

# مدلی مبتنی بر آنتروپی و اتوماتاهای یادگیر

## برای حل بازی‌های تصادفی

بهروز معصومی و محمدرضا میدی

مدل‌سازی این‌گونه مسایل و یافتن راه حل‌های بهینه برای آنها هستند. برای حل مسایل تصمیم‌گیری مارکوف، یادگیری تقویتی سیار مورد استفاده قرار گرفته است. بازی‌های غیر قطعی (تصادفی)<sup>۱</sup> (SG) به عنوان توسعه‌ای از فرآیندهای چندعامله در سیستم‌های چندعامله و مدل‌سازی آنها حائز اهمیت بوده و به عنوان چارچوبی مناسب در تحقیقات یادگیری تقویتی چندعامله به کار رفته‌اند. در حال حاضر اتوماتاهای یادگیر به عنوان ابزاری ارزشمند در طراحی الگوریتم‌های یادگیری چندعامله به کار رفته‌اند. در این مقاله مدلی مبتنی بر اتوماتای یادگیر و مفهوم آنتروپی برای حل بازی‌های غیر قطعی و پیداکردن سیاست بهینه در این بازی‌ها ارائه شده است. در مدل پیشنهادی به ازای هر عامل در هر حالت از محیط بازی یک اتوماتای یادگیر با ساختار متغیر از نوع S قرار داده شده است که اعمال بهینه را در هر حالت یاد می‌گیرند. تعداد اعمال هر اتوماتا با توجه به همسایگان مجاور هر حالت تعیین شده و ترکیب اعمال اتوماتاهای حالت بعدی محیط را انتخاب می‌کند. در مدل پیشنهادی از آنتروپی بردار احتمالات اتوماتای یادگیر حالت جدید برای کمک به پاداش‌دهی اتوماتاها و بهبود یادگیری استفاده شده است. برای بررسی و تحلیل رفتار الگوریتم یادگیری پارامتری بنام آنتروپی کلی تعریف گردیده که میزان همگرایی را در الگوریتم یادگیری بیان می‌کند. در نهایت الگوریتمی اصلاح یافته با ایجاد تعادل بین جستجو و استناد بر تجربیات پیشنهاد شده است. نتایج آزمایش‌های نشان می‌دهد الگوریتم ارائه شده از کارایی مناسبی از هر دو جنبه هزینه و سرعت رسیدن به راه حل بهینه برخوردار است.

بازی‌گیری تقویتی چندعامله به سرعت در حال توسعه بوده و روش‌های متنوع و مختلفی را در حوزه‌های رقابتی، همکارانه و ترکیبی در بر می‌گیرد به طوری که ارتباطی بین حوزه‌های مختلف علوم نظری تئوری بازی‌ها، بهینه‌سازی و یادگیری در بازی‌ها برقرار ساخته است. بازی‌های تصادفی مدلی توسعه یافته از بازی‌های تکرارشونده هستند که در هر لحظه از زمان، بازی در یک حالت قرار دارد. گذار از حالتی به حالت جدید بر پایه تابع احتمالاتی با توجه به حالت قبلی و تعامل بین عامل‌ها در حالت قبلی انجام می‌گیرد. هر حالت در یک بازی تصادفی می‌تواند به صورت یک فرآیند تصمیم‌گیری مارکوف دیده شود و هر بازی تصادفی با یک عامل به صورت یک فرآیند تصمیم‌گیری مارکوف است.

بازی‌های تصادفی دارای انواع متفاوتی بوده و از نظر پاداش به دو نوع بازی‌های با مجموع صفر<sup>۲</sup> (بازی‌های رقابتی) و بازی‌های با جمع کلی<sup>۳</sup> تقسیم‌بندی شده‌اند. در این بازی‌ها، عامل‌ها اعمالشان را به طور همزمان انجام داده و پاداش هر عامل به عمل گروهی انتخاب شده توسط همه عامل‌ها بستگی دارد. در بازی‌های رقابتی دونفره ساختار پاداش عامل‌ها محدودیتی است که پاداش هر عامل منفی مقدار پاداش عامل دیگر است. در بازی‌های با جمع کلی، فرض می‌شود که برای پاداش عامل‌ها محدودیتی مطرح نیست. در حالت خاص بازی‌های مارکوف کلی، در صورتی که پاداش یکسانی برای همه عامل‌ها در نظر گرفته شود آنها را کاملاً همکارانه<sup>۴</sup> گویند و به آن فرآیندهای تصادفی مارکوف چندعامله (MMDP)<sup>۵</sup> نیز گفته می‌شود.

در بازی‌های مارکوفی، راه حل به معنای پیداکردن سیاستی برای انتخاب اعمال توسط عامل‌ها در هر حالت است تا بتواند امید ریاضی مجموع کاهش یافته پاداش‌ها<sup>۶</sup> را برای همه عامل‌ها بیشینه نماید. در

چکیده: بازی‌های غیر قطعی (تصادفی) به عنوان توسعه‌ای از فرآیندهای چندعامله با چندین عامل در سیستم‌های چندعامله و مدل‌سازی آنها حائز به کار رفته‌اند. در حال حاضر اتوماتاهای یادگیر به عنوان ابزاری ارزشمند در طراحی الگوریتم‌های یادگیری چندعامله به کار رفته‌اند. در این مقاله مدلی مبتنی بر اتوماتای یادگیر و مفهوم آنتروپی برای حل بازی‌های غیر قطعی و پیداکردن سیاست بهینه در این بازی‌ها ارائه شده است. در مدل پیشنهادی به ازای هر عامل در هر حالت از محیط بازی یک اتوماتای یادگیر با ساختار متغیر از نوع S قرار داده شده است که اعمال بهینه را در هر حالت یاد می‌گیرند. تعداد اعمال هر اتوماتا با توجه به همسایگان مجاور هر حالت تعیین شده و ترکیب اعمال اتوماتاهای حالت بعدی محیط را انتخاب می‌کند. در مدل پیشنهادی از آنتروپی بردار احتمالات اتوماتای یادگیر حالت جدید برای کمک به پاداش‌دهی اتوماتاها و بهبود یادگیری استفاده شده است. برای بررسی و تحلیل رفتار الگوریتم یادگیری پارامتری بنام آنتروپی کلی تعریف گردیده که میزان همگرایی را در الگوریتم یادگیری بیان می‌کند. در نهایت الگوریتمی اصلاح یافته با ایجاد تعادل بین جستجو و استناد بر تجربیات پیشنهاد شده است. نتایج آزمایش‌های نشان می‌دهد الگوریتم ارائه شده از کارایی مناسبی از هر دو جنبه هزینه و سرعت رسیدن به راه حل بهینه برخوردار است.

**کلید واژه:** آنتروپی، اتوماتاهای یادگیر، بازی‌های تصادفی، سیستم‌های چندعامله.

### ۱- مقدمه

یک سیستم چندعامله، در برگیرنده جامعه‌ای از عامل‌های هوشمند و خودمحختار است که در یک محیط در کنار یکدیگر در حال کار بوده و سعی در انجام کاری خاص و رسیدن به هدفی مشخص دارند [۱]. سیستم‌های چندعامله برای مدل‌سازی، تحلیل و طراحی سیستم‌هایی که کنترل بین تصمیم‌گیرندهای خودمحختار به صورت توزیع شده است مناسب هستند. امروزه در بسیاری از کاربردها و در زمینه‌های مختلف صنعتی، نظامی، مخابراتی و اطلاعاتی از سیستم‌های پیچیده و توزیع شده چندعامله استفاده فراوانی می‌شود [۲] و [۳]. برای حل بسیاری از مسایل مهم دنیای واقعی مانند برخی از کاربردهای رباتیک، مسیریابی در شبکه، زمان‌بندی و تصمیم‌گیری اقتصادی نیازمند برنامه‌ریزی در حالت غیر قطعی هستیم. مدل‌های فرآیند تصمیم‌گیری مارکف<sup>۷</sup> (MDP) چارچوب مناسبی برای

این مقاله در تاریخ ۱۵ بهمن ماه ۱۳۸۷ دریافت و در تاریخ ۹ اردیبهشت ماه ۱۳۸۹ بازنگری شد.

بهروز معصومی، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه آزاد اسلامی قزوین، قزوین، (email: masoumi@Qiau.ac.ir).  
محمدرضا میدی، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، (email: mmeybodi@aut.ac.ir).

1. Markov Decision Process

2. Stochastic Games

3. Markov Game

4. Zero - Sum Games

5. General - Sum Games

6. Fully Cooperative

7. Multi - Agent MDP

8. Sum of Discounted Expected Rewards

مشخص نبوده و عامل‌ها از اطلاعات دیگران آگاهی کامل ندارند. در این راستا الگوریتمی با استفاده از اتوماتاهای یادگیر ارائه می‌شود تا با بهره‌گیری از مفهوم آنترپریز برای بهینه‌سازی استراتژی‌ها، الگوریتمی با قابلیت یادگیری بهتر و سریع‌تر برای یافتن خط مشی بهینه ارائه دهد. برای آزمایش مدل ارائه شده از محیط بازی‌های Grid Game استفاده شده است. در ادامه مقاله، بخش ۲ به تعریف یادگیری تقویتی چندعامله و تعریف بازی‌های تصادفی و مفهوم راه حل در آنها می‌پردازد. در بخش ۳ مفهوم اتوماتای یادگیر و الگوریتم پیشنهادی و در بخش ۴ نتایج و تحلیل آزمایش‌ها ارائه گردیده‌اند.

## ۲- تعریف بازی‌های تصادفی و مفهوم راه حل در آنها

در این بخش به طور خلاصه تعریف فرآیندهای تصادفی مارکوف و بازی‌های تصادفی مطرح می‌شود. در سیستم‌های چندعامله، الگوریتم‌های یادگیری تقویتی مانند یادگیری  $Q$ ، با موقیت در سیاری از کاربردها مورد استفاده قرار گرفته‌اند [۴]. اکثر الگوریتم‌های یادگیری تقویتی چندعامله، بر پایه روش‌های تکعامله بنا نهاده شده‌اند. در حالت یادگیری تکعامله که در آن عامل به طور مستقل در حال یادگیری باشد، فرآیند یادگیری تقویتی را می‌توان به صورت فرآیند تصمیم‌گیری مارکوف تعریف نمود.

