

ارائه‌ی الگوریتمی برای بیشینه سازی سود فروشنده‌گان در بازار دوانحصاری بدون دانستن سود رقبا

نادیا فرهادی^۱، بهروز معصومی^۱، محمدرضا میبدی^۲

^۱ دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه آزاد، قزوین

Nadia.Farhadi@gmail.com, Masoumi@qiau.ac.ir

^۲ دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران

Mmeybodi@aut.ac.ir

چکیده

امروزه استفاده از عامل‌های هوشمند در امور اقتصادی به ویژه در مبحث بازارها به منظور قیمت‌گذاری برای فروشنده‌گان و همچنین پیدا کردن بهترین گزینه برای خریداران، در حال گسترش است. برای تحلیل بهتر این گونه مسائل، از مدل کردن بازارها بصورت بازی استفاده شده است تا از مزایای نظریه بازی‌ها نیز بتوان بهره برد. با توجه به بررسی تحقیقات انجام شده در این زمینه دیده شده است که به مسئله‌ی عدم مشاهده تمامی اطلاعات رقبا، کمتر پرداخته شده است. هدف این مقاله ارائه‌ی الگوریتمی برای بازار دوانحصاری است که در آن ربات‌های قیمت‌گذار در حالیکه سود و عمل رقیب خود را نمی‌بینند، یاد می‌گیرند بهترین قیمت را برای فروشنده تعیین کنند تا بیشترین سود را حاصل کند. برای این منظور، از روش تاخیر در به روز رسانی سیاست عامل‌ها استفاده شده است که با ایجاد این تاخیر، عامل کننده در نقش راهبر و عامل سریعتر در نقش پیرو عمل می‌کند. نتایج آزمایش‌های انجام گرفته نشان می‌دهند که الگوریتم ارائه شده به نقطه تعادل استکلبرگ همگرا می‌شود.

کلمات کلیدی

عامل‌های هوشمند، یادگیری تقویتی، قیمت‌گذاری، بازار دوانحصاری

فروش) ایفا می‌کنند. دلیل استفاده از یادگیری در این مسائل این است که این عامل‌ها مانند تمام ساخته‌های دست بشر، هوش محدودی دارند و لازم است برای استفاده در محیط‌های مختلف (که در آینده پیچیده‌تر نیز خواهد شد)، خود را با اولویت‌های کاربران یا رفتارهای رقیبان تطبیق دهند. این تطبیق می‌تواند بصورت‌های مختلفی به عامل‌ها اعمال شوند. برای نمونه یکی از روش‌های تحت تاثیر قرار دادن عامل‌ها، اثر مستقیم روى عامل توسط خود کاربر است. روش دیگری که روش کاراتری می‌باشد، اینست که عامل‌ها یاد بگیرند چگونه استراتژی‌های خود را با استفاده از مشاهدات و تجربیات گذشته تغییر دهند. روش‌های یادگیری که تاکنون در قیمت‌گذاری در محیط‌های چند عامله ارائه شده اند یا بر ایستاد بودن سایر عوامل متکی هستند و یا مشاهده عمل رقبا را مفروض می‌دانند. در محیط‌هایی همچون بازار هیچیک از این دو پیش‌فرض برقرار نمی‌باشد. به عنوان مثال در یک بازار تمامی فروشنده‌گان سعی دارند از نوسانات بازار جهت یادگیری قیمت‌گذاری بهینه در آینده استفاده کنند. همچنین بسیاری از فرادرادهای بازار مخفیانه بوده و لزوماً رقبا از تراکنش‌های انجام شده

۱- مقدمه

مسئله قیمت‌گذاری یکی از مسائل اقتصادی است که امروزه ترکیب آن با هوش مصنوعی مورد توجه قرار گرفته و روش‌های مختلفی برای شبیه‌سازی این‌گونه مسائل و بدست آوردن سود بیشتر بکار رفته است. در مطالعات اقتصادی در زمینه بازار و قیمت‌گذاری با دو دسته کلی مواجه هستیم: مدل (یا بازی) کورنو و مدل برترند^[۱]. در مدل کورنو توکلیدکننده‌گان کمیت فروش خود را انتخاب می‌کنند در حالی که در مدل برترند توکلیدکننده‌گان، باید قیمت مناسب برای اجناس را انتخاب کنند^[۲]. یکی از ساده‌ترین حالت‌های اقتصادی که بتوان پارامترهای مورد نظر را در آن بررسی کرد، بازار دوانحصاری می‌باشد به این معنا که کالای مورد نظر در یک بازار منحصراً توسط دو فروشنده ارائه می‌گردد. استفاده از عامل‌های نرم‌افزاری تعاملی در این مسائل کمک چشمگیری به خریداران و حتی فروشنده‌گان خواهد داشت. در واقع عامل‌های نرم‌افزاری بعنوان تصمیم‌گیرنده‌های اقتصادی عمل کرده و نقش اساسی در جنبه‌های مختلف تجارت (شامل مذاکرات و خرید و

بر تعریف MDP است. در واقع یک MDP بازی تصادفی با یک بازیگر است.

اکنون که اقتصاد عاملی مورد نظر را به یک بازی مارکوف مدل کردیم، برای بدست آوردن بیشترین سود باید سیاست بهینه برای این بازی را پیدا کنیم. این سیاست بهینه در واقع راه حل این بازی محسوب می‌شود. یک گروه از راه حل هایی که برای بیشینه سازی در این گونه بازی ها بکار می‌رود استفاده از یادگیری می‌باشد.

