روشی نوین برای بهبود عملکرد یادگیری Q با افزایش تعداد به‌روز رسانی مقادیر Q برپایه عمل متضاد

مریم پویان[۱]، امین موسوی[۲]، شهرام گلزاری[۳]، احمد حاتم[۴]

[۱] دانشجوی کارشناسی ارشد، گروه برق و کامپیوتر ، دانشگاه هرمزگان، بندرعباس
pouyan.student@hormozgan.ac.ir

[۲، ۳، ۴] استادیار، گروه برق و کامپیوتر ، دانشگاه هرمزگان، بندرعباس
[2]mousavi@hormozgan.ac.ir
[3]golzari@hormozgan.ac.ir
[4]hatam@hormozgan.ac.ir

چکیده

الگوریتم یادگیری Q، یکی از بهترین الگوریتم‌های یادگیری مستقل از مدل می‌باشد. هدف از یادگیری، یافتن تخمینی از تابع ارزش – عمل بهینه می‌باشد که مقادیر Q نامیده می‌شود. یکی از عمده ترین مشکلات روش یادگیری Q در برخورد با مسائل دنیای واقعی، زیاد شدن تعداد حالت‌های محیط و در نتیجه کم شدن سرعت همگرایی است، زیرا برای تضمین همگرایی یادگیری، تمامی زوج‌های حالت – عمل باید بی‌نهایت بار بازدید شود. در این نوشتار، از روش ترکیبی بر پایه مفاهیم عمل متضاد استفاده شده است. مفاهیم تضاد در یادگیری تقویتی منجر به بهبود سرعت همگرایی می‌شود، زیرا در آن به‌روز رسانی مقادیر Q برای عمل و عمل متضاد متناظر آن، در یک مرحله و بصورت همزمان انجام می‌پذیرد. روش ارائه شده همراه با یافتن بهترین اثر متقابل بین اکتساب و اکتشاف در یادگیری Q، برای افزایش سرعت همگرایی یادگیری استفاده شده است. تکنیک ارائه شده برای مسئله Grid world شبیه سازی شده است. نتایج به دست آمده بهبود در فرایند یادگیری را نشان می‌دهد.

کلمات کلیدی

یادگیری تقویتی، یادگیری Q، عمل متضاد، اکتساب، اکتشاف

۱- مقدمه

امروزه بسیاری از مسائل دنیای واقعی به وسیله‌ی تکنیک‌های اتخاذ شده در ماشین‌های هوشمند بررسی و حل می‌شوند. یادگیری تقویتی شاخه‌ای از دانش هوش مصنوعی است که به بررسی حوزه‌ی یادگیری از طریق تعامل و به شیوه آزمون و خطا می‌پردازد. بنابراین اگر عملی منجر به بهبود وضعیت شود تمایل به انجام آن تقویت می‌شود و امکان انجام آن در صورت عدم بهبود تضعیف می‌شود.

برای حل مسائل یادگیری تقویتی سه راه حل کلی برنامه نویسی پویا، روش مونت کارلو و روش تفاضل زمانی مطرح است [۱]. روش تفاضل زمانی از اصلی‌ترین روش‌هایی است که در یادگیری تقویتی مورد استفاده قرار می‌گیرد و ترکیبی از روش‌های برنامه نویسی پویا و مونت کارلو می‌باشد.

یادگیری Q، الگوریتم کنترلی تفاضل زمانی off-policy است که به‌طور همزمان در محیط کاوش انجام می‌دهد و سیاست بهینه را یاد می‌گیرد. این الگوریتم به دلیل آسان بودن پیاده سازی و تئوری خوب توسعه یافته به صورت گسترده‌ای مورد استفاده قرار گرفته است.

در روش یادگیری Q، افزایش تعداد حالت‌های محیط به عنوان چالشی مطرح می‌باشد [۱]. زیرا افزایش تعداد حالت‌ها باعث کاهش سرعت همگرایی و درنتیجه افزایش هزینه در یادگیری عامل می‌شود. بنابراین ارائه روش‌هایی که باعث افزایش سرعت یادگیری عامل می‌شوند، ضروری می‌نماید.

تاکنون روش‌های گوناگونی برای بهبود یادگیری Q ارائه شده است. عمده‌ی این روش‌ها به پنج دسته اصلی تقسیم می‌شوند، که هر کدام تلاش بر بهبود یادگیری باتوجه به محدودیت‌های موجود در یادگیری Q دارند. ساختار کلی این پژوهش‌ها در پنج دسته زیر شرح داده شده است.

