

یک الگوریتم مبتنی بر آتاماتای یادگیر توزیع شده جدید

برای حل مساله کوتاهترین مسیر تصادفی

محمد رضا میبدی

meybodi@ce.aut.ac.ir

محمد رضا ملاخلیلی میدی

mrmollakhalili@yahoo.com

چکیده: در این مقاله یک الگوریتم جدید مبتنی بر آتاماتای یادگیر توزیع شده برای حل مساله کوتاهترین مسیر تصادفی ارایه می‌گردد. در این الگوریتم یک معیار جدید برای تعیین پاداش و یا جریمه برای عمل‌های انتخابی توسط یک آتاماتای یادگیر توزیع شده معرفی می‌شود. نتایج شیوه سازی‌ها الگوریتم جدید نشان می‌دهد که این معیار جدید در مقایسه با روش‌های دیگری که تاکنون برای حل مساله کوتاهترین مسیر تصادفی توسط آتاماتای یادگیر توزیع شده به کار گرفته شده است، از نقطه نظر تعداد نمونه‌های مورد نیاز و نیز تعداد تکرارهای لازم برای همگرایی عملکرد بهتری دارد.

واژه‌های کلیدی: گراف تصادفی - مساله کوتاهترین مسیر - آتاماتای یادگیر - آتاماتای یادگیر توزیع شده

۱ - مقدمه

پیدا کردن کوتاهترین مسیر بین دو گره مشخص از یک گراف وزندار (شبکه)، که به مساله کوتاهترین مسیر شهرت دارد، نحسین بار در سال ۱۹۵۶ توسط محققی به نام Ford مطرح و از این زمان به بعد، الگوریتم‌های مختلفی نیز برای حل آن ارائه شد [۱]. این مدل‌سازیها و الگوریتم‌های ارائه شده برای حل آنها، در حالتی است که هزینه مربوط به یال‌های شبکه یک مقدار مشخص و قطعی باشد. اما در خیلی از کاربردهای واقعی تعیین مقدار واقعی هزینه یال‌ها ممکن نیست. در این موارد استفاده از شبکه‌ای که طول یال‌ها در آن یک متغیر تصادفی است مدل مناسبتری برای مساله است. به عنوان مثال‌هایی از این نوع مدل‌سازی می‌توان شبکه‌های کامپیوتری را در نظر گرفت. ترافیک خطوط شبکه در زمانهای مختلف دارای یک توزیع احتمالی است که در عمل این تابع ممکن است ناشناخته باشد.

گراف تصادفی G به صورت ۳ تایی $G=(V,E,Q_{n \times n})$ تعریف می‌شود که $V=\{1,2,\dots,n\}$ مجموعه رئوس گراف، $E \subset V \times V$ مجموعه یال‌ها و ماتریس $Q_{n \times n}$ تابع توزیع احتمال مربوط به هزینه یال‌ها را مشخص می‌کند. هزینه q_{ij} یال (i,j) یک متغیر تصادفی با تابع چگالی q_{ij} فرض می‌شود. این تابع توزیع از قبل شناخته شده نیست. در گراف G مسیر π_i با طول m_i و هزینه مورد انتظار L_{π_i} از راس مبدأ v_s به راس مقصد v_d ، به عنوان دنباله $\{i_1, i_2, \dots, i_{m_i}\} \subset V$ تعریف می‌شود که $i_1 = V_s$ و $i_{m_i} = V_d$ به ترتیب رئوس مبدأ و مقصد هستند و برای تمام $1 \leq j < m$ $(i_j, i_{j+1}) \in E$ برقرار است. در نظر بگیرید که

Π مسیر مجازی $\{\pi_1, \pi_2, \dots, \pi_r\}$ میان رئوس v_s و v_d وجود داشته باشد. کوتاهترین مسیر میان v_s و v_d مسیری است که کمترین هزینه مورد انتظار را دارد. به عبارت دیگر کوتاهترین مسیر π^* مسیری است که $L_{\pi^*} = \min_{\pi \in \Pi} \{L_\pi\}$.

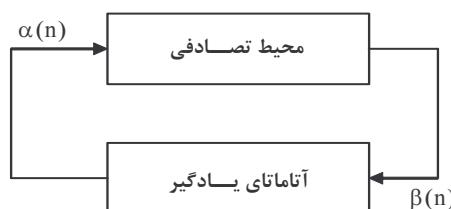
محققی یه نام Frank نخستین کسی بود که مطالعه شبکه‌های تصادفی را آغاز کرد. [2] تحلیل برخی از مسائل مربوط به شبکه‌های تصادفی در کارهای محقق دیگری به نام Pritskar دیده می‌شود. [3] همچنین مدل‌های دیگری از این مساله به همراه تحلیلهایی که بر روی این نوع از مدلها صورت گرفته است را می‌توان در [4] و [5] مشاهده کرد. دو محقق به نامهای Tsitsiklis و Bertsekas، یک مدل عمومی از مساله کوتاهترین مسیر تصادفی را مورد بررسی و مطالعه قرار داده‌اند [6].