روش های زیادی برای یادگیری در این محیط ها برای عامل های قیمت-گذار بکار رفته است که تعدادی از آنها عبارتند از روش های مبتنی بر تئوری بازی، روش های نزدیک بینانه بهینه، روش های پیروی اشتراقی و روش های یادگیری بدون پشمیمانی [۵]. یکی از روش هایی که عموماً در تئوری بازیها مورد مطالعه قرار گرفته است، یادگیری تقویتی می‌باشد.

یادگیری تقویتی راهی برای آموزش عامل ها از طریق آزمون و خطاست که در محیط های پویا انجام می‌گیرد. یعنی بدون اینکه نحوه انجام عمل را برای عامل مشخص کنیم از طریق پاداش و جریمه دادن به عامل، یادگیری صورت می‌گیرد. عامل بدون دانستن اطلاعاتی از محیط با استفاده از بهینه سازی بر اساس روش های تقریبی و تخمینی به هدف دست می‌یابد. انواع الگوریتم های یادگیری به تفصیل در [۶] شرح داده شده است.

یک الگوریتم بسیار معروف یادگیری تقویتی در محیط های MDP، یادگیری Q می‌باشد. این روش یادگیری تقویتی، مقادیر Q را به هر زوج (حالت، عمل) (s, a) نسبت می‌دهد. این مقدار هر بار که حالت s رویت شده و عمل a انجام شود به روز می‌گردد. قانون به روز کردن آن در رابطه ۱ آمده است.

$$Q(s, a) \leftarrow (1 - \alpha) Q(s, a) + \alpha[R(s, a) + \gamma V(s')] \\ V(s) = \max_a Q(s, a) \quad (1)$$

که در رابطه فوق γ نشان می‌دهد، عامل چقدر نزدیک نگر بوده و یا علاقمند به پاداش های بلند مدت است و α نشان دهنده میزان بستگی عامل به تجربیات اخیر در مقایسه با تجربیات گذشته می‌باشد. همانطور که در رابطه ۱ مشاهده می‌شود، به هر حالت نیز یک مقدار نسبت داده می‌شود که متناظر با ارزش پرازدش ترین عمل در آن حالت است.

یادگیری تقویتی در اقتصاد عاملی که موضوع مورد بحث ما می‌باشد، برای یادگیری [۷] عامل ها به منظور تطبیق با محیط بکار می‌رود [۹، ۸]. استفاده از یادگیری تقویتی به عنوان روشی برای یادگیری استراتژی مناسب توسعه عامل ها در این نوع اقتصادهای عاملی، ملزم به رعایت ویژگی مارکوف در محیطی است که این عامل ها به تعامل در آن محیط مشغول هستند [۸]. به عنوان مثال اگر بازاری دوانحصاری را در نظر بگیریم که فروشنده اول از استراتژی قیمت گذاری ثابت استفاده کند، محیط برای فروشنده دوم شرایط مارکوف را داشته و محیطی ایستا خواهد بود که به تاریخچه نیز وابستگی ندارد. (البته در صورتی که محیط برای فروشنده دوم کاملاً قابل مشاهده باشد).

توسط دیگران مطلع نیستند. دسته دیگری از مسایل که با روش های موجود قابل حل نیستند بازی هایی هستند مثل پوکر، که بازیگران از دست رقیب و استراتژی او بی خبرند. بنابراین در چنین محیط هایی روش های طرح شده گذشته، الزاماً کارایی نداشته و نیاز به الگوریتم های جدیدی برای یادگیری می‌باشد. در این مقاله روش برای حل مساله ای غیر ایستاد بودن محیط، با استفاده از تاخیر در به روز کردن سیاست بازیگران ارائه شده است. در این روش هر بازیگر برای مدت قابل قبولی استراتژی خود را حفظ کرده و به یادگیری بهترین سیاست برای واکنش به رقیبان خود می‌پردازد. سپس بازیگران استراتژی خود را به عکس العمل بهینه تغییر داده و به یادگیری ادامه می‌دهند. نقطه تعادل بازی بسته به زمان به روز کردن سیاست توسط عامل ها می‌تواند ترکیبی از تعادل نش [۳] یا استکلیرگ باشد. به این ترتیب که عوامل کند تر نقش راهبر و عوامل سریعتر نقش پیرو را در بازی به عهده گیرند. این مقاله به بررسی عامل های قیمت گذار در یک بازار دو انحصاری محدود می‌شود که فروشنده گان از عمل و سود رقیبان خود اطلاعی در دست ندارند. در ادامه مقاله چنین سازماندهی شده است: در بخش ۲ تعاریف اولیه و مرواری بر کارهای قبلی بیان می‌شود. در بخش ۳ الگوریتم پیشنهادی ارائه شده است. در بخش ۴ ارزیابی الگوریتم پیشنهادی و در آخر نتیجه گیری آورده شده است.

۲- مفاهیم اولیه و مرواری بر کارهای گذشته

یک اقتصاد عاملی به عنوان نوعی سیستم چند عامله است که می‌توان این سیستم را با بازیهای مارکوف مدل کرد [۴]. بازی مارکوف در واقع یک فرآیند تصمیم گیری مارکوف چند عامله است. که در ادامه به توضیح آنها می‌پردازیم:

تعريف ۱. فرآیند تصادفی مارکوف (MDP) به صورت چند تایی $\langle S, A, R, T \rangle$ نشان داده می‌شود که در آن S مجموعه متناهی از وضعیت ها، A مجموعه عملیات قابل دسترس برای عامل، γ ضریب کاهش و $T: S \times A \times S \rightarrow [0, 1]$: احتمال انتقال از وضعیت جاری به وضعیت بعدی با انجام عمل a است و $R: S \times A \rightarrow \mathbb{R}$: تابع پاداش است که یک مقدار عددی را بر می‌گرداند. MDP با این ویژگی مهم تعریف می‌گردد که تابع انتقال حالت، فقط بستگی به آخرین حالت و عمل عامل داشته و مستقل از سابقه حالات و اعمال عامل است.

