تنظیم خودکار پارامترهای مدل یادگیری Q با استفاده از اتوماتون‌های یادگیر

محمدرضا میبدی
دانشیار

سیامک حجت
کارشناسی ارشد

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر

چکیده

در این مقاله یك رهیافت جدید برای تنظیم پارامترهای مدل های یادگیر تقویتی ارائه می شود. در این رهیافت یك یا چند مدل یادگیر ساده مسئول تنظیم پارامترهای یك مدل یادگیر پیچیده تر خواهند بود. در اینجا برای تشریح این رهیافت تنظیم پارامترهای یك مدل یادگیر به نام یادگیری Q و دسته بندی آماری بررسی می شود و با استفاده از مثال هایی نسبتاً ساده نتایج تنظیم خودکار پارامترهای این مدل مورد مطالعه قرار می گیرند. این نتایج نشان خواهند داد که:

۱) مدل هایی که پارامترهای آنها تنظیم می شود می توانند در بسیاری موارد عملکرد بهتری نسبت به مدلهایی که پارامترهای آنها تنظیم نمی شود نشان دهند.

۲) تنظیم خودکار پارامترها باعث افزایش انعطاف پذیری مدل ها می شود.

Automatic Tuning of Q-Learning Model Using Learning Automata

M. R. Meybodi
Associate Prof.

Siamak Hodjat
M. Sc.

Computer Eng. Dept. Amirkabir Univ. of Tech.

Abstract

This paper describes a general approach for automatic tuning of reinforcement learning algorithms' parameters. In this approach a reinforcement learning agent's parameters are tuned by other more simple reinforcement learning algorithms. We will explain this approach by tuning the parameters of a Q-learning and statistical clustering algorithm. The results of tuning these parameters will be described by some simple examples. Comparing the result of an algorithm using automatically tuned parameters with an algorithm which uses fixed parameters will show that the former is generally more flexible and capable of performing better in most cases.

واژه‌های کلیدی

Reinforcement Learning, Q Learning, Learning Automata, Statistical Clustering

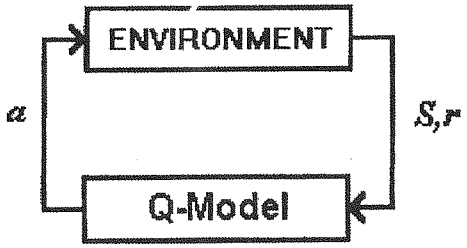
مقدمه

در یادگیری تقویتی[1] [5] [6] یک یادگیرنده در هر لحظه با اعمال فعالیتی در محیط پسخوری از آن دریافت می‌کند. هدف یادگیرنده در این محیط بیشینه کردن پسخورهای دریافت شده در طول زمان است. «یادگیری Q و دسته بندی آماری» [9] [10] (که آن را مدل Q می‌نامیم) نمونه‌ای از یک مدل یادگیری تقویتی است. مدل Q (مانند تمام مدل‌های یادگیر قدرتمند دیگر) دارای پارامترهای متعددی است و عملکرد بهینه آن به انتخاب مناسب این پارامترها وابسته است. مقادیر این پارامترها معمولاً با سعی و خطا و توسط طراح مدل انتخاب می‌شوند و در طول یادگیری به صورت مقادیر ثابت مورد استفاده قرار می‌گیرند. اما تعیین پارامترها به این روش اولاً بسیار نادقیق و وقتگیر است و ثانیاً انعطاف پذیری لازم را ندارد. بنابر این ما به یک مکانیزم برای تنظیم خودکار پارامترهای مدل یادگیر نیازمندیم. راه حلی که در این مقاله پیشنهاد می‌شود استفاده از مدل‌های یادگیر ساده‌تر برای تنظیم پارامترهای مدل‌های یادگیر پیچیده‌تر است. به این منظور برای تنظیم پارامترهای مدل Q از نمونه دیگری از مدل‌های یادگیر تقویتی به نام اتوماتون‌های یادگیر [23] استفاده شده است.

در بخش بعدی این مقاله مدل Q معرفی می‌شود. معرفی این مدل جهت آشنا شدن با نحوه کار و نقش پارامترهای آن ضروری است. این مدل عیناً همان است که در [10] جهت برنامه ریزی یک روبات رفتاری استفاده شده است. بخش سوم مقاله به معرفی چند نمونه از اتوماتون‌های یادگیر اختصاص دارد. شرح کاملتری از این اتوماتون‌ها را می‌توان در [23] پیدا کرد. در بخش چهارم روش پیشنهادی برای تنظیم پارامترهای مدل Q با استفاده از اتوماتون‌های یادگیر معرفی می‌شود و سپس در بخش ششم نتایج بعضی آزمایش‌های انجام شده به منظور ارزیابی این روش ارائه می‌شود. بخش ۷ مقاله جمعبندی می‌باشد.

۱ـ مدل Q

یادگیری Q یک نوع یادگیری تقویتی است. در یادگیری تقویتی یک یادگیرنده در هر لحظه از مجموعه فعالیت‌های ممکن فعالیتی را انتخاب کرده و در محیط اعمال می‌کند. با اعمال هر فعالیت پسخوری[2] از محیط دریافت می‌شود (شکل ۱). هدف یادگیرنده یافتن یک استراتژی انتخاب فعالیت برای ماکزیمم کردن پسخورهای دریافت شده از محیط در طول زمان است. یادگیری Q یک تکنیک برای انتشار پسخورهای بلافاصله روی توالی فعالیت‌ها می‌دهد. این الگوریتم یادگیری معمولاً به همراه روش‌های دسته بندی آماری[3] [16] استفاده می‌شود. در اینجا برای اختصار یادگیری Q با دسته بندی آماری را مدل Q می نامیم.



شکل (۱) رابطه مدل Q با محیط

در یادگیری Q از یک ساختمان داده به نام Q برای تخمین سودمندی اعمال فعالیت a در وضعیت حس شده S استفاده می‌شود: Q (S,a). ابتدا Q (S,a) برای تمام فعالیت‌های a و وضعیت‌های S برابر با صفر فرض می‌شود. سپس با اعمال هر فعالیت a در وضعیت X و دریافت پسخور بلافاصله r مقدار Q (X, a) با فرمول زیر بهنگام می‌شود:

$$Q(X, a) \leftarrow Q(X, a) + \lambda (r + \gamma e(Y) - Q(X, a)) \quad (۱)$$

در فرمول بالا Y وضعیت بعدی محیط (پس از اعمال فعالیت a در وضعیت X است) و e (Y) سودمندی وضعیت Y می‌باشد که با فرمول زیر محاسبه می‌شود: (m تعداد فعالیت‌هاست)

$$e(Y) \leftarrow \text{maximum } Q(Y, i) \text{ over all actions } i, (i = 1, 2, \ldots, m) \quad (۲)$$

پارامتر λ $(0 \le \lambda \le 1)$ میزان اصلاح خطا برای Q را تعیین می‌کند و پارامتر γ $(0 \le \gamma \le 1)$ میزان صرفنظر کردن از سودمندی وضعیت نتیجه شده را مشخص

می‌کند. در یادگیری Q تابع ارزیابی برای انتخاب بهترین فعالیت در وضعیت S باید فعالیتی را برگزیند که مقدار Q (S, a) را ماکزیمم کند. این سیاست انتخاب فعالیت‌ها هرگز تمام فعالیت‌های ممکن را امتحان نمی‌کند و معمولاً منجر به انتخاب غیر بهینه فعالیت‌ها می‌شود. بنابر این لازم است در درصدی از مواقع (θ) انتخاب فعالیت به طور تصادفی انجام گیرد [10].

زمانی که تعداد وضعیت‌های قابل تجربه زیاد باشد، به جای ذخیره کردن تمام تجربیات بهتر است آنها را دسته بندی کرد. در دسته بندی آماری تمام تجربیات مشابه در یک دسته قرار می‌گیرند و به جای ذخیره کردن همه آنها تنها اطلاعاتی آماری از آنها نگهداری می‌شود. در این تکنیک هر تجربه جدید با دسته‌های موجود مقایسه شده و در دسته (یا دسته‌های) مشابه ادغام می‌شود. در صورتی که تجربه جدید مشابه هیچکدام از دسته‌های موجود نباشد، یک دسته جدید برای آن تجربه ایجاد خواهد شد.

۱ـ۱ـ دسته‌ها

هر دسته نمایانگر گروهی از وضعیتهای مشابه است. یک دسته را می‌توان با n + 2 تایی زیر نشان داد:

$$C = < (z_1, o_1), (z_2, o_2), ..., (z_n, o_n), Q_c, M_c > \quad (۳)$$

z_i و o_i تعداد دفعاتی است که بیت iام از وضعیت S در دسته C, 0 یا 1 بوده است. n تعداد بیتهای یک وضعیت است، Q_c مقدار Q دسته را مشخص می‌کند و M_c نمایانگر تعداد تجربیاتی است که در این دسته قرار گرفته‌اند. با این نمایش می‌توان احتمال شرطی یک بودن بیت iام از وضعیت S در این دسته را با فرمول زیر محاسبه کرد: (s_i بیت i ام S است)

$$P (s_i = 1 | S \in C) = \frac{o_i}{o_i + z_i} \quad (۴)$$

۱ـ۲ـ مقایسه وضعیت‌ها با دسته‌ها

برای مقایسه وضعیت S با دسته C می‌توان از احتمال شرطی قرار گرفتن S در دسته C استفاده کرد: (v_i= 0 یا 1)

$$p (S \in C | s_1 = v_1, s_2 = v_2, ..., s_n = v_n) = \frac{p (s_1 = v_1, ..., s_n = v_n | S \in C) p (S \in C)}{p (s_1 = v_1, ..., s_n = v_n)} \quad (۵)$$

با فرض مستقل بودن بیت‌های یک وضعیت (که فرضی نادرست ولی با تقریب خوبی قابل قبول است) می‌توان مخرج کسر بالا را به صورت زیر نوشت:

$$p (s_1 = v_1, ..., s_n = v_n) = \prod_{i=1}^{n} p (s_1 = v_1) \quad (۶)$$

برای صورت کسر نیز می‌توان از فرمول زیر استفاده کرد:

$$p (s_1 = v_1, ..., s_n = v_n | S \in C) = \prod_{i=1}^{n} P (s_i = v_i | S \in C) \quad (۷)$$

سمت راست معادله (۶) را می‌توان با نگهداری اطلاعاتی آماری از حواس محاسبه کرد. مقدار p (S ∈ C) و سمت راست معادله (۷) را نیز می‌توان با استفاده از اطلاعات دسته‌ها به دست آورد. در نتیجه سمت چپ معادله (۵) قابل محاسبه خواهد بود. حال برای اینکه وضعیت S در دسته C قرار گیرد، باید داشته باشیم:

$$p (S \in C | s_1 = v_1, s_2 = v_2, ..., s_n = v_n) > \varepsilon \quad (۸)$$

$$|Q_c - Q_s| < \delta \quad (۹)$$

نامعادلات (۸) و (۹) تضمین می‌کنند ـ که اولاً مشابهت وضعیت S با دسته C از یک مقدار آستانه‌ای (ε) بیشتر باشد و ثانیاً Q محاسبه شده برای وضعیت S نسبت به مقدار Q ذخیره شده در دسته C از یک مقدار ثابت آستانه‌ای (δ) کمتر باشد.

۱ـ۳ـ ادغام وضعیت‌ها با دسته‌ها

بعد از اینکه مشخص شد که وضعیت S در دسته C قرار می‌گیرد، از آن برای بهنگام سازی دسته استفاده خواهد شد. فرض کنید:

$$C = < (z_1, o_1), (z_2, o_2), ..., (z_n, o_n), Q_c, M_c > \quad (۱۰)$$

اگر C_u نمایانگر دسته C پس از بهنگام سازی باشد و داشته باشیم:

$$C_u = < (z_{1u}, o_{1u}), (z_{2u}, o_{2u}), ..., (z_{nu}, o_{nu}), Q_{cu}, M_{cu} > \quad (۱۱)$$

برای هر بیت i از وضعیت S که برابر با 1 باشد، خواهیم داشت:

$$z_{iu} = \mu z_i,\ o_i = 1 + \mu\ o_i\ (s_i = 1) \qquad (۱۲)$$

و برای هر بیت i از وضعیت S که برابر با 0 باشد، خواهیم داشت:

$$z_{iu} = 1 + \mu\ z_i\ ,\ o_i = \mu\ o_i\ (s_i = 0) \qquad (۱۳)$$

در اینجا μ عددی حقیقی و بین صفر و یک است که برای افزایش اهمیت تجارب جدید به کار می‌رود. اگر $\mu = 1$ باشد، اهمیت تجارب جدید در نظر گرفته نخواهد شد. μ را معمولاً از فرمول $\mu = (2K - 1) / 2K$ به دست می‌آورند که در آن K عددی صحیح است (از K برای ایجاد دسته های جدید استفاده خواهد شد).

فرض کنید در وضعیت S فعالیت α اعمال شده باشد و مقدار محاسبه شده Q برای آن برابر با $Q\ (S, \alpha)$ باشد، آنگاه برای ساختن Q_{cu} می‌توان از مجموع Q_c و $Q\ (S,\alpha)$ استفاده کرد. معمولاً در این جمع از M_c به عنوان وزن استفاده می‌شود:

$$Q_{cu} = Q_u\ (\frac{M_c}{M_c + 1}) + Q\ (S, \alpha)\ (\frac{1}{M_c + 1}) \qquad (۱۴)$$

همچنین تعداد تجربیات دسته C_u به صورت زیر بهنگام می‌شود:

$$M_{cu} = M_c + 1 \qquad (۱۵)$$

۱ ـ ٤ ـ ایجاد دسته های جدید

اگر یک وضعیت S مشابه هیچ یک از دسته های موجود نباشد، باید یک دسته جدید C_{new} برای آن ساخته شود. برای اینکار ابتدا یک دسته خالی به شکل زیر ایجاد می‌گردد:

$$C = <(z_1, o_1), (z_2, o_2), ..., (z_n, o_n), Q_c, M_c> \qquad (۱۶)$$

که در آن:

$$z_i = o_i = K\ ,\ Q_c = 0\ ,\ M_c = 0 \qquad (۱۷)$$

سپس دسته خالی فوق با وضعیت S ادغام می‌شود و دسته C_{new} را می‌سازد.

۱ ـ ۵ ـ ادغام دسته های موجود

گاهی دو دسته به اندازه‌ای مشابه یکدیگرند که می‌توانند در هم ادغام شوند. برای محاسبه تشابه دو دسته می‌توان از اندازه‌گیری فاصله بین دو دسته استفاده کرد:

$$distance\ (C_1\ ,\ C_2) = \Sigma\ [\ p\ (s_i = 1 | S \in C_1) - p\ (s_i = 1 | S \in C_2)] \qquad (۱۸)$$

دو دسته C_1 و C_2 تنها زمانی با هم ادغام می‌شوند، که اولاً فاصله آنها کمتر از مقدار ثابت ρ باشد و ثانیاً مقادیر Q دو دسته اختلافی کمتر از δ داشته باشد. یعنی:

$$distance\ (C_1\ ,\ C_2) < \rho \qquad (۱۹)$$

$$|Q_{c1} - Q_{c2}| < \delta \qquad (۲۰)$$

حال اگر دو دسته زیر را داشته باشیم:

$$C_a = <(z_{1a}, o_{1a}),\ (z_{2a}, o_{2a}),\ ...,\ (z_{na}, o_{na}), Q_{ca}, M_{ca}, > \qquad (۲۱)$$

$$C_b = <(z_{1b}, o_{1b}),\ (z_{2b}, o_{2b}),\ ...,\ (z_{nb}, o_{nb}), Q_{cb}, M_{cb}, > \qquad (۲۲)$$

و این دو دسته به اندازه کافی مشابه باشند (یعنی روابط ۱۹ و ۲۰ برای آنها صادق باشد)، آنگاه دو دسته در هم ادغام می‌شوند و دسته جدید C را بوجود می‌آورند:

$$C = <(z_1, o_1),\ (z_2, o_2),\ ...,\ (z_n, o_n), Q_c, M_c> \qquad (۲۳)$$

عناصر دسته C به صورت زیر ساخته می‌شوند:

$$z_{ic} = z_{ia}\ (\frac{M_a}{M_a + M_b}) + z_{ib}\ (\frac{M_b}{M_a + M_b}) \qquad (۲۴)$$

$$o_{ic} = o_{ia}\left(\frac{M_a}{M_a + M_b}\right) + o_{ib}\left(\frac{M_b}{M_a + M_b}\right) \quad (۲۵)$$

$$Q_c = Q_a\left(\frac{M_a}{M_a + M_b}\right) + Q_b\left(\frac{M_b}{M_a + M_b}\right) \quad (۲۶)$$

$$M_c = M_a + M_b \quad (۲۷)$$

۱ـ۶ـ انتخاب فعالیت با استفاده از دسته های موجود

در یادگیری Q باید در هر وضعیت S فعالیت a را به گونه ای انتخاب کرد که مقدار Q (S,x) را به ازای تمام فعالیت های ممکن ماکزیمم کند $(x = 1, 2, ..., m)$. برای محاسبه Q (S,x) از روی دسته های موجود می توان از فرمول زیر استفاده کرد:

$$Q(S, x) = \frac{\sum_{C \in C_x} [Q_c \times P(S \in C \mid S_1 = V_1, ..., S_n = V_n)]}{\sum_{C \in C_x} [P(S \in C \mid S_1 = V_1, ..., S_n = V_n)]} \quad (۲۸)$$

در عبارت فوق C_x مجموعه دسته هایی می باشد که در آنها فعالیت x انتخاب شده است. صورت کسر بالا مجموع وزن دار مقادیر Q برای عناصر C_x است (از احتمال قرار گرفتن وضعیت S در دسته C به عنوان وزن این جمع استفاده شده است). مخرج کسر نیز برای نرمال کردن عبارت می باشد.

