اتوماتاهای یادگیر، راه حلی برای بازی های غیر قطعی بامجموع کلی

بهروز معصومی[۱]، محمدرضا میبدی[۲]، برنا جعفرپور[۲]

[۱] دانشگاه آزاد اسلامی واحد علوم و تحقیقات، تهران و مرکز تحقیقات MRL دانشگاه آزاد اسلامی قزوین

[۲] دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر،تهران ،ایران

Email: bmasoumi@Qazviniau.ac.ir, mmeybodi@aut.ac.ir, jafarpour@cic.aut.ac.ir

چکیده- بازی های غیر قطعی(اتفاقی) برای مدل سازی سیستمهای چند عامله بسیار مورد استفاده قرار گرفته اند.این بازیها توسعه ای از فرآیندهای تصادفی مارکوف با چندین عامل و بازی های ماتریسی با چندین حالت هستند.هدف هرعامل پیدا کردن سیاست بهینه ای است بطوریکه امید ریاضی مجموع کاهش‌یافته پاداشها را بیشینه نماید. در این مقاله یک مدل یادگیری تقویتی مبتنی بر اتوماتاهای یادگیر برای حل بازی های اتفاقی با مجموع کلی برای یافتن خط‌مشی بهینه پیشنهاد شده است.بازای هر حالت در محیط یک اتوماتا قرار داده شده بطوریکه تعداد اعمال هر اتوماتا با توجه به همسایگانش تعیین می‌گردد.هر اتوماتا مسوول انتخاب حالت بعدی محیط است. آزمایشهای انجام گرفته نشان داده‌اند که الگوریتم ارائه شده از کارایی مناسبی از هر دو جنبه هزینه و سرعت رسیدن به راه‌حل بهینه برخوردار است.

کلید واژه- سیستمهای چند عامله، اتوماتاهای یادگیر، یادگیری تقویتی چند عامله، بازی های اتفاقی

۱- مقدمه

امروزه در بسیاری از کاربردها و در زمینه‌های مختلف صنعتی، نظامی، مخابراتی، اطلاعاتی ، از سیستم‌های پیچیده و توزیع‌شده چندعامله استفاده فراوانی می-شود[۲،۱]. یک سیستم چندعامله، در برگیرنده جامعه‌ای از عامل‌های هوشمند و خود مختاراست که در یک محیط درکنار یکدیگر درحال کار بوده و سعی در انجام کاری خاص و رسیدن به هدفی مشخص دارند. برای حل بسیاری از مسائل مهم دنیای واقعی مانند برخی از کاربردهای رباتیک، مسیریابی در شبکه، زمان‌بندی و تصمیم‌گیری اقتصادی نیازمند برنامه‌ریزی در حالت غیرقطعی هستیم. مدلهای فرآیند تصمیم‌گیری مارکف[۱]، چارچوب مناسبی برای مدل‌سازی این مسائل و یافتن راه‌حل‌های بهینه برای آن‌ها می‌باشد. برای مدل سازی سیستمهای چند عامله مدلهای مدلهای مختلفی با توجه به مدل مارکوف پیشنهاد شده است که ازجمله آنها مدل بازی های مارکوفی[۲]، فرآیندهای تصادفی مارکوف با مشاهدات جزئی[۳]، فرآیندهای تصادفی مارکوف با مشاهدات جزئی نامتمرکز[۴] را نام برد[۴،۳].

بازی های غیر قطعی (اتفاقی[۵]) به عنوان توسعه ای از فرآیندهای تصادفی مارکوف با چندین عامل در سیستمهای چند عامله بسیار حائز اهمیت بوده و به عنوان چارچوبی مناسب در تحقیقات یادگیری های چند عامله به ویژه یادگیری‌تقویتی چندعامله *(MARL[۶])* به‌کاررفته‌اند [۷،۶،۵]. یادگیری تقویتی چندعامله به سرعت در حال توسعه بوده و روشهای متنوع و مختلفی را درحوزه های رقابتی، همکارانه و ترکیبی بر می گیرد بطوری که ارتباطی بین حوزه های مختلف علوم نظیر تئوری‌بازیها، بهینه سازی، یادگیری در بازی ها برقرار ساخته است[۸].

بازی های‌اتفاقی(*SG*) مدلی توسعه یافته از بازی‌های تکرارشونده هستند که در هر لحظه از زمان، بازی در یک‌حالت قرار دارد. گذار از حالتی به حالت جدید بر پایه تابع احتمالاتی با توجه به حالت قبلی و تعامل بین عامل ها در حالت قبل انجام می گیرد. هر حالت در یک *SG* می تواند بصورت یک فرآیند تصمیم گیری مارکوف دیده شودوهر*SG* با یک عامل بصورت یک فرآیند تصمیم گیری مارکوف می باشد. بازیهای اتفاقی دارای انواع متفاوتی

هستند. بازی های اتفاقی از نظر پاداش به بازی های با مجموع صفر(رقابتی) و مجموع کلی تقسیم بندی شده اند. برای حل بازی های اتفاقی اعم از بازی های رقابتی و غیر رقابتی الگوریتمهای یادگیری تقویتی متعددی به کاررفته اند . با توجه به پیچیدگی محاسباتی روش ها هر کدام در کاربرد خاص با شرایط خاصی استفاده شده اند. هدف اصلی این مقاله ارائه الگوریتمی با استفاده اتوماتاهای یادگیر برای حل مسائل بازی های اتفاقی مجموع کلی و یافتن خط‌مشی بهینه است. در بخش ۲ به مبحث یادگیری تقویتی و بازی های اتفاقی پرداخته شده در بخش ۳ اتوماتاهای یادگیر و در بخش ۴و۵ الگوریتم پیشنهادی و نتایج آزمایشها ارائه گردیده اند.