تعريف ۲. بازی مارکوف یا بازی تصادفی به صورت چند تایی مرتب $\langle n, S, \vec{A}, T, \vec{R} \rangle$ تعریف می‌گردد. در این تعریف n تعداد عامل هاست، \vec{A} بردار عمل مشترک است و \vec{R} بردار سود مشترک شامل سود تک S . $S \times A \rightarrow \mathbb{R}$ تک عامل هاست که تابعی از اعمال مشترک می‌باشد. S حالت بازی را مشخص می‌کند، و T تابع انتقال حالت است و بر اساس حالت قبلی و عملکرد بازیگران، حالت جدید بازی را تعیین می‌کند $[0, 1] \rightarrow S \times A \times S$. البته R علاوه بر عملکرد بازیگران تابع حالت بازی نیز هست. بنابراین می‌توان گفت تعریف بازی های تصادفی بسطی

توانند از ربات قیمت‌گذار استفاده کنند. ربات قیمت‌گذار به فروشنده‌گان کمک می‌کند با در نظر گرفتن شرایط بازار و تطبیق با این شرایط، قیمت مناسبی انتخاب کنند تا سود بیشتری بدست آورند و ربات خرید نیز به مشتریان کمک می‌کند قیمت‌ها و دیگر پارامترهای مورد نظرشان را بررسی کرده و بهترین محصول با مناسب‌ترین قیمت ممکن را انتخاب کنند (توضیحات بیشتر در [۵]). معمولاً قیمت‌های تقاضای بازار توسط دو حد آستانه (هزینه مرزی و هزینه رزوایشیون) محدود می‌شود که قیمت‌های تعیین شده باید در این محدوده قرار داشته باشند [۱۹].

معمولًاً فرض می‌شود تعدادی از خریداران از ربات‌های خرید استفاده می‌کنند و تعدادی دیگر نیز بصورت تصادفی خرید می‌کنند یا به یک فروشگاه خاص علاقمند هستند و همیشه از آن فروشگاه خاص خرید می‌کنند. همچنین در مقالات مختلف به این موضوع که خریداران نیز به عنوان بازیگران در نظر گرفته شوند و پارامترهای خاص مربوط به خریداران نیز پرداخته شده است [۲۰ و ۲۱] در سالهای اخیر نیز بررسی‌های نسبتاً زیادی در مورد محیط‌های نامعین انجام شده است که این نامعین بودن در مورد تقاضا در [۲۲، ۲۳] در نظر گرفته شده و با استفاده از یادگیری بیزین کارایی را بهبود بخشیده اند و یا در [۲۴] با استفاده از الگوریتم‌های تکاملی، پارامترهای مربوط به قیمت‌گذاری خود را بهبود بخشیده اند.

اما مبحث مورد علاقه ما ربات‌های قیمت‌گذار هستند که مورد استفاده فروشنده‌گان قرار می‌گیرند و به آنها کمک می‌کنند با تعیین یک استراتژی قیمت‌گذاری سود بیشتری بدست آورند. در این مقاله ما به بررسی این ربات‌ها در مدل برترنده می‌پردازیم. مدل برترنده دارای جزیاتی می‌باشد که در ادامه در مورد آنها صحبت می‌کنیم.

مدل برترنده همانطور که قبل اشاره شد تعاملات بین فروشنده‌گانی است که قیمت کالایی را تعیین می‌کنند. بنابراین از آنجا که ما بازار دو-انحصاری را مورد بررسی قرار می‌دهیم در واقع به بررسی تعاملات بین دو فروشنده برای فروش کالایی یکسان می‌پردازیم که فرضیات مدل برترنده را شامل می‌شود یعنی دو فروشنده نمی‌توانند تبانی داشته باشند و بر قیمت‌گذاری آن کالا با یکدیگر رقابت می‌کنند. اگر قیمتی که فروشنده ۱ برای کالا قرار می‌دهد را p_1 و قیمت فروشنده ۲ را p_2 بنامیم و تعداد کالایی که فروشنده ۱ می‌فروشد را q_1 و تعداد کالایی که فروشنده ۲ می‌فروشد را q_2 قرار دهیم، رابطه ۲ و ۳ بصورت زیر تعریف می‌شود:

$$q_1 = Xp_1 + Yp_2 + b \quad (2)$$

$$q_2 = Yp_1 + Xp_2 + b \quad (3)$$

همچنین فرض شده است که هر کالا هزینه مرزی C برای هر فروشنده دارد که اگر قیمت فروش هر فروشنده از آن هزینه‌ی مرزی بیشتر باشد، فروشنده‌ی آن از فروش این کالا طبق رابطه ۴ سود می‌برد.

$$\Pi_i = p_i q_i - c_i q_i \quad (4)$$

که c_i هزینه‌ی مرزی فروشنده‌ی i است.

