالت ازدحام و بدون ازدحام می‌باشد. مقدار هر دو آستانه در گره مقصد، صفر می‌باشد و در طول اجرای الگوریتم تغییری نمی‌کند. آستانه پویا برای گره i با استفاده از رابطه‌های (۷) و (۸) بروز رسانی می شود.

$$d_{cg}^i = w_{cg}(e_{ij}) + p_{(c,ug)_{cg}}^i t_{cg}^j + p_{(c,ug)_{ug}}^i t_{ug}^j \quad (5)$$

$$d_{ug}^i = w_{ug}(e_{ij}) + p_{(c,ug)_{cg}}^i t_{cg}^j + p_{(c,ug)_{ug}}^i t_{ug}^j \quad (6)$$

$$t_{cg}^i = \frac{d_{cg}^i + t_{cg}^i (k_{cg}^i - 1)}{k_{cg}^i} \quad (7)$$

$$t_{ug}^i = \frac{d_{ug}^i + t_{ug}^i (k_{ug}^i - 1)}{k_{ug}^i} \quad (8)$$

که $k_{cg}^i - 1$ و $k_{ug}^i - 1$ به ترتیب برابر است با تعداد دفعاتی که t_{cg}^i و t_{ug}^i تا بحال بروز رسانی شده است. $w(e_{ij})$ نمونه گرفته شده از یال e_{ij} می‌باشد. در ابتدا بجز گره مقصد مقدار آستانه دیگر گره‌ها به ۱- مقدار دهی اولیه می شود.

مرحله ۲: تمامی QLA ها در شبکه بطور همزمان فعال شده و هر کدام با توجه به اینکه حالت گره متناظر با QLA در حالت ازدحام یا در حالت بدون ازدحام باشد اتوماتاهای مربوط به آن حالت فعال شده و یکی از عمل های ممکن خود را انتخاب می‌کند. به عبارتی در هر QLA دو اتوماتا فعال می‌شود. اتوماتای یادگیرنده همبستگی یکی از دو عمل ممکن cg و ug را انتخاب می‌کند و اتوماتای یادگیرنده یال بهینه یک یال از یالهای خروجی گره متناظر با QLA را انتخاب می‌کند.

مرحله ۳: برای هر یک از یالهای انتخاب شده e_{ij} توسط اتوماتاهای یادگیر، نمونه $w(e_{ij})$ از وزن یال گرفته می‌شود. با توجه به مقدار نمونه گیری شده و مقادیر آستانه پویا در گره سمت دیگر یال e_{ij} ، نحوه بروز رسانی بردار احتمال دو اتوماتای فعال در QLA گره i بصورت زیر می‌باشد:

اتوماتای یادگیرنده همبستگی: اگر مقدار $w(e_{ij})$ از آستانه ازدحام (τ^{0ij}) بیشتر باشد، یال در حالت ازدحام و در غیر این صورت در حالت بدون ازدحام می‌باشد. با توجه به حالت یال انتخاب شده (دارای ازدحام یا بدون ازدحام) به اتوماتای یادگیرنده همبستگی پاداش یا جریمه تعلق می‌گیرد، چنانچه عمل "ازدحام" توسط اتوماتای یادگیر انتخاب شده باشد و یال نیز در حالت "ازدحام" باشد (وزن یال بیش از آستانه ازدحام باشد) به عمل انتخاب شده پاداش و در غیر اینصورت جریمه داده می‌شود. همچنین اگر اتوماتای یادگیر عمل

"بدون ازدحام" را انتخاب کرده باشد و یال نیز در حالت "بدون ازدحام" باشد، به عمل انتخاب شده پاداش و در غیر اینصورت جریمه داده می‌شود.

اتوماتای یادگیرنده یال بهینه: اگر مقدار استانه گره سمت دیگر یال انتخاب شده ۱- نباشد، با توجه به حالت گره یکی از مقادیر رابطه (۵) و (۶) محاسبه می‌شود. اگر گره در حالت بدون ازدحام باشد رابطه (۶) و در غیر این صورت رابطه (۵) محاسبه می‌شود. سپس این مقدار با مقدار استانه مربوط به حالت جاری گره مقایسه می‌شود. یعنی اگر گره در حالت ازدحام باشد با t_{cg} و اگر گره در حالت بدون ازدحام باشد با t_{ucg} مقایسه می‌شود. چنانچه کمتر و یا برابر با استانه پویا باشد و یا مقدار استانه گره جاری ۱- باشد به عمل انتخاب شده توسط اتوماتای یادگیر پاداش و در غیر این صورت جریمه داده می‌شود. سپس مقدار استانه بروز رسانی می‌شود. برای بروز کردن مقدار استانه می‌توان از روشهای مختلفی استفاده نمود که بعداً به آنها اشاره خواهد شد.