### ۱-۲ فرآیند تصادفی مارکوف (MDP)

به طور رسمی فرآیند تصادفی مارکوف را به صورت زیر تعریف می‌کنند: تعریف ۱: فرآیند تصادفی مارکوف به صورت چندتایی  $\langle S, A, R, T \rangle$  نشان داده می‌شود که در آن  $S$  مجموعه متناهی از وضعیت‌ها،  $A$  مجموعه عملیات قابل دسترس برای عامل،  $T: S \times A \times S \rightarrow [0, 1]$  احتمال انتقال از وضعیت جاری به وضعیت بعدی با انجام عمل  $a$  است و  $R: S \times A \rightarrow \mathbb{R}$  بیانگرتابع پاداش است که یک مقدار عددی را بر می‌گرداند.

علاوه بر این ما نیاز به مفهوم سیاست داریم که آنچه باید در وضعیت فعلی محیط انجام شود را بیان می‌کند. سیاست نگاشتی از وضعیت به عمل است. ما سیاست را در لحظه  $t$  به صورت  $A_t: \pi: S \rightarrow A$  نشان می‌دهیم. سیاستی را ایستا گویند اگر وابسته به زمان نباشد. در یک MDP هدف عامل پیداکردن استراتژی  $\pi: A \rightarrow S$  است که امید ریاضی مجموع کاهش یافته پاداش‌ها را بیشینه نماید. برای هر خط مشی  $\pi$  که عامل می‌تواند دنبال کند، بر روی وضعیت‌ها تابعی به نام تابع ارزیابی  $V$  به شکل رابطه زیر تعریف می‌شود

$$V(\pi, s) = \sum_{r=0}^{\infty} \gamma^r E(r | \pi, s) \quad (1)$$

این تابع، نگاشتی از مجموعه وضعیت‌ها به مقدار ارزش آنها می‌باشد.  $s$  یک وضعیت خاص،  $r$  وضعیت اولیه،  $\pi$  پاداش در زمان  $t$  و  $\gamma$  ضریب کاهش در محدوده  $(0, 1]$  است.  $V(\pi, s)$  ارزش وضعیت  $s$  تحت استراتژی  $\pi$  را نشان می‌دهد. بنابراین هدف، یادگیری خطا مشی بهینه  $\pi^*$  است. پیداکردن سیاست بهینه می‌تواند به صورت یک مسئله بهینه‌سازی مطرح شود که می‌تواند با استفاده از الگوریتم‌های برنامه‌ریزی یوپیا حل شود. در این صورت روشی تقریبی برای تخمین مقادیر بهینه  $V(s)$  به کار می‌رود که روش Value Iteration نام دارد. راه حل استاندارد از طریق یک روش جستجوی تکرارپذیر است که بر پایه استفاده از معادله Bellman استوار است و به صورت [۲] تعریف می‌شود [۱۶]

MMDP‌ها با توجه به این که همه عامل‌ها پاداش یکسان دریافت می‌کنند عامل‌ها بایستی یادگیرند تا در مورد سیاست بهینه توافق نمایند. در مقابل در بازی‌های مارکوفی کلی به دلیل وجود پاداش‌های متفاوت، پیداکردن راه حل بهینه مشکل بوده و لذا نقاط تعادل<sup>۱</sup> (سیاست تعادل) در بازی مورد جستجو قرار می‌گیرند. به عبارت دیگر مسئله پیداکردن سیاست بهینه در بازی‌های تصادفی در حوزه تئوری بازی‌ها بوده و راه حل پایه در آنها رسیدن به نقاط تعادل شش<sup>۲</sup> می‌باشد.

برای حل بازی‌های اتفاقی اعم از بازی‌های رقابتی و غیر رقابتی الگوریتم‌های یادگیری تقویتی متعددی به کار رفته‌اند. این الگوریتم‌ها از نظر خصوصیت ایستایی و پویایی محیط در دو رده قرار می‌گیرند. ما در این مقاله فرض محیط‌های پویا در بازی‌های تصادفی با چند حالت و غیر رقابتی را در نظر می‌گیریم. بیشتر الگوریتم‌های یادگیری چندعامله بر پایه پیداکردن نقاط تعادل عمل می‌کنند. Littman (۱۹۹۴) راه حل minmax را در بازی‌های با مجموع صفر ارائه می‌دهد [۵]. Hu و Wellman (۱۹۹۸) الگوریتمی برای حل بازی‌های با مجموع کلی ارائه دادند [۷]. آنها الگوریتمی به نام Q-Nash را پیشنهاد دادند که تحت شرایط خاص به سیاست تعادل نش همگرا می‌شد [۸]. Greenwald و همکارانش (۲۰۰۳) الگوریتم مبتنی بر تعادل همیستگی<sup>۳</sup> به جای تعادل را برای بازی‌های تصادفی ارائه می‌دهد [۹]. Meiping Song و همکاران (۲۰۰۵) الگوریتمی به نام Q-Pareto را برای بازی‌های با مجموع کلی همکارانه مطرح کردند که از بهینه پارتو بر اساس قوانین اجتماعی استفاده می‌کرد [۱۰] و [۱۱]. بررسی دقیق الگوریتم‌های مختلف و رده‌بندی‌های دیگر این الگوریتم‌ها در [۴] به تفصیل گزارش شده است.

اتوماتاهای یادگیر نیز در حال حاضر به عنوان ابزاری ارزشمند در طراحی الگوریتم‌های یادگیری تقویتی بوده و به واسطه ویژگی‌هایی که دارند در بسیاری از کاربردهای چندعامله و محیط‌های ناشناخته مناسب هستند [۱۲] و [۱۳]. سادگی ساختار، نیاز به اطلاعات و بازخورد کم از محیط، مناسب‌بودن در سیستم‌های توزیع شده و سیستم‌های چندعامله با اطلاعات ناکامل و ارتباطات محدود، انعطاف‌پذیری و قابلیت تحلیل در بیشتر کاربردها از جمله این ویژگی‌ها هستند. در [۱۳] الگوریتم‌های مبتنی بر اتوماتاهای یادگیر برای حل فرآیندهای تصادفی مارکوف با شرایط مختلف و بازی‌های مارکوفی مطرح شده‌اند. بازی‌های اتوماتاهای در حالت وجود چند عامل و یک حالت و بازی‌های تصادفی در شرایط وجود چندین عامل و چندین حالت از اهمیت خاصی برخوردار هستند. در [۱۴] روشی مبتنی بر اتوماتاهای یادگیر برای حل فرآیندهای تصادفی مارکوف چندعامله و بازی‌های مارکوفی در شرایط ارگودیک<sup>۴</sup> مطرح شده و نشان داده شده است که شبکه‌ای از اتوماتاهای یادگیر قادر به رسیدن به استراتژی‌های تعادل در بازی‌های مارکوفی می‌باشند.

در [۱۵] ترکیب تیمی اتوماتاهای برای حل بازی‌های  $n$ -نفره تیمی استفاده شده و نشان داده شده است در صورت استفاده از الگوریتم  $L_{RI}$  توسط تمام اعضای تیم الگوریتم به تعادل نش همگرایی پیدا می‌کند. هدف این مقاله طراحی الگوریتم یادگیری مبتنی بر اتوماتاهای یادگیر برای حل بازی‌های تصادفی با مجموع کلی است که در آنها پاداش کلی عامل‌ها از قبل

1. Equilibrium Points
2. Nash Equilibrium
3. Correlated Equilibrium
4. Ergodic
5. Network of Learning Automata

### ۳-۲ استراتژی‌های تعادل

در بازی‌های با جمع کلی، راه حل مینا تعادل نش است [۱۸]. نظریه تعادل‌های نش می‌گوید که در هر بازی به‌فرض آن که بازیکنان معقولانه استراتژی‌های خود را انتخاب کرده و به‌دنبال به‌دست آوردن حداکثر بهره (سود) از بازی باشند، حداقل یک استراتژی برای به‌دست آوردن بهترین نتیجه برای هر بازیکن قابل انتخاب است که اگر بازیکن راهکار دیگری به‌غیر از آن را انتخاب کند، نتیجه بهتری به‌دست نخواهد آورد [۱۹].

**تعريف ۳:** در یک بازی تصادفی  $\Gamma$ ، یک تعادل نش به صورت چندتایی استراتژی‌های  $(\pi_1^*, \dots, \pi_n^*)$  تعریف می‌شود به‌طوری که برای هر  $s \in S$

$$V^i(s, \pi_1^*, \dots, \pi_n^*) \geq V^i(s, \pi_1^*, \dots, \pi_{i-1}^*, \pi_i^*, \pi_{i+1}^*, \dots, \pi_n^*) \quad (4)$$

for all  $\pi^i \in \prod^i$

$\prod^i$  مجموعه استراتژی‌های قابل دسترس عامل  $i$  است. استراتژی‌هایی که یک تعادل نش را می‌سازند می‌توانند استراتژی‌های ایستا باشند. قضیه زیر نشان می‌دهد حداقل یک تعادل در استراتژی‌های ایستا وجود دارد:

**قضیه ۱:** (Fink ۱۹۶۴) هر بازی اتفاقی کاهش‌پذیر  $n$  نفره حداقل یک نقطه تعادل نش در استراتژی‌های ایستا را دارد [۲۰].

### ۴ حل بازی‌های تصادفی

در بازی‌های اتفاقی برخلاف MDP‌ها یک راه حل بهینه‌ای که مستقل از عامل‌های دیگر باشد متحمل نیست. لذا نسبت به حالت تک عامله مسئله مشکل‌تر است. همچنین ممکن است که چندین نقطه تعادل وجود داشته و همانگی عامل‌ها برای توافق کار مشکلی است. مفهوم راه حل در بازی‌های اتفاقی به معنای پیداکردن نقطه تعادل منحصر به‌فرد می‌باشد. الگوریتم‌های مختلفی برای حل این بازی‌ها پیشنهاد شده‌اند که در یک رده‌بندی می‌توان به روش‌هایی که از تئوری بازی‌ها استفاده می‌کنند و روش‌های مبتنی بر یادگیری تقویتی اشاره نمود. در روش‌های مبتنی بر تئوری بازی‌ها، نیاز به مدل محیط داشته و توابع  $T$  و  $R$  نیز بایستی مشخص باشند. هدف این الگوریتم‌ها محاسبه مقدار تعادل بازی (پاداش مورد انتظار کاهش‌یافته برای هر عامل) است، لذا نیاز به فرضیات قوی‌تری از رفتار عامل‌ها دارند. در مقابل، الگوریتم‌های مبتنی بر یادگیری تقویتی چندعامله فرض می‌کنند محیط ناشناخته بوده و مشاهدات  $T$  و  $R$  از طریق تجربی قابل مشاهده هستند. هدف این الگوریتم‌ها پیداکردن سیاست تعادل بوده و معمولاً نیازهای کمتری را درباره رفتار عامل‌های دیگر دارند. یک چارچوب کلی برای یادگیری  $Q$  چندعامله در شکل ۲ نشان داده شده است. در قالب کلی، الگوریتم‌های مختلف یک ورودی تابع انتخاب تعادل را دریافت می‌کنند تا بتوانند با توجه به بردار ماتریس نظری  $V = (Q_1, \dots, Q_n)$  تابع ارزش  $V$  را محاسبه نمایند. Hu و Wellman از تابع با نام  $Q$ -Nash برای محاسبه مقدار  $V$  طبق (۵) استفاده نمودند [۸].

