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

نادیا فرهادی[1]، بهروز معصومی[2]، محمدرضا میبدی[3]

چکیده

امروزه استفاده از عامل‌های هوشمند در امور اقتصادی به ویژه در مبحث بازارها به منظور قیمت‌گذاری برای فروشندگان و همچنین پیدا کردن بهترین گزینه برای خریداران، در حال گسترش است. برای تحلیل بهتر این گونه مسائل، از مدل کردن بازار ها 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 کنند. برای این منظور، از روش تاخیر در به روز رسانی سیاست عامل‌ها استفاده شده است که با ایجاد این تاخیر، عامل کندتر در نقش راهبر و عامل سریعتر در نقش پیرو عمل می کند. نتایج آزمایش های انجام گرفته نشان می‌دهند که الگوریتم ارائه شده، به نقطه تعادل استکلبرگ همگرا می‌شود.

کلمات کلیدی

عامل‌های هوشمند، یادگیری تقویتی، قیمت گذاری، بازار دوانحصاری، اقتصاد

An Algorithm for Maximizing Seller's Profit in a Duopolistic Market with No Information on Competitor's Profit

Nadia Farhadi, Behrooz Masoumi, Mohammad Reza Meybodi

Abstract – There is ongoing research on the applications of intelligent agents in economic systems. This includes optimal pricing of the goods for the sellers and finding the best options for the buyers. To analyze such systems, the market is often modeled as a game providing the benefits of well-known game theoretic models. However, one assumption in most of the previous research is perfect information on rival's pay-off function. In this paper an algorithm is developed by which the agents learn their optimal pricing strategies in a duopoly without knowing the profit or action of their rival. To do so, a delay is applied to policy update of one of the agents. With this delay, the slower agent acts as a leader while the faster agent behaves as a follower after the learning process. The results show that the algorithm converges to Stackelberg Equilibrium

Keywords: Intelligent agents, Reinforcement Learning, Pricing, Duopolistic Market, Economics

[1] دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه آزاد، قزوین - Nadia.Farhadi@gmail.com - 09122534952

[2] دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه آزاد، قزوین - Masoumi@qiau.ac.ir

[3] دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران - Mmeybodi@aut.ac.ir

1- مقدمه

مسئله قیمت‌گذاری یکی از مسائل اقتصادی است که امروزه ترکیب آن با هوش مصنوعی مورد توجه قرار گرفته و روش های مختلفی برای شبیه‌سازی اینگونه مسائل و بدست آوردن سود بیشتر بکار رفته است. در مطالعات اقتصادی در زمینه بازار و قیمت گذاری با دو دسته کلی مواجه هستیم: مدل (یا بازی) کورنو و مدل برترند[1]. در مدل کورنو تولیدکنندگان، کمیت فروش خود را انتخاب می‌کنند در حالی که در مدل برترند تولیدکنندگان، باید قمیت مناسب برای اجناس را انتخاب کنند[2]. یکی از ساده ترین 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 منحصرا توسط دو فروشنده ارائه می‌گردد. استفاده از عامل‌های نرم‌افزاری تعاملی در این مسائل کمک چشمگیری به خریداران و حتی فروشندگان خواهد داشت. در واقع عامل‌های نرم‌افزاری بعنوان تصمیم گیرنده‌های اقتصادی عمل کرده و نقش اساسی در جنبه‌های مختلف تجارت (شامل مذاکرات و خرید و فروش) ایفا می‌کنند. دلیل استفاده از یادگیری در این مسائل این است که این عامل ها مانند تمام ساخته‌های دست بشر، هوش محدودی دارند و لازم است برای استفاده در محیط‌های مختلف (که در آینده پیچیده‌تر نیز خواهند 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 روی 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. روش های یادگیری که تاکنون در قیمت‌گذاری در محیط‌های چندعامله ارائه شده اند یا بر ایستا 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 فروشندگان سعی دارند از نوسانات بازار جهت یادگیری قیمت‌گذاری بهینه در آینده استفاده کنند. همچنین بسیاری از قراردادهای بازار مخفیانه بوده و لزوما رقبا از تراکنش های انجام شده توسط دیگران مطلع نیستند. دسته دیگری از مسایل که با روش‌های موجود قابل حل نیستند، بازی‌هایی هستند مثل پوکر، که بازیگران از دست رقیب و استراتژی او بی‌خبرند. بنابراین در چنین محیط‌هایی روش‌های طرح شده‌ی گذشته، الزاما کارایی نداشته و نیاز به الگوریتم های جدیدی برای یادگیری می باشد. دراین مقاله روش برای حل مساله‌ی غیر ایستا 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 می پردازد. سپس بازیگران استراتژی خود را به عکس‌العمل بهینه تغییر داده و به یادگیری ادامه می‌دهند. نقطه تعادل بازی بسته به زمان به روز کردن سیاست توسط عامل ها می تواند ترکیبی از تعادل نش [3] یا استکلبرگ باشد. به این ترتیب که عوامل کندتر نقش راهبر و عوامل سریعتر نقش پیرو را در بازی به عهده گیرند. این مقاله به بررسی عامل‌های قیمت‌گذار در یک بازار دو انحصاری محدود می‌شود که فروشندگان از عمل و سود رقیبان خود اطلاعی در دست ندارند. در ادامه مقاله چنین سازماندهی شده است: در بخش 2 تعاریف اولیه و مروری بر کارهای قبلی بیان می‌شود. در بخش 3 الگوریتم پیشنهادی ارائه شده است. در بخش 4 ارزیابی الگوریتم پیشنهادی و در آخر نتیجه گیری آورده شده است.