فرض می‌کنیم که $P(\pi)$ احتمال اینکه مسیر π کوتاهترین مسیر باشد تعریف شود. یکی از روش‌های تکراری برای محاسبه $P(\pi)$ استفاده از آتمات‌ای یادگیر توزیع شده یا DLA است. مبتدی و بیگی در [8] برای نخستین بار از DLA برای حل مساله یافتن کوتاهترین مسیر در شبکه‌های تصادفی استفاده کرده‌اند. نتایجی که در [8] و [9] آمده حاکی از آن است که در میان الگوریتم‌های خطی، L_{RI} بیشترین تعداد دفعات هم‌گرایی را در مقایسه با سایر روش‌های خطی دارد. در روش ارائه شده در [8] و [9] معیار تعیین پاداش یا جریمه آتمات‌های موجود در یک مسیر انتخاب شده، مقایسه طول مسیر جدید با متوسط طول مسیرهای پیمایش شده تا این مرحله است. متوسط طول مسیرهای پیمایش شده یک مقدار آستانه را تشکیل می‌دهد که به صورت پویا در حال تغییر است. در [10] نشان داده شده است که با استفاده از الگوریتم یادگیری غیرخطی دنباله رو ۱ و استفاده از مقدار آستانه پویای معرفی شده در [9]، در یک آتمات‌ای یادگیر توزیع شده می‌توان مساله یافتن کوتاهترین مسیر در شبکه‌های تصادفی را با میانگین تعداد تکرار کمتر و با سرعت بیشتر حل کرد. در این مقاله نشان می‌دهیم که با استفاده از یک معیار جدید برای تعیین پاداش یا جریمه آتمات‌ها در آتمات‌ای یادگیر توزیع شده می‌توان میانگین تعداد تکرارها و متوسط تعداد نمونه‌های لازم از یالها، برای یافتن کوتاهترین مسیر تصادفی توسط DLA را کاهش داد.

ادامه مقاله به صورت زیر سازماندهی شده است. در بخش ۲ به معرفی آتمات‌ای یادگیر پرداخته و الگوریتم‌های یادگیری خطی و الگوریتم یادگیری دنباله رو را معرفی می‌کنیم. در بخش ۳ آتمات‌ای یادگیر توزیع شده را معرفی و در بخش ۴ الگوریتم پیشنهادی برای حل مساله کوتاهترین مسیر تصادفی به کمک آتمات‌ای یادگیر توزیع شده را بررسی خواهیم کرد. در بخش ۵ به کمک الگوریتم بخش ۴ و استفاده از روش‌های یادگیری خطی و دنباله رو، کوتاهترین مسیر تصادفی را در شبکه‌های استاندارد نمونه خواهیم یافت و سپس به ارزیابی شاخصهای معرفی شده می‌پردازیم.

۲- آتمات‌ای یادگیر^۲

آتمات‌ای یادگیر یک مدل انتزاعی است که تعداد محدودی عمل را می‌تواند انجام دهد. هر عمل انتخاب شده توسط محیطی احتمالی ارزیابی شده و پاسخی به آتمات‌ای یادگیر داده می‌شود. آتمات‌ای یادگیر از این پاسخ استفاده نموده و عمل خود را برای مرحله بعد انتخاب می‌کند.



شکل ۱: ارتباط بین آتمات‌ای یادگیر و محیط

¹ Pursuit Learning Algorithm

² Learning Automata

شکل ۱ ارتباط بین آتماتای یادگیر و محیط را نشان می دهد. محیط ۳ را می توان توسط سه تابع $E \equiv \{\alpha, \beta, c\}$ 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. c_i احتمال اینکه عمل α_i نتیجه نامطلوب داشته باشد می باشد.

آتماتای یادگیر با ساختار متغیر^۴: آتماتای یادگیر با ساختار متغیر توسط ۴ تابع $\{\alpha, \beta, p, T\}$ نشان داده می شود که در آن $p \equiv \{p_1, p_2, \dots, p_r\}$ مجموعه عملهای آتماتا، $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عملهای آتماتا، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودیهای آتماتا، $T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می باشد. در این نوع از آتماتاها، اگر 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 آتماتاها، اگر α_i در مرحله n ام انتخاب شود و پاسخ مطلوب^۵ از محیط دریافت نماید، احتمال (n) p_i افزایش یافته و سایر احتمالها کاهش می یابند. و برای پاسخ نامطلوب^۶ احتمال (n) کاهش یافته و سایر احتمالها افزایش می یابند.

