

طراحی مسیر حرکت عامل ها در سیستم شبیه ساز امداد با استفاده از اتوماتای یادگیر توزیع شده

مصطفی اصغری^۱، بهروز معصومی^۲، محمدرضا میبیدی^۳

۱- دانشگاه آزاد اسلامی - واحد میاندوآب-ایران

m_asghary86@yahoo.com

۲-دانشگاه آزاد اسلامی قزوین- دانشگاه آزاد اسلامی علوم و تحقیقات تهران- ایران

bmasoumi@Qazviniau.ir

۳-دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر- تهران- ایران

mmeybodi@aut.ac.ir

چکیده:

یکی از بسترهای مناسب برای تست و ارزیابی ایده ها و تکنیک های مربوط به سیستم های چند عامله سیستم شبیه ساز امداد می باشد. در سیستم شبیه ساز امداد، وقوع یک حادثه مخرب شبیه سازی شده و عامل های نرم افزاری برای مقابله با اثرات حادثه در محیط مصنوعی فعالیت می کنند. در این مقاله که اولین کاربرد اتوماتای یادگیر توزیع شده در سیستم شبیه ساز امداد است برای مسیریابی عامل ها از اتوماتای یادگیر توزیع شده استفاده شده و تاثیر آن در افزایش قدرت کاوش عامل های امداد ارزیابی شده است. نتایج بدست آمده حاکی از آن است که روش مذکور در مقایسه با روش های استفاده شده دیگر، قدرت جستجو و کاوش بیشتری را به عامل ها می دهد.

واژه های کلیدی:

اتوماتای یادگیر توزیع شده، شبیه ساز امداد، طراحی مسیر.

۱- مقدمه

ویرانگر و مخرب مانند زلزله در یک محیط شهری شبیه سازی می شود. ساختمانها فرو می ریزند و یا آتش می گیرند. شهروندان در زیر آوار گیر می کنند و توسط آتش تهدید می شوند. راههای ارتباطی نیز به دلیل فرو ریختن آوار ساختمانها مسدود می شود. در چنین شرایطی، آمبولانس ها باید در سریعترین زمان ممکن به یاری مصدومین بشتابند و آنها را به پناهگاه منقل نمایند. آتش نشانها وظیفه دارند آتش را خاموش کرده و از انتشار آن جلوگیری کنند. همچنانکه قبلا نیز بیان شد در تمام این موارد زمان نقش کلیدی دارد. یکی از عوامل مهم و اساسی برای رسیدن به اهداف فوق این است که عامل ها در کوتاهترین زمان ممکن بتوانند اطلاعات نسبتا کاملی از محیط به دست آورند. کامل بودن اطلاعات عامل ها از

سیستم شبیه ساز امداد نمونه ای از سیستم های چند عامله بلادرنگ است که در آن محیط عملیاتی عامل ها دارای پیچیدگی بالایی بوده و به شرایط واقعی نزدیکتر می باشد. فراهم بودن شرایط سیستم های چندعامله پیچیده و بزرگ در سیستم شبیه ساز امداد، آن را به بستر مناسبی برای تست و ارزیابی ایده ها و تکنیک های مربوط به سیستم های چند عامله تبدیل کرده است. با توجه به بلادرنگ بودن سیستم شبیه ساز امداد، یکی از پارامترهای اصلی که باید در ایده ها و تکنیک های استفاده شده در آن مورد توجه ویژه قرار گیرد زمان می باشد. در سیستم شبیه ساز امداد، رویداد یک حادثه



شکل ۱- ارتباط بین اتوماتای یادگیر و محیط

محیط را می توان توسط سه تایی $E = \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودی ها، $\beta = \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه خروجی ها و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمال های جریمه می باشد. هرگاه β مجموعه ای دو عضوی باشد محیط از نوع P است. در چنین محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته می شود. در محیط از نوع Q، $\beta(n)$ می تواند به طور گسسته یک مقدار از مقادیر محدود در فاصله $[0, 1]$ را اختیار کند و در محیط از نوع S، $\beta(n)$ متغیر تصادفی در فاصله $[0, 1]$ است. c_i احتمال اینکه عمل α_i نتیجه مطلوب داشته باشد می باشد. در محیط ایستا، مقادیر c_i بدون تغییر باقی می مانند حال آنکه در محیط غیر ایستا این مقادیر در طی زمان تغییر می کنند.