۱ـ۷ـ جمع بندی مدل Q

یادگیری Q با دسته بندی را می توان به صورت زیر خلاصه کرد:

۱ ـ مقادیر ثابتی برای پارامترهای Q$(\theta, \gamma, \lambda)$ و پارامترهای دسته بندی $(\rho, K, \delta, \varepsilon)$ در نظر بگیرید.

۲ ـ برای همیشه:

I. وضعیت فعلی محیط را مشاهده کنید (X).

II. در θ درصد از مواقع فعالیتی را به طور تصادفی انتخاب کنید. در مواقع دیگر فعالیتی را انتخاب کنید که مقدار Q (X, a) را ماکزیمم کند.

III. فعالیت a را در محیط اعمال کنید. فرض کنید وضعیت جدید Y باشد و پسخور بلافاصله اعمال این فعالیت r باشد.

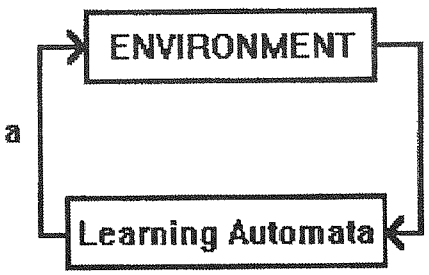
IV. میزان Q (X, a) را با معادله (۱) بهنگام کنید.

V. اگر دسته ای مانند C وجود داشت که به همراه X در نامعادلات (۸) و (۹) صدق کند، وضعیت X را در دسته C ادغام کنید. در غیر این صورت دسته جدیدی از روی X ایجاد نمایید.

VI. هر دو دسته C_1 و C_2 که در نامعادلات (۱۹) و (۲۰) صدق می کنند را در هم ادغام کنید.

۲ ـ اتوماتون های یادگیر

در اتوماتون های یادگیر یک تصمیم گیرنده در یک محیط تصادفی فعالیت هایی را براساس پسخورهای دریافت شده از محیط انتخاب می کند (شکل ۲). تصمیم گیرنده در چنین محیطی از پسخورهای دریافت شده یک استراتژی انتخاب فعالیت، ایجاد می کند و این استراتژی را با دریافت پسخورهای جدید بهنگام می سازد. یک اتوماتون یادگیر را می توان با پنج مؤلفه نمایش داد: ورودی یا پسخور (r)، خروجی ها (α)، وضعیت های درونی (ϕ) و دو تابع یکی برای نگاشت پسخور به وضعیت درونی فعلی و دیگری برای نگاشت وضعیت درونی فعلی به خروجی، هرگاه ساختار نگاشت پسخور به وضعیت فعلی و نگاشت وضعیت فعلی به خروجی در طول زمان (و با دریافت پسخورهای جدید) تغییر نکند، اتوماتون یادگیر از نوع ساختار ثابت خواهد بود و در غیر این صورت اتوماتون از نوع ساختار متغیر می باشد.



شکل (۲) رابطه اتوماتون یادگیری با محیط

۲ـ۱ـ اتوماتون های با ساختار متغیر (اتوماتون های LRI, LRεP, LRP)

در اتوماتون های یادگیر با ساختار متغیر به هر فعالیت a یک احتمال انتخاب P_a نسبت داده می شود. هرگاه اتوماتون فعالیت a را در زمان t انتخاب کند و پسخور محیط موفقیت آمیز باشد، آنگاه p_a افزایش داده می شود و احتمال انتخاب سایر فعالیت ها کاهش پیدا

می‌کند (در صورتی که پسخور محیط نشانگر عدم موفقیت باشد عکس این عمل اتفاق خواهد افتاد). در حالت کلی اتوماتون یک فعالیت (به عنوان مثال فعالیت a) را براساس احتمال انتخاب فعالیت ها انتخاب می‌کند. اگر پسخور حاصل از اعمال این فعالیت r باشد، اتوماتون مقادیر p_i ها را مطابق فرمول های زیر بهنگام می‌سازد:

$$p_i(t+1) \leftarrow p_i(t) + (1 - r(t)) \times \beta \times \left[\frac{1}{m-1} - p_i(t)\right] - r(t) \times \alpha \times p_i(t), \text{ for all } i \neq \alpha$$

$$p_a(t+1) \leftarrow p_a(t) - (1 - r(t)) \times \beta \times p_a(t) + r(t) \times \alpha \times (1 - p_a(t))$$

در معادلات بالا m تعداد فعالیت ها و α و β به ترتیب پارامترهای پاداش و جزا می باشند که مقادیری بین صفر و یک را اختیار می‌کنند. پارامتر پاداش نرخ افزایش احتمال انتخاب فعالیتی را نشان می‌دهد که پسخور موفقیت‌آمیز دریافت کرده و پارامتر جزا نرخ کاهش احتمال انتخاب فعالیتی را مشخص می‌کند که پسخور غیر موفق دریافت نموده است. در رابطه بالا مقدار r باید بین ۰ و ۱ باشد، در غیر این صورت می‌توان از رابطه زیر مقدار r را به عددی بین ۰ و ۱ نگاشت کرد (در این رابطه فرض شده است که مقدار بیشینه و کمینه پسخورهای دریافت شده از پیش مشخص نمی باشند):

$$r_{norm} = \frac{r(t) - \text{Min}(r(i))}{\text{Max}(r(i)) - \text{Min}(r(i))}$$

اتوماتون های یادگیر برحسب مقدار پارامتر جزا به سه اتوماتون اصلی تقسیم بندی می‌شوند. اگر $\alpha \neq 0$ و $\beta = 0$ اتوماتون را LRI[3] و در صورتی که $a = \beta \neq 0$ باشد، اتوماتون را LRP[5] و اگر $\alpha \neq 0$ و β به اندازه کافی کوچک باشد، اتوماتون را LRεP[6] می نامند.

فرض کنید احتمال انتخاب فعالیت ها در زمان t را با بردار P(t) نمایش دهیم: $P(t) = \{p_{a0}, p_{a1}, ..., p_{am}\}$، آنگاه P (n) یک حالت جذب شونده[7] خوانده می‌شود. اگر برای هر $k \geq n$ داشته باشیم: $p(k) = p(n)$. اتوماتونی که دارای حالت جذب شونده باشد را اتوماتون جذب شونده می‌خوانند. نشان داده شده است که اتوماتون LRI که در یک محیط ایستا عمل می‌کند، جذب شونده می‌باشد. اتوماتون های LRP و LRεP دارای خصوصیت جذب شوندگی نمی باشند [23] [12].

۲ـ۲ـ اتوماتون های با ساختار ثابت

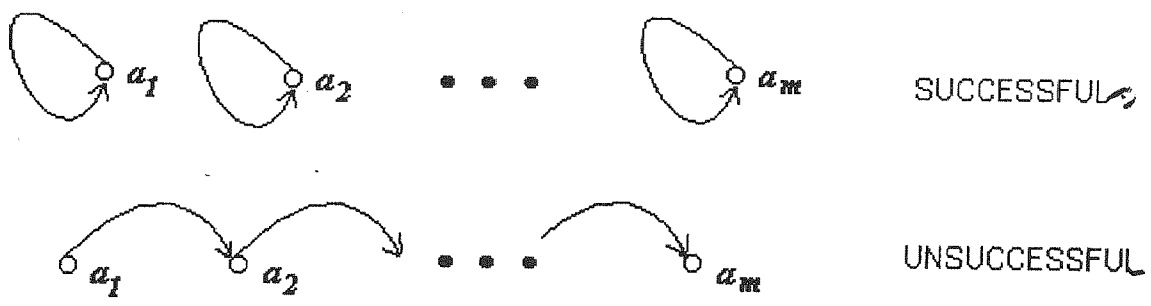
۲ـ۲ـ۱ـ اتوماتون های بدون حافظه (اتوماتون FA)

یک استراتژی ساده برای انتخاب فعالیت ها این است که اتوماتون تا زمانی که پسخور موفقیت‌آمیز دریافت می‌کند، به انتخاب فعالیتی که در لحظه قبل انتخاب کرده ادامه دهد. اما هرگاه پسخور ناموفق بود، یک فعالیت دیگر را انتخاب کند. در گراف های شکل ۳ این استراتژی انتخاب فعالیت مشخص شده است (در گرافها m تعداد فعالیت هاست). پیاده سازی این استراتژی مستلزم روشی برای برآورد موفقیت‌آمیز بودن یا نبودن یک پسخور است. فرض کنید پسخور دریافت شده توسط مأمور r باشد و مقداری بین c و d داشته باشد. در این صورت یک راه ساده برای تخمین موفقیت‌آمیز بودن پسخور مقایسه آن با (c+d)/2 است. از آنجایی که اتوماتون در ابتدای یادگیری اطلاعی از مقادیر c و d ندارد، می‌توان از روابط زیر برای تخمین آنها استفاده کرد:

$$c = \underset{i=0}{\text{MIN}}(r(i)), \quad d = \underset{i=0}{\text{MAX}}(r(i)) \qquad (۲۹)$$

$$\text{if } r(t) > \frac{c+d}{2} \text{ then SUCCESSFUL} \qquad (۳۰)$$

else UNSUCCESSFUL



شکل (۳) استراتژی انتخاب فعالیت اتوماتون FA

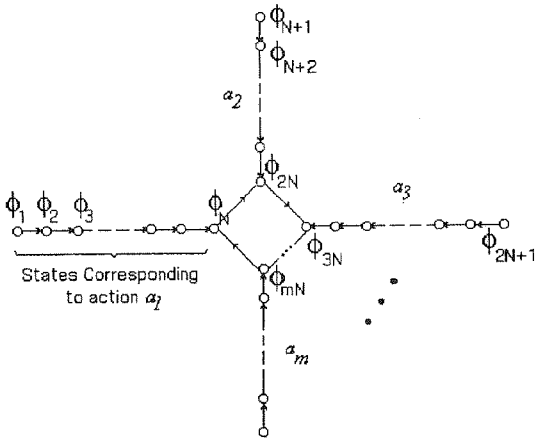
۲ـ۲ـ۲ـ اتوماتون‌های حافظه دار

۲ـ۲ـ۲ـ۱ـ اتوماتون اول تسلتسین (اتوماتون L) [18]

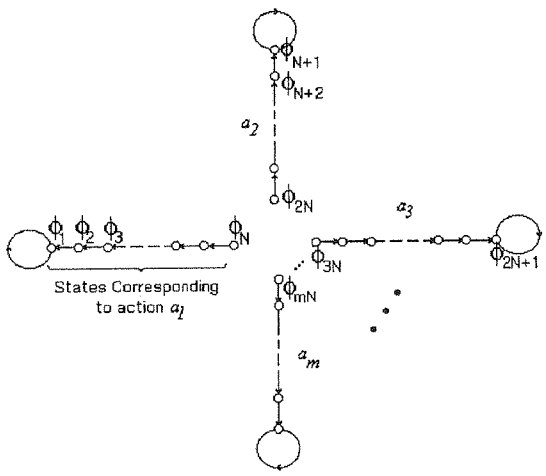
این اتوماتون تعداد پسخورهای موفق و ناموفق حاصل از هر فعالیت را ثبت می‌کند. در اینجا پس از انتخاب یک فعالیت تا زمانی که تعداد پسخورهای موفقیت‌آمیز بیشتر از تعداد پسخورهای ناموفق باشد، فعالیت انتخاب شده در لحظه قبل دوباره انتخاب خواهد شد. در شکل‌های ۴ و ۵ این استراتژی انتخاب فعالیت مشخص شده است (موفقیت‌آمیز بودن یا نبودن یک پسخور با استفاده از روابط ۲۹ و ۳۰ محاسبه می‌شود). در شکل N عمق اتوماتون می‌باشد و می‌توان آن را حافظه متعلق به هر فعالیت دانست (این اتوماتون در حالت N = 1 به اتوماتون بدون حافظه تبدیل می‌شود). همچنین ϕ وضعیت درونی اتوماتون را نمایش می‌دهد، به طوری که وضعیت‌های درونی ϕ_{iN+1}, ϕ_{iN+2}, ..., ϕ_{2Ni} مربوط به فعالیت a_{i+1} $(0 \le i \le m\text{-}1)$ می‌باشند و m تعداد فعالیت‌های قابل انتخاب را نشان می‌دهد. این اتوماتون با حرف L مشخص می‌شود.

۲ـ۲ـ۲ـ۲ـ اتوماتون دوم تسلتسین (اتوماتون G)[18]

نسخه‌های متعددی از اتوماتون حافظه دار ساخته شده است. یک نمونه از این اتوماتون‌ها که آن را اتوماتون G می‌نامیم پس از انتخاب یک فعالیت جدید حداقل N بار دیگر آن را انتخاب خواهد کرد. تفاوت این اتوماتون با اتوماتون L تنها پس از دریافت پسخور ناموفق می‌باشد (شکل ۶).



شکل (۴) استراتژی انتخاب فعالیت اتوماتون L به ازای دریافت پسخور ناموفق



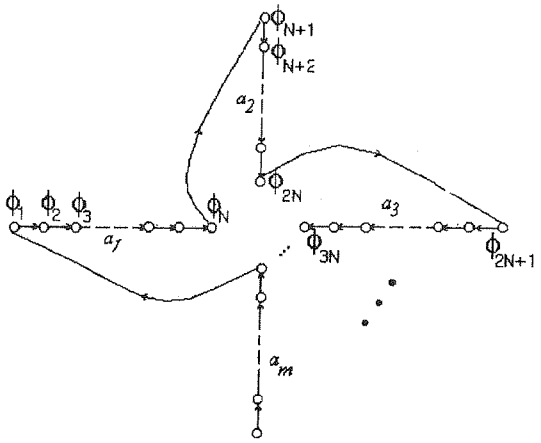
شکل (۵) استراتژی انتخاب فعالیت اتوماتون L به ازای دریافت پسخور موفقیت‌آمیز

۲ـ۲ـ۲ـ۳ـ اتوماتون کرینسکی (اتوماتون K1) [8]

این اتوماتون زمانی که پسخور محیط موفقیت‌آمیز نیست، مانند اتوماتون L رفتار می‌کند. اما هر گاه پسخور محیط موفقیت‌آمیز باشد، وضعیت درونی مأمور از ϕ_{iN+j} (for i =0, 1, 2, ..., m-1 and j = 1, 2, ..., N) به $\phi_{(j-1)N+1}$ منتقل می‌شود. یعنی این مدل تنها با دریافت N بار پسخور ناموفق فعالیتی که انتخاب کرده است را تغییر می‌دهد. این اتوماتون با K1 مشخص می‌شود.

