

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

اصغر قربانی      محمدرضا میبدی

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 صحیح 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 قبیل مشخص است برای اولین بار توسط والر<sup>۱</sup> و پس از آن توسط والر<sup>۲</sup> و زیلیاسکوپولس<sup>۳</sup> و فن<sup>۴</sup> 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<sup>۵</sup> (SSPCL) و همچنین توزیعهای احتمالی وزن یالها از قبل شناخته شده نیست پیشنهاد می‌گردد. در الگوریتم پیشنهادی از بازی بین اتماتاها یادگیر برای حل مساله کوتاهترین مسیر بین یک گره و دیگر گره‌های 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، اتماتاها یادگیر، بازی اتماتاها 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، الگوریتم‌های متفاوتی با زمان چندجمله‌ای، نظری دایجسترا و فلوبیدوارشال 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 مجدد بايستی بررسی شود. بدین جهت الگوریتم‌هایی برای مسائل کوتاهترین مسیر در گرافهای پویا<sup>۶</sup>، 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، گراف‌هایی 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<sup>۷</sup> [3] نخستین کسی بود که مطالعه شبکه‌های تصادفی را آغاز کرد. مسأله 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 وزنهای یالهای گراف از پیش مشخص باشد، در صورتی که در بسیاری از محیط‌های واقعی توزیع احتمال یالها از پیش مشخص نمی‌باشد. برای گرافهای تصادفی در شرایطی که وزن یالها از قبیل شناخته شده نیست و یا اطلاعات کاملی درباره انها در دسترس نمی‌باشد تا کنون ۴ الگوریتم مبتنی بر اتماتاها یادگیر برای حل مساله یافتن کوتاهترین مسیر گزارش شده است. اولین الگوریتم توسط میبدی-بیگی [2] ارائه شده

<sup>1</sup> Burton

<sup>2</sup> Waller

<sup>3</sup> Ziliaskopoulos

<sup>4</sup> Fan

<sup>5</sup> Stochastic Shortest Path Problem with Correlated Link Cost

<sup>6</sup> Dynamic Single-Source Shortest Path (DSSSP)

<sup>7</sup> 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-امن ارائه شده است برای یافتن کوتاهترین مسیر بین یک مبدأ و دیگر گره های گراف در یک گراف تصادفی طراحی شده است [3]. سومین الگوریتم نیز توسط میسرا-امن ارائه شده است که میتواند برای یافتن کوتاهترین مسیر بین تمام جفت گره های گراف در یک گراف تصادفی استفاده شود. در [6] یک الگوریتم مبتنی بر بازی بین اتوماتاهای یادگیر برای یافتن کوتاهترین مسیر بین یک مبدأ و دیگر گره ها در یک گراف تصادفی ارائه شده است.

در مدل های فوق فرض بر این است هزینه يالها در گراف مستقل از یکدیگر می باشد. در بسیاری از کاربردهای واقعی هزینه يالها کاملا مستقل از هم نمی باشد و با تغییر ترافیک در یک قسمت از شبکه، ترافیک در قسمتهای مجاور نیز تغییر پیدا میکند. مسأله کوتاهترین مسیر احتمالی با يالهای همبسته برای اولین بار توسط بارتون<sup>8</sup> [15] [17] و پس از آن توسط والر<sup>9</sup> و زیلیاسکوپولس<sup>10</sup> [16] و فن<sup>11</sup> [4] مورد بررسی قرار گرفت. فن در [4] مسأله یافتن کوتاهترین مسیر در گراف های تصادفی با يالهای همبسته در شرایطی که هر يال در یکی از دو حالت ازدحام و بدون ازدحام ممکن است قرار گیرد و در عین حال توابع توزیع هزینه يالها از قبل شناخته شده باشند مطرح و الگوریتمی جهت حل آن پیشنهاد نمود. در این مقاله برای اولین بار توانی کوتاهترین مسیر گرافهای تصادفی در شرایطی که همیستگی مابین هزینه يالها وجود دارد و همچنین توزیعهای احتمالی وزن يالها از قبل شناخته شده نیست پیشنهاد میگردد. در الگوریتم پیشنهادی از بازی بین اتوماتاهای یادگیر برای اولین بار توانی کوتاهترین مسیر بین یک گره و دیگر گره های گراف استفاده میشود. برای این منظور شبکه ای از اتوماتاهای یادگیر برای حل مسأله 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 يالهای گراف تصادفی درخت کوتاهترین مسیر را برای یک گره ریشه مشخص پیدا نماید.