Meiping Song نیز  $Q$ -همکاران نیز از مکانیزم چانه‌زنی<sup>۱</sup> [۲۱] و  $Q$ -Pareto استفاده کردند [۱۱]

$$V_i(s) \in \text{Nash}_i(Q_1(s), \dots, Q_n(s)) \quad (5)$$

$$\text{Nash}_i(Q_i(s)) = \pi^i(s) \dots \pi^n(s) Q_i(s)$$

$$V_i(s) \in \text{NBS}_i(Q_1(s), \dots, Q_n(s)) \quad (6)$$

$$\text{NBS}_i(Q_i(s)) = \max_{\vec{a}} (Q_i(s, \vec{a}) \times \dots, Q_n(s, \vec{a}))$$

1. Nash Bargaining

	T	L
T	۱،-۱	-۱،۱
L	-۱،۱	۱،-۱

	L	R
L	۶	۲۷
R	۲۷	۰۰

شکل ۱: دو نوع بازی ماتریسی به نام‌های تطابق‌سکه‌ها (سمت چپ) و بازی Chicken (سمت راست) [۲۱]

$$V(\pi^*, s) \equiv \max_a \{r(s, a) + \gamma \sum_{s'} P(s'|s, a) V(s', \pi^*)\} \quad (2)$$

$r(s, a)$  پاداش به‌دست آمده از انجام عمل  $a$  در وضعیت  $s$  است،  $s'$  وضعیت جدید و  $P(s'|s, a)$  احتمال رفتن به  $s'$  پس از انجام عمل  $a$  در وضعیت  $s$  است. این روش هنگامی قابل استفاده است که عامل، توابع پاداش و گذار وضعیت را بشناسد، در غیر این صورت نمی‌توان این روش را به کار گرفت. در چنین حالی از الگوریتمی به نام یادگیری  $Q$  استفاده می‌شود [۱۷].

### ۲-۲ بازی‌های تصادفی

بازی‌های تصادفی تعیین فرآیندهای تصادفی مارکوف به حالت چندعامله بوده و می‌توانند به عنوان چارچوبی مناسب برای بررسی و تحقیقات یادگیری چندعامله استفاده شوند. این بازی‌ها توسعه‌ای از بازی‌های ماتریسی با چندین حالت بوده و به آنها بازی‌های مارکوفی نیز گویند. دو مثال از بازی‌های ماتریسی در شکل ۱ آورده شده‌اند. در هر بازی دو بازیکن وجود دارند که به صورت سط्रی و ستونی اعمالشان نشان داده شده است. درایه‌های ماتریس همان پاداش بازیکنان است. در بازی اول که یک بازی با جمع صفر است پاداش بازیکن دوم منفی پاداش اولی است. در حالت کلی ماتریس پاداش برای بازی‌های دونفره با دو ماتریس نشان داده می‌شود. هر حالت در بازی اتفاقی می‌تواند به صورت یک بازی ماتریسی بیان شود. در هر حالت از بازی بازیکنان پس از بازی و دریافت پاداش با توجه به عمل گروهی انجامشده به حالت دیگری از بازی می‌روند. در یک بازی تصادفی، اعمال بازیکنان همزمان صورت می‌گیرد.

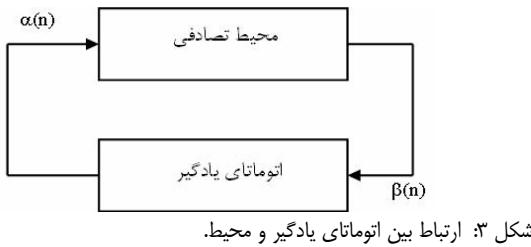
به‌طور کلی تعریف رسمی بازی‌های تصادفی به صورت زیر است [۱۸]:

**تعریف ۲:** یک بازی تصادفی به‌فرم چندتایی  $\langle n, S, A_1, \dots, A_n, T, R \rangle$  بیان می‌شود که  $n$  تعداد عامل‌ها،  $S$  مجموعه حالات،  $A_i$  مجموعه اعمال هر عامل  $i$  (در فضای اعمال گروهی  $\prod_{i=1}^n A_i$ )،  $T$  تابع انتقال  $[0, 1] \rightarrow [0, 1]^n$  و  $R$  تابع پاداش برای عامل  $i$  است.

در یک بازی تصادفی کاهش‌پذیر، هدف یک بازیکن بیشینه کردن مجموع پاداش کاهش‌یافته با پارامتر کاهش  $\gamma \in [0, 1]$  است. اگر  $\pi_i$  استراتژی بازیکن  $i$  باشد به‌ازای یک وضعیت اولیه  $s$ ، بازیکن  $i$  سعی در بیشینه‌نمودن رابطه زیر دارد

$$V(s, \pi^1, \pi^2, \dots, \pi^n) \equiv \sum_{t=1}^{\infty} \gamma^t E(r_t | \pi^1, \pi^2, \dots, \pi^n, s_t = s) \quad (3)$$

با نگاه به (۳) و (۱) تشابه با مدل MDP دیده می‌شود متنهای در اینجا چندین عامل وجود دارند و رفتن به وضعیت بعدی و دریافت پاداش به عمل گروهی عامل‌ها بستگی دارد. در حالت یادگیری تقویتی چندعامله، بیشینه‌نمودن سودمندی (پاداش) مورد انتظار هر عامل در بازی‌های مجموع کلی ممکن نبوده و هدف پیداکردن سیاست تعادل در بازی‌های مارکوف است. از جمله این سیاست‌ها می‌توان سیاست تعادل نش را نام برد. به عبارت دیگر پیداکردن یک سیاست متعادل به عنوان یک راه حل برای بازی‌های اتفاقی محسوب می‌شود.



اتوماتای یادگیر با ساختار متغیر را توسط چهارتایی  $\{\alpha, \beta, p, T\}$  داریم که  $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  مجموعه عملها (اقدامات)،  $\beta = \{\beta_1, \beta_2, \dots, \beta_m\}$  مجموعه ورودی‌های اتوماتا،  $p = \{p_1, \dots, p_r\}$  بردار احتمال انتخاب هر یک از عملها و  $T[\alpha(n), \beta(n), p(n)] = p(n+1)$  الگوریتم زیر بر اساس (۷) و (۸) یک نمونه از الگوریتم‌های یادگیری خطی است. فرض می‌کنیم عمل  $a_i$  در مرحله  $n$  انتخاب شود.

- پاسخ مطلوب از محیط

$$\begin{aligned} p_i(n+1) &= p_i(n) + a(1-p_i(n)) \\ p_j(n+1) &= (1-a)p_j(n) \quad \forall j, j \neq i \end{aligned} \quad (7)$$

- پاسخ نامطلوب از محیط

$$\begin{aligned} p_i(n+1) &= (1-b)p_i(n) \\ p_j(n+1) &= \frac{b}{r-1} + (1-b)p_j(n) \quad \forall j, j \neq i \end{aligned} \quad (8)$$

در (۷) و (۸)  $a$  پارامتر پاداش و  $b$  پارامتر جریمه می‌باشد. با توجه به مقادیر مختلف برای  $a$  و  $b$  سه حالت را می‌توان در نظر گرفت: اگر  $a$  و  $b$  با هم برابر باشند، الگوریتم را  $L_{RP}$ ، هنگامی که  $b$  از  $a$  خیلی کوچک‌تر باشد، الگوریتم را  $L_{REP}$  و اگر  $b$  مساوی صفر باشد آن را  $L_{RI}$  می‌نامیم. شما  $S - L_{RP}$  برای مدل‌های  $Q$  و  $S$  در (۹) بیان شده است. در مدل  $S$ ، اگر عمل  $a_i$  در مرحله  $n$  انتخاب شود در این صورت داریم

$$\begin{aligned} p_i(n+1) &= p_i(n) + a(1-\beta_i(n))(1-p_i(n)) \\ &\quad - b\beta_i(n)p_i(n) \\ p_j(n+1) &= p_j(n) - a(1-\beta_i(n))p_j(n) \\ &\quad + b\beta_i(n)\left(\frac{1}{r-1} - p_j(n)\right) \quad \forall j, j \neq i \end{aligned} \quad (9)$$

که در آن  $r$  تعداد اعمال ممکن،  $a$  و  $b$  همان پارامترهای پاداش و جریمه هستند. برای اطلاعات بیشتر درباره اتوماتاهای یادگیر می‌توان به [۲۳] مراجعه نمود.

### ۲-۳ بازی اتوماتا

در بازی اتوماتاها [۲۲] و [۲۴] تعدادی اتوماتا بدون آن که دانش کاملی از یکدیگر داشته باشند در محیط عمل می‌کنند. از نقطه نظر بازی‌ها بهتر است اعمال اتوماتا را به عنوان استراتژی آنها و ورویدشان را به عنوان نتایج در نظر بگیریم. با انجام نتایج هر اتوماتا یک ورودی را از محیط دریافت می‌کند. به دلیل طبیعت تصادفی بودن محیط‌ها، انتخاب اعمال توسط هر اتوماتا در یک بازی تعیین‌کننده احتمال نتایجشان است. همچنین فرض شده اتوماتای شرکت‌کننده در بازی، احتمالات نتایج را به عنوان دانش اولیه در اختیار ندارد. علاوه بر این فقط اطلاعاتی که هر اتوماتا در جریان بازی از ورویدش به دست می‌آورد به عنوان تابعی از اعمالش در بازی‌های موفق محسوب می‌شود. بنابراین در بحث تئوری بازی‌ها، در بازی‌های اتوماتا بازیکنان اطلاعات اولیه در ارتباط با بازی نظیر تعداد بازیکنان، اعمال احتمالی ایشان و عنصر ماتریس‌های بازی را در اختیار ندارند.