الگوریتم یادگیری خطی: قانون یادگیری مورد استفاده در این الگوریتم به صورت زیر می باشد:

الف - پاسخ مطلوب

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

ب - پاسخ نامطلوب

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

در روابط فوق، a پارامتر پاداش و b 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^۷ L_{RP} می نامیم. زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را^۸ L_{REP} می نامیم. زمانیکه b مساوی صفر باشد، الگوریتم را^۹ L_R می نامیم. [11]

الگوریتم یادگیری دنباله رو: این الگوریتم از سابقه پاداش و جریمه عملهای آتماتا برای پاداش دادن یا جریمه کردن آتماتاها استفاده می کند [12][13] در این الگوریتم (t) W_i نشانگر تعداد دفعاتی است که عمل α_i ام آتماتا تاکنون پاداش گرفته است. $Z_i(t)$ تعداد دفعاتی است که عمل α_i ام آتماتا تاکنون انتخاب شده است. $\Delta = \frac{1}{rN}$ طول گام مورد استفاده برای پاداش دادن به عملهای آتماتا است که در آن ۲ تعداد عملهای قابل انجام توسط آتماتا است و N پارامتر تفکیک ۱۰ نام دارد در هر مرحله یکی از عملهای قابل انجام توسط آتماتا انتخاب می شود. فرض کنید در لحظه t عمل $\alpha_i = \alpha(t)$ انتخاب شده باشد. همچنین فرض کنید عمل m بیشترین نسبت پاداش به انتخاب را داراست ($d_m(t) = \max_{i=1,2,\dots,r} \{d_i(t)\}$). در این صورت الگوریتم دنباله رو مطابق روابط زیر بردار احتمالات مربوط به عملها را اصلاح می کند:

3 Environment

4 Variable Structure Learning Automata

5 Favorable

6 Unfavorable

7 Linear Reward Penalty

8 Linear Reward Epsilon Penalty

9 Linear Reward Inaction

10 Resolution

$$d_m(t) = \max_{i=1,2,\dots,r} \{d_i(t)\}$$

if $\beta(t) = 0$ and $p_m(t) < 1$ then

$$\begin{cases} p_j(t+1) = \max_{j \neq m} \{p_j(t) - \Delta, 0\} \\ p_m(t+1) = 1 - \sum_{j \neq m} p_j(t+1) \end{cases}$$

else $p_j(t+1) = p_j(t)$

همچنین در هر مرحله بردارهای W , Z و d مطابق روابط زیر بهنگام می‌شوند:

$$W_i(t+1) = W_i(t) + (1 - \beta(t))$$

$$Z_i(t+1) = Z_i(t) + 1$$

$$d_i(t+1) = \frac{W_i(t+1)}{Z_i(t+1)}$$

۳- آتماتای یادگیر توزیع شده

آتماتای یادگیر توزیع شده شبکه‌ای است از آتماتاهای یادگیر که برای حل یک مساله خاص با یکدیگر همکاری دارند. در این شبکه از آتماتاهای همکار در هر زمان تنها یک آتماتا فعال است. تعداد اعمال قبل انجام توسط یک آتماتا در DLA برابر است با تعداد آتماتاهایی که به این آتماتا متصل شده‌اند. انتخاب یک عمل توسط آتماتا در این شبکه، باعث فعال شدن آتماتای متصل شده به این آتماتا و منتظر با این عمل می‌گردد. به عبارت معادل، انتخاب یک عمل توسط یک آتماتا در این شبکه منتظر با فعال شدن یک آتماتای دیگر در این شبکه است. [8]

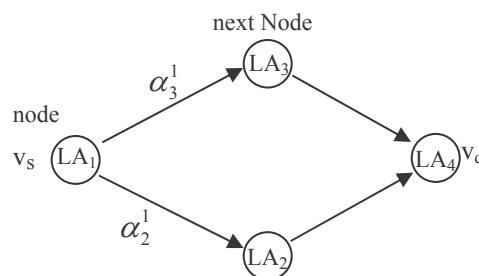
مدلی که برای شبکه DLA در نظر می‌گیریم، یک گراف است که هر یک از رئوس آن یک آتماتا است

$$DLA = (V, E)$$

$$V = \{LA_1, LA_2, \dots, LA_n\}$$

$$E \subset V \times V$$

$$(LA_i, LA_j) \in E$$



شکل ۲: اجزای یک سیستم DLA مورد استفاده برای حل مساله کوتاهترین مسیر تصادفی

وجود یال (LA_i, LA_j) در این گراف بدان معناست که انتخاب عمل α_j^i توسط LA_i باعث فعال شدن LA_j می‌گردد.

۴- الگوریتم پیشنهادی (مقایسه دو مسیر آخر)

برای حل مساله کوتاهترین مسیر تصادفی به کمک آتماتای یادگیر از DLA استفاده می‌کنیم. در این روش ، گراف تصادفی نقش محیط تصادفی (ناشناخته) را برای DLA و تمام LA های موجود در ساختار DLA بازی می‌کند. خروجی DLA، یک دنباله از

قدمه‌است که یک مسیر را در گراف تصادفی نمایش می‌دهد. محیط از طول مسیر مشخص شده، جهت تولید پاسخ برای DLA استفاده می‌کند (تابع ارزیابی عملکرد DLA). اگر عمل انتخاب شده قابل قبول باشد عملهای انتخاب شده توسط کلیه آتماتاها در طول مسیر پاداش می‌گیرند و گرنه کلیه اعمال انتخاب شده جریمه می‌شوند. برای یافتن کوتاهترین مسیر تصادفی در شبکه‌ها به کمک آتماتای یادگیر توزیع از الگوریتم زیر استفاده می‌کنیم:

گام اول: ساختن یک DLA یکریخت با گراف مساله. این DLA یک شبکه از LA هاست که هر کدام در یکی از رئوس گراف قرار گرفته است. هر یک از رئوس خروجی یک راس، یکی از قدمهای قابل انجام توسط LA متناظر با آن راس را نمایش می‌دهند. انتخاب یکی از عملهای قابل انجام توسط آتماتا منجر به فعال شدن آتماتای موجود در سوی دیگر یا اتصال دهنده دو آتماتا می‌شود. گام دوم: مسیریابی به کمک آتماتاها و تعیین پاداش یا جریمه آنها. الگوریتم به روش زیر پیمایش گراف را آغاز می‌کند تا کوتاهترین مسیر را بیابد: در هر مرحله آتماتای LA_s (آتماتای واقع در راس مبدأ) یکی از عملهای قابل انجام خود را انتخاب می‌کند (α_m^s) انتخاب این عمل باعث فعال شدن LA_m در سوی دیگر می‌شود. فرآیند انتخاب یک عمل و فعال شدن آتماتای متناظر آنقدر (pathCost) ادامه می‌یابد تا به آتماتای مقصد LA_d برسیم. بعد از آنکه به آتماتای LA_d رسیدیم، هزینه مسیر پیمایش شده (pathCost) محاسبه می‌گردد. سپس هزینه مسیر فعلی (pathCost) با هزینه مسیر قبلی (prevPathCost) مقایسه می‌گردد. بر حسب این مقایسه پاداش یا جریمه آتماتاهای موجود در مسیر انجام می‌شود. چنانچه هزینه متوسط مسیر فعلی از هزینه متوسط مسیر قبلی کوچکتر باشد، کلیه آتماتاهای مسیر فعلی پاداش می‌گیرند و کلیه آتماتاهای مسیر قبلی جریمه می‌شوند و بر عکس

Procedure DLA-SSPP

Input: Graph $G=(V,E)$, Threshold, v_s, v_d ;
Output: Path, PathLen, PathCost;

Begin

Construct The DLA from G;

Find firstPath;

Set prevPath=firstPath;

Repeat

pathCost=0; pathLen=1; Node=Path[1]= v_s ;
While node<> v_d and PathLen< n do

nextNode=node.getAdjNode();

pathLen=pathLen+1;

path[pathLen]=nextNode;

pathCost=pathCost+

edgecost(node, nextNode);

node=nextNode;

end while

if (path[pathLen]== v_d) then

if (pathCost < prevPathCost) then

rewardPath(path);

penalizePath(prevPath);

End if

If (pathCost > prevPathCost) then

rewardPath (prevPath);

penalizePath(path);

End if

prevPath=path;

Endif

Until probability of selected path >= Threshold

End DLA-SSPP;

الگوریتم پاداش یا جریمه هر یک از آتماتاهای موود در مسیر می‌تواند الگوریتم خطی یا الگوریتم دنباله رو باشد

۵- نتایج شبیه سازی

برای مقایسه عملکرد الگوریتم های مختلف ارائه شده برای حل مساله کوتاهترین مسیر و الگوریتم جدید دو گروه شبیه سازی صورت گرفته است. در هر یک از این شبیه سازیها از ۵ شبکه استاندارد نمونه Net1 تا Net5 استفاده شده است (شکل های ۳ تا ۶) [۱۰]. شبکه Net4 و Net5 مشابه یکدیگرند متنها توزیع طول یالها به گونه ای تغییر یافته که تفاوت میانگین طول مسیرهای میان مبدأ و مقصد محسوس نیست باشد [۱۰]. در هر یک از این شبکه ها، یک توزیع گسسته با مقادیر مثبت به عنوان طول یال درنظر گرفته شده است. مسیر بهینه، مسیری است که کمترین طول مورد انتظار را دارد. مسیرهای بهینه برای هر ۵ شبکه بر اساس میانگین طول یالها و به کمک الگوریتم دیجیسترا محاسبه شده است.

بر اساس الگوریتم های مختلفی که توضیح داده شد، چهار روش مختلف مبتنی بر DLA برای حل مساله کوتاهترین مسیر تصادفی می توان داشت که عبارتند از:

Alg1: الگوریتم یادگیری L_{RI} و معیار هزینه متوسط (مرجع [۹])

Alg2: الگوریتم یادگیری L_{RI} و معیار مقایسه دو مسیر اخیر (معیار پیشنهادی)

Alg3: الگوریتم یادگیری دنباله رو و معیار هزینه متوسط (مرجع [۱۰])

Alg4: الگوریتم یادگیری دنباله رو و معیار مقایسه دو مسیر اخیر. (معیار پیشنهادی)

در گروه اول این آزمایش ها (الگوریتم های Alg1 و Alg2)، مقایسه میان عملکرد الگوریتم پیشنهادی (مقایسه دو مسیر آخر) و الگوریتم پیشنهادی در [۹] صورت گرفته است. نتایج این مقایسه نشان می دهد که الگوریتم جدید پیشنهادی تعداد نمونه های موردنیاز را از ۱۱٪ در شبکه های کوچک تا ۶۵٪ در شبکه های بزرگ و پیچیده کاهش می دهد.