۲- یادگیری تقویتی و حل بازی های اتفاقی

در سیستمهای چند عامله ، الگوریتم‌های یادگیری تقویتی مانند یادگیری Q، با موفقیت در بسیاری از کاربردها مورد استفاده قرارگرفته‌اند[۸]. اکثر الگوریتم‌های یادگیری تقویتی چندعامله، بر پایه روش‌های تک‌عامله بنا نهاده شده‌اند. درحالت یادگیری تک عامله که در آن عامل به‌طور مستقل درحال یادگیری باشد، فرآیند یادگیری تقویتی را می‌توان به‌صورت فرآیند تصمیم‌گیری مارکوف تعریف نمود.

تعریف ۱. فرآیند تصادفی مارکوف بصورت چندتایی $<S,A,P,r,\gamma>$نشان داده می شود که در آن Sمجموعه متناهی از وضعیت‌ها ؛ A مجموعه عملیات قابل دسترس برای عامل ،γضریب‌کاهش‌و $P : S \times S \times A \mapsto [0,1]$ احتمال انتقال از وضعیت جاری به وضعیت بعدی با انجام عمل a است و $r : S \times A \mapsto \Re$ تابع پاداش است که یک مقدار عددی را بر می گرداند.

در یکMDP هدف عامل پیدا کردن استراتژی $\pi : S \to A$ است که امید ریاضی مجموع کاهش‌یافته پاداشها را بیشینه نماید. یادگیری تقویتی برپایه اصل بهینگی *Bellman* استوار است. برای هر خط‌مشی π که عامل می‌تواند دنبال کند، بر روی وضعیت‌ها تابعی به نام تابع ارزیابی[۷] به‌شکل زیر تعریف می‌شود:

$$V(\pi,s) \equiv \sum_{t=0}^{\infty} \gamma^t E\left(r_t | \pi, s_0 = s\right) \qquad (۱)$$

این تابع، نگاشتی از مجموعه وضعیت‌ها به مقدار ارزش آن‌ها می‌باشد. r_t پاداش در زمان t وγضریب کاهش در محدوده $[0,1)$ است. . بنابراین در اینجا هدف، یادگیری خط‌مشی بهینه است که به‌صورت زیر تعریف می‌شود:

$$\pi^*(s) = \arg\max_{\pi} [V(\pi(s,a))] \qquad (۲)$$

عامل باید سعی کند تا مقادیرV را به‌ازای خط‌مشی بهینه یاد بگیرد. این مقادیر را به‌طور خلاصه به‌شکل V^* نشان می‌دهیم.

$$\pi^*(s) = \arg\max_{a} [r(s,a) + \gamma V^*(\delta(s,a))] \qquad (۳)$$

برای محاسبه تابع $V(s)$از تکنیک برنامه‌سازی پویا[۸] استفاده می‌شود. در این صورت، روشی تقریبی برای تخمین مقادیر بهینه $V(s)$ به‌کار می‌رود که روش *Value Iteration* نام دارد. این روش هنگامی قابل استفاده است که عامل، توابع $\delta : S \times A \to S$ و $r : S \times A \to R$ را بشناسد، در غیراین‌صورت نمی‌توان این روش را به‌کارگرفت. در چنین حالتی، از الگوریتمی به‌نام یادگیریQ استفاده می‌نماییم.در یادگیری Q استاندارد یک عامل مقادیر Q را طبق معادله زیر یاد می گیرد که در آن α نرخ یادگیری و β ضریب کاهش است.

$$Q_{k+1}(s_k, a) = (1 - \alpha_t) Q_k(s_k, a) + \alpha_t \left[r_t + \beta \max_b Q_k(s_{k+1}, b) \right] \qquad (۴)$$

بازی های اتفاقی تعمیم فرآیند تصادفی مارکوف به‌حالت چندعامله‌وهمچنین‌توسعه‌ای‌از بازی‌های‌ماتریسی‌با چندین حالت بوده و به آنها نام بازیهای مارکوفی نیز گویند.

تعریف۲. یک بازی اتفاقی (مارکوف) بصورت‌چندتایی $< n, S, \vec{A}, \vec{R}, T >$بیان میشود که در آن S مجموعه حالات، n تعداد عامل ها ، $\vec{A}$ مجموعه اعمال هر عامل i، $\vec{R}$ تابع پاداش عامل های مختلف $S \times \vec{A} \to R$ و T تابع تبدیل اتفاقی است.در حالت چند عامله تابع تبدیل با توجه به عمل گروه عاملها تعیین می شود.اگر پاداش تمام

عاملها برابر باشد بازی را کاملا همکارانه[۹] و اگر $n=2$و پاداش یک بازیکن مخالف پاداش دیگری باشد بازی را رقابتی و در حالت کلی بازی را ترکیبی گویند.