مرحله ۴: اگر شرط خاتمه الگوریتم برقرار باشد، اجرای الگوریتم متوقف می‌شود در غیر اینصورت کنترل به مرحله ۲ انتقال پیدا می‌کند.

شبه کد الگوریتم پیشنهادی در شکل ۵ آمده است.

Procedure GA- SSPCL

Input: Graph $G=(V,E)$,

begin

Create a network of QLA corresponding to the graph of the problem

Let t_{state}^i be the dynamic threshold for node i in state $state$ which can be congested or uncongested

Let $ActiveLA_c^i$ be the active learning automata in node i responsible for congestion.

Let $ActiveLA_e^i$ be the active learning automata in node i responsible for optimal edge.

repeat

For every node i except for the destination node do in parallel

If node i is congested then

$ActiveLA_c^i = A_{c,cg}^i$;

$ActiveLA_e^i = A_{e,cg}^i$;

end if

If node i is not congested then

$ActiveLA_c^i = A_{c,ucg}^i$

$ActiveLA_e^i = A_{e,ucg}^i$

end if

$state \leftarrow ActiveLA_c^i$ selects an action based on its probability vector.

$edge \leftarrow ActiveLA_e^i$ selects an action based on its probability vector.

if (the state of selected edge = congestion) then

Reward the action selected by the $ActiveLA_c^i$ according to the L_{RP} learning Algorithm;

else

penalize the action selected by the $ActiveLA_c^i$ according to the L_{RP} learning Algorithm;

end if

if (state = congested)

Compute d using equation 5

end if

if (state = uncongested)

Compute d using equation 6

end if

if $d \leq t_{state}^i$ then;

Reward the action selected by $ActiveLA_e^i$ according to the L_{RI} learning Algorithm;

end if

//update dynamic threshold

if (state = congested)

Compute t_{state}^i using equation 7

end if

if (state = uncongested)

Compute t_{state}^i using equation 8

end if

end for

until algorithm converges

end GA- SSPCL

شکل ۵) شبه کد الگوریتم پیشنهادی برای حل مسأله SSPCL

۵. نتایج شبیه‌سازی ها

برای شبیه سازی و مقایسه عملکرد میان الگوریتم های مختلف از ۲ شبکه که در [4] معرفی شده استفاده می‌شود. برای طول یالها در هر یک از این شبکه ها از یک تابع توزیع احتمال گسسته با مقادیر مثبت استفاده شده است. یالهای بهینه و مقدار هزینه لازم برای

رفتن به گره مقصد در هر دو حالت برای هر گره توسط (۳) و (۴) مشخص میشود که با استفاده از روش تخمین پیکارد که در [5][4] ارائه شده است، براساس میانگین طول یالها محاسبه میشود.

در این بخش برای بررسی عملکرد الگوریتم های پیشنهادی معیارهای زیر استفاده میشود.

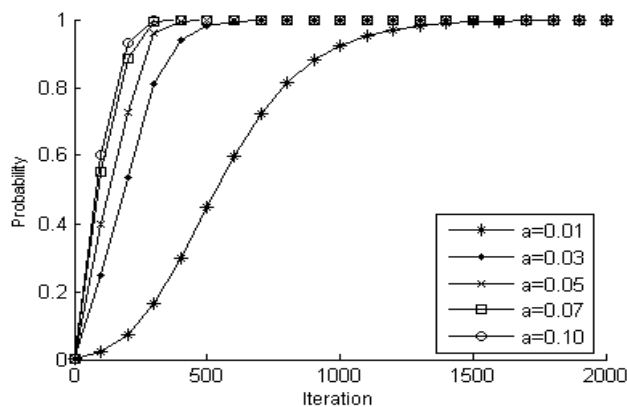
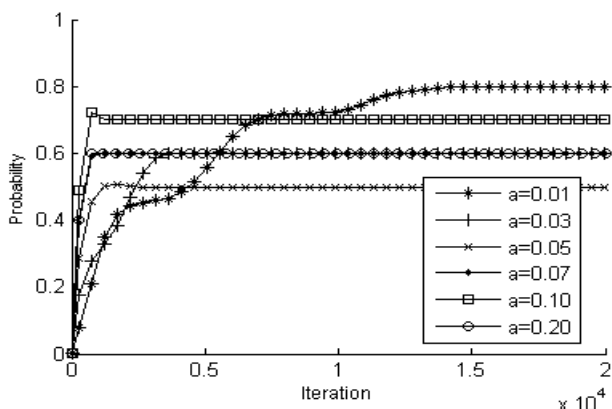
- تعداد نمونه هایی که توسط الگوریتم از یالهای گراف گرفته شده است. با پیشرفت الگوریتم نمونه گیری بر روی یالهایی متمرکز میشود که جز یال بهینه در هر گره می باشند و نمونه گیریها از یالهایی که جز یالهای بهینه نیستند کاهش میابد.
- احتمال انتخاب یالهای بهینه که حاصل ضرب احتمالات انتخاب یالهای بهینه توسط اتوماتاهای یادگیرنده یال بهینه در هر گره می باشد. این احتمال با پیشرفت الگوریتم بایستی به سمت یک میل کند.