- استراتژی به‌روز رسانی مقادیر Q: در پژوهش[۲] الگوریتم Q(λ)، به عنوان یک الگوریتم چندگامه افزایشی پیشنهاد شده است. در پژوهش [۳] برای دستیابی به مسیر بهینه در محیط ناشناخته، تعداد به‌روز رسانی مقادیر Q پس از انجام یک عمل افزایش داده شده است. در این روش، زمانی که یک عمل توسط عامل انجام می‌گیرد زنجیره‌ای از حالت‌ها که در طول فرایند جستجو تشکیل شده و شامل وضعیت جاری تا نقطه شروع می‌باشد، برای به‌روز رسانی درنظر گرفته می‌شود. در پژوهش‌های نظیر [۴, ۵]، یادگیری Q مبتنی بر تضاد معرفی شده است. محور اصلی این پژوهش‌ها افزایش تعداد به‌روز رسانی تابع ارزش می‌باشد، بدین گونه که اگر عامل ارزش عمل مخالف را نیز بداند به جای یک مقدار می‌تواند دو مقدار از تابع ارزش را به‌طور همزمان به‌روز رسانی کند. کاهش زمان اکتشاف و افزایش سرعت همگرایی به عنوان دستاوردهای این روش بیان شده است.
- استراتژی کاهش فضای حالت: یکی از محدودیت‌های استفاده از روش یادگیری تقویتی همان طور که قبلا ذکر شد در کاربردهای واقعی رویارویی با فضای حالت بسیار بزرگ می‌باشد که باعث می‌شود زمان یادگیری طولانی شود و حافظه مورد انتظار برای ذخیره جدول Q زیاد شود. اکثر تحقیقات نظیر [۸-۶] ، در زمینه کاهش حالات یا تجزیه وظایف به زیر وظایف کوچکتر و یا تعمیم تجربه‌های عامل یادگیری انجام شده است.
- استراتژی استفاده از دانش پیشین و مقداردهی اولیه Q: اگرچه استفاده از دانش اولیه باعث افزایش سرعت یادگیری تقویتی گزارش شده است [۹]. اما اگر این دانش حاوی اطلاعات اشتباه باشد دارای اثرات سوء مانند جلوگیری از رسیدن به سیاست بهینه می‌شود که منجر به کم شدن سرعت یادگیری می‌شود [۱۰]. در [۱۰]، روشی را با استفاده از کنترل بر روی دانش پیشین ارائه کرده‌اند که تاثیر بد را سرکوب کند. در این روش با درنظر گرفتن فاکتور فراموشی برای عامل باعث بهبود یادگیری شدند. در[۱۱]، یادگیری Q مبتنی بر شبکه عصبی را پیشنهاد داده‌اند. که در آن برای مقداردهی اولیه مقادیر Q از شبکه عصبی استفاده کرده‌اند. نتایج بیان شده نشان دهنده‌ی بهبود عملکرد الگوریتم با استفاده از مقداردهی اولیه اکتشافی می‌باشد.
- استراتژی شکل‌دهی تابع پاداش: برای تسریع فرایند یادگیری روش‌هایی هم‌چون یادگیری Q بیزی، یادگیری Q کندرو و یادگیری Q نسبی، برای دستیابی به پاداش بیشتر مورد بررسی قرار گرفته است[۱۲،۱۳]. در [۱۴]، برای طراحی تابع تقویت از تابع پاداش پیوسته و تخمین‌زن پیشرفت برای سرعت بخشیدن به یادگیری بهره گرفته شده است.
- استراتژی انتخاب عمل: در[۱۵] پیدا کردن سیاست بهینه در یادگیریQ، به جستجوی یک راه‌حل در مسائل بهینه سازی ترکیبی تبدیل شده است. در این روش از معیار Metropolis الگوریتم شبیه سازی تبرید به منظور مصالحه بین اکتشاف و بهره برداری استفاده شده است. در [۱۶]، اکتشاف برپایه تفاضل ارزش برای تعادل بین اکتشاف و بهره برداری در یادگیری تقویتی درنظر گرفته شده است. در[۱۷]، اکتشاف بر پایه تفاضل ارزش که با انتخاب عمل سافت‌مکس ترکیب شده به عنوان سیاست تطبیقی در یادگیری تفاضل زمانی پیشنهاد شده است.

در این پژوهش، از ترکیب استراتژی‌های به‌روز رسانی مقادیر Q و انتخاب عمل استفاده شده است. با الهام از الگوریتم یادگیری Q مبتنی بر تضاد و تغییر قسمت به‌روز رسانی مقادیر Q یک الگوریتم جدید پیشنهاد می‌شود. که در آن به منظور رسیدن به رویه بهینه از اکتشاف بر پایه تفاضل ارزش ترکیب شده با رویه سافت‌مکس استفاده شده است. مشکل موجود مقادیر Q گمراه کننده که در پژوهش پیشین [۵] وجود داشته است و باعث سردرگمی عامل بوده است در این کار برطرف شده است.

سازماندهی ساختار مقاله به اینصورت است که در بخش۲، ابتدا یادگیریQ به صورت مختصر شرح داده شده است. سپس، مروری بر پژوهش پیشین در زمینه یادگیریQ مبتنی بر تضاد شده است. در بخش۳، روش کار توضیح داده شده است‌که درآن الگوریتم پیشنهادی بیان می‌شود. ارزیابی و نتایج آزمایش‌ها در بخش۴ آورده شده است. در نهایت نتیجه‌گیری در بخش۵ آورده شده است.

۲- یادگیری Q و یادگیری Q مبتنی بر تضاد

۲-۱- یادگیری Q

یادگیری Q اولین بار توسط واتکینز معرفی شد[۱۸]. یادگیری Q تک گامی به این صورت تعریف می‌شود که عامل در هر تکرار یکی از مقادیر Q را برای هر جفت حالت – عمل مطابق رابطه (۱) به‌روز رسانی می‌کند:

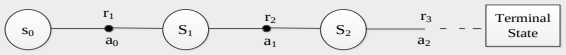
$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \quad (۱)$$

نمادهایی که برای محاسبه یک گام یادگیری Q در فرمول(۱) استفاده شده است به صورت زیر تعریف می‌شوند که این نمادها در [۱] استفاده شده‌اند.

- s= حالت فعلی
- a= عمل
- s'= حالت بعدی
- a'= عمل حالت بعدی
- r= پاداش فوری
- α= پارامتر نرخ یادگیری
- γ= فاکتور تخفیف یا ضریب تنزیل
- Q(s, a) = ارزش حالت - عمل برای حالت s و عمل a

فاکتور گاما میزان آینده‌نگری عامل را بیان می‌کند که نشان دهنده‌ی اثربخشی ارزش پاداش‌های آینده در روند تصمیم‌گیری فعلی است. افزایش مقدار این پارامتر باعث می‌شود که در روند تصمیم‌گیری به پاداش‌های آینده اهمیت بیشتری داده شود و ارزش حالت بعدی تاثیر بیشتری نسبت به ارزش حالت فعلی داشته باشد. بنابراین، زمانیکه انتخاب عمل به‌درستی صورت نگیرد موجب کاهش سرعت همگرایی می‌شود. اگر مقدار این پارامتر کوچک در نظر گرفته شود تاثیر پاداش‌های آینده کمتر شده و همگرایی را تحت تاثیر قرار می‌دهد چون از گسترش پاداش حالت هدف جلوگیری می‌شود. فاکتور آلفا، برای تاثیر خطای اختلاف زمانی در هنگام به‌روز رسانی ارزش فعلی حالت– عمل استفاده می‌شود که سرعت تغییر مقدار Q در به‌روز رسانی را مشخص می‌کند. افزایش فاکتور آلفا باعث می‌شود که بخش بیشتری از مقدار به‌روز رسانی به‌وسیله‌ی ارزش حالت فعلی تقویت شود. انتخاب مقادیر کوچکتر برای آلفا، مزایایی چون کنترل تاثیر منفی یک به‌روز رسانی نادرست را دارا است.

