دهمین
کنفرانس سالانه انجمن کامپیوتر ایران
۲۷ تا ۲۹ بهمن ۱۳۸۳

اتوماتای اختلاف زمانی سلولی و کاربردهای آن

بهرنگ عاصمی محمدرضا میبدی
آزمایشگاه محاسبات نرم
دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیرکبیر
تهران ایران

چکیده: در این مقاله، مدل جدیدی با نام *اتوماتای اختلاف زمانی سلولی* پیشنهاد و جزئیات آن مورد مطالعه قرار می‌گیرد. این مدل، از ترکیب یادگیری اختلاف زمانی با اتوماتای سلولی حاصل شده است و لذا سادگی اتوماتای سلولی را همراه با ویژگی‌های یادگیری اختلاف زمانی بطور همزمان دارا است. بارزترین ویژگی این مدل، ظهور رفتار برایندی است که حاصل به کارگیری *اتوماتای سلولی* می‌باشد. رفتاری که امکان یافتن پاسخ مسأله را با حل بخش‌بخش آن بصورت محلی و سپس پیداکردن برایند نتایج محلی، فراهم می‌سازد. یادگیری اختلاف زمانی، دیگر ویژگی مهم این مدل است که قدرت حل مسائل را توسط اتوماتای سلولی افزایش می‌دهد. برای نشان دادن قابلیت های مدل پیشنهادی، حل مسأله شکار و شکارچی، توسط آن مورد مطالعه قرار گرفته است.

کلمات کلیدی: اتوماتای سلولی، یادگیری اختلاف زمانی، اتوماتای اختلاف زمانی سلولی، شکار و شکارچی

۱. مقدمه

اتوماتای اختلاف زمانی سلولی[۱]، تلفیقی از اتوماتای سلولی[۲] و یادگیری اختلاف زمانی[۳] است، لذا ضروریست در ابتدا به بررسی این دو بپردازیم و سپس مدل پیشنهادی تلفیقی را مورد مطالعه قرار دهیم. از نظر تئوری، اتوماتای سلولی، اواخر دهۀ ۴۰ میلادی توسط جان وون نیومن[۴] و استانیسلاو اولام[۵] معرفی شد. از نظر کاربردی، استفاده از اتوماتای سلولی به اواخر دهۀ ۶۰ میلادی، هنگامی که کانوی[۶]، بازی حیات[۷] را ایجاد کرد، باز می‌گردد. اتوماتای سلولی، یک سیستم گسستۀ پویا است.

---

[1] Cellular Temporal Difference Automata (CTDA)
[2] Cellular Automata (CA)
[3] Temporal Difference Learning (TD)
[4] John von Neumann
[5] Stanislaw Ulam
[6] John Horton Conway
[7] Game of Life

مفهوم گسسته، این است که فضا، زمان و دیگر ویژگیهای اتوماتا، می‌توانند حالات محدود قابل شمارشی اختیار کنند.یک اتوماتای سلولی، شبکه‌ای از سلولها است که هر یک می‌تواند k ارزش مختلف اختیار کند. هر محل اتوماتا، در گام‌های زمانی گسسته، توسط یک ماشین حالت متناهی که در آن محل وجود دارد و ارزشی متناسب با ارزش محل‌های اطراف آن نسبت می‌دهد، بروز می‌شود. تعریف همسایگی در اتوماتای سلولی، ضروری است. این ویژگی اتوماتای سلولی به آن قدرت فوق‌العاده‌ای می‌دهد [1]. اما محدودیت‌های اتوماتای سلولی نظیر تعداد بسیار اندک حالات سلول، قوانین بسیار ساده و نداشتن حافظه مانع از بکارگیری آن در حل مسائل پیچیده می‌شود. این محدودیت‌ها با افزودن قابلیت یادگیری به سلول از بین می‌روند. برای نخستین بار، اتوماتای یادگیر[8] برای از بین بردن این محدودیت‌ها، توسط تسلین[9] ارائه گردید. لیکن به دلیل نداشتن ساختار سلولی، از مزایای اتوماتای سلولی بی‌بهره بود. اتوماتای یادگیر سلولی[10] مدل جدیدی است که در آن اتوماتای سلولی با اتوماتای یادگیر تلفیق گشته است و بنابراین مزایای یادگیری و رفتار براینده را داراست. برای اطلاعات بیشتر به [2] و [3] و جهت آشنایی با برخی کاربردهای اتوماتای یادگیر سلولی به [4] و [5] مراجعه کنید. همچنین در [6] یادگیری و رفتار براینده به عنوان دو ویژگی مهم این مدل، بصورت کاربردی مورد مطالعه قرار گرفته است.

انواع مختلفی از یادگیری‌ها در جهت افزودن توانمندی‌های اتوماتای سلولی با آن تلفیق شده است. از جمله نمونه‌های خاصی از یادگیری تقویتی که در [7] راجع به آنها بحث شده است. یادگیری Q، نوعی دیگر از یادگیری تقویتی است که در [8] با اتوماتای سلولی تلفیق گشته است. یادگیری اختلاف زمانی، نوع خاصی از یادگیری تقویتی است و در آن، همانند دیگر انواع یادگیری های تقویتی، وظیفه عامل پیدا کردن سیاست Π است که نگاشت حالتها به اعمال را به نحوی انجام می‌دهد که مقدار پاداش دریافتی در بلند مدت بیشینه شود. محیط، عموماً غیر قطعی در نظر گرفته می‌شود، یعنی انجام یک عمل در یک حالت، در دو موقعیت متفاوت، می‌تواند منجر به حالتهای بعدی متفاوت و/یا دریافت پاداش متفاوت شود. فرض دیگر آن است که محیط ثابت[11] باشد، یعنی احتمال انجام تغییر حالت و یا دریافت سیگنالهای تقویتی خاص در طی زمان تغییر نکند.

اتوماتای اختلاف زمانی سلولی از تلفیق دو مدل اتوماتای سلولی و یادگیری اختلاف زمانی حاصل می‌شود. در واقع برای افزایش قابلیت‌های اتوماتای سلولی در حل مسائل پیچیده، به هر سلول آن، ویژگی یادگیری اختلاف زمانی اعطاء می‌شود. اجزاء این سیستم، مشابه اجزای اتوماتای سلولی است، با این تفاوت که هر سلول، یک مکانیسم یادگیری نیز دربردارد. به عبارت دیگر، هر سلول را می‌توان یک یادگیر اختلاف زمانی محسوب کرد. هر مدل یادگیری اختلاف زمانی، شامل مجموعه‌ای از حالت‌ها، اعمال و تابع توزیع احتمالی انتقال از یک حالت به حالات دیگر، پس از انجام یک عمل است. سیگنال تقویتی[12] برای هر عمل در هر سلول، بر اساس اعمالی که همزمان در سلولهای همسایه انجام شده‌اند، تعیین می‌شود. هر اتوماتای اختلاف زمانی سلولی، دارای یک قانون تقویتی است.