در گروه دوم از آزمایش ها، مقایسه میان الگوریتم پیشنهادی و الگوریتم پیشنهادی در [۱۰] صورت گرفته است. (Alg3 و Alg4) نتایج این مقایسه حاکی از بهبود عملکرد روش جدید در مقایسه با روش پیشنهادی در [۱۰] است. نتایج محاسبه شاخصهای، میانگین تعداد تکرارهای لازم برای رسیدن به مسیر بهینه (Avg Iter)، درصد تکرارهای همگرا شده به مسیر بهینه (Conv. Runs)، میانگین Short. Samp (Avg Samp)، میانگین تعداد نمونه های گرفته شده از مسیر بهینه در هر اجرا (Path) را در جداول ۱ تا ۱۰ مشاهده می کنید.

مقایسه جداول ۱ با ۳ و نیز جدولهای ۲ با ۴ نشان می دهد که معیار جدید تعداد تکرارهای لازم برای همگرایی را کاهش می دهد. مقایسه جداول ۶ با ۸ و ۷ با ۹ نیز حاکی از بهبود روش مقایسه دو مسیر اخیر در مقایسه با روش هزینه متوسط است. میزان بهبود هریک از این روشها نیز در جداول ۶ و ۱۱ محاسبه شده است.

a	Net1		Net2		Net3		Net4	
	Avg Iter	Conv. Runs						
0.001	4445.64	100	4438.85	100	49270.2	100	74985.72	100
0.051	99.41	100	90.05	100	989.3059	85	1431.831	89
0.101	60.18	100	45.71	100	420.8158	76	670.875	80
0.151	41.17172	99	30.87	100	246.4219	64	296.9342	76
0.201	35.48	100	24	100	169.5857	70	182.7586	58

جدول شماره ۱: میانگین تعداد تکرارها و درصد تکرارهای همگرا شده در الگوریتم Alg1

a	Net1		Net2		Net3		Net4	
	Avg sam	Short. path	Avg sam	Short. path	Avg sam	Short. path	Avg sam	Short. path
0.001	8895.28	6453.34	10365.14	7002.14	160321.6	75557.96	255315.7	155167.6
0.051	202.82	141.24	215.53	144.27	3326.03	1368.9	4968.26	2426.65
0.101	124.36	82.96	111.41	75.22	1502.46	582.84	2229.16	917.15
0.151	86.36	55.66	76.81	51.71	875.7	328	1059.9	397.13
0.201	74.96	46.46	61.41	40.73	630.6	233.06	763.7998	245.17

جدول شماره ۲: کل تعداد نمونه گیریها و نمونه های گرفته شده از کوتاهترین مسیر در الگوریتم Alg1

a	Net1		Net2		Net3		Net4	
	Avg Iter	Conv. Runs						
0.001	4484.86	100	3920.35	100	14082.58	100	16898.52	100
0.051	98.91	100	77.4	100	289.82	100	384.3441	93
0.101	51.57	100	38.16	100	155.7347	98	178.9882	85
0.151	34.30612	98	26.56	100	96.43434	99	129.2963	81
0.201	26.95876	97	20.09	100	78.64211	95	89.41428	70

جدول شماره ۳: میانگین تعداد تکرارها و درصد تکرارهای همگرا شده در الگوریتم Alg2

a	Net1		Net2		Net3		Net4	
	Avg sam	Short. path	Avg sam	Short. path	Avg sam	Short. path	Avg sam	Short. path
0.001	8973.72	6506.28	9113.891	6231.73	50294.34	26391.91	67131.52	27189.29
0.051	201.82	138.94	185.54	124.94	1040.1	530.33	1499.78	527.16
0.101	107.14	73.14	93.02	64.07	553.3	264.6	714.1201	246.48
0.151	72.54	48.02	66.88001	44.74	346.82	172.57	504.81	163.07
0.201	57.48	36.54	52.07	33.99	284.27	131.76	369.37	105.26

جدول شماره ۴: کل تعداد نمونه گیریها و نمونه های گرفته شده از کوتاهترین مسیر در الگوریتم Alg2

شبکه	درصد کاهش تعداد تکرارها به صورت متوسط	درصد کاهش تعداد نمونه گیریها
Net1	٪۱۰	٪۱۱
Net2	٪۱۴	٪۱۴
Net3	٪۶۳	٪۶۲
Net4	٪۶۶	٪۶۳

جدول شماره ۵: میزان درصد کاهش تعداد تکرارها و نمونه گیریها در الگوریتم Alg2 نسبت به Alg1 (این نتایج از جداول ۱ تا ۴ استخراج شده است)

resolution	Net1		Net2		Net1		Net2	
	Avg. Iter	Conv. Runs	Avg. Iter	Conv. Runs	Avg. samp.	Short. path	Avg. amp	Short. Path
2	12.75	100	5.39	100	29.5	18.28	18.35	10.15
4	14.41	100	8.25	100	32.82	22.02	24.69	15.25
6	19.56	100	9.97	100	43.12	26.36	29.57	16.91
8	18.76	100	12.68	100	41.52	26.56	35.92	20.94
10	18.39	100	13.77	100	40.78	26.26	37.88	23.51