برای پیاده سازی این روش یک جدول Q درنظر گرفته می‌شود که هر خانه جدول به یک جفت حالت– عمل تعلق دارد. یادگیری در این الگوریتم بدین صورت است که در هر دوره عامل در یک حالت تصادفی قرار داده می‌شود و تا رسیدن به حالت پایانی مقادیر جدول برای هر جفت حالت– عمل بر اساس رابطه (۲) به‌روز می‌شود. شکل(۱) این فرایند را نشان می‌دهد.



شکل(۱): یک دوره یادگیری

همان‌طور که در شکل(۱) نشان داده شده است، با این فرض که عامل در حالت s_0 قرار دارد عمل a_0 را طبق سیاست مشتق شده از مقادیر Q انتخاب نموده و پاداش r_1 را از محیط دریافت می‌کند. حالت بعدی محیط s_1 را مشاهده می‌کند و تا رسیدن به حالت پایانی این روند را تکرار می‌کند.

۲-۲- یادگیریQ مبتنی بر تضاد

زمان مورد نیاز برای همگرایی روش Q، متناسب با سایز جدول Q است. با افزایش سایز جدول Q زمان پر کردن آن نیز افزایش می‌یابد. الگوریتم‌های مبتنی بر تضاد دارای این مزیت می‌باشند که با افزایش تعداد به‌روز رسانی مقادیر Q باعث افزایش سرعت یادگیری می‌شوند. زیرا عامل با انجام یک عمل، عمل متضاد متناظر را نیز درنظر می‌گیرد و به‌صورت همزمان می‌تواند دو مقدار از جدول Q را پر کند. برای یافتن درجه تضاد می‌توان از رابطه(۲) استفاده نمود.

$$\breve{\varphi}(a_1|s_i, a_2|s_j) = \eta \times [1 - e^{\left(-\frac{Q(s_i,a_1)-Q(s_j,a_2)}{\max_k(Q(s_i,a_j)-Q(s_j,a_k)}\right)}] \quad (۲)$$

$\breve{\varphi}$ معیاری است که تضاد بین دو عمل a_1 و a_2 را مشخص می‌کند و درجه تضاد نامیده می‌شود. η تشابه حالت نامگذاری شده که بر اساس کلاستر حالت یا طبق رابطه (۳) اندازه‌گیری می‌شود[۵].

$$\eta(s_i, s_j) = 1 - \frac{\sum_k |\, Q(s_i, s_k) - Q(s_j, s_k)\,|}{\sum_k \max\left(Q(s_i, s_k) - Q(s_j, s_k)\right)} \quad (۳)$$

در پژوهش[۵]، سه الگوریتم بر مبنای تضاد مشتق شده از یادگیری Q معرفی شده که عبارتند از OQL1'، OQL2 و OQL3. ایده اصلی این الگوریتم‌ها این است که اگر عامل به ازای یک عمل پاداش دریافت کند، برای عمل متضاد متناظر با آن یک جریمه دریافت می‌کند.

اولین نسخه الگوریتم (OQL1)، بدین صورت است که در هر گام عامل با انجام عملa و دریافت پاداش r، برای عمل متضاد متناظر ã، مجازات r̃ دریافت می‌کند. به‌روز رسانی برای مقادیر حالت – عمل و حالت– عمل متضاد مطابق رابطه (۴) انجام می‌گیرد:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$
$$Q(s,\tilde{a}) \leftarrow Q(s,\tilde{a}) + \alpha[\tilde{r} + \gamma \max_{a''} Q(s'',a'') - Q(s,\tilde{a})] \quad (۴)$$

در دومین نسخه الگوریتم(OQL2)، نرخ یادگیری برای به‌روز رسانی مقادیر ارزش حالت– عمل متضاد به صورت تابع کاهشی مطابق فرمول (۵) به‌روز می‌شود:

$$\tilde{\alpha} = \sqrt{1 - \frac{i}{n_E}} \quad (۵)$$

که در آن i نشان دهنده‌ی تکرار و n_E تعداد اپیزود می‌باشد[۵]. به‌روز رسانی برای مقادیر حالت– عمل انجام می‌گیرد، سپس نرخ یادگیری طبق رابطه (۵) به‌روز می‌شود و مقدار تابع ارزش برای جفت حالت– عمل متضاد نیز به‌روز می‌شود. بنابراین همانند رابطه(۶) داریم:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

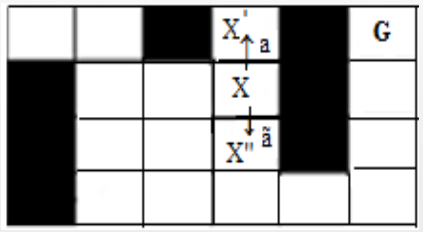
به‌روز رسانی نرخ یادگیری طبق رابطه(۵)

$$Q(s,\tilde{a}) \leftarrow Q(s,\tilde{a}) + \tilde{\alpha}[\tilde{r} + \gamma \max_{a''} Q(s'',a'') - Q(s,\tilde{a})]$$

(۶)

در سومین نسخه الگوریتم(OQL3) ، برای تعداد محدودی از اپیزودها در آغاز یادگیری، مثلا $\frac{۱}{۴}$ تعداد کل اپیزودها به‌روز رسانی اضافی انجام می‌گیرد.