ادامه این مقاله بدین صورت سازماندهی شده است. در بخش دوم اتوماتای سلولی به اختصار مورد مطالعه قرار می‌گیرد. در بخش سوم یادگیری اختلاف زمانی بررسی می‌شود. در بخش چهارم، مدل پیشنهادی، یعنی اتوماتای اختلاف زمانی سلولی، تعریف شده و مشخصات و الگوریتم‌های مورد استفاده با جزئیات تبیین می‌گردد. بخش پنجم به بررسی مسأله شکار و شکارچی اختصاص دارد و سرانجام در بخش ششم نتایج آزمایش‌ها گردآوری شده است. بخش نهایی، نتیجه‌گیری است.

---

[8] Learning Automata (LA)
[9] Tsetlin
[10] Cellular Learning Automata (CLA)
[11] Stationary
[12] Reinforcement Signal

۲. اتوماتای سلولی

اتوماتای سلولی شبکه‌ای از سلول‌هاست که هر کدام می‌توانند k وضعیت مختلف داشته باشند. سلول‌ها در اتوماتای سلولی می‌توانند در شبکه‌ای با هر ابعادی قرار گیرند. تابع Φ که تعیین‌کننده وضعیت هر سلول در زمان t+1 یعنی $a_i^{(t+1)}$ براساس وضعیت همسایه‌های آن سلول (و احتمالاً وضعیت خود آن سلول) در زمان t است، قانون اتوماتای سلولی نامیده می‌شود.

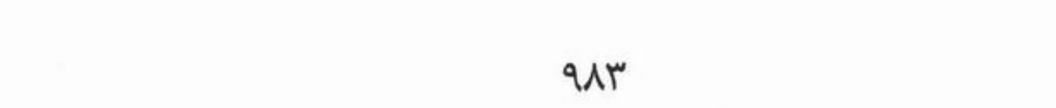
a_i ها: همسایه‌های a_j $$a_i^{(t+1)} = \Phi(a_j^{(t)}) \quad (۱)$$

در گروهی از قوانین، مقدار یک سلول در مرحله بعدی به مقادیر همسایه‌ها و در گروهی دیگر، تنها به مجموع همسایه‌ها وابسته است. قوانین گروه اول، عمومی[۱۳] و قوانین گروه دوم، مجموعی[۱۴]، نامیده می‌شوند. برای نمایش قوانین عمومی معمولاً از شماره‌گذاری استفاده می‌کنند و قوانین مجموعی را به کمک توابع نشان می‌دهند. هر دسته از این قوانین برای حل مجموعه‌ای از مسائل مناسب هستند و لذا براساس نوع مسأله، قانون معینی انتخاب می‌شود.

ویژگی‌های اساسی اتوماتای سلولی عبارتند از: فضا و زمان بصورت گسسته می‌باشند، هر سلول تعداد محدودی وضعیت همگن را اختیار می‌کند، تمام سلول‌ها یکسان هستند، بروز درآوردن سلول‌ها همگام است، قوانین بطور قطعی اعمال می‌شوند، قانون انتخاب شده در هر سلول فقط به مقدار همسایه‌های اطراف آن بستگی دارد و مقدار جدید هر سلول فقط بستگی به مقادیر تعداد محدودی (معمولاً یک مرحله) از مراحل قبل دارد [9]. یکی از مهمترین ویژگیهای اتوماتای سلولی، رفتار برایندی[۱۵] است. پیتر کاریانی[۱۶]، در [10]، رفتار برایندی را این گونه تعریف کرده است: "*رفتار برایندی، شامل خلق ساختارها و رفتارهای کیفی جدیدی است که قابل تجزیه به اجزای تشکیل دهنده خود نمی‌باشند*".

۳. یادگیری اختلاف زمانی

در مدل استاندارد یادگیری تقویتی[۱۷]، یک عامل[۱۸] با محیطش تعامل دارد. تعامل به این صورت است که عامل محیط را حس می‌کند و براساس ورودی گرفته شده از حسگرها، یک عمل انتخاب می‌کند. عمل انتخابی، به نحوی باعث تغییر در محیط می‌شود و این تغییرات از طریق یک سیگنال تقویتی عددی به عامل گزارش داده می‌شود. هر مسأله یادگیری تقویتی دارای سه جزء است: محیط، تابع تقویتی[۱۹] و تابع ارزش[۲۰] [11]. در ابتدا تخمین تابع ارزش بهینه، یعنی نگاشت از حالت‌ها به ارزش‌های حالت‌ها معتبر نیست. هدف اولیه یادگیری پیدا کردن نگاشت صحیح است. هنگامی که به تابع ارزش بهینه دست یافتیم، سیاست بهینه را بسادگی می‌توانیم از آن استخراج کنیم. زنجیرهٔ مارکوف شکل ۱ را در نظر بگیرید. حالت اولیه، صفر و حالت پایانی ۹۹۹ است. هر انتقال، هزینه (پاداش) دارد و ارزش حالت ۹۹۹، صفر است.



شکل ۱ : زنجیره مارکوف با ۱۰۰۰ حالت؛ صفر، حالت شروع و ۹۹۹ حالت پایانی است.

[13] General
[14] Totalistic
[15] Emergent Behavior
[16] Peter Cariani
[17] Reinforcement Learning
[18] Agent
[19] Reward Function
[20] Value Function

هدف یادگیری تقویتی، پیش‌بینی پاداش کلی دریافت شده به هنگام شروع از یک حالت n است (n حالتی است در بازه [۹۹۸..۱]). TD(λ) این مسأله را سریعتر از دیگر روشهای یادگیری تقویتی حل می‌کند، چرا که TD(λ)، به جای آنکه تنها از ارزش حالت بعدی استفاده کند، ارزش حالت جاری را براساس ترکیبی وزن‌دار از ارزش حالتهای آتی، بروز می‌رساند. $0 \le \lambda \le 1$، ضریب وزن‌دهی است. رابطه زیر، نمایش‌دهنده نحوه بروزرسانی پارامتر وزن، در TD(λ) است.

$$\Delta w_t = \alpha(r(x_t) + V(x_{t+1}, w_t) - V(x_t, w_t)) \sum_{k=1}^{t} \lambda^{t-k} \nabla_w V(x_k, w_t) \qquad (۲)$$