۲ـ۲ـ۲ـ۴ـ اتوماتون کریلوف (اتوماتون K2) [Krylov64]

این اتوماتون زمانی که پسخور محیط موفقیت‌آمیز باشد مانند اتوماتون L رفتار می‌کند. اما هرگاه پسخور محیط موفقیت‌آمیز نباشد، وضعیت درونی مأمور از ϕ_{iN+j} $(j \neq 1, N)$ به احتمال 50% به ϕ_{iN+j+1} و به احتمال 50% به ϕ_{iN+j-1} منتقل می‌شود. هرگاه j = 0 باشد، وضعیت درونی به احتمال 50% تغییر نمی‌کند و به احتمال 50% به وضعیت درونی ϕ_{iN+j+1} منتقل می‌شود. هرگاه j = N باشد، وضعیت درونی به احتمال 50% به یکی از وضعیت‌های درونی ϕ_{iN+j-1} یا $\phi_{(i+1)N+j}$ منتقل می‌شود. این اتوماتون K2 نامیده شده است.



شکل (۶) استراتژی انتخاب فعالیت اتوماتون G به ازای دریافت پسخور ناموفق

۳ـ تنظیم پارامترهای مدل Q توسط اتوماتون‌های یادگیر [4] [3]

برای تنظیم پارامترهای مدل Q با استفاده از اتوماتون های یادگیر این اتوماتون ها باید مجهز به فعالیت هایی جهت تغییر مقادیر پارامترهای مدل Q باشند (به این ترتیب مدل Q نقش محیط اتوماتون یادگیر را ایفا می‌کند ،شکل ۷). شمای کلی روش تنظیم پارامترها توسط اتوماتون ها به این صورت است:

۱)اتوماتون ها با اعمال فعالیت های خود مقدار پارامترهای مدل Q را تغییر می دهند.

۲) مدل Q در طول یک پریود زمانی (period) از این مقادیر استفاده می کند.

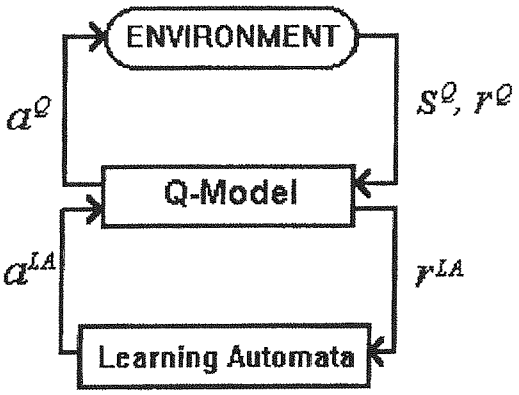
۳) در انتهای پریود میزان کارا بودن هر یک از پارامترها براساس عملکرد مدل ارزیابی می‌شود و به صورت پسخورهایی در اختیار اتوماتون های یادگیر قرار می گیرند.

۴) اتوماتون ها با استفاده از پسخورهای دریافتی احتمال انتخاب فعالیت های خود را بهنگام می کنند.

۵) تکرار عملیات.

برای تنظیم هر پارامتر به روش بالا باید مجموعه ای از مقادیر را برای آن درنظر گرفت، تا اتوماتون با اعمال فعالیت خود یکی از اعضای این مجموعه را برای پارامتر انتخاب کند. تمام مقادیری که ممکن است در این محیط برای پارامتر مناسب باشند باید در این مجموعه قرار بگیرند. اما این مقادیر از پیش مشخص نیستند. بنابراین باید برای تعیین اعضای مجموعه ها از روش های هیورستیک استفاده نمود. در اینجا برای تعیین این مجموعه ها تنها سه نکته مدنظر بوده است: اولاً سعی شده است تا هر کدام از اعضای مجموعه ها نمایانگر یک بازه از مقادیر قابل قبول برای پارامتر باشد، ثانیاً این بازه ها بر روی هم تمام دامنه مقادیر معقول برای پارامتر در محیط را پوشش دهند و ثالثاً تعداد اعضای مجموعه ها زیاد نباشد.

به عنوان مثال فرض کنید از یک اتوماتون یادگیر برای تنظیم مقدار θ استفاده شود. برای تعیین مقدار مناسب این پارامتر که عددی بین صفر و یک است، بازه [0, 1] به بازه های کوچکتری تقسیم می‌شود. فرض کنید بازه ها با فاصله یک دهم انتخاب شوند (یعنی ده بازه) و هر بازه با کوچکترین عدد آن بازه مشخص شود. وظیفه اتوماتون یادگیر انتخاب بازه ای است که عدد مشخص کننده آن مناسبترین مقدار را برای پارامتر انتخاب تصادفی یادگیری Q داشته باشد. برای این منظور باید از یک اتوماتون یادگیر که مجهز به ده فعالیت (برای انتخاب هر کدام از بازه ها) است استفاده کرد. اتوماتون یادگیر پس از انتخاب یک مقدار برای این پارامتر آن را در اختیار مدل Q می‌گذارد و مدل Q در یک پریود زمانی (مثلاً ده واحد زمانی) از این مقدار استفاده خواهد کرد. پس از انقضای این پریود میزان کارآیی پارامتر تنظیم شده توسط یک تابع ارزیابی به اتوماتون یادگیر پسخور می‌گردد و دوباره این عملیات تکرار می‌شود.



شکل (۷) رابطه مدل Q با اتوماتون یادگیر و محیط

تابع ارزیابی مورد استفاده برای اتوماتون تنظیم کننده پارامتر θ به صورت زیر تعریف شده است: (این روابط پسخور اتوماتون در زمان t را مشخص می‌کنند، فرض شده که اتوماتون در زمان T پسخور بیشینه را دریافت کرده باشد.)

$$\text{IF} \sum_{i=t-p}^{t} r_i^Q \geq \sum_{i=t-p}^{T} r_i^Q \text{ THEN } r^{IA} = MAX_{i=1}^{t}(r_i^Q)$$

$$\text{ELSE IF} \sum_{i=t-p}^{t} \frac{r_i^Q}{P} \leq \sum_{i=1}^{t} \frac{r_i^Q}{t} \text{ THEN } r^{IA} = MIN_{i=1}^{t}(r_i^Q)$$

$$\text{ELSE } r^{IA} = MIN_{i=1}^{t}(r_i^Q) + \frac{\sum_{i=t-p}^{t} \frac{r_i^Q}{P} - \sum_{i=1}^{t} \frac{r_i^Q}{t}}{\sum_{i=T-p}^{T} \frac{r_i^Q}{P} - \sum_{i=1}^{t} \frac{r_i^Q}{t}} \times (MAX_{i=1}^{t}(r_i^Q) - MIN_{i=1}^{t}(r_i^Q))$$

سایر پارامترهای مدل Q علاوه بر عملکرد مدل بر تعداد دسته های مدل نیز مؤثرند. در تابع ارزیابی اتوماتون های تنظیم کننده این پارامترها در صورتی که تعداد دسته ها در یک پریود تغییر نکنند، پسخور با روابط بالا محاسبه شده است. اما اگر تعداد دسته ها در یک پریود زمانی افزایش یابد و میانگین پسخورهای دریافتی در این پریود نسبت به میانگین کل بهتر نشود، آنگاه پسخور کمینه و در صورتی که تعداد دسته ها در یک پریود زمانی کاهش یابد و میانگین پسخورهای دریافتی در این پریود نسبت به پریود قبلی کاهش نیابد، پسخور بیشینه برای اتوماتون ها در نظر گرفته شده است.

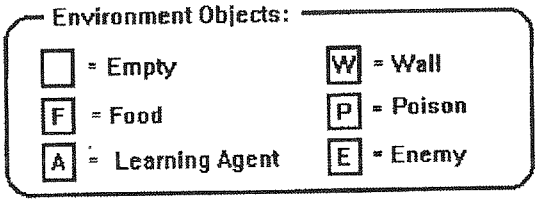
٤ ـ آزمایش ها و نتایج

در آزمایش های انجام شده از اتوماتون های K1, G, L, FA, LRI, LRεP, LRP و K2 برای تنظیم پارامترهای مدل Q استفاده شده است. نام مدل هایی که پارامترهای آنها با این اتوماتون ها تنظیم شده اند، به ترتیب با سمبل های Q-K1, Q-G, Q-L, Q-LRI, Q-LRεP, Q-LRP و Q-K2 نمایش داده شده اند. در این مقاله طی آزمایش هایی پارامترهای θ (تتا)، ρ (رو)، δ (دلتا) و ε (اپسیلون) تنظیم شده اند. برای نمایش پارامتر یا پارامترهایی که در هر آزمایش تنظیم می شوند، از حرف اول هر کدام از پارامترها (به ترتیب D, R, T و E) استفاده شده است. به این ترتیب که حرف اول پارامتر یا پارامترهایی که توسط اتوماتون تنظیم شده است، داخل پرانتز و در مقابل نام مدل قرار گرفته شده است. برای مثال منظور از مدل Q-L(ERT) در این متن این است که پارامترهای ε ,ρ و θ مدل Q توسط اتوماتون L تنظیم شده است.

در آزمایش هایی که شرح آنها خواهد آمد، محیط یک صفحه شطرنجی شکل است که در هر خانه آن ممکن است یک شیء وجود داشته باشد. در شرح آزمایش ها ابتدا از یک محیط با قوانین نسبتاً ساده استفاده خواهد شد. برای این محیط ابتدا مدل هایی که تنها پارامتر θ در آنها تنظیم شده است، بررسی می شوند. برای این منظور عملکرد، انعطاف پذیری، قدرت اکتشاف و قدرت یادگیری توالی عملیات مدل ها مقایسه خواهد شد. سپس در این محیط ساده تنظیم همزمان بیش از یک پارامتر بررسی می شود و تأثیر آن در عملکرد و انعطاف پذیری مدل ها مطالعه می شود. سپس از یک محیط کلاسیک برای بررسی تنظیم پارامترهای مدل Q استفاده خواهد شد. این محیط بر اساس مثال کلاسیک هل دادن جعبه ها طراحی شده است.

٥ ـ یک محیط ساده

مأمور یادگیرنده در این محیط مجهز به چهار حس برای مشاهده اجسام خانه های مجاور خود (بالا، راست، پایین و چپ) و چهار فعالیت برای حرکت به این خانه های مجاور است. این فعالیت ها تنها زمانی موجب حرکت به یک خانه مجاور می شوند که در آن خانه شیئی وجود نداشته باشد. در این صورت حرکت به این خانه انجام می گردد و مأمور پسخور صفر (r=0) را از محیط دریافت می کند. در هر خانه از محیط یکی از اشیاء دیوار، غذا، زهر یا دشمن می تواند وجود داشته باشد. اعمال یک فعالیت برای حرکت به خانه ای که در آن دیوار، غذا، زهر یا دشمن وجود دارد به ترتیب پسخورهای 40- ,100 + ,100- و 100- را ایجاد می کند. اشیاء دیوار، غذا و زهر نمی توانند در محیط حرکت کنند، اما دشمن می تواند در هر واحد زمانی به یکی از چهار خانه مجاور (بالا، راست، پایین یا چپ) حرکت کند (حرکت دشمن به خانه های مجاور به طور تصادفی انجام می گیرد). برای ساده کردن توصیف محیط در آزمایش های این بخش از علامت های زیر برای نمایش هر کدام از اشیاء استفاده شده است:



شکل (۸) اشیاء درون محیط

۵ـ۱ـ تنظیم پارامتر انتخاب تصادفی (θ)

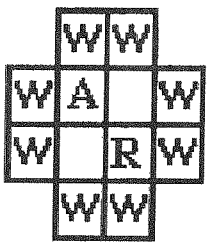
در این آزمایش‌ها برای تنظیم پارامتر θ از یک اتوماتون یادگیر مجهز به ده فعالیت برای انتخاب یکی از مقادیر {0, 0.1, 0.2, ..., 0.9} برای θ استفاده شده است. در آزمایش‌های انجام شده مقدار θ برای مدل Q با پارامترهای ثابت 0.1 در نظر گرفته شده است (این مقداری است که معمولاً برای این پارامتر توصیه می‌شود [10]. مقادیر سایر پارامترهای یادگیری نیز براساس مقادیر متعارف آنها انتخاب شده‌اند (در توضیح هر یک از آزمایش‌ها مقدار پارامترهای مورد استفاده ذکر گردیده است).

۵ـ۱ـ۱ـ مقایسه عملکرد

برای بررسی تأثیر تنظیم θ در عملکرد مدلها از محیط ساده با پیکربندی شکل ۹ استفاده شده است. در این پیکربندی یک مأمور یادگیر (A) به همراه یک مأمور تصادفی (R) توسط اشیاء دیوار (W) محاصره شده‌اند. در این آزمایش‌ها مقادیر پارامترهای ثابت مدل‌های Q با پارامتر θ ثابت و «تنظیم شده» بدین صورت انتخاب شده‌اند:

$\theta = 0.1, k = 5, \delta = 10, \varepsilon = 0.000001, \rho = 2, \lambda = 0.5, \gamma = 0.9$

(فقط برای مدل‌های Q با پارامترهای ثابت) و period = 1 (فقط برای مدل‌های Q با پارامتر تنظیم شده). در آزمایش‌های این بخش هر مدل سه بار در محیط شبیه‌سازی شده ومتوسط نتایج سه بار شبیه‌سازی در نمودارها و جداول امده است.

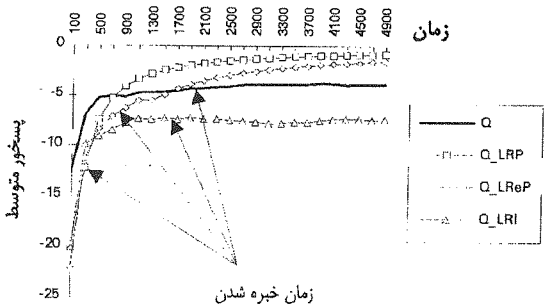


شکل (۹) پیکربندی محیط ساده در آزمایش مقایسه عملکرد

۵ـ۱ـ۱ـ تنظیم پارامتر انتخاب تصادفی با اتوماتون‌های ساختار متغیر

در این آزمایش از مأمورهای Q-LRP (T), Q-LRI (T), Q - LRεP (T) و به عنوان مأمور یادگیر محیط شکل ۲۲ استفاده شده است. برای تمام آزمایش‌ها مقدار α و β برای اتوماتون‌ها برابر با 0.1 در نظر گرفته شده است. در نمودار ۲ تغییرات عملکرد مدل‌ها مقایسه شده است، همچنین در این نمودار زمان خبرگی در محیط برای هر یک از مدل‌ها با پیکان‌هایی مشخص شده‌اند. مدل زمانی در محیط خبره می‌شود که تمام دسته‌های لازم برای انتخاب بهینه فعالیت‌ها را کشف کرده باشد. می‌توان زمانی را که از آن پس دسته‌های کشف شده، ثابت باقی می‌مانند را معادل زمان خبرگی دانست (مطابق نمودار مدل‌های Q-LRI و Q-LRεP, Q-LPR, Q به ترتیب پس از ۱۹۰۰، ۳۰۰، ۸۰۰ و ۱۶۰۰ واحد زمان بر خبره شده‌اند).

