

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

اصغر قربانی

آزمایشگاه سیستمهای نرم افزاری، دانشکده مهندسی کامپیوتر و فناوری  
اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران

aghorbani@aut.ac.ir

محمد رضا میبدی

آزمایشگاه سیستمهای نرم افزاری، دانشکده مهندسی کامپیوتر و فناوری  
اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران

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 (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.

**واژه های 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، الگوریتمهای متفاوتی با زمان چندجمله ای، نظیر دایجسترا و فلویدارشال 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 الگوریتمهایی برای مسائل کوتاهترین مسیر در گرافهای پویا<sup>۱</sup>، گرافهایی که وزن یالها و یا ساختار گراف در طی زمان بطور پویا تغییر کنند، مطرح شده اند [1-4]. این الگوریتمها نیز برای گرافهایی

تصادفی، گراف هایی که وزن یالها متغییرهای تصادفی باشند، از کارایی بالایی برخوردار نیستند. بهمین دلیل برای حل مساله کوتاهترین مسیر در گرافهایی تصادفی الگوریتمهایی طراحی شده اند. فرانک<sup>۲</sup> [5] نخستین کسی بود که مطالعه شبکه‌های تصادفی را آغاز کرد. 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 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 میبدی-یگی [6] ارائه شده است که برای حل مساله یافتن کوتاهترین مسیر بین دو گره در گرافهای تصادفی مورد استفاده قرار گرفته است. نسخه های اصلاح شده این الگوریتم در [7-9] 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 [10]. سومین الگوریتم نیز توسط میسرا-اومن ارائه شده است که میتواند برای یافتن کوتاهترین مسیر بین تمام جفت گره‌های گراف در یک گراف تصادفی استفاده شود [11]. در [۱۲] یک الگوریتم مبتنی بر بازی بین اتوماتاهای یادگیر برای یافتن کوتاهترین مسیر بین یک مبدأ و دیگرگره‌ها در یک گراف تصادفی ارائه شده است.

در مدل های فوق فرض بر این است هزینه یالها در گراف مستقل از یکدیگر می‌باشد. در بسیاری از کاربردهای واقعی هزینه یالها کاملاً مستقل از هم نمی‌باشد 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 بارتون<sup>۳</sup> [13,14] و پس از آن توسط والر<sup>۴</sup> و زیلیاسکوپولس<sup>۵</sup> [15] و فن<sup>۶</sup> [16,17] مورد بررسی قرار گرفت. فن در [16] مساله 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

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

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

$$u_N, v_N = 0$$

چنانچه فرض شود:

$$I_{ij} = \int_0^{t_{0ij}} q_{ij}(t) dt \quad , \quad a_{ij} = \int_0^{t_{0ij}} p_{ij}(t) dt$$

$$t_{ij} = \int_0^{\infty} t q_{ij}(t) dt \quad , \quad t_{ij} = \int_0^{\infty} t p_{ij}(t) dt$$

خواهیم داشت :

$$u_i = \min_{j \neq i} \left\{ t_{ij} + a_{ij} u_j + (1 - a_{ij}) v_j \right\} \quad (3)$$

$$v_i = \min_{j \neq i} \left\{ t_{ij} + I_{ij} u_j + (1 - I_{ij}) v_j \right\} \quad (4)$$

$$, i = 1, 2, \dots, N - 1$$

در رابطه های (۳) و (۴)  $a_{ij}$  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  $I_{ij}$  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 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 [16] و [17] مراجعه کنید.

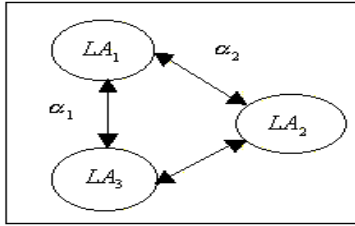
### ۳- اتوماتاهای یادگیر، اتوماتای یادگیر توزیع شده و بازی بین اتوماتاهای یادگیر

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.