در این رابطه، α، نرخ یادگیری، x_t و x_{t+1}، بترتیب حالت فعلی و حالت بعدی، r، تابع تقویتی و V، تخمین تابع ارزش است. در معادله بروزرسانی TD(λ)، عبارت بیشینه یا کمینه وجود ندارد. این بدان مفهوم است که TD(λ)، تنها در زمینه پیش‌بینی (در زنجیره‌های مارکوف) استفاده می‌شود. یک راه تعمیم TD(λ) به حیطه فرآیندهای تصمیم‌گیری مارکوف این است که بروزرسانی‌ها را بر طبق معادله TD(λ) انجام دهیم در حالیکه مجموع را بر طبق معادله زیر با دنبال کردن سیاست فعلی بدست می‌آوریم [12]،[13]. g، همان عبارت Σ در رابطه (۲) است.

$$\begin{aligned} g_{t+1} &= \sum_{k=1}^{t+1} \lambda^{t+1-k} \nabla_w V(x_k, w_t) \\ &= \nabla_w V(x_{k+1}, w_t) + \sum_{k=1}^{t} \lambda^{t+1-k} \nabla_w V(x_k, w_t) \\ &= \nabla_w V(x_{k+1}, w_t) + \lambda g_t \end{aligned} \qquad (۳)$$

۴. اتوماتای اختلاف زمانی سلولی

اتوماتای اختلاف زمانی سلولی، مدلی برای سیستم‌هایی است که از اجزاء ساده‌ای تشکیل شده‌اند و رفتار هر جزء بر اساس رفتار همسایگان آن جزء و نیز تجربیات گذشته‌اش، تعیین و اصلاح می‌شود. یک اتوماتای اختلاف زمانی سلولی، دارای سه خصوصیت اصلی زیر است:

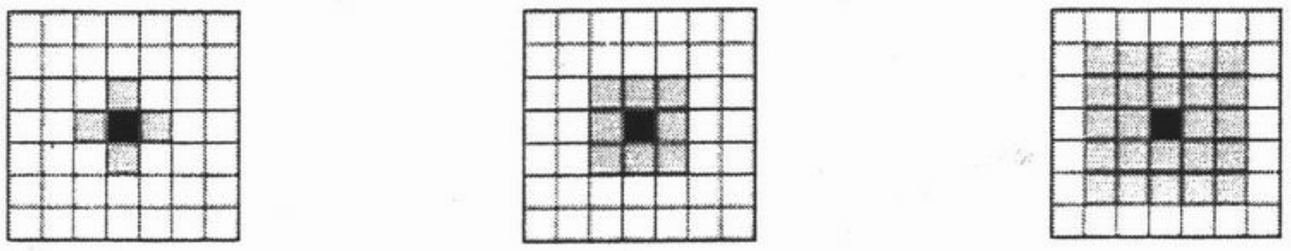
- یک شبکه *n-بعدی منظم*[۲۱] (n غالباً یک یا دو است) که هر *سلول* این شبکه یک حالت گسسته دارد،
- یک *رفتار پویا* که از آن با عنوان *قوانین*، نام می‌بریم. این قوانین، سیگنال تقویتی ارسالی به هر یک از سلول‌ها را در گام بعدی، بر اساس حالت همسایه‌های آن سلول (ممکن است خود سلول نیز شامل شود) تعیین می‌کنند.
- *یادگیری اختلاف زمانی* در هر سلول، که مستقل از سلولهای دیگر است؛ اگر چه در تمام سلول‌ها دارای پارامترهای یکسان می باشد. *سیگنال تقویتی*، بر اساس قوانین CTDA، به هر سلول ارسال می‌شود.

لازم به ذکر است که در این مدل، الگوریتم Sarsa که حالت خاصی از یادگیری اختلاف زمانی است برای فرایند یادگیری سلول‌ها به کار گرفته شده است، که در ادامه به توصیف آن خواهیم پرداخت. نکته‌ای که در شبکه CTDA حائز اهمیت است، تعریف مرزهای شبکه است. مرزهای شبکه، ممکن است ایستا[۲۲] یا پریودیک[۲۳] باشند. چنانچه مرزهای شبکه، ایستا باشد، شبکه به مرزها، محدود است. اما در شبکه با مرزهای پریودیک، مرزهای شبکه بر روی همدیگر قرار می گیرد و یک لبه در کنار لبه دیگر در نظر گرفته می‌شود. هم‌چنین ممکن است، یک شبکه هیچ مرزی نداشته باشد و نامحدود درنظر گرفته شود.

---

[21] Regular n-Dimensional Lattice
[22] Static
[23] Periodic

برای پویا نمودن سیستم، لازم است *قوانین* را بدان بیفزاییم. وظیفه قوانین، تعیین سیگنال تقویتی ارسالی به سلول‌ها در هر گام زمانی است. قوانینی که سیگنال تقویتی هر سلول را مشخص می‌سازند، وابسته به وضعیت همسایه‌های آن سلول هستند. همسایگی را می‌توان به شکل‌های مختلف، تعریف نمود. تعریف همسایگی، به شرایط مسأله مورد نظر بستگی دارد. در ابعاد مختلف، همسایگی گونه‌های مختلفی پیدا می‌کند. برای مثال، در یک شبکه دو بعدی، می توان تعاریف متعارف زیر را که در اتوماتای سلولی وجود دارند، برای همسایگی به کار برد.



شکل ۲ : چند همسایگی متداول، از راست به چپ؛ وون نیومن توسعه‌یافته، مور[۲۴]، وون نیومن

قوانین، در اتوماتای اختلاف زمانی سلولی، بر اساس وضعیت همسایه‌های هر سلول، مقدار سیگنال تقویتی ارسالی به آن سلول را مشخص می‌سازند. وضعیت سلول‌های همسایه، ممکن است عمل انتخابی توسط آنها، در گام زمانی فعلی باشد که غالباً همین‌گونه است، یا ممکن است حالت فعلی این سلول‌ها مدنظر باشد. ویژگی‌های مسأله مورد بررسی، تعیین‌کننده نحوه تعریف قوانین است. این قوانین، از طریق ارسال سیگنال تقویتی مناسب به هر سلول، سبب هدایت فرایند یادگیری سلولی می‌شوند و در نهایت سیستم را به سمت پاسخ مسأله، سوق می‌دهند. قوانین اتوماتای اختلاف زمانی سلولی را همانند قوانین اتوماتای سلولی به دو دسته تقسیم می‌کنیم:

- قوانین عمومی[۲۵]: هر گروه از وضعیت‌های سلول‌های همسایه، به یک وضعیت سلول مرکزی، نگاشت می‌شود. برای مثال، یک CTDA یک بعدی را درنظر بگیرید: قانون "$x-1x\rightarrow 011$"، بدین مفهوم است که سلول مرکزی، تنبیه می شود (سیگنال تقویتی منفی، ۱-، دریافت می‌کند)، اگر سلول سمت چپ آن، در حالت صفر، سلول سمت راست آن، در حالت یک و خود سلول در حالت یک باشد؛ یا می‌توان گفت، سلول مرکزی، تنبیه می‌شود، اگر سلول سمت چپ، عمل صفر، سلول سمت راست، عمل ۱ و خود سلول نیز عمل شماره یک را انجام داده باشد.
- قوانین مجموعی[۲۶]: سیگنال تقویتی ارسالی به سلول مرکزی، تنها به مجموع حالتهای سلولهای همسایه وابسته است. معمولاً قوانین مجموعی، به شکل تابع تعریف می‌شوند. برای مثال، چنانچه سیگنال تقویتی ارسالی به سلول i را در لحظه t، با r_t^i و اعمال انجام‌شده توسط سلولهای همسایه این سلول را در لحظه t با a_t نشان دهیم، رابطه ذیل یک قانون مجموعی می‌باشد (در آن φ یک تابع معین است).

$$r_t^i = \varphi(a_t^{i-1}, a_t^i, a_t^{i+1}) \qquad (۴)$$

در مدل پیشنهادی، هدف ما این است که از پیمایش شایستگی[۲۷]، نه تنها به منظور پیش‌بینی، بلکه برای کنترل استفاده کنیم. ایده اصلی، یادگیری ارزش اعمال، $Q_t(s,a)$، بجای ارزش حالت‌ها، $V_t(s)$، است. در اینجا، با ترکیب پیمایش شایستگی و الگوریتم Sarsa، راهی برای ایجاد یک الگوریتم کنترل TD با سیاست برخط[۲۸]، معرفی می‌کنیم. ایده Sarsa(λ)، بکارگیری روش پیش‌بینی TD(λ) برای دوتایی‌های حالت-عمل، بجای حالت تنها است. واضح است که در چنین شرایطی، ردگیری برای هر حالت کفایت نمی‌کند و لازم است هر دوتایی حالت-عمل، ردگیری شود. فرض کنید $e_t(s,a)$، بیانگر ردگیری

[24] Moore
[25] General Rules
[26] Totalistic Rules
[27] Eligibility Traces
[28] On-Policy TD Control Method

دوتایی حالت-عمل، s,a، است. الگوریتم مورد استفاده، دقیقاً همانند TD(λ) است، بجز آنکه متغیرهای حالت-عمل، جایگزین متغیرهای حالت می‌شوند (یعنی $Q(s,a)$ جایگزین $V(s)$ و $e_t(s,a)$ جایگزین $e_t(s)$ می‌شود):

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha\delta_t e_t(s,a),\ for\ all\ s,a \quad (۵)$$

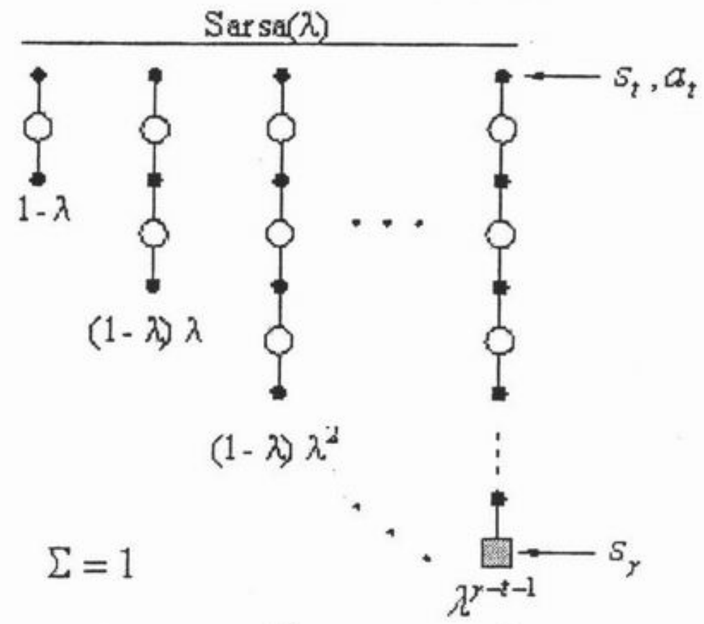
که در آن:

$$\delta_t = r_{t+1} + \gamma Q_t(s_{t+1},a_{t+1}) - Q_t(s_t,a_t) \quad (۶)$$

و:

$$e_t(s,a) = \begin{cases} \gamma\lambda e_{t-1}(s,a)+1 & if\ s=s_t\ and\ a=a_t; \\ \gamma\lambda e_{t-1}(s,a) & otherwise. \end{cases} \quad for\ all\ s,a \quad (۷)$$

شکل ۳، نمودار عملکرد الگوریتم Sarsa(λ) را نمایش می‌دهد. در اولین مرحله، الگوریتم به اندازه یک گام کامل جلوتر را نظاره می‌کند تا دوتایی حالت-عمل بعدی را ببیند، در دومین مرحله، دو گام جلوتر را بررسی می‌کند و الی‌آخر. آخرین مرحله، برپایه بازگشت کامل قرار گرفته است. ارزش هر مرحله، دقیقاً مشابه TD(λ)، محاسبه می‌شود.



شکل ۳ : نمودار الگوریتم Sarsa

الگوریتم Sarsa(λ)، یک الگوریتم از نوع سیاست برخط می‌باشد، بدین مفهوم که در آن، ارزش هر عمل برای سیاست فعلی π، یعنی $Q^{\pi}(s,a)$، تخمین زده می‌شود، آنگاه سیاست را بتدریج و براساس ارزشهای تخمینی سیاست فعلی، اصلاح می‌کند. اصلاح سیاست، از طرق مختلف امکانپذیر است، به عنوان مثال، ساده‌ترین رهیافت، استفاده از روش ε-greedy، با توجه به تخمین‌های فعلی عمل-ارزش است. در ذیل، شبه کد الگوریتم Sarsa که الگوریتم یادگیری به کارگرفته شده در هر سلول است، مشاهده می‌شود [14].