در پژوهش[۵]، با فرض اینکه موقعیت هدف شناخته شده است تابع پاداش این‌گونه تعریف شده که زمانی‌که عامل یک عمل را انجام می‌دهد اگر فاصله اقلیدسی بین عامل و هدف کاهش یابد یا تغییر نکند عامل پاداش ۱+ دریافت می‌کند و زمانی‌که فاصله ذکر شده افزایش یابد پاداش ۱– دریافت می‌نماید. پاداش به طور ضمنی رفتار بهینه را برای عامل توصیف می‌کند به همین دلیل به کارگیری غلط آن می‌تواند باعث گمراه نمودن عامل شود. در شکل(۲) نمایی از یک محیط نشان داده شده است که نشان دهنده‌ی این امر می‌باشد. فرض شده که عامل در موقعیت x قرار دارد و عمل a را انجام می‌دهد و به حالت x' می‌رود چون فاصله تا هدف کاهش یافته بنابراین پاداش ۱+ دریافت می‌کند و همزمان برای عمل مخالف ã مجازات ۱– دریافت می‌کند زیرا فاصله تا هدف برای انتساب پاداش و جریمه برای عامل استفاده شده است. ایجاد مقادیر اشتباه Q باعث گمراهی عامل در انتخاب عمل می‌باشد و منجر به شکست در رسیدن به هدف می‌شود.



شکل(۱): مثالی از مقادیر Q اشتباه. موقعیت هدف با G مشخص شده است. عامل در موقعیت x قرار دارد، عمل a را انجام می‌دهد به حالت x' منتقل می‌شود، پاداش دریافت می‌کند و همزمان برای عمل متضاد ã مجازات می‌شود. بنابراین، مقادیر Q این حالت‌ها اشتباه می‌باشد.

۳- روش کار

۳-۱- به‌روز رسانی مقادیرQ

در روش پیشنهادی مقادیر Q برای هر جفت حالت- عمل مطابق فرمول (۷) به‌روز رسانی می‌شود.

$$Q(s, a) \leftarrow Q(s, a) + \alpha\,[r + \gamma \max_{a'} Q(s', a') + (1-\gamma)\min_{a''} Q(s', a'') - Q(s, a)] \qquad (۷)$$

شرط لازم برای همگرایی مقادیرQها این است که $\max_{a'} Q(s', a') = Q_{t+1}$ و $\min_{a''} Q(s', a'') = Q'_{t+1}$ دارای مقادیر ثابتی باشند یا به عبارت دیگر مقادیرQها با Q_{t+1} و Q'_{t+1} تغییر کند بنابراین داریم:

گام اول: $Q_t = (1-\alpha)\,Q_t + \alpha\,(r + \gamma\, Q_{t+1} + (1-\gamma) Q'_{t+1})$

گام دوم: $Q_t = (1-\alpha)^2\,Q_t + (1-\alpha)\,\alpha(r + \gamma\, Q_{t+1} + (1-\gamma) Q'_{t+1}) + \alpha\,(r + \gamma\, Q_{t+1} + (1-\gamma) Q'_{t+1})$

.

.

.

گام nام: $Q_t = (1-\alpha)^n\,Q_t + (1-\alpha)^{n-1}\,\alpha(r + \gamma\, Q_{t+1} + (1-\gamma) Q'_{t+1}) + (1-\alpha)^{n-2}\,\alpha(r + \gamma\, Q_{t+1} + (1-\gamma) Q'_{t+1}) + \ldots + \alpha(r + \gamma\, Q_{t+1} + (1-\gamma)\, Q'_{t+1})$

$= (1-\alpha)^n\, Q_t + \alpha(r + \gamma\, Q_{t+1} + (1-\gamma) Q'_{t+1}) \times [(1-\alpha)^{n-1} + (1-\alpha)^{n-2} + \ldots + 1]$

$= (1-\alpha)^n\, Q_t + (r + \gamma\, Q_{t+1} + (1-\gamma) Q'_{t+1})(1-(1-\alpha)^n)$

چون

$0 < \alpha < 1 \rightarrow 0 < 1-\alpha < 1$

$(1-\alpha)^n \rightarrow 0,\ Q_t = r + \gamma\, Q_{t+1} + (1-\gamma)\, Q'_{t+1}$

مقدارQ، با به‌روز رسانی کافی همگرا می‌شود.

در روش پیشنهادی عامل مقادیر Q را برای هر عمل و عمل متضاد متناظر با آن به‌روز رسانی می‌کند. عمل متضاد دارای جهتی مخالف جهت عمل اصلی می‌باشد. به‌عنوان مثال هنگامی‌که عمل اصلی دارای جهت رو به بالا است، جهت عمل متضاد متناظر با آن رو به پایین است. روش پیشنهادی نیز بطور همزمان دو مقدار از جدول Q را برای مقادیر حالت- عمل و حالت - عمل متضاد به‌روز رسانی می‌کند. تابع پاداش به صورت ماتریسی از حالت - عمل در نظر گرفته شده است. با فرض این‌که عامل در جهت دلخواه باشد ارزش حالت- عمل متضاد، با دریافت پاداش جفت حالت – عمل متضاد، برای عمل با کمترین ارزش، ضریب بالاتری نسبت به عملی که بیشترین ارزش در حالت بعدی را دارد درنظر می‌گیرد. در روش پیشنهادی دو مقدار Q مطابق (۸) به‌روز رسانی می‌شود.