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 فعال شده و سپس هر یک از آنها یکی از اعمال خود را انتخاب می نمایند. اعمال انتخاب شده در محیط اجرا و با توجه به نتیجه اعمال به آنها پاداش و یا جریمه داده می شود. در ادامه مقاله، 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 می باشد.

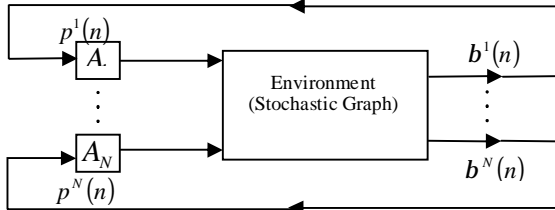
### ۲- تعریف مساله

در این مساله 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<sup>۱۱</sup> و بدون ازدحام<sup>۱۲</sup>. 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  $t_{ij}$ ، که به ترتیب متوسط  $p_{ij}(t)$  و  $q_{ij}(t)$  می باشد. برای هر یال یک مقدار آستانه ازدحام  $t^{0ij}$  وجود دارد. در صورتی که وزن یال بیشتر از  $t^{0ij}$  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 [16] آمده است.



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

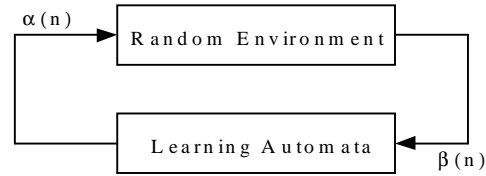
بازی بین اتوماتاهای یادگیر<sup>۴</sup>: در بازی بین اتوماتاهای یادگیر که در این مقاله استفاده شده است شبکه ای از اتوماتاهای یادگیر برای حل مساله همکاری مینمایند. در این شبکه در هر مرحله تمامی اتوماتاهای یادگیر فعال شده و سپس هر یک از آنها یکی از اعمال خود را انتخاب مینمایند. اعمال انتخاب شده در محیط 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 می باشد نشان داده شده است. برای اطلاعات بیشتر در باره بازیهای بین اتوماتاهای یادگیر میتوان به [21] مراجعه کرد.



شکل (۳): بازی بین اتوماتاهای یادگیر

#### ۴- الگوریتم پیشنهادی

برای حل این مسئله هر گره شبکه به یک گروه ۴ اتوماتای یادگیر (QLA)<sup>۵</sup> که طبق شکل (۴) با محیط تصادفی در ارتباط می باشد 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 زمانی که 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 می باشد (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 می باشد.



شکل (۱): ارتباط بین اتوماتای یادگیر و محیط

محیط را می توان توسط سه تایی  $E \equiv \{a, b, c\}$  نشان داد که در آن  $a$  مجموعه ورودیها،  $b$  مجموعه خروجیها و  $c$  مجموعه احتمالات 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. اتوماتاهای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم بندی می گردند. یک اتوماتای یادگیر با ساختار متغیر توسط ۴ تایی  $\{a, b, p, T\}$  نشان داده می شود که در آن  $a \equiv \{a_1, a_2, \mathbf{L}, a_r\}$  مجموعه عملهای اتوماتا،  $b \equiv \{b_1, b_2, \mathbf{L}, b_m\}$  مجموعه ورودیهای اتوماتا،  $p \equiv \{p_1, p_2, \mathbf{L}, p_r\}$  بردار احتمال انتخاب هر یک از عملها، و  $p(n+1) = \Pi[\alpha(n), \beta(n), p(n)]$  الگوریتم یادگیری می باشد. الگوریتم زیر یک نمونه از الگوریتمهای یادگیری خطی برای اتوماتای یادگیر با ساختار متغیر می باشد.

الف- پاسخ مطلوب  $p_i(n+1) = p_i(n) + a[1 - p_i(n)]$

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

ب- پاسخ نامطلوب

$$p_i(n+1) = (1-b)p_i(n) \quad \forall j \quad j \neq i$$

$$p_j(n+1) = \frac{b}{r-1} + (1-b)p_j(n)$$

در روابط فوق، پارامتر پاداش و a پارامتر پاداش و b پارامتر جریمه می باشد. برای مطالعه بیشتر در باره اتوماتاهای یادگیر می توان به [18-22] 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 [9]. 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  $a_2$  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 می باشد. برای اطلاعات بیشتر در باره DLA میتوان به مراجع [8] و [9] مراجعه کرد.

