بهبود کارایی در سیستمهای چند عامله مبتنی بر اتوماتای یادگیربا استفاده از مفهوم آنتروپی

بهروز معصومی[۱]؛ محمد رضا میبدی[۲]

چکیده

تا به حال برای مدل سازی سیستمهای چند عامله مدلهای مختلفی مبتنی بر مدل مارکوف پیشنهاد شده است که ازجمله آنها مدل بازی های مارکوفی را می توان نام برد . در این مقاله روشی جدید با استفاده از مفهوم آنتروپی در سیستمهای چند عامله مبتنی بر اتوماتاهای یادگیر با هدف بهبود کارایی ارائه شده است. سیستم چند عامله مورد نظر، برای پیدا کردن خطمشی بهینه بازی های مارکوف مورد استفاده قرار می گیرد. در الگوریتم پیشنهادی، در هر حالت ازمحیط به ازای هر عامل بازی یک اتوماتای یادگیر قرارداده می‌شود تا بتوانند عاملهای محیط را کنترل نمایند. تعداد اعمال هر اتوماتای یادگیر با توجه به حالت های مجاور با آن تعیین می‌گردد وترکیب اعمال اتوماتای یادگیرهر حالت از محیط، حالت بعدی را تعیین می کند. درالگوریتم مطرح شده اعمال انتخابی اتوماتاهای درون مسیر با توجه به هزینه مسیر طی شده و آنتروپی بدست آمده از بردارهای احتمالات اعمال اتوماتای یادگیر هر حالت، پاداش یا جریمه می گیرند.
آزمایشهای انجام گرفته نشان دادهاند که الگوریتمارائه شده از کارایی مناسبی از نظرسرعت رسیدن به راهحل بهینه برخورداراست.

کلمات کلیدی

سیستمهای چند عامله، اتوماتای یادگیر، بازی های مارکوف.

Improving Learning Automata based Multi-agent System using Entropy Concept

Behrooz Masoumi[۱]; Mohammad Reza Meybodi[۲]

[1] Department of IT& Computer Engineering, Islamic Azad University, Qazvin Branch. Qazvin, Iran.

[2] Department of Computer Engineering and IT, Amirkabir University of Technology, Tehran, Iran
mmeybodi@aut.ac.ir

ABSTRACT

So far, many models have been proposed for multi-agent systems based on Markov Models. One of these models is Markov Games. In this paper a new method based on the entropy concept is presented to improvement of learning automata based multi-agent systems. Multi-agent system is used for finding optimal policies in Markov games.. In the proposed algorithm, each agent residing in every state of the environment is equipped with a learning automaton. Each joint-action of the learning automaton corresponds to moving to one of the adjacent states. Each agent moves from one state to another and tries to reach the goal state. The actions taken by the learning automata along the path traversed by the agent are then rewarded or penalized using the cost of the traversed path or the entropy of probability vector of learning automaton of each agent in the next state according to a learning algorithm. The results of experiments have shown that the proposed algorithms perform better than the existing algorithms in terms of cost and the speed of reaching the optimal policy.

KEYWORDS

Markov Games, Learning Automata, Entropy.

[1] دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه آزاد قزوین Email: masoumi@Qiau.ac.ir

[۲] دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران. Email: mmeybodi@aut.ac.ir

۱. مقدمه

یک سیستم چندعامله، در برگیرنده جامعه‌ای از عاملهای هوشمند و خود مختاراست که در یک محیط درکنار یکدیگر سعی در انجام کاری خاص و رسیدن به هدفی مشخص دارند [۱]. امروزه در بسیاری از کاربردها ودر زمینههای مختلف صنعتی، نظامی، مخابراتی، اطلاعاتی، از سیستمهای پیچیده و توزیعشده چندعامله استفاده فراوانی میشود [۲]. تاکنون‌برای مدل سازی سیستمهای چند عامله مدلهای مختلفی مبتنی بر مدل مارکوف پیشنهاد شده است که ازجمله آنها مدل بازیهای مارکوفی[۱] را می توان نام برد [۳] [۴]. این بازی ها توسعه ای از فرآیندهای تصادفی مارکوف با چندین عامل بوده و به عنوان چارچوبی مناسب در تحقیقات یادگیری های چند عامله به ویژه یادگیری‌تقویتی چندعامله[۲] به‌کاررفته‌اند[۵] [۶]. بازیهای مارکوف، به صورت چند تایی (N,S,A,R,T) تعریف می گردند که در آن N تعداد عاملها، S مجموعه حالات، A مجموعه اعمال گروهی تمام عاملها، T تابع گذار حالات و R تابع پاداش با توجه به اعمال انتخابی عاملها در هر حالت تعریف می گردد. در این بازی ها، پاداش هر عامل وابسته به اعمال مشترک تمام عاملها بوده و حالت فعلی و گذار ازحالتی به حالت دیگر از ویژگی مارکوف تبعیت می کنند. در حالت خاص، در صورتیکه که فقط یک عامل وجود داشته باشد بازی مارکوفی را فرآیند تصادفی مارکوف[۳] (MDP) گویند و درصورتیکه فقط یک حالت وجود داشته باشد به آن بازی های نرمال[۴] گفته می شود.

