حل فرآيندهاي تصميم‌گيري مارکف چندعامله با استفاده از اتوماتاهاي يادگير

فرناز ابطحي   محمدرضا ميبدي

آزمايشگاه محاسبات نرم

دانشکده مهندسي کامپيوتر و فناوري اطلاعات

دانشگاه صنعتي اميرکبير

تهران   ايران

abtahi@aut.ac.ir   mmeybodi@aut.ac.ir

چکيده: فرآيندهاي تصميم گيري مارکف چندعامله بطور گسترده براي مدل کردن سيستم‌هاي چندعامله بکار مي‌رود. در اين مقاله الگوريتمهايي مبتني بر اتوماتاهاي يادگير براي حل فرآيندهاي تصميم گيري مارکف چندعامله و يافتن خط‌مشي بهينه پيشنهاد مي‌گردد. در الگوريتم‌هاي پيشنهادي محيط مسئله مارکف به صورت يک گراف جهت‌دار مدل مي‌شود که در آن، گره‌هاي گراف نشان‌دهنده وضعيت‌ها و يال‌هاي جهت‌دار بين گره‌ها به منزله اعمالي مي‌باشند که منجر به انتقال از وضعيتي به وضعيت ديگر مي‌گردند. هر گره از گراف، به يک اتوماتاي يادگير مجهز مي‌شود که اعمال آن، يال‌هاي خروجي آن گره مي‌باشد. عامل‌ها بر روي اين گراف حرکت کرده، از يک گره به گره ديگر حرکت مي‌کنند و در هر گره، از اتوماتاي يادگير متناظر با آن گره براي انتخاب عمل و انتقال به وضعيت بعدي کمک مي‌گيرند. با توجه به مسير طي شده توسط عامل‌ها و ميزان مطلوبيت مسير از لحاظ هزينه و مدت زمان لازم براي رسيدن به وضعيت هدف، بردار احتمال اعمال اتوماتاهاي يادگير بهنگام مي‌شود و از اين طريق اتوماتاهاي يادگير به‌تدريج به اعمال بهينه خود همگرا مي‌گردند. آزمايشهاي انجام گرفته نشان داده‌اند که الگوريتم‌هاي ارائه شده، در مقايسه با روش‌هاي مبتني بر اتوماتاهاي يادگير موجود از کارايي بالاتري از هر دو جنبه هزينه و سرعت رسيدن به راه‌حل بهينه برخوردار هستند.

کلمات کليدي: سيستمهاي چند عامله، فرآيند تصميم‌گيري مارکف چندعامله، اتوماتاهاي يادگير، خط‌مشي بهينه

Solving Multi-Agent Markov Decision Processes Using Learning Automata

F. Abtahi   M. R. Meybodi

Soft Computing Laboratory
Department of Computer Engineering and Information Technology
Amirkabir University of Technology, Tehran ,Iran

abtahi@aut.ac.ir   mmeybodi@aut.ac.ir

Abstract: Multi-Agent Markov Decision Processes (MMDPs) are widely used to model many types of multi-agent systems. In this paper, several algorithms based on learning automata for solving MMDPs and finding a policy for selecting actions are proposed. In the proposed algorithms, Markov problem is described as a directed graph. The nodes of this graph are the states of the problem, and the directed edges represent the actions that result in transition from one state to another. Each node in the graph is equipped with a learning automaton whose actions are the outgoing edges of that node. Each agent moves from one node to another and tries to reach the goal state. In each node, the agent with the help of the learning automaton in that node chooses the next transition. The actions taken by the learning automata along the path traveled by the agent is then rewarded or penalized based on the cost of the traveled path according to a learning algorithm. This way the optimal policy for the agent will be gradually reached. The results of experiments have shown that the proposed algorithms perform better than the existing learning automata based algorithms in terms of cost and the speed of reaching the optimal policy.

Keywords: Multi- Agent Systems, Multi-Agent Markov Decision Process, Learning Automata, Optimal Policy

۱- مقدمه

تا به حال مدل‌هاي مختلفي براي سيستم‌هاي چندعامله پيشنهاد شده است. يكي از اين مدل‌ها، فرآيند تصميم‌گيري ماركف چندعامله[1] مي‌باشد. راه‌حل فرآيند تصميم‌گيري ماركف چندعامله يك خط‌مشي[2] است كه سياستي براي انتخاب اعمال توسط عاملها در هر وضعيت ميباشد[۱]. يك فرآيند تصميم‌گيري ماركف *(MDP)* تك‌عامله را مي‌توان به‌صورت چندتايي $< S, A, P, R >$ تعريف كرد كه $S = \{i\}$ و $A = \{a\}$، به‌ترتيب مجموعه متناهي وضعيت‌ها و اعمال مي‌باشند. $P : S \times A \times S \to [0,1]$ تابع انتقال را تعريف مي‌نمايد؛ به‌اين معنا كه احتمال اين‌كه عامل با انجام عمل a در وضعيت i، به وضعيت j تغيير حالت دهد، برابر $P(i,a,j)$ است. $R : S \to \mathbb{R}$ تابع پاداش مي‌باشد؛ يعني هنگامي‌كه عامل، وضعيت i را مشاهده مي‌كند، پاداشي به اندازه $R(i)$ دريافت مي‌نمايد[۹]. راه‌حل يک مسئله *MDP*، يك خط‌مشي است كه به‌صورت رويه‌اي براي انتخاب اعمال تعريف مي‌شود. براي چنين*MDP* هايي، هميشه خط‌مشي‌هايي وجود دارند كه بهينه‌يكنواخت بوده (با شروع از هر وضعيتي بهينه هستند)، پايدار، مستقل از زمان، قطعي (هميشه در يك وضعيت خاص، عمل مشخصي انتخاب مي‌شود) و مستقل از آن‌چه از گذشته تا به حال انجام شده مي‌باشند. اين سياست‌ها، كه آن‌ها را با π نشان مي‌دهيم، مي‌توانند به‌صورت نگاشتي از وضعيت‌ها به اعمال به‌شكل $\pi : S \to A$ تعريف شوند.