در ادامه مقاله، ابتدا در بخش ۲ تعریف مسأله، 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 مسأله

در این مسأله فرض می شود که یک شبکه با تعداد  $n$  گره و یک گره مقصد وجود دارد و يالها در دو حالت می توانند قرار داشته باشند؛ دارای ازدحام<sup>12</sup> و بدون ازدحام<sup>13</sup>. هدف از این مسأله یافتن استراتژی بهینه (یال بهینه) در هر گره جهت رسیدن به گره مقصد می باشد. نحوه همیستگی بین يالها توسط تابع احتمال شرطی بیان می شود. در صورتی که يالی که توسط یک متحرک طی شده است تا به گره  $i$  برسد بدون ازدحام باشد تابع چگالی احتمال زمان سفر در يال  $e_{ij}$   $p_{ij}(t)$  نشان داده می شود و در صورتی که يالی که طی شده است تا به گره  $i$  برسد در حالت ازدحام باشد تابع چگالی احتمال زمان سفر در يال  $e_{ij}$   $q_{ij}(t)$  بیان می شود. متوسط زمان سفر در يال  $e_{ij}$  در حالت بدون ازدحام برابر است با  $t_{ij}$  و در حالت ازدحام برابر است با  $\tau_{ij}$ ، که به ترتیب متوسط  $p_{ij}(t)$  و  $q_{ij}(t)$  می باشد. برای هر يال یک مقدار آستانه ازدحام  $\tau^{0ij}$  وجود دارد. در صورتی که وزن يال بیشتر از  $\tau^{0ij}$  باشد يال در حالت ازدحام خواهد بود و در غیر اینصورت يال در حالت بدون ازدحام می باشد. برای هر گره  $i$  چنانچه يالی 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 يالی 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.

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

<sup>8</sup> Burton

<sup>9</sup> Waller

<sup>10</sup> Ziliaskopoulos

<sup>11</sup> Fan

<sup>12</sup> Congested

<sup>13</sup> Un-Congested

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

$$u_N, v_N = 0$$

چنانچه فرض شود:

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

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

خواهیم داشت:

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

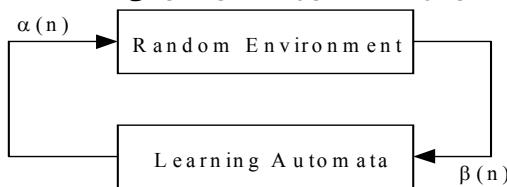
$$v_i = \min_{j \neq i} \left\{ \tau_{ij} + \lambda_{ij} u_j + (1 - \lambda_{ij}) v_j \right\} i = 1, 2, \dots, N-1 \quad (4)$$

در رابطه های (3) و (4)  $\alpha_{ij}$  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  $\lambda_{ij}$  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. اتماتاهای یادگیر و اتماتای یادگیر توزیع شده و بازی بین اتماتاهای یادگیر

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

**اتماتاهای یادگیر:** اتماتای 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 اتماتای یادگیر داده می شود. اتماتای یادگیر از این پاسخ استفاده نموده و عمل خود را برای مرحله بعد انتخاب می کند. شکل ۱ ارتباط بین اتماتای یادگیر و محیط را نشان می دهد.



شکل ۱: ارتباط بین اتماتای 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  $\beta$  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 گستته یک مقدار از مقدار محدود در فاصله  $[0, 1]$  باشد. در محیط ایستا<sup>14</sup> مقدار  $\beta(n)$  متغیر تصادفی در فاصله  $[0, 1]$  است.  $c_i$  احتمال اینکه عمل  $\alpha_i$  نتیجه نامطلوب داشته باشد می باشد. در محیط ایستا<sup>15</sup> مقدار  $c_i$  بدون تغییر می مانند، حال آنکه در محیط غیر ایستا<sup>16</sup> این مقدار در طی زمان تغییر می کنند. اتماتاهای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم بندی میگردند. در ادامه به شرح مختصری درباره اتماتای یادگیر با ساختار متغیر که در این مقاله از آنها استفاده شده است، می پردازیم.

<sup>14</sup> Stationary

<sup>15</sup> Non-Stationary

اتوماتای یادگیر با ساختار متغیر<sup>۱۶</sup> : یک اتوماتای یادگیر با ساختار متغیر توسط ۴ تائی  $\{\alpha, \beta, p, T\}$  نشان داده می شود که در آن  $p \equiv \{p_1, p_2, \dots, p_r\}$  مجموعه عملهای اتوماتا،  $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  مجموعه ورودیهای اتوماتا،  $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$  مجموعه الگوریتم یادگیری می باشد. در این نوع از اتوماتاهای اگر بردار احتمال انتخاب هر یک از عملهای  $p(n+1) = T[\alpha(n), \beta(n), p(n)]$  افزايش یافته و سایر احتمالها کاهش عمل  $\alpha_i$  در مرحله  $n$  ام انتخاب شود و پاسخ مطلوب از محیط دریافت نماید، احتمال  $p_i(n)$  افزایش یافته و سایر احتمالها افزایش می یابند. و برای پاسخ نامطلوب احتمال  $p_j(n)$  کاهش یافته و سایر احتمالها 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