Initialize ***Q(s,a)*** *arbitrarily and* ***e(s,a) = 0****, for all* ***s, a***
Repeat (for each episode):
  Initialize ***s, a***
  Repeat (for each step of episode):
    Take action ***a****, observe* ***r****, s´*
    Choose a´ from s´ using policy derived from ***Q*** *(e.g., ε-greedy)*
    $\delta \leftarrow r + \gamma Q(s',a') - Q(s,a)$
    $e(s,a) \leftarrow e(s,a) + 1$
    For all ***s, a****:*
      $Q(s,a) \leftarrow Q(s,a) + \alpha\delta e(s,a)$
      $e(s,a) \leftarrow \gamma\lambda e(s,a)$
    $s \leftarrow s'$; $a \leftarrow a'$
  until ***s*** *is terminal*

الگوریتم Sarsa(λ)، یک الگوریتم از نوع سیاست برخط می‌باشد، بدین مفهوم که در آن، ارزش هر عمل برای سیاست فعلی π، یعنی $Q^{\pi}(s,a)$، تخمین زده می‌شود، آنگاه سیاست را بتدریج و براساس ارزشهای تخمینی سیاست فعلی، اصلاح می‌کند. اصلاح سیاست، از طرق مختلف امکانپذیر است، به عنوان مثال، ساده‌ترین رهیافت، استفاده از روش ε-greedy، با توجه به تخمین‌های فعلی عمل-ارزش است. در ذیل، شبه کد الگوریتم Sarsa که الگوریتم یادگیری به کارگرفته شده در هر سلول است، مشاهده می‌شود [14].

اینک به تعریف رسمی اتوماتای اختلاف زمانی سلولی، می‌پردازیم. یک اتوماتای اختلاف زمانی سلولی، CTDA، بصورت چهارتایی $\{X, A, \Omega, \Phi\}$ تعریف می‌شود که در آن، $X = \{\chi_1, \chi_2, ..., \chi_n\}$، مجموعه سلول‌ها،

$A=\{\alpha_1,\alpha_2,...,\alpha_k\}$، مجموعه اعمال مجاز اتوماتا، Ω، رابطه همسایگی سلول‌ها و Φ، قانون تقویتی حاکم بر اتوماتا است. Ω، رابطه همسایگی، رابطه‌ای است که برای هر سلول، بر اساس نوع همسایگی تعریف شده و مرزهای شبکه CTDA، همسایه‌های آنرا مشخص می‌سازد. برای مثال در یک اتوماتای دوبعدی، با همسایگی وون نیومن، داریم:

$$\Omega(\chi_{i,j})=\{\chi_{i-1,j},\chi_{i,j-1},\chi_{i,j+1},\chi_{i+1,j}\} \quad (۸)$$

در همسایگی، معمولاً خود سلول را درنظر نمی‌گیریم. به عبارت دیگر، رابطه همسایگی، دارای ویژگی‌های زیر است:

$$\begin{cases}\chi_i\notin\Omega(\chi_i); & \forall\chi_i\in X\\ \chi_i\in\Omega(\chi_j), iff\ \chi_j\in\Omega(\chi_i); & \forall\chi_i,\chi_j\in X\end{cases} \quad (۹)$$

هم‌چنین، برای هر سلول، Ψ، را به شکل زیر تعریف می‌کنیم:

$$\Psi(\chi_i)=\Omega(\chi_i)\cup\{\chi_i\} \quad (۱۰)$$

هر سلول χ_i، در اتوماتای اختلاف زمانی سلولی بصورت پنج‌تایی $\{\Sigma,\Delta,\sigma,\Gamma,P\}$ نشان داده می‌شود که در آن، $\Sigma=\{\sigma_1,\sigma_2,...,\sigma_r\}$، حالات ممکن هر سلول است. این مجموعه، برای سلول‌های متفاوت می‌تواند متفاوت باشد، اگرچه در این مقاله، برای تمام سلول‌ها یکسان درنظر گرفته شده است. σ، حالت اولیه سلول، عضوی از مجموعه Σ است. هم‌چنین، Γ، حالات نهایی سلول، زیرمجموعه‌ای از این مجموعه می‌باشد. Δ، مجموعه اعمال مجاز برای هر سلول است که با مجموعه اعمال مجاز اتوماتا، A، رابطه‌ای به شکل $\Delta\subseteq A$ دارد. در نهایت، Γ، تابع توزیع احتمالی تغییر حالت هر سلول است. هر سلول، که در زمان t، در حالت σ_i قرار دارد، پس از انجام عمل δ_i، به حالت یا یکی از حالات در زمان t+1 منتقل می‌شود. چنانچه اتوماتا، قطعی باشد، هر سلول در هر گام زمانی، با احتمال ۱، به یک حالت جدید، تغییر وضعیت می‌دهد. اما اگر اتوماتا، غیرقطعی باشد، تابع توزیع احتمال Γ، تعیین می‌کند که شانس رسیدن به هر حالت، چقدر است. در زیر، تعریف این تابع را مشاهده می‌کنید (در آن $\sum_{i=1}^{r}\rho_i=1$):

$$\Gamma(\sigma_i^t,\alpha_j^t)=\begin{cases}\{(\sigma_k^{t+1},1)\}, & if\ CTDA\ is\ Deterministic\\ \{(\sigma_1^{t+1},\rho_1),(\sigma_2^{t+1},\rho_2),...,(\sigma_r^{t+1},\rho_r)\}, & if\ CTDA\ is\ nonDeterministic\end{cases} \quad (۱۱)$$

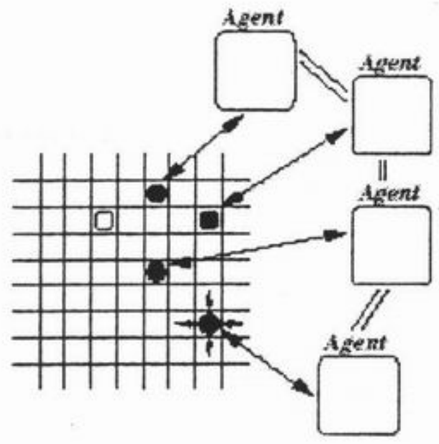
هر سلول، دارای یک الگوریتم یادگیری Sarsa(λ) است. حال فرض کنید عمل انجام شده توسط سلول i، در زمان t، با $A^t(\chi_i)$ نمایش داده شود. قانون تقویتی که یک قانون عمومی یا مجموعی می‌باشد، براساس اعمال انجام شده توسط همسایه‌های سلول در همان زمان t، یک سیگنال تقویتی به سلول ارسال می‌نماید. سیگنال تقویتی ارسال شده به هر سلول در زمان t، $R^t(\chi_i)$ بصورت زیر تعریف می‌شود:

$$R^t(\chi_i)=\Phi\{A^t(\chi)|\chi\in\Psi(\chi_i)\} \quad (۱۲)$$

۵. مسأله شکار و شکارچی (تعقیب)

مسأله تعقیب[۲۹]، نخستین‌بار توسط بندا[۳۰] و دیگران [15] مطرح شد و طی سالیان متمادی، پژوهشگران گونه‌های متعددی از مسألهٔ اولیه را مورد مطالعه قرار دادند. در اینجا، تنها به ذکر نمونهٔ ساده‌ای از مسأله بسنده می‌کنیم. برای اطلاع از دیگر حالات مسأله، به [16] رجوع کنید. مسألهٔ تعقیب، معمولاً با چهار شکارچی و یک شکار مورد مطالعه قرار می‌گیرد. به شکل تاریخی، شکارچی‌ها، آبی و شکار قرمز است (در شکل ۴ بترتیب سیاه و سفید).