بازی های مارکوف دارای انواع مختلفی بوده و از نظر پاداش به دو رده بازی های کاملا رقابتی (با مجموع صفر) و بازی جمع‌کلی تقسیم بندی شده اند. در بازی های رقابتی دونفره ساختار ماتریس پاداش هر عامل قرینه دیگری است. در بازی های مارکوف کلی، فرض می شود که برای پاداش‌عامل ها محدودیتی مطرح نیست. درحالت خاص، در صورتیکه پاداش یکسانی برای همه عاملها در نظر گرفته شودآنهارا کاملا همکارانه[۵] گویند و به آن فرآیندهای تصادفی مارکوف چند عامله (MMDP[۶]) نیزگفته می شود. دریک بازی های مارکوفی راه حل به معنای پیداکردن سیاستی برای انتخاب اعمال توسط عاملها در هر حالت است تا بتواند امید ریاضی مجموع کاهشیافته پاداشها[۷] را برای همه عاملها بیشینه نماید. درMMDP ها با توجه به اینکه همه عاملها پاداش یکسان دریافت می کنند عامل ها بایستی یادبگیرند تا درمورد سیاست بهینه توافق نمایند. در مقابل در بازی های مارکوفی کلی به دلیل وجود پاداش های متفاوت، پیدا کردن راه حل بهینه مشکل بوده و لذا نقاط تعادل[۸] (سیاست تعادل) در بازی مورد جستجو قرار می گیرند، وضعیتی که هیچ عاملی به تنهایی نمی تواند برای بهبود پاداشش سیاستش را تغییر دهد تا زمانیکه تمام عاملهای دیگر سیاستشان را ثابت نگه می دارند.

برای پیداکردن راه حل بهینه در بازی های مارکوفی، در شرایطی که شناخت کاملی از محیط وجود نداشته باشد و توابع پاداش و انتقال حالت مشخص نبوده و به صورت تجربی قابل مشاهده باشند استفاده از روشهای یادگیری تقویتی برای پیدا کردن نقاط تعادل مناسب است. در شکل کلی، این روش ها یک ورودی تابع انتخاب تعادل رادریافت نموده تا بتوانند با توجه شرایط تابع ارزش[۹] را محاسبه نموده و سیاست بهینه را یادبگیرند. این روشها با توجه به پیداکردن نقاط تعادل در هر مرحله از نظر محاسباتی پیچیدگی نسبتا بالایی را دارند و در بسیاری از مواقع شرایط خاصی را بایستی در نظر گرفت. از جمله این روشها می‌توان به روش NASH-Q [۷]، PARETO-Q [۸] [۹] وNASH BARGAINING [۱۰] اشاره نمود.

اتوماتاهای یادگیرنیز در حال حاضر به عنوان ابزاری ارزشمند در طراحی الگوریتمهای یادگیری تقویتی بوده به واسطه ویژگیهایی که دارند در بسیاری از کاربردهای جند عامله و محیط های ناشناخته مناسب هستند [۱۱] [۱۲]. سادگی ساختار، نیاز به اطلاعات و بازخورد کم از محیط، قابلیت مناسب برای سیستمهای توزیع شده و سیستمهای چند عامله با اطلاعات ناکامل و ارتباطات محدود، انعطاف پذیری و قابلیت تحلیل در بیشتر کاربرد ها از جمله این ویژگیها هستند. برای حل بازی های مارکوفی نیز الگوریتمهای مختلفی مبتنی بر اتوماتاهای یادگیر ارائه شده است. در [۱۳] روشی مبتنی براتوماتاهای یادگیر برای حل فرآیندهای تصادفی مارکوف چند عامله وبازی های مارکوفی در شرایط ارگودیک[۱۰] مطرح شده اند و نشان داده شده است که شبکه ای از اتوماتاهای یادگیر[۱۱] قادر به رسیدن به استراتژی های تعادل در بازی های مارکوفی می باشند. در [۱۴] یک راه حل کلی برای بازی های مارکوفی با مجموع کلی با استفاده از اتوماتای یادگیر ارائه شده است که در آن با توجه به بردارهای احتمالات اعمال اتوماتای یادگیر و محاسبه آنتروپی اطلاعاتی به دست آمده در هر حالت، اعمال انتخابی اتوماتای یادگیر پاداش یا جریمه دریافت می دارند. در [۱۵] روشی برای حل بازی های مارکوفی‌همکارانه (MMDP) با استفاده از پیداکردن هزینه کوتاهترین مسیر و پاداش دادن به اعمال اتوماتای مسیر بدست آمده ارائه شده است.