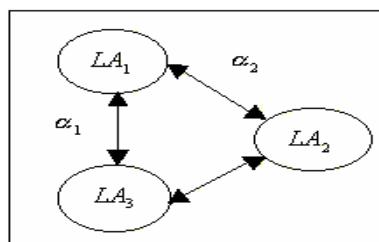
$$p_i(n+1) = p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) = (1 - a)p_j(n) \quad j \neq i \quad \forall j$$

ب- پاسخ نامطلوب

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

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

اتوماتای یادگیر توزیع شده (DLA): اتوماتای یادگیر توزیع شده شبکهای 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  $\alpha_2$  توسط  $LA_1$ ، اتوماتا یادگیر  $LA_3$  را فعال خواهد کرد. اتوماتای یادگیر فعال شده ( $LA_3$ ) بنوبهی 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  $DLA = (V, E)$  که  $V = \{LA_1, LA_2, \dots, LA_n\}$  مجموعه اتوماتای یادگیر و  $E \subset V \times V$  تعداد اتوماتاهای در  $DLA$  و  $E$  مجموعه لبههای گراف می باشد، تعريف کرد. لبه  $(i, j)$  اقدام  $j$  اتوماتا  $i$  را نشان می دهد.  $LA_j$  زمانی فعال خواهد شد که اقدام  $j$  اتوماتون  $LA_i$  انتخاب شود. تعداد اقدامهای اتوماتا  $LA_k$  ( $k = 1, 2, \dots, n$ ) برابر درجهی خروجی آن گره می باشد. برای اطلاعات بیشتر درباره DLA میتوان به مراجع [1] و [6] مراجعه کرد.



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

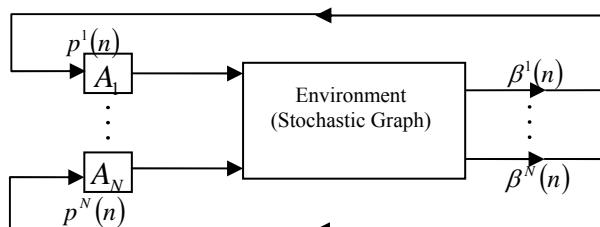
<sup>16</sup> Variable Learning Automata

<sup>17</sup> Linear Reward Penalty

<sup>18</sup> Linear Reward Epsilon Penalty

<sup>19</sup> Linear Reward Inaction

بازی بین اتوماتاهای یادگیر<sup>۲۰</sup>: 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 مینماید. 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 درنظر 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 درباره 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 ( $OLA^i$ )<sup>۲۱</sup> که طبق شکل ۴ با محیط تصادفی در ارتباط میباشد مجهر میباشد. در هر زمان با توجه به شرایط محیط فقط دو اتوماتای یادگیر از ۴ اتوماتای یادگیر در هر گره فعال خواهد 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 داردند که اتوماتای یادگیرنده همبستگی نامیده میشوند و اتوماتای یادگیر فعال دیگر وظیفه یادگیری یال بهینه را به عهده دارد که اتوماتای یادگیرنده یال بهینه نامیده میشوند. در این شبکه هر اتوماتای یادگیر با ۲ اندیس مشخص میشود. اندیس اول مشخص کننده نوع اتوماتای یادگیر(یادگیر نده 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.

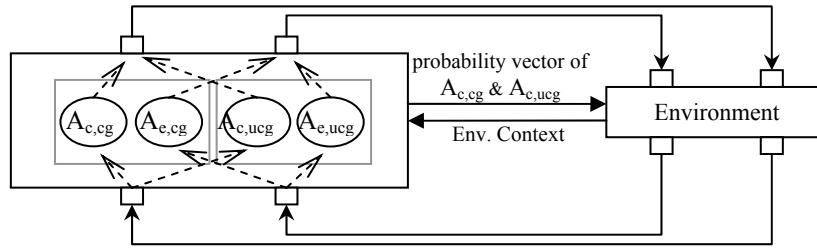
$A_{c,cg}^i$ : اتوماتای یادگیر برای یادگیری همبستگی بین یالها در شرایطی که محیط دارای ازدحام باشد.

$A_{e,cg}^i$ : 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  $OLA^i$  علاوه بر پاداش و جریمه هایی که توسط محیط به عملهای هر اتوماتای یادگیر داده میشود و ضعیت جاری محیط(ازدحام و یا بدون ازدحام) نیز میباشد که باعث فعال شدن اتوماتای یادگیر مربوط به هر حالت میشود. خروجی های  $OLA^i$  علاوه بر عملهای انتخاب شده توسط هر کدام از اتوماتاهای یادگیر، وضعیت(بردار احتمال) اتوماتای یادگیرنده همبستگی  $OLA^i$  نیز میباشد.