۲-۱- اتوماتای یادگیر با ساختار متغیر

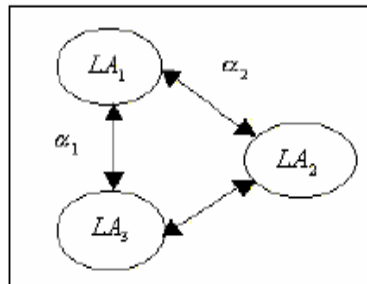
اتوماتای یادگیر با ساختار متغیر توسط ۴ تایی $\{\alpha, \beta, p, T\}$ نشان داده می شود که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عمل های اتوماتا، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودی های اتوماتا، $p \equiv \{p_1, p_2, \dots, p_r\}$ بردار احتمال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

محیط، باعث می شود عاملها تصمیمات مناسب تری بگیرند. با توجه به محدودیت پهنای باند و تعداد پیغام های مبادله شده بین عامل ها، هر عامل سعی می کند تا با حرکت و کاوش در محیط، اطلاعات بیشتری در اختیار داشته باشد. عامل ها حتی در صورت داشتن اطلاعات کامل از محیط، برای انجام اعمال مورد نظرشان نیاز به جابجا شدن و حرکت در محیط دارند. بنابراین طراحی مسیر حرکت عامل ها یکی از موارد مهم و تاثیرگذار در کارایی سیستم است. اکثر روش های استفاده شده برای طراحی مسیر حرکت عامل ها، بر اساس الگوریتم های کوتاهترین مسیر می باشند [1]. عیب استفاده از روش های مسیریابی بر اساس الگوریتم های کوتاهترین مسیر این است که مکانهایی که عامل ها ملاقات می کنند محدود به مواردی می شود که در کوتاهترین مسیر بین مبدا و مقصد قرار گیرد. در این مقاله الگوریتمی ارائه می شود که برای طراحی مسیر حرکت عامل ها از اتوماتای یادگیر توزیع شده استفاده نموده و قدرت کاوش عامل ها را افزایش می دهد. بخش بندی بقیه مطالب این مقاله به صورت زیر است: ابتدا در بخش ۲ اتوماتای یادگیر و اتوماتای یادگیر توزیع شده به طور مختصر و خلاصه معرفی می شوند. بخش ۳ به معرفی الگوریتم ارائه شده می پردازد. در بخش ۴ الگوریتم ارائه شده ارزیابی شده است. بخش ۵ مقایسه و نتیجه گیری می باشد.

۲- اتوماتای یادگیر

اتوماتای یادگیر ماشینی است که می تواند تعدادی متناهی عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی می شود و نتیجه ارزیابی در قالب سیگنالی مثبت یا منفی به اتوماتا داده می شود و اتوماتا از این پاسخ در انتخاب عمل بعدی تاثیر می گیرد. هدف نهایی این است که اتوماتا یاد بگیرد تا از بین اعمال خود، بهترین عمل را انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداکثر برساند. کارکرد اتوماتای یادگیر در تعامل با محیط در شکل ۱ مشاهده می شود.

خود را انتخاب می کند که در نتیجه آن یکی از اتوماتاهای یادگیر متصل به آن اتوماتا که متناظر با عمل انتخاب شده می باشد فعال می شود. در هر زمان فقط یک اتوماتای یادگیر در شبکه فعال می باشد.



شکل ۲- اتوماتای یادگیر توزیع شده (DLA) با ۳ اتوماتای یادگیر

به طور رسمی DLA را می توان توسط گراف $DLA = (V, E)$ که $V = \{LA_1, LA_2, \dots, LA_n\}$ مجموعه اتوماتاهای یادگیر و n تعداد اتوماتاهای یادگیر در DLA و $E \subset V \times V$ مجموعه لبه های گراف می باشد، تعریف کرد. لبه (i, j) عمل ز اتوماتای یادگیر LA_i را نشان می دهد. LA_j زمانی فعال خواهد شد که عمل ز اتوماتای یادگیر LA_i انتخاب شود. تعداد اعمال برای اتوماتای یادگیر LA_k برابر درجه خروجی گره متناظر با اتوماتای یادگیر LA_k می باشد. برای اطلاعات بیشتر در باره اتوماتای یادگیر توزیع شده می توان به مراجع [5,6,7] مراجعه نمود.

