به کارگیری اتوماتای یادگیر[1] در سیستمهای چندعاملهٔ همکار

محمدرضا آیت اله زاده شیرازی محمدرضا میبدی

آزمایشگاه محاسبات نرم، دانشکده مهندسی کامپیوتر و فنآوری اطلاعات

دانشگاه صنعتی امیرکبیر

ashirazi@ce.aut.ac.ir

چکیده

یکی از مشکلاتی که الگوریتمهای یادگیری تقویتی در سیستمهای چندعامله با آن مواجه هستند وجود چندین نقطهٔ موازنه می باشد.در یادگیری تقویتی سیستمهای چندعامله، بیشترین تمرکز برروی تضمین همگرایی الگوریتمهای یادگیری به نقطهٔ موازنهٔ مطلوب است. این روشها با این مشکل مواجه هستند که عاملها باید انتخاب نقطه موازنه را با یکدیگر هماهنگ کنند. در این مقاله، رفتار اتوماتای یادگیر به عنوان استراتژی تصمیم گیری عاملها در سیستمهای چندعامله به منظور دستیابی به رفتاری هماهنگ مورد بررسی و ارزیابی قرار می گیرد. بدین منظور همگرایی الگوریتم یادگیری اتوماتای یادگیر به عنوان استراتژی تصمیم گیری عاملهایی که در ساختار یک بازی همکاری فعالیت می کنند، در دو حالت وجود یادگیرنده های مستقل و یادگیرنده های مشترک ارزیابی و کارآیی آن با الگوریتم یادگیری Q مقایسه شده است. نتایج حاصل نشان می دهند که استراتژی اتوماتای یادگیر با سرعت خوبی به عمل مشترک بهینه همگرا می شود. دیده می شود که اتوماتای یادگیر در مقایسه با یادگیری Q با سرعت بیشتری احتمالات عمل مشترک بهینه را یاد می گیرد. در رابطه با یادگیرنده های مشترک نیز که بر اعمال یکدیگر نظارت دارند، اتوماتای یادگیر و یادگیری Q در هردو حالت کارآیی یکسانی از خود نشان می دهند. همچنین، در این مقاله، ایدهٔ به کارگیری اتوماتای یادگیر به منظور پیاده سازی استراتژی مذاکره در عاملهای مذاکره کننده ارائه و مورد بررسی قرار می گیرد.

واژه های کلیدی: استراتژی مذاکره- اتوماتای یادگیر با ساختار متغیر- سیستم چندعامله- همکاری - یادگیری تقویتی - یادگیری Q.

۱- مقدمه

شرکت در جامعه ای از عاملها بایستی هم محدودکننده و هم تقویت کننده باشد. محدود کننده از این نظر که عاملها باید در جامعه حضور داشته باشند و تقویت کننده از این نظر که حضور و شرکت در جامعه، منابع و فرصتهایی را در اختیار عامل قرار می دهد که به تنهایی این عامل نمی توانست به آنها دست یابد. هماهنگی[2] کلید دستیابی به این مقصود است. موضوع هماهنگی در سیستمهای چندعامله[3] یکی از مسائل کلیدی در این حوزه است. هماهنگی فرآیندی است که توسط آن عامل به منظور سعی در راه عمل یکپارچهٔ جامعهٔ عاملها و تضمین این یکپارچگی دربارهٔ اعمال محلی خودش و اعمال (پیش بینی شدهٔ) دیگران استنتاج می کند[۸,۹]. همکاری[4] هماهنگی میان عاملهایی با هدف مشترک است در حالی که مذاکره هماهنگی میان عاملها رقابتی می باشد[۹] . مذاکره خودکار نوعی تعامل است که میان عاملهایی با اهداف یا مقاصد متفاوت یا حتی مشترک صورت می گیرد[۷] .

یادگیری در مسئلهٔ هماهنگی در سیستمهای چندعامله کاربردهای زیادی یافته است[۲,۳,۱۵] . در این بین از یادگیری تقویتی به شکل زیادی در سیستمهای چندعامله استفاده می شود. استفاده از یادگیری تقویتی به عنوان ابزاری برای دستیابی

---

[1] Learning Automata
[2] coordination
[3] Multi-agent systems
[4] cooperation

به رفتار هماهنگ به خاطر عمومیت و استحکامی[5] کـه دارد، بسـیار جـذاب اسـت[۱,۲,۳,۶,۱۵,۱۸] . هـدف عاملهـایی کـه در محیطهای پویا عمل می کنند، این است که تصمیمهای بهینه بگیرند. اگر عاملها از پاداشهای متناظر با اعمال مختلف مشترکی که در محیط انجام می دهند، آگاه نباشند، انتخاب عمل مشکل می شود. یادگیری بـا تنظـیم انتخـاب عمـل عاملهـا براسـاس اطلاعات جمع آوری شده در طی زمان چنین مقصودی را برآورده می سازد. در یادگیری تقویتی، عامل نیازی بـه مـدل سـازی صریح محیط ندارد، زیرا اعمالش می توانند مستقیما براساس پاداشهای دریافتی از محیط پایه گذاری شوند. بنابراین ایـن روش یادگیری به خصوص در زمانی مفید است که عاملها دانش اندکی از محیط دارند.عامل در سیستم چندعاملـه ممکـن اسـت بـه علت توزیع شدگی اطلاعات آگاهی اندکی از سایرین داشته باشد. حتـی اگـر عامـل اطلاعـات از قبـل دانسـته شـده ای دربـارۀ عاملهای دیگر داشته باشد، به خاطر این که عاملهای دیگر نیز در حال یادگیری هسـتند، محـیط غیرایسـتا مـی باشـد و رفتـار عاملهای دیگر ممکن است در طی زمان تغییر کند. بنابراین استفاده از یادگیری تقویتی در این سیستمها امری طبیعی است.

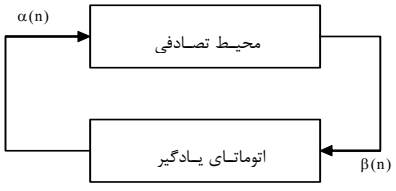
باید توجه داشت که در سیستمهای چندعامله مسئله یادگیری پیچیدگی بیشتری دارد و عامل نه تنها باید اثـرات اعمـال خودش را بیاموزد، بلکه باید چگونگی هماهنگ نمودن اعمالش با اعمال سایر عاملهای دیگر را نیز بیاموزد. کارهای جاری نشـان می دهد که یادگیری تقویتی اغلب منجر به دستیابی به رفتار متوازن یا هماهنگ می گردد. از الگوریتمهـای یـادگیری تقـویتی که تجربه کمتری دربارۀ آن در یادگیری تقویتی سیستمهای چندعامله وجود دارد، اتوماتای یادگیر می باشد.