[29] The Predator/Prey (Pursuit) Domain
[30] Benda

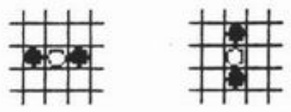


شکل ۴: یک نمونه معمول از مسأله تعقیب. جهت نماهای اطراف شکارچی حرکتهای ممکن را نمایش می‌دهد.

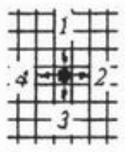
برای سهولت بیشتر، دو شکارچی و یک شکار در نظر می‌گیریم و هم‌چنین فرض‌های زیر را در مسأله به کار می‌گیریم:

- شکارچی‌ها، همدیگر را می‌بینند.
- شکار بصورت تصادفی حرکت می‌کند (در حالت پیچیده‌تر، شکار همیشه از شکارچی‌ها می‌گریزد).
- حرکت‌ها بصورت همزمان و گسسته می‌باشد.

هدف شکارچی‌ها، "گرفتن" شکار یا احاطه کردن آن به نحوی است که نتواند به یک فضای آزاد بگریزد. حالت‌های مختلف گرفتن، در شکل ۵ نمایش داده شده است.

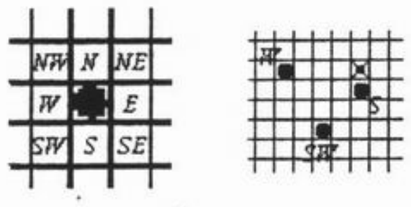


شکل ۵: گرفتن شکار

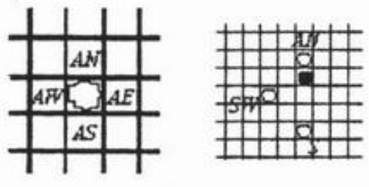


شکل ۶: حرکتهای مجاز شکارچی

با بیانی از مسأله که ارائه نمودیم، اینک به مدلسازی مسأله برای حل آن به کمک CTDA می‌پردازیم. برای مدلسازی مسأله، یک CTDA با یک سطر و دو ستون بکار می‌گیریم. هر سلول این CTDA معادل یکی از شکارچی‌ها (عامل‌ها) است. با بکارگیری CTDA، یادگیری هر عامل از نوع TD(λ) خواهد بود. عامل‌ها دارای چهار حرکت مجاز هستند که در شکل ۶، نمایش داده شده است. هدف مشترک عامل‌ها، گرفتن شکار است که پیشتر آنرا تعریف نمودیم. حالت‌های داخلی هر شکارچی، براساس موقعیت آن در دنیا و مشاهداتش تعیین می‌شود. هر شکارچی در دنیا، شکارچی دیگر و شکار را در موقعیتی نسبت به خود مشاهده می‌کند. شکارچی دیگر در یکی از ۹ موقعیت ممکن نسبت به آن قرار دارد. این موضوع در شکل ۷ نمایان است.



شکل ۷: موقعیت هر شکارچی نسبت به دیگری



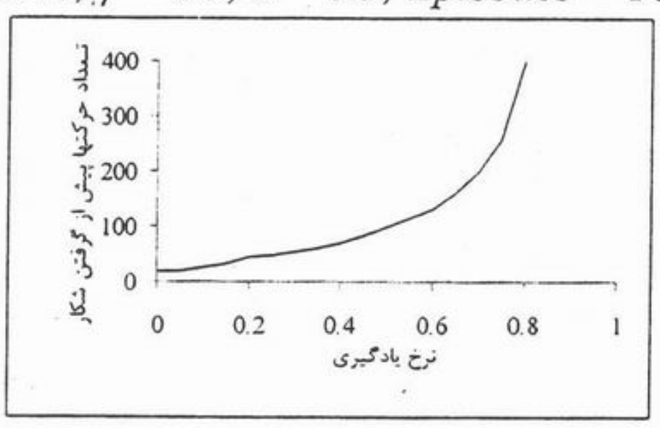
شکل ۸: موقعیت شکار نسبت به شکارچی

شکار نیز در یکی از این ۹ موقعیت نسبت به هر شکارچی واقع است یا ممکن است دقیقاً در ۴ موقعیت مجاور یک شکارچی قرار داشته باشد و امکان گرفتن آن برای شکارچی وجود داشته باشد. هم‌چنین حالت نهایی یک شکارچی که گرفتن شکار است به مجموعه حالت‌ها افزوده می‌شود. لذا هر شکارچی دارای $(9+4)\times 9+1=13\times 9+1=118$ حالت مختلف است و ۴ عمل مجاز دارد. حالت اولیه هر شکارچی، با قرار گرفتن آن در محیط بصورت تصادفی، تعیین می‌شود و حالت نهایی آن گرفتن شکار است. با انتخاب هر عمل در هر حالت، حالت نتیجه نیز مشخص است. سیستم تنها در صورتی به شکارچی‌ها پاداش می‌دهد که هر دو در حالت گرفتن شکار قرار داشته باشند.