راه حل بازی‌های مارکوف از آنجا که محیط شامل چندین عامل می‌باشد، باید تعادلی شامل عملکرد تمامی عامل‌ها باشد. یکی از این تعادل‌ها، تعادل استکلبرگ است که در ادامه به توضیح آن می‌پردازیم. تعادل استکلبرگ یا مدل راهبر – پیرو، به این صورت است که بازیگران به صورت متواتی تصمیم می‌گیرند و در بازار شرکت می‌کنند. به این ترتیب که راهبر بازی را شروع کرده و پیرو با اطلاع از تصمیم واقعی اخذ شده توسط راهبر، بهترین عمل را برای افزایش سود خود انجام می‌دهد. بنابراین راهبر با دانستنتابع عکس‌العمل پیرو نسبت به عمل راهبر، بهترین عمل خود را انتخاب می‌کند و به این صورت، نقطه تعادل استکلبرگ بازی بدست می‌آید. در واقع مدل استکلبرگ بیانگر توالی تصمیم‌گیری بازیگران و تبادل اطلاعات در بازی است.

اما ممکن است در شرایطی که اطلاعات کاملی از محیط یا مدل آن در دست نیست به این عامل‌ها برای قیمت‌گذاری نیاز داشته باشیم. مثلاً در [۱۰] فرآیند مارکوف با قابلیت دید جزئی برای عدم قطعیت‌های محیط در مورد پارامترهای مربوط به مشتریان (مثلًاً نرخ ورود، تعداد واقعی مشتریان، ویژگی‌های آماری مربوط به قیمت رزوایشیون و ...) مورد بررسی قرار گرفته است. در اینکونه شرایطی که اطلاعات کاملی از محیط در دسترس نباشد، یادگیری Q نتایج خوبی می‌دهد زیرا نیازی به مدل ندارد [۱۱] و می‌تواند برای یادگیری برخط بکار رود. اما می‌دانیم که همگرایی یادگیری Q در شرایطی تضمین می‌شود که شرایط مارکوف برقرار باشد در اینصورت می‌توان از جداول Q استفاده کرد و نیز میدانیم که استفاده از این جداول جاییکه فضای حالت بزرگ باشد مناسب نیست و در مقالات مختلف روش‌های یادگیری تقویتی در فضای حالت بزرگ بصورت ترکیب با تقریب‌زننده‌های تابع بکار می‌روند تا کارایی خوبی ارائه دهند. برای مثال در [۱۲] درخت رگرسیون و در [۱۳] شبکه‌های عصبی، بررسی شده اند.

در شرایطی که همه فروشنده‌ها (با تعداد ۲ یا بیشتر) بخواهند یاد بگیرند چگونه قیمت‌گذاری کنند، مسئله تبدیل به یادگیری در سیستم‌های چندعامله [۱۴] می‌شود که این مسئله مارکوفی نیست. زیرا حالت بعدی برای هر عامل نه تنها به حالت فعلی و عمل خود که به عمل دیگر عامل‌ها نیز وابسته است به این مفهوم که هر عامل برای دیگر عامل‌ها محیط غیرایستایی فراهم می‌کند که در چنین شرایطی گارانتی برای همگرایی وجود ندارد. این مسئله در [۲] برای یادگیری Q و در [۱۵] برای یادگیری نش [۳] بررسی شده است که البته نقص‌هایی هم دارد. همچنین بسته به اینکه چه اطلاعاتی از بازار موجود باشد (اولویت‌های خریداران یا قیمت رقبا و اطلاعاتی از این دست) فروشنده‌گان از استراتژی قیمت‌گذاری که مناسب با آن شرایط باشد استفاده می‌کنند. [۱۶، ۱۷] استفاده از یادگیری‌های نامتقارن نیز در مقالات قیمت‌گذاری بکار رفته است که در [۱۸] می‌توان نمونه‌ای از آن را مورد بررسی قرار داد. در اقتصادهای عاملی عموماً تعداد زیادی خریدار و تعدادی فروشنده وجود دارند که خریدارها می‌توانند از ربات‌خرید و فروشنده‌گان نیز می‌

۳- الگوریتم پیشنهادی

های کندر استراتژی خود را تغییر دهنده، عامل‌های سریع‌تر بهترین پاسخ خود را نسبت به عمل رقیبان یاد گرفته باشد.

بنابراین اولین مرحله الگوریتم تشخیص یک T_i به عامل A می‌باشد که این عامل پس از گذشت هر T_i زمان سیاست خود را به روز کند. بعد از آن، بازیگران جداول Q خود را فقط بر حسب عمل خود تشکیل داده و آن را مقدار دهی اولیه می‌کنند. انتخاب عمل بازیگران باید بر اساس سیاست ϵ -greedy یا هر سیاست تصادفی دیگری باشد تا به آنها اجازه دهد همه اعمال خود را بیازمایند. این مساله وقتی اهمیت می‌یابد که عامل‌های سریعتر که عمل بهینه خود را یافته‌اند مجبور به تغییر آن به خاطر تغییر رفتار عوامل کند تر شده و عمل بهینه جدید خود را جستجو می‌کنند. در هر تکرار بازی عامل‌ها مقادیر Q خود را طبق رابطه ۵ به روز می‌کنند:

$$Q_i(a_i) = \alpha_i Q_i(a_i) + (1 - \alpha_i)r_i \quad (5)$$

از آنجا که بازی ماتریسی تکراری است، متغیر حالت در رابطه ۱ وجود ندارد. α باید بر اساس توزیع احتمالی سودها و درجه نامعینی سیستم تعیین گردد. به علاوه بهتر است که عوامل سریعتر α کوچکتری داشته باشد و بالعکس.