### Multi-agent $Q$ -learning (Stochastic Game, $\alpha, \gamma, M$ )

**Inputs:** discount factor  $\gamma$ , learning rate  $\alpha$ ,  $M$  total training time

**Output:** state-value functions, action-value  $V_i^*$  function  $Q_i^*$

**Initialize:**  $s, a_1 \dots a_N, Q_1 \dots Q_N$

1. for  $k = 1$  to  $M$  do
2. simulate actions  $a_1, \dots, a_N$  in state  $s$
3. observe rewards  $R_1, \dots, R_N$  and next state  $s'$
4. for  $i = 1$  to  $N$  do
  - (a) compute  $V_i(s')$
  - (b)  $Q_i(s, a_i) \leftarrow (1-\alpha)Q_i(s, a_i) + \alpha[(1-\gamma)R_i + \gamma V_i(s')]$
5. agents choose action  $a'_1, \dots, a'_N$
6.  $s = s', a_1 = a'_1, \dots, a_N = a'_N$
7. decay  $\alpha$

شکل ۲: الگوریتم یادگیری  $Q$  چندعامله.

با توجه به پیچیدگی بالای محاسبات در هر یک از الگوریتم‌های محاسبه تعادل نش در بازی‌های اتفاقی، به نظر می‌رسد استفاده از الگوریتم‌های مبتنی بر اتوماتاهای یادگیر با توجه به سادگی و پیچیدگی محاسباتی کم بتوانند کارایی مطلوبی را از خود ارائه دهند. با توجه به این موضوع در بخش بعدی مدلی مبتنی بر اتوماتاهای یادگیر برای حل بازی‌های مارکوفی پیشنهاد و نتایج آن ارائه گردیده‌اند.

### ۳-۱ اتوماتاهای یادگیر و مدل پیشنهادی

با توجه به این که مدل‌های اتوماتاهای یادگیر ارتباطی نزدیک با یادگیری تقویتی چندعامله دارند، در این بخش اتوماتاهای یادگیر به عنوان مدلی از یادگیری تقویتی به اختصار شرح داده می‌شود.

### ۳-۲ اتوماتاهای یادگیر

اتوماتای یادگیر، ماشینی است که می‌تواند تعدادی متناهی عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی شده 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 [۲۲]. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداقل برساند. کارکرد اتوماتای یادگیر در تعامل با محیط، در شکل ۳ مشاهده می‌شود.

محیط را می‌توان توسط سه‌تایی  $E = \langle \alpha, \beta, c \rangle$  نشان داد که در آن  $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  مجموعه ورودی‌ها،  $\beta = \{\beta_1, \beta_2, \dots, \beta_m\}$  مجموعه خروجی‌ها و  $c = \{c_1, c_2, \dots, c_r\}$  مجموعه احتمال‌های جریمه می‌باشد. هرگاه  $\beta$  مجموعه‌ای دواعضوی باشد، محیط از نوع  $P$  است. در چنین محیطی  $\beta_1 = \beta_2 = \dots = \beta_m$  به عنوان پاداش در نظر گرفته می‌شود. در محیط از نوع  $Q$ ،  $\beta$  می‌تواند به طور گسته‌یک مقدار از مقادیر محدود در فاصله  $[0, 1]$  را اختیار کند و در محیط از نوع  $S$ ،  $\beta(n)$  مقداری تصادفی در فاصله  $[0, 1]$  است.  $c_i$  احتمال این که عمل  $i$  نتیجه نامطلوب داشته باشد است. در محیط ایستا، مقادیر  $c_i$  بدون تغییر می‌مانند، حال آن که در محیط غیر ایستا این مقادیر در طی زمان تغییر می‌کنند. اتوماتاهای یادگیر می‌توانند به دو دسته اتوماتای یادگیر با ساختار ثابت اتوماتای یادگیر با ساختار متغیر (VSLA) دسته‌بندی شود [۲۲].

#### ۴-۳ معرفی مدل پیشنهادی

در این بخش روش پیشنهادی حل بازی‌های تصادفی با استفاده از اتوماتای یادگیر ارائه می‌شود که آن را  $MLA^s$  می‌نامیم. می‌دانیم در یک بازی تصادفی عمل انتخاب شده در هر حالت با توجه به نتیجه اعمال گروهی عامل‌های مستقل در سیستم است. در مدل پیشنهادی، فرض می‌گردد که محیط شامل مجموعه‌ای از حالات گستته است. به ازای هر حالت  $S$  در محیط بازی و به ازای هر عامل نظیر  $K$ ، یک اتوماتای یادگیر با ساختار متغیر و مدل  $S$  نظیر  $LA_k^s$  قرار داده می‌شود. حالتهای مجاور با هر حالت، همسایگان آن در نظر گرفته می‌شود. با توجه به تعداد همسایگان هر اتوماتا، تعداد اعمال اتوماتا در هر حالت در نظر گرفته می‌شود. هر عمل گروهی ناشی از اعمال اتوماتاهای هر حالت متناظر با انتقال به یکی از حالات همسایه است. هر عامل برای تعیین حالت بعدی خود از اتوماتای یادگیر با توجه به عمل انتخابی عامل‌های محیط کمک می‌گیرد. عامل‌های یادگیرنده از یک حالت شروع کننده آغاز و به حالت پایانی یا هدف می‌رسند. در ابتدا اتوماتاهای یادگیر تمام عمل‌های خود را با احتمالی یکسان متناسب با تعداد اعمال آنها، انتخاب می‌کنند. در صورتی که عمل اتوماتها منجر به ورود عامل به حالت نهایی (هدف) شود اتوماتای یادگیر پاداش کامل می‌گیرد، در غیر این صورت از آنتروپی بردار احتمال اتوماتای یادگیر حالت بعد برای تعیین پاداش یا جریمه میانی استفاده می‌شود.

در روش پیشنهادی، آنتروپی بردار احتمال میزان عدم قطعیت اتوماتای یادگیر حالت بعد را در انتخاب عمل خود نشان می‌دهد. هرچه 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 نبوده و عمل‌های خود را به صورت تصادفی انتخاب می‌کند (جستجو). ولی چنانچه عدم قطعیت کم باشد به این معنی است که اتوماتا با احتمال بالایی یکی از اعمال خود را انتخاب می‌کند و دارای اطلاعات مفیدی برای رسیدن به هدف می‌باشد و از این اطلاعات بهره‌برداری می‌نماید. فرض کنید بردار احتمال اعمال اتوماتای یادگیر  $i$  با عمل در حالت  $s$  به صورت  $\{P_i^s(1), P_i^s(2), \dots, P_i^s(r)\}$  باشد.

آنتروپی این بردار احتمال برای حالت  $s$  به شکل زیر تعیین می‌شود

$$H_i^s = \sum_{j=1}^r p_i^s(j) \log(p_i^s(j)) \quad (13)$$

آنtronوپی زمانی بیشترین مقدار را خواهد داشت که تمام اعمال اتوماتای یادگیر احتمالی یکسان باشند.  $P_{equal} = \{p_1 = \dots = p_r = 1/r\}$  زمانی کمترین مقدار (برابر با  $0$ ) را خواهد داشت که بردار احتمالات یک بردار یکه باشد. برای این که مقدار آنتروپی را به مقداری بین  $0$  و  $1$  تبدیل کنیم تا به عنوان بردار تقویتی در اتوماتای ساختار متغیر مدل  $S$  قابل استفاده باشد از  $(14)$  استفاده می‌شود. فرض کنید عامل  $i$  در حالت  $s$  و اتوماتای یادگیر متناظرش یعنی  $LA_i^s$  به حالت  $'s$  می‌رود در این صورت بردار تقویتی اتوماتای یادگیر  $\beta_i^s$  به صورت زیر محاسبه می‌شود

$$\beta_i^s = \frac{H_i^s}{\max(H_i^{s'})} \quad (14)$$

در بازی‌های با مجموع صفر الگوریتم  $L_{RI}$  به نقطه تعادل همگرا می‌شود اگر در استراتژی‌های مخصوص باشد در حالی که در الگوریتم  $L_{REP}$  به استراتژی‌های تعادل مخلوط همگراشی دارد. در  $[15]$  نشان داده شده است در بازی‌های با مجموع غیر صفر وقتی که اتوماتاهای یادگیر از الگوریتم  $L_{RI}$  استفاده می‌نمایند و بازی دارای یک نقطه تعادل مخصوص باشد، همگراشی تضمین شده است.

#### ۳-۳ مفهوم آنتروپی در یادگیری

آنtronوپی مفهومی اساسی در ترمودینامیک است که بیانگر درجه بی‌نظمی در یک سیستم ترمودینامیکی است. این مفهوم توسط شانون در حوزه تئوری اطلاعات تحت عنوان آنتروپی اطلاعاتی<sup>۱</sup> مطرح گردید. آنتروپی اطلاعاتی حالتی از آنتروپی متغیرهای تصادفی است که درجه عدم قطعیت را بیان می‌کند و به صورت زیر تعریف می‌شود  $[25]$  و  $[26]$

$$H(X) = - \sum_{x_i \in \Theta} P(x_i) \log(P(x_i)) \quad (10)$$

$X$  متغیر تصادفی با مجموعه مقادیر  $\Theta$  و تابع تجمعی احتمال  $P(x_i)$  است. آنتروپی متغیر تصادفی بسته به کاربردهای مختلف، معانی مختلف وابسته به مسأله دارد. در این مقاله آنتروپی در حوزه یادگیری در فضای گستته به عنوان معیاری جدید در فرآیند یادگیری معرفی می‌شود. در یک فضای حالات گستته، هر حالت  $s$  با در برگیرنده مجموعه‌ای از مقادیر احتمالات تصمیم است. هر حالت  $s$  متناظر با متغیر تصادفی  $D_i$  است. در فرآیند یادگیری آنتروپی  $D_i$  بیانگر عدم قطعیت تصمیم در حالت  $s$  است. مقادیر بیشتر آنتروپی عدم قطعیت بیشتر تصمیم‌گیرنده را مشخص می‌کند.