در حالت یادگیری تقویتی چند عامله ، بیشینه نمودن سودمندی(پاداش) مورد انتظار هر عامل به تنهایی کافی نیست بطوریکه هدف پیدا کردن سیاست متعادل در بازی های مارکوف است.از جمله این سیاست ها می توان سیاست تعادل نش[۱۰] را نام برد. به عبارت دیگر پیدا کردن یک سیاست متعادل به عنوان یک راه حل برای بازی های اتفاقی محسوب می شود. استراتژی یک عامل، برنامه و طرح آن در بازی محسوب می گردد. استراتژی بصورت $\pi=(\pi_0,\cdots\pi_t,\ldots)$ روی کل بازی تعریف می شود که π_t قانون تصمیم گیری در لحظه t نامیده می شود. قاتون πرا ایستا گویند اگر مستقل از زمان باشد.در این مقاله ما روی استراتژی‌های ایستا تاکید داریم. استراتژی بصورت معادله ۵ تعریف می شود که توزیع احتمالاتی را بر روی اعمال قابل دسترس عاملها برای هر حالت $s_j, j=1..m$و m تعداد حالات است تعیین می کند.

$$\overline{\pi}=\left(\overline{\pi}(s1),\ldots,\overline{\pi}(sm)\right) \qquad (۵)$$

برای هر حالت s_j طبق معادله ۶ داریم، بطوریکه$P(a_k)$ احتمال انجام عمل a_k را نشان می دهد .

$$\overline{\pi}(s_j)=\{P(a_1),\ldots,P(a_m)\} \qquad (۶)$$

$\sum_{k=1}^{M}P(a_k)=1$ و M تعداد اعمال هر عامل‌است

تعریف۳. در یک بازی اتفاقی یک تعادل نش بصورت زوج استراتژی $\left(\pi_*^1,\pi_*^2\right)$ تعریف می شود بطوریکه برای هر $s\in S$ داریم :

$$v^1\left(S,\pi_*^1,\pi_*^2\right)\geq v^1\left(S,\pi^1,\pi_*^2\right)\forall\pi^1\in\Pi^1 \qquad (۷)$$
$$v^2\left(S,\pi_*^1,\pi_*^2\right)\geq v^2\left(S,\pi_*^1,\pi^2\right)\forall\pi^2\in\Pi^2$$

یادگیری Q چند عامله مشابه یادگیری تک عامله است با این تفاوت که در اینجا اعمال و استراتژی های گروهی[۱۱] مد نظر است و پیدا کردن مقدار Q بهینه بستگی به استراتژی و اعمال دیگران دارد. بطورکلی معادله بروز رسانی Q را می توان بصورت زیر نوشت :

$$Q_i(s,\vec{a})=(1-\alpha)\,Q_i(s,\vec{a})+\alpha\left(r+\gamma V(s')\right) \qquad (۸)$$
$$V(s)=Value\left(\left[Q_i(s,\vec{a})\right]\right)$$

در این معادله r_t^iپاداش عامل i در لحظه t وγ ضریب کاهش است.الگوریتم یادگیری Q چند عامله در شکل (۱) دیده می شود: انتخاب توابع مختلف متناسب با نوع بازی باعث ایجاد الگوریتمهای مختلف شده است. ازجمله این توابع می توان به *minimaxQ* برای بازی های رقابتی[۵] ، *Nash(Q)* [۷]و*NashBargaining* [۹] و *ParetoQ* [۱۰] برای بازیهای با مجموع کلی استفاده شده است .

Multi-Q-learning (MarkovGame,α,γ)
Inputs: discount factor γ , learning rate α
func Q_i^* Output : state–value func V_i^*,action–value
Initialize : s, $a_1..a_n$, $Q_1..Q_n$
1. for i=1 to M
2. simulate actions a1..an in state s
3. observe rewards R1,,Rn and Next State s'
4. for i = 1 to N
(a) compute $V_i(S')$
5. (b)
$Q_i(s,\vec{a})=(1-\alpha)Q_i(s,\vec{a})+\alpha(1-\gamma)(R_i+\gamma V_i(s')$
agents choose action $a'_1..a'_n$
6. s= s' ,a_1= a'_1 , a_n= a'_n
7. decay α

شکل ۱: الگوریتم یادگیری Q چند عامله

۳- اتوماتاهای یادگیر

با توجه به اینکه مدل های اتوماتاهای یادگیر ارتباطی نزدیک با بحث یادگیری تقویتی چند عامله دارند ،در این بخش اتوماتاهای یادگیر به عنوان مدلی از یادگیری تقویتی به اختصار شرح داده میشود.

۳-۱- اتوماتاهای یادگیر

اتوماتای یادگیر، ماشینی است که می‌تواند تعدادی متناهی عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی می‌شود و نتیجه ارزیابی در قالب سیگنالی مثبت یا منفی به اتوماتا داده می‌شود و اتوماتا از این پاسخ در انتخاب عمل بعدی تأثیر می‌گیرد.

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 شکل۲ مشاهده می‌شود.