پس از گذشت زمان T_i برای هر بازیگر A ، این عامل سیاست خود را به روز خواهد کرد. برای یک سیاست ϵ -greedy، سیاست جدید انتخاب بهترین عمل با بالاترین مقدار Q با احتمال بالا (exploitation) و انتخاب تصادفی یک عمل دیگر با احتمال کم (exploration) خواهد بود. این مساله در رابطه ۶ نشان داده شده است:

$$\begin{cases} a_i = \operatorname{argmax}_{a_i} Q_i(a_i) & \text{Probability } 1 - \epsilon \\ a_i = \operatorname{rand}(A_i) & \text{Probability } \epsilon \end{cases} \quad (6)$$

این عمل، برای عامل A در دوره بعد بازی ثابت خواهد بود. می‌توان اثبات کرد که وقتی کندرین عامل تمام عمل‌های خود را امتحان کرده و بهترین عمل خود را باید همه عامل‌ها، با فرض اینکه عوامل کندر در نقش راهبر برای عوامل تندتر هستند، به مقادیر Q خود تحت تعادل استکلبرگ همگرا شده‌اند.

اثبات این مساله وقتی که بدانیم الگوریتم یادگیری Q به مقادیر Q تحت سیاست بهینه همگرا می‌شود، ساده خواهد بود. ابتدا فرض می‌شود الگوریتم روی یک بازی تک عامله با مجموعه عمل‌های ممکن A_1 کار می‌کند. بازی تک عامله در واقع همان MDP با یک حالت است. ثابت شده است که چنین مساله‌ای با روش یادگیری Q به مقادیر Q تحت سیاست بهینه همگرا می‌گردد. در این مساله هر چیزی می‌تواند باشد، پس ما آن را به سادگی با ۱ برابر می‌گیریم. این الگوریتم در زمان مشخصی همگرا می‌گردد. می‌توان زمانی به قدر کافی بزرگتر از آن را در نظر گرفت که مطمئن باشیم این روش در این زمان قطعاً همگرا می‌شود. این زمان را $T_{\text{conv},1}$ می‌نامیم. با اضافه کردن یک بازیگر دیگر به سیستم، یک بازی دو عامله ایجاد خواهد شد. به این جهت باید T_2 را به گونه‌ای تنظیم کرد که شرایطی که قبلاً به آنها اشاره کردیم برقرار شود. T_2 می‌تواند به ترتیب رابطه ۷ تعیین گردد:

در روش پیشنهادی ارائه شده برای حل مساله‌ی غیر ایستا بودن محیط، از تاخیر در به روز کردن سیاست بازیگران استفاده شده است. در این روش هر بازیگر برای مدت قابل قبولی استراتژی خود را حفظ کرده و به یادگیری بهترین سیاست برای واکنش به رقیبان خود می‌پردازد. سپس بازیگران استراتژی خود را به عکس العمل بهینه تغییر داده و به یادگیری ادامه می‌دهند. این مقاله به بررسی عامل‌های قیمت‌گذار در یک بازار دو انحصاری محدود می‌شود که فروشنده‌گان از عمل و سود رقیبان خود اطلاعی در دست ندارند. در واقع ما در این بخش الگوریتمی برای بیشینه سازی سود فروشنده‌گان در بازار دوانحصاری بدون دانستن قیمت و سود رقبا ارائه می‌دهیم و با ایجاد تاخیر در به روز رسانی سیاست عامل‌ها در یادگیری تقویتی مربوط به بازی‌های تصادفی باعث بهبود یادگیری شده است به گونه‌ای که نیازی به مشاهده سود و حتی استراتژی رقیب نداشته و کافیست هر عامل تنها بتواند سود خود را مشاهده کند.

همانطور که بیان شد، یکی از مشکلات روش‌های یادگیری در محیط‌های چند عامله ایستا نبودن محیط است که به دلیل تاثیر عمل رقبا در سود دیگر بازیگران بوجود می‌آید. در اکثر تحقیقات پیشین، این مشکل، با یادگیری روی بردار اعمال گروهی \vec{a} به جای عمل تک تک عامل‌ها رفع شده است. این راه حل اگرچه کلی است ولی به این معنی است که هر بازیگر عمل رقبای خود را مشاهده می‌کند، فرضی که در محیط‌هایی مثل بازار درست به نظر نمی‌رسد. به علاوه مشکل دیگر این روش‌ها که عمدتاً بر اساس انتخاب نقطه تعادل عمل می‌کنند، عدم توانایی آنها در بازی مناسب در مقابل رقیبان ایستایی است که استراتژی‌های تعادل را بازی نمی‌کنند. در یادگیری تقویتی یکی از مهمترین فرضیات اینست که عامل‌ها در پی ماقزیم کردن سود خود هستند، بنابراین به نظر می‌رسد بهتر باشد اپراتور ماقزیم‌گیری در رابطه ارزش حالت، را حفظ کرده و آن را با اپراتور تعادل جایگزین نکنیم. خواه بازی به نقطه تعادل نش همگرا بشود و خواه نشود، به نظر می‌رسد ماقزیم کردن سود نباید دستخوش تغییر شود.