برای بررسی رفتار یادگیرنده، دو پارامتر آنتروپی استراتژی محلی و آنتروپی کلی را به عنوان پارامتر مستقل از مسأله به صورت زیر تعریف می‌کنیم: آنتروپی استراتژی محلی بیانگر عدم قطعیت تصمیم در یک حالت خاص بوده و به صورت زیر تعریف می‌شود

$$H_{local}^{s_i} = - \sum_{j=1}^N p_j(s_i) \log(p_j(s_i)) \quad (11)$$

که در آن  $N$  تعداد عناصر در مجموعه اعمال و  $P_j(s_i)$  احتمال انتخاب  $j$  امین عمل تحت حالت  $s_i$  است. آنتروپی کلی محیط را میانگین تمام آنتروپی‌های استراتژی‌های محلی به صورت زیر تعریف می‌کنیم

$$H_{global}^{s_i} = - \frac{1}{M} \sum_{i=1}^M H_{local}^{s_i} = - \frac{1}{M} \sum_{i=1}^M \sum_{j=1}^N p_j(s_i) \log(p_j(s_i)) \quad (12)$$

که در آن  $M$  تعداد عناصر در مجموعه حالات محیط می‌باشد. این پارامتر یک معیار مستقل از مسأله برای فرآیند یادگیری است به طوری که درجه همگراشی استراتژی را بیان می‌کند. برای بررسی رفتار الگوریتم این معیار در هر مرحله نمایش داده می‌شود. در یادگیری، بهینه‌سازی استراتژی فرآیندی است که احتمال مقادیر انتخاب عمل بهینه را افزایش می‌دهد به این معنا که استراتژی تصادفی اولیه به استراتژی بهینه تعریف شده همگرا می‌گردد. در نتیجه در فرآیند یادگیری به مرور عدم قطعیت استراتژی یادگیری و بالطبع آن آنتروپی کاهش می‌یابند. لذا مفهوم آنتروپی درجه همگراشی را در فرآیند یادگیری بیان می‌نماید.

2. Multi Learning Automata
3. Exploration
4. Exploitation

1. Information Entropy

**Modified MLA (Stochastic Game SG,  $a, b, M$ )**

**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. Coefficient  $c$  is an increment parameter.

The following steps are taken

1.  $K$  is set to initial value
2. for episode = 1 to  $M$  do
3.      $s$  = Start State
4.     while not done
5.         for each agent  $i$  do concurrently
6.             Activate  $LA_i^s$  ( $LA$  residing in current state( $s$ ) of agent  $i$ )
7.             Choose action  $a_i^s$  in state  $s$
8.             Observe Rewards  $r_i^s$  and Next State  $s'$
9.             Compute cumulative-reward1, cumulative-reward2
10.            Compute  $\beta_i^s$  signal base on EQ (15)
11.            Train  $LA_i^s$  residing in state  $s$  according  $\beta_i^s$
12.         end for
13.          $s = s'$
14.     end while
15.     Increment  $K$  with coefficient  $c$
16. end for

شکل ۵: الگوریتم MLA تغییریافته.

#### ۴- آزمایش‌های انجام‌گرفته در محیط بازی‌های GRID GAME و نتایج آنها

چارچوب عمومی به کار رفته در سیستم‌های چندعامله شامل بازی‌های تصادفی و ماتریسی است. در یادگیری چندعامله، یکی از انواع بازی‌های تصادفی غیر رقابتی برای آزمایش، بازی‌های Grid Game است که توسط Hu و Wellman ارائه شده و بسیار مورد استفاده قرار گرفته است [۸]. این بازی دارای انواع متفاوتی است که همه آنها یک بازی دونفری از نوع جمع کلی می‌باشند.

#### ۴-۱ دو نوع بازی Grid Game

دو نوع بازی Grid Game در شکل ۶ نشان داده شده‌اند. بازی نوع اول (GG1) نسخه‌ای از بازی Chicken است که دارای چندین حالت بوده و در آن فقط یک هدف وجود دارد. بازی دوم (GG2) یک بازی همانگی چندحالته‌ی است که دو عامل و دو نوع هدف وجود دارند. در بازی ۱، دو عامل از دو گوشی یک صفحه شروع کرده و سعی می‌کنند با کمترین تعداد حرکت به هدف برسند. اعمال بازیکنان به طور همزمان انجام می‌گیرند. اعمال بازیکنان یعنی A1 و A2 به صورت چهار عمل (بالا، پایین، چپ، راست) تعریف می‌شود. مجموعه فضای حالات به صورت  $S = \{s / s = (l_1, l_2) \text{ تعریف می‌شود که هر حالت } l = (l_1, l_2) \in S\}$  مختصات عامل‌های ۱ و ۲ را نشان می‌دهد. عامل‌ها نمی‌توانند در یک مختصات یکسان قرار گیرند. اگر دو عامل سعی در حرکت به یک مربع یکسان داشته باشند حرکت هر دو با شکست مواجه می‌شود. تعداد حالات ممکن در نوع اول برابر  $8 \times 7 = 56$  و نوع دوم برابر  $7 \times 7 = 49$  است. اگر عامل‌ها به دو خانه مختلف غیر هدف بروند هر دو پاداش صفر را دریافت می‌کنند. اگر هر یک از عامل‌ها به هدف برسد واحد پاداش می‌گیرد و در صورت رفتن به خانه یکسان (تدالو) هر دو یک واحد جریمه می‌شوند و در موقعیت قبلی باقی می‌مانند. در هر دو بازی گذار از

**Multi Leaning Automata (Stochastic Game SG,  $a, b, K, M$ )**

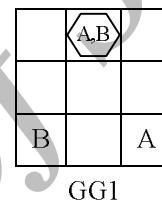
**Inputs:**  $a, b$ : reward and penalty parameter for each LA,  
 $K$ : exploration Parameter,  $M$ : total training time

**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.

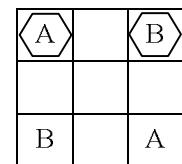
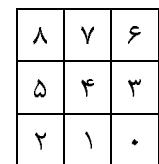
The following steps are taken

1. for episode = 1 to  $M$  do
2.      $s$  = Start State
3.     while not done
4.         for each agent  $i$  do concurrently
5.             Activate  $LA_i^s$  ( $LA$  residing in current state( $s$ ) of agent  $i$ )
6.             Choose action  $a_i^s$  in state  $s$
7.             Observe Rewards  $r_i^s$  and Next State  $s'$
8.             Compute cumulative-reward1, cumulative-reward2
9.             Compute  $\beta_i^s$  signal base on EQ (15)
10.            Train  $LA_i^s$  residing in state  $s$  according  $\beta_i^s$
11.         end for
12.          $s = s'$
13.     end while
14. end for

شکل ۶: الگوریتم MLA.



GG1

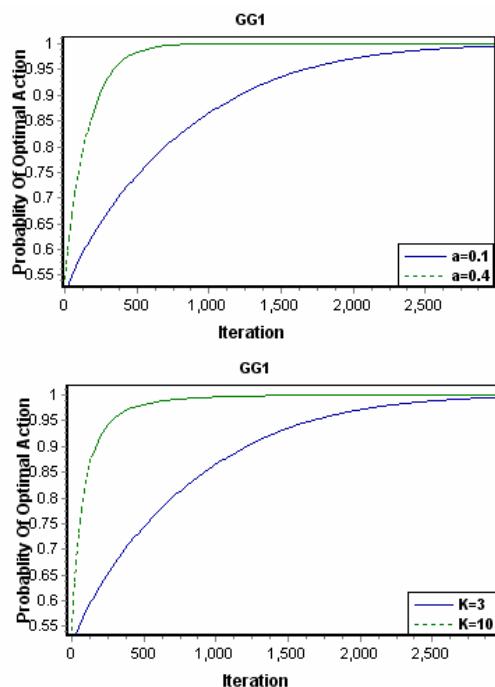


شکل ۶ نمایش مختصات بازی دو نوع بازی Grid World.

که در آن  $\max(H_i^s)$  بیشترین مقدار آنتربوی برای عامل  $i$  در حالت  $s$  است و  $K$  پارامتر روش می‌باشد. میزان این پارامتر بر حسب کاربرد و نیاز مسئله باید به دقت تعیین شود. مقادیر بالای این پارامتر باعث می‌شود که  $\beta_i^s$  ها مقادیر کمی داشته باشند و اتوماتاهای یادگیر بیش از حد پاداش بینند. این به معنی جستجوی بیشتر در محیط است. هرچه میزان  $K$  کمتر باشد  $\beta_i^s$  ها مقادیر بیشتری خواهند داشت. این امر باعث می‌شود که اتوماتاهای بیشتر جریمه شوند. این مسئله باعث می‌شود که حتی حالتهای مطلوب نیز پاداش لازم را نگیرند. فرض کنید که عامل  $i$  در حالت  $s$  باشد و اتوماتای یادگیر آن یعنی عامل را به حالت  $s'$  هدایت کند. در این صورت سیگنال تقویتی با توجه به (۱۵) تعیین می‌شود. الگوریتم پیشنهادی در شکل ۷ نشان داده شده است

$$\beta_i^s = \begin{cases} \cdot & \text{if } s' = \text{EndState} \\ \frac{H_i^s}{\max(H_i^{s'})^K} & \text{otherwise} \end{cases} \quad (15)$$

تنظیم پارامتر  $K$ : در روش پیشنهادی مقدار پارامتر  $K$  ثابت در نظر گرفته شده بود. در روش بهبودیافته که مشابه با روش بولتزمن در یادگیری Q عمل می‌کند پارامتر  $K$  برای تنظیم حالت جستجو یا استناد بر تجربیات قبلی استفاده شده است. با تنظیم اولیه پارامتر  $K$  در ابتدا عامل‌ها جستجو را انجام داده و به مرور زمان با توجه به تغییر پارامتر امکان بهره‌برداری از دانسته‌های خود را پیدا می‌کنند. الگوریتم بهبودیافته در شکل ۸ نشان داده شده است. یادگیری موقعی پایان می‌باید که عامل به هدف برسد یا تعداد مراحل پایان می‌باید.