اكنون محيطي چندعامله را درنظر مي‌گيريم كه داراي مجموعه‌اي از n عامل به‌صورت $M = \{m\}$ مي‌باشد. هر عامل، مجموعه وضعيت‌هاي $S_m = \{i_m\}$ و مجموعه اعمال $A_m = \{a_m\}$ را دارد. ساده‌ترين و رايج‌ترين راه براي توسعه مفهوم *MDP* تك‌عامله به حالت چندعامله اين است كه فرض كنيم هريك از عامل‌ها روي تغيير وضعيت و پاداش ساير عامل‌ها مؤثر مي‌باشد. تحت اين شرايط، يك *MDP* چندعامله را مي‌توان به‌راحتي به‌شكل يك *MDP* بزرگ $< S_M, A_M, P_M, R_M >$ تعريف نمود كه در آن، فضاي وضعيت‌هاي جمعي S_M، به‌صورت ضرب خارجي فضاي وضعيت تمام عامل‌ها، يعني $S_M = S_1 \times \ldots \times S_n$، و فضاي اعمال جمعي A_M، به‌صورت $A_M = A_1 \times \ldots \times A_n$ تعريف مي‌شوند. در اين حالت، تابع انتقال و تابع پاداش روي فضاي اعمال جمعي و فضاي وضعيت جمعي عامل‌ها، به‌شكل استاندارد $P_M : S_M \times A_M \times S_M \to [0,1]$ و $R_M : S_M \to \mathbb{R}$ خواهد بود. [۱۰] روش نمايش بالا، كه به آن روش مسطح مي‌گوييم، عمومي‌ترين روش مي‌باشد؛ زيرا با داشتن فضاي وضعيت و اعمال گروهي، عامل‌ها مي‌توانند به‌طور دلخواه با يكديگر در تعامل باشند. فرآيندهاي ماركف چندعامله، بازي‌هاي ماركف و يا زنجيره‌هاي ماركف نيز ناميده مي‌شوند.

براي يافتن خط‌مشي بهينه در فرآيند تصميم‌گيري ماركف چندعامله، از روش‌هاي مختلفي مي‌توان استفاده نمود. براي مثال در [۳]، الگوريتمي برمبناي يادگيري*Q* براي حل اين مسائل مورد استفاده قرار گرفته است. در [۴] روشي براي حل *MDP* با تجزيه آن به *MDP*هاي محلي، حل آن‌ها به‌صورت مستقل، و تخمين راه‌حل كلي با توجه به راه‌حل *MDP* هاي محلي ارائه گرديده است. همچنين در [۵] روشي برخط[3] براي حل مسائل *MDP* مطرح شده كه در آن، ابتدا يک خط‌مشي تقريبي مثلاً به‌صورت حريصانه[4] براي *MDP* درنظر گرفته مي‌شود و سپس با روش‌هاي جستجوي برخط، اين خط‌مشي تصفيه شده و به خط‌مشي بهينه نزديک مي‌گردد. يكي از روش‌هايي كه براي چنين كاربردي مناسب بوده و برمبناي يادگيري تقويتي عمل مي‌كند، اتوماتاي يادگير مي‌باشد. در [۶]، مدلي مبتني بر اتوماتاي يادگير اتصالي[5] براي حل مسائل ماركف چندعامله به‌كار رفته است.

در اين مقاله الگوريتم‌هايي مبتني بر اتوماتاهاي يادگير براي حل فرآيندهاي تصميم گيري ماركف چندعامله و يافتن خط‌مشي بهينه پيشنهاد مي‌گردد. در الگوريتم‌هاي پيشنهادي محيط مسئله ماركف به‌صورت يک گراف جهت‌دار مدل مي‌شود كه در آن، گره‌هاي گراف نشان-دهنده وضعيت‌ها و يال‌هاي جهت‌دار بين گره‌ها به منزله اعمالي مي‌باشند كه منجر به انتقال از وضعيتي به وضعيت ديگر مي‌گردند. هر گره از گراف، به يک اتوماتاي يادگير مجهز مي‌شود كه اعمال آن، يال‌هاي خروجي آن گره مي‌باشد. عامل‌ها بر روي اين گراف حركت كرده و در هر گره، از اتوماتاي يادگير متناظر با آن گره براي انتخاب عمل و انتقال به وضعيت بعدي كمک مي‌گيرند. با توجه به مسير طي شده توسط عامل‌ها و ميزان مطلوبيت مسير از لحاظ هزينه و مدت زمان لازم براي رسيدن به وضعيت هدف، بردار احتمال اعمال اتوماتاهاي يادگير بهنگام مي‌شود. اين فرآيند كه به‌طور موازي توسط تمام عامل‌ها اجرا مي‌شود، براي هر عامل بارها تكرار مي‌گردد تا جايي‌كه مسير طي‌شده توسط هريک از آن-

[1] Multi-Agent Markov Decision Process
[2] Policy
[3] Online
[4] Greedy
[5] Interconnected Learning Automata

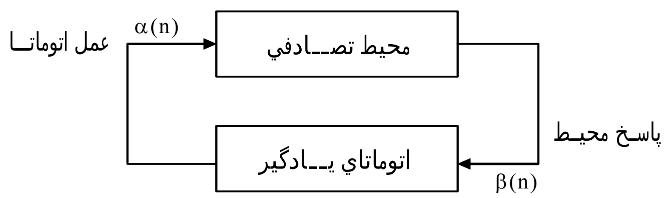
ها به تدريج به سمت بهترين مسير همگرا شود. آزمايش‌هاي انجام گرفته نشان داده‌اند كه الگوريتم‌هاي ارائه شده در مقايسه با روش‌هاي موجود مبتني بر اتوماتاهاي يادگير از كارايي بالاتري از جنبه هزينه و سرعت رسيدن به راه‌حل بهينه برخوردار هستند. ادامه مقاله بدين‌صورت سازماندهي شده است. در بخش۲ مروري كوتاه بر اتوماتاهاي يادگير خواهيم داشت. در بخش۳ به چگونگي حل فرآيندهاي تصميم‌گيري ماركف چندعامله با استفاده از اتوماتاهاي يادگير و نتايج آزمايش‌ها مي‌پردازيم. بخش۴ به نتيجه گيري اختصاص دارد.