شکل ۴) یک QLA و نحوه رابطه آن با محیط

در ابتدای الگوریتم شبکه‌ای از  $OLA^i$  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 مینماید. اعمال انتخاب شده در محیط که همان گراف تصادفی

<sup>20</sup> Game of Learning Automata

<sup>21</sup> Quadra Learning Automata

می‌باشد اجرا و با توجه به نتیجه اعمال به انها پاداش و یا جریمه داده می‌شود. الگوریتم پیشنهادی سعی میکند با حداقل تعداد نمونه گیری از یالهای گراف تصادفی در شرایطی که 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 می‌آورد ( $\alpha_{ij}$  و  $\lambda$ ). 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 می‌شود.

مراحل الگوریتم به شرح زیر می‌باشد.

**مرحله ۱:** یک شبکه از  $QLA$  متناظر با گراف مسئله ایجاد می‌شود. تمام اتماتاهای یادگیرنده یال بهینه از الگوریتم یادگیری  $L_{R-I}$  و تمام اتماتاهای یادگیرنده همبستگی از الگوریتم یادگیری  $L_{R-P}$  استفاده میکنند. نرخ یادگیری دو اتماتای یادگیرنده یال بهینه یکسان در نظر گرفته می‌شود و همچنین ضرایب جریمه و پاداش در دو اتماتای یادگیرنده همبستگی نیز یکسان می‌باشد. اقدامهای هر اتماتای یادگیرنده یال بهینه، یالهای خروجی از گره متناظر با  $QLA$  می‌باشد؛ در گره  $i$  اقدامهای اتماتای یادگیرنده یال بهینه در شرایط ازدحام  $\alpha_{e,cg}^i = \{\alpha_{(e,cg)_1}^i, \alpha_{(e,cg)_2}^i, \dots, \alpha_{(e,cg)_{r_i}}^i\}$  و اقدامهای اتماتای یادگیرنده یال بهینه در شرایط بدون ازدحام  $\alpha_{e,ucg}^i = \{\alpha_{(e,ucg)_1}^i, \alpha_{(e,ucg)_2}^i, \dots, \alpha_{(e,ucg)_{r_i}}^i\}$  میباشد. بردارهای احتمال اتماتاهای یادگیرنده یال بهینه برای شرایط ازدحام و عدم ازدحام به ترتیب  $p_{e,ucg}^i = (p_{(e,ucg)_1}^i, p_{(e,ucg)_2}^i, \dots, p_{(e,ucg)_{r_i}}^i)$  و  $p_{e,cg}^i = (p_{(e,cg)_1}^i, p_{(e,cg)_2}^i, \dots, p_{(e,cg)_{r_i}}^i)$  میباشند. مقدار اولیه هر کدام از اعضای این دو بردار برابر با  $r_i/1$  می‌باشد که تعداد یالهای خروجی از گره  $i$  است. عملهای مجاز هر یک از ۲ اتماتای یادگیرنده همبستگی "ازدحام" ( $cg$ ) و "بدون ازدحام" ( $ucg$ ) می‌باشد؛ یعنی  $\{cg, ucg\}$  و  $\{c, cg\}$  و  $\{c, ucg\}$ . برای هر یک از گره‌ها دو مقدار آستانه پویا در نظر گرفته می‌شود،  $t_{cg}^i$  و  $t_{ucg}^i$  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 می‌شود.

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

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

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

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

که  $k_{cg}^i - 1$  و  $k_{ucg}^i - 1$  به ترتیب برابر است با تعداد دفعاتی که  $t_{cg}^i$  و  $t_{ucg}^i$  تا حال 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  $1 -$  مقدار دهی اولیه می‌شود.

**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 اتماتاهای مربوط به آن حالت فعال شده و یکی از عمل های ممکن خود را انتخاب می‌کند. به عبارتی در هر  $QLA$  دو اتماتاً فعال می‌شود. اتماتای یادگیرنده همبستگی یکی از دو عمل ممکن  $cg$  و  $ucg$  را انتخاب می‌کند و اتماتای 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 اتماتاهای 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 اتماتای فعال در  $QLA$  گره  $i$  بصورت زیر می‌باشد:

**اتوماتای یادگیرنده همبستگی:** اگر مقدار  $(w(e_{ij})\tau^{0ij})$  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 اتماتای یادگیرنده همبستگی پاداش یا جریمه تعلق می‌گیرد، چنانچه عمل "ازدحام" توسط اتماتای 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 اتماتای 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 یا ل بھینه:** 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 آستانه 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 آستانه پویا باشد و یا مقدار استانه گره جاری ۱ - باشد به عمل انتخاب شده توسط اتماتای یادگیر پاداش و در غیر این صورت جریمه داده می‌شود. سپس مقدار آستانه بروز رسانی می‌شود. برای بروز کردن مقدار آستانه 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.