در مدل اتوماتای یادگیر، تصمیم گیر در محیطی تصادفی عمـل مـی کنـد و اسـتراتژی خـودش بـرای انتخـاب اعمـال را براساس پاسخ دریافتی بهنگام سازی می کند. اتوماتا دارای تعداد محدودی عمل می باشد و متناظر با هر عمل پاسخ محیط با درجه ای از اطمینان می تواند مطلوب یا نامطلوب باشد. اتوماتا با به کارگیری استراتژیهای قطعی یا تصادفی می تواند به مقاصد متفاوتی دست پیدا کند. این مقاله، کارآیی اتوماتای یادگیر را به عنوان استراتژی تصمیم گیری عاملهـا در سیسـتم چندعاملـه مورد ارزیابی قرار می دهد و آن را با تکنیک یادگیری Q مقایسه می کند. در این بررسی دو پرسش اصلی وجود دارد. اول ایـن که آیا عاملهای یادگیرنده با کمک اتوماتای یادگیر در سیستمهای چندعامله همگرا می شوند؟ دوم این که آیـا تفاوتهـایی بـین عاملهای یادگیرنده مستقل و یادگیرندۀ مشترک وجود دارد؟ همچنین، نتایج حاصل، در راستای به کـارگیری اتوماتـای یـادگیر برای پیاده سازی استراتژی مذاکره در عاملهای مذاکره کننده استفاده شده است.

بدین منظور ابتدا در بخش ۲، اتوماتای یادگیر با ساختار متغیر و شماهای تقویتی در محیطهای S مورد بررسی قـرار مـی گیرد. بخش ۳ به توضیح مسئلۀ یادگیری تقویتی در سیستمهای چندعامله و تفاوت بین یادگیرنده های مستقل و مشـترک در بازیهای همکاری می پردازد. بخش ۴ ارزیابی یادگیری Q در ساختار مسئله به عنوان یکـی از الگوریتمهـای یـادگیری تقـویتی سیستمهای چندعامله را ارائه می کند. بخش ۵ به ارزیابی اتوماتای یـادگیر و بررسـی چگـونگی همگرایـی آن در سیسـتمهای چندعامله اختصاص دارد. بخش ۶ نیز نتایج را ارائه می کند.

۲- اتوماتای یادگیر

شاخه ای از نظریۀ کنترل تطبیقی به اتوماتای یادگیر اختصاص دارد. اتوماتای یادگیر یک مدل انتزاعـی اسـت کـه تعـداد معدودی عمل را می تواند انجام دهد. هر عمل انتخاب شده توسط محیطی احتمالی ارزیابی شده و پاسخی به اتوماتـای یـادگیر داده می شود. اتوماتای یادگیر از این پاسخ استفاده نموده وعمل خود را برای مرحله بعد انتخـاب مـی کنـد [۱۰,۱۴]. شـکل ۱ ارتباط بین اتوماتای یادگیر و محیط را نشان می دهد.



شکل ۱. ارتباط بین اتوماتای یادگیر و محیط

[5] robustness

اتوماتای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم می گردد [۱۰،۱۳،۱۹] . اتوماتای یادگیر با ساختار ثابت با احتمالات گذر وضعیت ثابت مشخص[۶] می شود. نظریهٔ زنجیره های مارکوف ابزار اصلی تجزیه و تحلیل این کلاس از اتوماتا می باشد و در اغلب موارد، رفتار مقتضی با انتخاب احتمالات گذر وضعیت اتوماتون در پاسخ به خروجی محیط به دست می آید. رفتار کلی سیستم توسط ماتریس گذر وضعیت زنجیرهٔ مارکوف تعیین می شود. با توجه به این که در این مقاله از اتوماتای ساختار متغیر استفاده شده است، در ادامه توضیحاتی در رابطه با اتوماتای ساختار متغیر داده می شود. برای مطالعه بیشتر در رابطه با اتوماتاهای ساختار ثابت و متغیر می توان به [۱۰،۱۱،۱۲،۱۳،۱۴،۱۹] مراجعه نمود. اتوماتای یادگیر با ساختار متغیر[۷] توسط ۴ تایی $\{\alpha,\beta,p,T\}$ نشان داده می شود که در آن $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه عملهای اتوماتا، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ مجموعه ورودیهای اتوماتا و $p = \{p_1, p_2, \ldots p_n\}$ بردار احتمال انتخاب هر یک از اعمال و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می باشد. در این نوع از اتوماتاها، اگر عمل α_i در مرحله n ام انتخاب شود و این عمل، پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمالها کاهش می یابند. برای پاسخ نامطلوب احتمال $p_i(n)$ کاهش یافته و سایر احتمالها افزایش می یابند. در هر حال، تغییرات به گونه ای صورت می گیرد تا حاصل جمع $p_i(n)$ ها همواره ثابت و مساوی یک باقی بماند. الگوریتم (۱) نمونه ای از الگوریتمهای یادگیری خطی در اتوماتای با ساختار متغیر است[۱۴].

الف- پاسخ مطلوب برای عمل i:

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)] \qquad (۱)$$
$$p_j(n+1) = (1-a)\,p_j(n) \qquad j \neq i \quad \forall j$$

ب- پاسخ نامطلوب برای عمل i:

$$P_i(n+1) = P_i(n) - (1-b)\,P_i(n) \qquad p_j(n+1) = \frac{b}{r-1} + (1-b)\,p_j(n) \quad \forall j \quad j \neq i$$

در روابط فوق، پارامتر پاداش و a پارامتر پاداش و b پارامتر جریمه می باشد. با توجه به مقادیر a و b سه حالت زیر را می توان در نظر گرفت. زمانی که a و b با هم برابر باشند، الگوریتم را L_{R-P}[۸] می نامیم. زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را $L_{R-\varepsilon P}$[۹] می نامیم و زمانی که b مساوی صفر باشد، الگوریتم را L_{R-I}[۱۰] می نامیم.