۲- اتوماتاي يادگير

اتوماتاي يادگير، ماشيني است كه مي‌تواند تعدادي متناهي عمل را انجام دهد. هر عمل انتخاب شده توسط یک محيط احتمالي ارزيابي مي‌شود، نتيجه ارزيابي در قالب سيگنالي مثبت يا منفي به اتوماتا داده مي‌شود و اتوماتا از اين پاسخ در انتخاب عمل بعدي تأثير مي‌گيرد. هدف نهايي اين است كه اتوماتا ياد بگيرد تا از بين اعمال خود، بهترين عمل را انتخاب كند. بهترين عمل، عملي است كه احتمال دريافت پاداش از محيط را به حداكثر برساند. كاركرد اتوماتاي يادگير در تعامل با محيط، در شكل۱ مشاهده مي‌شود. [۶،۱۱،۱۲]

محيط را مي‌توان توسط سه تايي $E \equiv \{\alpha, \beta, c\}$ نشان داد كه در آن $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه ورودي‌ها، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ مجموعه خروجي‌ها و $c \equiv \{c_1, c_2, ..., c_r\}$ مجموعه احتمال‌هاي جريمه مي‌باشد. هرگاه β مجموعه‌اي دوعضوي باشد، محيط از نوع P است. در چنين محيطي $\beta_1 = 1$ به عنوان جريمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته مي‌شود. در محيط از نوع Q، $\beta(n)$ مي‌تواند به طور گسسته يك مقدار از مقادير محدود در فاصله [۰،۱] را اختيار كند و در محيط از نوع S، $\beta(n)$ متغير تصادفي در فاصله [۰،۱] است. c_i احتمال اين‌كه عمل α_i نتيجه نامطلوب داشته باشد مي‌باشد. در محيط ايستا، مقادير c_i بدون تغيير مي‌مانند، در حالي‌كه در محيط غيرايستا اين مقادير در طي زمان تغيير مي‌كنند.

اتوماتاي يادگير با ساختار ثابت توسط پنج‌تايي $\{\alpha, \beta, F, G, \phi\}$ نشان داده مي‌شود كه در آن $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه عمل‌هاي اتوماتا، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_r\}$ مجموعه ورودي‌هاي اتوماتا، $\phi(n) \equiv \{\phi_1, \phi_2, ..., \phi_k\}$ مجموعه وضعيت‌هاي داخلي اتوماتا در لحظه n، $F : \phi \times \beta \rightarrow \phi$ تابع توليد وضعيت جديد اتوماتا و $G : \phi \rightarrow \alpha$ تابع خروجي مي‌باشد كه وضعيت كنوني اتوماتا را به خروجي بعدي مي‌نگارد.



شكل ۱- ارتباط بين اتوماتاي يادگير و محيط

اتوماتاي يادگير با ساختار متغير را مي‌توان توسط چهارتايي $\{\alpha, \beta, p, T\}$ نشان داد كه $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه عمل‌هاي اتوماتا، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_r\}$ مجموعه ورودي‌هاي اتوماتا، $p = \{p_1, .., p_r\}$ بردار احتمال انتخاب هريک از عمل‌ها و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوريتم يادگيري مي‌باشد. الگوريتم زير یک نمونه از الگوريتم‌هاي يادگيري خطي است. فرض مي‌كنيم عمل α_i در مرحله nام انتخاب شود.

- پاسخ مطلوب از محيط

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1-a)p_j(n) \quad \forall j \quad j \neq i \end{aligned} \tag{۱}$$

- پاسخ نامطلوب از محيط

$$\begin{aligned} p_i(n+1) &= (1-b)p_i(n) \\ p_j(n+1) &= (b/r - 1) + (1-b)p_j(n) \quad \forall j \quad j \neq i \end{aligned} \tag{۲}$$

در روابط (۱) و (۲)، a پارامتر پاداش و b پارامتر جريمه مي‌باشند. با توجه به مقادير a و b سه حالت را مي‌توان درنظر گرفت: اگر a و b با هم برابر باشند، الگوريتم را L_{RP} ، هنگامي‌كه b از a بسيار كوچک‌تر باشد، الگوريتم را $L_{R\varepsilon P}$ و اگر b مساوي صفر باشد آن را L_{RI} مي‌ناميم.