شبیه 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 $_o^i$  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 $_c^i$  =  $A_{c,ucg}^i$  ;
        end if
        If node  $i$  is not congested then
            ActiveLA $_c^i$  =  $A_{c,ucg}^i$ 
            ActiveLA $_c^i$  =  $A_{c,ucg}^i$ 
        end if
        state  $\leftarrow$  ActiveLA $_c^i$  selects an action based on its probability vector.
        edge  $\leftarrow$  ActiveLA $_c^i$  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 LRP learning Algorithm;
        else
            penalize the action selected by the ActiveLA $_c^i$  according to the LRP 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 $_c^i$  according to the LRI 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) شبیه کد الگوریتم پیشنهادی برای حل مسئله SSPCL

## ۵. نتایج شبیه‌سازی ها

برای شبیه سازی و مقایسه عملکرده 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 یکتابع توزیع احتمال گستته با مقادیر مثبت استفاده شده است. یالهای بھینه 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 پیکارده که در [۵][۶] ارائه شده است، براساس میانگین طول یالها محاسبه میشود.

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

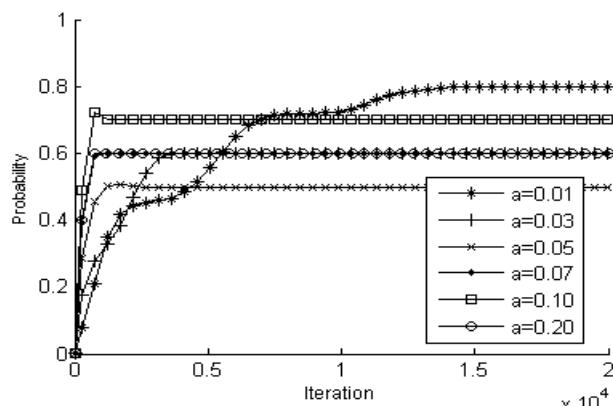
- تعداد نمونه هایی که توسط الگوریتم از یالهای گراف گرفته شده است. با پیشرفت الگوریتم نمونه گیری بر روی یالهایی متوجه میشود که جز یال بهینه در هر گره می باشند و نمونه گیریها از یالهایی که جز یالهایی بهینه نیستند کاهش میابد.
- احتمال انتخاب یالهایی بهینه که حاصل ضرب احتمالهای انتخاب یالهایی بهینه توسط اتماتاهای 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 تکراهای 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 یالهایی بهینه (SPS) نشان داده می شود.

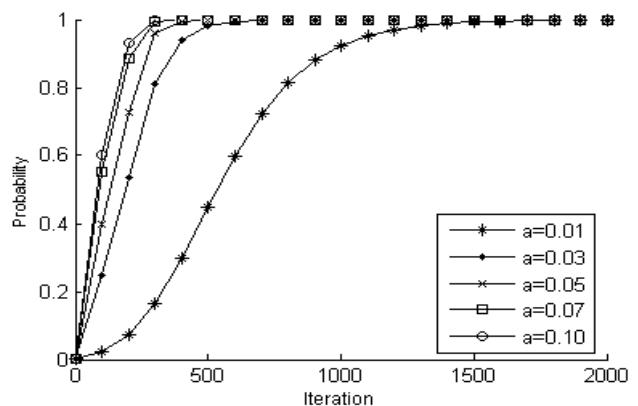
**الگوریتم ۱:** ضریب یادگیری اتماتاهای در تمام  $QLA$  ها ثابت در نظر گرفته می شود و در طول اجرای الگوریتم تغییر نمیکند. در هر مرحله مقادیر آستانه در هر  $QLA$  مطابق با رابطه (۵) و (۶) بروز میشود. نتایج اجرای این الگوریتم که حاصل میانگین ۱۰۰ مرتبه اجرا بر روی گرافهای ۱ و ۲ می باشد در جدول زیر نشان داده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

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

| $a_{p,\psi}$ | Graph 1 |     |        |        | Graph 2 |    |         |         |
|--------------|---------|-----|--------|--------|---------|----|---------|---------|
|              | 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 مسأله SSPCL در گراف ۱