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r(s, a) + \gamma \max_{a'} Q(s', a') + (1-\gamma) \min_{a''} Q(s', a'') - Q(s, a)]$$
$$Q(s, \tilde{a}) \leftarrow Q(s, \tilde{a}) + \alpha[r(s,\tilde{a}) + \gamma \min_{a'} Q(s', a') + (1-\gamma) \max_{a''} Q(s', a'') - Q(s, \tilde{a})] \qquad (۸)$$

۳-۲- استراتژی انتخاب عمل

رفتار عامل در هر زمان توسط رویه عمل تعریف می‌شود که به بیانی دیگر حالت را به عمل نگاشت می‌کند. اپسیلون- گریدی و سافت‌مکس دو رویه معروف هستند که اغلب مورد استفاده قرار می‌گیرند [۱]. در رویه اپسیلون- گریدی در هر گام زمانی، عمل تصادفی با احتمال ثابت $0 \le \varepsilon \le 1$ و عمل با بالاترین ارزش با احتمال ε-۱ انتخاب می‌شود که این نوع انتخاب عمل به عنوان انتخاب حریصانه شناخته می‌شود. اگرچه رویه اپسیلون- گریدی به دفعات مورد استفاده قرار می‌گیرد، یکی از اشکال‌های این رویه درنظر گرفتن احتمال مساوی برای انتخاب عمل‌های غیر بهینه است. در رویه سافت‌مکس احتمال انتخاب عمل طبق فرمول (۹) درنظر گرفته می‌شود:

$$P(a) = \frac{e^{\frac{Q(s,a)}{\tau}}}{\sum_{a_j \in A(s)} e^{\frac{Q(s,a_j)}{\tau}}} \qquad (۹)$$

که در آن τ ضریب دما می‌باشد که مقدار آن مثبت است [۱].

اکتشاف بر پایه تفاضل ارزش که با انتخاب عمل سافت‌مکس ترکیب می‌شود به‌عنوان یک رویه تطبیقی برای روش‌های یادگیری تفاضل زمانی مطرح شده است که آن را VDBE-Softmax[۲] نامیده‌اند. تفاوت توزیع بولتزمن مقادیر قبل و بعد از یادگیری طبق رابطه (۱۰) محاسبه می‌شود [۱۷].

$$f(s,a,\sigma) = \left| \frac{e^{\frac{Q_t(s,a)}{\sigma}}}{e^{\frac{Q_t(s,a)}{\sigma}} + e^{\frac{Q_{t+1}(s,a)}{\sigma}}} - \frac{e^{\frac{Q_{t+1}(s,a)}{\sigma}}}{e^{\frac{Q_t(s,a)}{\sigma}} + e^{\frac{Q_{t+1}(s,a)}{\sigma}}} \right| = \frac{1 - e^{\frac{-|\alpha.\Delta|}{\sigma}}}{1 + e^{\frac{-|\alpha.\Delta|}{\sigma}}} \qquad (۱۰)$$

که σ ثابت مثبت می‌باشد.

از مزایای این روش این است که عمل‌های اکتشافی در موقعیت‌هایی که مقادیر ارزش در فرایند یادگیری دارای نوسان است و دانش در مورد محیط به قطعیت نرسیده است انتخاب می‌شود. در آغاز فرایند یادگیری انتظار می‌رود عامل بیشتر اکتشاف انجام دهد و زمانی‌که عامل به شناختی از محیط برسد مقدار اکتشاف کاهش یابد. چنین رفتار انطباقی با استفاده از محاسبه احتمال اکتشاف وابسته به حالت بعد از هر گام یادگیری مانند (۱۱) محاسبه می‌شود.

$$\varepsilon_{t+1}(s) = \delta.\, f(s_t, a_t, \sigma) + (1-\delta).\, \varepsilon_t(s) \qquad (۱۱)$$

$\delta(s) = \frac{1}{|A(s)|}$ می‌باشد، که در آن |A(s)| تعداد عمل‌ها می‌باشد. ε (s) برای تمامی حالت‌ها در آغاز با یک مقدار دهی شده است[۱۷].

در روش پیشنهاد شده برای محاسبه احتمال اکتشاف وابسته به حالت، فرمول(۱۲)، میانگینی از $f(s, a, \sigma)$ و $f(s, \tilde{a}, \sigma)$ درنظر گرفته شده است.

$$f(s,a,\sigma) = \frac{1-e^{\frac{-|\alpha.\Delta_1|}{\sigma}}}{1+e^{\frac{-|\alpha.\Delta_1|}{\sigma}}}$$

$$\Delta_1 = r(s,a) + \gamma \max_{a'} Q(s',a') + (1-\gamma)\min_{a''} Q(s',a'') - Q(s,a)$$

$$f(s,\tilde{a},\sigma) = \frac{1-e^{\frac{-|\alpha.\Delta_2|}{\sigma}}}{1+e^{\frac{-|\alpha.\Delta_2|}{\sigma}}}$$

$$\Delta_2 = r(s,\tilde{a}) + \gamma \min_{a'} Q(s',a') + (1-\gamma)\max_{a''} Q(s',a'') - Q(s,\tilde{a})$$

$$\varepsilon_{t+1}(s) = \frac{1}{2}\delta.(f(s,a,\sigma) + f(s,\tilde{a},\sigma)) + (1-\delta).\varepsilon_t(s) \qquad (۱۲)$$

الگوریتم اصلی این مقاله در ادامه با عنوان الگوریتم (۱) ارائه شده است.