اتوماتاي يادگير داراي تعداد عمل ثابتي است، اما در بعضي از كاربردها نياز به اتوماتايي با تعداد عمل متغير مي‌باشد. اين اتوماتا در لحظه n، عمل خود را فقط از يک زير مجموعه غير تهي ($V(n)$) از اعمال، كه اعمال فعال ناميده مي‌شوند، انتخاب مي‌كند. انتخاب مجموعه $V(n)$ توسط يک عامل خارجي و به‌صورت تصادفي انجام مي‌شود. نحوه كار اين اتوماتا به اين صورت است: براي انتخاب يک عمل در زمان n، ابتدا اتوماتا مجموع احتمال اعمال فعال خود ($K(n)$) را محاسبه كرده و سپس بردار $p(n)$ را مطابق رابطه۳ محاسبه مي‌كند. آنگاه اتوماتا يک عمل مانند α_i از مجموعه اعمال فعال خود را به‌صورت تصادفي و مطابق بردار احتمال $p(n)$ انتخاب كرده و بر محيط اعمال مي‌نمايد. پس از دريافت پاسخ محيط، اتوماتا بردار احتمال $p(n)$ اعمال خود درصورت دريافت پاداش بر اساس رابطه۴ و درصورت دريافت جريمه طبق رابطه۵ به‌روز مي‌كند.

$$p_i(n) = prob[\alpha(n) = \alpha_i \mid V(n) \text{ is the set of active actions}, \alpha_i \in V(n)] = \frac{p_i(n)}{K(n)} \qquad (۳)$$

- پاسخ مطلوب از محيط

$$\begin{aligned} p_i(n+1) &= p_i(n) + a.(1 - p_i(n)) && \alpha(n) = \alpha_i \\ p_i(n+1) &= p_j(n) + a.p_i(n) && \alpha(n) = \alpha_i,\ \forall j \quad j \neq i \end{aligned} \qquad (٤)$$

- پاسخ نامطلوب از محيط

$$\begin{aligned} p_i(n+1) &= (1-b).p_i(n) && \alpha(n) = \alpha \\ p_i(n+1) &= \frac{b}{r-1} + (1-b)p_j(n) && \alpha(n) = \alpha_i,\ \forall j \quad j \neq i \end{aligned} \qquad (۵)$$

سپس اتوماتا بردار احتمال اعمال $p(n)$ را با استفاده از بردار $p(n+1)$ و به‌صورت زير به‌روز مي‌نمايد:

$$\begin{aligned} p_j(n+1) &= p_j(n+1).K(n) && \text{for all j}, \alpha_j \in V(n) \\ p_j(n+1) &= p_j(n) && \text{for all j}, \alpha_j \notin V(n) \end{aligned} \qquad (۶)$$

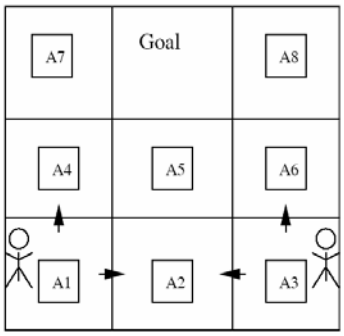
۳- كنترل فرآيند تصميم‌گيري ماركف چندعامله با استفاده از اتوماتاهاي يادگير

مسئله كنترل فرآيند تصميم‌گيري ماركف (زنجيره ماركف) تک‌عامله را مي‌توان با استفاده از شبكه‌اي از اتوماتاهاي يادگير متصل به هم مدل نمود كه در آن، كنترل از يک اتوماتاي يادگير به اتوماتاي يادگير ديگر منتقل مي‌شود. هر وضعيت از زنجيره ماركف داراي يک اتوماتاي يادگير است كه طي مراحل فرآيند و با استفاده از قوانين (۱) و (۲) سعي مي‌كند توزيع احتمال بهينه اعمال در آن وضعيت را ياد بگيرد. بنابراين در اين مدل، اعمال متعلق به وضعيت‌ها مي‌باشد و نه متعلق به عامل‌ها. عامل با شروع از وضعيت اوليه تا رسيدن به هدف بر روي اين شبكه از اتوماتاهاي يادگير حركت مي‌كند و در هر وضعيت، از اتوماتاي يادگير در آن وضعيت براي انتقال به يكي از وضعيت‌هاي مجاور كمک مي‌گيرد كه اين كار با استفاده از بردار احتمال اعمال اتوماتاي يادگير انجام مي‌شود. در هر لحظه فقط يک اتوماتاي يادگير فعال بوده و انتقال از يک وضعيت به وضعيت ديگر، اتوماتاي مربوط به وضعيت جديد را فعال مي‌نمايد. اين فرآيند تا زماني‌كه بردار احتمال‌هاي كليه اتوماتاهاي يادگير به پايداري برسد و يا شرط خاصي برقرار گردد تكرار مي‌شود. با افزايش تعداد عامل‌ها و توسعه مدل به حالت چندعامله، لازم است بيش از يک اتوماتاي يادگير در هر لحظه فعال گردند، زيرا با توجه به اين‌كه وضعيت‌ها وابسته به مسئله و محيط مي‌باشند، عامل‌ها ممكن است در وضعيت‌هاي مختلفي قرار داشته باشند.

در اين مقاله، دو الگوريتم جديد براي حل مسائل ماركف چندعامله ارائه مي‌شود. در اين الگوريتم‌ها، محيط مسئله ماركف روي يک گراف جهت‌دار نگاشت مي‌گردد. اين نگاشت به‌صورتي انجام مي‌شود كه گره‌هاي گراف نشان‌دهنده وضعيت‌هاي مسئله و يال‌هاي جهت‌دار بين گره‌ها

نشان‌دهنده اعمالي مي‌باشند كه منجر به انتقال عامل از وضعيتي به وضعيتي ديگر مي‌گردند. سپس در هر گره از اين گراف، يک اتوماتاي يادگير قرار مي‌گيرد که متناظر با وضعيتي است که روي آن گره نگاشت شده است. عامل‌ها با شروع از وضعيت اوليه تا رسيدن به وضعيت هدف، به‌طور موازي بر روي گراف متناظر با مسئله حركت مي‌كنند. در هر وضعيت، عامل‌ها از اتوماتاي يادگير آن وضعيت در انتخاب جهت حرکت کمک مي‌گيرند و با توجه به بردار احتمال اعمال اتوماتا، به وضعيت جديدي منتقل مي‌گردند. حرکت هر عامل روي گراف تا جايي تكرار مي‌شود كه به وضعيت هدف برسد. سپس با توجه به مسير طي‌شده توسط عامل و ميزان مطلوبيت مسير از لحاظ زمان و هزينه مصرفي، احتمال انتخاب هريک از اعمال براي اتوماتاهاي متناظر با وضعيت‌هايي که عامل در طول مسير از آن‌ها عبور نموده، به‌روز مي‌شود. اين فرآيند براي هر عامل بارها تکرار مي‌گردد تا جايي‌که مسير طي‌شده توسط هريک از آن‌ها به تدريج به سمت بهترين مسير همگرا شود و يا يك معيار از پيش تعيين شده‌اي ارضاء گردد.