با درنظر گرفتن طبیعت پاسخ محیط، مدلهای P، Q و S برای محیطی که اتوماتای یادگیر در آن عمل می کند،درنظرگرفته می شود[۱۴]. پاسخ در محیطهای مدل P دارای مقدار دودویی می باشد. در مدل Q متناظر با عمل α_i، خروجی محیط ممکن است تعداد متناهی از مقادیر اختیار کند. با نرمال سازی مقادیر خروجی، هر مدل Q با مقادیر متناهی از خروجیهای محیط در فاصلهٔ واحد [0,1] مشخص می گردد. تعداد این مقادیر خروجی از عملی به عمل دیگر متفاوت است و با m_i برای عمل α_i $(i=1,2,\ldots,\gamma)$ بیان می شود. در مدل S، پاسخها می توانند مقادیری پیوسته در یک فاصله مشخص را اختیار کنند. با نرمال سازی پاسخها، می توان آنها را در فاصله [0,1] درنظرگرفت. اگر پاسخ محیط در مدل Q برای عمل α_i با $\beta^i_1, \beta^i_2, \ldots, \beta^i_{mi}$ مشخص شود که در آن $\beta^i_1 < \beta^i_2 < \ldots < \beta^i_{mi}$ ، مجموعه نرمال شدهٔ پاسخها $\{\beta^i_j\}$ به شکل $\beta^i_j = \beta^i_j - a \,/\, b-a \quad j=1,2,\ldots,m_i$ تعریف می گردد که در آن $a = \min_i\{\beta^i_j\}$ و $b = \max_i\{\beta^i_{mi}\}$ است. نرمال سازی مشابهی را نیز می توان در مدل S انجام داد. نگارشهای مدل S و Q برای شمای L_{R-I} و L_{R-P} به صورت زیر می باشند. باید توجه داشت که مدل S بازنمایی عمومی تری ازدو نگارش قبلی است. با داشتن مدل S می توان مدلهای Q و P را نیز به دست آورد. بهنگام سازی احتمالات در شمای SL_{R-I} براساس معادلهٔ (۲) بیان می شود:

$$P_i(n+1) = P_i(n) - a\,(1-\beta(n))\,P_i(n) \qquad \alpha(n) <> \alpha_I \qquad (۲)$$
$$P_i(n+1) = P_i(n) + a\,(1-\beta(n))\,\Sigma_{j \neq i}\,P_j(n) \qquad \alpha(n) = \alpha_i$$

---

[6] fixed state transition probablities
[7] Variable Structure Learning Automata
[۸] Linear Reward Pealty
[۹] Linear Reward Epsilon Penalty
[۱۰] Linear Reward Inaction

شمای SL_{R-P} برای مدلهای Q و S براساس معادلهٔ (۳) بیان می شود:

$$P_i(n+1) = P_i(n) + \beta(n)\,[(a/r-1) - a\,P_i(n)] - [1-\beta(n)]\,a\,P_i(n) \qquad \alpha(n) <> \alpha_I \qquad (۳)$$
$$P_i(n+1) = P_i(n) + \beta(n)\,a\,P_i(n) + (1-\beta(n))\,a\,(1-P_i(n)) \qquad \alpha(n) = \alpha_i$$

مدلهای اتوماتای یادگیر در تصمیم گیریهای تطبیقی که در آنها تصمیم گیر باید به منظور بهینه سازی کارآیی کلی سیستم از بین چندین عمل، عمل مناسبی را به شکل برخط انتخاب کند، مانند تحصیص کانال به شکل پویا، پرهیز از تصادم در شبکه های ستاره ای، مسیریابی در شبکه تلفن، کاربرد دارد. بحث این مقاله نیز در این حوزه از کاربردهای اتوماتای یادگیر قرار می گیرد. همچنین مفیدبودن مدلهای اتوماتای یادگیر در مسائل بهینه سازی اتفاقی مانند یادگیری توابع تفکیک کننده در بازشناسی الگو به اثبات رسیده است. ازجمله دیگر کاربردهای مدلهای اتوماتای یادگیر در کنترل تطبیقی، پردازش سیگنال پردازش تصویر، بخش بندی اشیاء می باشد[۱۹]. برای مطالعه بیشتر در باره اتوماتاهای یادگیر می توان به [۱۰,۱۱,۱۲,۱۳,۱۴,۱۹] مراجعه کرد.

۳- یادگیری تقویتی در سیستمهای چندعامله

یکی از مسائلی که هم اکنون در تحقیقات در زمینهٔ سیستمهای چندعامله مورد توجه قرار گرفته است، استفاده از تکنیکهای یادگیری و تجهیز سیستمهای چندعامله با تواناییهای یادگیری می باشد[۱۵] . معمولا سیستمهای چندعامله در محیطهای پیچیده- بزرگ، باز، پویا و غیرقابل پیش بینی- عمل می کنند. در چنین محیطهایی مشخص نمودن سیستمهای چندعامله به طور کامل و درست در زمان طراحی و پیش از استفاده مشکل و گاهی غیرممکن است. یعنی طراح باید مشخص کند که چه شرایطی در محیط پیش خواهد آمد، در زمان نیاز چه عاملهایی در دسترس هستند و عاملهای دردسترس چگونه باید در پاسخ به شرایط محیطی واکنش نشان بدهند و بایکدیگر ارتباط برقرار کنند. تنها راه ممکن برای حل این مشکل این است که هر عامل، توانایی بهبود کارآیی خود و کل سیستم را داشته باشد.

هدف عاملهایی که در محیطهای پویا عمل می کنند، این است که تصمیمهای بهینه بگیرند. اگر عاملها از پاداشهای متناظر با اعمال مختلف مشترکی که در محیط انجام می دهند، آگاه نباشند، انتخاب عمل مشکل می شود. یادگیری با تنظیم انتخاب عمل عاملها براساس اطلاعات جمع آوری شده در طی زمان چنین مقصودی را برآورده می سازد. یادگیری این امکان را می دهد که عاملها براساس تجربهٔ گذشته، پاداش موردانتظار برای اعمال فردی یا مشترک را تخمین بزنند. عامل انتخاب عمل خودش را براساس پیش بینی محیط یا مستقیما براساس پاداش دریافتی از محیط قرار می دهد. یادگیری تقویتی روشی سیستماتیک است که عمل عامل را با پاداش دریافتی از محیط مربوط می سازد[۱,۶,۱۸]. در یادگیری تقویتی، عامل نیازی به مدل سازی صریح محیط ندارد زیرا اعمالش می تواند مستقیما براساس پاداشهای دریافتی از محیط پایه گذاری شود. بنابراین این روش یادگیری به خصوص در زمانی مفید است که عاملها دانش اندکی از محیط دارند.