الگوریتم(۱): OQL-VDBE

1. Initialize Q(s, a) arbitrarily
2. Initialize e(s) arbitrarily, e.g. $\varepsilon(s) = 1$ for all s
3. Repeat (for each episode):
4. Initialize s
5. Repeat (for each step of episode):
6. $\xi \leftarrow$ rand(0..1)
7. if $\xi < \varepsilon(s)$ then
8. $a \leftarrow$ softmax(A(s))
9. else
10. $a \leftarrow \text{argmax}_{b \in A(s)} Q(s, b)$
11. endif
12. Take action a, observe reward r and next state s'
13. Determine opposite action $\tilde{a}$
14. $a^* \leftarrow \text{argmax}_{i \in A(s')} Q(s', i)$
15. $\tilde{a}^* \leftarrow \text{argmin}_{j \in A(s')} Q(s', j)$
16. $\Delta_1 = r(s,a) + \gamma Q(s', a^*) + (1-\gamma)Q(s', a^*) - Q(s,a)$
17. $\Delta_2 = r(s,\tilde{a}) + \gamma Q(s', \tilde{a}^*) + (1-\gamma)Q(s', a^*) - Q(s,\tilde{a})$
18. $\Delta_3 = r(s,a) + \gamma Q(s', \tilde{a}^*) + (1-\gamma)Q(s', a^*) - Q(s,a)$
19. $\Delta_4 = r(s,\tilde{a}) + \gamma Q(s', a^*) + (1-\gamma)Q(s', \tilde{a}^*) - Q(s,\tilde{a})$
20. if $Q(s, a) < Q(s', a^*)$
21. $Q(s, a) = Q(s, a) + \alpha.\ \Delta_1$
22. $Q(s, \tilde{a}) = Q(s, \tilde{a}) + \alpha.\ \Delta_2$
23. $\varepsilon(s) = \frac{1}{2}.\delta.\left(\frac{1-e^{\frac{-|\alpha.\Delta_1|}{\sigma}}}{1+e^{\frac{-|\alpha.\Delta_1|}{\sigma}}} + \frac{1-e^{\frac{-|\alpha.\Delta_2|}{\sigma}}}{1+e^{\frac{-|\alpha.\Delta_2|}{\sigma}}}\right) + (1-\delta).\varepsilon(s)$
24. else
25. $Q(s, a) = Q(s, a) + \alpha.\ \Delta_3$
26. $Q(s, \tilde{a}) = Q(s, \tilde{a}) + \alpha.\ \Delta_4$
27. $\varepsilon(s) = \frac{1}{2}.\delta.\left(\frac{1-e^{\frac{-|\alpha.\Delta_3|}{\sigma}}}{1+e^{\frac{-|\alpha.\Delta_3|}{\sigma}}} + \frac{1-e^{\frac{-|\alpha.\Delta_4|}{\sigma}}}{1+e^{\frac{-|\alpha.\Delta_4|}{\sigma}}}\right) + (1-\delta).\varepsilon(s)$
28. endif
29. $s \leftarrow s'$
30. until s is terminal state
31. until a desired number of episode terminated

٤- آزمایش‌ها و ارزیابی

به منظور مقایسه روش پیشنهاد شده (OQL-VDBE) ، با روش‌های تضاد قبلی[۵]، روش QL-VDBE-softmax [۱۷] و روش یادگیری Q استاندارد دو Grid world نشان داده شده در شکل(۳) استفاده شده است. انتخاب عمل نیز بوسیله رویه سافت‌مکس انجام شده است.

عامل در هر دوره به‌صورت تصادفی در یکی از خانه‌های سفید رنگ نشان داده شده در شکل(۳) یادگیری را آغاز می‌کند. در هر قدم عامل می‌تواند در یکی از هشت جهت که شامل: شمال، شمال شرق، شرق، جنوب شرق، جنوب، جنوب غرب، غرب و شمال غرب است، باشد. عامل مسیر را برای رسیدن به خانه هدف که با G مشخص شده، پیمایش می‌کند.

هدف از یادگیری این است که عامل بتواند با پرداخت کمترین هزینه به خانه هدف برسد. فرض شده که هر حرکت پاداشی به اندازه ۱- دارد. حرکت‌هایی که باعث برخورد عامل به مانع یا دیوار می‌شود محل عامل را تغییر نمی‌دهد و پاداش ۱۰- را در پی دارد. زمانی‌که عامل به خانه هدف برسد پاداش ۱+ دریافت می‌کند. برای پیاده سازی پژوهش پیشین مبتنی بر تضاد از تابع پاداش ذکر شده در پژوهش [۵] استفاده شده است.

به منظور مقایسه روش‌ها نرخ موفقیت، میانگین درصد حالت‌های بهینه و متوسط تعداد گام‌های عامل برای رسیدن به هدف به عنوان معیارهای اندازه‌گیری درنظر گرفته شده است. نرخ موفقیت مطابق(۱۳) محاسبه می‌شود.

$$\zeta = \frac{n_S}{epoch \times \max\_episode} \qquad (۱۳)$$

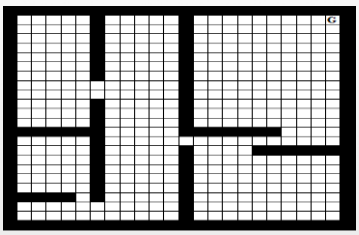
که n_s تعداد دفعاتی است که عامل توانسته به خانه هدف برسد. مخرج تعداد دفعاتی که روش‌ها برای هر محیط اجرا شده است را نشان می‌دهد که شامل تعداد تکرار در تعداد دوره‌ها می‌باشد. هر دوره یادگیری، زمانی‌که عامل به خانه هدف برسد یا به حداکثر تعداد حرکات درنظر گرفته شده برای هر محیط برسد پایان می‌یابد. پارامترهای استفاده شده برای تمامی پیاده سازی‌ها در جدول(۱) آورده شده است. درصد نرخ موفقیت در جدول(۲) نشان داده شده است.

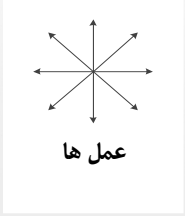
جدول (۱) : مقداردهی پارامترها

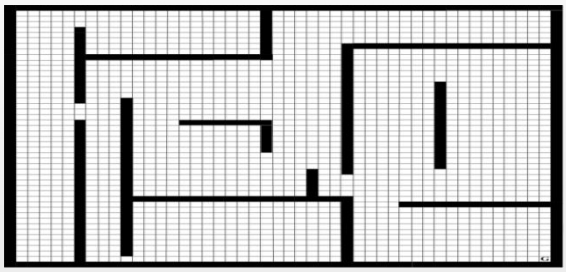
پارامتر	اندازه
حداکثر تعداد گام در محیط (أ)	۳۰۰۰
حداکثر تعداد گام در محیط (ب)	۷۰۰۰
تعداد دوره	۴۰۰
تعداد تکرار	۵۰
دما (τ)	./۱
δ	./۱۲۵
σ	۱۰