در این مقاله روشی مبتنی بر اتوماتاهای یادگیر برای حل بازی های مارکوف کلی برای خطمشی بهینه پیشنهاد شده است . در روش پیشنهادی، در هر حالت ازمحیط به ازای هر عامل، یک اتوماتای یادگیر قرارداده می شود تا بتوانند عاملهای محیط را کنترل نمایند. با توجه به هزینه مسیر بدست آمده و نیز آنتروپی[۱۲] بردار احتمالات اتوماتای یادگیر حالت جدید اعمال انتخابی اتوماتاها پاداش می گیرند. برای بررسی و ارزیابی روش پیشنهادی از محیط بازی های Grid Game برای شبیه سازی آزمایش ها استفاده شده است. درادامه مقاله، در بخش ۲ مفهوم بازی های مارکوفی توضیح داده شده و در بخش ۳ اتوماتاهای یادگیر و در بخش ۴ الگوریتم پیشنهادی مطرح و در بخش ۵ آزمایش های انجام شده در محیط بازی های Grid-Game و نتایج آزمایشها ارائه گردیده اند.

$$J(\alpha)=\lim_{l\to\infty}\frac{1}{l}E\left[\sum_{t=0}^{l-1}R^{x(t)x(t+1)}(\alpha)\right] \qquad (۱)$$

تعریف بازی های مارکوفی

بازی های مارکوف تعمیم فرآیندهای تصادفی مارکوف به‌حالت چندعامله است و بصورت زیر تعریف میشوند[۱۸]:

تعریف۲. بازی‌مارکوف بصورت چندتایی ⟨n,S,A۱..n ,T,R۱..n⟩ بیان میشود که‌درآن n تعداد عامل ها، S مجموعه حالات، Ai مجموعه اعمال هر عامل i (در فضای اعمال گروهی An× . . . ×A۲ × A۱) ، T تابع انتقال [۰,۱]→S×A×S و R تابع پاداش برای عامل iام با توجه به اعمال انتخابی در هر حالت است.

هر بازی مارکوفی با یک حالت بصورت یک بازی نرمال تکراری در تئوری بازی ها شناخته شده و ه‌ربازی مارکوفی با یک عامل بصورت یک فرآیند تصمیم گیری مارکوفی است. علاوه بر این هر عامل تابع پاداش خاص خودش را داراست. در حالتی که هر عامل پاداش مختلفی را داراست پیدا کردن سیاست بهینه برای تمام عاملها بسیار مشکل بوده لذا بجا ی آن نقاط تعادل بازی جستجو می شوند، وضعیتی که هیچ عاملی به تنهایی نمی تواند تا زمانیکه تمام عاملهای دیگر سیاستشان را ثابت نگه می دارند برای بهبود پاداش سیاستش را تغییر دهد.

۲. اتوماتاهای یادگیر

اتوماتای یادگیر، ماشینی است که میتواند تعدادی متناهی عمل را انجام دهد [۱۹]. 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 شکل ۱ مشاهده می‌شود.

ه‌محیط 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 [۱۳] (VSLA) 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 (۲) و (۳)یک‌نمونه‌ازالگوریتم‌های یادگیری خطی است. فرض میکنیم عمل α_i در مرحله nام انتخاب شود. در اینصورت پاسخ مطلوب از محیط بصورت:

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)]$$
$$p_j(n+1) = (1-a)p_j(n) \qquad \forall j \quad j \neq i \qquad (۲)$$

و پاسخ نامطلوب از محیط بصورت زیر می باشد.

$$p_i(n+1) = (1-b)p_i(n)$$
$$p_j(n+1) = (b/r - 1) + (1-b)p_j(n) \qquad (۳)$$
$$\forall j \quad j \neq i$$

در روابط (۲) و (۳)، a پارامتر پاداش و b پارامتر جریمه میباشند. با توجه به مقادیر a و b سه حالت را می‌توان درنظر گرفت: اگر a و b باهم برابر باشند، الگوریتم را $L\text{-}RP$، هنگامیکه b از a خیلی کوچکتر باشد، الگوریتم را $LR\text{-}\varepsilon P$ و اگر b مساوی صفر باشد آن را $L\text{-}RI$ مینامیم [۱۹]. شمای $S\text{-}LRP$ برای مدلهای Q و S براساس رابطه (۴) بیان‌می شود:

شکل۱- ارتباط بین اتوماتای یادگیر و محیط

اگر عمل α_i در مرحله nام انتخاب شود در این صورت طبق معادله (۴) داریم :

$$p_i(n+1) = p_i(n) + a(1-\beta_i(n))(1-p_i(n)) - a\beta_i(n)p_i(n)$$
$$p_j(n+1) = p_j(n) + a(1-\beta_i(n))(p_j(n)) + a\beta_i(n)\left[\frac{1}{r-1} - p_j(n)\right] - a.(1-\beta_i(n))p_j(n) \ if \ j <> i \qquad (۴)$$

r تعداد اعمال ممکن، aپارامتر پاداش وb پارامتر جریمه میباشند. برای اطلاعات بیشتر در باره اتوماتاهای یادگیر می توان به [۲۰] مراجعه نمود.

