



یافتن استراتژی غالب در بازی‌های YADGIR با استفاده از اتوماتاهای Blotto

سعید شیری

دانشکده مهندسی کامپیوتو و فناوری اطلاعات

دانشگاه صنعتی امیرکبیر، تهران، ایران

shiry@aut.ac.ir

محمد رضا میبدی

دانشکده مهندسی کامپیوتو و فناوری اطلاعات

دانشگاه صنعتی امیرکبیر، تهران، ایران

mmeybodi@aut.ac.ir

فرناز ابطحی

دانشکده مهندسی کامپیوتو و فناوری اطلاعات

دانشگاه صنعتی امیرکبیر، تهران، ایران

abtahi@aut.ac.ir

در آن، بازیکنان باید به طور همزمان منابع محدودی را بین چندین کار یا شیء توزیع نمایند و امتیاز هر بازیکن برابر مجموع امتیازی خواهد بود که از هر کار یا شیء به دست می‌آورد.

بازی‌های Blotto براساس یک شخصیت خیالی به نام Colonel Blotto نامگذاری شده‌اند که در جنگ وظیفه توزیع بهینه سربازان بین N میدان مبارزه را به عهده داشته است، با دانستن این‌که: ۱) در هر میدان، هر کدام از طرفین جنگ که سرباز بیشتری به آن محل اختصاص داده است پیروز خواهد شد، ۲) هیچ‌یک از طرف اطلاع ندارند که رقیب چه تعداد سرباز را به هر میدان اختصاص می‌دهد، ۳) طرفی که در اکثر میدان‌های جنگ بر حرف پیروز شود، در کل برنده جنگ خواهد بود.

أنواع مختلفی از بازی‌های Blotto وجود دارد. یکی از نمونه‌های ممکن که دارای دو بازیکن می‌باشد به این صورت است که هر یک از بازیکن‌ها سه عدد را روی گذاشتند، به طوری که مجموع این سه عدد برابر عدد معینی شود. این کار بدون اطلاع از اعدادی که طرف مقابل می‌نویسد صورت می‌گیرد. فرض می‌کنیم بازیکن‌ها مجاز به نوشتن جایگشت‌های مختلف سه عدد مشخص نمی‌باشند. مثلاً (۱،۲،۳)، (۱،۲،۴) و (۱،۳،۴) دو انتخاب به حساب نمی‌آیند. برای محاسبه امتیاز هر بازیکن، در صورتی که بیش از یک عدد از سه عدد بازیکن، از یکی از سه عددی که حریف نوشته بزرگتر باشد، یک امتیاز مثبت به بازیکن تعلق خواهد گرفت. در غیر این صورت اگر کمتر از دو عدد از اعداد بازیکن دارای این ویژگی باشد، بازیکن یک امتیاز منفی می‌گیرد. در صورتی که تمام اعداد بازیکن با اعداد رقیب مساوی باشد، بازیکن هیچ امتیازی دریافت نمی‌کند. اگر مجموع سه عدد باشد برابر ۶ شود، با توجه به قانون بازی، جدول (۱) ماتریس نتیجه برای بازیکن اول را نشان می‌دهد:

جدول (۱): ماتریس نتیجه برای بازیکن اول در یک نمونه بازی Blotto

(۱,۱,۴)	(۱,۲,۳)	(۲,۲,۲)
(۱,۱,۴)	۰	-۱
(۱,۲,۳)	۱	-۱
(۲,۲,۲)	۱	۰

در ماتریس بالا، سطرها اعمال بازیکن اول و ستون‌ها اعمال بازیکن دوم را نشان می‌دهند. از آن جایی که مجموع امتیازات دو بازیکن برابر با صفر است، ماتریس نتیجه بازیکن دوم نیز مشابه همین ماتریس است، تنها با این تفاوت که علامت امتیازات برای بازیکن دوم معکوس خواهد