نتایج حاصل از اجرای هر الگوریتم در یک جدول که نشاندهنده تعداد کل تکرارهای لازم جهت اجرای الگوریتم (AVI)، میانگین تعداد اجراهای همگرا شده (PC)، میانگین تعداد کل نمونه های گرفته شده (TS) و تعداد نمونه های گرفته شده از یالهای بهینه (SPS) نشان داده می شود.

الگوریتم ۱: ضریب یادگیری اتوماتاها در تمام QLA ها ثابت در نظر گرفته می شود و در طول اجرای الگوریتم تغییر نمیکند. در هر مرحله مقادیر آستانه در هر QLA مطابق با رابطه (۵) و (۶) بروز میشود. نتایج اجرای این الگوریتم که حاصل میانگین ۱۰۰ مرتبه اجرا بروی گرافهای ۱ و ۲ می باشد در جدول زیر نشان داده شده است.

جدول (۱) میانگین تعداد تکرار ها، تکرارهای همگرا شده، تعداد کل نمونه گیری ها و تعداد نمونه های گرفته شده از یالهای بهینه در الگوریتم ۱

	Graph 1					Graph 2			
$a_{p,\psi}$	AVI	PC	TS	SPS		AVI	PC	TS	SPS
0.005	824.5	100	2456.1	1525.8		1730.2	99	13770.7	11126.0
0.010	408.5	100	1207.3	748.2		695.9	95	5498.3	4269.5
0.030	139.2	100	400.7	245.7		178.8	84	1360.7	967.9
0.050	85.3	100	237.1	147.8		112.8	85	833.3	600.7
0.070	65.3	100	178.3	111.4		98.9	82	723.4	536.8
0.100	48.1	100	127.6	81.4		67.3	73	467.8	340.0
0.200	28.2	92	67.9	44.6		39.3	48	253.6	191.8



نمودار (۲) همگرایی الگوریتم ۱ برای مسئله SSPCL در گراف ۲

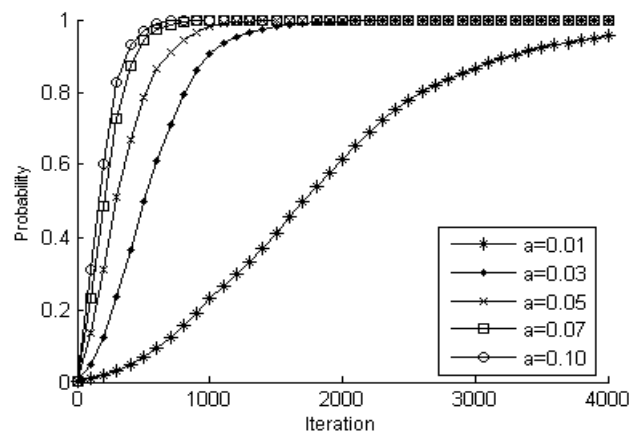
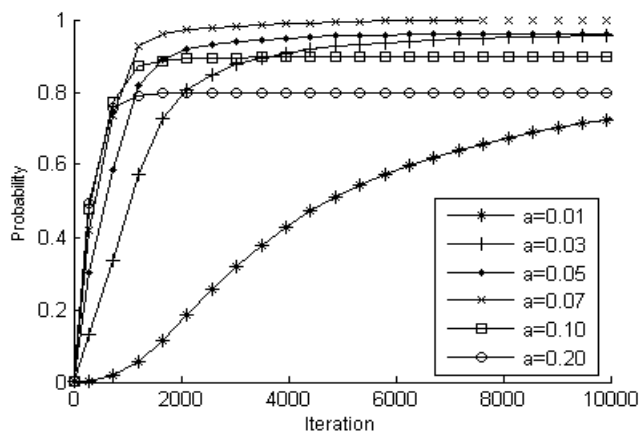
نمودار (۱) همگرایی الگوریتم ۱ برای مسئله SSPCL در گراف ۱