۳. روش پیشنهادی

در این بخش روش پیشنهادی برای حل بازی های مارکوف کلی به کمک اتوماتای یادگیر بررسی می‌گردد. مسئله کنترل فرآیند تصمیمگیری مارکف (زنجیره مارکف) تکعامله را میتوان با استفاده از شبکهای از اتوماتاهای یادگیر متصل به هم مدل نمود که در آن، کنترل از یک اتوماتای یادگیر به اتوماتای یادگیر دیگر منتقل میشود [۲۱]. هر وضعیت از زنجیره مارکف دارای یک اتوماتای یادگیر است که طی مراحل فرآیند و با استفاده از روابط (۲) و (۳) سعی میکند توزیع احتمال بهینه اعمال در آن وضعیت را یاد بگیرد. عامل با شروع از وضعیت اولیه تا رسیدن به هدف بر روی این شبکه از اتوماتاهای یادگیر حرکت میکند و در هر وضعیت، از اتوماتای یادگیر در آن وضعیت برای انتقال به یکی از وضعیتهای مجاور کمک میگیرد که این کار با استفاده از برداراحتمال اعمال اتوماتای یادگیر انجام میگیرد. در هر لحظه فقط یک اتوماتای یادگیر فعال بوده و انتقال از یک وضعیت به وضعیت دیگر، اتوماتای مربوط به وضعیت جدید را فعال مینماید. این فرآیند تا زمانیکه بردار احتمالهای کلیه اتوماتاهای یادگیر به پایداری برسد و یا شرط خاصی برقرار گردد تکرار میشود.

در یک بازی مارکوفی تغییر حالت با توجه به ترکیب اعمال مستقل انجام شده توسط اتوماتای یادگیردرهر حالت از محیط است. شبکه اتوماتای یادگیر به کار رفته برای MDP ها می تواند برای بازی های مارکوفی نیز با قرار دادن یک اتوماتا بازای هر عامل در هر حالت توسعه یابد [۱۳]. در الگوریتم پیشنهادی در هر حالت Si (i=۱..m و m تعداد حالات) از محیط بازی بازای هر عامل k یک اتوماتای یادگیر نظیر $LA(i, k)$ با ساختار متغیر قرار داده می‌شود. با توجه به تعداد حالت های مجاور با هر حالت از محیط، تعداد اعمال اتوماتای یادگیر مشخص می گردند. هر عامل با توجه به عمل انتخاب شده توسط اتوماتای یادگیر آن حالت و ترکیب گروهی اعمال انتخابی به حالت بعدی می رود. در ابتدا فرض می شود اتوماتاهای یادگیر تمام عملهای خود را با احتمالی یکسان انتخاب می‌کنند. در صورتیکه تغییر حالت ناشی از اعمال اتوماتای هر حالت منجر به ورود عامل به حالت نهایی (هدف) شود هزینه مسیر طی شده با هزینه مسیر قبلی مقایسه می شود اگر کمتر بود، هریک از اعمال انتخاب شده توسط اتوماتاهای یادگیر آن مسیر با توجه به میزان هزینه پاداش می گیرند و در غیر این صورت همه اتوماتاهای مسیر جریمه میشوند.

هزینه مسیر طی شده π_i ، طبق رابطه(۵)محاسبه‌می‌شود. در این رابطه، R_G پاداش مربوط به رسیدن به وضعیت هدف و $t_\pi(k)$ مدت زمان سپری شده از شروع حرکت عامل k تا رسیدن آن به وضعیت هدف، با طی مسیر π می‌باشد.

در صورتیکه حالت جدید حالت نهایی(هدف) نیست هر اتوماتا با توجه به بردارهای احتمال اعمال اتوماتای یادگیر و محاسبه آنتروپی اطلاعاتی بردارهای احتمالات بصورت زیر تعریف می شود پاداش یا جریمه می گیرند.

$$L_k = t_\pi / R_G \qquad (۵)$$

$$H(S_i) = -\sum_{j=1}^{N} P_j(S_i)\log(P_j(S_i)) \qquad (۶)$$

N تعداد اعمال اتوماتا در حالت S_i و $P_j(s_i)$ احتمال انجام عمل j در حالت S_i است. 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. در ابتدای الگوریتم که بردارهای احتمال اعمال اتوماتای یادگیر دارای مقادیر یکسان هستند یعنی $P۱=P۲=...Pr=۱/r$ میزان آنتروپی بیشترین مقدار است. لذا اتوماتای یادگیر دارای اطلاعات مفیدی برای رسیدن به هدف نبوده و عملهای خود را به صورت تصادفی انتخاب می‌کند (جستجو۱۴). با ادامه الگوریتم با تغیر احتمالات اعمال اتوماتای یادگیر میزان آنتروپی کم می شود و در حالت مینیم به صفر می رسد یعنی : $\exists i,\ P_i = 1 \wedge \forall\ j \neq i\ P_j = 0$. کاهش آنتروپی به این معنی است که اتوماتا ی یادگیر با احتمال بالایی یکی از اعمال خود را انتخاب می‌کند و دارای اطلاعات مفیدی برای رسیدن به هدف بوده و از این اطلاعات بهره برداری ۱۵می نماید. برای اینکه مقدار آنتروپی به مقداری بین ۰ و ۱ تبدیل شده تا به عنوان پاداش یعنیβدر اتوماتای یادگیر در محیطهای مدل S قابل استفاده باشد، آنتروپی آن حالت را به آنتروپی ماکزیمم تقسیم کرده و به آن آنتروپی نسبی گویند. الگوریتم نهایی در شکل ۲ نشان داده شده است.