چکیده: در این مقاله، رویکردی مبتنی بر اتوamatاهای YADGIR برای یافتن استراتژی غالب در بازی‌های Blotto ارائه می‌گردد. اهمیت این دسته از بازی‌ها در تئوری بازی از دو جهت است. اولاً این بازی‌ها در دنیای واقعی برای مدل‌سازی فرآیندهایی به کار می‌روند که در آن‌ها برای غلبه بر حرف، نیاز به توزیع بهینه منابع محدود بین چندین کار وجود دارد. ثانیاً رویکردی که در این بازی‌ها برای یافتن استراتژی غالب مورد استفاده قرار می‌گیرد را می‌توان برای مدل کردن هر فرآیند چندعامله رقابتی دیگر که دارای استراتژی غالب برای هر یک از عامل‌ها می‌باشد به کار برد. با دانستن استراتژی غالب می‌توان تضمین کرد که عامل، همواره سودی بیشتر یا مساوی با سایر عامل‌ها به دست خواهد آورد. در روش پیشنهادی در این مقاله، هر یک از بازیکنان دارای یک اتوamatای YADGIR می‌باشد که از آن، برای یادگیری و تصمیم‌گیری در مرور انتخاب اعمال کمک می‌گیرد. آزمایشات انجام‌شده نشان می‌دهند که با استفاده از این روش، استراتژی بازیکنان مجهز به اتوamatای YADGIR به تدریج به استراتژی غالب همگرا شده و این بازیکنان قادر به یافتن بهترین حالت تقسیم منابع و برد در بازی خواهند بود.

واژه‌های کلیدی: بازی‌های Blotto، یادگیری تقویتی، اتوamatای YADGIR، استراتژی غالب.

۱ - مقدمه

در تئوری بازی، غلبه هنگامی روی می‌دهد که یک استراتژی برای یک بازیکن، بدون توجه به این که رقبا چگونه بازی کنند، از سایر استراتژی‌ها بهتر بوده و سود بیشتری را نصیب آن بازیکن نماید. این استراتژی، استراتژی غالب نامیده می‌شود [۳]. این مفهوم در بسیاری از سیستم‌های چندعامله رقابتی، بازی‌ها و مسائل دنیای واقعی به یافتن مناسب‌ترین راه حل مسئله کمک می‌کند، زیرا در صورت وجود استراتژی غالب، بازیکن قادر است در تمام تعادل‌های Nash بازی، آن استراتژی را اجرا نماید و اطمینان داشته باشد که در بدترین شرایط نتیجه‌ای برابر با رقبای خود به دست خواهد آورد.

گروهی از بازی‌های رقابتی که در مبحث تئوری بازی مطرح بوده و با استفاده از آن‌ها به خوبی می‌توان مفهوم استراتژی غالب را مورد مطالعه قرار داد، بازی‌های Blotto می‌باشند [۱,2,8,9,10,11]. این بازی‌ها، دسته‌ای از بازی‌های ماتریسی جمع‌صفر را شامل می‌شوند که

شده برای حل مسئله تخصیص بهینه منابع در بازی‌های Blotto را بر می‌شماریم.

۲- اتوماتای یادگیر

اتوماتای یادگیر [4,5,6,7]، ماشینی است که می‌تواند تعدادی متنه ای عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی می‌شود، نتیجه ارزیابی در قالب سیگنالی مثبت یا منفی به اتوماتا داده می‌شود و اتوماتا از این پاسخ در انتخاب عمل بعدی تأثیر می‌گیرد. هدف نهایی این است که اتوماتا یاد بگیرد تا از بین اعمال خود، بهترین عمل را انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداقل برساند. کارکرد اتوماتای یادگیر در تعامل با محیط، در شکل (۱) مشاهده می‌شود.