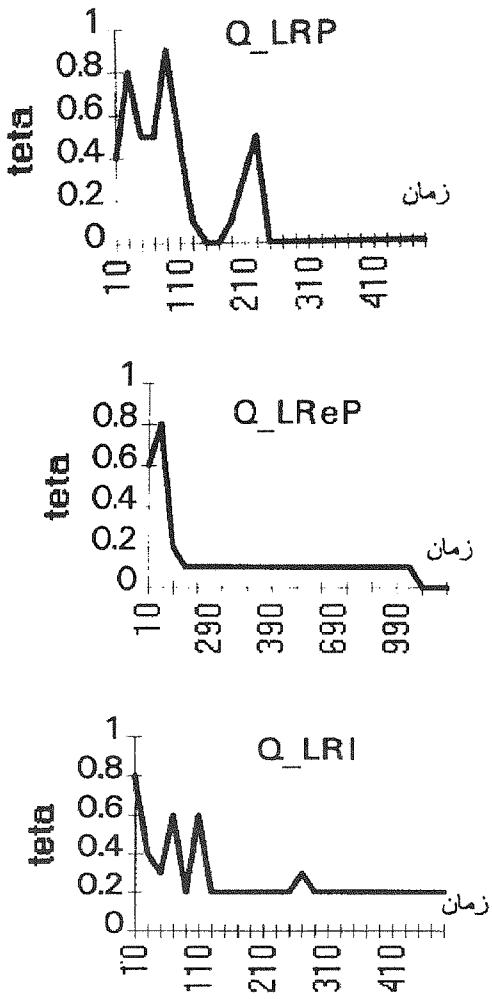
در مقایسه انجام یافته Q-LRP و Q-LRεP عملکرد نسبتاً خوبی داشته‌اند. این امر به این دلیل است که اولاً، این مدل‌ها افزایش انتخاب‌های تصادفی در واحدهای زمانی اولیه باعث کاهش زمان لازم برای خبرگی در محیط می‌شوند و ثانیاً Q-LRεP و Q-LRI پس از خبرگی در محیط دیگر انتخاب تصادفی انجام نمی‌دهند (با ثابت ماندن مقدار θ در مقدار صفر). در نمودارهای ۳ ـ الف، ۳ ـ ب و ۳ ـ پ نحوه تغییرات پارامتر θ با زمان در مدل‌های Q-LRεP, Q-LRP و Q-LRI نشان داده شده است. همانطور که در نمودار ۳ ـ پ مشخص است، مقدار θ در Q-LRI به سرعت به مقدار 0.2 همگرا شده و در همین مقدار ثابت باقی مانده است و در نتیجه عملکرد این مدل تقریباً معادل عملکرد یک مدل Q با پارامترثابت θ = 0.2 می‌باشد. در آزمایش انجام شده پارامتر θدر مدل Q همواره ثابت و برابر با 0.1 در نظر گرفته شده است، این مدل به دلیل اینکه حتی پس از خبرگی کامل انتخاب‌های تصادفی انجام می‌دهد، نسبت به Q-LRP و Q-LRεP عملکرد ضعیف‌تری را نشان داده است.



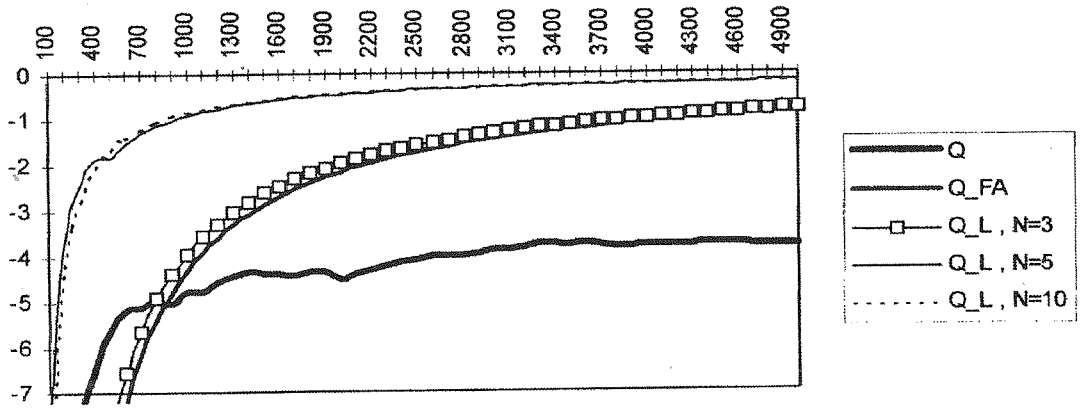
نمودار (۲) مقایسه عملکرد و زمان خبرگی در مدل‌های Q-LRI(T) و Q-LRεP(T), Q-LRP(T), Q

۵-۱-۱-۲-تنظیم پارامتر انتخاب تصادفی توسط اتوماتون‌های با ساختار ثابت

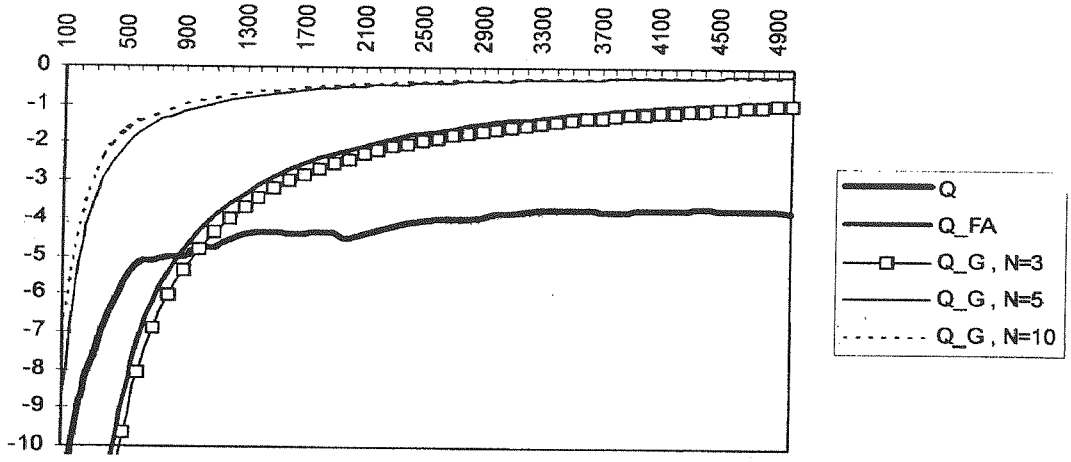
در این آزمایش‌ها مقادیر پارامترهای ثابت مدل‌های Q با پارامتر θ «ثابت» و «تنظیم شده» بدین صورت انتخاب شده‌اند: $\theta = 0.1, k = 5, \delta = 10, \varepsilon = 0.000001$ $\rho = 2, \lambda = 0.5, \gamma = 0.9$ (فقط برای مدل‌های Q با پارامترهای ثابت) و period = 1 (فقط برای مدل‌های Q با پارامتر تنظیم شده). در نمودارهای ۴ تا ۷ به ترتیب عملکرد هر کدام از مدل‌های Q-L, Q-G, Q-K1 و Q-K2 با عمق‌های حافظه مختلف با مدل‌های Q و Q-FA مقایسه شده‌اند. مطابق نمودارها مشخص است که اولاً تنظیم پارامترها باعث افزایش عملکرد شده است (دلایل افزایش عملکرد با تنظیم پارامترها در بخش قبل بررسی شده است) و ثانیاً افزایش عمق حافظه در مدل‌ها موجب عملکرد بهتر آنها شده است. در نمودار ۸ نحوه تغییر عملکرد با افزایش عمق حافظه آورده شده است. چنانچه ملاحظه می‌شود برای این محیط عمق حافظه بین ۵ تا ۱۰ بهترین نتیجه را در بر داشته است. نمودار ۹ عملکرد مدل‌های Q-L, Q-G, Q-K1 و Q-K2 هر یک با عمق حافظه ۵ را با یکدیگر و با مدل‌های Q-FA, Q و مدل Q-LRP مقایسه می‌کند. مدل‌های Q-L, Q-G, Q-K1 و Q-K2 عملکرد بهتری نسبت به سایر مدل‌ها نشان داده‌اند و مطابق نمودار ۹ عملکرد این مدل‌ها نسبتاً مشابه بوده است.



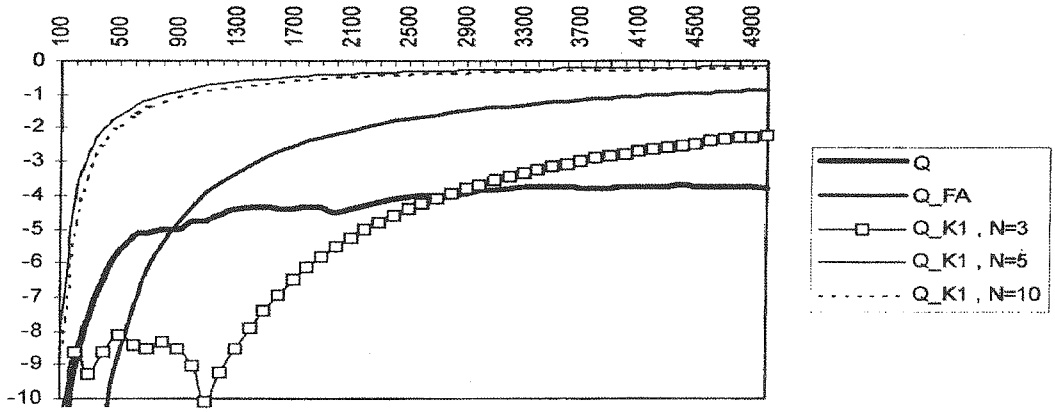
نمودار (۳) الف. (بالا سمت راست) تغییرات θ و Q-LRP
ب- (بالا سمت چپ) تغییرات θ در Q-LRεP
پ- (پایین) تغییرات θ در Q-LR



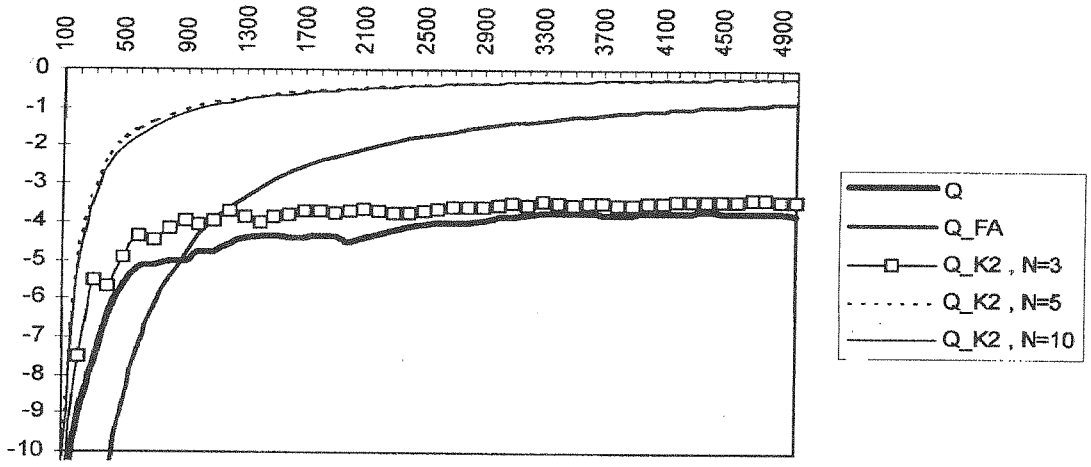
نمودار (۴) عملکرد مدل Q-L با عمق‌های حافظه مختلف



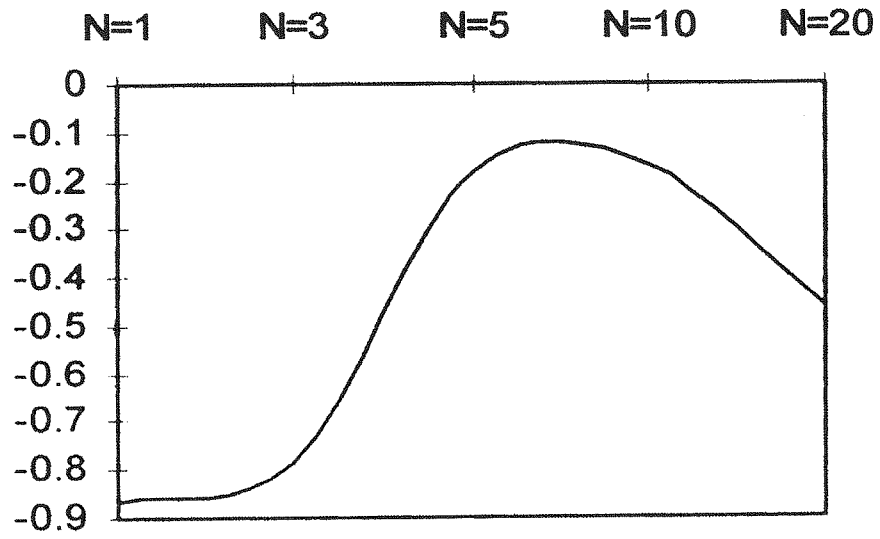
نمودار (۵) عملکرد مدل Q-G با عمق های حافظه مختلف



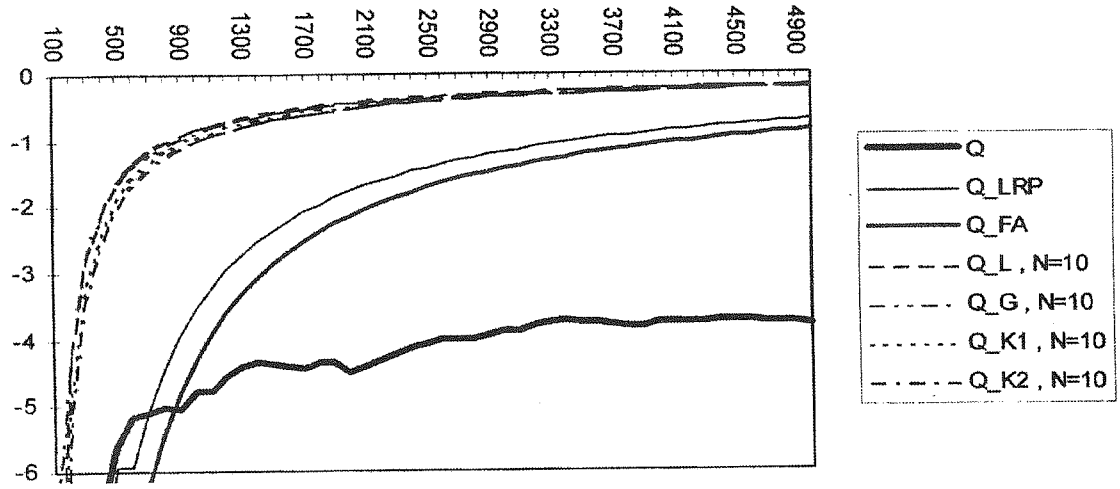
نمودار (۶) عملکرد مدل Q-K1 با عمق های حافظه مختلف



نمودار (۷) عملکرد مدل Q-K2 با عمق های حافظه مختلف



نمودار (۸) تغییرات عملکرد مدل Q-L با تغییر عمق حافظه



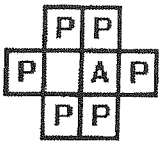
نمودار (۹) مقایسه عملکرد مدل های مختلف با یکدیگر

۵ـ۱ـ۲ـ مقایسه انعطاف پذیری

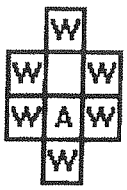
منظور از انعطاف پذیری مدل های یادگیری قابلیت انطباق آنها با محیط است. درجه انعطاف پذیری مدل یادگیری را می توان متناسب با زمان لازم برای یافتن بهترین فعالیت برای اعمال در یک وضعیت تجربه نشده دانست. برای اندازه گیری درجه انعطاف پذیری مدل در برخورد با وضعیت های جدید از پیکربندی های شکل ۱۰ استفاده شده است. در این آزمایش ابتدا از محیط شکل ۱۰ ـ الف برای خبره کردن یادگیرنده استفاده شده است. برای این منظور یادگیرنده به مدت ۲۰۰۰ واحد زمان در این محیط قرار گرفته است. سپس یادگیرنده در محیط شکل ۱۰ ـ ب قرار داده شده و در این وضعیت جدید زمان لازم برای انتخاب بهترین فعالیت (حرکت به بالا) اندازه گیری شده است. برای آزمایش انجام گرفته مقادیر پارامترهای ثابت مدل های Q با «پارامتر ثابت» و «تنظیم شده» بدین صورت انتخاب شده اند:

$\theta = 0.1, k = 5, \delta = 10, \varepsilon = 0.000001, \rho = 2, \lambda = 0.5, \gamma = 0.9$

(فقط برای مدل Q با پارامتر ثابت) و period = 1 (برای مدل های Q با پارامتر تنظیم شده). در آزمایش های این بخش زمان متوسط یادگیری وضعیت جدید با معدل گیری از پنج بار اجرای هر آزمایش محاسبه شده است.