2- مفاهیم اولیه و مروری بر کارهای گذشته

یک اقتصاد عاملی به عنوان نوعی سیستم چندعامله است که می‌توان این سیستم را با بازیهای مارکوف مدل کرد[4]. بازی مارکوف در واقع یک فرآیند تصمیم گیری مارکوف چند عامله است. که در ادامه به توضیح آنها می‌پردازیم.

تعریف 1. فرآیند تصادفی مارکوف (MDP[1]) 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، γ ضریب کاهش و $T : S\times A\times S \to [0,1]$ احتمال انتقال از وضعیت جاری به وضعیت بعدی با انجام عمل a است و $R: S\times A\to \Re$ 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.

تعریف 2. 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 تک عامل هاست که تابعی از اعمال مشترک می‌باشد $S . S\times A\to \Re$ حالت بازی را مشخص می کند، و T تابع انتقال حالت است و بر اساس حالت قبلی و عملکرد بازیگران، حالت جدید بازی را تعیین می‌کند $S\times A\times S \to [0,1]$. البته R علاوه بر عملکرد بازیگران تابع حالت بازی نیز هست. بنابراین می‌توان گفت تعریف بازی‌های تصادفی بسطی بر تعریف MDP است. در واقع یکMDP 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 راه‌حل این بازی محسوب می‌شود. یک گروه از راه‌حل‌هایی که برای بیشینه سازی در این گونه بازی‌ها بکار می‌رود، استفاده از یادگیری می‌باشد.

روش‌های زیادی برای یادگیری در این محیط‌ها برای عامل‌های قیمت‌گذار بکار رفته است که تعدادی از آنها عبارتند از: روش‌های مبتنی بر تئوری بازی، روش‌های نزدیک بینانه بهینه، روش‌های پیروی اشتقاقی و روش‌های یادگیری بدون پشیمانی[5]. یکی از روش‌هایی که عموما

در تئوری بازیها مورد مطالعه قرار گرفته است، یادگیری تقویتی می‌باشد.

یادگیری تقویتی راهی برای آموزش عامل‌ها از طریق آزمون و خطاست که در محیط‌های 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 بهینه‌سازی بر اساس روش‌های تقریبی و تخمینی به هدف دست می‌یابد. انواع الگوریتم‌های یادگیری به تفصیل در[6] شرح داده شده است.

یک الگوریتم بسیار معروف یادگیری تقویتی در محیط‌های MDP، یادگیری Q می‌باشد. این روش یادگیری تقویتی، مقادیر Q را به هر زوج (حالت،عمل)=(s,a) نسبت می‌دهد. این مقدار هربار که حالت s رویت شده و عمل a انجام شود به روز می‌گردد. قانون به روز کردن آن در رابطه 1 آمده است.