روش ما پیشنهاد می‌دهد بازیگران از مرتبه‌های مختلف زمانی برای به روز کردن سیاست خود استفاده کنند تا محیط برای مدتی ایستا باقی بماند و گارانتی همگرایی به سیاست بهینه در MDP ها برقرار بماند. این روش قرار است در بازی‌های تکراری که با چند تابی مرتب $(n, A_1, \dots, A_n, R_1, \dots, R_n)$ تعریف می‌شود، عمل کند. هر بازیگر باید زمانی برای به روز رسانی سیاست خود داشته باشد که به طوری که هر بازیگر باید در مرتبه‌ی زمانی از دیگران متفاوت باشد. عامل‌هایی که مرتبه‌ی زمانی طولانی‌تری دارند کندر بوده و به دینامیک محیط که شامل عملکرد رقیبان است، کمتر حساس هستند. در مقابل، عوامل با مرتبه‌ی زمانی کمتر، سریعتر بوده و خود را با سرعت بیشتری با تغییرات محیط تطبیق می‌دهند. مرتبه‌های زمانی باید با توجه به مجموعه عمل‌های ممکن عامل‌ها تعیین گردد تا پیش از آنکه عامل-

وجود دارد که دو فروشنده ای انحصاری آن برای بدست آوردن سود بیشتر رقابت می کنند. در واقع ما به بررسی مدل ربات خرید پرداختیم. در مدل ما فرض می شود مطابق با یکتابع تقاضا خریدارانی وجود دارند که با جستجوی کمترین قیمت به خرید آن کالا می پردازند. پس از اینکه خریداران فروشنده را برای خرید انتخاب کردند، سودی از این خرید نسبیت فروشنده می شود. قسمت مورد توجه در مقاله ما اینست که فرض می شود هیچ کدام از فروشنده ها از این تابع تقاضا و همچنین قیمت و پاداش فروشنده دیگر اطلاعی ندارد.

فرض می کنیم عامل ۱ عامل کندر است که مرتبه زمانی آن برابر ۱۰۰۰ واحد زمانی درنظر گرفته شده و عامل شماره ۲ یعنی عامل سریعتر در هر ۱۰۰۰ واحد که استراتژی عامل ۱ ثابت است به یادگیری می پردازد تا بهترین قیمت را نسبت به قیمت عامل کندر انتخاب کند. هر ۱۰۰۰ گام زمانی عامل کندر (راهبر) قیمت خود را تغییر می دهد و باز عامل سریعتر یادمی گیرد بهترین پاسخ خود را انتخاب کند. از آنجاییکه در این ۱۰۰۰ گام زمانی عامل کندر قیمت خود را ثابت نگه می دارد، محیط برای عامل سریعتر ایستا بوده و میتواند از یادگیری Q تک عامله برای یادگیری خود استفاده کند. تابع سود به این صورت محاسبه می شود که میزان درآمد حاصل از تعداد کالاهای فروخته شده را بدست آورده و به همان تعداد فروش، هزینه مرزی، از این مقدار کم می شود تا سود خالص محاسبه شود. هزینه مرزی در پیاده سازی ما برابر با ۱ در نظر گرفته شده است یعنی داریم $c_1 = c_2 = 1$. دامنه استراتژی بازیگران ۱ تا ۲۵ است یعنی قیمت های مجاز برای هر یک از عامل ها می باشد که آن را در بازه های واحد، گسسته کرده ایم. به نظر می رسد الگوریتم احتیاج به ۵۰ دوره ۱۰۰ راندی برای همگرا شدن به نقطه تعادل خود دارد. مقادیر اولیه جداول Q را نیز طوری مقداردهی کردیم که از سودها بسیار بیشتر باشد تا تمامی اعمال ممکن برای عامل ها بررسی شود.

برای بررسی دقیقت، به مقایسه نتایج تئوری محاسبات برای نقطه تعادل استکلبرگ با نتایج بدست آمده پرداختیم. همانطور که در رابطه ۲ و ۳ و ۴ بیان شد توابع میزان فروش هر فروشنده و سود حاصل از فروش در مدل برترند مشخص شده است، ما برای مدل خود پارامترهای زیر را تعیین کرده و شبیه سازی را انجام دادیم:

$$q_1 = -10p_1 + 10p_2 + 100 \quad (9)$$

$$q_2 = 10p_1 - 10p_2 + 100 \quad (10)$$

بنابراین طبق رابطه ۴ داریم:

$$\Pi_1 = p_1 q_1 - c_1 q_1 = (p_1 - c_1) q_1 \quad (11)$$

$$\Pi_2 = p_2 q_2 - c_2 q_2 = (p_2 - c_2) q_2 \quad (12)$$

اکنون برای بدست آوردن ماکزیمم قیمت پیرو از سود آن نسبت به قیمت مشتق گرفته و برابر با صفر قرار می دهیم:

$$\frac{\partial \Pi_2}{\partial p_2} = q_2 + (p_2 - c_2) \frac{\partial q_2}{\partial p_2}$$

$$T_2 = T_{conv.1} \quad (7)$$

در ابتدای بازی بازیگر ۲، یک عمل را انتخاب کرده و آن را برای T_2 راند بازی می کند، زمانی که مطمئن هستیم بازیگر ۱ وقت کافی برای یافتن بهترین عکس العمل به عمل بازیگر ۲ داشته است. در این زمان بازیگر ۱ به مقادیر Q بهینه نسبت به عمل بازیگر ۲، و بازیگر ۲ به مقادیر Q عمل خودش بر حسب عکس العمل بهینه بازیگر ۱ همگرا شده است. در این زمان عامل ۲، استراتژی خود را به علت مقدار دهی اولیه با مقادیری بالاتر از سودهای موجود در سیستم، عوض کرده و عمل جدیدی را برای T_2 مرحله دیگر انجام می دهد. این فرایند آنقدر تکرار می شود تا همه استراتژی های بازیگر ۲ آزموده شده و عامل ۲ بهترین عمل خود را بر اساس عکس العمل رقبا بدست آورد. این نقطه تعادل استکلبرگ با راهبری بازیگر ۲ و پیروی بازیگر ۲ خواهد بود. بعثت آزمودن تمام استراتژی های بازیگر ۲ و انتخاب بهترین استراتژی براساس عکس العمل بازیگر ۱ است. دقیقا مانند حالت تک عامله می توان زمانی را تعیین کرد که مطمئن باشیم بازی در آن زمان $T_{conv.2}$ به نقطه تعادل خود رسیده است.