الگوریتم ۲: الگوریتم ۲ همان الگوریتم ۱ است که در آن مقدار t_{ψ}^i زمانی بروز میشود که اتوماتای یادگیرنده یال بهینه QLA^i پاداش دریافت کرده باشد. نتایج اجرای این الگوریتم که حاصل میانگین ۱۰۰ مرتبه اجرا بروی گرافهای ۱ و ۲ می باشد در جدول زیر نشان داده شده است.

جدول (۲) میانگین تعداد تکرار ها، تکرارهای همگرا شده، تعداد کل نمونه گیری ها و تعداد نمونه های گرفته شده از یالهای بهینه در الگوریتم ۲

	Graph 1					Graph2			
$a_{p,\psi}$	AVI	PC	TS	SPS		AVI	PC	TS	SPS
0.005	2929.2	100	8768.7	5623.3		2232.6	100	17790.2	12017.9
0.010	1340.7	100	4003.9	2624.3		1015.9	100	8060.9	5495.4

0.030	407.0	100	1204.7	797.4		277.0	100	2150.1	1458.6
0.050	217.9	100	636.5	425.9		146.8	100	1104.7	740.8
0.070	147.2	99	423.7	285.7		106.2	100	781.1	528.7
0.100	93.3	98	261.9	176.6		77.9	99	553.6	381.2
0.200	42.4	86	121.0	83.1		50.1	73	333.9	249.6



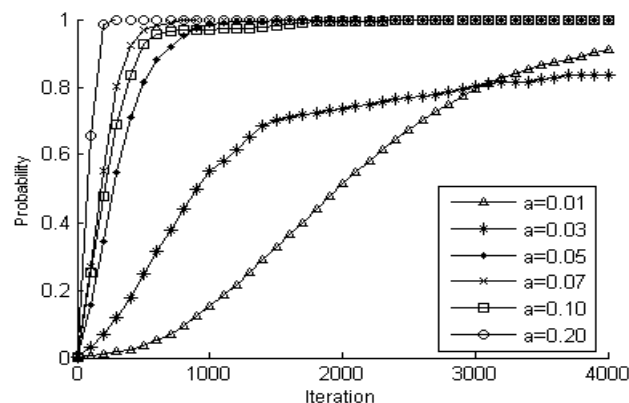
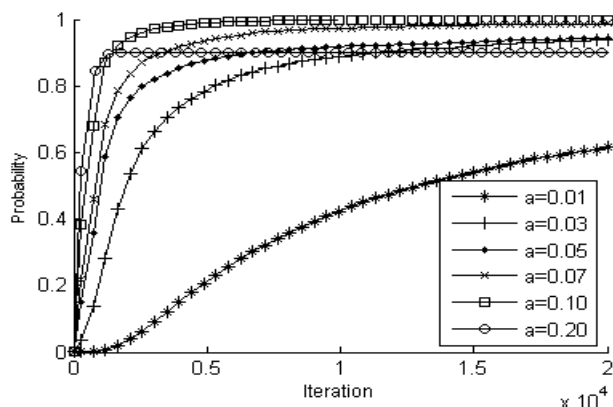
نمودار ۴) همگرایی الگوریتم ۲ برای مسئله SSPCL در گراف ۲

نمودار ۳) همگرایی الگوریتم ۲ برای مسئله SSPCL در گراف ۱

الگوریتم ۳: همان الگوریتم ۲ است که ضریب یادگیری برای دو اتوماتای یادگیرنده یال بهینه در گره i برابر با $a/dist[i]$ می باشد که $dist[i]$ حداقل تعداد یالهایی است که برای رسیدن به گره i از مبدا بایستی طی شود. $dist[i]$ در ابتدای الگوریتم به تعداد گره های گراف مقدار دهی میشود. در هر مرحله چنانچه برای یال e_{ij} که توسط یکی از اتوماتاهای یادگیر گره i انتخاب شده است، مقدار $dist[i]$ از $dist[j]+1$ بیشتر باشد مقدار $dist[i]$ به $dist[j]+1$ تغییر پیدا میکند. به بیان دیگر هر چه یک گره به گره مقصد نزدیکتر باشد ضریب یادگیری اتوماتای یادگیر ن گره بیشتر میباشد. البته لازم به ذکر است که اتوماتاهای یادگیر مربوط به یادگیری همبستگی در تمام QLA دارای مقدار ضریب یادگیری ثابت می باشند زیرا همبستگی بین یالها به دور یا نزدیک بودن به مقصد ارتباطی ندارد.