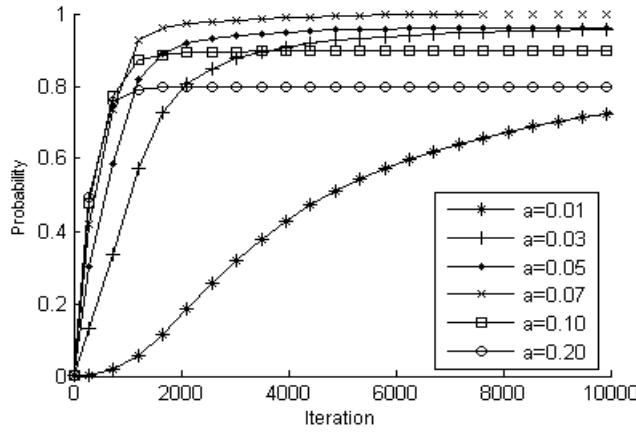
نمودار ۱) همگرایی الگوریتم ۱ برای مسأله SSPCL در گراف ۱

**الگوریتم ۲:** همان الگوریتم ۱ است که در آن مقدار  $t^i$  زمانی بروز میشود که اتماتای یادگیرنده یال بهینه  $i$  پاداش دریافت کرده باشد. نتایج اجرای این الگوریتم که حاصل میانگین ۱۰۰ مرتبه اجرا بر روی گرافهای ۱ و ۲ می باشد در جدول زیر نشان داده شده است.

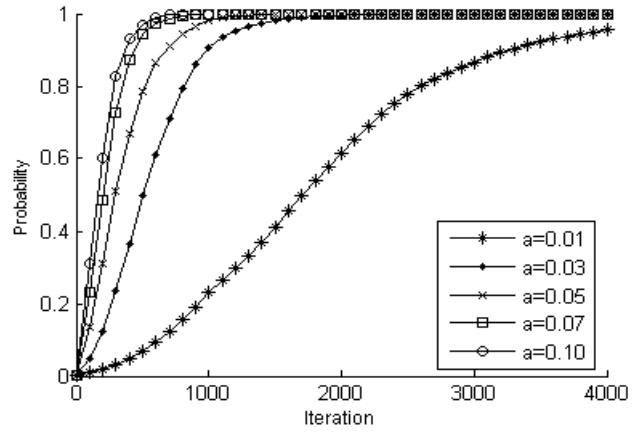
جدول ۲) میانگین تعداد تکرارها، تکرارهای همگرا شده، تعداد کل نمونه گیری ها و تعداد نمونه های گرفته شده از یالهایی بهینه در الگوریتم ۲

| $a_{p,\psi}$ | Graph 1 |     |        |        | Graph2 |     |         |         |
|--------------|---------|-----|--------|--------|--------|-----|---------|---------|
|              | 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 ۲) همگرایی الگوریتم ۲ برای مسئله SSPCL در گراف ۲

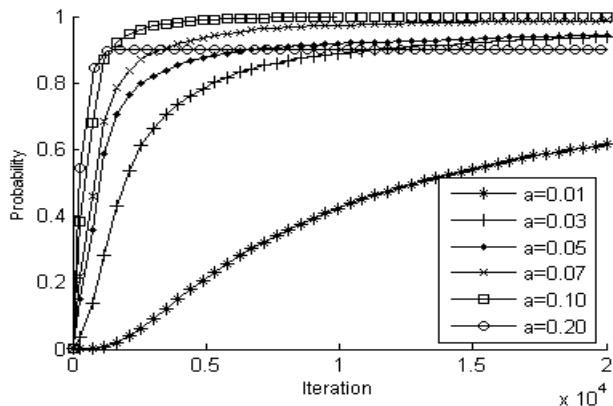


نمودار ۳) همگرایی الگوریتم ۲ برای مسئله SSPCL در گراف ۱

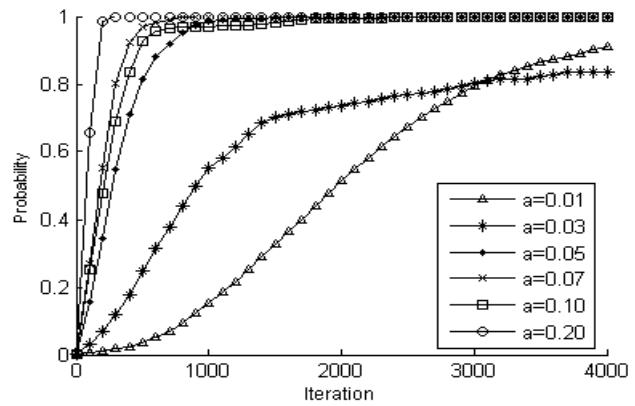
الگوریتم ۳ : الگوریتم ۳ همان الگوریتم ۲ است که ضریب یادگیری برای دو اتوماتای یادگیرنده یال بهینه در گره  $i$  برابر با  $a/dist[i]$  میباشد که  $dist[i]$  حداقل تعداد یالهایی است که برای رسیدن به گره  $i$  از مبدا بایستی طی شود.  $[dist[i]]$  در ابتدای الگوریتم به تعداد گره های گراف مقدار دهی میشود. در هر مرحله چنانچه برای یال  $e_{ij}$  که توسط یکی از اتوماتاهای یادگیر گره  $i$  انتخاب شده است، مقدار  $dist[i] + 1$  بیشتر باشد مقدار  $dist[j] + 1$   $dist[i] + 1$  به  $dist[j]$  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 با 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 مسأله SSPCL در گراف ۲