۴. شبیه سازی انجام گرفته در محیط بازی *Game—Grid*

یکی از انواع بازیهایاتفاقیغیر رقابتی بازی های*Game-Grid* است که توسط Hu,Wellman ارائه شده است[۷]. این بازی یک بازی مارکوف دو نفری از نوع جمع کلی است. درشکل۳ دو نوع بازی Grid Game نشان داده شده اند. در بازی نوع اول (GG۱) دوعامل وجود دارندکه هردو می خواهند به یک هدف مشترک برسند و در بازی نوع دوم (GG۲) که یک بازی هماهنگی چند حالتی است، دو عامل و دو هدف وجود دارند.فرض بر این است که د وعامل از دو گوشه یک صفحه شروع کرده و سعی دارندتا باکمترین تعداد حرکت به هدف برسند. اعمال بازیکنان بطور همزمان انجام گرفته و هر یک از بازیکنان می توانند یکی از اعمال شمال، جنوب، شرق و یا غرب را انتخاب نمایند. مجموعه فضای‌حالات بصورت S={(۰ ۱), (۰ ۲), ... , (۸ ۷)} تعریف می شوند که هر حالت $S=(l_۱,l_۲)$ مختصات عاملهای ۱و۲ را نشان می دهد. عاملها همزمان نمی توانند در یک مختصات یکسان قرار گیرند. اگر دو عامل سعی در حرکت به یک مربع یکسان داشته باشند حرکت هردو باشکست مواجه شده و هردو یک واحد جریمه گردیده و در موقعیت قبلی باقی می مانند. اگر عاملها به دو مربع مختلف غیر هدف بروند هر دو پاداش صفر را دریافت می کنند و اگر یکی به هدف برسد ۱۰۰ واحد پاداش می گیرد.

The Proposed Algorithm

***Inputs*: a ,b**: reward and penalty parameter for each LA **M** : total training time
Initialize : In each state, a Learning Automaton of type S for each agent is placed. The set of actions of this LA is the set of permissible movements to other states.

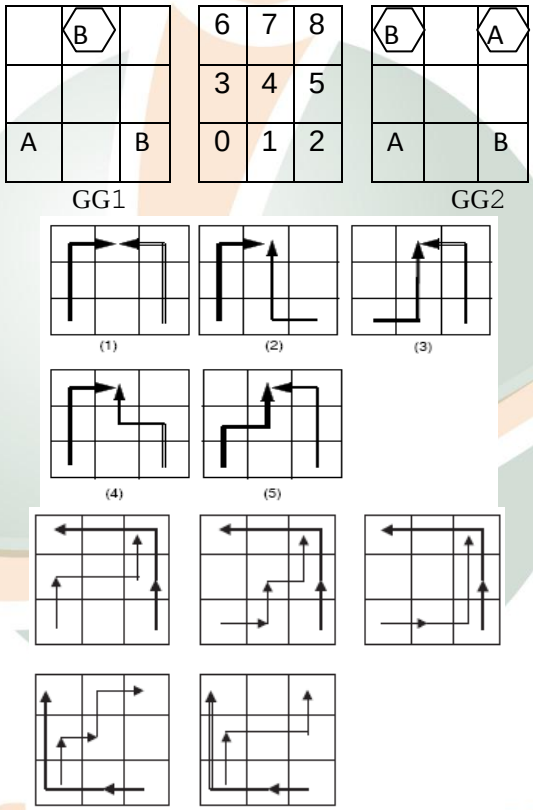
```
 for all states s, agents k do
    P(s,k) =1/number of permissible actions
    TotalReward(k) = 0  // total reward received by each agent
    LastAction(s,k) = 0  // last action played in each state
    T[k] = 0;              // Threshold for agent K ;
  end for
//Main Loop
for episode =1  to  M  do
    S = StartState;      //random or fixed
    Iteration=1
  while not done
     JointAction =∅;
      for all  agent k do cuncurrently
          Activate LA(S,k)
           Action = SelectAction (p( S, agent))
          JointAction = JointAction ∪  Action;
          NewState = GetNextState (S,JointAction)
          LastAction (state,agent) = action;
          TotalReward (agent)  = TotalReward(agent) +  GetReward(S,JointAction)
         StorePath(NewState);
        if  NewState = Goal state  then
          if CostLk < TK  then
            reward LA(S,k)in Path with signal Lk based on  Eq(5)
            else  penalize  LA_k^s  with  (1.0- Lk)
```

$$T_k \leftarrow T_k + \frac{1}{Iteration}(L_k - T_k)$$

endif
else Train *LA(S,k)* with reward signal Eq(6)
endif
end for
S=s',
Increment (Iteration)
end while
end for episode

شکل ۲. الگوریتم پیشنهادی

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.