عامل در سیستم چندعامله ممکن است به علت توزیع شدگی اطلاعات آگاهی اندکی از سایرین داشته باشد. حتی اگر عامل اطلاعات از قبل دانسته شده ای دربارهٔ عاملهای دیگر داشته باشد، به خاطر این که عاملهای دیگر نیز در حال یادگیری هستند، محیط غیرایستا می باشد و رفتار عاملهای دیگر ممکن است در طی زمان تغییر کند. بنابراین استفاده از یادگیری تقویتی در این سیستمها امری طبیعی است. فوتبال روباتها [۱۶]، بازی تعقیب و گریز [۱۸] و هماهنگ سازی توزیع شده تیم [۳] مثالهایی از کاربردهایی هستند که یادگیری تقویتی در آنها می تواند موثر باشد. موضوع هماهنگی[11] در سیستمهای چندعامله موضوع مهمی می باشد. به کارگیری یادگیری در مسئلهٔ هماهنگی در سیستمهای چندعامله در هوش مصنوعی و نظریه بازیها بسیار معروف شده است. یکی از این مدلهای یادگیری برای دستیابی به همکاری، بازی ساختگی[12] است[۱۵] . هر عامل i برای هر $j \in a$ و $a_j \in A_j$ شمارندهٔ C^i_{aj} تعداد دفعاتی که عامل j عمل a_j را در گذشته انجام داده است، نگهداری

[11] coordination
[12] ficitious play

می کند. در بازی، عامل i از تناوب نسبی هر عمل j به عنوان نشانگری از استراتژی فعلی j استفاده می کند. یعنی برای هر عامل j، عامل i حساب می کند که عامل j عمل $a_j \in A_j$ را با احتمال $Pr^i_{aj} = C^j_{aj} / (\Sigma_{bj\ \varepsilon\ Aj}\ C^j_{bj})$ انجام می دهد. این مجموعه از استراتژیها، استراتژی کاهش یافته Π-i را شکل می دهد که برای آن عامل i بهترین پاسخ را انتخاب می کند. بعد از هر تکرار بازی، عامل i شمارشگرهای خود را براساس اعمال انجام شده توسط عاملهای دیگر بهنگام سازی می کند. این شمارشگرها منعکس کنندهٔ باور عامل در رابطه با بازی دیگر عاملها می باشند. این استراتژی تطبیقی ساده به نقطهٔ موازنه همگرا می شود. روش دیگر استفاده از الگوریتم یادگیری Q برای عاملها می باشد که توسط بوتیلیر و کراوس در [۳] مورد ارزیابی قرار گرفته است. یادگیری تقویتی به شکل زیادی در سیستمهای چندعامله استفاده می شود. استفاده از یادگیری تقویتی به عنوان ابزاری برای دستیابی به رفتار هماهنگ به خاطر عمومیت و استحکامی که دارد، بسیار جذاب است[۶] . یکی از مشکلاتی الگوریتمهای یادگیری تقویتی در سیستمهای چندعامله با آن مواجه هستند وجود چندین نقطهٔ موازنه می باشد [۲] . در چنین حالتی، این روشها با این مشکل مواجه هستند که عاملها باید انتخاب نقطه موازنه را با یکدیگر هماهنگ کنند. در این مقاله، بررسی می شود که چگونه با به کارگیری اتوماتای یادگیری با این مسئله برخورد می شود.

۳-۱- یادگیرنده های مستقل و مشترک در بازیهای تکراری

در این مقاله، به منظور ارزیابی کارآیی استراتژی اتوماتای یادگیر از بازیهای تکراری که در آنها علائق مشترک یا همکاری وجود دارد، استفاده می شود. به شکلی رسمی تر، مجموعهٔ a از n عامل وجود دارد که در آن هر عامل مجموعهٔ محدودی از اعمال A_i را در دسترس دارد. عاملها به شکل تکراری یک بازی مرحله ای را تکرار می کنند که در آن هرکدام به طورمستقل عمل مجزایی را برای انجام انتخاب می کند. اعمال انتخاب شده در هر مرحله عملی مشترک را شکل می دهند که مجموعهٔ آنها را می توان به شکل $A = \times\ A_i\ (i \in a)$ نشان داد. برای هر $a \in A$ توزیع متناظری برروی پاداشهای ممکن وجود دارد. مسئلهٔ تصمیم گیری در اینجا همکاری می باشد زیرا که پاداش هر عامل از توزیع یکسانی به دست می آید. عاملها تمایل به انتخاب اعمالی را دارند که پاداش را حداکثر سازد. کارآیی الگوریتمهای یادگیری تقویتی یادگیری Q و الگوریتم اتوماتای یادگیر براساس بازی هماهنگی با ساختار سادهٔ زیر بررسی می شود:

	a_0	a_1
b_0	10	0
b_1	0	10

علاوه بر بررسی کارآیی دو الگوریتم یادگیری فوق در این بازی، هدف دیگر بررسی کارآیی دو راه به کارگیری الگوریتمهای تقویتی در در سیستمهای چندعامله می باشد. دو راه برای به کارگیری الگوریمتهای تقویتی یادگیری در سیستمهای چندعامله وجود دارد. اگر هر عاملی احتمالات انجام عمل را براساس اعمال مجزایی که خودش انجام می دهد، بهنگام سازی کند یا یادبگیرد، الگوریتم یادگیری تقویتی سیستم چندعامله، *الگوریتم یادگیرندهٔ مستقل* است. یعنی، عاملها اعمالشان را انجام می دهند، بازخوردی از محیط دریافت می کنند و بدون درنظرگرفتن اعمال صورت گرفته توسط سایر عاملها، احتمالات انجام اعمالشان را بهنگام سازی می کنند. تجربیات هر عامل i شکل $<a_i,b>$ را به خود می گیرد که در آن a_i عمل انجام شده توسط عامل i و b پاسخ دریافت شده از محیط است.اگر عاملها از وجود عاملهای دیگر آگاه نباشند، نتوانند اعمال یکدیگر را تشخیص دهند یا دلیلی نداشته باشند که عاملهای دیگر به شکل استراتژیک عمل می کنند، آنگاه این الگوریتم برای یادگیری روشی مناسب است.

یادگیرندهٔ عمل مشترک، احتمالات اعمال مشترک را بهنگام سازی می کند. تجربیات هر عامل به شکل $<a,r>$ است که در آن a عمل مشترک است. در این حالت، اعمال در اختیار هر عامل ترکیبهای دوتایی از تعداد اعمال هر عامل می باشد. در حالتی که هر عامل بتواند دو عمل انجام دهد، هر عامل چهار عمل مشترک به شکل $<a_0,b_0>$، $<a_0,b_1>$ و ... در اختیار دارد.