شکل(۱): ارتباط بین اتوماتای یادگیر و محیط [5]

محیط را می‌توان توسط سه‌تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودی‌ها، $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه خروجی‌ها، و $c \equiv \{c_1, c_2, \dots, c_s\}$ مجموعه احتمال‌های جریمه می‌باشد. هرگاه β مجموعه‌ای دواعضوی باشد، محیط از نوع P است. در چنین محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته می‌شود. در محیط از نوع Q ، $\beta(n)$ می‌تواند به‌طور گسته یک مقدار از مقادیر محدود در فاصله $[0, 1]$ را اختیار کند و در محیط از نوع K $\beta(n)$ متغیر تصادفی در فاصله $[0, 1]$ است. c_i احتمال این‌که عمل α_i نتیجه نامطلوب داشته باشد می‌باشد. در محیط ایستا، مقادیر c_i بدون تغییر می‌مانند، حال آن‌که در محیط غیرایستا این مقادیر در طی زمان تغییر می‌کنند. اتوماتای یادگیر با ساختار ثابت توسط پنج-تایی $\{\alpha, \beta, F, G, \phi\}$ نشان داده می‌شود که در آن، $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عمل‌های اتوماتا، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_s\}$ مجموعه ورودی‌های اتوماتا، $\phi(n) \equiv \{\phi_1, \phi_2, \dots, \phi_k\}$ وضعیت‌های داخلی اتوماتا در لحظه n ، $F: \phi \times \beta \rightarrow \phi$ تابع تولید وضعیت جدید اتوماتا و $G: \phi \rightarrow \alpha$ تابع خروجی می‌باشد که وضعیت کنونی اتوماتا را به خروجی بعدی می‌نگارد. اتوماتای یادگیر با ساختار متغیر را می‌توان توسط چهارتایی $\{\alpha, \beta, p, T\}$ نشان داد که $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه اعمال اتوماتا، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_s\}$ مجموعه ورودی‌های اتوماتا، $p = \{p_1, \dots, p_r\}$ بردار احتمال انتخاب هریک از عمل‌ها و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می‌باشد. الگوریتم زیر یک نمونه از الگوریتم‌های یادگیری خطی است. فرض می‌کنیم عمل α در مرحله n انتخاب شود.

شد. با توجه به ماتریس نتایج، بهترین استراتژی (تعادل Nash) در اینجا (۲,۲) می‌باشد و انتخاب آن، همواره منجر به دریافت امتیاز بیشتر یا مساوی با حریف می‌گردد.

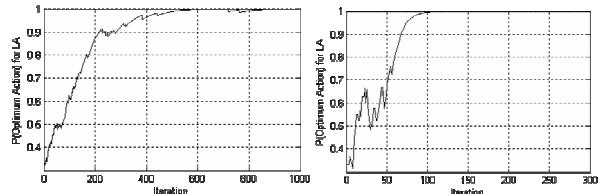
از بازی‌های Blotto برای مدل‌سازی برخی از مسائل دنیای واقعی استفاده شده است. برای مثال در [10]، این بازی برای مدل‌کردن انتخابات سال ۲۰۰۰ آمریکا که یکی از نزدیک‌ترین رقابت‌ها در تاریخ انتخابات ریاست جمهوری این کشور بوده به کار رفته است. همچنین در [11]، از این بازی در مبارزه با تروریسم استفاده شده و این کار با محاسبه احتمال موفقیت تروریست در حمله خود، از طریق مدل‌سازی حمله و دفاع توسط بازی Blotto صورت گرفته است.

در بازی بالا و یا در هر فرآیند رقابتی دیگر، بسیار مطلوب است که بتوان استراتژی‌های غالب را مشخص کرد. بدین منظور می‌توان از تکنیک‌های یادگیری استفاده نمود. در این‌گونه بازی‌ها، بازیکنان هم‌زمان در حال یادگیری می‌باشند و سود هر عامل، علاوه بر اعمال خود او به اعمال سایر عامل‌ها نیز وابسته است. به همین دلیل، محیط بازی غیرقطعی بوده و مناسب‌ترین روش برای یادگیری، استفاده از تکنیک‌های مبتنی بر یادگیری تقویتی می‌باشد. در بازی‌های ماتریسی، یادگیری از طریق تکرار بازی به دفعات متعدد صورت می‌گیرد. در هر تکرار، هریک از بازیکنان عملی را انتخاب کرده و انجام می‌دهد. سپس با توجه به عمل خود و اعمال سایر بازیکنان، امتیازی به بازیکن تعلق می‌گیرد. با توجه به این امتیاز، بازیکن یاد می‌گیرد که عمل انجام شده تا چه حد مناسب بوده است و نتیجه این یادگیری را در دفعات بعدی اجرای بازی به کار می‌برد.