۸: نمودار تغییرات احتمال انتخاب عمل بهینه برای اتماماتای یادگیر ۱ در خانه شروع با توجه به مقادیر مختلف پارامتر یادگیری و مقادیر متفاوت  $K$ .

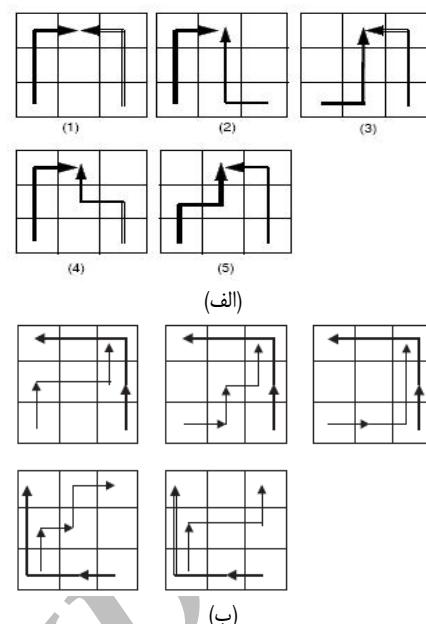
و با توجه به (۱۵) مقدار  $\beta$  (پارامتر پاداش یا جریمه) تعیین شده و توماتای آن حالت به روز می‌گردد. عامل‌ها اعمالشان را هم‌زمان انجام داده و هر دو قابلیت مشاهده حالت جدید، پاداش آنی به دست آمده و عمل انجام‌شده توسط دیگری را دارند. شکل ۷ راه حل‌های بھینه را برای دو بازی نشان می‌دهد.

## ۴- نتایج آزمایش‌ها و تحلیل آنها

برای بررسی رفتار الگوریتم پیشنهادی و ارزیابی آن چند سری آزمایش نجاح گرفته است. بازی مارکوفی در نظر گرفته شده همان دو نوع بازی Grid Game مطرح شده در بخش قبلی است.

ازمایش ۱: بررسی همگرایی اتوماتاتی یادگیری. این آزمایش برای بررسی همگرایی اتوماتاتی انجام گرفته است. برای این منظور تغییرات احتمال انتخاب عمل حرکت بهینه در خانه شووع برای عامل ۱ در دو حالت ثابت گرفتن پارامتر  $K$  و ثابت نگه داشتن پارامتر یادگیری مشاهده می‌شود. در حالت اول پارامترهای یادگیری  $a$  برابر  $\{0, 1, 0, 4\}$  و پارامتر  $b$  برابر صفر و با فرض  $(K = 3)$  ثابت) و حالت دوم پارامتر  $a$  ثابت و پارامتر  $k = \{10, 3\}$  بررسی شده است. شکل ۸ نمودار تغییرات را برای این عامل نشان می‌دهد. با توجه به نمودار می‌بینیم هرچه ضربی پارامتر یادگیری افزایش یابد احتمال انجام عمل نیز افزایش یافته و در نتیجه همگرایی نیز سریع‌تر خواهد شد و از طرفی هرچه پارامتر جستجو بیشتر شود سعت همگاری بیشتر می‌شود.

ازمايش ۲: مقايسه کاريي الگوريتم پيشنهادي با ساير الگوريتم‌ها.  
برای مقايسه الگوريتم با الگوريتم‌های مشابه از الگوريتم Q - Nash که يكی از مهم‌ترین و عمومی‌ترین الگوريتم‌ها در اين زمينه می‌باشد، استفاده شده است. در بخش اول ابتدا مقايسه‌ای بين الگوريتم پيشنهادي و ساير الگوريتم‌ها از نظر درصد رسیدن به مسیر بهينه انجام می‌گيرد. جدول ۱ مقايسه نتایج به دست آمده را از نظر ميزان همگرائي نشان می‌دهد.  
ازمايش‌های جدول ۱، ۴۰۰۰۰ مرحله و برای استحکام بیشتر ۵۰ مرتبه جرا در نظر گرفته شده است.



شکل ۷: نمونه راه حل های بهینه برای دو بازی Grid World، (الف) GG۱ و (ب) GG۲.

جدول ۱: مقایسه کارایی الگوریتم‌های یادگیری.

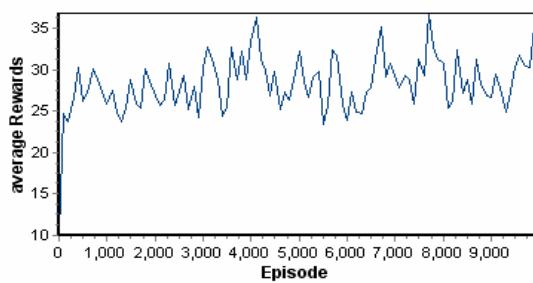
استراتژی یادگیری	نتیجه یادگیری
درصد رسیدن به مسیر بهینه	عامل ۱
۲۳	Single Q -learning
۴۵	Single Q -learning
۵۴	Single Q -learning
۱۰۰	First Nash
۱۰۰	First Pareto
۱۰۰	MLA

حالتی به حالت دیگر با قطعیت انجام می‌شود، یعنی حالت جاری و عمل مشترک عامل‌ها منحصرًا حالت بعدی را تعیین می‌کنند.

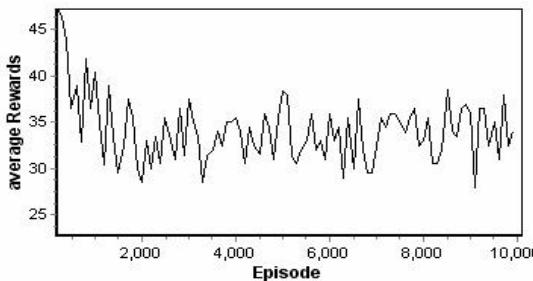
در این بازی فرض می‌شود عامل‌ها از موقعیت هدف در ابتدای بازی و همچنین از پاداش کلی یکدیگر اطلاع ندارند. عامل‌ها اعمالشان را هم‌زمان انتخاب نموده و فقط می‌توانند از اعمال قبلی عامل‌های دیگر و حالت فعلی (موقعیت مشترک هر دو عامل) آگاهی داشته باشند.

یک مسیر دنباله‌ای از اعمال از نقطه شروع تا پایان را نشان می‌دهد. در اصطلاح بازی چنین مسیری را استراتژی یا سیاست می‌نامند. کوتاه‌ترین مسیری که با مسیر عامل دیگر تداخل داشته باشد را استراتژی بهینه می‌نامند که یک موازنہ نش را می‌سازند زیرا هر مسیر (استراتژی) بهترین پاسخ در مقابل دیگری است. استراتژی‌های ایستا به صورت  $\pi^i$  (برای هر حالت  $s_i$ ) تعریف می‌شود که برای  $(s_m, \dots, s_1)$  توزیع احتمالاتی را اختصاص می‌دهد.  $\pi(s_i)$  را استراتژی محض گویند هرگاه احتمال ۱ را به اعمال بدهد. یک موازنہ نش شامل زوج استراتژی‌های ایستا به صورت  $(\pi_*, \pi^*)$  است که هر استراتژی بهترین پاسخ به دیگری، باشد.

برای حل مسأله با توجه به روش ارائه شده بهازای هر حالت (l<sub>1</sub>, l<sub>2</sub>) برای هر عامل یک اتوماتی یادگیر در نظر گرفته شده است. هر عامل با توجه به عمل مشترک گروهی از یک حالت به حالت جدید می‌رود (در صورت عدم برخورد دو عامل با یکدیگر و با توجه به عدم برخورد با موانع

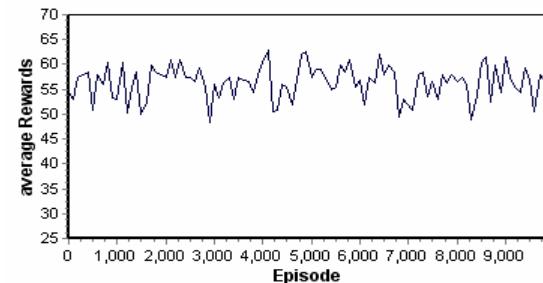


(الف)

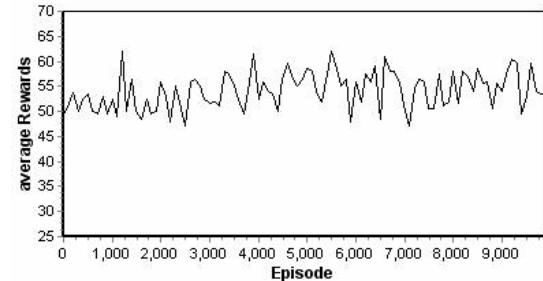


(ب)

شکل ۱۰: مقایسه روش پیشنهادی با الگوریتم Q-Nash از نظر میانگین پاداش دریافتی برای عامل ۱ در بازی ۲، (الف) GG1 و (ب) الگوریتم پیشنهادی.

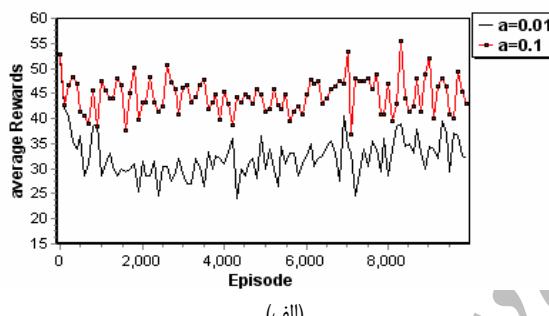


(الف)

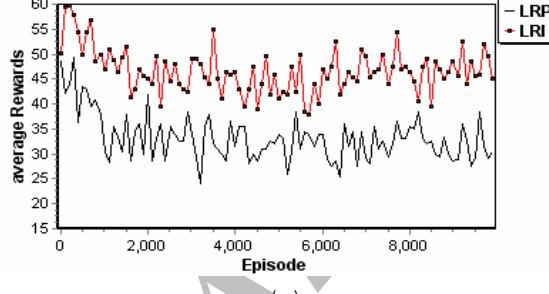


(ب)

شکل ۹: مقایسه روش پیشنهادی با الگوریتم Q-Nash از نظر میانگین پاداش دریافتی برای عامل ۱ در بازی ۲، (الف) GG1 و (ب) الگوریتم پیشنهادی.