محیط الف



محیط ب

شکل (۱۰) پیکربندی های محیط ساده در آزمایش انعطاف پذیری

۵ - ۱ - ۲ - ۱ - تنظیم پارامتر انتخاب تصادفی توسط اتوماتون های با ساختار متغیر

در این آزمایش از اتوماتون های با ساختار متغیر برای تنظیم پارامترهای مدل Q در محیط شکل ۱۰ استفاده شده است. برای آزمایش انجام شده مقدار α برای اتوماتون ثابت ($\alpha = 0.1$) در نظر گرفته شده است. در اینجا مأمور یادگیرنده در محیط ۱۰ - الف یاد می گیرد که انتخاب فعالیت های «حرکت به بالا» و «حرکت به پایین» در هر شرایط پسخور ناموفق دریافت خواهد کرد. اما هرگاه خانه سمت راست (یا چپ) مأمور خالی باشد، انتخاب فعالیت «حرکت به راست» (یا چپ) پسخور موفق دریافت خواهد نمود. اما وقتی مأمور در محیط ۱۰ - ب قرار می گیرد، باید یاد بگیرد که در وضعیت جدید خود فعالیت «حرکت به بالا» را انتخاب کند. بنابر این آنچه که مأمور در محیط ۱۰ - الف یاد گرفته است (دسته های کشف شده در محیط ۱۰ - الف) برای یادگیری وضعیت جدید در محیط ۱۰ - ب کافی نمی باشد و در نتیجه یادگیری بهترین فعالیت برای اعمال در محیط ۱۰ - ب فقط با انتخاب تصادفی فعالیت ها امکان پذیر خواهد بود. مطابق آزمایش پس از خبرگی یادگیرنده ها در محیط ۱۰ - الف تمام اتوماتون های یادگیر تنظیم کننده مدل های Q-LA وضعیت درونی یکسانی خواهند داشت. در تمام این اتوماتون ها «احتمال انتخاب» فعالیت قرار دادن صفر برای $\theta(0 \leftarrow \theta)$ برابر با 100% و «احتمال انتخاب» سایر فعالیت ها 0% شده است. بنابراین برای انتخاب تصادفی در محیط ۱۰ - ب لازم بوده تا اتوماتون ها از این وضعیت درونی خارج شوند (شکل ۱۱ وضعیت درونی اولیه اتوماتون ها را در محیط ۱۰ - ب نشان می دهد). وقتی مأمور در وضعیت مشخص شده در محیط ۲۳ - ب قرار می گیرد، انتخاب فعالیت ها براساس آموخته های قبلی (یعنی انتخاب فعالیت های «حرکت به راست» و «حرکت به چپ») موجب دریافت پسخور ناموفق می شود. این پسخور ناموفق به اتوماتون های تنظیم کننده θ منتقل می شود و سپس مدل با استفاده از این پسخور «احتمال انتخاب» فعالیت های خود را اصلاح می کند.

$$P(\theta \leftarrow 0.0) = \%100$$
$$P(\theta \leftarrow 0.1) = \%0$$
$$\cdots$$
$$P(\theta \leftarrow 0.9) = \%0$$

شکل (۱۱) وضعیت درونی اولیه اتوماتون های LA از مدل های Q-LA در محیط شکل ۱۰ - «ب»

نتایج آزمایش برای مدل های Q با مقادیر متفاوت پارامتر جزا (β) در اتوماتون های تنظیم کننده آنها در جدول ۱ خلاصه شده است. در این آزمایش همانگونه که انتظار می رفت با کاهش مقدار پارامتر β درجه انعطاف پذیری یادگیرنده کاهش پیدا کرده است، چرا که کاهش این پارامتر موجب کاهش میزان اصلاح مقدار «احتمال انتخاب» فعالیت ها پس از دریافت پسخور ناموفق می شود. به طوری که در اتوماتون LRI هیچگونه اصلاحی در «احتمال انتخاب» فعالیت ها رخ نمی دهد. در نتیجه مقدار θ در مدل Q-LRI همواره صفر باقی می ماند و این مدل هیچگاه فعالیت «حرکت به بالا» را در محیط جدید (۱۰ - ب) امتحان نخواهد کرد. لازم به ذکر است که انجام این آزمایش با مدل Q (با پارامتر ثابت) نشان داد که این مدل برای یادگیری وضعیت جدید به طور متوسط به 32.1 واحد زمانی نیاز دارد.

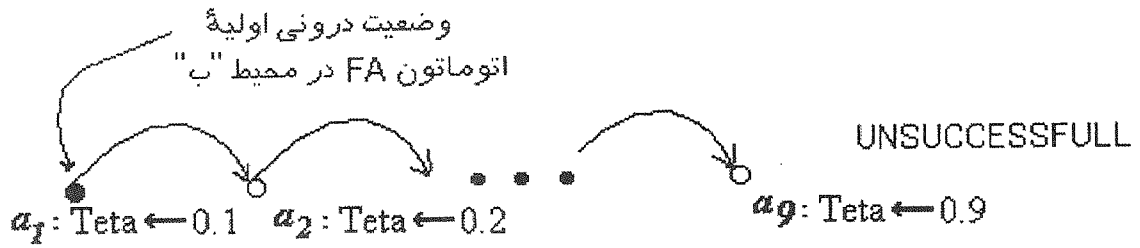
واضح است که افزایش N در اتوماتون های ساختار ثابت باعث افزایش زمان یادگیری وضعیت جدید در مدل های Q-L, Q-k2, Q-k1, Q - G خواهد شد. باید توجه داشت که هرچند انتخاب مقادیر بزرگتر برای q احتمال بیشتری برای یادگیری وضعیت جدید ایجاد می کند اما انتخاب هر مقدار غیر صفر برای آن می تواند باعث کشف وضعیت جدید شود و بنابراین نسبت زمان لازم برای یادگیری یک وضعیت جدید توسط مدل Q-G با افزایش N به طور کلی افزایش پیدا نخواهد کرد. لازم به ذکر است که مدل Q (با پارامتر ثابت) به طور متوسط 32.1 و مدل Q-LRP به 14.3 واحد زمانی برای یادگیری وضعیت جدید در محیط ۱۰ - ب نیاز داشته اند.

جدول (۱) مقایسه زمان یادگیری وضعیت جدید برای مدل های Q-LA

β	متوسط زمان لازم برای یادگیری وضعیت جدید
0	نامحدود
0.01	33.30
0.05	22.70
0.1	14.30
0.2	15.00
0.4	12.50
0.6	13.75
0.8	7.25
1.0	9.00

۵ـ۱ـ۲ـ۲ـ تنظیم پارامتر انتخاب تصادفی با اتوماتون های با ساختار ثابت

به منظور بررسی انعطاف پذیری «مدل های Q با پارامتر انتخاب تصادفی تنظیم شده توسط اتوماتون های ساختار ثابت » آزمایش قبل برای مدل های Q-K1, Q-G, Q-L, Q-FA و Q-K2 (با عمق های حافظه ۵، ۱۰ و ۲۰ برای چهار مدل آخر) تکرار شد. نتایج این آزمایش در جدول ۲ خلاصه شده است. آنچه که در اینجا اتفاق می افتد درست مانند مطالب گفته شده در بخش قبل است، با این تفاوت که وضعیت درونی اولیه اتوماتون های ساختار ثابت در محیط ۱۰ ـ ب متفاوت است. شکل های ۱۲، ۱۳ و ۱۴ به ترتیب وضعیت درونی اولیه اتوماتون های تنظیم کننده θ در مدل های Q-L, Q-FA و Q-Gرا در محیط ۱۰ ـ ب نشان می دهد (این وضعیت پس از ۲۰۰۰ واحد زمان شبیه سازی در محیط ۱۰ ـ الف ایجاد شده است). باتوجه به این شکل ها نرخ تغییر پارامتر θ با دریافت پسخورهای ناموفق متوالی در محیط جدید در مدل Q-FA بیشتر از سایر مدل هاست. مطابق شکل ۱۲ اتوماتون FA با دریافت هر پسخور ناموفق پارامتر θ در Q - FA را تغییر خواهد داد و همانطور که در جدول ۲ هم پیداست این مدل بیشترین انعطاف را داشته است.

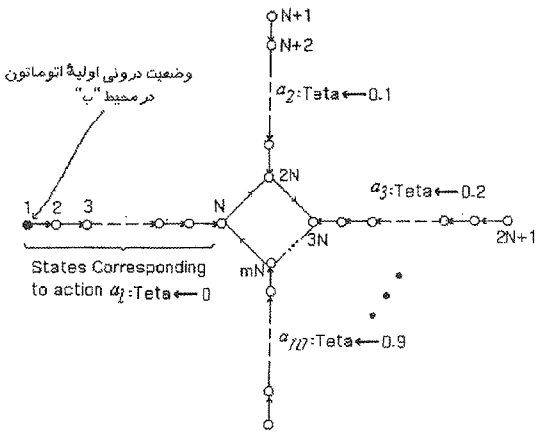


شکل (۱۲) وضعیت درونی اولیه اتوماتون FA از مدل Q-FA

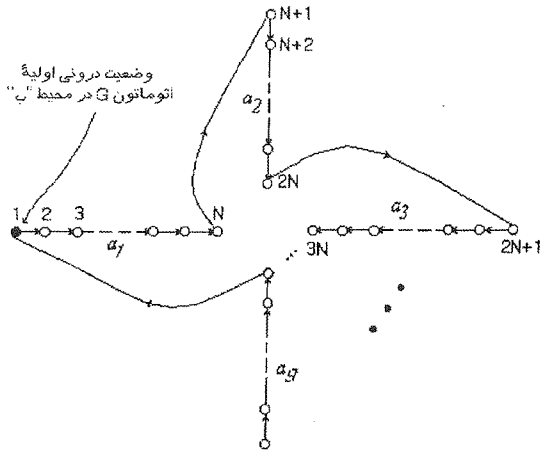
پس از مدل Q-FA مدل Q-L بیشترین نرخ تغییر پارامتر θ را خواهد داشت. باتوجه به شکل ۱۳ اتوماتون L در N واحد زمان اولیه با دریافت پسخورهای ناموفق تغییری در پارامتر نخواهد داد ولی از آن پس با دریافت هر پسخور ناموفق اتوماتون پارامتر θ در مدل Q-L را تغییر خواهد داد. اما مطابق شکل ۱۴ اتوماتون G برای هر تغییر در پارامتر θ نیاز به دریافت N پسخور ناموفق دارد و بنابراین مدل Q-G کمترین انعطاف را نشان داده است. از آنجایی که رفتار مدل Q-K1 در هنگام دریافت پسخورهای ناموفق معادل رفتار مدل Q-L است، نتیجه آزمایش ها بر روی این مدل مشابه مدل Q-L بوده است. با استدلال های مشابهی می توان انتظار داشت که انعطاف پذیری مدل Q-K2 (که به حداقل N بار پسخور ناموفق برای تغییر فعالیت انتخاب شده نیاز دارد) بسیار پایین باشد. همچنین واضح است که افزایش N در اتوماتون های با ساختار ثابت باعث افزایش زمان یادگیری وضعیت جدید در مدل های Q-K1, Q-G, Q-L و Q-K2 خواهد شد. باید توجه داشت که هر چند انتخاب مقادیر بزرگتر برای θ احتمال بیشتری برای یادگیری وضعیت جدید ایجاد می کند، اما انتخاب هر مقدار غیر صفر برای آن می تواند باعث کشف وضعیت جدید شود و بنابراین نسبت زمان لازم برای یادگیری یک وضعیت جدید توسط مدل Q-G با افزایش N به طور خطی افزایش پیدا نخواهد کرد. لازم به ذکر است که مدل Q (با پارامتر ثابت) به طور متوسط به 32.1 و مدل Q-LRP به 14.3 واحد زمان برای یادگیری وضعیت جدید در محیط ۱۰ ـ ب نیاز داشته اند.

جدول (۲) مقایسه زمان یادگیری وضعیت جدید
برای مدل های Q-LA

مدل	N	متوسط زمان لازم برای یادگیری وضعیت جدید
R	1	5.4
Q - FA	1	9.6
Q-L	5	12
Q - L	10	18.2
Q - L	20	27.2
Q - G	5	12.8
Q - G	10	27
Q - G	20	54.4
Q - K1	5	10.6
Q - K1	10	22.3
Q - K1	20	31.6
Q - K2	5	39.3
Q - K 2	10	67.3
Q - K2	20	446



شکل (۱۳) وضعیت درونی اولیه اتوماتون L از مدل
Q-L در محیط ۱۰ ـ ب



شکل (۱۴) وضعیت درونی اولیه اتوماتون G از مدل
Q-G در محیط ۱۰ ـ ب

۵ ـ ۱ ـ ۳ ـ مقایسه قدرت اکتشاف

در مدل های یادگیری از پارامتر انتخاب تصادفی (θ) به منظور کشف وضعیت های جدید استفاده می شود. تکنیک اکتشاف به وسیله انتخاب تصادفی فعالیت ها به تکنیک اکتشافی غیرمستقیم [14] معروف است. پیاده سازی این تکنیک اکتشافی ساده، اما غیر کارا است [11] (در [10] ثابت شده است که در این تکنیک با افزایش تعداد وضعیت ها زمان یادگیری به طور نمایی افزایش پیدا می کند). بنابراین معمولاً هنگامی که تعداد وضعیت ها زیاد است از تکنیک های اکتشافی مستقیم [7] [17] [15] استفاده می گردد. در این تکنیک ها برای اکتشاف محیط از اطلاعات آماری کسب شده در تجربیات استفاده می شود (ثابت شده است که با افزایش تعداد وضعیت ها زمان یادگیری با این تکنیک ها به صورت یک چند جمله ای درجه پایین رشد می کند). در آزمایش های انجام شده در این قسمت از یک نمونه از این مدل ها که آن را DQ می نامیم استفاده شده است [4].

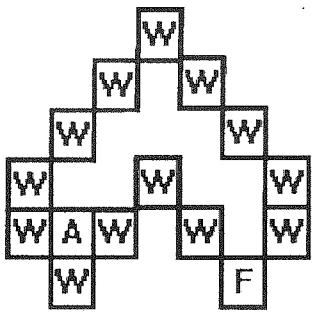
برای گنجاندن قدرت اکتشاف بیشتر در مدل های Q با پارامترهای تنظیم شونده روش محاسبه پسخور مأمورهای یادگیری تنظیم کننده پارامتر θ به صورت زیر تغییر داده شده است:

$$\text{IF} \sum_{i=t-p}^{t} r_i^Q > \sum_{i=T-p}^{T} r_i^Q \text{ THEN } r^{LA} = \text{MAX}_{i=1}^{t}(r_i^Q) \quad (۳۴)$$

$$\text{ELSE } r^{LA} = \text{MIN}_{i=1}^{t}(r_i^Q)$$

شرایط بالا تضمین می کنند تا در صورت عدم

افزایش بیشترین پسخور دریافتی در پریودها مقدار پارامتر تصادفی مدل (و در نتیجه میزان اکتشاف آن) پسخور کمینه دریافت کند. برای مقایسه قدرت اکتشاف از محیط شکل ۱۵ استفاده شده است. در این آزمایش قدرت اکتشاف مدل ها برحسب زمان لازم برای یافتن غذا در محیط اندازه گیری شده است. جدول ۳ قدرت اکتشاف و عملکرد در طی زمان اکتشاف مدل های Q - L, Q - FA, DQ , Q (به ترتیب با عمق های حافظه ۵، ۱۰ و ۲۰)، Q - G (به ترتیب با عمق های حافظه ۵، ۱۰ و ۲۰) و Q - LRP (با پارامترهای $\alpha = \beta = 0.1$) را مقایسه می کند. مقادیر پارامترهای ثابت مدل های Q با «پارامتر ثابت »(Q و DQ) و «تنظیم شده» (Q-G, Q-L, Q-FA, Q و Q-LRP) بدین صورت انتخاب شده اند: $k = 5, \delta = 10, \varepsilon = 0.000001, \rho = 2, \lambda = 0.5, \gamma = 0.9$ و $\theta = 0.1$ (فقط برای مدل های Q با پارامتر ثابت) و period = 1 (برای مدل های Q با پارامتر تنظیم شده). در آزمایش های این بخش زمان متوسط برای یافتن غذا در محیط با معدل گیری از پنج بار اجرای هر آزمایش محاسبه شده است.



شکل (۱۵) آزمایش قدرت اکتشاف

همانطور که جدول ۳ نشان می دهد، مدل های «Q با پارامتر تنظیم شده توسط اتوماتون های با ساختار ثابت» (Q-K1, Q-G, Q-L و Q-K2) بیشترین قدرت اکتشاف را در محیط داشته اند. اما متوسط پسخور دریافتی در طی زمان اکتشاف برای این مدل ها بسیار پایین بوده است (عملکرد این مدل ها حتی از مأمور تصادفی هم بدتر بوده است). در بین مدل های آزمایش شده عملکرد مدل Q در محیط بهترین بوده است. اما این مدل ضعیفترین قدرت اکتشاف را داشته است. در مقابل مدل هایی که از اتوماتون های با ساختار ثابت برای تنظیم پارامتر استفاده کرده اند، بهترین اکتشاف را داشته اند. اما عملکرد مناسبی را در زمان اکتشاف نمایش نداده اند (همانطور که پیداست قدرت اکتشاف این مدل ها با عمق حافظه آنها نسبت عکس داشته است). در این آزمایش مدل های Q-LRP و DQ مناسبترین جفت قدرت اکتشاف و عملکرد را نشان داده اند.