۳-۲- به کارگیری اتوماتای یادگیر در مذاکره عاملها

همانگونه که بیان شد، مذاکره خودکار یکی از روشهای هماهنگی در سیستمهای چندعامله می باشد. در فرآیند مذاکره خودکار گروهی از عاملها در رابطه با موضوع خاصی، به تصمیمی مشترک یا توافقی قابل قبول برای همه دست می یابند. سیستم مذاکره با چندتایی $N = <Ag,S,\mathcal{P}>$ تعریف می شود که در آن Ag مجموعهٔ متناهی عاملهای مذاکره کننده، Sاستراتژی مذاکره و $\mathcal{P}$ پروتکل مذاکره است. پروتکل مذاکره مجموعه قوانین حاکم برتعامل است و استراتژی مذاکره راهنمایی برای تصمیم گیری در رابطه با اعمال متفاوت در مرحله ای معین است. اگر A مجموعه اعمال مذاکره باشد. استراتژی مذاکره به شکل ساده به صورت تابع $S: 2^A \rightarrow A$ تعریف می شود، به طوری که اگر $T \subseteq A$باشد، $S(T) \in T$ است.

اگر بخواهیم مسئلهٔ انتخاب شده برای ارزیابی را در حوزهٔ کاربردی مذاکرهٔ خودکار تصویر کنیم، می توان گفت که در این مسئله، مجموعه $Ag=\{A,B\}$، پروتکل مذاکره مشخص می کند که مجموعه اعمال مجاز برای عامل A مجموعهٔ $\{a_0,a_1\}$ و برای عامل B مجموعهٔ $\{b_0,b_1\}$ می باشد. در هر مرحله، عاملها باید به طور همزمان یکی از دو عمل مجاز را انجام دهند. پس از دستیابی به نقطهٔ موازنه، فرآیند مذاکره پایان می یابد. استراتژی مذاکره هر عامل، الگوریتم یادگیری اتوماتای یادگیر خواهد بود. به عنوان مثال، در یک بازار الکترونیکی که عاملها در آن معامله می کنند، اعمال هر عامل ممکن است، بالا بردن قیمت، پایین آوردن قیمت یا ثابت نگه داشتن قیمت باشد و بازار نیز براساس مکانیزم مشخصی پاداش یا جریمه ای را به هر عمل بدهد. عاملها براساس الگوریتم اتوماتای یادگیر احتمال انجام عمل را بهنگام سازی می کنند، تا زمانی که عاملها در رابطه با قیمت به یک نقطه موازنه همگرا شوند.

۴- ارزیابی یادگیری Q در همکاری عاملها

برای پیاده سازی بازی فوق توسط الگوریتم یادگیری Q از حالت بدون وضعیت آن استفاده می شود[۱۷] . در این حالت فرض می گردد که مقدار Q(a) تخمینی از ارزش انجام عمل فردی یا مشترک a ارائه می کند. عامل مقدار Q(a) را براساس نمونهٔ <a,r> براساس معادلهٔ ۴ بهنگام سازی می کند:

$$Q(a) \leftarrow Q(a) + \lambda(r-Q(a)) \qquad (۴)$$

نمونهٔ <a,r> تجربهٔ حاصل شده توسط عامل است. عمل a انجام شده و پاداش r را دریافت کرده است. در اینجا λ نرخ یادگیری می باشد که $0 < \lambda < 1$ است و میزان جایگزینی نمونهٔ جدید با تخمین فعلی است. دو شکل برای انجام عمل وجود دارد. در حالت nonexploitive exploration عامل به طور تصادفی با احتمال یکنواخت عمل خودش را انجام می دهد. هیچ تلاشی برای استفاده از آنچه که فراگرفته شده است، نمی شود. هدف یادگیری مقادیر Q می باشد. در exploitive exploration عامل بهترین عمل را با احتمال p_x و عمل دیگر را با احتمال $1-p_x$ انتخاب می کند. یک استراتژی برای این کار Boltzman exploration می باشد. عمل a با احتمال زیر انتخاب می شود[۱۷] :

$$e^{Q(a)/T} / \Sigma_{a'} e^{Q(a')/T}$$

پارامتر حرارت T را در طی زمان می توان کاهش داد به طوری که احتمال exploitation افزایش یابد[۴,۱۷] .

همانگونه که بیان شد، یکی از اهداف انجام پیاده سازیها در این مقاله، بررسی کارآیی عاملهای یادگیرندهٔ مستقل و مشترک در سیستم چندعامله می باشد. در رابطه با عاملهایی که از الگوریتم یادگیری Q استفاده می کنند، در حالت یادگیری مستقل، عاملها مقادیر Q را برای اعمال فردی خود براساس رابطه بیان شده در بالا یاد می گیرند. عامل عمل را انجام می دهد، پاداش از محیط دریافت می کند و بدون درنظر گرفتن عمل انجام شده توسط عامل دیگر، مقدار Q خود را بهنگام سازی می کند.

در حالت عاملهای یادگیرندهٔ مشترک، عامل مقادیر Q را برای اعمال مشترک یاد می گیرد. هر عامل می تواند اعمال عاملهای دیگر را مشاهده کند. در مورد ساختار بازی پیاده سازی شده، عامل یادگیرندهٔ مشترک، مقادیر Q را برای چهار عمل

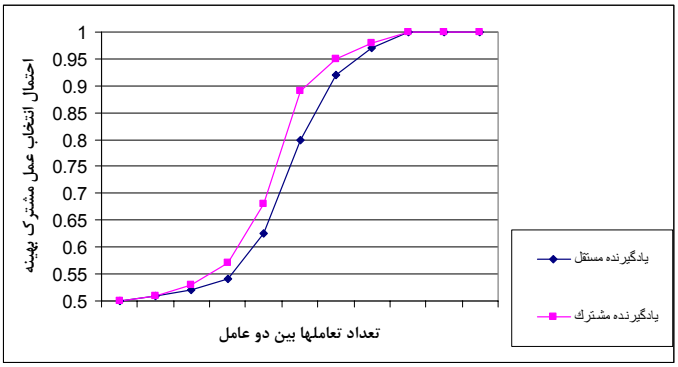
مشترک <a_1,b_1> , <a_0,b_0>,<a_0,b_1>,<a_1,b_0> یاد می گیرد. در حالت یادگیرندهٔ مشترک، اگر عامل A دارای مقادیر Q برای تمام چهار عمل مشترک باشد، ارزش موردانتظار انجام a_0 یا a_1 بستگی به استراتژی به کار رفته توسط عامل B دارد. به طور کلی، مقدار موردانتظار عامل i برای عمل a_i به شکل زیر محاسبه خواهد شد[۱] :

$$EV(a^i) = \Sigma\ Q(a^{-i}\ U\ \{a^i\})\ \Pi_{j<>i}\ \{Pr^j_{a^{-i}[j]}\} \quad (۵)$$

عامل i می تواند از این مقادیر مانند مقادیر Q در پیاده سازی استراتژی کاوش استفاده کند.