۶. نتایج آزمایشها

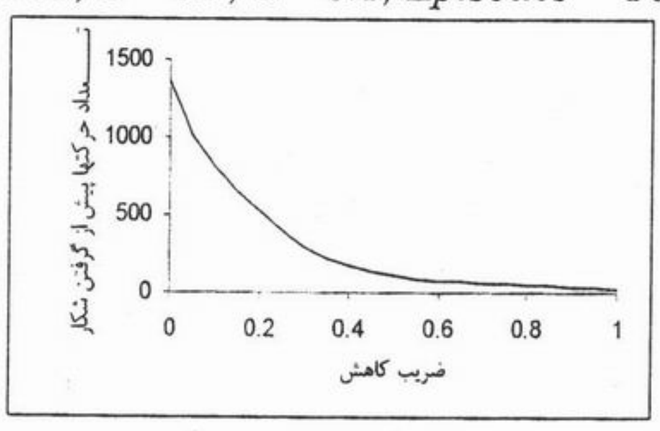
پارامترهای متغیر برنامه، عبارتست از: نرخ یادگیری، α، ضریب کاهش، γ، احتمال اکتشاف، ε، لامبدا، λ و تعداد دورهای یادگیری در هر سلول، پیش از رسیدن به هدف. با ترتیب دادن آزمایشهایی بر اساس مدل مطرح شده برای مسأله تعقیب، به تغییر هر یک از پارامترهای فوق می‌پردازیم و رفتار سیستم را بررسی می‌کنیم. لازم است توجه شود که هنگام تغییر مقدار هر پارامتر، مقدار پارامترهای دیگر ثابت و مشخص در نظر گرفته شده است. آزمایش‌ها، برای هر پارامتر، چندین بار انجام شده و در نمودارها، میانگین نتایج مورد استفاده قرار گرفته است. در ادامه، نتایج این آزمایشها را مشاهده می‌کنیم.

$\varepsilon = 0.01, \gamma = 0.9, \lambda = 0.9, Episodes = 10000$



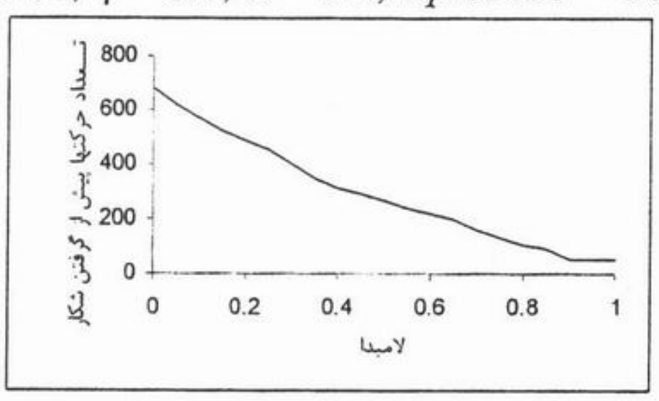
شکل ۹ : تأثیر تغییر α در رفتار CTDA

$\varepsilon = 0.01, \lambda = 0.9, \alpha = 0.1, Episodes = 10000$



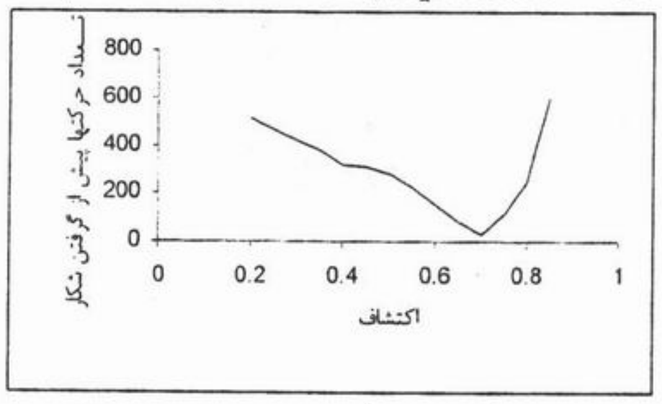
شکل ۱۰ : تأثیر تغییر γ در رفتار CTDA

$\varepsilon = 0.01, \gamma = 0.9, \alpha = 0.1, Episodes = 10000$



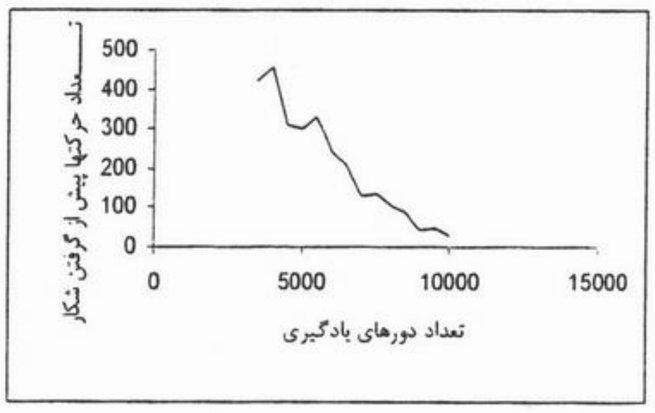
شکل ۱۱ : تأثیر تغییر λ در رفتار CTDA

$\gamma = 0.9, \lambda = 0.9, \alpha = 0.1, Episodes = 10000$



شکل ۱۲: تأثیر ε در رفتار CTDA

$\varepsilon = 0.01, \gamma = 0.9, \lambda = 0.9, \alpha = 0.1$



شکل ۱۳: تأثیر تعداد دورهای یادگیری بر رفتار CTDA

نرخ یادگیری، پارامتری در یادگیری‌های تقویتی است که میزان تأثیرگذاری یادگیری را در سیاست فعلی تعیین می‌کند. نرخ یادگیری، α، عددی بین ۰ و ۱ است که هر چه میزان آن کمتر باشد، به مفهوم تأثیرگذاری کندتر یادگیری در سیاست فعلی برای انتخاب اعمال است و هر چه میزان آن به ۱ نزدیک‌تر باشد، سبب استفاده سریعتر از یادگیری در سیاست انتخاب اعمال می‌گردد. اثر تغییر این پارامتر، در شکل ۹ مشاهده می‌شود.

ضریب کاهش، γ، تعیین کننده میزان تأثیر پاداش‌های آینده است. هرچه این ضریب، به صفر نزدیکتر باشد، سیگنال‌های تقویتی نزدیک‌تر در فرایند یادگیری در نظر گرفته می‌شوند. به عکس هرچه این ضریب بزرگ‌تر باشد، سیگنال‌های تقویتی در زمان،

بیشتر منتشر می‌شوند و در حالت‌های گذشته بیشتری تأثیر می‌گذارند. در شکل ۱۰، تأثیر تغییر ضریب کاهش در رفتار سیستم، آمده است. لامبدا، λ، ضریبی در یادگیری اختلاف زمانی است که در بازه [۱..۰] تعریف می‌شود. تأثیر تغییرات این ضریب را در رفتار سیستم، در شکل ۱۱، مشاهده می‌کنیم.