یادگیرنده همبستگی "ازدحام" ( $cg$ ) و "بدون ازدحام" ( $ucg$ ) می باشد؛ یعنی  $\alpha_{c,cg} = \{cg, ucg\}$  و  $\alpha_{c,ucg} = \{cg, ucg\}$  و  $P_{c,ucg}^i = (P_{(c,ucg)cg}^i, P_{(c,ucg)ucg}^i)$  و  $P_{c,cg}^i = (P_{(c,cg)cg}^i, P_{(c,cg)ucg}^i)$  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 (7) و (8) 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  $t_{ucg}^i$  و  $t_{cg}^i$  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 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 اتوماتاهای مربوط به آن حالت فعال شده و یکی از عمل‌های ممکن خود را انتخاب می‌کند. به عبارتی در هر  $QLA$  دو اتوماتا فعال می‌شود. اتوماتای یادگیرنده همبستگی یکی از دو عمل ممکن  $cg$  و  $ucg$  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 یادگیرنده، 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:

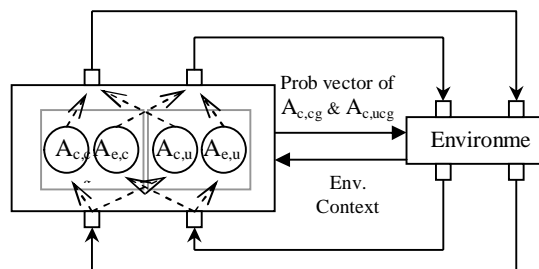
اتوماتای یادگیرنده همبستگی: اگر مقدار  $w(e_{ij})$  از آستانه ازدحام ( $t_{0ij}$ ) بیشتر باشد، یال در حالت ازدحام و در غیر این صورت در حالت بدون ازدحام می‌باشد. با توجه به حالت یال انتخاب شده (دارای ازدحام یا بدون ازدحام) به اتوماتای یادگیرنده همبستگی پاداش یا جریمه تعلق می‌گیرد، چنانچه عمل "ازدحام"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

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

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

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

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



شکل (۴) یک  $QLA$  و نحوه رابطه آن با محیط

ورودی های  $QLA$  علاوه بر پاداش و جریمه‌هایی که توسط محیط به عملهای هر اتوماتای یادگیرنده داده می‌شود و وضعیت جاری محیط (ازدحام و یا بدون ازدحام) نیز می‌باشد که باعث فعال شدن اتوماتای یادگیرنده مربوط به هر حالت می‌شود. خروجی های  $QLA$  علاوه بر عملهای انتخاب شده توسط هر کدام از اتوماتاهای یادگیرنده، وضعیت (بردار احتمال) اتوماتای یادگیرنده همبستگی  $QLA$  نیز می‌باشد. مراحل الگوریتم به شرح زیر می‌باشد.

**مرحله ۱:** یک شبکه از  $QLA$  متناظر با گراف مسأله ایجاد می‌شود. تمام اتوماتاهای یادگیرنده یال بهینه از الگوریتم یادگیری  $L_R-1$  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 یادگیرنده یال بهینه در شرایط بدون ازدحام  $a_{e,cg}^i = \{a_{(e,cg)_1}^i, a_{(e,cg)_2}^i, \dots, a_{(e,cg)_r_i}^i\}$  و اقدامهای اتوماتای یادگیرنده یال بهینه در شرایط بدون ازدحام  $a_{e,ucg}^i = \{a_{(e,ucg)_1}^i, a_{(e,ucg)_2}^i, \dots, a_{(e,ucg)_r_i}^i\}$  می‌باشد. بردارهای احتمال اتوماتاهای یادگیرنده یال بهینه برای شرایط ازدحام و عدم ازدحام به ترتیب  $P_{e,cg}^i = (P_{(e,cg)_1}^i, P_{(e,cg)_2}^i, \dots, P_{(e,cg)_r_i}^i)$  و  $P_{e,ucg}^i = (P_{(e,ucg)_1}^i, P_{(e,ucg)_2}^i, \dots, P_{(e,ucg)_r_i}^i)$  می‌باشند. مقدار اولیه هر کدام از اعضای این دو بردار برابر با  $1/r_i$  می‌باشد که  $r_i$  تعداد یالهای خروجی از گره  $i$  است. عملهای مجاز هر یک از ۲ اتوماتای