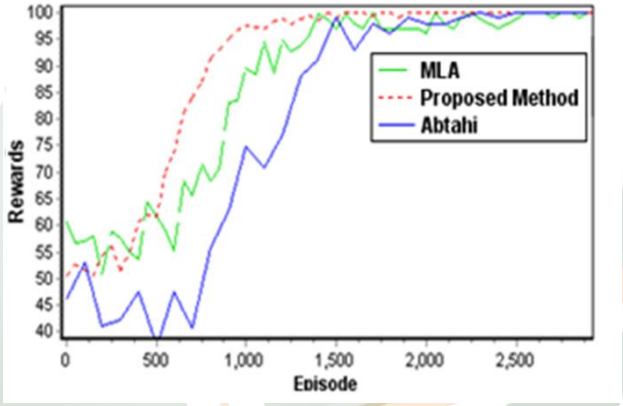
شکل ۳. بازی Grid World و نمایش مختصات بازی به همراه راه حلهای بهینه دردوبازی

یک مسیر دنباله ای از اعمال انتخاب شده از نقطه شروع تا پایان را نشان می دهد. به کوتاهترین مسیری که یک عامل بدون تداخل با عاملهای دیگرطی می نماید مسیر بهینه گفته می شود. در این بازی فرض می شود عاملها از موقعیت هدف در ابتدای بازی آگاهی نداشته و همچنین از پاداش سراسری یکدیگر اطلاع ندارند. عاملها اعمالشان را همزمان انتخاب نموده و فقط می توانندازاعمال قبلی عاملهای دیگروحالت فعلی (موقعیت مشترک هر دو عامل) آگاهی داشته باشند.
برای حل مساله با توجه به روش ارائه شده در هر حالت از محیط بازای هر عامل یک اتوماتای یادگیر در نظر گرفته شده است. هرعامل با توجه به عمل مشترک گروهی ناشی از اعمال انتخابی اتوماتای یادگیر هر حالت به حالت جدید میرود و مسیر انتخابی نگهداری می شود. در صورتیکه حالت جدید حالت نهایی باشد هزینه مسیر طی شده با هزینه مسیر قبل مقایسه و در صورت بهبود به تمام اعمال انتخابی اتوماتاهای یادگیر آن مسیر پاداش داده می شود. در این مساله با توجه به اینکه حالت تکراری باعث افزایش طول مسیر می شود در صورتی که حالت جدید قبلا مشاهده شده باشد اعمال انتخابی اتوماتاهای موجود جریمه تعلق می گیرد و درغیراینصورت از آنتروپی بردار احتمالات اعمال اتوماتای یادگیر حالت جدید به عنوان پاداش یا جریمه اتوماتا استفاده می شود.

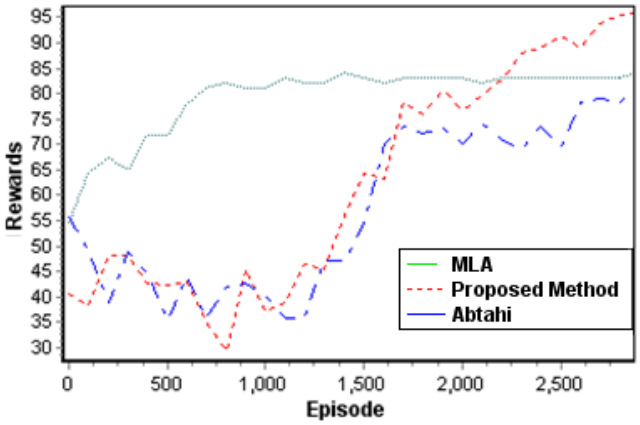
مقایسه الگوریتم پیشنهادی با دیگر روشها

در ابتدا روش پیشنهادی در حل بازی های GG۱وGG۲ با الگوریتمهای ارائه شده در [۱۴] یعنی الگوریتم MLA و [۱۵] Abtahi مقایسه می گردد. پارامترهای یادگیری را در تمام الگوریتمهای مطرح شده یکسان در نظر گرفته و از الگوریتم LRI در اتوماتای یادگیر استفاده می شود. آزمایشها به خاطر استحکام نتایج ۲۰۰ بار تکرار شده ودر هر بار ۳۰۰۰ اپیزود آزمایش شده است. درابتدا فرض می شود که نقطه شروع مختصات $s.=(۰\ ۲)$ بوده و در هر اپیزود جدید عامل ها از همان نقطه مبدا مجددا شروع می کنند. برای بررسی میانگین پاداش تجمعی بدست آمده در هر اپیزود در ۲۰۰ آزمایش نشان داده شده است. شکل ۴ نمونه اجرای الگوریتمها را در بازی GG۱ و شکل ۵ اجرای الگوریتمهاردربازی GG۲ نشان می دهد. همانطور که در شکل ۵ دیده می شود پس از تعدادی تکرار میانگین پاداش به سمت بیشترین افزایش می یابد. در آزمایش انجام گرفته پارامتر یادگیری $a=۰.۰۱$ در نظر گرفته شده است و پاداش بدست آمده در هر اپیزود نشان داده شده است. همانطور که در شکل نیز ۵ دیده می شود در هر دو بازی الگوریتم پیشنهادی رفتار بهتری را ارائه می دهد.