براي بررسي عملکرد شبکه اتوماتاي يادگير در کنترل زنجيره مارکف آزمايشي را انجام داده‌ايم كه در آن، محيط عملياتي يک صفحه ۳×۳ با ۹ خانه مي‌باشد و دو عامل با شروع از دو محل اوليه روي اين صفحه، سعي دارند خود را به خانه هدف برسانند. اين فرآيند چندعامله ابتـدا در [۳] و سپس در [۶] مطرح گرديده است. اين محيط، محل اوليه عامل‌ها و محل خانه هدف در شكل۲ نشان داده شده است.



شكل ۲- فرآيند تصميم‌گيري ماركف چندعامله مورد استفاده در آزمايش

براي تبديل مسئله به زنجيره ماركف، هر خانه از صفحه را يک وضعيت فرض كرده و انتقال‌هاي مجاز به خانه‌هاي مجاور را متناظر با اعمال در هر وضعيت درنظر مي‌گيريم. در اين زنجيره ماركف، هدف اين است كه هر دو عامل به خانه هدف برسند و براي رسيدن به آن مسيرهاي مجزايي را انتخاب نمايند، زيرا هنگامي كه هر دو هم‌زمان تصميم به ورود به خانه يكساني (به‌جز خانه هدف) مي‌گيرند، از حركت آن‌ها جلوگيري مي‌شود. علاوه بر اين هدف، مطلوب است كه عامل‌ها كوتاه‌ترين مسيرهاي ممكن را طي نمايند.

بنابراين، اين فرآيند چندعامله از چند جهت قابل توجه است؛ اول اين‌که اين فرآيند، يک زنجيره مارکف است که مي‌تواند توسط شبکه‌اي از اتوماتاهاي يادگير حل شود؛ دوم اين‌که اين مسئله را مي‌توان يک مسئله بهينه‌سازي چندعامله براي يافتن کوتاه‌ترين مسير دانست که در آن، عامل‌ها براي پيدا کردن مسير بهينه با يکديگر همکاري مي‌کنند؛ و سوم اين‌که به دليل وجود محدوديت در حرکت عامل‌ها هنگامي که هم‌زمان قصد حرکت به خانه يکساني را دارند، ايجاد هماهنگي بين عامل‌ها لازم مي‌باشد.

اولين الگوريتم پيشنهادي، که آن را الگوريتم۱ مي‌ناميم، به شرح زير مي‌باشد.

الگوريتم۱:

۱- گراف جهت‌داري متناظر با مسئله ايجاد شده، وضعيت‌ها روي گره‌ها و اعمال روي يال‌ها نگاشت مي‌شوند.

۲- در هر گره از گراف، يک اتوماتاي يادگير از نوع L_{RP} قرار مي‌گيرد. مجموعه اعمال اين اتوماتاي يادگير برابر با حرکت‌هاي مجاز به وضعيت‌هاي بعدي در نظر گرفته مي‌شود. اعضاي بردار احتمال هر يك از اتوماتاهاي يادگير مساوي و برابر يك تقسيم بر تعداد حركت-هاي مجاز به وضعيت‌هاي بعدي مقدار دهي اوليه مي‌شود.

۳- مراحل زير به‌طور موازي براي هر يک از عامل‌ها اجرا مي‌گردد.

۳-۱- عامل j در وضعيت شروع قرار مي‌گيرد.

۳-۲- حد آستانه پويا T_k^j براي عامل برابر صفر قرار داده مي‌شود.

۳-۳- در مرحله k^jام:

۳-۳-۱- با قرار گرفتن عامل j در وضعيت s، اتوماتاي يادگير A_s متناظر با اين وضعيت فعال شده و عامل براي انتقال به وضعيت بعدي از آن کمک مي‌گيرد. بدين منظور، با توجه به بردار احتمال اتوماتا، يک عمل مانند a_m را که متناظر با يال جهت‌دار (s,m) از گراف و به معناي حرکت به وضعيت بعدي m است انتخاب مي‌کند.

۳-۳-۲- عامل j به وضعيت m منتقل مي‌شود.

۳-۳-۳- $s \leftarrow m$ و به مرحله ۳-۳-۱ بازمي‌گرديم تا زماني که عامل j به وضعيت هدف برسد.

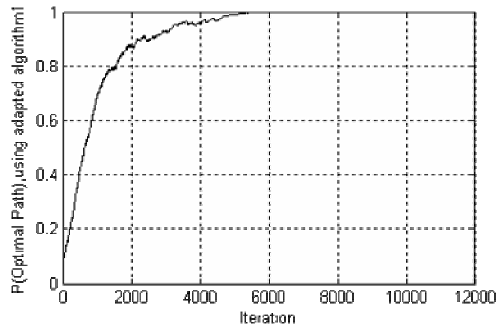
۳-۳-۴- با رسيدن عامل j به وضعيت هدف، هزينه مسير طي‌شده π_i، طبق رابطه $L_{\pi_i}^j = R_G / t_{\pi_i}^j$ محاسبه مي‌شود. در اين رابطه، R_G پاداش مربوط به رسيدن به وضعيت هدف و t_{π_i} مدت زمان سپري شده از شروع حرکت عامل j تا رسيدن آن به وضعيت هدف، با طي مسير π_i مي‌باشد.