در پیاده سازی انجام شده، هر دو یادگیرندهٔ مستقل و مشترک از استراتژی کاوش بولتزمن استفاده می کنند. پارامتر حرارت یا T در ابتدا برابر ۱۶ درنظر گرفته شد و در هر تکرار با فاکتور ۰/۹ تقلیل یافت. نتیجه حاصل در شکل ۲ دیده می شود. یادگیرنده های مستقل به سرعت و در تعامل ۴۰ام همگرا شدند. بازی فوق دو نقطهٔ موازنه دارد که در این پیاده سازی دیده شد که اولویتی بین این دو نقطهٔ موازنه یعنی <a_0,b_0> و <a_1,b_1>وجود نداشت. مقادیر Q برای اعمال موجود در نقطهٔ موازنه برابر ۱۰ شدند در حالی که برای اعمال دیگر برابر صفر گردیدند. البته می توان با تغییر دادن پارامترهای یادگیری Q مانند پارامتر T سرعت همگرایی را نیز بیشتر کرد.

در مورد یادگیرنده های مشترک در شکل ۲ دیده می شود که عملکرد آنها با داشتن اطلاعات بیشتر در رابطه با اعمال عامل دیگر، تاحدودی بهتر است، اما خیلی هم تغییری زیادی در سرعت همگرایی به وجود نیامده است. شاید دلیل این امر این باشد که عاملهای یادگیرندهٔ مشترک می توانند مقادیر Q را برای اعمال مشترک تمایز دهند، اما توانایی آنها برای استفاده از این اطلاعات با مکانیزم انتخاب عمل محدود می گردد. عامل دارای باوری از استراتژی عامل دیگر است و اعمال را براساس مقدار موردانتظار محاسبه شده براساس این باورها انتخاب می کند.



شکل ۲. همگرایی همکاری برای یادگیرنده های فردی و مشترک (در ۵۰ تکرار)

۵ ارزیابی اتوماتای یادگیر در همکاری عاملها

در این بخش، کارآیی اتوماتای یادگیر به منظور ایجاد همکاری در سیستمهای چندعامله مورد بررسی قرار می گیرد. پرسشهای اساسی در این ارزیابی به شرح زیر هستند:

- آیا تضمین می شود که اتوماتای یادگیر در سیستمهای چندعامله همگرا شود؟
- آیا تفاوتی بین یادگیرنده های مستقل و یادگیرنده های مشترک وجود دارد؟

به شکل رسمی می توان سیستم فوق را به این شکل توصیف نمود[۱۷] . $P^j(n) = [P^j_1(n), P^j_2(n),\ldots,P^j_{rj}(n)]$ که در آن $P^j(n)$ توزیع احتمالی است که انتخاب اعمال در اتوماتای A_j در مرحلهٔ n ام را اداره می کند و $Pji(n) = Pr[a^j(n) = a_{ij}]$. در

مرحلهٔ n احتمال مشترک $Pr[a(n) = [a^1_{i1},a^2_{i2},\ldots,a^N_{iN}]]$ را به شکل $p_{i1,i2,\ldots,iN}(n)$ نشان می دهیم. اگر بازی براساس این توزیع $a(n) = [a^1_{i1},a^2_{i2},\ldots,a^N_{iN}]$ باشد و برای این بازی

$$E[\beta^j(n) \mid a(n) = [a^1_{i1},a^2_{i2},\ldots,a^N_{iN}]] = d^j_{i1,i2,\ldots,iN}$$

آنگاه خروجی موردانتظار اتوماتای Aj در مرحلهٔ n برابر است با

$$Mj(n) = E[b^j(n) \mid p^1(n), p^2(n), \ldots, p^N(n)]$$

$$= \Sigma_{i1,i2,\ldots,iN}\, p^1_{i1}(n)p^2_{i2}(n)\ldots p^N_{iN}(n)d^j_{i1,i2,\ldots,iN}$$

با استفاده از تعریف بالا، سیستم فوق با استفاده از فرآیند مارکوف که فضای حالت آن S فضای ضرب سیمپلکس است، توصیف می گردد. یعنی

$$S = S^1_{r1} * S^2_{r2} * \ldots. * S^N_{rN}$$

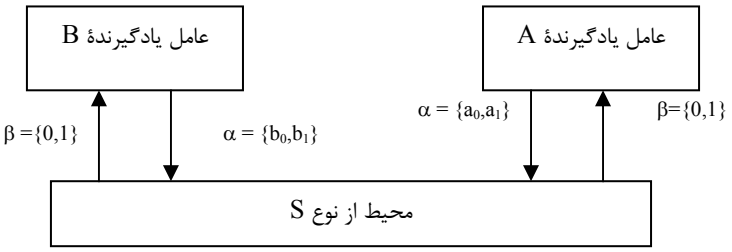
که در آن S^i_{ri} سیمپلکس r_i بعدی برای اتوماتای A_j است. در هر لحظه n، براساس توزیعهای احتمال $p^1(n), p^2(n),\ldots, p^N(n)$ اتوماتا عمل a(n) را انتخاب می کند و براساس پاسخ دریافتی از محیط و شماهای یادگیری مورد استفاده در فضای S حرکت می کند.

در ارزیابی انجام شده این بازی را براساس ساختار بیان شده در بخش ۳-۱، دو عامل که به عنوان استراتژی تصمیم گیری خود از الگوریتم اتوماتای یادگیر استفاده می کنند، با یکدیگر انجام می دهند. این عاملها در هر مرحله از الگوریتم یا شمای تقویتی L_{R-P} برای تصمیم گیری استفاده می کنند. با توجه به این که پاسخهای محیط یعنی $b = \{0,10\}$ می باشد، محیط از نوع S می باشد که با نرمال سازی و تغییر مکان دو پاسخ محیط الگوریتم L_{R-P} عمل می کند. با درنظر گرفتن مجموعهٔ اعمال دو عامل به شکل $A= \{a_0,a_1\}$ و $B= \{b_0,b_1\}$ و پاسخ محیط به شکل نرمال شدهٔ $b = \{0,1\}$ شمای تقویتی مورد استفادهٔ دو عامل برای انتخاب بهترین عمل در هر مرحله از بازی به شکل زیر می باشد.

$$P_1(n+1) = P_1(n) + a\,(1-P_1(n)) \quad \alpha(n) = \alpha_1$$
$$P_2(n+1) = (1-a)\,P_2(n) \quad \beta(n) = 0$$

$$P_1(n+1) = (1-b)\,P_1(n) \quad \alpha(n) = \alpha_1$$
$$P_2(n+1) = P_2(n) + b\,(1-P_2(n)) \quad \beta(n) = 1$$