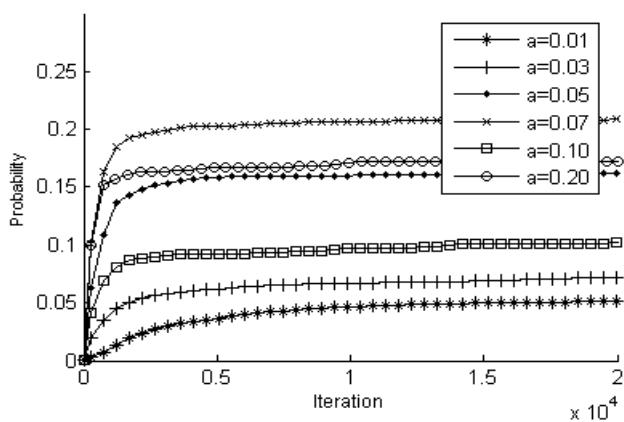


نمودار ۵) همگرایی الگوریتم ۳ برای مسأله SSPCL در گراف ۱

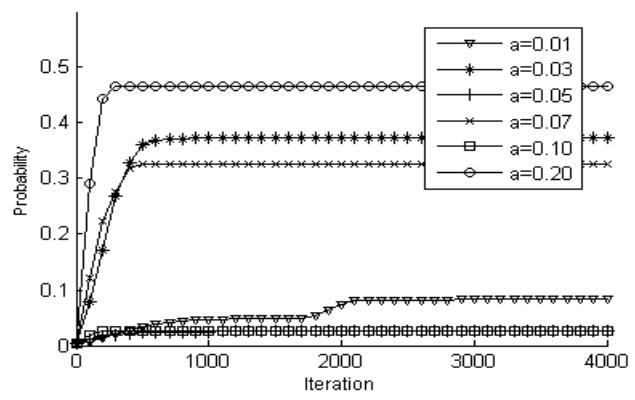
الگوریتم ۴: این الگوریتم همانند الگوریتم ۱ است با این تفاوت که در آن برای محاسبه مقدار استانه، از متوسط نمونه های یال  $e_{ij}$  که تا حال گرفته شده است به جای نمونه گرفته شده فلی استفاده می شود. آزمایشها نشان داده است که استفاده از تمامی نمونه های گرفته شده از یک یال برای محاسبه استانه منجر به نتایج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

جدول ۴) میانگین تعداد تکرار ها، 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 مسأله SSPCL در گراف ۲



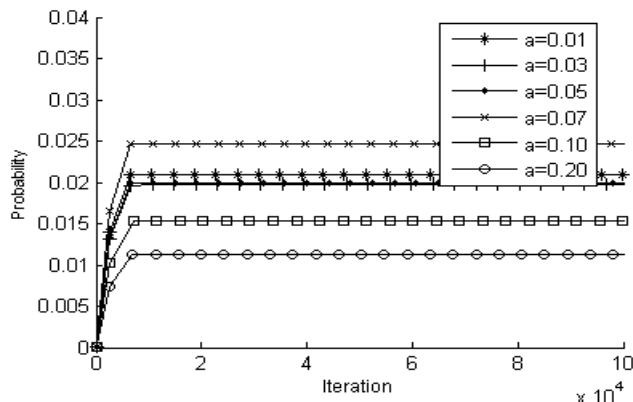
نمودار ۷) همگرایی الگوریتم ۴ برای مسأله SSPCL در گراف ۱

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

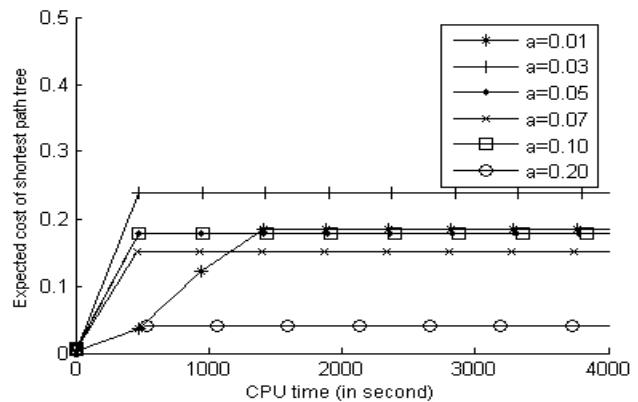
زیر نشان داده شده است. منظور از همگرا شدن یک اتوماتای یادگیر، رسیدن احتمال انتخاب یکی از اعمالش از یک مقدار از پیش تعیین شده بیشتر گردد. این مقدار برای ازمایش‌های که نتایج آنها در جدول ۵ آمده ۰,۸۵ در نظر گرفته شده است.