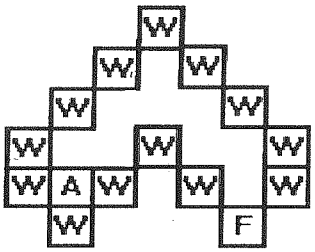
جدول (۳) مقایسه قدرت اکتشاف مدل ها

نام مدل	زمان متوسط برای یافتن غذا (قدرت اکتشاف)	متوسط پسخور طی زمان اکتشاف (عملکرد)
R	147	-22
Q	10357	-2.2
Q-L , N = 5	147	-22.6
Q - L , N = 10	109	-23
Q - L, N = 20	85	-24
Q - G , N = 5	119	-24.3
Q - G , N = 10	112	-24
Q - G , N = 20	113	-22.6
Q - K1 , N = 5	39	-25.2
Q - K1 , N = 10	72	-20.8
Q - K1 , N = 20	151	-22.4
Q - K2 , N = 5	63	-18.0
Q - K 2 , N = 10	113	-22
Q - K2 , N = 20	147	-22.4
Q - LRP	537	-8.4
DQ	361	-12.8

۵ـ۱ـ٤ـ یادگیری توالی عملیات

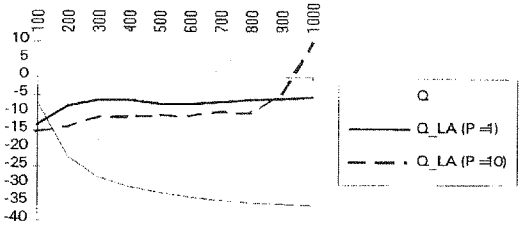
برای بررسی قدرت یادگیری یک توالی از عملیات از پیکربندی شکل ۱۶ استفاده شده است. در این آزمایش مأمور یادگیر باید توالی عملیات برای یافتن غذا در محیط را کشف می کرد (توالی این عملیات از خانه شروع عبارتند از: حرکت به بالا، به راست، به پایین و به بالا) این آزمایش به مدت ۱۰۰۰ واحد زمان برای هر مأمور شبیه سازی شده و در طول شبیه سازی هر بار مأمور به غذا می رسید، دوباره در خانه شروع قرار می گیرد. بدیهی است که یادگیری توالی عملیات باعث افزایش عملکرد مأمور در این محیط می شود و در نتیجه عملکرد

هر مأمور در طول شبیه سازی می تواند نمایانگر میزان یادگیری توالی عملیات توسط مأمور باشد. در نمودار ۱۰ تغییرات عملکرد مدل های Q و Q-LRP (θ) آورده شده است. مقادیر پارامترها در این آزمایش مانند آزمایش بخش قبل انتخاب شده اند، با این تفاوت که مدل Q-LRP (θ) یک بار با مقدار period = 1 و یک بار با مقدار period = 10 در نظر گرفته شده است.



شکل (۱۶) یادگیری توالی عملیات

مأمورهای یادگیر آزمایش شده در محیط شکل ۱۶ برای یادگیری توالی عملیاتی که منتهی به یافتن غذا می شود، نیاز به چندین بار تکرار این توالی عملیات دارند و اولین باری که این مأمورها به غذا می رسند، نه تنها موفق به فراگیری توالی عملیات نمی شوند، بلکه به خاطر تعمیم هایی که در دسته بندی آماری انجام می گیرد، عملکرد مأمورها پس از قرار گرفتن در خانه شروع کاهش پیدا خواهد کرد. اما کاهش عملکرد مأمورهایی که پارامتر θ در آنها تنظیم می شود، به طور غیر مستقیم باعث افزایش مقدار θ در آنها می شود. بنابراین مأمورها انتخاب های تصادفی بیشتری انجام خواهند داد. انجام انتخاب های تصادفی بیشتر منتهی به اکتشاف بیشتر و در نتیجه تکرار عملیات لازم برای یافتن غذا در محیط می شود و در نتیجه این مأمورها قادر خواهند بود تا توالی عملیات را فرا بگیرند.



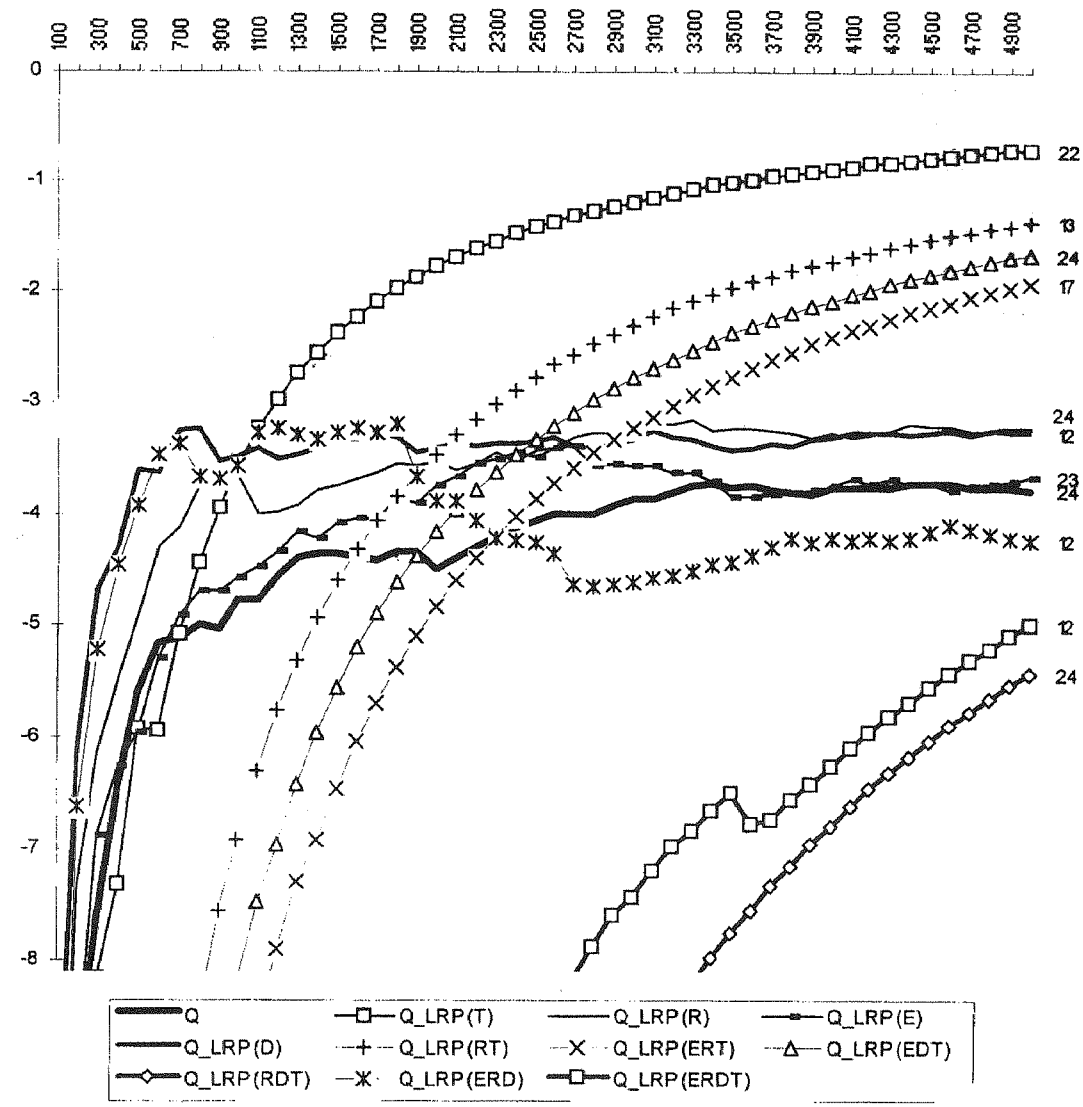
نمودار (۱۰) مقایسه قدرت یادگیری توالی عملیات

۵-۱-۵- تنظیم سایر پارامترهای یادگیری

پیکربندی محیط در این آزمایش ها همان پیکربندی شکل ۹ است. مشخصات اتوماتون های تنظیم کننده پارامتر (های) مدل ها در این آزمایش ها از این قرار می باشد: اتوماتون تنظیم کننده θ شامل ۱۰ فعالیت است که یکی از مقادیر 0, 0.1, 0.2, ... و 0.9 را برای θ انتخاب می کنند. اتوماتون تنظیم کننده ε دارای ۱۰ فعالیت برای انتخاب یکی از مقادیر 0, 0.2, 0.4, 0.6, 0.8 و 1.0 برای ε است، اتوماتون تنظیم کننده ρ یکی از ۵ مقدار 0, 1, 2, 3 و 4 را برای ρ و اتوماتون تنظیم کننده σ یکی از ۱۰ مقدار 1, 5, 10, 15, ... و 45 را برای σ انتخاب می کند. در این آزمایش ها پارامترهای K, λ و γ تنظیم نشده اند و مقدار این پارامترها برای تمام مدل ها ثابت و به ترتیب برابر با 0, 1, 5 در نظر گرفته شده است. همچنین هر جا پارامترهای θ, ε, ρ و σ تنظیم نشده باشند، مقدار آنها به ترتیب برابر با 10, 0.000001, 2, 20 قرار داده شده است (این مقادیر برای آزمایش مشابهی در [10] پیشنهاد شده اند). مقدار period برای تمام اتوماتون ها برابر با 10 واحد زمان در نظر گرفته شده است. در آزمایش های این بخش هر مدل سه بار در محیط شبیه سازی شده و متوسط این سه آزمایش در نمودارها و جداول آمده است.

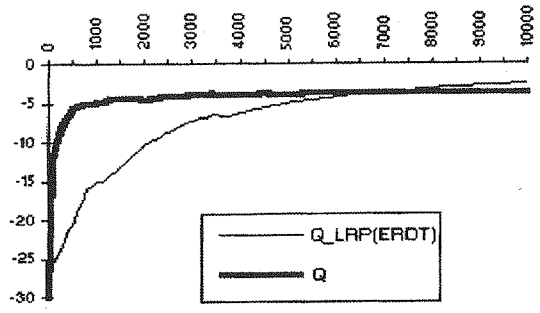
۵-۱-۵-۱- تنظیم سایر پارامترها توسط اتوماتون های با ساختار متغیر

تمام اتوماتون های مورد استفاده در آزمایش های این قسمت از نوع LRP می باشند. در نمودار ۱۱ نحوه تغییر پسخورهای کل دریافتی از محیط در طول ۵۰۰۰ واحد زمانی برای ۱۱ مدل مختلف مقایسه شده است. مزایای تنظیم پارامتر θ قبلاً بررسی شد. با محاسبه می توان نشان داد که در محیط ساده با پیکربندی شکل ۹ یک مدل با پارامتر ثابت θ = 10 حداکثر قادر است به پسخور متوسط 2.83- برسد، اما مدلها با تنظیم θ می توانند در تئوری به پسخور متوسط 0 برسند (که بهترین عملکرد ممکن در این محیط است). در نمودار ۱۱ همگرا شدن عملکرد مدلهای Q-LRP (RT), Q-LRP (T) و Q-LRP (ERT) به سمت 0 قابل مشاهده است. گرچه افزایش تعداد پارامترهای تنظیم شده باعث شده تا سرعت همگرایی در مدل های Q-LRP (RDT), Q-LRP (EDT) و Q-LRP (ERDT) تا حدودی کاهش یابد، ولی آزمایش ها نشان داده اند که این مدلها نیز با صرف زمان بیشتر عملکرد بهتری را نسبت به Q نشان می دهند (نمودار ۱۲ تغییرات پسخور متوسط مدل Q-LRP (ERDT) را در ۱۰۰۰۰ واحد زمان دنبال می کند).

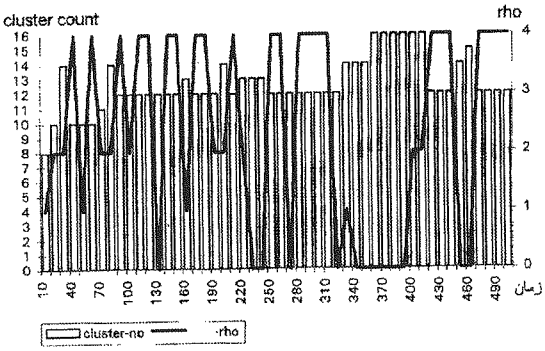


نمودار (۱۱) مقایسه تغییرات عملکرد مدل های مختلف در زمان

در کنار هر کدام از منحنی های نمودار ۱۱ تعداد نهایی دسته هایی که مدل پس از خبرگی به آنها رسیده، نوشته شده است. می توان نشان داد که حداقل تعداد دسته هایی که مأمور در این محیط احتیاج دارد ۱۲ است (این دسته ها معنی دارترین دسته ها برای یادگیری این محیط را نیز تشکیل می دهند). همانطور که مشاهده می شود مدلهایی که پارامتر ρ در آنها تنظیم شده است، اکثراً قادر به خبرگی با ۱۲ دسته شده اند. اما سایرپارامترها این خاصیت را نداشته اند. در نمودار ۱۳ نحوه تغییر تعداد دسته ها با مقدار ρ در طول زمان آورده شده است. همانطور که در نمودار مشخص است، مقدار $\rho = 4$ موجب کاهش تعداد دسته ها در طول یادگیری شده است. در آزمایش دیگری از مدل Q با مقدار پارامتر ثابت $\rho = 4$ استفاده شد. در این آزمایش مدل با ۱۲ دسته به خبرگی رسیده است. نمودار ۱۳ همچنین نشان می دهد که مدلهای Q LRP (R), Q-LRP (E) و Q-LRP (D) عملکرد بهتری نسبت به مدل Q داشته اند.



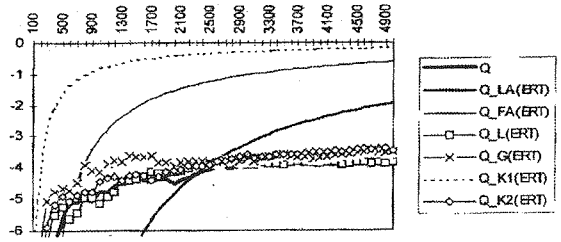
نمودار (۱۲) مقایسه تغییرات پسخور متوسط مدل های Q-LRP (ERDT) و Q در ۱۰۰۰۰ واحد زمان



نمودار (۱۳) تغییرات ρ و تعداد دسته با زمان در مدل Q-LRP (R)

۵-۱-۵-۲-تنظیم سایر پارامترها با اتوماتون های ساختار ثابت

پیکربندی محیط ساده در این آزمایش ها همان پیکربندی شکل ۹ است و مقادیر ثابت و متغیر پارامترها مانند بخش قبل انتخاب شده اند. نتایج آزمایش ها در نمودارهای ۱۴ تا ۱۶ آورده شده است . نتایج آزمایش ها نشان می دهند که در بعضی موارد مدل ها نمی توانند بهترین مقادیر را برای پارامترهای تنظیم کننده پیدا کنند و در بهینه های محلی قرار می گیرند. به عنوان مثال در جدول ۴ مقدار پارامتر θ در انتهای شبیه سازی نمودار ۱۴ آورده شده است. همانطور که در این جدول ملاحظه می شود، پس از خبره شدن مدل در محیط پارامتر θ در بعضی موارد به جای آنکه به مقدار 0 همگرا شود به مقدار 0.1 همگرا شده است. اما با این وجود عملکرد مدل های تنظیم کننده پارامترها در اکثر موارد معادل و یا بهتر از مدل Q با پارامتر ثابت بوده است. به نظر می رسد که با در نظر گرفتن توابع بهتری برای تعیین پسخور اتوماتون های تنظیم کننده پارامترها و تنظیم عمق حافظه این اتوماتونها بتوان به نتایج بهتری نیز دست یافت.