جدول ۳) میانگین تعداد تکرار ها، تکرارهای همگرا شده، تعداد کل نمونه گیری ها و تعداد نمونه های گرفته شده از یالهای بهینه در الگوریتم ۳

$a_{p,\psi}$	Graph 1					Graph 2			
	AVI	PC	TS	SPS		AVI	PC	TS	SPS
0.005	3407.9	100	10205.6	6455.1		3641.1	100	29060.3	19158.5
0.010	1550.1	100	4631.5	2953.4		1584.6	100	12605.0	8293.3
0.030	444.7	100	1314.9	844.5		439.8	100	3451.2	2259.1
0.050	229.2	100	668.9	435.9		240.9	100	1861.9	1224.2
0.070	164.1	99	474.3	309.7		168.7	100	1282.0	847.7
0.100	109.4	99	310.3	204.3		120.6	100	898.5	608.6
0.200	50.5	90	134.4	90.8		50.1	73	333.9	249.6



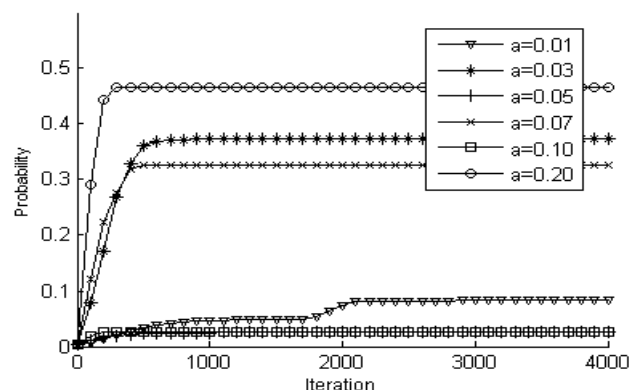
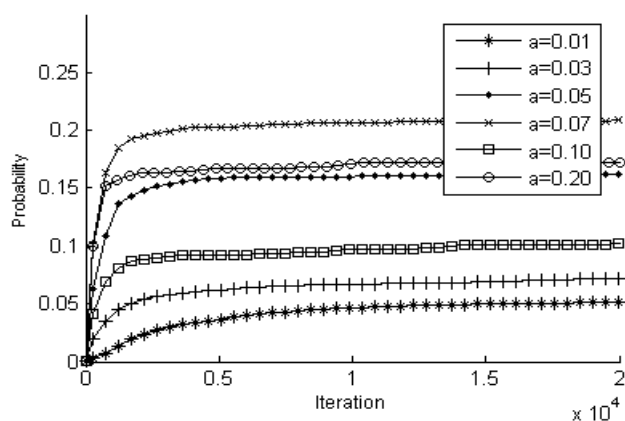
نمودار ۶) همگرایی الگوریتم ۳ برای مسئله SSPCL در گراف ۲

نمودار ۵) همگرایی الگوریتم ۳ برای مسئله SSPCL در گراف ۱

الگوریتم ۴: این الگوریتم مانند الگوریتم ۱ است با این تفاوت که در آن برای محاسبه مقدار استاندارد، از متوسط نمونه های یال e_{ij} که تا بحال گرفته شده است به جای نمونه گرفته شده فعلی استفاده می شود. آزمایشها نشان داده است که استفاده از تمامی نمونه های گرفته شده از یک یال برای محاسبه استاندارد منجر به نتایج بهتری از نظر تعداد همگراییها به جواب بهینه میشود. نتایج در جدول ۴ آمده است.