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

که در رابطه‌ی فوق γ نشان می دهد، عامل چقدر نزدیک نگر بوده و یا علاقمند به پاداش‌های بلند مدت است و α نشان دهنده‌ی میزان بستگی عامل به تجربیات اخیر در مقایسه با تجربیات گذشته می‌باشد. همانطور که در رابطه 1 مشاهده می‌شود، به هر حالت نیز یک مقدار نسبت داده می‌شود که متناظر با ارزشِ پرارزش‌ترین عمل در آن حالت است.

یادگیری تقویتی در اقتصاد عاملی که موضوع مورد بحث ما می‌باشد، برای یادگیری[7] عامل‌ها به منظور تطبیق با محیط بکار می‌رود[9،8]. استفاده از یادگیری تقویتی به عنوان روشی برای یادگیری استراتژی مناسب توسط عامل‌ها در این نوع اقتصادهای عاملی، ملزم به رعایت ویژگی مارکوف در محیطی است که این عامل‌ها به تعامل در آن محیط مشغول هستند[8]. به عنوان مثال اگر بازاری دوانحصاری را در نظر بگیریم که فروشنده اول از استراتژی قیمت‌گذاری 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.)

از آنجا که در بازی‌های مارکوف محیط شامل چندین عامل می‌باشد، راه حل 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 متوالی 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 دانستن تابع 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 تصمیم گیری 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. مثلا در [10] فرآیند مارکوف با قابلیت دید جزئی برای عدم قطعیت های محیط در مورد پارامترهای مربوط به مشتریان (مثلاً نرخ ورود، تعداد واقعی مشتریان، ویژگی‌های آماری مربوط به قیمت رزرواسیون و ...) مورد بررسی قرارگرفته است. در اینگونه شرایطی که اطلاعات کاملی از محیط در دسترس نباشد، یادگیری Q نتایج خوبی می‌دهد؛ زیرا نیازی به مدل ندارد[11] 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 [12] درخت رگرسیون و در[13] شبکه‌های عصبی، بررسی شده اند.

در شرایطی که همه فروشنده‌ها (با تعداد 2 یا بیشتر) بخواهند یاد بگیرند چگونه قیمت‌گذاری کنند، مسأله تبدیل به یادگیری در سیستم‌های چندعامله [14] می‌شود که این مسأله مارکوفی نیست؛ زیرا حالت بعدی برای هر عامل، نه تنها به حالت فعلی و عمل خود که به عمل دیگر عامل ها 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 مسأله در[2] برای یادگیری Q و در[15] برای یادگیری نش[3] 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) فروشندگان از استراتژی قیمت‌گذاری که مناسب با آن شرایط باشد استفاده می‌کنند[17،16]. استفاده از یادگیری‌های نامتقارن نیز در مقالات قیمت‌گذاری بکار رفته است که در[18] می‌توان نمونه ای از آن را مورد بررسی قرار داد.

در اقتصادهای عاملی عموماً تعداد زیادی خریدار و تعدادی فروشنده وجود دارند که خریدارها می‌توانند از ربات خرید، و فروشندگان نیز می‌توانند از ربات قیمت‌گذار استفاده کنند. ربات قیمت‌گذار به فروشندگان 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 قیمت ها و دیگر پارامترهای مورد نظرشان را بررسی کرده و بهترین محصول با مناسب‌ترین قیمت ممکن را انتخاب کنند (توضیحات بیشتر در[5]). معمولاً قیمت‌های تقاضای بازار توسط دو حد آستانه (هزینه مرزی و هزینه رزرواسیون) محدود می‌شود که قیمت های تعیین شده باید در این محدوده قرار داشته باشند [19].

معمولاً 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 [10و20و21]. در سالهای اخیر نیز بررسی‌های نسبتا زیادی در مورد محیط‌های نامعین انجام شده است که این نامعین بودن در مورد تقاضا در [21و22] در نظر گرفته شده و با استفاده از یادگیری بیزین کارایی را بهبود بخشیده اند و یا در [23] با استفاده از الگوریتم‌های تکاملی، پارامترهای مربوط به قیمت گذاری خود را بهبود بخشیده اند.

اما مبحث مورد علاقه ما ربات‌های قیمت‌گذار هستند که مورد استفاده فروشندگان قرار می‌گیرند و به آنها کمک می‌کنند با تعیین یک استراتژی قیمت‌گذاری، سود بیشتری بدست آورند. در این مقاله ما به بررسی این ربات ها در مدل برترند می‌پردازیم. مدل برترند دارای جزییاتی می‌باشد که در ادامه در مورد آنها صحبت میکنیم.