جدول شماره ۶: میانگین تعداد تکرارها و درصد تکرارهای همگرا شده ، کل تعداد نمونه گیریها و نمونه های گرفته شده از کوتاهترین مسیر در الگوریتم Alg3

resolution	Net3		Net5		Net3		Net5	
	Avg. Iter	Conv. Runs	Avg. Iter	Conv. Runs	Avg. samp.	Short. path	Avg. amp	Short. Path
20	63.56	100	876.2421	95	271.11	104.69	5999.35	3672.49
22	71.07	100	985.7957	93	300.64	119.4	6870.3	4141.67
24	71.17	100	826.9663	89	301.2	119.5	7317.541	4442.29
26	75.27	100	1003.516	93	321.42	127.44	7427.559	4330.92
28	85.03	100	1182.022	91	359.24	142	7529.25	4410.08

جدول شماره ۷: میانگین تعداد تکرارها، درصد تکرارهای همگرا شده ، کل تعداد نمونه گیریها و نمونه های گرفته شده از کوتاهترین مسیر در الگوریتم Alg3

resolution	Net1		Net2		Net1		Net2	
	Avg Iter	Conv. Runs	Avg Iter	Conv. Runs	Avg samp	Short. path	Avg amp	Short. Path
2	9.59	100	3.49	100	23.18	13.82	13.39	7.71
4	9.36	100	5.53	100	22.72	15.52	18.22	11.06
6	10.27	100	6.67	100	24.54	14.76	21.01	12.73
8	12.03	100	8.5	100	28.06	18.46	25.28	15.59
10	13.76	100	9.68	100	31.52	19.52	28.14	17.21

جدول شماره ۸: میانگین تعداد تکرارها ، درصد تکرارهای همگرا شده ، کل تعداد نمونه گیریها و نمونه های گرفته شده از کوتاهترین مسیر در الگوریتم Alg4

resolution	Net3		Net5		Net3		Net5	
	Avg Iter	Conv. Runs	Avg Iter	Conv. Runs	Avg samp	Short. path	Avg amp	Short. Path
20	40.09	100	761.8283	99	169.64	70.15	3407.77	2704.4
22	40.15	100	745.8145	97	172.07	72.43	4008	3001.86
2	42.24	100	444.2887	97	181.38	76.14	2842	2024.87
26	44.93	100	221.6939	98	191.17	82.46	1640.63	1025.85
28	44.5	100	422.5155	97	190.24	84.02	2772.08	1889.62

جدول شماره ۹ : میانگین تعداد تکرارها ، درصد تکرارهای همگرا شده ، کل تعداد نمونه گیریها و نمونه های گرفته شده از کوتاهترین مسیر در الگوریتم Alg4

شبکه	درصد کاهش تعداد تکرارها به صورت متوسط	درصد کاهش تعداد نمونه گیریها
Net1	%۳۳	%۳۰
Net2	%۳۲	%۲۷
Net3	%۴۲	%۴۱
Net5	%۴۵	%۵۷

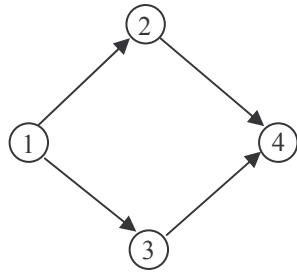
جدول شماره ۱۰: میزان درصد کاهش تعداد تکرارها و نمونه گیریها در الگوریتم Alg4 نسبت به Alg3 (این نتایج از جداول ۶ تا ۹ استخراج شده است)

۶- نتیجه گیری

در این مقاله یک الگوریتم جدید مبتنی بر آتماتاتی یادگیر توزیع شده که در آن از یک معیار جدید برای تعیین پاداش و یا جریمه برای عملهای انتخابی توسط یک آتماتاتی یادگیر توزیع شده استفاده می‌شود برای حل مساله کوتاهترین مسیر تصادفی ارایه گردید. نتایج شبیه سازیها نشان داد که این معیار جدید تعداد نمونه‌های مورد نیاز و نیز تعداد تکرارهای لازم برای همگرایی الگوریتم‌های حل مساله کوتاهترین مسیر تصادفی را، در مقایسه با دیگر روش‌های مبتنی بر آتماتاتی یادگیر توزیع شده، کاهش می‌دهد. همچنین نتایج شبیه سازیها نشان میدهد استفاده از معیار مقایسه دو مسیر آخر به همراه الگوریتم یادگیری غیرخطی دنباله رو (Alg4) بهترین نتیجه را در میان روش‌های یافتن کوتاهترین مسیر تصادفی مبتنی بر DLA دارد.