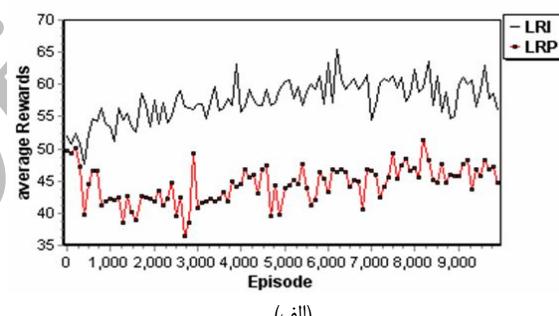


(الف)

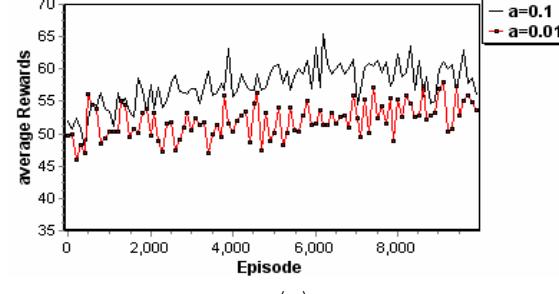


(ب)

شکل ۱۱: بررسی رفتار روش پیشنهادی با توجه به (الف) الگوریتم یادگیری و (ب) پارامتر یادگیری برای بازی ۲. GG2



(الف)



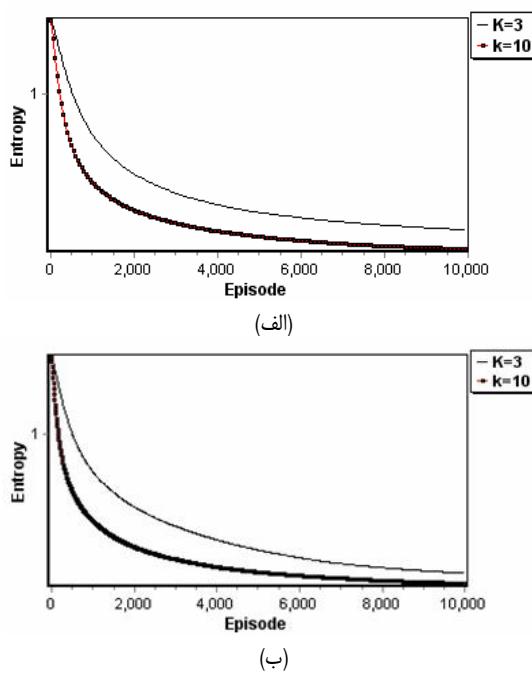
(ب)

شکل ۱۲: بررسی رفتار روش پیشنهادی با توجه به (الف) الگوریتم یادگیری و (ب) پارامتر یادگیری برای بازی ۱. GG1

پیشنهادی. در این آزمایش نقش پارامترهای یادگیری  $a$  و  $b$  در همگرایی و کارایی الگوریتم پیشنهادی در دو نوع بازی مورد بررسی قرار می‌گیرد. با در نظر گرفتن مقدار  $\{a = 0.1, 0.01\}$  و  $b = 0$  می‌بینیم با افزایش پارامتر  $a$  شبیه رسیدن به نقطه تعادل بالا رفته و در فاصله زمانی کمتری به پاداش مطلوب‌تر می‌رسد. در بخش دوم آزمایش به نقش پارامتر  $b$  در کارایی الگوریتم پرداخته می‌شود. با در نظر گرفتن مقدار  $\{a = 0.1, 0.01\}$  و  $b = 0$  می‌بینیم با افزایش  $L_{RI}$  و سپس الگوریتم  $L_{RP}$  از نظر میانگین پاداش برای عامل ۱ مشاهده می‌گردد. با توجه به نتایج به دست آمده دستاورد قضیه مطرح شده در [۱۵] الگوریتم  $L_{RI}$  در بازی‌های مارکوفی نیز تأیید می‌گردد و الگوریتم  $L_{RI}$  رفتار بهتری را از نظر همگرایی ارائه می‌دهد. شکل‌های ۱۱ و ۱۲ این مقایسه را نشان می‌دهند.

در بخش دوم این آزمایش، از میانگین پاداش به دست آمده در هر اپیزود برای مقایسه استفاده شده است. شکل‌های ۹ و ۱۰ میانگین پاداش را در الگوریتم Q-Nash و MLA برای هر دو نوع بازی نشان می‌دهند. آزمایش‌ها به خاطر استحکام در نتایج ۲۰۰ بار تکرار شده و در هر بار ۱۰۰۰ اپیزود آزمایش شده است. در هر اپیزود جدید هر عامل به طور تصادفی یک مختصات جدید (به جز هدف) را خواهد گرفت. پارامتر  $L_{RI}$  بوده و برای یادگیری برای روش پیشنهادی  $a = 0.1$  و الگوریتم  $L_{RI}$  نشان می‌دهد. برای روش Q-Nash پارامتر یادگیری همان  $a = 0.1$  در نظر گرفته شده است. همان‌طور که در شکل‌های ۹ و ۱۰ دیده می‌شود رفتارهای مشابهی از الگوریتم پیشنهادی و الگوریتم Q-Nash دیده می‌شود.

آزمایش ۳: بررسی نقش پارامتر یادگیری در کارایی الگوریتم



شکل ۱۴: تأثیر پارامتر  $K$  بر آنتروپی سراسری محیط برای الگوریتم MLA در دو بازی GG2 (الف) و (ب) GG1 (الف).

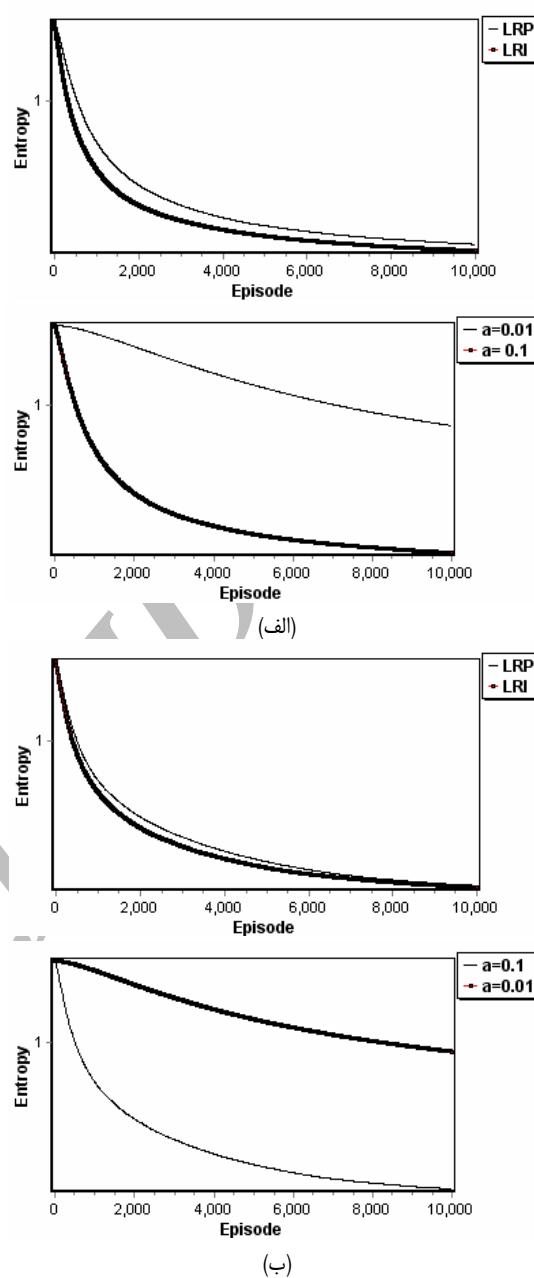
**آزمایش ۵:** بررسی نقش پارامتر  $K$  در آنتروپی کلی محیط. در این آزمایش به بررسی نقش پارامتر  $K$  در تغییرات آنتروپی کلی پرداخته می‌شود. برای این منظور پارامتر یادگیری  $\alpha = 0.1$  و پارامتر  $K = \{3, 10\}$  در نظر گرفته شده است. با توجه به شکل ۱۴ می‌بینیم افزایش پارامتر  $K$  باعث کاهش سریع تر آنتروپی کلی می‌گردد.

**آزمایش ۶:** در این آزمایش به بررسی الگوریتم اصلاح شده پرداخته می‌شود. برای این منظور برای بررسی رفتار بازی در هر اپیزود فرض می‌شود که عامل‌ها از همان نقطه شروع دوباره آغاز می‌کنند (تصادفی) و میانگین پاداش دریافتی در هر اپیزود ترسیم می‌شود. ابتدا مقدار  $K$  ثابت و مساوی مقدار ۱ نگه داشته شده و نقش پارامتر  $c$  مورد بررسی قرار می‌گیرد. آزمایش‌ها در دو نوع بازی با مقادیر  $\{c = \{0.001, 0.01, 0.1\}\}$  انجام گرفته‌اند. در حالت  $c = 0$  همان الگوریتم اولیه است. با توجه به نتایج بهدست آمده در شکل ۱۵ بهترین حالت موقعی است که مقدار  $c = 0.1$  داشته باشد.

همان‌طور که در شکل دیده می‌شود با مقدار دهی اولیه  $K$  برابر یک، ابتدا الگوریتم حالت جستجوگرانه داشته و با افزایش مقدار  $K$  به مرور زمان، امکان بهره‌برداری عامل از دانسته‌هایش میسر شده و کارایی آن بهبود می‌یابد.