مدل برترند همانطور که قبلا اشاره شد تعاملات بین فروشندگانی 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 برترند را شامل می‌شود؛ یعنی دو فروشنده نمی‌توانند تبانی داشته باشند و بر قیمت‌گذاری آن کالا با یکدیگر رقابت می‌کنند. اگر قیمتی که فروشنده1 برای کالا قرار می‌دهد را p_1 و قیمت فروشنده2 را p_2 بنامیم و تعداد کالایی که فروشنده1 می‌فروشد را q_1 و تعداد کالایی که فروشنده2 می‌فروشد را q_2 قرار دهیم، رابطه 2 و 3 بصورت زیر تعریف می‌شود:

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

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

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 i از فروش این کالا طبق رابطه 4 سود می‌برد.

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

که c_i هزینه‌ی مرزی فروشنده‌ی i است.

3- الگوریتم پیشنهادی

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 عکس‌العمل بهینه تغییر داده و به یادگیری ادامه می‌دهند. این مقاله به بررسی عامل‌های قیمت‌گذار در یک بازار دو انحصاری محدود می‌شود که فروشندگان از عمل و سود رقیبان خود اطلاعی در دست ندارند. در واقع ما در این بخش، الگوریتمی برای بیشینه سازی سود فروشندگان در بازار دوانحصاری، بدون دانستن قیمت و سود رقبا ارائه می‌دهیم و نیز در یادگیری تقویتی مربوط به بازی‌های تصادفی، با ایجاد تاخیر در به روز رسانی سیاست عامل‌ها، باعث بهبود یادگیری شده، 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 محیط-های 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 عامل ها 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 عمدتا 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 ماکزیمم کردن سود خود هستند، بنابراین به نظر می‌رسد، بهتر باشد که اپراتور ماکزیمم‌گیری 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 ماکزیمم 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 بازی های تکراری که با چند تایی مرتب $(n,A_1,\ldots,A_n,R_1,\ldots,R_n)$ تعریف می‌شود، عمل کند. هر بازیگر باید زمانی برای به روز رسانی سیاست خود داشته باشد، به طوری که هر بازیگر باید در مرتبه‌ی زمانی از دیگران متفاوت باشد. عامل‌هایی که مرتبه‌ی زمانی طولانی‌تری دارند، کندتر 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-های کندتر استراتژی خود را تغییر دهند، عامل‌های سریع‌تر بهترین پاسخ خود را نسبت به عمل رقیبان یاد گرفته باشند.

بنابراین اولین مرحله الگوریتم تخصیص یک T_i به عامل i 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 می-یابد که عامل‌های سریعتر که عمل بهینه خود را یافته‌اند مجبور به تغییر دادن آن، به خاطر تغییر رفتار عوامل کند تر شده و عمل بهینه‌ی

جدید خود را جستجو می‌کنند. در هر تکرار بازی عامل ها مقادیر Q خود را طبق رابطه 5 به روز می‌کنند:

$$Q_i(a_i) = \alpha_i Q_i(a_i) + (1-\alpha_i)\eta_i \qquad (5)$$

از آنجا که بازی، ماتریسی تکراری است، متغیر حالت در رابطه 1 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 باشند و بالعکس.

پس از گذشت زمان T_i برای هر بازیگر i، 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 6 نشان داده شده‌است:

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

این عمل، برای عامل i در T_i دوره بعد بازی ثابت خواهد بود. می‌توان اثبات کرد که وقتی کندترین عامل، تمام عمل‌های خود را امتحان کرده و بهترین عمل خود را بیابد، با فرض اینکه عوامل کندتر در نقش راهبر برای عوامل تندتر هستند، همه عامل ها 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. T_i در این مساله هر چیزی می‌تواند باشد، پس ما آن را به سادگی با 1 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 قطعا همگرا می‌شود. این زمان را $T_{conv.1}$ 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 قبلا به آنها اشاره کردیم برقرار شود. T_2 می‌تواند به ترتیب رابطه 7 تعیین گردد:

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

در ابتدای بازی، بازیگر 2 یک عمل را انتخاب کرده و آن را برای T_2 راند بازی می‌کند؛ زمانی که مطمئن هستیم بازیگر1 وقت کافی برای یافتن بهترین عکس العمل به عمل بازیگر2 داشته است. در این زمان بازیگر1 به مقادیر Q بهینه نسبت به عمل بازیگر2، و بازیگر2 به مقدار Q عمل خودش بر حسب عکس العمل بهینه بازیگر1 همگرا شده است. در این زمان عامل2، 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 می‌دهد. این فرایند آنقدر تکرار می‌شود تا همه استراتژی‌های بازیگر2 آزموده شده و عامل2 بهترین عمل خود را بر اساس عکس‌العمل رقبا بدست آورد. این نقطه، تعادل استکلبرگ با راهبری بازیگر2 و پیروی بازیگر1 خواهد بود. بعلت آزمودن، تمام استراتژی‌های بازیگر2 و انتخاب بهترین استراتژی، براساس عکس‌العمل بازیگر1 است. دقیقا مانند حالت تک‌عامله می‌توان زمانی را تعیین کرد که مطمئن باشیم بازی در آن زمان $T_{conv.2}$ به نقطه تعادل خود رسیده است.

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

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

این روش، به نقطه تعادل استکلبرگ که در آن بازیگران کندتر، راهبر بازیگران سریع‌تر هستند همگرا می‌گردد. بنابراین عوامل کندتر سود بلندمدت بیشتری بدست آورده و در عوض بازیگران سریع‌تر زودتر با تغییرات تطبیق یافته و سود کوتاه‌مدت بیشتری کسب می‌کنند. همانطور که ملاحظه شد، بازیگران نیازی به رؤیت استراتژی رقیبان خود ندارند و فقط سود خود را رؤیت می‌کنند. این مهمترین ویژگی الگوریتم می‌باشد. شبه کد این الگوریتم در شکل 1 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:
$$\begin{cases} a_i = arg\ \underset{a_i}{\text{Max}}\ Q_i(a_i) & probability\ 1-\epsilon \\ a_i = rand(A_i) & probability\ \epsilon \end{cases}$$
3. Decrease ϵ

شکل (1) : شبه‌کد الگوریتم برای بازی‌های ماتریسی تکراری

4- پیاده‌سازی و نتایج آزمایش‌ها

برای بررسی مدل پیشنهادی از مدل برترند برای بازار استفاده شده است. به این صورت که فرض می‌شود یک نوع کالای فاسد نشدنی وجود دارد که دو فروشنده‌ی انحصاری آن، برای بدست آوردن سود بیشتر رقابت می‌کنند. در واقع ما به بررسی مدل ربات خرید پرداختیم. در مدل ما فرض می‌شود مطابق با یک تابع تقاضا، خریدارانی وجود دارند که با جستجوی کمترین قیمت به خرید آن کالا می‌پردازند. پس از اینکه خریداران فروشنده را برای خرید انتخاب کردند، سودی از این خرید نسیب فروشنده می‌شود. قسمت مورد توجه در مقاله ما اینست که فرض می‌شود هیچ کدام از فروشنده ها از این تابع تقاضا و همچنین قیمت و پاداش فروشنده دیگر اطلاعی ندارد.

فرض می‌کنیم عامل1 عامل کندتر است که مرتبه‌ی زمانی آن برابر 1000 واحد زمانی درنظر گرفته شده و عامل شماره2 یعنی عامل

سریعتر در هر 1000 واحد که استراتژی عامل 1 ثابت است به یادگیری می‌پردازد تا بهترین قیمت را نسبت به قیمت عامل کندتر انتخاب کند. هر 1000 گام زمانی، عامل کندتر (راهبر) قیمت خود را تغییر می‌دهد و باز عامل سریعتر یادمی‌گیرد بهترین پاسخ خود را انتخاب کند. از آنجاییکه در این 1000 گام زمانی، عامل کندتر قیمت خود را ثابت نگه می‌دارد، محیط برای عامل سریعتر ایستا بوده و میتواند از یادگیری Q تک عامله برای یادگیری خود استفاده کند. تابع سود به این صورت محاسبه می‌شود که میزان درآمد حاصل از تعداد کالاهای فروخته شده را بدست آورده و به همان تعداد فروش، هزینه مرزی، از این مقدار کم می‌شود تا سود خالص محاسبه شود. هزینه مرزی در پیاده سازی ما برابر با 1 در نظر گرفته شده است یعنی داریم $c_1=c_2=1$. دامنه استراتژی بازیگران 1 تا 25 است یعنی قیمت‌های مجاز برای هر یک از عامل ها می‌باشد که آن را در بازه‌های واحد، گسسته کرده‌ایم. به نظر می‌رسد الگوریتم احتیاج به 50 دوره 100 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 دقیقتر، به مقایسه‌ی نتایج تئوری محاسبات برای نقطه‌ی تعادل استکلبرگ با نتایج بدست آمده پرداختیم. همانطور که در رابطه‌ی2 و3 و4 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:

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

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

بنابراین طبق رابطه‌ی 4 داریم:

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

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

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

$$\frac{\partial \Pi_2}{\partial p_2} = q_2 + (p_2 - c_2)\frac{\partial q_2}{\partial p_2}$$
$$= 10p_1 - 10p_2 + 100 - 10p_2 + 10c_2 = 0$$

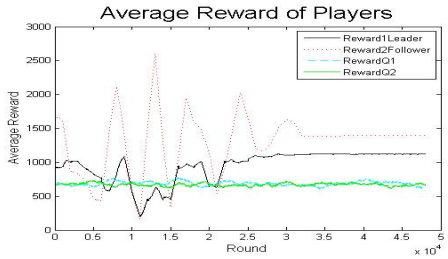
$$\Rightarrow \quad p_2^* = 5 + \tfrac{1}{2} p_1 + \tfrac{1}{2} \qquad (13)$$

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

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

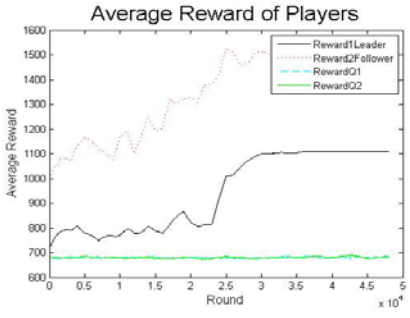
لازم است به این نکته اشاره شود که در مدل بازار ما، فرض می‌کنیم هیچ یک از فروشندگان این اطلاعات را ندارند و قادر به محاسبه‌ی نقطه‌ی تعادل استکلبرگ نیستند و فقط با یادگیری Q به نتایجی مشابه به تعادل استکلبرگ می‌رسند.

شکل2 مقایسه‌ی بازاری با 2 عامل فوق را با بازاری شامل 2 فروشنده که هر دو به سادگی از یادگیری Q بدون تاخیر 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 (2) : نمودار میانگین سود فروشنده ها با اجرای الگوریتم راهبر-پیرو در مقایسه با عامل‌های الگوریتمQ در یک اجرا

برای اطمینان از عملکرد غیر تصادفی الگوریتم آن را در 100 اجرای متوالی نیز آزموده و میانگین مقدار را با تنظیمات قبلی در شکل 3 نشان داده ایم.



شکل (3) : نمودار میانگین سود فروشنده ها با اجرای الگوریتم راهبر-پیرو در مقایسه با عامل‌های الگوریتمQ در 100 اجرا

5- 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 قیمت‌گذاری 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 عدم‌قطعیت‌ها در محیط‌های پیچیده که استفاده از یادگیری تقویتی تک عامله میسر نیست، مشکل می‌نماید و اکثر بررسی عدم‌قطعیت‌ها در زمینه تابع درخواست و یا ورود مشتریان به بازار بوده است. همچنین مشاهده کردیم توجه اندکی معطوف به نا-معین بودن پاداش‌ها و حتی اعمال رقیبان شده است و بیشتر نامعین

بودن محیط را مربوط به مشتریان مورد بررسی قـرار داده انـد. در ایـن مقاله با فرض نامشخص بودن عمل رقیب و پاداش او برای یک عامل به بررسی بازاری دو انحصاری پرداختیم و با ایستا کردن محیط برای بازه-های زمانی ثابتی به عامل دیگر فرصت یادگیری داده و مشاهده کـردیم که با محاسبات انجام شده مشاهده کردیم که در این صـورت بـازار بـه 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 multi-agent 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.), *Unpublished survey. http://robotics. stanford. edu/shoham*, 2003.

[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. Papayiannis, 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.

[1] Markov Decision Process