نمودار (۱۴) مقایسه عملکرد مدل های مختلف با تنظیم همزمان پارامترهای ε, ρ و θ

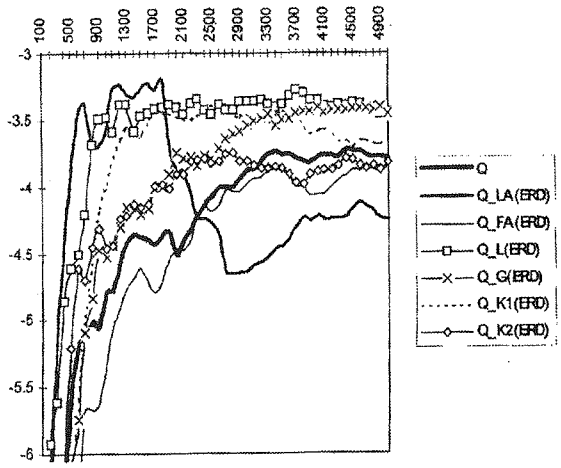
جدول (۴) مقدار نهایی θ پس از شبیه سازی نمودار ۸.

نام مدل	Q-LRP	Q-FA	Q-L	Q-G	Q-K1	Q-K2
مقدار θ	۰	۰	۱۰	۱۰	۰	۱۰

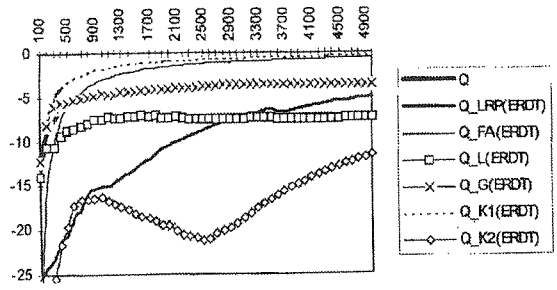
در جدول ۵ تعداد دسته ها در انتهای شبیه سازی های نمودارهای ۱۴ تا ۱۶ آورده شده است. می توان نشان داد که حداقل تعداد دسته هایی که مأمور در محیط شکل ۹ احتیاج دارد ۱۲ است (این دسته ها معنی دارترین دسته ها برای یادگیری این محیط را نیز تشکیل می دهند). اما همانطور که مشاهده می شود مدل هایی که پارامترهای آنها توسط اتوماتون های با ساختار ثابت تنظیم شده است نتوانسته اند به تعداد دسته های بهینه همگرا شوند و در این مورد تنظیم پارامترها با اتوماتون های ساختار متغیر نتایج بهتری را در برداشته است.

جدول (۵) تعداد دسته ها پس از شبیه سازی

	Q	Q - LRP	Q - FA	Q -L	Q - G	Q-K1	Q-K2
ε, ρ, θ	۲۴	۱۷	۳۲	۱۲	۳۲	۱۰	۳۲
ε, ρ, σ	۲۴	۱۲	۱۲	۲۱	۲۰	۱۶	۲۳
ε, ρ, σ, θ	۲۴	۱۲	۳۲	۲۹	۴۶	۱۱	۱۵



نمودار (۱۵) مقایسه عملکرد مدل های مختلف با تنظیم همزمان پارامترهای ε ,ρ و σ



نمودار (۱۶) مقایسه عملکرد مدل های مختلف با تنظیم همزمان پارامترهای θ ,ε ,ρ و σ

۶ـ محیط رفتاری (محیط جعبه ها)

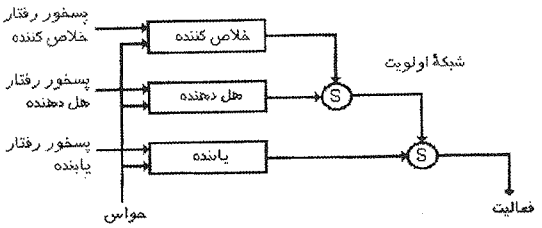
مسئله هل دادن جعبه ها یک آزمایش کلاسیک برای مقایسه رفتار مأمورهای یادگیر است [13] [10]. در نمونه ساده شده این مسئله یک مدل یادگیر مأموریت دارد تا جعبه هایی را در یک اتاق جابجا کند. در آزمایش های انجام شده در این بخش مأمورهای یادگیری بررسی می شوند که دارای ساختار رفتاری هستند. در [10] یک ساختار رفتاری برای مسئله هل دادن جعبه ها آمده است که در اینجا عیناً از آن استفاده می شود. در این ساختار مأمور یادگیر دارای سه رفتار است:

۱ ـ رفتار یافتن جعبه (یابنده)

۲ ـ رفتار هل دادن جعبه (هل دهنده)

۳ ـ رفتار خلاص کردن در صورت گیر افتادن (خلاص کننده)

مأمور برای یادگیری هر کدام از رفتارها سه کپی از مدل یادگیری را در خود دارد. مطابق معماری لایه ای هر مأمور یادگیر باید ترتیبی برای انتخاب رفتار برتر در هر موقعیت را داشته باشد. برای این منظور به رفتار خلاص سازی بیشترین اولویت داده شده است، هل دادن جعبه اولویت بعدی را دارد و نهایتاً پیدا کردن جعبه ها دارای کمترین اولویت است.



شکل (۱۷) نمای کلی از معماری لایه ای محیط جعبه ها

در جدول های ۶ تا ۱۰ قسمت فعال کننده و سیاستگزاری هر کدام از رفتارها آورده شده است [10]. تابع سیاستگزاری در واقع پسخور دریافت شده توسط هر رفتار را مشخص می کند. جدول ۶ تابع سیاستگزاری رفتار یابنده را مشخص می کند. همانطور که پیداست این سیاست مأمور را به روبرو شدن با اشیاء (که ممکن است جعبه باشند) تشویق می کند. رفتار یابنده دارای تابع فعال کننده نیست و هرگاه رفتارهای با اولویت بیشتر (یعنی رفتارهای هل دهنده و خلاص کننده) فعال نباشند، این رفتار فعال خواهد شد.

جدول ۷ تابع سیاستگزاری رفتار هل دهنده را نشان می دهد. مطابق این سیاستگزاری مأمور تا زمانی که شیئی در مقابلش باشد و به جلو حرکت کند پسخور مثبت دریافت خواهد کرد و هرگاه شیء مقابل خود را از دست بدهد، پسخور منفی دریافت می کند. تابع فعال کننده رفتار هل دهنده (جدول ۸) تا زمانی که شیئی در مقابل مأمور باشد، فعال خواهد بود. در طی هل دادن جعبه ممکن است مأمور اشتباهاتی بکند و جعبه مقابل خود را از دست بدهد. برای آنکه مأمور قادر به تصحیح چنین اشتباهاتی باشد، تابع فعال کننده مأمور به گونه ای طراحی شده است تا رفتار هل دهنده حتی وقتی مأمور جعبه ای در مقابل ندارد نیز تا چند واحد زمانی معین فعال باقی بماند.

جدول (۶) تعریف تابع سیاستگزاری برای رفتار یابنده در محیط جعبه ها با معماری رفتاری

فعالیت	شرط دریافت پسخور	پسخور
حرکت به جلو	پس از اعمال فعالیت یک شیء در مقابل مأمور قرار گیرد	+ 300
حرکت به جلو، تغییر جهت	پس از اعمال فعالیت شیئی در مقابل مأمور نباشد	-100
حرکت به جلو، تغییر جهت	–	0

جدول (۷) تعریف تابع سیاستگزاری برای رفتار هل دهنده در محیط جعبه ها با معماری رفتاری

فعالیت	شرط دریافت پسخور	پسخور
حرکت به جلو	پس از اعمال فعالیت یک شیء در مقابل مأمور قرار گیرد	+ 300
حرکت به جلو، تغییر جهت	پس از اعمال فعالیت شیئی در مقابل مأمور نباشد	-100
حرکت به جلو، تغییر جهت	–	0

جدول (۸) تعریف تابع فعال کننده رفتار هل دهنده در محیط جعبه ها با معماری رفتاری

شرط	نتیجه	خروجی
یک شیء در مقابل مأمور باشد	Pusher-history = 5	TRUE
pusher - history > 0	pusher - history = pusher - history - 1	TRUE
هیچکدام از موارد بالا	-	FALSE

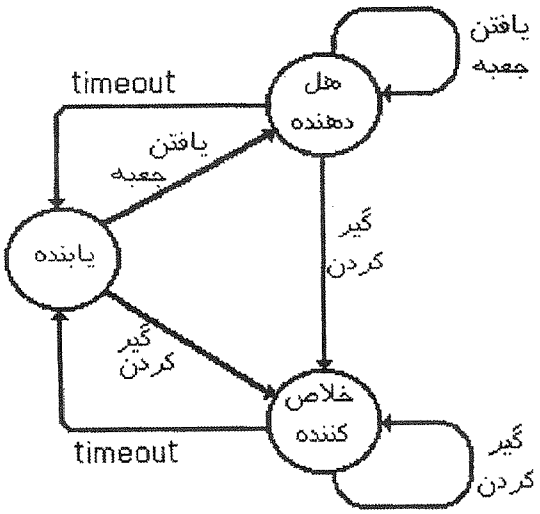
جدول (۹) پسخور خلاص کننده

فعالیت	شرط دریافت پسخور	پسخور
حرکت به جلو	مکان مأمور تغییر نکند	+ 100
حرکت به جلو، تغییر جهت	مکان مأمور تغییر نکند	- 300
حرکت به جلو، تغییر جهت	–	0

جدول (۱۰) تابع فعال کننده رفتار هل دهنده

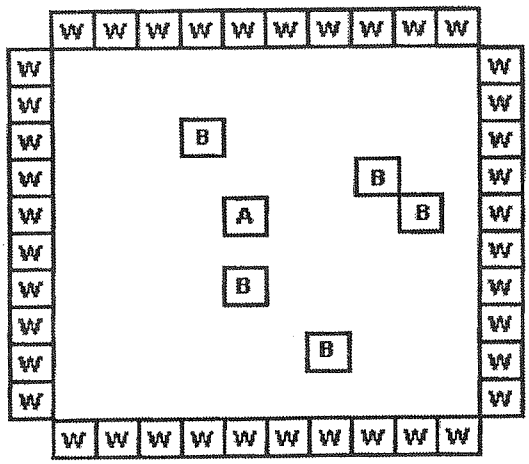
شرط	نتیجه	خروجی
مأمور گیر کرده باشد	unwedger-history = 5	TRUE
unwedger - history > 0	unwedger - history = unwedger - history - 1	TRUE
هیچکدام از موارد بالا	-	FALSE

تابع سیاستگزاری رفتار خلاص کننده در جدول ۹ آورده شده است. این تابع هرگاه مأمور گیر افتاده باشد، پسخوری منفی ایجاد خواهد کرد. گیر افتادن مأمور در حالتی است که مأمور فعالیت حرکت به جلو را انتخاب کند، اما تغییر مکان ندهد (مقدار مشخصه های "x" و "y" مأمور ثابت بماند). تابع سیاستگزاری هر وقت مأمور از این حالت خارج شود، پسخور مثبت ایجاد خواهد کرد. تابع فعال کننده رفتار خلاص کننده (جدول ۱۰) تا زمانی که مأمور گیر افتاده باشد فعال می ماند. این تابع نیز مانند تابع فعال کننده هل دهنده تا مدت زمان معینی پس از خلاص شدن مأمور فعال باقی می ماند. شکل ۱۸ چگونگی جریان فعال شدن هر رفتار را در معماری لایه ای محیط جعبه ها را خلاصه کرده است. در ضمن نحوه عملکرد حواس در این محیط دقیقاً مانند محیط ساده (جدول ۳) است.

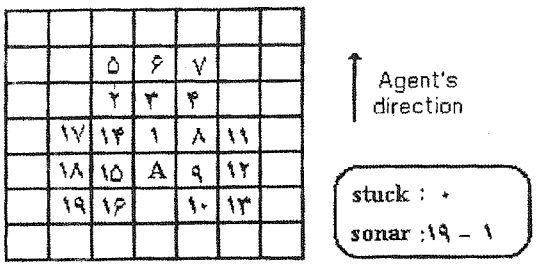


شکل (۱۸) جریان فعال شدن رفتارها

پیکربندی اولیه محیط در آزمایش هایی که ارائه خواهد شد، مطابق شکل ۱۹ می باشد. در این پیکربندی مربعهایی که در آنها حرف (B) قرار گرفته است نمایانگر جعبه، مربعهایی که در آنها حرف (W) قرار گرفته نمایانگر دیوار و مربعی که در آن حرف (A) قرار دارد نشان دهنده مأمور می باشد. مأمور یادگیر در این آزمایش ها مجهز به فعالیت های تغییر جهت به راست (به اندازه ۹۰ درجه)، تغییر جهت به چپ (به اندازه ۹۰ درجه) و حرکت به جلو است. مأمور یادگیر با اعمال فعالیت حرکت به جلو می تواند جعبه ها را در محیط جابجا کند. این جابجایی وقتی اتفاق می افتد که یک جعبه در راستای حرکت مأمور قرار داشته باشد و خانه ای که جعبه به طرف آن هل داده می شود، خالی باشد. همچنین حواس مأمور یادگیر در این محیط شامل ۲۰ بیت می باشد. اولین بیت زمانی یک می شود که مأمور یادگیر «گیر» بیفتد. این زمانی است که مأمور فعالیت حرکت به جلو را انتخاب کند، اما تغییر مکان ندهد (به عبارتی در راستای حرکت مأمور یا جعبه ای که هل می دهد شیئی قرار گرفته باشد). ۱۹ بیت دیگر هر کدام به خانه ای در اطراف مأمور اشاره می کنند. هر زمان که در خانه ای که یک بیت به آن اشاره می کند، شیئی قرار داشته باشد آن بیت یک شود. در شکل ۲۰ مکانهایی که بیتها به آن اشاره می کنند، مشخص شده است. در آزمایش های این بخش هر مدل سه بار در محیط شبیه سازی شده و نتیجه شبیه سازی با عملکرد متوسط در نمودارها و جدول ها استفاده شده است.



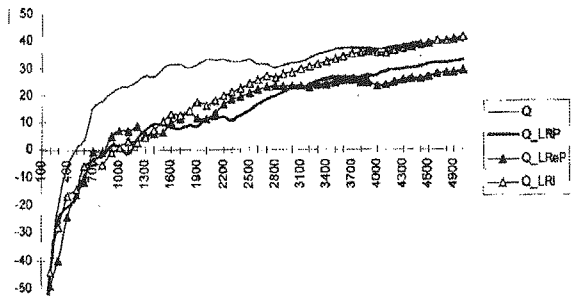
شکل (۱۹)



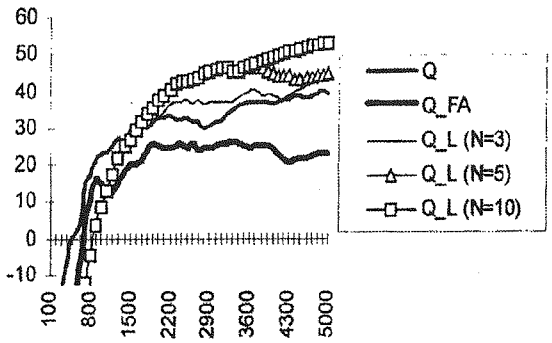
شکل (۲۰)

هدف معماری رفتاری این است که با شکستن یک مسئله پیچیده به رفتارها (یا ماجول های) کوچکتر و ساده تر موجب افزایش سرعت یادگیری مأمور یادگیر شود. در این معماری برای بروز رفتار دلخواه توسط مأمور یادگیر لازم است تا یادگیری تمام رفتارها باعث یادگیری کل مسئله شود. اما تحقیق صحت این امر کار بسیار مشکلی است. به عنوان مثال در انتهای تقریباً تمام شبیه سازی های این محیط مأمورهای یادگیر به جای آنکه بیشتر وقت خود را به هل دادن جعبه ها بپردازند، در کنار دیوارها حرکت می کردند (تعداد کم جعبه های هل داده شده که قبلاً نیز در [10] گزارش شده است). چرا که در اینجا حرکت مأمورهای یادگیر در کنار دیوار موجب بیشینه سازی پسخور دریافتی از هر ماجول می شود، اما مسلماً رفتار دلخواه را ایجاد نمی کند. بنابراین در اینجا مقایسه مأمورهای یادگیر براساس تعداد جعبه های هل داده شده نادرست خواهد بود و نتیجتاً در مقایسه های انجام شده تنها عملکرد نهایی مأمورها ملاک قرار گرفته است.