جدول(۲): درصد نرخ موفقیت

روش ها	محیط(أ)			محیط(ب)		
	α=۰/۱ γ= ۰/۷	α=۰/۳ γ= ۰/۸	α=۰/۸ γ= ۰/۹	α=۰/۱ γ= ۰/۷	α=۰/۳ γ= ۰/۸	α=۰/۸ γ= ۰/۹
QL	۸۹/۹۵	۹۸/۱۷	۹۹/۴۷	۶۷/۸۲	۸۳/۳۷	۹۶/۳۲
QL_VDBE	۹۶/۶۱	۹۸/۹۳	۹۹/۵۳	۷۹/۸۵	۹۳/۲۶	۹۷/۲۷
OQL1	۲۴/۷۸	۷/۱۷	۱/۳	۱۲/۲۸	۴/۱۵	۱/۲۳
OQL2	۱/۳۶	۰/۸۸	۳/۶۵	۰/۶۷	۱/۵۷	۱/۹۳
OQL3	۲۳/۷۲	۱۶/۲۶	۶/۶۷	۱۰/۴۱	۹/۱	۴/۴۵
OQL-VDBE	۹۸/۶۷	۹۹/۶۲	۹۹/۶۷	۹۴/۶۱	۹۷/۰۷	۹۷/۶۳







(آ) (ب)

شکل(۳): نمایی از دو محیط شبیه‌سازی. موقعیت شروع یکی از خانه‌های سفید رنگ می‌باشد و موقعیت هدف با G مشخص شده است. عامل می‌تواند در هشت جهت حرکت کند. شکل(آ): محیط ۲۴×۲۴ ، شکل(ب): محیط ۴۸×۴۸.

درصد نرخ موفقیت آمده در جدول (۲) نشان می‌دهد که این نرخ برای روش پیشنهادی (OQL-VDBE) دارای مقادیر بیشتری می‌باشد. اگرچه کارهای مبتنی بر تضاد قبلی (OQL1، OQL2 و OQL3)، در محیط‌های بدون مانع به خوبی جواب می‌دهد و باعث افزایش سرعت یادگیری می‌شود، اما در محیط‌هایی که مانع وجود دارد به‌دلیل وجود مقادیر Q اشتباه، منجر به شکست در رسیدن به هدف می‌شود.

درصد حالت‌های بهینه، با درنظر گرفتن نسبت بین، تعداد گام‌های مسیر بهینه به تعداد گام‌های موجود در مسیر هر روش اندازه‌گیری می‌شود [۱۵]. به منظور مقایسه روش‌ها، درصد حالت‌های بهینه و تعداد گام‌های عامل در رسیدن به هدف در هر دوره یادگیری ثبت شده است. پس از پایان یافتن دوره‌ها، فرایند یادگیری برای تمامی روش‌ها ۵۰ مرتبه تکرار شده است. میانگین درصد بهینه و متوسط تعداد گام‌ها گزارش شده است.

ارزیابی روش پیشنهاد شده با پژوهش‌های پیشین در شکل‌های (۹-۴) نشان داده شده است. درصد حالت‌های بهینه در شکل‌های(۴،۶،۸) و متوسط تعداد گام‌های عامل تا رسیدن به هدف در شکل‌های (۵،۷،۹) برای دو محیط نشان داده شده است. در شکل‌های ذکر شده زیرنویس(آ) نتایج به دست آمده برای محیط (آ) را نشان می‌دهد و زیرنویس (ب) متناظر با نتایج به دست آمده در محیط (ب) می‌باشد.

شکل ۴ (آ)، نتایج فرایند یادگیری را با پارامترهای α= ۰/۱، γ=۰/۷ و تعداد دوره ۴۰۰ برای شش روش نشان می‌دهد. میانگین درصد حالت بهینه در روش پیشنهاد شده دارای مقادیر بالاتری نسبت به دیگر روش‌ها می‌باشد. دلیل این بهبود این است که عامل مسیر کوتاهتری را برای رسیدن به هدف پیمایش می‌کند. روش پیشنهاد شده (OQL_VDBE) چون در هر گام یادگیری، بطور همزمان دو مقدار از مقادیر Q را به‌روز می‌کند نسبت به روش QL_VDBE برتری دارد. روش QL_VDBE نیز به‌دلیل بهبود سیاست انتخاب عمل از روش QL بهتر عمل می‌کند. روش‌های مبتنی بر تضاد (OQL1,OQL2,OQL3) به دلیل انتساب پاداش‌های اشتباه، باعث ورود مقادیر اشتباه به جدول Q می‌شوند. بنابراین عامل در بسیاری از موارد نمی‌تواند به خانه هدف برسد، و برای به پایان رساندن فرایند یادگیری در هر دوره، حداکثر تعداد گام‌های درنظر گرفته شده را طی می‌کند.

شکل ۴(ب)، فرایند یادگیری را برای محیط (ب) نشان می‌دهد. همان‌طور که دیده می‌شود درصد حالت بهینه نسبت به محیط (آ) کمتر شده است، چون محیط (ب) دارای تعداد حالت‌های بیشتری نسبت به محیط (آ) می‌باشد.

شکل ۵(آ) و ۵(ب) متوسط تعداد گام‌های عامل تا هدف را برای دو محیط نشان می‌دهد. روش ارائه شده نسبت به روش‌های دیگر بهتر عمل می‌کند.

همان‌گونه که دیده می‌شود روش پیشنهادی با تعداد گام‌های کمتری به خانه هدف می‌رسد. همین بهبود حاصل شده منجر به افزایش سرعت یادگیری می‌شود.

برای کارایی الگوریتم، آزمایش‌ها با پارامترهای α= ۰/۳، γ=۰/۸ و α= ۰/۸، γ=۰/۹ تکرار شده است. نتایج در شکل‌های (۹-۶) آورده شده است. با افزایش این پارامترها، فرایند یادگیری نیز بهبود یافته است.