محیط را می‌توان توسط سه‌تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه ورودی‌ها، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ مجموعه خروجی‌ها و $c \equiv \{c_1, c_2, ..., c_r\}$ مجموعه احتمال‌های جریمه می‌باشد. هرگاه β مجموعه‌ای 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 می‌شود. در محیط از نوع Q، $\beta(n)$ می‌تواند به طور گسسته یک مقدار از مقادیر محدود در فاصله [۰،۱] را اختیار کند و در محیط از نوع S، $\beta(n)$ متغیر تصادفی در فاصله [۰،۱] است. c_i احتمال اینکه عمل α_i نتیجه نامطلوب داشته باشد می‌باشد. در محیط ایستا، مقادیر c_i بدون تغییر می‌مانند، حال آن‌که در محیط غیرایستا این مقادیر در طی زمان تغییر می‌کنند. اتوماتاهای یادگیر به دو دسته اتوماتای یادگیر با ساختار ثابت اتوماتای یادگیر با ساختار متغیر($VSLA$[۱۲])دسته بندی می شوند .

اتوماتای یادگیر با ساختارمتغیر را می‌توان توسط چهارتایی $\{\alpha, \beta, p, T\}$ نشان‌دادکه $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه عمل‌های اتوماتا، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_r\}$ ورودی‌های اتوماتا، $p = \{p_1, .., p_r\}$ بردار احتمال انتخاب هریک از عمل‌ها و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می‌باشد.

الگوریتم‌زیربراساس‌روابط(۹)و(۱۰)یک‌نمونه‌از الگوریتم‌های یادگیری خطی است. فرض می‌کنیم عمل α_i در مرحله nام انتخاب شود.

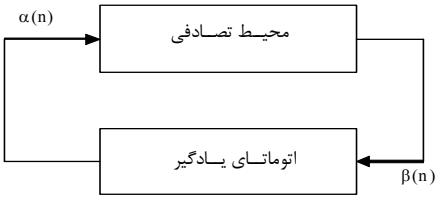
- پاسخ مطلوب از محیط

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)]$$
$$p_j(n+1) = (1-a)p_j(n) \qquad \forall j \;\; j \neq i \qquad (۹)$$

- پاسخ نامطلوب از محیط

$$p_i(n+1) = (1-b)p_i(n)$$
$$p_j(n+1) = (b/r-1) + (1-b)p_j(n)$$
$$\forall j \;\; j \neq i \qquad (۱۰)$$

در روابط (۹) و (۱۰)،a پارامتر پاداش وb پارامتر جریمه می‌باشند. با توجه به مقادیرa وb سه حالت را می‌توان درنظر گرفت: اگرa وb با هم برابر باشند، الگوریتم را L_{RP}، هنگامی‌کهb ازa خیلی کوچکتر باشد، الگوریتم را $L_{R\varepsilon P}$ و اگر b مساوی صفر باشد آن را L_{RI} می‌نامیم[11]. شمای $SLR\text{-}P$ برای مدلهای Q و S براساس‌رابطه(۱۱)بیان‌می شود:



شکل۲- ارتباط بین اتوماتای یادگیر و محیط

اگر عمل α_i در مرحله nام انتخاب شود در این صورت طبق معادله (۱۱) داریم :

r تعداد اعمال ممکن ، aپارامتر پاداش وb پارامتر جریمه

$$p_i(n+1) = p_i(n) + a(1-\beta_i(n))(1-p_i(n)) - a\beta_i(n)p_i(n)$$
$$p_j(n+1) = p_j(n) - a(1-\beta_i(n))p_j(n) + a\beta_i(n)\left[\frac{1}{r-1} - p_j(n)\right] - a(1-\beta_i(n))p_j(n) \qquad (۱۱)$$
$$j \neq i$$

می‌باشند.برای اطلاعات بیشتر در باره اتوماتاهای یادگیر می توان به[12] مراجعه نمود.

۴- استفاده از اتوماتهای یادگیربرای حل بازی های اتفاقی

در این بخش روش پیشنهادی برای یادگیری در سیستمهای چند عاملی به کمک اتوماتای یادگیر را بررسی می‌کنیم.ما این روش راMLA[۱۳]می نامیم .در یک بازی مارکوفی عمل انتخاب شده در هر حالت با توجه به نتیجه اعمال گروهی عاملهای مستقل در سیستم است. در این مدل در هر حالت s_i ($i=1..m$ وm تعداد حالات)از محیط بازی هر عامل k یک اتوماتای یادگیر نظیر LA_k^i

با ساختار متغیر و مدل S قرار داده می‌شود. با توجه به تعداد همسایگان هر اتوماتا ، تعداد اعمال اتوماتا در هر حالت در نظر گرفته شده و هر عمل مشترک متناظر با انتقال به یکی از حالات همسایه می‌باشد. هر عامل برای تعیین حالت بعدی خود از اتوماتای یادگیر با توجه به عمل انتخابی عاملهای محیط کمک می‌گیرد.