۳-۳-۵- اگر شرط $L_{\pi_i}^j < T_k^j$ برقرار باشد، آنگاه به مسير طي‌شده پاداش داده مي‌شود و در غير اين صورت مسير طي‌شده جريمه مي‌شود. ميزان پاداش را برابر با $L_{\pi_i}^j$ و ميزان جريمه را برابر با $-L_{\pi_i}^j$ درنظر مي‌گيريم. اين پاداش يا جريمه به هريک از اتوماتاهايي که در مسير مشارکت داشته‌اند تعلق مي‌گيرد و بردار احتمال آن‌ها بهنگام مي‌گردد.

۳-۳-۶- $k^j \leftarrow k^j + 1$ و $T_k^j \leftarrow T_{k-1}^j + \frac{1}{k^j}(L_{\pi_i}^j - T_{k-1}^j)$.

۳-۳-۷- شرط خاتمه الگوريتم (رسيدن احتمال مسير بهينه براي عامل j به حد آستانه) بررسي مي‌شود. در صورت عدم برقراري شرط خاتمه، به مرحله ۳-۲ باز مي‌گرديم.

اگر احتمال مسير بهينه از حاصل‌ضرب احتمال يال‌هاي آن به‌دست آيد و حد آستانه مطلوب براي اين احتمال ۰/۹۵ درنظر گرفته شود، با استفاده از اين الگوريتم با نرخ يادگيري ۰/۰۱، همگرايي به مسير بهينه به‌طور متوسط پس از ۲۱۰۰ بار تکرار اتفاق مي‌افتد. اين مسئله را مي‌توان در نمودار شکل۳ مشاهده نمود.

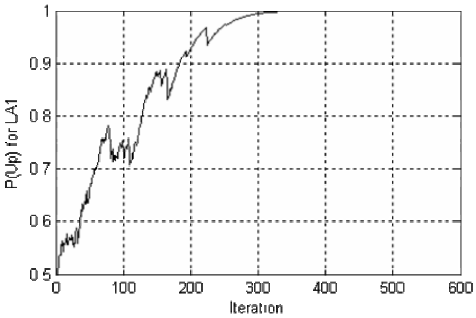


شکل ۳- نمودار تغييرات احتمال انتخاب مسير بهينه براي عامل اول با استفاده از الگوريتم۱

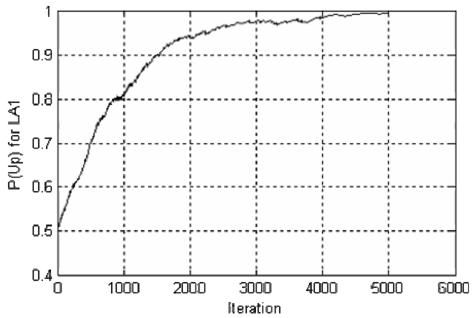
به‌علاوه، منطقي است که احتمال انتخاب عمل حرکت به بالا در خانه شروع هريک از عامل‌ها بايد به‌تدريج افزايش يابد (به اين دليل که عامل‌ها هم‌زمان قادر به حرکت به خانه ۲ نيستند). اين موضوع در نمودارهاي شکل۴ ديده مي‌شود.

مسئله‌اي که در الگوريتم بالا ممکن است مشکل‌ساز شود، احتمال ايجاد دور در مسير طي شده توسط عامل‌ها است که اين احتمال با افزايش نرخ يادگيري و يا پاداش افزايش مي‌يابد. الگوريتم بعدي که آن را الگوريتم۲ مي‌ناميم، اين مسئله را حل مي‌کند.

(۱)



(۲)



شکل ۴- نمودار تغييرات احتمال انتخاب عمل حرکت به بالا در خانه شروع به ازاء نرخ يادگيري (۱) ۰/۱ و (۲) ۰/۰۱ اتوماتاي يادگير۱

الگوریتم ۲:

۱- گراف جهت‌داري متناظر با مسئله ايجاد شده، وضعيت‌ها روي گره‌ها و اعمال روي يال‌ها نگاشت مي‌شوند.

۲- در هر گره از گراف، يک اتوماتاي يادگير با تعداد اعمال متغير و با شماي L_{RP} قرار مي‌گيرد. مجموعه اعمال اين اتوماتا برابر با حرکت‌هاي مجاز به وضعيت‌هاي بعدي در نظر گرفته مي‌شود. اعضاي بردار احتمال هر يك از اتوماتاهاي يادگير مساوي و برابر يك تقسيم بر تعداد حرکت‌هاي مجاز به وضعيت‌هاي بعدي مقدار دهي اوليه مي‌شود.

۳- مراحل زير به‌طور موازي براي هر يک از عامل‌ها اجرا مي‌گردد.

۳-۱- عامل j در وضعيت شروع قرار مي‌گيرد.

۳-۲- حد آستانه پويا T_k^j براي عامل برابر صفر قرار داده مي‌شود.

۳-۳- در مرحله k^jام:

۳-۳-۱- با قرار گرفتن عامل j در وضعيت s، اتوماتاي يادگير A_s متناظر با اين وضعيت فعال شده و عامل براي انتقال به وضعيت بعدي از آن کمک مي‌گيرد. بدين منظور، با توجه به بردار احتمال اتوماتا، يک عمل مانند a_m را که متناظر با يال جهت‌داري (s,m) از گراف و به معناي حرکت به وضعيت بعدي m است انتخاب مي‌کند. انتخاب عمل به اين صورت انجام مي‌شود که A_s اعمالي که منجر به ورود به وضعيت‌هاي موجود در مسير طي‌شده تا کنون (و فعال شدن اتوماتاي مربوطه) مي‌گردند را در مجموعه اعمال خود غيرفعال مي‌کند، سپس از بين اعمال باقي‌مانده با توجه به احتمال آن‌ها يک عمل را انتخاب مي‌نمايد. درصورتي که تمام اعمال غيرفعال شده باشند و عملي براي انتخاب باقي نمانده باشد، $k^j \leftarrow k^j + 1$، اعمال فعال مي‌گردند و به مرحله ۳-۳-۷ مي‌رويم. در غير اين‌صورت، پس از انتخاب عمل، کليه اعمالي که غيرفعال شده‌اند مجدداً فعال مي‌گردند (تا براي ساير عامل‌ها قابل استفاده شوند).