۳- الگوریتم پیشنهادی

الگوریتم ارائه شده برای طراحی مسیر حرکت عامل ها، از اتوماتای یادگیر به صورت توزیع شده استفاده می کند. به این صورت که در هر یک از تقاطع ها مانند دوراه، سه راه، چهارراه و غیره یک اتوماتای یادگیر وجود دارد که تعداد اعمال آن برابر با تعداد خیابانها یا راههای متصل به آن تقاطع می باشد. اتوماتای یادگیر به عاملی که قصد عبور از آن تقاطع را دارد یکی از خیابان های متصل به تقاطع را به صورت احتمالی نشان می دهد. احتمال

بماند. الگوریتم زیر نمونه ای از الگوریتمهای یادگیری خطی برای اتوماتای یادگیر با ساختار متغیر است.

الف- پاسخ مطلوب:

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)] \quad (1-الف)$$

$$j \neq i \quad \forall j$$

$$p_j(n+1) = (1-a)p_j(n) \quad (1-ب)$$

ب- پاسخ نامطلوب:

$$p_i(n+1) = (1-b)p_i(n) \quad (2-الف)$$

$$(2-ب)$$

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

$$j \neq i \quad \forall j$$

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. زمانی که a و b با هم برابر باشند، الگوریتم را L_{RP} می نامیم. زمانی که b از a خیلی کوچکتر باشد، الگوریتم را L_{REP} می نامیم. زمانی که b مساوی صفر باشد، الگوریتم را L_{RI} می نامیم. در این مقاله از الگوریتم L_{RP} برای یادگیری در اتوماتای یادگیر استفاده شده است. برای اطلاعات بیشتر در باره اتوماتاهای یادگیر با ساختار متغیر و اتوماتاهای یادگیر با ساختار ثابت می توان به [2,3,4] مراجعه کرد.

۲-۲- اتوماتای یادگیر توزیع شده

اتوماتای یادگیر توزیع شده (DLA) شبکه ای از اتوماتاهای یادگیر است که برای حل یک مسئله با یکدیگر همکاری می نمایند [4]. تعداد اعمال یک اتوماتا در DLA برابر تعداد اتوماتاهای متصل به این اتوماتای یادگیر می باشد. انتخاب یک عمل توسط یک اتوماتای یادگیر در شبکه، اتوماتای یادگیر متناظر با این عمل در طرف دیگر یال را فعال می سازد. به عنوان مثال در شکل ۱، هر اتوماتای یادگیر دارای دو عمل می باشد. انتخاب عمل α_2 توسط LA_1 اتوماتای یادگیر LA_2 را فعال خواهد کرد. سپس اتوماتای یادگیر فعال شده (LA_2) یکی از اعمال



انتخاب گره بعدی (V_{i+1}) برای V_i در این الگوریتم طبق شکل ۳ انجام می شود:

- ۱- برای تمام یال های واقع در گره V_i ، مقدار احتمال هر یال را با توجه به عکس طول یال محاسبه کن.
- ۲- یکی از یالهای گرهی را که بر روی آن قرار داری به تصادف انتخاب کن.
- ۳- برای تمام یال های متصل به گره V_i موارد زیر را انجام بده:
- اگر یال مورد نظر به عنوان مسیر انتخاب شده است: بر اساس معادله ۲-ب به آن پاداش بده.
- در غیر اینصورت: طبق معادله ۲-الف، آن را جریمه کن.

شکل ۳- الگوریتم انتخاب گره بعدی (V_{i+1}) در طراحی مسیر عامل ها

الگوی یادگیری اتوماتاهای استفاده شده در این الگوریتم LRP است که در آن مقادیر پارامترهای a و b با هم برابر است. برای محاسبه مقدار این پارامترها از روش آزمایش و خطا استفاده شده است. جدول زیر نتایج حاصل از شبیه سازی به ازای مقادیر مختلف پارامترها را در مقاطع مختلف زمانی نشان می دهد. با توجه به مقادیر جدول ۳ می توان گفت مقدار مناسب a و b در این مسئله، مقدار ۳، ۱۰ است.