در ابتدا اتوماتاهای یادگیر تمام عملهای خود را با احتمالی یکسان انتخاب می‌کنند. درصورتیکه عمل اتوماتا هامنجر به ورود عامل به حالت هدف شود اتوماتا پاداش می‌گیرد، در غیر اینصورت از آنتروپی بردار احتمال اتوماتای یادگیر حالت بعد برای تعیین پاداش یا جریمه استفاده می‌شود. آنتروپی بردار احتمال میزان عدم قطعیت اتوماتای یادگیر حالت بعد را در انتخاب عمل خود نشان می‌دهد. هر چه آنتروپی بیشتر باشد میزان عدم قطعیت بیشتر است. عدم قطعیت بالا در بردار احتمال اتوماتای یادگیر به این معنی است که این اتوماتا دارای اطلاعات مفیدی برای رسیدن به هدف نیست و عملهای خود را به صورت تصادفی انتخاب می‌کند(جستجو[۱۴]). ولی چنانچه عدم قطعیت کم باشد به این معنی است که اتوماتا با احتمال بالایی یکی از اعمال خود را اتخاب می‌کند و دارای اطلاعات مفیدی برای رسیدن به هدف می‌باشدو از این اطلاعات بهره برداری می نماید[۱۵] . فرض کنید که $\{p_1,p_2,...p_r\}$ بردار احتمال اعمال یک اتوماتای یادگیر باشد . آنتروپی این بردار احتمال به شکل زیر تعیین می‌شود.

$$E(P) = -\sum_{i=1}^{r} p_i \log(p_i) \qquad (۱۲)$$

زمانی‌آنتروپی‌بیشترین مقدار را خواهد داشت‌که‌تمام اعمال احتمالی‌یکسان‌داشته‌باشند$P_{equal}=\{p_1=p_2=...p_r=1/r\}$ و زمانی‌کمترین مقدار (برابربا۰) را خواهد داشت که $\exists i,\ p_i = 1 \wedge \forall\ j \neq i\ p_j = 0$. برای اینکه مقدار آنتروپی را به مقداری بین ۰ و ۱ تبدیل کنیم تا به عنوان بردار تقویتی در اتوماتای ساختار متغیر مدل S قابل استفاده باشد از فرمول ۱۳ استفاده می‌شود[۱۳].

$$\beta = E((P) / E(P_{equal}))^K \qquad (۱۳)$$

K پارامتر روش می‌باشد. میزان این پارامتر باید به دقت تعیین شود. مقادیر بالای این پارامتر باعث می‌شود که β ها مقادیر کمی داشته باشند و اتوماتاهای یادگیر بیش از حد پاداش ببینند. این به معنی جستجوی بیشتر در محیط است. هر چه میزان K کمتر باشد βها مقادیر بیشتری خواهند داشت. این امر باعث می‌شود که اتوماتاها بیشتر جریمه شوند. این مسئله باعث می‌شود که حتی حالتهای مطلوب نیز پاداش لازم را نگیرند. در الگوریتم پیشنهادی به گونه ای عمل می شود مشابه با روش بولتزمن در یادگیری Qدر اینجا ابتدا عاملها جستجو انجام داده و به مرور زمان با توجه به تغییر پارامتر بهره برداری از دانسته های خود داشته باشند. فرض کنید که عاملk در حالت s باشد و اتوماتای یادگیر آن LA_k^s)عامل را به حالت s' هدایت کند. در اینصورت تعیین سیگنال تقویتی طبق معادله ۱۴ تعیین می‌شود بطوریکه $P\left(LA_i^s\right)$ نشان دهنده بردار احتمال اتوماتای یادگیر i قرار گرفته در حالت s می‌باشد. الگوریتم MLA در شکل (۳) آورده شده است.

MAL (StochasticGame,a,b,k,M)
Inputs: a ,b: reward and penalty parameter for LA ,k:exploration Parameter , M : total training time
Initialize : *$s_0, a_1 ,...,a_n$, initialization probablity of all of LA*
1. *For Episod =1 to M do*
2. *While not done*
3. *k= initial Value*
4. *for each agent k do cuncurrently*
5. *Active LA_k^i*
6. *Choose action a_k^i in state s_i*
7. *Observe Rewards r_k^i and Next State s'*
8. *Compute β_k^i signal basedon EQ (14)*
9. *Train LA_k^i residing in state S_i according β_k^i*
10. *s=s'*
11. *End while*
12. *Increment k*
13. *End for*

شکل ۳- الگوریتم MLA

$$\beta_i^s = \begin{cases} 1 & s' = Out\ of\ bound \\ & s' = Goal \\ (E(P(LA_i^{s'})) / E(P_{equal}))^K & otherwise \end{cases} \qquad (۱۴)$$

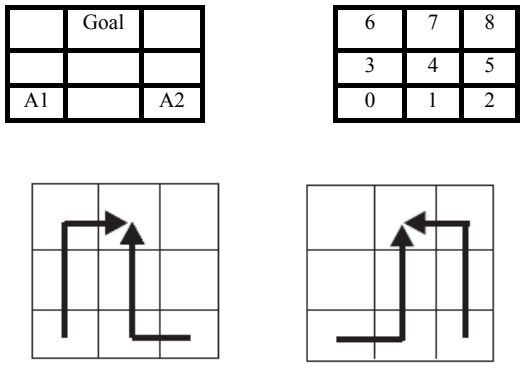
۴-۱- حل بازی های اتفاقی با اتوماتای یادگیر

یکی از انواع بازی‌های‌اتفاقی‌غیر رقابتی بازی های *Grid Game* است که توسط *Hu,Wellman* ارائه شده است [۶]. این بازی یک بازی دو نفری از نوع جمع کلی است. در این بازی دوعامل از دو گوشه یک صفحه شروع کرده و سعی می کنند تا به مربع هدف برسند . این بازی در چند شکل مختلف مطرح شده است .در یک نوع آن فقط یک هدف وجود دارد و و دارای ۹ خانه است .حرکت بازیکنان قطعی بوده وهدف این است که باکمترین تعداد حرکت به هدف برسیم.شکل(۴) مدل بازی رابه همراه مختصات مورد نظر بازی و همچنین راه حل های بهینه را نشان می‌دهد.