در آزمایش بعدی نقش پارامتر یادگیری در سرعت همگرایی مورد بررسی قرار می گیرد. همانطور که در شکلهای ۶و۷ دیده می شود، میزان پارامتر یادگیری نقش مهمی را در همگرایی الگوریتم ایفا می نماید. افزایش مقدار پارامتر a در بسیاری از مواقع باعث افزایش سرعت همگرایی می گردد. همچنین باید در نظر داشت در مواقعی که افزایش پارامتر یادگیری را داریم ممکن است باعث قرارگرفتن در نقاط بهینه محلی شده و باعث شود همگرایی بهینه دچار مشکل گردد. این مساله در بازی GG۲ کاملا مشهود است. کاهش میزان پارامتر یادگیری اگرچه ممکن است نتایج بهتری ارائه دهد ولی باعث می شود تعداد تکرار ها برای رسیدن به همگرایی بیشتر گردد.

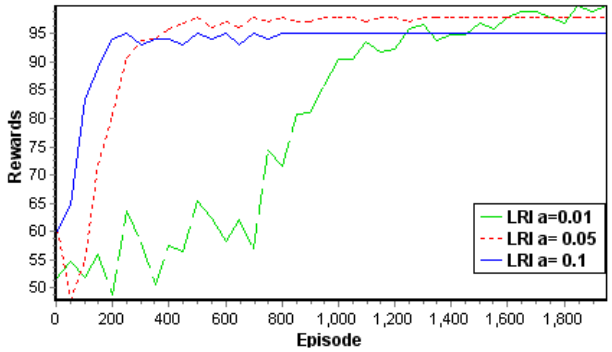


شکل ۴. مقایسه الگوریتم های مختلف برروی مساله GG۱ در ۳۰۰۰ اپیزود با پارامتر a=۰.۰۱,b=۰



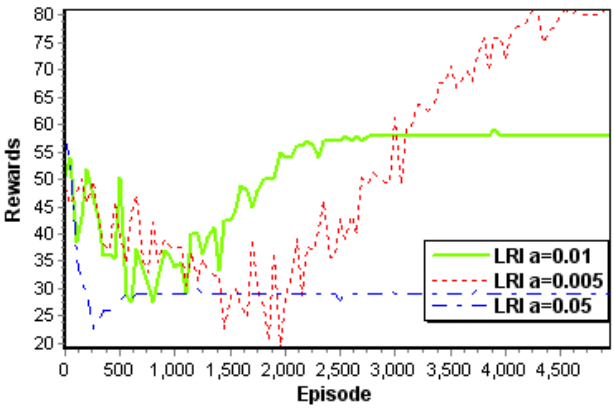
شکل ۵. مقایسه الگوریتم های مختلف برروی مساله GG۲

در ۳۰۰۰ اپیزود با پارامتر a=۰.۰۱,b=۰

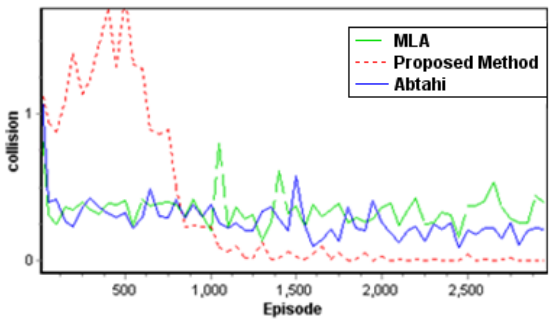


شکل ۶. بررسی رفتار پارامتر یادگیری در همگرایی الگوریتم درGG۱

برای بررسی رفتار الگوریتمها آزمایش دیگری انجام گرفت. که در آن متوسط تعداد برخوردهایی که بین دو عامل در هر اپیزود در ۲۰۰ آزمایش رخ می دهد محاسبه می شود. همانطور که در شکل ۸ دیده می شود در ابتدا الگوریتم پیشنهادی در حالت جستجو (Explore) بوده و به مرور طی فرآیند یادگیری به حالت استناد بر تجربه (Exploit) رسیده و لذا سریع همگرا می شود.