۷- مراجع

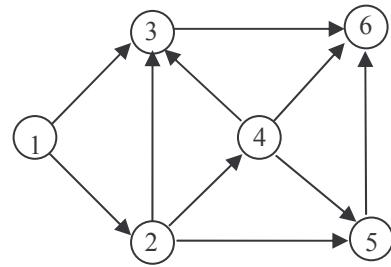
- [1] N. Deo and C. Pang, “**Shortest path algorithms: Taxonomy and annotation,**” Networks 14 (1984) pp.275-323.
- [2] H. Frank, “**Shortest paths in probabilistic graphs ,**” Oper. Res. 17 (1969), pp. 583-599.
- [3] C. C. Sigal, A. A. B. Pritsker, “**The Stochastic Shortest Route Problem,**” Operations Research, 28(1980), pp.1122-1129.
- [4] G.Andretta and L.Romeo, ”**Stochastic shortest paths with recourse,**” Networks 18 (1988) pp. 193-204.
- [5] C.Alexopoulos, “**State space partitioning methods for stochastic shortest path problems,**” Networks 30 (1997) pp. 9-21.
- [6] D.P.Bertsekas, and J.N.Tsitsiklis, “**An Analysis of Stochastic Shortest Path Problems,**” Technical Report LIDS-P-1815, Laboratory of Information and Decisions Systems, M.I.T., 1990.
- [7] D.P.Bertsekas and J.N.Tsitsiklis, “**An analysis of stochastic shortest path problems,**” Mathematics of Operations Researches, 16 (1991), pp.580-595.
- [8] M.R.Meybodi and H. Beigy, “**Solving Stochastic Shortest Path Problem Using Distributed Learning Automat,**” in Proceedings of CSICC-2001, Isfahan, Iran, 2001, pp. 70-86.
- [9] M.R.Meybodi and H.Beigy, “**A New Distributed Learning Automat Based Algorithm For Solving Stochastic Shortest Path Problem,**” in Proceeding of the Sixth International Joint Conference on Information Science, Durham U.S.A , 2002, pp. 334-343.
- [10] محمدرضا ملاخیلی مبیدی و محمدرضا مبیدی ”یک روش جدید مبتنی بر آتماتاتی یادگیر توزیع شده برای حل مساله کوتاهترین مسیر تصادفی،” گزارش فنی، آزمایشگاه سیستم‌های نرم‌افزاری دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر، مهرماه ۱۳۸۲ (<http://elc.maybod.info/downloads/softlabTR782.pdf>)
- [11] K. S. Narendra and M. A. L. Thatachar, “**Learning Automata: an Introduction,**” Englewood cliffs, NJ, Prentice-Hall, 1989.
- [12] B. J. Oomen and Martin Agache, “**A Comparison of Continuous and Discretized Pursuit Learning Schemes,**” Technical Report, Carleton University, Ottawa, 2002.
- [13] Mariana Agache and B. John Oomen, “**Generalized Pursuit Learning Schemes: New Families of Continuous and Discretized Learning Automata,**” IEEE Transactions on systems man. And cybernetics, Vol 32, No. 6, December 2002, pp. 738-749.



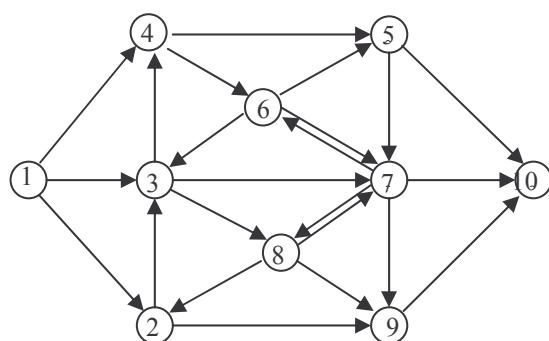
يال	وزن			احتمال		
(1,2)	6	14	15	0.30	0.45	0.25
(1,3)	2	9	10	0.10	0.70	0.20
(3,4)	6	13	15	0.15	0.30	0.55
(2,4)	4	11	18	0.25	0.10	0.30
						0.35

شكل (٣) شبكة net1 و توزيع وزن يالها

يال	وزن			احتمال		
(1,3)	1	10		0.8	0.2	
(1,2)	4	10	100	0.2	0.4	0.4
(2,3)	10	50	100	0.1	0.8	0.1
(2,4)	20	100		0.1	0.9	
(2,5)	10	90		0.1	0.9	
(3,6)	1	100		0.9	0.1	
(4,3)	10	100		0.2	0.8	
(4,5)	50	50		0.5	0.5	
(4,6)	100	100		0.2	0.8	
(5,6)	1	10	100	0.6	0.2	0.2