این روال را می توان به بازی های n نفره نیز بسط داد:

$$T_i = T_{conv.i-1} \quad (8)$$

این روش، به نقطه تعادل استکلبرگ که در آن بازیگران کندر راهبر بازیگران سریع تر هستند همگرا می گردد. بنابراین عوامل کندر سود بلندمدت بیشتری بدست آورده و در عوض بازیگران سریع تر زودتر با تغییرات تطبیق یافته و سود کوتاه مدت بیشتری کسب می کنند. همانطور که ملاحظه شد، بازیگران نیازی به رؤیت استراتژی رقیبان خود ندارند و فقط سود خود را رؤیت می کنند. این مهمنترین ویژگی الگوریتم می باشد. شبیه کد این الگوریتم در شکل ۱ آمده است.

Initialization for each agent i :

1. Initialize $Q_i(a_i)$
2. Assign T_i
3. Select a random action for each agent

Loop for ever :

1. Update Q-Values at each stage : $Q_i(a_i) = \alpha_i Q_i(a_i) + (1 - \alpha_i)r_i$
2. In each T_i update agent i 's policy:

$$\begin{cases} a_i = \arg \max_{a_i} Q_i(a_i) & \text{probability } 1 - \epsilon \\ a_i = \text{rand}(A_i) & \text{probability } \epsilon \end{cases}$$
3. Decrease ϵ

شکل (۱) : شبیه کد الگوریتم برای بازی های ماتریسی تکراری

۴- پیاده سازی و نتایج آزمایش ها

برای بررسی مدل پیشنهادی از مدل برترند برای بازار استفاده شده است. به این صورت که فرض می شود یک نوع کالای فاسد نشدنی

۵- نتیجه گیری

در این مقاله به ارائه‌ی الگوریتمی برای بازارهای دو انحصاری بدون دانستن سود و عمل رقبا پرداختیم. این کار را با بررسی موضوع قیمت گذاری در بازار شروع کرده و با مشاهده‌ی اینکه مسئله قیمت گذاری با بکار بردن یادگیری برای بهینه سازی سود، بطور گسترده‌ای مورد مطالعه قرار گرفته است و نیز اینکه این مطالعات نشان داده است که قیمت گذاری عاملی بخصوص با استفاده از یادگیری تقویتی تک عامله به نتایج خوبی رسیده است، به سمت ارائه‌ی چنین روشی سوق داده شدیم. زیرا مشاهده کردیم عدم قطعیت‌ها در محیط‌های پیچیده که استفاده از یادگیری تقویتی تک عامله میسر نیست، مشکل می‌نماید و اکثر بررسی عدم قطعیت‌ها در زمینه تابع درخواست و یا ورود مشتریان به بازار بوده است. همچنین مشاهده کردیم توجه اندکی معطوف به نامعین بودن پاداش‌ها و حتی اعمال رقیبان شده است و بیشتر نامعین بودن محیط را مربوط به مشتریان مورد بررسی قرار داده اند. در این مقاله با فرض نامشخص بودن عمل رقیب و پاداش او برای یک عامل به بررسی بازاری دو انحصاری پرداختیم و با ایستادن محیط برای بازارهای زمانی ثابتی به عامل دیگر فرصت یادگیری داده و مشاهده کردیم که با محاسبات انجام شده مشاهده کردیم که در این صورت بازار به تعادل استکلبرگ همگرا می‌شود.

مراجع

- [1] H. R. Varian, *Microeconomic analysis*, vol. 2. Norton New York, 1992.
- [2] G. Tesauro and J. O. Kephart, “Pricing in agent economies using multi-agent Q-learning,” *Autonomous Agents and Multi-Agent Systems*, vol. 5, no. 3, pp. 289–304, 2002.
- [3] J. Hu and M. P. Wellman, “Nash Q-learning for general-sum stochastic games,” *The Journal of Machine Learning Research*, vol. 4, pp. 1039–1069, 2003.
- [4] M. L. Littman, “Markov games as a framework for multiagent reinforcement learning,” in *Proceedings of the eleventh international conference on machine learning*, 1994, vol. 157, p. 163.
- [5] V. Könönen, “Dynamic pricing based on asymmetric multiagent reinforcement learning,” *International journal of intelligent systems*, vol. 21, no. 1, pp. 73–98, 2006.
- [6] A. Lipson and K. Leyton-Brown, “Empirically evaluating multiagent reinforcement learning algorithms,” Citeseer, 2005.
- [7] D. Fudenberg and D. K. Levine, *The theory of learning in games*, vol. 2. MIT press, 1998.
- [8] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1. Cambridge Univ Press, chap3-6, 1998.
- [9] Y. Shoham, R. Powers, and T. Grenager, “Multi-agent reinforcement learning: a critical survey,” (Tech. Rep.),

$$= 10p_1 - 10p_2 + 100 - 10p_2 + 10c_2 = 0$$

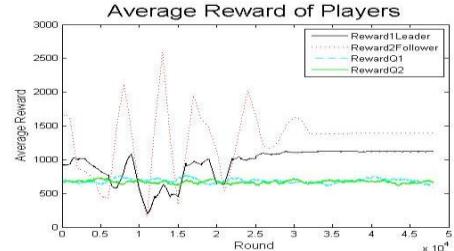
$$\Rightarrow p_2^* = 5 + \frac{1}{2}p_1 + \frac{1}{2} \quad (13)$$