شکل ۷. بررسی رفتار پارامتر یادگیری در همگرایی الگوریتم در GG۲



شکل ۸. تعداد برخوردهای دو عامل در هر اپیزود برای بازی GG۱

۵. نتیجه گیری

در این مقاله روشی مبتنی بر اتوماتاهای یادگیر برای حل بازی های مارکوفی ارائه گردید. در این روش با توجه به هزینه مسیر طی شده تا رسیدن به هدف، اعمال انتخابی اتوماتاهای یادگیر در طول مسیر پاداش یا جریمه می گیرند. در طول مسیرنیز با توجه به آنتروپی به دست آمده از بردارهای احتمال اتوماتای یادگیرحالت جدید به عنوان پاداش های کمکی جهت پاداش یا جریمه اتوماتا ها استفاده می شود و این کار باعث افزایش کارایی می گردد. با توجه به الگوریتم ارائه شده ونتایج بدست آمده می بینیم روش مورد نظر رفتار مناسبتری را نسبت به روشهای دیگربرای حل بازی های مارکوفی نشان می دهد. تعداد تکرارها، پارامترهای یادگیری و سرعت رسیدن به نقاط تعادل را تعیین می نمایند. تنظیم پارامترهای پاداش و جریمه اتوماتاها می تواند کارایی رسیدن به راه حل بهینه را افزایش دهد. با توجه به نتایج به دست آمده اتوماتاهای یادگیر مدل مناسب یادگیری و هماهنگی بین عاملها در سیستمهای چندعامله بوده و می تواند به عنوان راه حلی مناسب و کارا در بازی های مارکوف به کار روند.

۶. مراجع

[**1**] G. Weiss; Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, Cambridge, MA: MIT Press, **1999** .

[**2**] A. K. Goel; V. Kummar and S. Irivasan; "Application of Multi-Agent System & Agent Coordination", **2**nd National Conference Mathematical Techiniques: Emerging Paradigms for Electronics and IT Industries(MATEIT-**2008**), New Dehli, India, **2008** .

[**3**] Y. Shoham; Multiagent Systems: Algorithmic Game Theoretic and Logical Foundations, Cambridge University Press, **2009** .

[**4**] M. L. Littman; "Markov Games as a Framework for Multi-agent Reinforcement Learning", In Proceedings of the **11**th International Conference on Machine Learning, pp. **322** – **328** , **1994** .

[**5**] J. Osborne; A. Rubinstein. A Course in Game Theory, MIT Press, Cambridge, MA, **1994** .

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

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

[**8**] M. Song; G. Gu and G. Zhang; "Pareto-Q Learning Algorithm for Cooperative Agents in General-Sum Games", CEEMAS**2005** , pp. **576-578** , ۲۰۰۵.

[**9**] M. Song; J. Bai and 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** .

[**10**] H. Qio; F. Szidarovszky; Rozenblit and L. Yong; "Multi-agent Learning Model with Bargaining ", Proceedings of the **38**th conference on winter simulation, pp. **934 - 940** , **2006** .

[**11**] M. R. Khojasteh and M. R. Meybodi; "Evaluating Learning Automata as a Model for Cooperation in Complex Multi-Agent Domains", Lecture Notes in Artificial Intelligence, Springer Verlag, LNAI **4434** , pp. **409-416** , **2007**

[**12**] A. Now´e; K. Verbeeck and M. Peeters; "Learning Automata as a basis for Multi-agent Reinforcement Learning", Lecture Notes in Computer Science, vol. **3898** , pp. **71**–**85** , **2006** .

[**13**] P. Vrancx; K. Verbeeck and A. Nowe; "Decentralized Learning in Markov Games", IEEE Transactions on Systems, Man and Cybernetics (Part B: Cybernetics), vol. **38** , iss. **4**, pp. **976-81** , **2008** .

[**14**] B. Masoumi ; M. R. Meybodi and B. Jafarpour; "Solving General Sum Stochastic Games using Learning Automata", Proceedings of the second Joint Congress on Fuzzy and Intelligent Systems, Malek Ashtar University of Technology, Tehran, Iran, pp. **28-30** , **2008** .

[**15**] F. Abtahi and M. R. Meybodi; "Solving Multi-Agent Markov Decision Processes Using Learning Automata", Proceedings of the **6**th International Symposium on Intelligent Systems (SISY**2008**), Subotica, Serbia, September **26-27** , **2008** .

[**16**] M. L. Puterman; Markov Decision Processes: Discrete Stochastic Dynamic Programming, John Wiley & Sons, Inc. New York, NY, USA, **1994** .

[**17**] J. C. H. Watkins and P. Dayan; Q-Learning, Machine Learning. Vol. **3**, pp.**279** –**292** , **1992** .

[**18**] F. Thusijsman; Optimality and Equilibria in Stochastic Games, (Centrum voor Wiskunde en Informatica, Amsterdam, **1992**).

[**19**] K. Narendra and M. Thathachar; Learning Automata: An Introduction, Prentice-Hall International, Inc, **1989** .

[**20**] M. A. L. Thathachar, P. Sastry, "Varieties of Learning Automata: An Overview", IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. **32** , No. **6**, pp. **711-722** , **2002** .

[**21**] R. M. Wheeler and K. S. Narendra; "Decentralized Learning in Finite Markov Chains", IEEE Transactions on Automatic Control, pp. **519** – **526** , **1986** .

زیر نویس ها

---

۱ Markov Game
۲ Multi Agent Reinforcement Learning
۳ Markov Decision Process
۴ Normal Game
۵ Fully Cooperative
۶ Multi-Agent MDP
۷ Sum of discounted expected rewards
۸ Equilibrium Point
۹ Value Function
۱۰ Ergodic
۱۱ Network of Learning Automata
۱۲ Entropy
۱۳ Variable Sturcture Learning Automata
۱۴ Exploration
۱۵ Exploitation