در این مقاله، رویکردی مبتنی بر اتوماتای یادگیر برای پیاده‌سازی بازی‌های Blotto ارائه می‌گردد. اتوماتای یادگیر که بر مبنای یادگیری تقویتی عمل می‌کند، ابزاری مناسب برای کمک به فرآیند یادگیری عامل‌ها در سیستم‌های چندعمله از جمله سیستم‌های رقابتی است. در رویکرد پیشنهادی، هریک از بازیکنان در بخش یادگیری خود دارای یک اتوماتای یادگیر می‌باشد. اتوماتا به بازیکن کمک می‌کند تا در هر مرحله، با توجه به نتایجی که تا به حال به دست آمده، بهترین عمل را انتخاب نماید و سپس براساس پاداش یا جریمه‌ای که در اثر انجام این عمل دریافت می‌کند، احتمال انجام آن را در مراحل بعدی بازی افزایش یا کاهش دهد. آزمایشات انجام‌شده نشان می‌دهند که با استفاده از این روش، استراتژی بازیکنان مجهز به اتوماتای یادگیر به تدریج به استراتژی غالب همگرا می‌گردد. استراتژی غالب که در بازی‌های Blotto به معنای بهترین حالت تخصیص منابع محدود به چندین کار می‌باشد، همواره بهترین نتیجه را برای بازیکن دریی خواهد داشت.

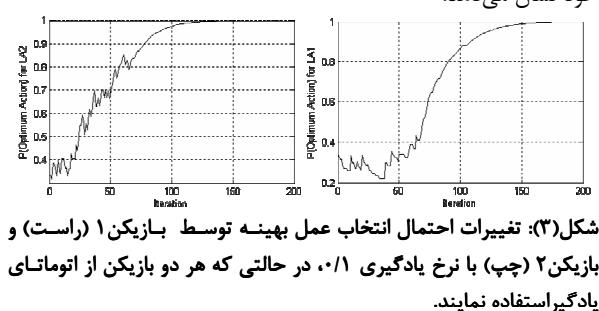
ادامه مقاله بدین صورت سازماندهی شده است. در بخش ۲ مروری بر اتوماتای یادگیر و الگوریتم یادگیری مورد استفاده در آن خواهیم داشت. بخش ۳ شامل آزمایشات انجام‌شده برای بررسی عملکرد روش پیشنهادی می‌باشد. در بخش ۴ نتایج حاصل از به کار گیری روش ارائه-

که در نمودارهای شکل (۲) مشاهده می‌شود، احتمال انتخاب عمل بهینه توسط اوتوماتا به سرعت به یک می‌رسد و استراتژی بازیکن به استراتژی غالب همگرا می‌گردد. مشاهده می‌کنیم که سرعت همگرایی و همواری نمودار به پارامتر پاداش، یعنی a وابسته است.



شکل (۲): تغییرات احتمال انتخاب عمل بهینه توسط بازیکن دارای اوتوماتای یادگیر با نرخ یادگیری $0.1/0.1$ (راست) و $0.01/0.1$ (چپ)، در حالتی که فقط یکی از بازیکنان از اوتوماتای یادگیر استفاده نماید.

در حالت دوم، در ساختار هر دو بازیکنان از اوتوماتای یادگیر استفاده می‌کنیم. در این وضعیت به دلیل این که بازیکنان به طور همزمان در حال یادگیری هستند، همگرایی با سرعت کمتری صورت می‌گیرد؛ زیرا در هر بار تکرار بازی، هر بازیکن سعی می‌کند بهترین رفتاری که تا آن لحظه یاد گرفته است را از خود نشان دهد و انتخاب عملی با سود بیشتر را برای حریف مشکل می‌سازد. این مسئله در نمودارهای شکل (۳) قابل مشاهده می‌باشد. در این حالت نیز با کاهش پارامتر پاداش، همگرایی رفتار بازیکنان، کنترل شده ولی نوسانات کمتری از خود نشان می‌دهد.