۴- ارزیابی

برای ارزیابی کارایی الگوریتم ارائه شده از دو سناریوی متفاوت استفاده شده است: نقشه RandomSmall-1 (مسابقات روبوکاپ ۲۰۰۸ چین) و نقشه دیگر که بخشی از گراف مربوط به شهر Kobe می باشد. اهداف انتخاب شده برای عاملها در تمام سناریوها و تمام شرایط کاملاً یکسان است. نتایج یادداشت شده برای سناریوی Kobe مربوط به ۶۰ سیکل اول شبیه سازی با دوره های زمانی ۱۵ ثانیه ای می باشد و برای ۲۰۰ ثانیه دوم با دوره های ۵۰ ثانیه ای یادداشت شده است. عملکرد الگوریتم ارائه شده (از نظر exploration) با عملکرد الگوریتم استفاده شده در تیم Persia که از کوتاهترین مسیر استفاده می کند مقایسه شده است [8,9]. شکل های ۴ و ۵ این مقایسه را به ترتیب در

انتخاب هر یک از خیابان های متصل به تقاطع نیز با توجه به طول مسیر از آن خیابان و نیز قانون احتمال محاسبه می شود. همچنین اتوماتای یادگیر حق انتخاب خیابانی را که عامل از طریق آن به تقاطع وارد می شود را ندارد و عملی را که متناظر با آن است غیرفعال می کند. اتوماتاهای دیگر نیز اگر دارای این عمل در مجموعه اعمال خود هستند آن را غیرفعال می کنند. با انتخاب یکی از خیابانها، اتوماتای یادگیر واقع در طرف دیگر خیابان انتخاب شده که در یکی از تقاطع های همسایه واقع است فعال می شود و همین روال تا رسیدن عامل به مقصد ادامه دارد. نکته ای که وجود دارد این است که هر یک از اتوماتاهای یادگیر که فعال می شوند، عمل انتخاب شده خودشان را مطابق فرمول ۲-الف جریمه می کنند. این عمل باعث می شود تا هر خیابانی که به عنوان مسیر انتخاب می شود و عامل از آن عبور می کند در آینده شانس کمتری برای انتخاب شدن به عنوان مسیر حرکت عامل داشته باشد. همچنین اتوماتای یادگیر فعال، اعمال انتخاب نشده خودشان را مطابق فرمول ۲-ب پاداش می دهند تا این خیابانها در آینده شانس بیشتری برای انتخاب شدن به عنوان مسیر عامل داشته باشند. جریمه و پاداش اتوماتای یادگیر باعث می شود تا از تمام خیابان های متصل به تقاطع متناظر با اتوماتای یادگیر استفاده شود. از آنجا که احتمال انتخاب شدن یک خیابان، بر اساس طول مسیر عبوری از آن خیابان می باشد بنابراین با خیابانها با احتمال بالایی به صورت صعودی و بر اساس طول مسیر انتخاب خواهند شد. یعنی برای بار اول که عامل از یک تقاطع عبور می کند، اولین کوتاهترین مسیر و برای بار دوم، دومین کوتاهترین مسیر به صورت احتمالی پیشنهاد می شود. هر یالی که انتخاب می شود بر اساس قوانین اتوماتای یادگیر جریمه می شود و یال هایی که انتخاب نشده اند پاداش می گیرند. این عمل باعث می شود احتمال انتخاب یال ملاقات شده در آینده کمتر شده و یال های ملاقات نشده احتمال بیشتری برای انتخاب شدن در آینده داشته باشند. هر عامل سعی می کند در آینده از یالهایی که عبور کرده است عبور نکند.

۵- نتیجه‌گیری

در این مقاله مسیریابی عامل‌های پلیس در سیستم شبیه‌ساز امداد، با استفاده از اتوماتای یادگیر توزیع شده انجام شد. شبیه‌سازی‌های انجام گرفته و نتایج بدست آمده نشان می‌دهند الگوریتم ارائه شده قدرت کاوش عامل‌ها را در مقایسه با الگوریتم‌های بر مبنای کوتاهترین مسیر افزایش می‌دهد. همچنین مسیرهای انتخاب شده توسط اتوماتای یادگیر توزیع شده با احتمال زیاد دارای حداقل اختلاف با کوتاهترین مسیر هستند. یعنی در یک گره برای یک مقصد خاص، برای بار اول، اولین مسیر کوتاه به عنوان مسیر انتخاب می‌شود و برای بار دوم، دومین مسیر کوتاه و الی آخر. با توجه به نتایج بدست آمده می‌بینیم نقطه اوج کاوش برای الگوریتم پیشنهاد شده در شروع شبیه‌سازی بوده و رفته رفته تقریباً به صورت منظم کم می‌شود. موارد فوق نشان می‌دهند الگوریتم ارائه شده از نظر کاوش و جستجو در محیط نسبت به الگوریتم‌های دیگر از سرعت و قدرت بیشتری برخوردار است.