جدول ۵) میانگین تعداد تکرارها، تکرارهای همگرا شده، تعداد کل نمونه گیری‌ها و تعداد نمونه‌های گرفته شده از یالهای بهینه در الگوریتم ۵

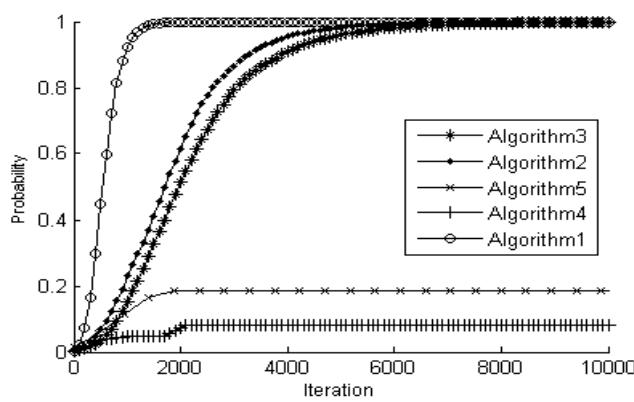
| $a_{p,\psi}$ | Graph 1 |     |        |        | Graph 2 |     |         |        |
|--------------|---------|-----|--------|--------|---------|-----|---------|--------|
|              | AVI     | PC  | TS     | SPS    | AVI     | PC  | TS      | SPS    |
| 0.005        | 2942.5  | 100 | 6959.8 | 4085.3 | 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  |



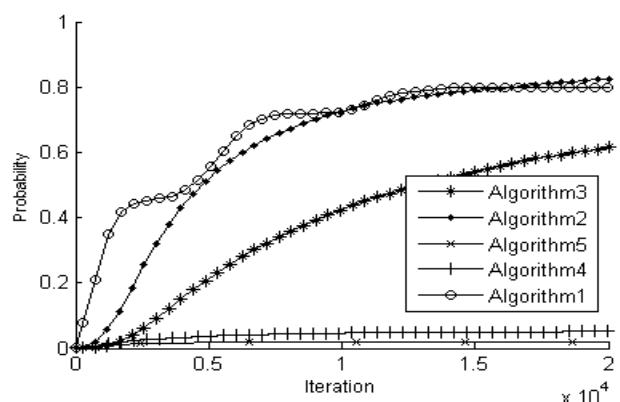
نمودار ۱۰) همگرایی الگوریتم ۵ برای مسئله SSPCL در گراف ۲



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 می‌یابد ولی در مقابل درصد اجراها همگرا شده به جواب بهینه نیز کاهش میابد. همچنین با توجه به نتایج ارائه شده در ستونهای TS و SPS مشاهده می‌شود که تعداد نمونه گیری‌ها از یالهای جواب بهینه بیشتر از دیگر یالها در گراف ۱ ارائه شده است. همانطور که در نمودار ۱۱ و نمودار ۱۲ مشخص است الگوریتم ۱ از دیگر الگوریتمهای ارایه شده دارای نرخ همگرایی بالاتر 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 الگوریتم‌های ۲ و ۳ نسبت به دیگر الگوریتم‌ها از دقت بالاتری برخوردار هستند ولی دارای سرعت همگرایی پایین‌تری میباشند. 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 براى این منظور شبکه ای از اتماتاهای 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 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 Automat", 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 میبدی، " حل مساله کوتاهترین مسیر بین یک گره و دیگر گره ها در گرافهای تصادفی با استفاده از بازی بین اتماتاهای یادگیر"، گزارش فنی، آزمایشگاه سیستم های نرم افزاری 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. Lakshminarayanan, Learning Algorithms: Theory and Applications. New York: Springer-verlag, 1981.
- [10] M. R. Meybodi and S. Lakshminarayanan, "On a Class of Learning Algorithms which have Symmetric Behavior under Success and Failure", 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 میبدی و محمدرضا میبدی "یک روش جدید مبتنی بر اتماتای یادگیر توزیع شده برای حل مسأله کوتاهترین مسیر تصادفی" ، گزارش فنی، آزمایشگاه سیستم های نرم افزاری 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.