شکل (۳): تغییرات احتمال انتخاب عمل بهینه توسط بازیکن ۱ (راست) و بازیکن ۲ (چپ) با نرخ یادگیری $0.1/0.01$ در حالتی که هر دو بازیکن از اوتوماتای یادگیر استفاده نمایند.

بازی مورد استفاده در آزمایش بالا، فقط دارای یک استراتژی غالب و تعادل *Nash* بوده و تعداد اعمال در آن محدود به سه عمل می‌باشد. هنگامی که مسئله بزرگتر بوده و یا قانون بازی متفاوت باشد، مجموعه اعمال نیز بزرگتر خواهد بود و ممکن است چندین استراتژی غالب برای بازیکن وجود داشته باشد. برای بررسی تأثیر افزایش اندازه مجموعه اعمال و همین‌طور افزایش تعداد استراتژی‌های غالب، نوع دیگری از بازی را مطرح می‌نماییم. در این بازی نیز که بسیار شبیه به بازی قبل می‌باشد، دو بازیکن سه عدد را روی کاغذ می‌نویسند، به طوری که مجموع آن‌ها برابر با عدد معینی شود. این کار بدون اطلاع از انتخاب حریف صورت می‌گیرد. تفاوت این بازی با بازی قبل در این است که اولاً بازیکن‌ها مجاز به نوشتن جایگشت‌های مختلف سه عدد می‌باشند؛ به این معنا که مثلًا $(1,1,3)$ و $(3,1,1)$ دو انتخاب متفاوت بهشمار می‌روند. ثانیاً در بازی جدید، هریک از سه عدد با عددی در همان جایگاه

- پاسخ مطلوب از محیط

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

- پاسخ نامطلوب از محیط

$$\begin{aligned} p_i(n+1) &= (1-b)p_i(n) \\ p_j(n+1) &= (b/r-1) + (1-b)p_j(n) \quad \forall j \neq i \end{aligned} \quad (2)$$

در روابط بالا، a پارامتر پاداش و b پارامتر جرمیه می‌باشد. با توجه به مقادیر a و b سه حالت را می‌توان در نظر گرفت: اگر b و a با هم برابر باشند، الگوریتم را L_{RP} ، هنگامی که a از b خیلی کوچکتر باشد، الگوریتم را L_{RRI} و اگر b مساوی صفر باشد آن را L_{RI} می‌نامیم.

۳- یافتن استراتژی غالب در بازی Blotto با استفاده از اوتوماتای یادگیر

هدف از این بخش یادگیری استراتژی غالب در بازی Blotto است، به‌طوری که بازیکن با اجرای آن، تحت هر شرایطی و بدون توجه به روش بازی حریف در بازی پیروز شود. برای پیاده‌سازی بازی از نمونه‌ای که در مقدمه توضیح داده شد، یعنی بازی اراده می‌دهیم، بازیکنان روبرکردی که در این بخش برای حل این بازی اراده می‌دهیم، بازیکنان از اوتوماتای یادگیر در فرآیند یادگیری و تصمیم‌گیری خود برای انتخاب بهترین عمل کمک می‌گیرند.

در این روش، هر بازیکن دارای یک اوتوماتای یادگیر با ساختار متغیر می‌باشد. همان‌طور که در بخش ۲ گفته شد، اوتوماتای یادگیر با ساختار متغیر با چهارتایی $\{\alpha, \beta, p, T\}$ نشان داده می‌شود. برای مدل کردن این بازی با اوتوماتای یادگیر، α برابر با مجموعه اعمال هر عامل، یعنی $\{(1,1), (1,2), (1,3), (2,2), (2,3), (3,2), (3,3)\}$ قرار داده می‌شود. β برابر با امتیازی در نظر گرفته می‌شود که بازیکن در هر مرحله دریافت می‌کند. p بردار احتمال اعمال اوتوماتا است که در ابتدا مقادیر آن به ازاء تمام اعمال، یکسان در نظر گرفته می‌شود. T نیز نشان‌دهنده الگوریتم یادگیری می‌باشد که در پیاده سازی این بازی، از یادگیری خطی مطابق با روابط (۱) و (۲)، و با شمای L_{RP} استفاده شده است.