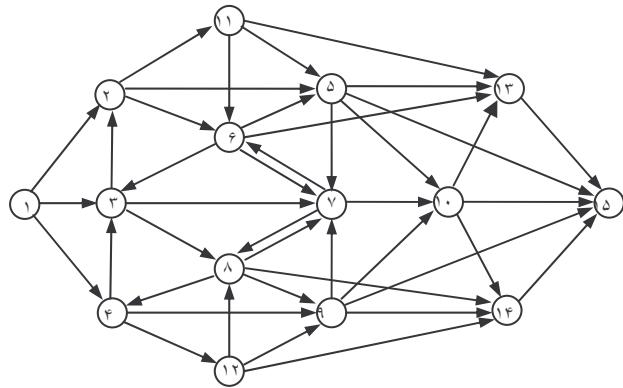
شكل (٤) شبكة net2 و توزيع وزن يالها



يال	وزن				احتمال		
(1,2)	3.0	5.3	7.4	9.7	0.2	0.2	0.3
(1,3)	3.5	6.2	7.9	8.5	0.3	0.3	0.2
(1,4)	4.2	6.1	6.9	8.9	0.2	0.3	0.2
(2,5)	2.6	4.1	5.5	9.0	0.2	0.2	0.4
(2,6)	5.8	7.0	8.5	9.6	0.3	0.3	0.2
(3,2)	1.5	2.3	3.6	4.5	0.2	0.2	0.3
(3,7)	6.5	7.2	8.3	9.4	0.5	0.2	0.2
(3,8)	5.9	7.8	8.6	9.9	0.4	0.3	0.2
(4,3)	2.1	3.2	4.5	6.8	0.2	0.2	0.3
(4,9)	1.1	2.2	3.5	4.3	0.2	0.3	0.4
(5,7)	3.2	4.8	6.7	8.2	0.2	0.2	0.3
(5,10)	6.3	7.8	8.4	9.1	0.2	0.2	0.4
(6,3)	6.8	7.7	8.5	9.6	0.4	0.1	0.1
(6,5)	0.6	1.5	3.9	5.8	0.2	0.2	0.3
(6,7)	2.1	4.8	6.6	7.5	0.2	0.4	0.2
(7,8)	1.6	2.8	5.2	6.0	0.2	0.3	0.2
(7,10)	1.6	3.4	8.2	9.3	0.2	0.3	0.2
(8,4)	7.0	8.0	8.8	9.4	0.2	0.2	0.4
(8,7)	2.1	4.6	8.5	9.6	0.4	0.2	0.2
(8,9)	1.7	4.9	6.5	7.8	0.2	0.2	0.4
(7,9)	3.5	4.0	5.0	7.7	0.1	0.2	0.4
(9,10)	4.6	6.4	7.6	8.9	0.4	0.1	0.2

شكل (٥) شبكة net3 و توزيع وزن يالها

یال	وزن	احتمال
(1,2)	16 25 36	0.6 0.3 0.1
(1,3)	21 24 25 39	0.5 0.2 0.2 0.1
(1,4)	11 13 26	0.4 0.4 0.2
(2,11)	24 28 31	0.5 0.3 0.2
(2,5)	11 30	0.7 0.3
(2,6)	13 37 39	0.6 0.2 0.2
(3,2)	11 20 24	0.6 0.3 0.1
(3,7)	23 30 34	0.4 0.3 0.3
(3,8)	14 23 34	0.5 0.4 0.1
(4,3)	22 30	0.7 0.3
(4,9)	35 40	0.6 0.4
(4,12)	16 25 37	0.5 0.4 0.1
(5,13)	28 35 37 40	0.4 0.3 0.2 0.1
(5,15)	25 32	0.7 0.3
(5,10)	27 33 40	0.4 0.3 0.3
(5,7)	15 17 19 26	0.3 0.3 0.3 0.1
(6,5)	18 25 29	0.5 0.3 0.2
(6,13)	21 23	0.5 0.5
(6,7)	11 31 37	0.5 0.4 0.1
(6,3)	18 24	0.7 0.3
(7,10)	19 23 37	0.6 0.2 0.2
(7,8)	12 15 22 4	0.3 0.3 0.2 0.2
(7,6)	12 23 31	0.5 0.3 0.2
(8,7)	14 34 39	0.6 0.2 0.2
(8,14)	14 15 27 32	0.3 0.3 0.2 0.2
(8,9)	13 31 32	0.8 0.1 0.1
(8,4)	13 23 34	0.4 0.3 0.3
(9,7)	10 17 20	0.6 0.3 0.1
(9,10)	16 18 36 9	0.3 0.3 0.2 0.2
(9,15)	12 13 25 2	0.4 0.3 0.2 0.1
(9,14)	19 24 29	0.4 0.3 0.3
(10,13)	14 20 25 2	0.3 0.3 0.2 0.2
(10,15)	15 19 25	0.4 0.3 0.3
(10,14)	23 34	0.9 0.1
(11,13)	13 31 25	0.6 0.3 0.1
(11,5)	18 19 20 23	0.3 0.3 0.3 0.1
(11,6)	10 19 39	0.5 0.4 0.1
(12,8)	15 36 39	0.5 0.3 0.2
(12,9)	16 22	0.7 0.3
(12,14)	10 13 18 34	0.3 0.3 0.3 0.1
(13,15)	12 31	0.9 0.1
(14,15)	14 19 32	0.5 0.3 0.2



شکل ۶) شبکه net4 و توزیع وزن یالها

مسیر	طول
1- 2- 5 - 15	67.6
1- 4 - 9 - 15	75.2
1- 4 - 12 - 14 - 15	71.3
1- 2 - 11 - 13 - 15	78.8

طول متوسط برخی از مسیرهای میان مبدا و مقصد در شبکه

Net5

جدول شماره ۱۱) تفاوت طول مسیرها در شبکه های Net4 و Net5

مسیر	طول
1- 2- 5 - 15	67.6
1- 4 - 9 - 15	71.2
1- 4 - 12 - 14 - 15	71.3
1- 2 - 11 - 13 - 15	70.8

طول متوسط برخی از مسیرهای میان مبدا و مقصد در شبکه

Net4