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 رابطه (۳) و (۴) مشخص می شود که با استفاده از روش تخمین پیکارد [16,17]، براساس میانگین طول یالها محاسبه می شود. در این بخش برای بررسی عملکرد الگوریتم های پیشنهادی معیارهای زیر استفاده می شود.

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 گیرها از یالهایی که جز یالهای بهینه نیستند کاهش می یابد.

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.

جریمه داده می شود. 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 می شود.

اتوماتای یادگیرنده یال بهینه: اگر مقدار آستانه گره سمت دیگر یال انتخاب شده ۱- نباشد، با توجه به حالت گره یکی از مقادیر رابطه (5) و (6) محاسبه می شود. اگر گره در حالت بدون ازدحام باشد رابطه (6) و در غیر این صورت رابطه (1) محاسبه می شود. سپس این مقدار با مقدار آستانه مربوط به حالت جاری گره مقایسه می شود. یعنی اگر گره در حالت ازدحام باشد با  $t_{cg}$  و اگر گره در حالت بدون ازدحام باشد با  $t_{ucg}$  مقایسه می شود. چنانچه کمتر و یا برابر با آستانه پویا باشد و یا مقدار آستانه گره جاری ۱- باشد به عمل انتخاب شده توسط اتوماتای یادگیر پاداش و در غیر این صورت جریمه داده می شود. سپس مقدار آستانه بروز رسانی می شود. برای بروز کردن مقدار آستانه می توان از روشهای مختلفی استفاده نمود که بعداً به آنها اشاره خواهد شد.

**مرحله ۴:** اگر شرط خاتمه الگوریتم برقرار باشد، اجرای الگوریتم متوقف می شود در غیر اینصورت کنترل به مرحله ۲ انتقال پیدا می کند. شبه کد الگوریتم پیشنهادی در شکل (۵) آمده است.

#### ۵- نتایج شبیه سازی ها

برای شبیه سازی و مقایسه عملکرد میان الگوریتم های مختلف از ۲ شبکه که در [16] معرفی شده استفاده می شود. برای طول یالها در هر

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  $L_{Rl}$  learning Algorithm;

**end if**

**if (state = congested)**

Compute  $t_{state}^i$  using equation 7

**end if**

```

if (state = uncongested)
  Compute  $t'_{state}$  using equation 8
end if
end for
until algorithm converges
end GA- SSPCL
    
```

شکل (۵) شبه کد الگوریتم پیشنهادی برای حل مسأله SSPCL

83.1	121.0	86	42.4	0.200
Graph 2				
SPS	TS	PC	AVI	$a_{p,y}$
12017.9	17790.2	100	2232.6	0.005
5495.4	8060.9	100	1015.9	0.010
1458.6	2150.1	100	277.0	0.030
740.8	1104.7	100	146.8	0.050
528.7	781.1	100	106.2	0.070
381.2	553.6	99	77.9	0.100
249.6	333.9	73	50.1	0.200

**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. لازم به ذکر است که اتوماتاهای یادگیرنده مربوط به یادگیری همبستگی در تمام  $QLA$  دارای مقدار ضریب یادگیری ثابت می‌باشند زیرا همبستگی بین یالها به دور یا نزدیک بودن به مقصد ارتباطی ندارد. نتایج اجرای این الگوریتم که حاصل میانگین ۱۰۰ مرتبه اجرا بروی گرافهای ۱ و ۲ می‌باشد در جدول (۳) نشان داده شده است.