بعد از پیرو نوبت به تعیین قیمت راهبر در پاسخ به قیمت قبلی پیرو می‌رسد، باز هم برای ماکریم‌گیری مشتق سود را نسبت به قیمت برابر با صفر قرار می‌دهیم:

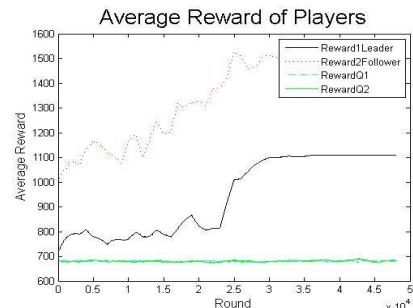
$$\begin{aligned} \Pi_1 &= p_1 q_1 - c_1 q_1 \\ &= (p_1 - c_1)(-10p_1 + 50 + 5p_1 + 5c_2) \\ &= (p_1 - c_1)(50 - 5p_1 + 5c_2) = 0 \\ \Rightarrow p_1^* &= \frac{50 + 5c_2 + 5c_1}{10} \end{aligned} \quad (14)$$

لازم است به این نکته اشاره شود که در مدل بازار ما فرض می‌کنیم هیچ یک از فروشنده‌گان این اطلاعات را ندارند و قادر به محاسبه‌ی نقطه‌ی تعادل استکلبرگ نیستند و فقط با یادگیری Q به نتایجی مشابه به تعادل استکلبرگ می‌رسند.

شکل ۲ مقایسه‌ی بازاری با ۲ عامل فوق در بازاری شامل ۲ فروشنده که هر دو به سادگی از یادگیری Q بدون تأخیر استفاده می‌کنند، نشان می‌دهد. از آنجا که تحقیقات انجام گرفته روی یادگیری‌های چندعاملی فرضیاتی متفاوت با فرضیات ما داشته اند با یادگیری Q تک عامله مقایسات را انجام دادیم که هیچ یک از اطلاعات رقیب آگاهی نداشته باشند.



شکل (۲): نمودار میانگین سود فروشنده‌ها با اجرای الگوریتم راهبر-پیرو در مقایسه با عامل‌های الگوریتم Q در یک اجرا برای اطمینان از عملکرد غیر تصادفی الگوریتم آن را در ۱۰۰ اجرای متولی نیز آزموده و میانگین مقدار را با تنظیمات قبلی در شکل ۳ نشان داده ایم.



شکل (۳): نمودار میانگین سود فروشنده‌ها با اجرای الگوریتم راهبر-پیرو در مقایسه با عامل‌های الگوریتم Q در ۱۰۰ اجرا

¹ Markov Decision Process

- [10] Y. Aviv and A. Pazgal, “A partially observed Markov decision process for dynamic pricing,” *Management Science*, vol. 51, no. 9, pp. 1400–1416, 2005.
- [11] Y. Shoham, R. Powers, and T. Grenager, “If multi-agent learning is the answer, what is the question?,” *Artificial Intelligence*, vol. 171, no. 7, pp. 365–377, 2007.
- [12] M. Sridharan and G. Tesauro, “Multi-agent Q-learning and regression trees for automated pricing decisions,” *Game Theory and Decision Theory in Agent-Based Systems*, pp. 217–234, 2002.
- [13] G. Tesauro, “Pricing in agent economies using neural networks and multi-agent Q-learning,” *Sequence learning*, pp. 288–307, 2001.
- [14] V. Kononen, “MULTIAGENT REINFORCEMENT LEARNING IN MARKOV GAMES: ASYMMETRIC AND SYMMETRIC APPROACHES,” Phd., Helsinki University of Technology, Finland, 2004.
- [15] J. Hu, “Dynamic Pricing Agents and Multi agent Learning,” 2003.
- [16] P. Dasgupta and R. Das, “Dynamic pricing with limited competitor information in a multi-agent economy,” in *Cooperative Information Systems*, 2000, pp. 299–310.
- [17] I. Popescu and Y. Wu, “Dynamic pricing strategies with reference effects,” *Operations Research*, vol. 55, no. 3, pp. 413–429, 2007.
- [18] V. Kononen and E. Oja, “Asymmetric multiagent reinforcement learning in pricing applications,” in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, 2004, vol. 2, pp. 1097–1102.
- [19] A. Greenwald and J. Kephart, “Shopbots and pricebots,” *Agent Mediated Electronic Commerce II*, pp. 1–23, 2000.
- [20] O. Besbes and A. Zeevi, “Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms,” *Operations Research*, vol. 57, no. 6, pp. 1407–1420, 2009.
- [21] K. Berg and H. Ehtamo, “Learning in nonlinear pricing with unknown utility functions,” *Annals of Operations Research*, vol. 172, no. 1, pp. 375–392, 2009.
- [22] J. M. Harrison, N. B. Keskin, and A. Zeevi, “Bayesian dynamic pricing policies: Learning and earning under a binary prior distribution,” *Management Science*, vol. 58, no. 3, pp. 570–586, 2012.
- [23] P. Johnson, A. Papaiyannis, and K. Poquet, “Dynamic Pricing Strategies in an Uncertain World,” 2012.
- [24] S. Ramezani, P. A. N. Bosman, and H. La Poutré, “Adaptive strategies for dynamic pricing agents,” in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, 2011, vol. 2, pp. 323–328.