جدول ۴) میانگین تعداد تکرار ها، تکرارهای همگرا شده، تعداد کل نمونه گیری ها و تعداد نمونه های گرفته شده از یالهای بهینه در الگوریتم ۴

$a_{p,\psi}$	Graph 1					Graph 2			
	AVI	PC	TS	SPS		AVI	PC	TS	SPS
0.005	3700.2	99	11082.7	7467.2		6692.0	100	53464.7	38711.9
0.010	2842.7	99	8510.2	5911.2		2129.1	100	16965.1	12146.5
0.030	2059.6	98	6161.6	4435.2		840.4	100	6653.7	4884.9
0.050	425.9	98	1257.9	890.3		820.6	100	6498.3	4884.4
0.070	439.2	100	1300.5	935.2		238.5	99	1836.5	1336.4
0.100	407.1	96	1205.9	891.3		188.9	93	1445.9	1080.7
0.200	1156.4	78	3449.3	2617.9		68.8	68	487.3	368.5



نمودار ۸) همگرایی الگوریتم ۴ برای مسئله SSPCL در گراف ۲

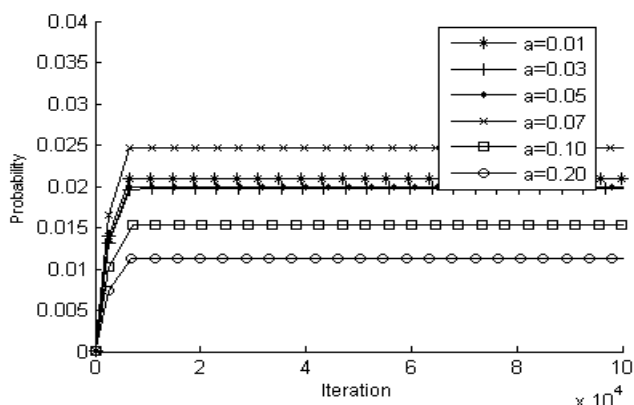
نمودار ۷) همگرایی الگوریتم ۴ برای مسئله SSPCL در گراف ۱

الگوریتم ۵: این الگوریتم همانند الگوریتم ۲ می باشد با این تفاوت اگر هر دو اتوماتای یادگیرنده یال بهینه یک گره همگرا شوند، ان گره غیر فعال میگردد که این باعث کاهش تعداد نمونه گیری های لازم جهت اجرای الگوریتم میشود. نتایج اجرای این الگوریتم در جدول

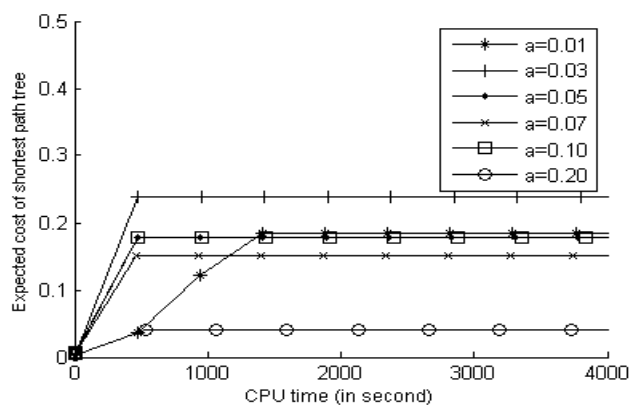
زیر نشان داده شده است. منظور از همگرا شدن یک اتوماتای یادگیر، رسیدن احتمال انتخاب یکی از اعمالش از یک مقدار از پیش تعیین شده بیشتر گردد. این مقدار برای آزمایشهای که نتایج آنها در جدول ۵ آمده ۰,۸۵ در نظر گرفته شده است.

جدول ۵) میانگین تعداد تکرار ها، تکرارهای همگرا شده، تعداد کل نمونه گیری ها و تعداد نمونه های گرفته شده از یالهای بهینه در الگوریتم ۵

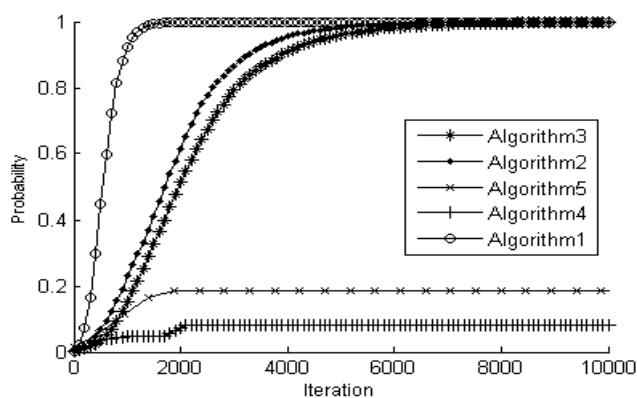
	Graph 1					Graph 2			
$a_{p,\psi}$	AVI	PC	TS	SPS		AVI	PC	TS	SPS
0.005	2942.5	100	6959.8	4085.3		2245.3	100	13865.2	8544.2
0.010	1421.4	100	3359.5	1980.0		1030.5	100	6396.6	3978.7
0.030	411.0	100	917.2	544.9		280.3	100	1741.1	1092.6
0.050	219.5	100	491.8	293.6		156.2	100	925.9	579.6
0.070	145.7	99	319.3	190.2		106.6	95	630.6	398.2
0.100	92.5	94	199.6	122.5		75.0	84	403.4	256.9
0.200	41.9	73	86.0	55.8		41.0	46	196.0	133.2