جدول (۳) نتایج شبیه سازی در الگوریتم ۳

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

**الگوریتم ۴:** این الگوریتم مانند الگوریتم ۱ است با این تفاوت که در آن برای محاسبه مقدار استانه، از متوسط نمونه های یال  $e_{ij}$  که

**الگوریتم ۱:** ضریب یادگیری اتوماتاها در تمام  $QLA$  ها ثابت در نظر گرفته می‌شود و در طول اجرای الگوریتم تغییر نمی‌کند. در هر مرحله مقادیر استانه در هر  $QLA$  مطابق با رابطه (5) و (6) بروز میشود. نتایج اجرای این الگوریتم که حاصل میانگین ۱۰۰ مرتبه اجرا بروی گرافهای ۱ و ۲ می‌باشد در جدول (۱) نشان داده شده است.

جدول (۱) نتایج شبیه سازی در الگوریتم ۱

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

**الگوریتم ۲:** الگوریتم ۲ همان الگوریتم ۱ است که در آن مقدار  $t_y^i$  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 می‌باشد در جدول (۲) نشان داده شده است.

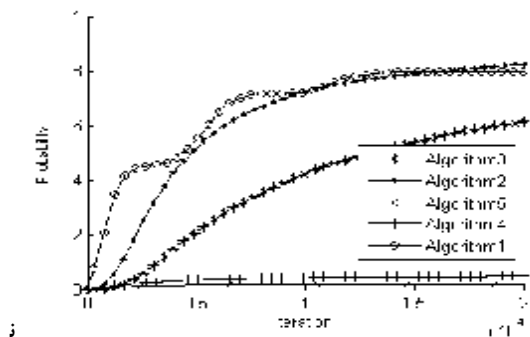
جدول (۲) نتایج شبیه سازی در الگوریتم ۲

Graph 1				
SPS	TS	PC	AVI	$a_{p,y}$
5623.3	8768.7	100	2929.2	0.005
2624.3	4003.9	100	1340.7	0.010
797.4	1204.7	100	407.0	0.030
425.9	636.5	100	217.9	0.050
285.7	423.7	99	147.2	0.070
176.6	261.9	98	93.3	0.100

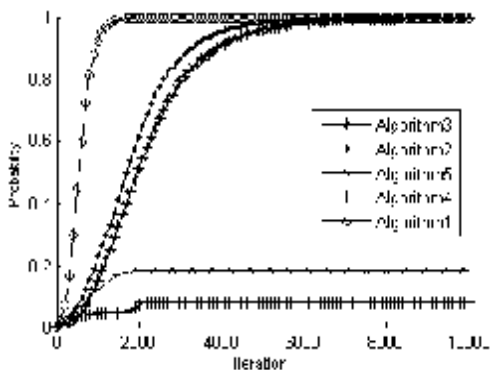
256.9	403.4	84	75.0	0.100
133.2	196.0	46	41.0	0.200

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 می باشد.

همانطور که در نمودار (۱) و (۲) مشخص است الگوریتم ۱ از دیگر الگوریتمهای ارائه شده دارای نرخ همگرایی بالاتری می باشد. 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 پایین تری می باشند.



مودار (۱) همگرایی الگوریتمهای مختلف در گراف ۲



نمودار (۲) همگرایی الگوریتمهای مختلف در گراف ۱

### 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 شبکههای اتوماتاهای

تا بحال گرفته شده است به جای نمونه گرفته شده فعلی استفاده می شود. نتایج در جدول (۴) آمده است.