یک مسیر دنباله ای از اعمال از نقطه شروع تا پایان را نشان می دهد. در اصطلاح بازی چنین مسیری را استراتژی یا سیاست می نامند. کوتاهترین مسیری که با مسیر عامل دیگر تداخل نداشته باشد را استراتژی بهینه می نامند بطوریکه یک‌موازنه نش را می سازند زیرا هر مسیر(استراتژی) بهترین پاسخ در قبال دیگری است.در این بازی فرض می شود عاملها از موقعیت هدف در ابتدای بازی آگاهی نداشته و همچنین از پاداش کلی یکدیگر اطلاع ندارند.عاملها اعمالشان را همزمان انتخاب نموده و فقط می توانندازاعمال قبلی عاملهای دیگر و حالت فعلی (موقعیت مشترک هر دو عامل) آگاهی داشته باشند.

اعمال بازیکنان یعنی *A2,A1* بصورت چهار عمل(شمال،جنوب،شرق و غرب یا بالا،پایین،چپ وراست)تعریف می شود . مجموعه فضای حالات بصورت $S=\{s/s=(l_1,l_2)\}$ تعریف می شود که هر حالت $s=(l_1,l_2)$ مختصات عاملهای ۱و۲ را نشان می دهد.عاملها نمی توانند در یک مختصات یکسان قرار گیرند. اگر دو عامل سعی در حرکت به یک مربع یکسان داشته باشند حرکت هردو باشکست مواجه می شود. تعداد حالات ممکن در بازی برابر۵۶=۷×۸ است. اگر عاملها به دو مربع مختلف غیر هدف بروند هر دو پاداش صفر را دریافت می کنند و اگر یکی به هدف برسد ۱۰۰واحد پاداش می گیردو در صورت برخورد هردو یک واحد جریمه می شوندو در موقعیت قبلی می مانند. در هر دو بازی گذار از حالتی به حالت دیگر با قطعیت انجام می شود .یعنی حالت جاری وعمل مشترک عاملها منحصرا حالت بعدی را تعیین می کنند .

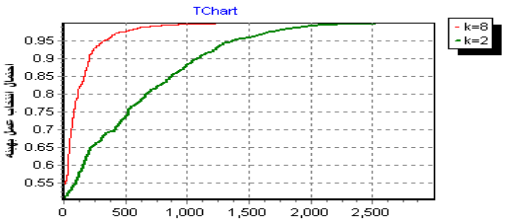
برای حل مساله با توجه به روش ارائه شده بازای هر حالت برای هر عامل یک اتوماتای یادگیر در نظر گرفته شده است . هرعامل با توجه به عمل مشترک گروهی از یک حالت با حالت جدید میرود (درصورت عدم برخورد دو عامل با یکدیگرو با توجه به عدم برخورد با موانع(دیوارها) وبراساس‌معادله ۱۴ مقدار β (پارامتر پاداش یا جریمه)تعیین شده و اتوماتای آن حالت بروز می‌گردد.عاملها اعمالشان را همزمان انجام داده و هردو قابلیت مشاهده حالت جدید ، پاداش آنی بدست آمده و عمل انجام شده توسط دیگری را دارند.



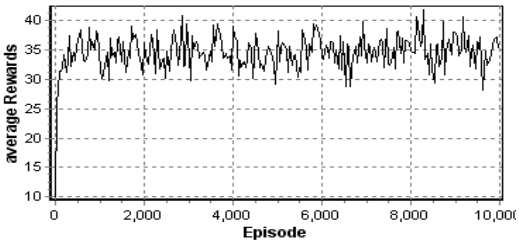
شکل ۴. بازی Grid World و نمایش مختصات بازی به همراه راه حلهای بهینه

۵- آزمایشهای انجام گرفته و نتایج آنها

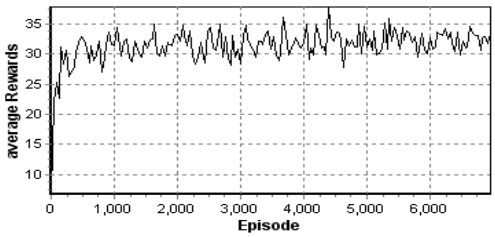
هدف اصلی آزمایشها بررسی این است که آیا اتوماتای یادگیر می تواند برای راه حل بازی های مارکوفی مناسب باشد؟ آیا به راه حل تعادل نش بهینه می شود؟ در آزمایش اولیه بررسی همگرایی اتوماتا انجام گرفته است. تغییرات احتمال انتخاب عمل حرکت به بالا در خانه شروع با توجه به پارامترهای در نظر گرفته شده یعنی پارامتر a برابر۰/۰۲ و پارامتر b برابر ۰/۰۰۲ در عامل ۱ بررسی شد. شکل ۵ نمودار تغییرات را برای این عامل نشان می دهد. با توجه به پارامترهای $k=\{2,8\}$ می بینیم همگرایی در حالت $k=2$ کمتر است. زیرا مقدار پارامتر بردار تقویتی بیشتر بوده و پاداش کمتری به عاملها داده می شود. به عبارتی هر چه پارامتر جستجو بیشتر شود سرعت همگرایی بیشتر می شود.