بازی به صورت تکراری و با توجه به قانون بازی و ماتریس نتیجه در دفعات متعدد اجرا می‌گردد. در هر بار اجرای بازی، ابتدا اوتوماتای یادگیر مربوط به هر بازیکن، مستقلًا یک عمل از مجموعه اعمال خود را با توجه به بردار احتمال انتخاب می‌کند. پس از اجرای اعمال توسط بازیکنان، با توجه به ماتریس نتایج امتیازی به هریک تعلق می‌گیرد. با توجه به این امتیاز و با استفاده الگوریتم یادگیری اوتوماتای یادگیر، بردار احتمال هر دو اوتوماتا به روز می‌شود. این روند مرتباً تکرار می‌شود تا جایی که بردار احتمال اوتوماتاها به پایداری برسد.

برای انجام آزمایش دو حالت را در نظر می‌گیریم. ابتدا حالتی را بررسی می‌کنیم که یکی از بازیکنان، دارای اوتوماتای یادگیر است و بازیکن دیگر اعمال خود را به طور تصادفی انتخاب می‌نماید. همان‌طور

یادگیری استراتژی غالب توسط آنها و همگرایی سریع رفتار آنها به رفتار بهینه می‌گردد. این بازی را می‌توان مدلی از بسیاری از فرآیندهای دنیای واقعی که در آنها نیاز به تقسیم بهینه منابع محدود بین چندین کار با محل وجود دارد درنظر گرفت. بهاین ترتیب روش ارائه شده، در بسیاری از کاربردهای بزرگ و برای حل مسائل مطرح در سیستم‌های چندعامله رقابتی که دارای استراتژی غالب می‌باشند قابل استفاده بوده و دستیابی به حداکثر سود ممکن را تضمین می‌نماید.

۵- مراجع

- 1- Roberson, B., "The Colonel Blotto Game", Journal of Economic Theory, vol. 29, no. 1, pp. 1-24, Springer Berlin, 2006.
- 2- Golman, R., Page, S. E., "General Blotto: Games of Allocative Strategic Mismatch", Center for Complex Systems, University of Michigan, 2006.
- 3- Tuyls, K., Nowe, A., "Evolutionary Game Theory and Multi-Agent Reinforcement Learning", the Knowledge Engineering Review, vol. 20, pp. 63-90, 2005.
- 4- Nowe, A., Verbeeck, K., Peeters, M., "Learning Automata as a Basis for Multi-Agent Reinforcement Learning", Proceedings of First International Workshop on Learning and Adaptation in Multi-Agent Systems (LAMAS), Utrecht, the Netherlands, 2005, pp. 71-85, 2006.
- 5- Shirazi, M.R., Meybodi, M.R. "Application of Learning Automata to Cooperation in Multi-Agent Systems", Proceedings of First International Conference on Information and Knowledge Technology (IKT2003), pp. 338-349, 2003.
- 6- Shirazi, M. R., Meybodi, M. R., "Solving Iterated Prisoner's Dilemma Using Learning Automata", Proceedings of First Iranian Conference on Mechatronics Engineering, ICME2003, pp. 349-362, Qazvin, Iran, 2003.
- 7- Verbeek, K., Nowe, A., Peeters, M., Tuyls, K. "Multi-Agent Reinforcement Learning in Stochastic Single and Multi-Stage Games", Lecture Notes in Computer Science, Springer Berlin, vol. 3394, pp. 275-294, 2005.
- 8- Banerjee, B., Peng, J., "Convergence of No-Regret Learning in Multi-Agent Systems", Proceedings of the First International Workshop on Learning and Adaptation in Multi-Agent Systems (LAMAS), Utrecht, the Netherlands, 2005.
- 9- Bowling, M., Veloso, M. M., Existence of Multi-Agent Equilibria with Limited Agents, Technical Report CMU-CS-02104, Computer Science Department, Carnegie Mellon University, 2002.
- 10- Merolla, J., Munger, M., Tofias, M., "Lotto, Blotto, or Frontrunner: The 2000 U. S. Presidential Election and the Nature of Mistakes", Duke University, 2003.
- 11- Powers, M. R., Shen, Z., Colonel Blotto in the War on Terror: Implications for Event Frequency, American Risk and Insurance Association (ARIA) Annual Meeting, Washington D.C., August 2006.