جدول (۴) نتایج شبیه سازی در الگوریتم ۴

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

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

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

- [13] Burton, D., *On the Inverse Shortest Path Problem*, Ph.D., Faculties University of Notre-Dame De La Paix De Namur. 1993.
- [14] Burton, D., "On the Use of an Inverse Shortest Paths Problem for Recovering Linearly Correlated Cost," *Mathematical Programming*, vol. 63, pp. 1-22, 1994.
- [15] Waller, S. T., Ziliaskopoulos, A. K., "On the Online Shortest Path Problem with Limited Arc Cost Dependencies," *Networks*, vol. 40, pp. 216-227, 2002.
- [16] Fan, Y., *Optimal Routing Through Stochastic Networks*, Ph.D., SOUTHERN CALIFORNIA. 2003.
- [17] Fan, Y., Kalaba, R., Moore, J., "Shortest Paths in Stochastic Networks with Correlated Link Costs," *Computers and Mathematics with Applications*, vol. 49, pp. 1549-1564, 2005.
- [18] Lakshminarayanan, S., *Learning Algorithms: Theory and Applications*. New York: Springer-Verlag, 1981.
- [19] Meybodi, M. R., Lakshminarayanan, S., "On a Class of Learning Algorithms which have Symmetric Behavior under Success and Failure," in *Lecture Notes in Statistics*: Springer-Verlag, 1984, pp. 145-155.
- [20] Mars, P., Chen, J. R., Nambir, R., *Learning Algorithms: Theory and Applications in Signal Processing, Control, and Communication*: CRC Press Inc., 1996.
- [21] Narendra, S., Thathachar, K. S., *Thathachar, Learning Automata: An Introduction*. New York: Prentice-Hall, 1989.
- [22] Baba, N., "New Topics in Learning Automata Theory and Applications," in *Lecture Notes in Control and Information Sciences*. vol. 71 Berlin: Springer-Verlag, 1985.

یادگیر که متناظر با گراف مساله می‌باشد برای حل مساله همکاری مینماید. 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] Mcquillan, J., Richer, I., Rosen, E., "The New Routing Algorithm for the ARPANET," *IEEE Transaction and Communication*, vol. COM-28, pp. 711-719, 1980.
- [2] Spira, P., Pan, A., "On Finding and Updating Spanning Trees and Shortest Paths," *SIAM Journal Computation*, vol. 4, pp. 375-380, 1975.
- [3] Ramalingam, G., Reps, T., "On the Computational Complexity of Dynamic Graph Problems," *Theoretical Computer Science*, vol. 158, pp. 233-277, 1996.
- [4] Frigioni, D., Marchetti-Spaccamela, A., Nanni, U., "Fully Dynamic Output Bounded Single Source Shortest Path Problem," in *Symposium on Discrete Algorithms*, 1996.
- [5] Frank, H., "Shortest Paths in Probabilistic Graphs," *Operations Research*, vol. 17, pp. 583-599, 1969.
- [6] Meybodi, M. R., Beigy, H., "Solving Stochastic Shortest Path Problem Using Distributed Learning Automata," in *Proceedings of CSICC*, Isfahan, Iran, 2001, pp. 70-86.

[۷] 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.

- [8] Beigy, H., Meybodi, M. R., "Utilizing Distributed Learning Automata to Solve Stochastic Shortest Path Problem" *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 14, pp. 591-615, 2006.
- [9] Beigy, H., Meybodi, M. R., "A New Distributed Learning Automata for Solving Stochastic Shortest Path Problem," in *International Joint Conference on Information Science*, Durham, USA, 2002, pp. 339-343.
- [10] Misra, S., 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, pp. 1179-1192, 2005.
- [11] Misra, S., Oommen, B. J., "An Efficient Dynamic Algorithm for Maintaining All-Pairs Shortest Paths in Stochastic Networks," *IEEE Transactions on Computers*, vol. 55, 2006.

[۱۲] اصغر قربانی و محمدرضا میبیدی، "حل مساله کوتاهترین مسیر بین یک گره و دیگر گره‌ها در گرافهای تصادفی با استفاده از بازی بین اتوماتاهای یادگیر"، گزارش فنی، آزمایشگاه سیستمهای نرم‌افزاری دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر، مرداد ماه ۱۳۸۶.

## زیرنویس‌ها

- <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
- <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
- <sup>14</sup> Game of Learning Automata
- <sup>15</sup> Quadra Learning Automata