۳-۳-۲- عامل j به وضعيت m منتقل مي‌شود.

۳-۳-۳- $s \leftarrow m$ و به مرحله ۳-۳-۱ بازمي‌گرديم تا زماني که عامل j به وضعيت هدف برسد.

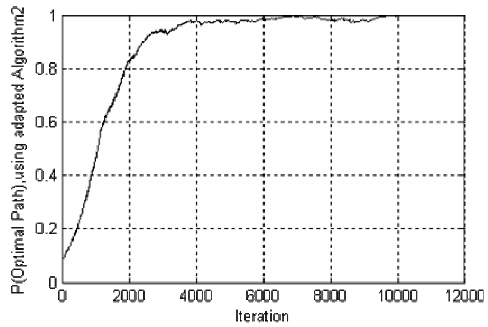
۳-۳-۴- با رسيدن عامل j به وضعيت هدف، هزينه مسير طي‌شده π_i، طبق رابطه $L_{\pi_i}^j = R_G / t_{\pi_i}^j$ محاسبه مي‌شود. در اين رابطه، R_G پاداش مربوط به رسيدن به وضعيت هدف و t_{π_i} مدت زمان سپري شده از شروع حرکت عامل j تا رسيدن آن به وضعيت هدف، با طي مسير π_i مي‌باشد.

۳-۳-۵- اگر شرط $L_{\pi_i}^j < T_k^j$ برقرار باشد، آنگاه به مسير طي‌شده پاداش داده مي‌شود و در غير اين صورت مسير طي‌شده جريمه مي‌شود. ميزان پاداش را برابر با $L_{\pi_i}^j$ و ميزان جريمه را برابر با $-L_{\pi_i}^j$ درنظر مي‌گيريم. اين پاداش يا جريمه به هريک از اتوماتاهايي که در مسير مشارکت داشته‌اند تعلق مي‌گيرد و بردار احتمال آن‌ها بهنگام مي‌شود.

۳-۳-۶- $k^j \leftarrow k^j + 1$ و $T_k^j \leftarrow T_{k-1}^j + \frac{1}{k^j}(L_{\pi_i}^j - T_{k-1}^j)$.

۳-۳-۷- شرط خاتمه الگوريتم (رسيدن احتمال مسير بهينه براي عامل j به حد آستانه) بررسي مي‌شود. در صورت عدم برقراري شرط خاتمه، به مرحله ۲-۳ باز مي‌گرديم.

مطابق شکل۵، با استفاده از اين الگوريتم با نرخ يادگيري ۰/۰۱ و حد آستانه ۰/۹۵ براي احتمال مسير بهينه، به‌طور متوسط پس از ۲۵۰۰ بار تکرار، مسير طي‌شده توسط عامل‌ها به مسير بهينه همگرا مي‌گردد. دليل اين‌که الگوريتم۲ ديرتر از الگوريتم۱ به همگرايي مي‌رسد، کوچک بودن محيط اين زنجيره مارکف و احتمال بالاي قرار گرفتن در خانه‌اي که به‌دليل ايجاد دور، تمام اعمال آن غيرفعال شده‌اند مي‌باشد که در اين صورت عامل ناچار است مسير خود را مجدداً از مبدأ آغاز کند و اتوماتا در آن مرحله بهنگام نمي‌شود.



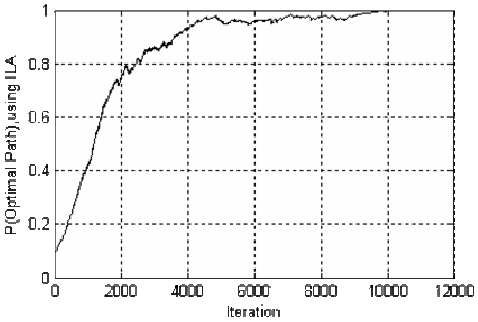
شکل ۵- نمودار تغييرات احتمال انتخاب مسير بهينه براي عامل اول با استفاده از الگوريتم۲

براي مقايسه عملکرد الگوريتم‌هاي ارائه‌شده با ساير رويکردهاي مبتني بر اتوماتاي يادگير، آزمايش را با روشي که در [۶] براي حل مسائل مارکف چندعامله مورد استفاده قرار گرفته است تکرار مي‌نماييم. در اين روش نيز هر خانه از صفحه داراي يک اتوماتاي يادگير است و مجموعه اعمال آن شامل حرکت‌هاي مجاز به خانه‌هاي مجاور مي‌باشد. روش کار به اين صورت است که اتوماتاي يادگير A_i که در وضعيت i فعال مي‌شود، بلافاصله از پاداش مقطعي $r_j^i(k)$ حاصل از عمل k خود که منجر به انتقال به وضعيت j شده آگاه نمي‌گردد؛ بلکه هنگامي‌که وضعيت i مجدداً ملاقات مي‌شود، A_i دو اطلاع را دريافت مي‌نمايد: پاداش افزايشي که از ابتدا تا آن لحظه توسط فرآيند توليد شده است، و زمان سراسري جاري. با استفاده از اين اطلاعات، A_i پاداش دريافتي پس از آخرين ملاقات وضعيت i و مدت زمان سپري‌شده از آن ملاقات تا کنون را محاسبه مي‌کند. سپس ورودي A_i به‌صورت زير خواهد بود:

$$\beta^i(n_i+1) = \frac{\rho_k^i(n_i+1)}{\eta_k^i(n_i+1)} \qquad (۷)$$

که در آن، $\rho_k^i(n_i+1)$ پاداش کلي براي عمل k در وضعيت i و $\eta_k^i(n_i+1)$ مدت زمان سپري‌شده مي‌باشد.