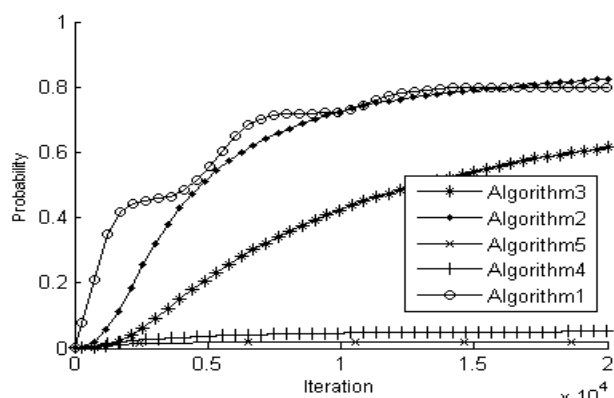
نمودار ۱۰) همگرایی الگوریتم ۵ برای مسئله SSPCL در گراف ۲



نمودار ۹) همگرایی الگوریتم ۵ برای مسئله SSPCL در گراف ۱



نمودار ۱۲) همگرایی الگوریتمهای مختلف برای مسئله SSPCL در گراف ۱



نمودار ۱۱) همگرایی الگوریتمهای مختلف برای مسئله SSPCL در گراف ۲

با توجه به نتایج آزمایشهای انجام شده میتوان نتیجه گرفت که با افزایش پارامتر یادگیری اتوماتای یادگیر یال بهینه تعداد تکرارهای لازم جهت اجرای الگوریتم کاهش می یابد ولی در مقابل درصد اجراهای همگرا شده به جواب بهینه نیز کاهش میابد. همچنین با توجه به نتایج ارائه شده در ستونهای TS و SPS مشاهده می شود که تعداد نمونه گیری ها از یالهای جواب بهینه بیشتر از دیگر یالها در گراف میباشد. همانطور که در نمودار ۱۱ و نمودار ۱۲ مشخص است الگوریتم ۱ از دیگر الگوریتمهای ارائه شده دارای نرخ همگرایی بالاتری میباشد. البته لازم به ذکر است که داشتن سرعت همگرایی بالا به معنی دقت بالاتر نیست، همانطور که در جداول فوق نشان داده شده است الگوریتم های ۲ و ۳ نسبت به دیگر الگوریتم ها از دقت بالاتری برخوردار هستند ولی دارای سرعت همگرایی پایین تری میباشد. در

الگوریتم ۵. بدلیل اینکه بعضی از گره ها در حین اجرای الگوریتم غیر فعال میشوند، تعداد نمونه گیری های لازم جهت اجرای الگوریتم ۵ نسبت به دیگر الگوریتمها کمتر می باشد.

۶. نتیجه گیری

. در این مقاله برای اولین بار الگوریتمی برای حل مساله کوتاهترین مسیر گرافهای تصادفی در شرایطی که همبستگی مابین هزینه یالها وجود دارد و همچنین توزیعهای احتمالی وزن یالها از قبل شناخته شده نیست پیشنهاد گردید برای این منظور شبکه ای از اتوماتاهای یادگیر که متناظر با گراف مساله میباشد برای حل مساله همکاری مینمایند. الگوریتم پیشنهادی سعی میکند با حداقل تعداد نمونه گیری از یالهای گراف تصادفی در شرایطی که توزیعهای احتمالی وزن یالها از قبل شناخته شده مسیر بهینه را در هر گره جهت ادامه سفر با توجه به وضعیت یال پیموده شده تا گره جاری را در هر دو حالت دارای ازدحام و بدون ازدحام مشخص کند. نتایج آزمایشها کارایی الگوریتم پیشنهادی را نشان داد.