## ۵- نتیجه‌گیری

در این مقاله مدلی مبتنی بر مفهوم آنتروپی و اتوماتاهای یادگیر برای حل بازی‌های تصادفی پیشنهاد گردید و سپس الگوریتمی مبتنی بر آن با در نظر گرفتن شرایط جستجو و استناد بر تجربیات عامل‌ها ارائه گردید. در روش پیشنهادی بهمازی هر عامل یک اتوماتای یادگیر در هر حالت از محیط قرار می‌گیرد که این یادگیرنده‌ها، عامل‌ها را برای رسیدن به اهداف خود هدایت می‌کنند. در روش پیشنهادی از مفهوم آنتروپی بردار احتمالات اتوماتای یادگیری برای پاداش دادن به عامل‌ها استفاده گردید. نتایج آزمایش‌های انجام‌گرفته نشان می‌دهند آنتروپی می‌تواند به عنوان یک عامل تسريع‌کننده در یادگیری به یادگیری بهتر و تطبیق‌پذیرتر کمک



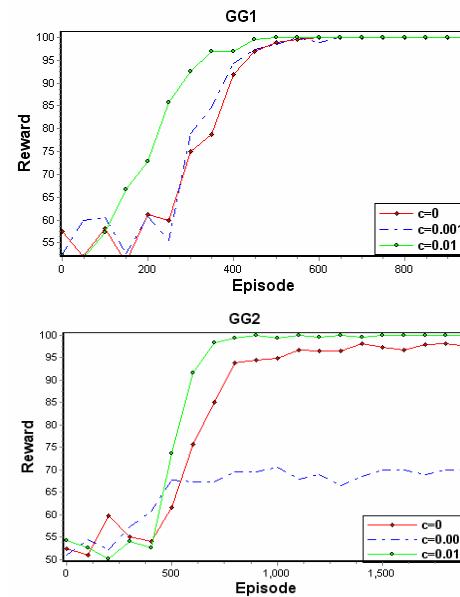
شکل ۱۳: تأثیر پارامترهای مختلف بر آنتروپی سراسری محیط برای دو بازی GG2 (الف) و (ب) GG1 (الف).

**آزمایش ۷:** بررسی رفتار الگوریتم با توجه به معیار آنتروپی کلی. در این آزمایش از معیار آنتروپی کلی سیستم به عنوان یک پارامتر کمی مستقل از مسئله در یاگیری استفاده شده که بیانگر درجه همگرای سیستم است. برای این منظور آنتروپی کلی محیط با توجه به پارامترهای مختلف یادگیری برای هر دو نوع بازی مورد بررسی قرار گرفته است. همان‌طور که در شکل ۱۳ دیده می‌شود میزان آنتروپی کلی سیستم در هر مرحله از یادگیری به مرور نشان داده شده است. با توجه به گذشت زمان یادگیری می‌بینیم رفتار سیستم به مرور از آنتروپی سراسری بالا که بیانگر بی‌نظمی سیستم است به سمت آنتروپی پایین تغییر می‌کند. در ابتدا که تمام احتمالات اعمال اتوماتاها با یکدیگر برابرند آنتروپی هر حالت دارای بیشترین مقدار بوده و لذا باعث می‌شود آنتروپی کلی محیط نیز در حد اکثر باشد. به مرور با توجه به یادگیری عامل‌ها می‌بینیم از میزان آنتروپی کلی محیط کاسته می‌شود. این تغییرات همان همگرای الگوریتم یادگیری را نشان می‌دهد. همان‌طور که در شکل دیده می‌شود رفتار الگوریتم در مورد آنتروپی کلی نیز در اینجا صدق می‌کند.

- [10] M. Song, G. Gu, and G. Zhang, "Pareto-Q learning algorithm for cooperative agents in general-sum games," in *Proc. CEMAS 2005*, pp. 576-578, Sep. 2005.
- [11] M. Song, J. Bai, and R. Chen, "A new learning algorithm for cooperative agents in general-sum games," in *Proc. of the Sixth Int. Conf. on Machine Learning and Cybernetics*, pp. 50-54, Hong Kong, Aug. 2007.
- [12] 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.
- [13] 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.
- [14] P. Vrancx, K. Verbeeck, and A. Nowe, "Decentralized learning in markov games," *IEEE Trans. on Systems, Man and Cybernetics- pt. B: Cybernetics*, vol. 38, no. 4, pp. 976-981, Aug., 2008.
- [15] P. S. Sastry, V. Phansalkar, and M. A. L. Thathachar, "Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 24, no. 5, pp. 769-777, May 1994.
- [16] R. A. Howard, *Dynamic Programming and Markov Processes*, MIT Press, 1960.
- [17] R. S. Sutton and A. G. Barto, *Reinforcement Learning I: an Introduction*, MIT Press, 1998.
- [18] F. Thuijsman, *Optimality and Equilibria in Stochastic Games*, CWI-tract 82, Centre for Mathematics and Computer Science, Amsterdam, 1992.
- [19] J. F. Nash Jr., "Non-cooperative games," *Annals of Mathematics*, vol. 54, no. 2, pp. 286-295, Sep. 1951.
- [20] A. M. Fink, "Equilibrium in a stochastic N-person game," *J. of Science in Hiroshima University, Series A - I*, vol. 28, pp. 89-93, 1964.
- [21] H. Qio, F. Szidarovszky, J. Rozenblit and L. Yong, "Multi-agent learning model with bargaining," in *Proc. of the 38th Conf. on Winter Simulation*, pp. 934-940, Dec. 2006.
- [22] K. S. Narendra and M. A. L Thathachar, *Learning Automata: an Introduction*, Prentice Hall, 1989.
- [23] M. A. L. Thathachar and P. S. Sastry, "Varieties of learning automata: an overview," *IEEE Trans. on Systems, Man, and Cybernetics - Pt B.: Cybernetics*, vol. 32, no. 6, pp. 711-722, Dec. 2002.
- [24] M. A. L. Thathachar and P. S. Sastry, *Networks of Learning Automata: Techniques for Online Stochastic Optimization*, Kluwer Academic Publishers, 2004.
- [25] M. Costa, A. L. Goldberger, and C. K. Peng, "Multi-scale entropy analysis of complex physiologic time series," *Physical Review Letters*, vol. 89, pp. 68101-68104, Aug. 2002.
- [26] Z. Dianhu, F. Shaohui, and D. Xiaojun, "Entropy - a measure of uncertainty of random variable," *Systems Engineering and Electronics*, vol. 11, pp. 1-3, Dec. 1997.

**بهروز معصومی** در سال ۱۳۷۴ مدرک کارشناسی مهندسی کامپیوتر-نرم افزار خود را از دانشگاه شهید بهشتی تهران و در سال ۱۳۷۷ مدرک کارشناسی ارشد مهندسی کامپیوتر خود را از دانشگاه آزاد اسلامی واحد تهران جنوب دریافت کرد. ایشان در سال ۱۳۸۹ موفق به اخذ درجه دکترا در مهندسی کامپیوتر-نرم افزار از دانشگاه آزاد اسلامی علوم و تحقیقات تهران گردید. نامبرده از سال ۱۳۷۷ تاکنون، عضو هیات علمی دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه آزاد اسلامی قزوین می باشد. زمینه های علمی مورد علاقه ایشان عبارتند از: سیستم های چند عامله، یادگیری در سیستم های چند عامله، طراحی سیستم های پایگاه داده و محاسبات نرم.

**محمد رضا میبدی** تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد اقتصاد بتریب در سالهای ۱۳۵۲ و ۱۳۵۶ از دانشگاه شهید بهشتی و در مقاطع کارشناسی ارشد و دکتری علوم کامپیوتر بتریب در سالهای ۱۳۵۹ و ۱۳۶۲ از دانشگاه اوکلاهوما امریکا به پایان رسانده است و هم‌اکنون استاد دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر می باشد. نامبرده قبل از پیوستش به دانشگاه صنعتی امیرکبیر در سالهای ۱۳۶۲ الی ۱۳۶۴ استادیار دانشگاه میشیگان غربی و در سالهای ۱۳۶۴ الی ۱۳۷۰ دانشیار دانشگاه اوهایو در ایالات متحده امریکا بوده است. زمینه های تحقیقاتی مورد علاقه ایشان عبارتند از: الگوریتم های موازی، پردازش موازی، محاسبات نرم و کاربردهای آن، شبکه های کامپیوتری و مهندسی نرم افزار.



شکل ۱۵: بررسی رفتار بازی از نظر پاداش دریافتی با توجه به الگوریتم یادگیری تغییریافته.

نمایید. همچنین معیاری به نام آنتروپی کلی برای بررسی رفتار الگوریتم یادگیری در محیط چندعامله پیشنهاد گردید. تنظیم پارامترهای پاداش و جریمه اutomataهای یادگیر می تواند کارایی حل مسئله را افزایش داده به طوری که استفاده از تغییرات پارامتر  $K$  در طول اجرای بازی می تواند زمان رسیدن به نقاط تعادل را بهبود بخشد. مدل ارائه شده می تواند با ایجاد تغییرات در نحوه پاداش دهی عامل ها و با توجه به پویایی محیط مسائل غیر همکارانه و همکارانه نیز مورد استفاده قرار گیرد. با توجه به نتایج آزمایش ها و مقایسه با مدل های مشابه اutomataهای یادگیر مدل مناسب یادگیری بین عامل ها در سیستم های چندعامله بوده و می تواند به عنوان راه حلی مناسب و کارا در بازی های مارکوفی به کار روند.

## مراجع

- [1] G. Weiss, *Multi - Agent Systems: A Modern Approach to Distributed Artificial Intelligence*, Cambridge, MA: MIT Press, 1999.
- [2] H. Van Dyke Parunak, "A practitioners' review of industrial agent applications, autonomous agents and multi-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 3, no. 4, pp. 389-407, Dec. 2000.
- [3] A. K. Goel, V. Kummar, and S. Sirivasan, "Application of multi - agent system & agent coordination," in *Proc. 2nd National Conf. Mathematical Techniques: Emerging Paradigms for Electronics and IT Industries, MATEIT'2008*, pp. 328-337, New Dehli, India, Sep. 2008.
- [4] L. Busnui, R. Babuska, and B. Schutter, "A comprehensive survey of multi-agent reinforcement learning," *IEEE Trans. on System, Man, Cybern.*, vol. 38, no. 2, pp. 156-171, Mar. 2008.
- [5] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. ICML'94*, pp. 157-163, Mar. 1994.
- [6] J. Osborne and A. Rubinstein, *A Course in Game Theory*, Cambridge, MA: MIT Press, 1994.
- [7] J. Hu and M. P. Wellman, "Online learning about other agents in a dynamic multi-agent system," *J. of Cognitive System Research*, vol. 2, no. 1, pp. 67-79, Apr. 2001.
- [8] J. Hu and M. P. Wellman, "Nash Q-learning for general - sum stochastic games," *J. of Machine Learning Research*, vol. 4, pp. 1039-1069, Nov. 2003.
- [9] A. Greenwald and K. Hall, "Correlated Q - learning," in *Proc. of the Twentieth Int. Conf. on Machine Learning, ICML'2003*, pp. 242-249, Aug. 2003.