نتيجه اين آزمايش در نمودار شکل۶ مشاهده مي‌شود. همان‌طور که در اين نمودار مي‌توان ديد، با حد آستانه ۰/۹۵ براي احتمال مسير بهينه و نرخ يادگيري ۰/۰۱، عامل‌ها به‌طور متوسط در حدود ۴۰۰۰ تکرار به مسير بهينه همگرا مي‌شوند.



شكل ۶- نمودار تغييرات احتمال انتخاب مسير بهينه براي عامل اول با استفاده از اتوماتاي يادگير اتصالي

با توجه به جدول۱ مي‌توان مقايسه دقيق‌تري بين الگوريتم‌هاي ارائه‌شده انجام داد. لازم به ذكر است كه مسير بهينه‌اي كه با استفاده از روش‌هاي ارائه شده حاصل مي‌شود، در واقع خط‌مشي بهينه براي مسئله ماركف چندعامله مي‌باشد.

الگوريتم مورد استفاده	تعداد تكرار لازم براي همگرايي
الگوريتم پيشنهادي ۱	۲۰۱۶
الگوريتم پيشنهادي ۲	۲۸۶۳
الگوريتم مرجع [۶]	۳۹۹۵

جدول ۱- تعداد متوسط دفعات تكرار براي رسيدن احتمال مسير بهينه به حد آستانه (۰/۹۵) با نرخ يادگيري ۰/۰۱ براي عامل۱

۴- نتيجه‌گيري

در اين مقاله دو الگوريتم براي حل مسائل ماركف چندعامله پيشنهاد گرديد. اين الگوريتم‌ها از شبكه‌اي از اتوماتاهاي يادگير متصل به هم استفاده مي‌كند. در الگوريتم‌هاي ارائه شده، عامل‌ها از اتوماتاهاي يادگير براي كمك به فرآيند تصميم‌گيري خود به‌منظور انتقال بين وضعيت‌هاي مختلف زنجيره ماركف و يافتن بهترين و كم‌هزينه‌ترين خط‌مشي بهره مي‌گيرند. براي ارزيابي الگوريتم‌هاي ارائه شده، به حل نمونه‌اي از مسئله ماركف چندعامله كه درعين‌حال نوعي مسئله بهينه‌سازي و همچنين داراي جنبه هماهنگي و همكاري بين عامل‌ها مي‌باشد پرداختيم. نتايج آزمايش‌ها نشان داد كه اين الگوريتم‌ها از جهت سرعت همگرايي و هزينه مسير نهايي، در مقايسه با ساير روش‌هاي مبتني بر اتوماتاي يادگير كه تا به حال براي حل مسائل ماركف چندعامله و يافتن خط‌مشي بهينه در اين مسائل به‌كار رفته‌اند برتر مي‌باشند.

مراجع

1. *Weiss, G., "Multi-Agent Systems, a Modern Approach to Distributed Artificial Intelligence", MIT Press, 2000.*
2. *Nowe, A., Verbeeck, K., and Peeters, M., "Learning Automata as a Basis for Multi-Agent Reinforcement Learning", First International Workshop on Learning and Adaptation in Multi-Agent Systems (LAMAS), pp. 71-85, 2006.*
3. *Hu, J. and Wellman, M. P., "Multi-Agent Reinforcement Learning: Theoretical Framework and an Algorithm", Proceedings of the Fifteenth International Conference on Machine Learning, 1998.*
4. *Laroche, P. Boniface, Y. and Schott, R., "A New Decomposition Technique for Solving Markov decision Processes", Proceedings of the 2001 ACM Symposium on Applied Computing, Las Vegas, Nevada, United States, pp. 12 – 16, 2001.*
5. *Peret, L. and Garcia, F., "On-line search for solving large Markov Decision Processes", Sixth European Workshop on Reinforcement Learning (EWRL-6), 2003.*
6. *Nowe, A. and Verbeeck, K., "Colonies of Learning Automata", IEEE Transactions on Systems, Man and Cybernetics, vol. 32, pp. 772-780, 2002.*

7. *Beigy, H. and 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.*
8. *Busoniu, L., De Schutter, B. and Babuska, R., "Learning and Coordination in Dynamic Multi-Agent Systems", Delft Center for Systems and Control, 2005.*
9. *Hanna, H., Mouaddid, A., "Markov Decision Process for Allocation Task in Multi-Agent Systems", International Conference IPMU, France, 2002.*
10. *Spaan, M., Vlassis, N. and Groen, F., "High Level Coordination of Agents based on Multi-Agent Markov Decision Processes with Roles", IROS'02 Workshop on Cooperative Robotics, 2002.*
11. *Shirazi, M. R. and Meybodi, M. R., "Application of Learning Automata to Cooperation in Multi-Agent Systems", Proceedings of First International Conference on Information and Knowledge Technology (IKT2003), pp. 338-349, 2003.*
12. *Hetland, H. and Eriksen, T. L., "Interconnected Learning Automata Playing Iterated Prisoner's Dilemma", Master Thesis, Agder University College, 2004.*
13. *Verbeek, K., Nowe, A., Peeters, M. and Tuyls, K., "Multi-Agent Reinforcement Learning in stochastic Single and Multi-Stage Games", Lecture Notes in Computer Science, Springer Berlin, vol. 3394, pp. 275-294, 2005.*