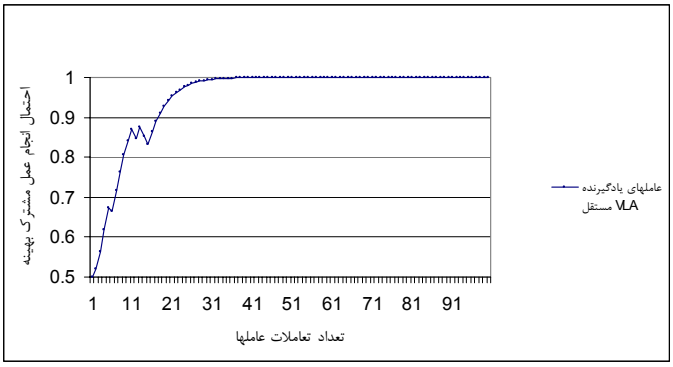
در پیاده سازی انجام شده، مقادیر مختلف پارامترهای a و b و تاثیر آنها در سرعت همگرایی و هماهنگ شدن عاملها مورد بررسی قرار گرفت.



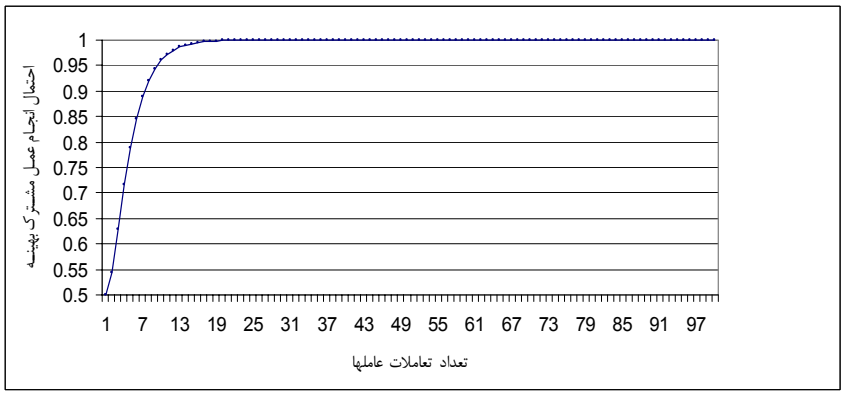
شکل ۳. دو عامل یادگیرنده براساس اتوماتای یادگیر که در محیط عمل می کنند

در پیاده سازی انجام شده، سعی شد تا همگرایی دو عامل همگن فوق، براساس مقادیر مختلفی از پارامترهای پاداش و جریمه a و b بررسی شود. نتیجهٔ بررسی برای دو مقدار از پارامترهای a و b در شکلهای ۴ و ۵ دیده می شود. شکل ۴ سرعت

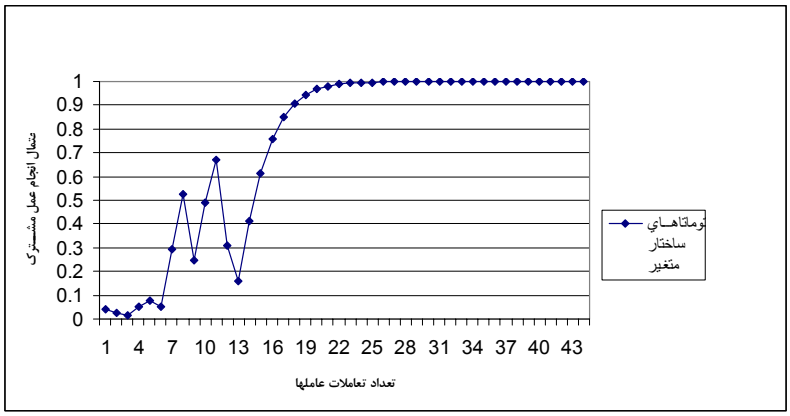
همگرایی و شیوهٔ همگرایی به ازاء مقادیر a=0.3, b=0.03 برای عاملهای یادگیرندهٔ مستقل نشان می دهد. شکل ۵ سرعت همگرایی و شیوهٔ همگرایی را برای عاملهای یادگیرندهٔ مستقل به ازاء a=0.4, نشان می دهد. دیده می شود که سرعت همگرایی افزایش یافته است و حرکت به سمت نقطهٔ موازنه روان تر شده است.



شکل ۴. سرعت و شیوهٔ همگرایی دو عامل یادگیرنده مستقل براساس شمای تقویتی L_{R-P} به عمل مشترک بهینه به ازاء a=0.3 , b=0.03 (با تعداد تعاملات ۱۰۰)



شکل ۵ سرعت و شیوهٔ همگرایی دو عامل یادگیرنده مستقل براساس شمای تقویتی L_{R-P} به عمل مشترک بهینه به ازاء a=0.4 , b=0.03 (با تعداد تعاملات ۱۰۰)



شکل ۶. سرعت و شیوهٔ همگرایی دو عامل یادگیرنده مشترک براساس شمای تقویتی L_{R-P} به عمل مشترک بهینه به ازاء a=0.4 (با تعداد تعاملات ۵۰)

انتخاب مقدارهای پارامترهای پاداش و جریمه a و b در شمای تقویتی در سرعت همگرایی و همچنین مستقیم بودن مسیر همگرایی عاملها به عمل مشترک تاثیر می گذارد. در اینجا هم مشخص گردید که دو عامل هیچ اولویتی برای نقطهٔ موازنه قائل نمی شوند و هر کدام از دو نقطهٔ موازنه $<a_0,b_0>$ و $<a_1,b_1>$ در نیمه دوم آزمایشات به دست آمدند. همگرایی و کارآیی یادگیرنده های مستقلی که از شمای تقویتی L_{R-P} برای تصمیم گیری استفاده می کنند به خوبی یادگیرنده های مستقل Q می باشد و می توان گفت که با انتخاب دقیق تر پارامترهای a و b می توان این کارآیی و سرعت همگرایی را افزایش داد. پس نتیجهٔ آزمایشات تجربی، نشان می دهند که *اتوماتای یادگیر در سیستمهای چندعامله همگرا می شود.*

در رابطه با اتوماتاهای یادگیری که مشترک یادمی گیرند از شمای تقویتی با چهار عمل استفاده شد که در آن احتمالات انجام عمل براساس فرمول زیر بهنگام سازی می گردیدند:

$$P_i(n+1) = P_i(n) + \beta(n)\,[(a/r-1) - a\,P_i(n)] - [1 - \beta(n)]\,\alpha\,P_i(n) \quad when \quad \alpha(n) <> \alpha_i$$
$$P_i(n+1) = P_i(n) + \beta(n)\,a\,P_i(n) + (1- \beta(n))\,a\,(1-P_i(n)) \quad when \quad \alpha(n) = \alpha_i$$