مراجع

- [1] H. Beigy and M. R. Meybodi, "Utilizing Distributed Learning Automata to Solve Stochastic Shortest Path Problem", International Journal of Uncertainty, Fuzziness and Knowledge-based Systems, Vol. 14, No. 5, pp. 591-615, 2006.
- [2] M. R. Meybodi and H. Beigy, "Solving Stochastic Shortest Path Problem Using Distributed Learning Automata", Proceedings of CSICC-2001, Isfahan, Iran, pp. 70-86, 2001.
- [3] H. Frank, "Shortest Paths in Probabilistic Graphs", Oper. Res., No. 17, pp.583-599, 1969.
- [4] Y. Fan, "Optimal Routing Through Stochastic Networks", PhD Dissertation, may 2003.
- [5] Y. Fan, R. Kalaba, and J. Moore, "Shortest Paths in Stochastic Networks with Correlated Link Costs", Computers and Mathematics with Applications (CAMWA), Vol. 49, pp. 1549-1564, 2005.
- [6] H. Beigy and M. R. Meybodi, "A New Distributed Learning Automata for Solving Stochastic Shortest Path Problem", Proceedings of the Sixth International Joint Conference on Information Science, Durham, USA, pp. 339-343, 2002.
- [7]. اصغر قربانی و محمدرضا میبیدی، "حل مساله کوتاهترین مسیر بین یک گره و دیگر گره ها در گرافهای تصادفی با استفاده از بازی بین اتوماتاهای یادگیر"، گزارش فنی، آزمایشگاه سیستمهای نرمافزاری دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر، مرداد ماه ۱۳۸۶.
- [8] K. S. Narendra and K. S. Thathachar, Learning Automata: An Introduction, New York: Prentice-Hall, 1989.
- [9] S. Lakshmivarahan, Learning Algorithms: Theory and Applications. New York: Springer-verlag, 1981.
- [10] M. R. Meybodi and S. Lakshmivarahan, "On a Class of Learning Algorithms which have Symmetric Behavior under Success and Failer", Lecture Notes in Statistics, Berlin: Springer-Verlag, pp. 145-155, 1984.
- [11] P. Mars, J. R. Chen, and R. Nambir, Learning Algorithms: Theory and Applications in Signal Processing, Control, and Communication. CRC Press Inc., 1996.
- [12] N. Baba, "New Topics in Learning Automata Theory and Applications", Number 71, Lecture Notes in Control and Information Sciences, Berlin: Springer-Verlag, 1985.
- [13]. J. McQuillan, I. Richer, and E. Rosen, "The New Routing Algorithm for the ARPANET", IEEE Trans. Commun., vol. COM-28, pp. 711-719, 1980.
- [14] P. Spira and A. Pan, "On Finding and Updating Spanning Trees and Shortest Paths", SIAM J. Comput., vol. 4, no. 3, pp. 375-380, 1975.
- [15] D. Burton, On the Inverse Shortest Path Problem, PhD Dissertation, Faculties University of Notre-Dame De La Paix De Namur, 1993.
- [16] S. T. Waller and A. K. Ziliaskopoulos, On the Online Shortest Path Problem with Limited Arc Cost Dependencies, Networks, 40 (4), 216-227 (2002).
- [17] D. Burton, On the Use of an Inverse Shortest Paths Problem for Recovering Linearly Correlated Cost, Mathematical Programming, No. 63, pp. 1-22, 1994.
- [18] Misra, S. and Oommen, B. J., "Dynamic Algorithms for the Shortest Path Routing Problem: Learning Automata-based Solutions", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-35(B), pp. 1179-1192, June 2005.
- [19] S. Misra and B. J. Oommen, "An Efficient Dynamic Algorithm for Maintaining All-Pairs Shortest Paths in Stochastic Networks", IEEE Transactions on Computers, Vol. 55, No. 6, June 2006.
- [20]. محمدرضا ملاخلیلی میبیدی و محمدرضا میبیدی "یک روش جدید مبتنی بر اتوماتای یادگیر توزیع شده برای حل مساله کوتاهترین مسیر تصادفی"، گزارش فنی، آزمایشگاه سیستمهای نرمافزاری دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر، مهرماه ۱۳۸۲.
- [21] G. Ramalingam and T. Reps, "On the Computational Complexity of Dynamic Graph Problems," Theoret. Comput. Sci., Vol. 158, No. 1, pp. 233-277, 1996.
- [22] D. Frigioni, A. Marchetti-Spaccamela, and U. Nanni, "Fully Dynamic Output Bounded Single Source Shortest Path Problem", ACM-SIAM Symp. Discrete Algorithms, pp. 212-221, 1996.