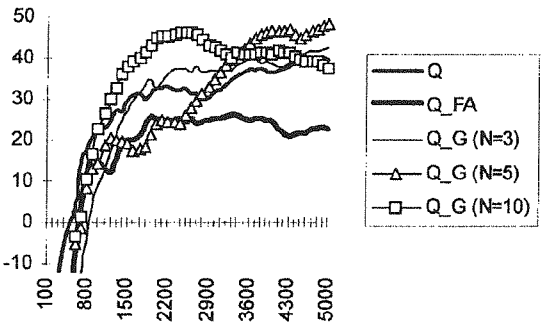
نمودارهای ۱۷ تا ۲۱ تغییرات عملکرد مدل های مختلفی که تنها پارامتر θ در آنها تنظیم شده است، را نشان می دهد. در نمودار ۱۷ تغییرات عملکرد مدل هایی که از اتوماتون های با ساختار متغیر (LReP, LRP) و (LRI) برای تنظیم پارامتر θ استفاده کرده اند، آورده شده است و در نمودارهای ۱۸ تا ۲۱ به ترتیب تغییرات عملکرد مدل هایی که از اتوماتون های K1, G, L و K2 با عمق های حافظه به ترتیب ۳، ۵ و ۱۰ استفاده کرده اند آورده شده است. در هر یک از این نمودارها عملکرد هر مدل با عملکرد مدل Q (بدون تنظیم پارامتر) مقایسه شده اند. همانطور که نمودار ۱۷ نشان می دهد استفاده از اتوماتون های با ساختار متغیر در تنظیم پارامتر θ باعث کاهش عملکرد مدل شده است. اما نمودارهای ۱۸ تا ۲۱ نشان می دهد که اتوماتون های ساختار ثابت عملکرد موفقی در تنظیم پارامتر θ داشته اند (در نمودار ۲۵ عملکرد اتوماتون های ساختار ثابت و متغیر در تنظیم پارامتر θ با یکدیگر مقایسه شده اند). نمودارهای ۲۲، ۲۳ و ۲۴ به ترتیب تغییرات پارامتر θ برای هر کدام از رفتارهای یافتن، هل دادن و خلاص کردن را در مدل Q-L (T) را نشان می دهد. همانطور که در نمودار ۲۲ مشخص است مقدار θ برای رفتار پیدا کردن جعبه ها در طول زمان شبیه سازی در نوسان بوده است، این امر منتهی به جستجوی تصادفی جعبه ها توسط مأمور می شود و این نوع جستجو احتمالاً باتوجه به محدود بودن حواس مأمور می تواند بهتر از جستجوی قانونمند باشد. نمودارهای ۲۳ و ۲۴ نشان می دهند که مقدار پارامتر θ برای رفتارهای هل دادن جعبه و خلاص کردن اکثراً صفر باقی مانده است که این امر باتوجه به ساده بودن یادگیری این دو رفتار قابل توجیه است.



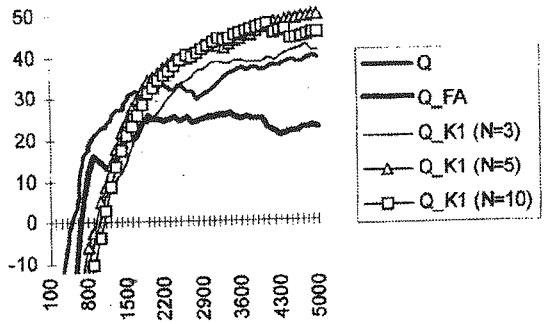
نمودار (۱۷)



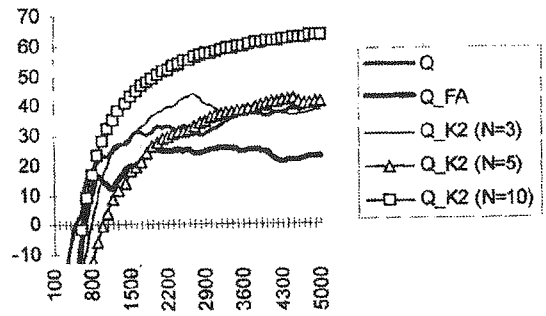
نمودار (۱۸)



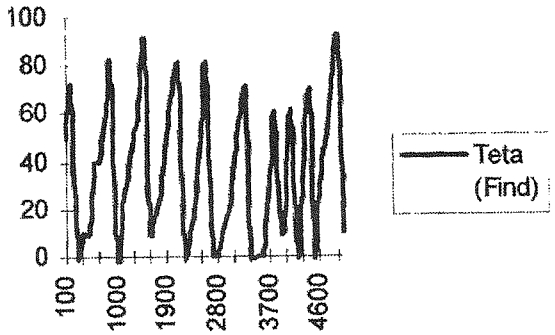
نمودار (۱۹)



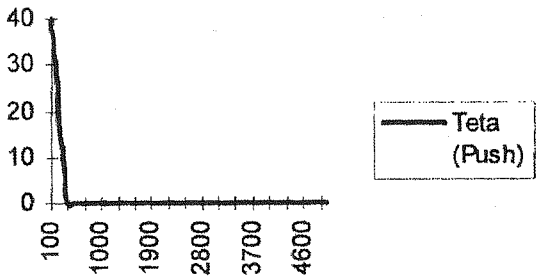
نمودار (۲۰)



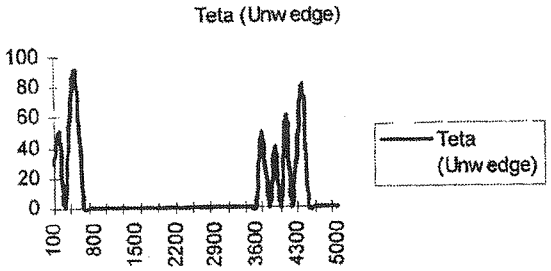
نمودار (۲۱)



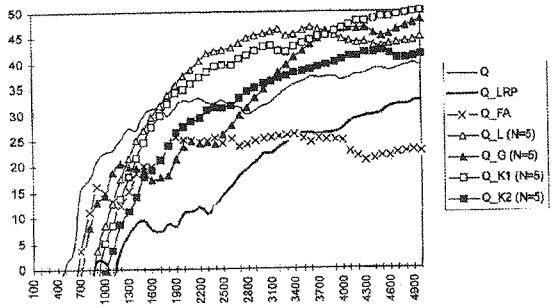
نمودار (۲۲) رفتار یافتن جعبه



نمودار (۲۳) رفتار هل دادن جعبه



نمودار (۲۴) رفتار خلاص کردن

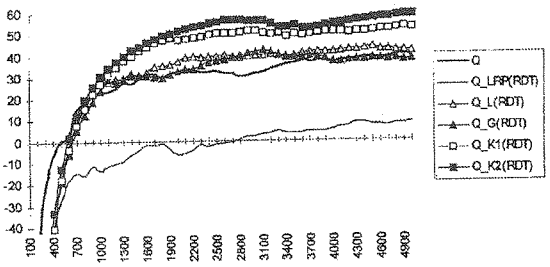


نمودار (۲۵) مقایسه عملکرد مدل های مختلف در تنظیم پارامتر انتخاب تصادفی

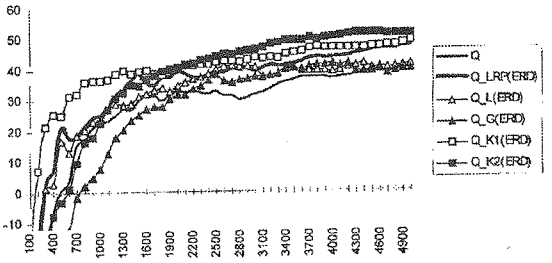
در نمودارهای ۲۶ تا ۲۹ عملکرد مدل هایی که از اتوماتون های ساختار ثابت (K_1, G, L) و(K_2) و متغیر (Q-LRP) برای تنظیم همزمان چند پارامتر استفاده می کنند با مدل Q مقایسه شده است. همانطور که ملاحظه می شود عملکرد مجهز به اتوماتون های با ساختار ثابت اکثراً بهتر از مدل های Q -LRP بوده است. برتری تنظیم پارامترها با اتوماتون های ساختار ثابت نسبت به اتوماتون های با ساختار متغیر بخصوص در مواردی که یکی از پارامترهای تنظیم شده θ باشد (نمودارهای ۲۶، ۲۸ و ۲۹) مشهودتر است. همچنین عملکرد مدل Q-K1 در نمودارهای ۲۶ و ۲۹ و عملکرد مدل Q-G در نمودار ۲۹ نشان می دهد که در این محیط نیز احتمال گرفتار شدن مدل ها در بهینه های محلی وجود دارد. آزمایش هایی که برای تنظیم ترکیبات مختلف پارامترهای مدل Q در این محیط انجام گرفته است، نشان می دهند که تنظیم پارامترهای ρ, δ و ε (به ترتیب از راست به چپ) موجب بیشترین افزایش ها در عملکرد

مأمور شده‌اند. این در حالی است که در محیط ساده‌ای که در بخش قبل شرح داده شد تنظیم پارامتر θ بیشترین سهم را در افزایش عملکرد مأمور را ایفا می‌کرد و تنظیم پارامترهای دیگر تأثیر چندانی در عملکرد مأمور نداشتند. همچنین برخلاف نتایج آزمایش‌های انجام شده در محیط ساده بخش قبلی، تنظیم پارامترهای δ ,ρ و ε در این محیط موجب تفاوت چندانی در تعداد دسته‌های نهایی نشده‌اند. بنابراین به نظر می‌رسد که تنظیم پارامترها در محیط‌های مختلف نتایج متفاوتی را در بردارد.

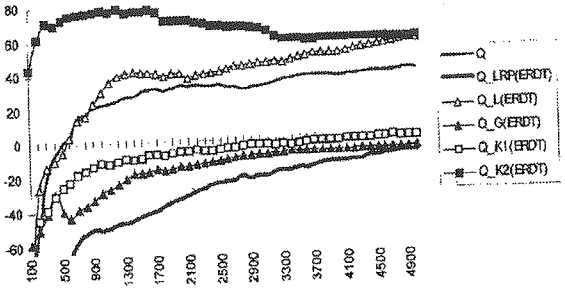
نمودار (۲۶)



نمودار (۲۷)



نمودار (۲۸)



نمودار (۲۹)

۷ ـ جمعبندی

مدل یادگیری Q دارای پارامترهای متعددی است و عملکرد بهینه این مدل به انتخاب مناسب این پارامترها بستگی دارد. مقادیر این پارامترها معمولاً با سعی و خطا و توسط طراح مدل انتخاب می‌شوند و در طول یادگیری بدون تغییر باقی می‌مانند. با تنظیم خودکار پارامترهای مدل یادگیری Q نه تنها مشکلات استفاده از روش‌های تجربی و سعی و خطا در یافتن مقادیر بهینه این پارامترها از بین می‌رود بلکه انعطاف‌پذیری بیشتری به مدل به منظور برخورد با یک محیط ناشناخته جدید می‌دهد. در این مقاله استفاده از اتوماتون‌های یادگیر با ساختار متغیر یا با ساختار ثابت برای تعیین و تنظیم پارامترهای مدل یادگیری Q مورد بررسی قرار گرفت. آزمایشات نشان می‌دهند که تنظیم پارامترها توسط این روش باعث افزایش عملکرد، انعطاف‌پذیری، قدرت اکتشاف و یادگیری توالی عملیات می‌گردد. نتایج آزمایشات نشان می‌دهند که تنظیم پارامترها توسط اتوماتون‌های با ساختار ثابت تقریباً در تمام مواقع نتایج بهتری نسبت به مدل کلاسیک (با پارامترهای ثبات) و مدلی که در آن از اتوماتون‌های با ساختار متغیر برای تنظیم پارامترها استفاده شده نشان می‌دهد.

زیرنویس‌ها

1 - Reinforcement Learning
2 - Feedback
3 - Statistical Clustering
4 - Linear - Reward - Inaction
5 - Linear - Reward - Penalty
6 - Linear - Reward - Epsilon Penalty
7 - Absorbing State

مراجع

[1] A. G. Barto, S. J. Bradthe and S. P. Singh, "Learning to Act Using Real-time Dynamic Programming", Artificial Intelligence, vol 72 (1), pp 81-138, 1995.

[2] M. Dorigo and H. Bersini, "A Comparison of Q learning & Classifier Systems," Proceedings of from Animates to Animals, International Conference on Simulation of Adaptive Behavior, SAB 1994.

[3] S. Hodjat and M. R. Meybodi, "Fine Tuning of Q-learning Parameters Using Learning Automata," Proceedingsot Second Annual Conference of Computer Society of Iran, pp 33-44, 1996.

[4] S. Hodjat, An Artificial Lab for Creating and Comparing Learning Algorithms , M. S. Thesis, Amirkabir University, 1997.

[5] L. P. Kaelbling, Learning in Embedded Systems, PhD. Thesis, Stanford University, Stanford, CA, 1990.

[6] L. P. Kaelbling, M.L. Littman and A.W. Moore, Reinforcement Learning, Artificial Intelligence Journal," Vol4, PP. 237-285, 1996.

[7] S. Koenig and R. G. Simmons. "Complexity Analysis of Real Time Reinforcement Learning," Proceedings of Eleventh Conference on Artificial Intelligence, AAAI-93, Menlo-park, CA, pp 99-105, 1993.

[8] V. I. Krinsky, "An Asymptotically Optimal Automaton with Exponential Convergence," Biofizica, vol 9, pp 99-105, 1964.

[9] V. Yu. Krylov, "On Stochastic Automaton which is Asymptotically Optimal in Random Medium," Automata and Remote Control, vol 24, pp. 1114-16, 1964.

[10] S. Mahadevan and J. Connel, "Automatic Programming of Behavior-based Robots Using Reinforcement learning, Artificial Intelligence Journal, vol 55, pp. 311-365, 1992.

[11] A. K. McCallum, Efficient Exploration in Reinforcement Learning with Hidden State, University of Rochester, 1996.

[12] M. R. Meybodi and S. Lakshmicarahan, "e - Optimality of a General Class of Obsorbing Barrier Learning Algorithms," Information Sciences, vol 28, pp. 1 - 20, 1982.

[13] T. M. Mitchell, "Generalization as Search, "Artificial Intelligence Journal, vol 18 (2), pp. 203-226, 1988.

[14] M. C. Mozar and J. R. Bacjrach, Discovering the Structure of a Reactive Environment by Exploration," Technical Report. CU-CS-451-8, Dept. of Computer Science, University of Colorado, Boulder, November 1989.

[15] M. Sato, K. Abe and H. Taheda, "Learning Control of Finite Markov Chains with an Explicit Trade off Between Estimation and Control," IEEE Transactions on Systems, Man and Cybernetics, vol 18 (5), September 1988.

[16] R. Schalkoff, Pattern Recognition, Wiley International Editions, 1991.

[17] S. B. Thrun, The Role of Exploration in Learning Control, In Handbook of Intelligent Control: Neuro, Fuzzy and Adaptive Approaches, Nostrand Reinhold, 1992.

[18] M. L. Tseltsin, On the Behavior of Finite Automata in Random Media: Automata and Remote Control, vol 22, pp. 1345-54, 1962.

[19] C. Watkins, Learning from Delayed Rewards, PhD. Thesis, Kings College, USA, 1989.

[20] S. D. Whitehead, "Complexity and Cooperaion in Q Learning," Proceedings of the Eighth International Workshop on Machine Learning, San Mateo, CA, Morgan Kaufmann, pp. 363-367, 1991.

[21] S. Hodjat and M. R. Meybodi, "Automatic Tuning of Q - Learning Parameters Using Game of Automata," Proceedings of Third Annual Conference of Computer Society of Iran, pp. 33-48, 1997.

[22] S. Hodjat and M. R. Meybodi, Tuning of Q-Learning Parameters Using Fixed Structure Automota, Proceedings of Third Annual Conference of Computer Society of Iran, pp. 1 - 16, 1997.

[23] k. S. Narendra and M. A. L. Tatachar, Learning Automata, Prentice Hall, Englewood cliffs, 1989.