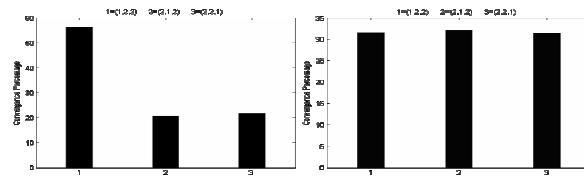
مقایسه می‌شود؛ یعنی عدد اول بازیکن با عدد اول حریف، عدد دوم او با عدد دوم حریف و عدد سوم او با عدد سوم حریف مقایسه شده و در این مقایسه‌ها اگر هریک از اعداد بازیکن از اعداد حریف بزرگتر باشد، بهاره آن عدد یک امتیاز مثبت و اگر کوچکتر باشد یک امتیاز منفی به بازیکن تعلق می‌گیرد. در صورت تساوی، بازیکن امتیازی برای آن عدد دریافت نمی‌کند. درنهایت امتیازی که به بازیکن داده می‌شود، برابر با مجموع امتیازی خواهد بود که از هریک از سه عدد به دست می‌آورد.

اگر مجموع سه عدد می‌باشد برابر با ۵ شود، ماتریس نتیجه برای بازیکن اول به صورت جدول (۲) خواهد بود. همان‌طور که در بازی قبل گفته شد، به دلیل جمع صفر بودن بازی‌های Blotto، ماتریس نتیجه بازیکن دوم مشابه همین ماتریس می‌باشد، اما علامت امتیازات معکوس می‌گردد.

جدول(۲): ماتریس نتیجه برای بازیکن اول در یک نمونه بازی Blotto

	(۱,۱,۳)	(۱,۲,۲)	(۱,۳,۱)	(۲,۱,۲)	(۲,۲,۱)	(۳,۱,۱)
(۱,۱,۳)	•	•	•	•	-1	•
(۱,۲,۲)	•	•	•	•	•	1
(۱,۳,۱)	•	•	•	-1	•	•
(۲,۱,۲)	•	•	1	•	•	•
(۲,۲,۱)	1	•	•	•	•	•
(۳,۱,۱)	•	-1	•	•	•	•

با توجه به جدول بالا، هر سه عمل (۱,۲,۲)، (۲,۱,۲) و (۲,۲,۱) تعادل Nash و استراتژی غالب برای این بازی می‌باشند. اگر احتمال انتخاب اعمال در ابتدای بازی یکسان درنظر گرفته شوند، منطقی است که در دفعات متعدد اجراء، احتمال همگرایی به هریک از سه استراتژی تقریباً یکسان باشد. اما اگر احتمال اولیه یک استراتژی را مثلاً با توجه به اطلاعات اولیه‌ای که درباره دامنه مسئله در دست داریم، به میزان جزئی افزایش دهیم، احتمال همگرایی به آن استراتژی بیشتر خواهد شد. نمودارهای شکل (۴) این مسئله را نشان می‌دهند.



شکل(۴): درصد دفعات همگرایی به هریک از سه استراتژی غالب در ۱۰۰۰۰ بار اجرای الگوریتم با احتمال اولیه برابر برای تمام اعمال (راست) و پس از افزایش جزئی احتمال اولیه عمل (۱,۲,۲) (چپ)

۴- نتیجه گیری

در این مقاله، روشی مبتنی بر اتوماتی یادگیر برای یافتن استراتژی غالب در بازی‌های Blotto ارائه گردید. با توجه به نتایج آزمایشات انجامشده، استفاده از اتوماتی یادگیر در ساختار بازیکنان منجر به