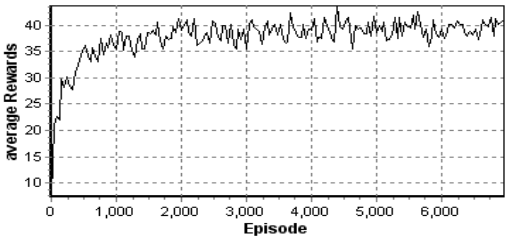
شکل ۵- نمودار تغییرات احتمال انتخاب عمل در خانه شروع به ازاء نرخ یادگیری ۰/۰۲ در ۳۰۰۰تکراربرای اتوماتای یادگیر۱



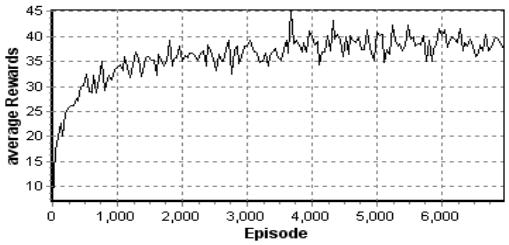
الف) الگوریتم NASHQ



ب) الگوریتم ارائه شده با K=1



K=3

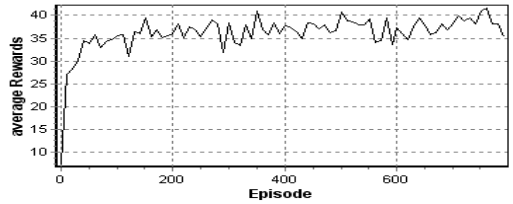


K=6

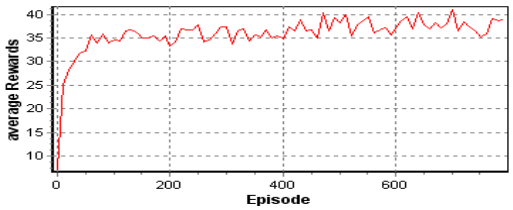
شکل ۶. میانگین پاداش بدست آمده در هر اپیزود در ۵۰۰ آزمایش بازای مقادیر مختلف K و مقایسه با NASHQ (a=0.02, b=0.002)

در آزمایشهای سری دوم نیز پارامتر a برابر۰/۰۲ و پارامتر b برابر ۰/۰۰۲ برای هر یک از اتوماتاها در نظر گرفته شده است. پارامتر $K=\{1,3,6\}$ در نظر گرفته شده است. برای نمایش نتایج ازمیانگین پاداش به دست آمده در هر اپیزود استفاده شده است. ابتدا مقایسه ای بین نتایج تولیدی با نتایج *Nash-Q* آورده شده است بطوریکه دیده می شود میانگین پاداش بدست آمده در هر دو الکوریتم تقریبا برابر هستند. با تغییرات پارامتر K نتایج در لحظه شروع متفاوت و همچنین شیب رسیدن به نقاط تعادل متفاوت میباشد.آزمایشها به خاطر استحکام نتایج ۵۰۰ بار تکرار شده ودر هر بار ۷۰۰۰ اپیزود آزمایش شده است.در هر اپیزود جدید هر عامل بطورتصادفی یک مختصات جدید (بجز هدف) را خواهد گرفت. شکل ۶ میانگین پاداش بدست آمده را در هر اپیزود نشان می دهد. همانطور که در شکل دیده میشود در اینجا نیز نقش k در میزان گرفتن پاداش عاملها مشهود است.

در سری سوم آزمایشها نقش پارامترهای پاداش و جریمه یعنی a,b مورد بررسی قرار میگیرد. با در نظر گرفتن مقدار $a=0.2,b=0.002$ می بینیم شیب رسیدن به نقطه تعادل بالا رفته و در فاصله زمانی کمتری به پاداش مطلوبتر می رسدبا توجه به تغییر پارامتر $b=0$ می بینیم اتوماتاهای *LRI* رفتار بهتری را ازخود نشان می دهند. شکل ۷ نتایج بدست آمده را با پارامتر های مختلف نشان می دهد.