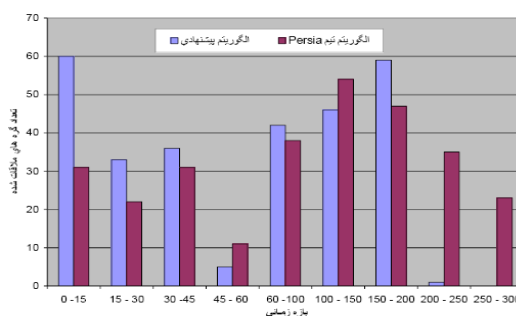
مراجع

- [1] S. B. M. Post and M. L. Fassaert, "A Communication and Coordination Model For 'RobocupRescue' Agents", *M.Sc. thesis, Department of Computer Science, University of Amsterdam*, 2004.
- [2] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction*, Prentice Hall, Inc., 1989.
- [3] M. A. L. Thathachar, P. S. Sastry, "Varieties of Learning Automata: An Overview", *IEEE Transaction on System, Man and Cybernetics, Part B*, Vol. 32, No. 6, P. 711-722, 2002.
- [4] S. Lakshmivarahan, "Learning Algorithms: Theory and Applications", New York: Springer-verlag, 1981.
- [5] M. R. Meybodi and H. Beigy, "Solving Stochastic Shortest Path Problem Using Distributed Learning Automata", *Proceedings 6th Annual CSI Computer Conference, University of Isfahan's Computer Engineering Department*, 2001.
- [6] H. Beigy and M. R. Meybodi "A New Distributed Learning Automata for Solving Stochastic Shortest Path Problem", *Proceedings of the Sixth*

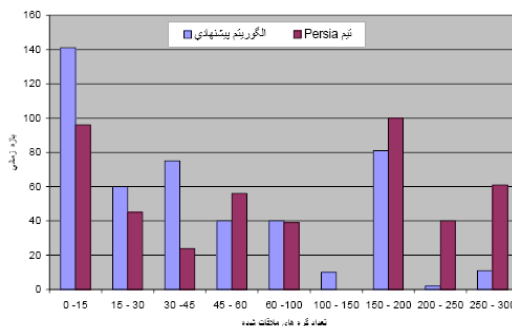
سناریو های Kobe و RandomSmall نشان می‌دهند.

جدول ۱- مقادیر مختلف برای پارامترهای a و b و تاثیر آن در تعداد گره‌های ملاقات شده توسط عامل در زمان‌های مختلف

زمان	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
15 - 0	139	125	178	75	167	139	155	184
30 - 15	8	40	47	66	25	50	33	8
45 - 30	63	41	5	24	51	85	57	15
60 - 45	6	9	18	65	6	15	11	25
75 - 60	42	3	29	60	5	1	0	4
90 - 75	22	0	15	6	14	0	0	42
105 - 90	12	19	0	0	22	0	25	0
120 - 105	0	26	15	15	0	1	0	0
135 - 120	0	5	0	0	0	1	0	0
150 - 135	0	0	0	0	0	0	0	0



شکل ۴- مقایسه الگوریتم ارائه شده با الگوریتم تیم Persia از نظر قدرت کاوش - سناریوی Kobe



شکل ۵- مقایسه الگوریتم ارائه شده با الگوریتم تیم Persia از نظر قدرت کاوش - سناریوی RandomSmall



- International Joint Conference on Information Science*, Durham, USA, pp. 339- 343, 2002,
- [7] M. Alipour and M. R. Meybodi, "Solving Traveling salesman Problem Using Distributed Learning Automata" , *Proceedings of 10th Annual CSI Computer Conference, Computer Engineering Department, Iran Telecommunication Research Center*, Tehran, Iran, pp. 271-280, Feb. 2005.
- [8] M. R. Khojasteh and H. Heidari, "Persia 2005 Team Description", *Team Description Paper*, 2005.
- [9] M. R. Khojasteh, A. Kazimi and Z. Ghaseminik, "Persia 2006, Towards a Full Learning Automata-Based Cooperative Team", *Team Description Paper*, 2006.
- [10] The Robocup Rescue Technical Committee, "Robocup-Rescue Simulator Manual", 2000, Web page: <http://robomec.cs.kobe-u.ac.jp/robocup-rescue>.
- [11] M. Bowling, "Robocup Rescue: Agent Development Kit", Version0.4, 2001.