احتمال اکتشاف، ε، تعیین کننده میزان اکتشافی بودن انتخاب اعمال در مقابل استفاده از تجربه گذشته است. هر چه میزان این احتمال کمتر باشد، اعمال، بیشتر بر اساس تجربه به دست آمده برگزیده می‌شوند و هرچه میزان این احتمال بیشتر باشد، از تصادف در انتخاب اعمال بهره بیشتری گرفته می‌شود. شکل ۱۲ نشان‌دهنده تأثیر تغییرات این پارامتر در رفتار سیستم است. همانگونه که قابل پیش‌بینی است، چنانچه ε به مرزهای بازه نزدیک باشد، پاسخ مسأله دیرتر به‌دست می‌آید (ممکن است اصلاً به دست نیاید). چنانچه ε به صفر نزدیک باشد، سیستم، تنها از تجربه به دست آمده برای انتخاب اعمال خود استفاده می‌کند. به همین علت، ممکن است برخی اعمال مفید، هیچ گاه در نظر گرفته نشوند. در چنین شرایطی، ممکن است دفعات بیشتر یادگیری، سبب همگرا شدن به سمت پاسخ گردد. به عکس، هنگامی که ε به ۱ نزدیک است، از تجربه گذشته، در انتخاب اعمال استفاده چندانی نمی‌شود. بنابراین می‌توان حدس زد که فرایند یادگیری، کاربرد چندانی پیدا نمی‌کند.

در شکل ۱۳، تأثیر تعداد دورهای یادگیری در رفتار اتوماتای اختلاف زمانی سلولی، مشاهده می‌شود. همانگونه که واضح است، دورهای بیشتر یادگیری، غالباً در همگرایی سریعتر به سمت پاسخ، مؤثر است، اگر چه لزوماً اینگونه نیست. مسأله اصلی که در افزایش تعداد دورهای یادگیری با آن مواجه هستیم، کاهش قابل ملاحظه سرعت سیستم است؛ به نحوی که گاهی امکان ادامه عملیات یادگیری، غیرممکن شود. از طرف دیگر، تعداد کم دورهای یادگیری، سیستم را به سمت پاسخ همگرا نمی‌کند. بنابراین، لازم است بین این دو مرز، یک نقطه مناسب، پیدا کنیم که ضمن آنکه ما را به پاسخ می‌رساند، ضرورتاً بهینه‌ترین پاسخ به دست نمی‌دهد.

در دیگر آزمایشهای مشابه، غالباً ارتباط بین تعداد شکار و شکارچی، مدنظر بوده است. لذا، معیار مناسبی برای مقایسه وجود ندارد [16]. اما همانگونه که در آزمایشهای صورت پذیرفته مشاهده می‌شود، مدل پیشنهادی، قادر به حل مسأله تعقیب، به شکلی قابل قبول می‌باشد. تنها باید توجه داشت که به دلیل آنکه همگرایی یادگیری اختلاف زمانی، اثبات نمی‌شود، ممکن است این سیستم نیز به سمت پاسخ همگرا نشود.

۷. نتیجه‌گیری

در این مقاله، مدل جدیدی تحت عنوان اتوماتای اختلاف زمانی سلولی پیشنهاد و جزئیات آن مورد مطالعه قرار گرفت. این مدل که از تلفیق یادگیری اختلاف زمانی با اتوماتای سلولی حاصل شده است، دارای سادگی اتوماتای سلولی و ویژگی‌های فوق‌العاده یادگیری اختلاف زمانی می‌باشد. با توجه به این ویژگی‌ها، مسائل متعددی توسط این مدل قابل بررسی هستند که برای نمونه، مسأله شکار و شکارچی، مورد مطالعه قرار گرفت.

۸ مراجع

[1] Alexander Schatten, Cellular Automata, Digital Worlds, 1999, Available Online At: http://www.ifs.tuwien.ac.at/Naschatt/info/ca/ca-print.html

[2] M. R. Meybodi, H. Beigy, and M. Taherkhani, Cellular Learning Automata, *Proceedings of the 6th Annual CSI Computer* Conference, University of Isfahan, 2001, Pages 153-163.

[3] H. Beigy, and M. R. Meybodi, A Mathematical Framework for Cellular Learning Automata, *Advances on Complex Systems*, Vol. 7, No. 3, , 2004, Pages 1-25.

[4] M. R. Meybodi, H. Beigy, and M. Taherkhani, Cellular Learning Automata and Its Applications, *Journal of Science and Technology*, Sharif University of Technology, No. 25, Autumn/Winter 2003-2004, Pages 54-77.

[5] M. R. Meybodi, and M. R. Kharazmi, Cellular Learning Automata and Its Application to Image Processing, *Journal of Amirkabir*, Vol. 14, No. 56A, 2004, Pages 1101-1126.

[6] M. R. Khojasteh, and M. R. Meybodi, Learning Automata as a Model for Cooperation in a Team of Agents, *Proceedings of the 8th Annual CSI Computer* Conference, Ferdowsi University, 2003, Pages 116-126.

[7] Cem Ünsal, Chapter 6, New Reinforcement Schemes for Stochastic Learning Automata, *Intelligent Navigation of Autonomous Vehicles in an Automated Highway System: Learning Methods and Interacting Vehicles Approach*, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1997, Pages 92-100.

[8] R. Rastegar, and M. R. Meybodi, Cellular Q Learning Model and Its Application to Channel Assignment in Telecommunication Cellular Networks, *Journal of Amirkabir*, 2004.

[9] Cristoph Adami, *Introduction to Artificial Life*, Springer-Verlag, New York, 1998.

[10] Peter Cariani, Emergence and Artificial Life, *Proceedings of the Second International Conference on Artificial Life*, Addison-Wesley, 1991.

[11] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore, Reinforcement Learning: A Survey, *Journal of Artificial Intelligence Research*, No. 4, 1996, Pages 237-285.

[12] Mance E. Harmon, and Stephanie S. Harmon, Reinforcement Learning: A Tutorial, US Air Force, Office of Scientific Research, June 2000, Available Online At: http://citeseer.ist.psu.edu/harmon96reinforcement.htm

[13] Richard S. Sutton, Learning to Predict by the Methods of Temporal Differences, *Machine Learning, No. 3*, 1988, Pages 9-44.

[14] Richard S. Sutton, and Andrew G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, Massachusetts, 1998.

[15] M. Benda, V. Jagannathan, and R. Dodhiawala, On Optimal Cooperation of Knowledge Sources – an empirical investigation, *Technical Report BCS-G2010-28*, Boeing Advanced Technology Center, Boeing Computing Services, Seattle, Washington, July 1986.

[16] Peter Stone, and Manuela Veloso, Multiagent Systems: A Survey from a Machine Learning Perspective, *Autonomous Robotics*, Volume 8, No. 3, July 2000.