a=0.2,b=0.005, k=2

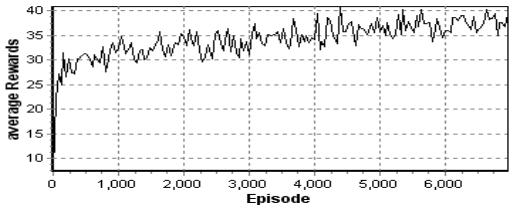


a=0.2,b=0.0, k=2

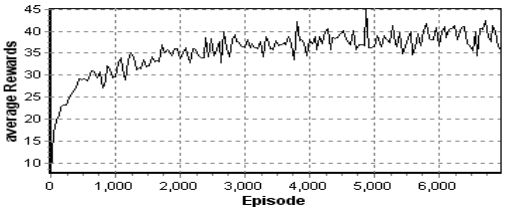
شکل ۷. بررسی رفتار پارامترهای a, b

در آزمایشهای نهایی نقش پارامتر k در بهبود نتایج

بررسی میشود. با توجه به ایده روش بولتزمن در یادگیری Q ابتدا با پارامتر k=۱ نتایج بررسی شده (قابلیت جستجوی بیشتر و به ابتدا عاملها جستجو انجام داده و به مرور زمان با توجه به تغییر پارامتر (افزایش مقدار k(امکان بهره برداری از دانسته های عامل میسر شده است.شکل ۸ این نتایج را نشان می دهد.



مقدار اولیه k برابر ۱ و افزایش 0.0004 در هر مرحله



مقدار اولیه k برابر ۶ و کاهش 0.0004 در هر مرحله

شکل ۸. بررسی رفتارالگوریتم با تغییر پارامتر k در طول اجرای الگوریتم.

۶-نتیجه گیری

در این مقاله الگوریتمی مبتنی بر اتوماتاهای یادگیر برای حل بازی های مارکوفی ارائه گردید با توجه به الگوریتم ارائه شده ونتایج بدست آمده می بینیم الگوریتم برای حل بازی های مارکوفی با مجموع کلی مناسب است. تعداد تکرار ها ، پارامترهای یادگیری و جرائم و نیز مقدار پارامتر k سرعت رسیدن به تعادل را تعیین می نمایند. تنظیم پارامترهای پاداش و جریمه اتوماتاها می تواند کارایی رسیدن به راه حل بهینه را افزایش داده بطوریکه استفاده ازتغییرات پارامتر k در طول اجرای بازی می تواند زمان رسیدن به راه حل بهینه را بهبود بخشد. با توجه به نتایج به دست آمده اتوماتاهای یادگیر مدل مناسب یادگیری و هماهنگی بین عاملها در سیستمهای چندعامله بوده بطوریکه می تواند به عنوان راه حلی مناسب و کارا در بازی های مارکوفی به کار روند.

مراجع

[1] H. Van Dyke Parunak, "A Practitioners' Review of Industrial Agent Applications, Autonomous Agents and Multi-Agent Systems", v.3 n.4, p.389-407, 2000.

[2] P. Stone,M. Veloso, “Multiagent systems: A survey from the machine learning perspective,” Auton. Robots, vol. 8, no. 3, pp. 345–383, 2000.

[3] D. Bernstein,S. Zilberstein, , N .Immerman, "The complexity of decentralized control of Markov decision processes, Mathematics of Operations Research, Vol. 27, No. 4 pp. 819-840, 2002.

[4] C. Claus and C. Boutilier," Sequential optimality and coordination in multiagent systems". In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, 1999.

[5] M. L. Littman, “Markov games as a framework for multi-agent reinforcement learning,” ICML-94,pp157–163,1994.

[6] J. Hu and M. P. Wellman, Online Learning about Other Agents in a Dynamic Multiagent System, Journal of Cognitive System Research 2(1), pp. 67-79, Elsevier Science,2001.

[7] J. Hu and M. P. Wellman," Nash Q-Learning for General-Sum Stochastic Games" , Journal of Machine Learning Research, 4(Nov):pp. 1039-1069, 2003.

[8] L. Busniu, R. Babuska, B. Schutter "A Coprehensive Survey of Multiagent Reinforcement Learning ".IEEE Transaction on System, Man, Cybern., vol. 38, no.2, pp.156–171,2008.

[9] H. Qio , F. Szidarovszky , Rozenblit and L. Yong," Multi-agent learning model with bargaining ",Proceedings of the 38th conference on Winter simulation .pp.934 - 940 ,2006.

[10] M. Song, J. Bai, R. Chen , "A New Learning Algorithm for Cooperative Agents in General-Sum Games", Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 2007.

[11] K.S. Narendra and M. A. L Thathachar, Learning automata: An introduction, Prentice Hall, 1989.

[12] M.A.L Thathachar and P.S. Sastry, “Varieties of Learning Automata: An Overview”, IEEE Transaction on Systems,

Man, and Cybernetics-Part B: Cybernetics, Vol. 32, No. pp. 6, 711-722, 2002.
[13] B. Jafarpour ,multi agent Cooperatiion using LA and PSO ,Msc Thesis, Computer Engineering and Information Technology Department,Amirkabir University of Technology, 2007.

زیر نویس ها

[1] Markov Decision Process (MDP)
2 Markov Game
3 Partial Observability Markov decision process
4Decentralized Partially Observable Markov decision process
5 Stochastic Game
6 Multi Agent Reinforcement Learning
7 Evaluation Function
8 Dynamic Programming
9 Fully Cooperative
10 Nash Equilibrium
11 Joint Action and Joint Strategy
12 Variable Sturcture Learning Automata
13 Multi Learning Automata
14 Exploration
15 Exploitation