همانگونه که در شکل ۵ دیده می شود، این دو اتوماتا بعد از حدود ۴۵ تکرار به عمل بهینه مشترک همگرا شده اند. اما همانگونه که دیده می شود، مسیر همگرایی چندان مستقیم و راحت نمی باشد. یکی از دلایل می تواند تعداد اعمال هر اتوماتا و کوچک شدن احتمال هر عمل باشد. در اینجا هم مقدار پارامتر a برروی سرعت و مسیر همگرایی تاثیر می گذارد. در پیاده سازیهای انجام شده مقدار a=0.4 یک مقدار مناسب برای پارامتر a بود که نمودار همگرایی شکل ۶ براساس این پارامتر به دست آمده است. همچنین، در اینجا هم آگاهی از عمل عامل دیگر برروی سرعت همگرایی تاثیر چندانی نگذاشته است.

۶ نتیجه گیری

یکی از مشکلاتی که الگوریتمهای یادگیری تقویتی در سیستمهای چندعامله با آن مواجه هستند، وجود چندین نقطهٔ موازنه می باشد. دریادگیری تقویتی سیستمهای چندعامله بیشترین تمرکز برروی تضمین همگرایی الگوریتمهای یادگیری به نقطهٔ موازنهٔ مطلوب است. این روشها با این مشکل مواجه هستند که عاملها باید انتخاب نقطه موازنه را با یکدیگر هماهنگ کنند. در این مقاله، با انجام آزمایشهای تجربی بررسی شد که چگونه عاملهایی که استراتژی تصمیم گیری آنها اتوماتای یادگیر می باشد، به نقطهٔ موازنهٔ مطلوب همگرا می شوند.

بدین منظور همگرایی الگوریتم یادگیری اتوماتای یادگیر به عنوان استراتژی تصمیم گیری عاملهایی که در ساختار یک بازی همکاری فعالیت می کنند، در دو حالت وجود یادگیرنده های مستقل و یادگیرنده های مشترک ارزیابی و کارآیی آن با الگوریتم یادگیری Q مقایسه شد. به عنوان کاربرد عملی نیز، چگونگی به کارگیری اتوماتای یادگیر به عنوان استراتژی مذاکره در مذاکرهٔ خودکار بین عاملها توضیح داده شد. نتایج حاصل نشان می دهند که الگوریتم اتوماتای یادگیر با سرعت خوبی به عمل مشترک بهینه همگرا می شوند. در این بین اتوماتای یادگیر با سرعت بیشتری احتمالات عمل مشترک بهینه را نسبت به الگوریتم یادگیری Q یاد می گیرد. در رابطه با یادگیرنده های مشترک نیز که بر اعمال یکدیگر نظارت دارند، در این دو الگوریتم تفاوت چندانی در ارزیابی صورت گرفته بین این یادگیرنده ها با یادگیرنده های مستقل دیده نشد. همچنین، در این مقاله، ایدهٔ به کارگیری اتوماتای یادگیر برای پیاده سازی استراتژی مذاکره در عاملهای مذاکره کننده ارائه گردید و چگونگی استفاده از این تکنیک در مذاکرهٔ عاملها بررسی شد.

مراجع

[1] C. Boutilier, “Planning, learning and coordination in multiagent decision processes”, *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge*, pp. 195-210, 1996.

[2] G. Chalkiadakis and Boutilier C., "Coordination in Multiagent Reinforcement Learning: A Bayesian Approach", *Proceedings of 2nd Intl. Conf. on Autonomous Agents and Multiagent Systems* (AAMAS-03), 2003.

[3] C. Claus, C. Boutilier, “The Dynamics of Learning Reinforcement in Cooperative Multiagent Systems”, *American Association for Artificial Intelligence*, 1998.

[4] L. Kaelbling, L., Littman, A. Moore, “Reinforcement Learning: A Survey”, In: *Journal of Artificial Intelligence Research*, 1996.

[5] D. Fudenberg and D. M. Kreps, “Lectures on Learning and Equilibrium in Strategic Games”, CORE Foundation, Belgium, 1992.

[6] H. Junling and M. Wellman, “Multiagent Reinforcement Learning in Stochastic Games”, 1999.

[7] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra and M. Wooldridge, Automated Negotiation: Prospects, Methods and Challenges, *Int. Journal of Group Decision and Negotiation*, 2000.

[8] N. R. Jennings, Sycara, and Wooldridge, A roadmap of agent research and development, *Autonomous Agents and Multiagent Systems Jouranl*, 1:7-38, 1998.

[9] M. Huhns, and L. Stephens, Multi-agent Systems and Societies of Agents, In: Gerhard Weiss (ed), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 1999.

[10] S. Lakshmivarahan, “Learning Algorithms: Theory and Applications”, New York, Springer Verlag, 1981.

[11] P. Mars, Chen, J. R. and Nambir, R., "Learning Algorithms: Theory and Applications in Signal Processing”, *Control and Communications,* CRC Press, Inc, 1996.

[12] M. R. Meybodi, and S. Lakshmivarahan: “Optimality of a Generalized Class of Learning Algorithm”, *Information Science*, Vol. 28, pp. 1-20, 1982.

[13] M. R. Meybodi, and S. Lakshmivarahan: “On a Class of Learning Algorithms which have a Symmetric Bahavior under Success and Failure”, *Lecture Notes in Statistics*, Springer Verlag, pp. 145-155, 1984.

[14] K. Narenrdra, S., M. A. L. Thathachar, “*Learning Automata: An Introduction*”, Prentice Hall, 1989.

[15] S. Sen, G. Weiss,”Chapter 6: Learning in Multiagent Systems”, In: Gerhard Weiss (ed), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 1999.

[16] P. Stone, “Layered Learning in Multi-Agent Systems”, *PhD thesis*, Carnegie Mellon University, 1998.

[17] R. Sutton, S., Barto A., G., “*Reinforcement Learning, An Introduction*”, MIT Press, 2000.

[18] M. Tan, “Multi-agent Reinforcement Learning: Independent vs. Cooperative Agents”, *Proceedings of the Tenth International Conference on Machine Learning*, pp. 330-337, 1993.

[19] M. A. L. Thathachar, Sastry P.S., "Varieties of Learning Automata: An Overview", *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics*, Vol. 32, No. 6, 2002.