همان‌طور که در شکل‌های(۴،۶،۸) برای دو محیط مشاهده می‌شود، درصد حالت‌های بهینه در روش پیشنهاد شده در این مقاله به طور میانگین مطابق جدول(۳) نسبت به روش‌های QL و QL_VDBE افزایش یافته است. که دلیل این افزایش همان‌گونه که در بخش۳ اشاره شده است موارد مانند: افزایش تعداد به‌روز رسانی و بهبود استراتژی انتخاب عمل می‌باشد. همان‌طور که قبلا ذکر شد درصد حالت بهینه، نسبت بین تعداد گام‌های کوتاهترین مسیر موجود به تعداد گام‌های موجود در مسیر هر روش می‌باشد. سه روش ذکر شده در پژوهش[۵] در بسیاری از موارد نمی‌توانند به خانه هدف برسند. بنابراین به دلیل عملکرد ضعیف، در جدول(۳) از نشان دادن نتایج آن خودداری شده است. برای مثال در شکل‌های (۴،۶،۸) عملکرد ضعیف این روش‌ها قابل مشاهده است.

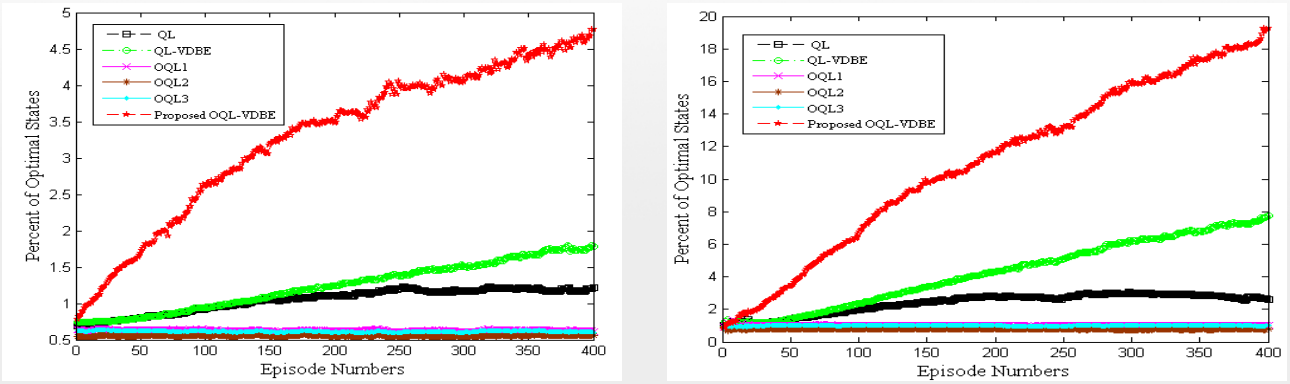
جدول(۳): مقایسه‌ی نرخ بهبود

روش	محیط(آ)			محیط(ب)		
	α=۰/۱ γ= ۰/۷	α=۰/۳ γ= ۰/۸	α=۰/۸ γ= ۰/۹	α=۰/۱ γ= ۰/۷	α=۰/۳ γ= ۰/۸	α=۰/۸ γ= ۰/۹
QL	۹/۲۴	۲۴/۳۸	۴۲/۳۹	۲/۲۳	۷/۴۸	۱۱/۴۷
QL_VDBE	۷/۱۴	۱۶/۵۸	۳۲/۶۱	۲/۰۴	۶/۴۸	۸/۸۲

۵- نتیجه گیری

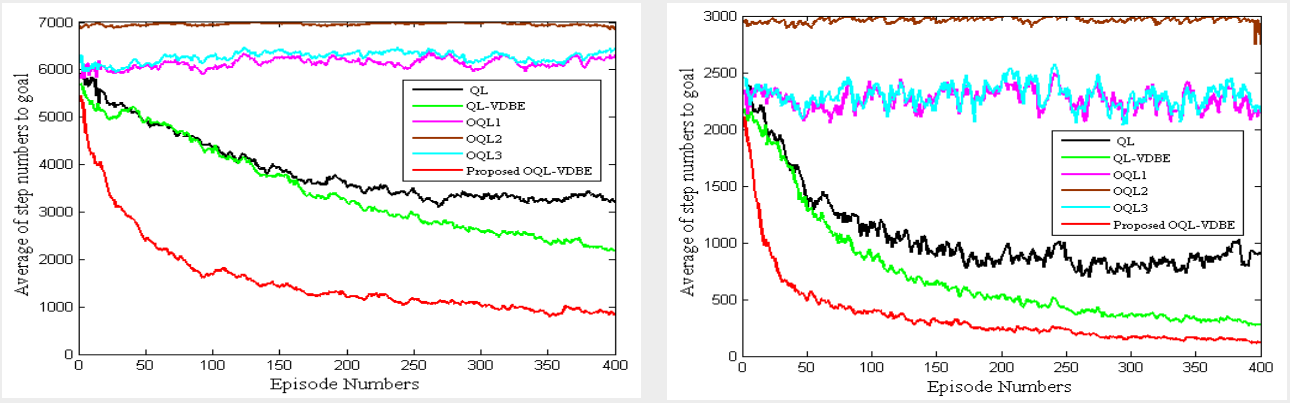
در مقاله ارائه شده، الگوریتمی برای افزایش سرعت یادگیری Q مطرح شده است. در الگوریتم ارائه شده از روشی ترکیبی بر پایه افزایش تعداد به‌روز رسانی مقادیر Q و بهبود انتخاب عمل استفاده شده است.

ارزیابی روش ارائه شده به وسیله مقداردهی گوناگون پارامترهای تاثیرگذار در دو محیط انجام گرفته شد. نتایج به دست آمده حاکی بر بهبود فرایند یادگیری و تسریع در آن می‌باشد. با توجه به اینکه یادگیری تقویتی دارای روش‌های مختلفی می‌باشد و در این مقاله یادگیری Q مد نظر گرفته شده است، الگوریتم ارائه شده می‌تواند در روش‌های دیگر یادگیری تقویتی به عنوان کار آتی مطرح باشد.



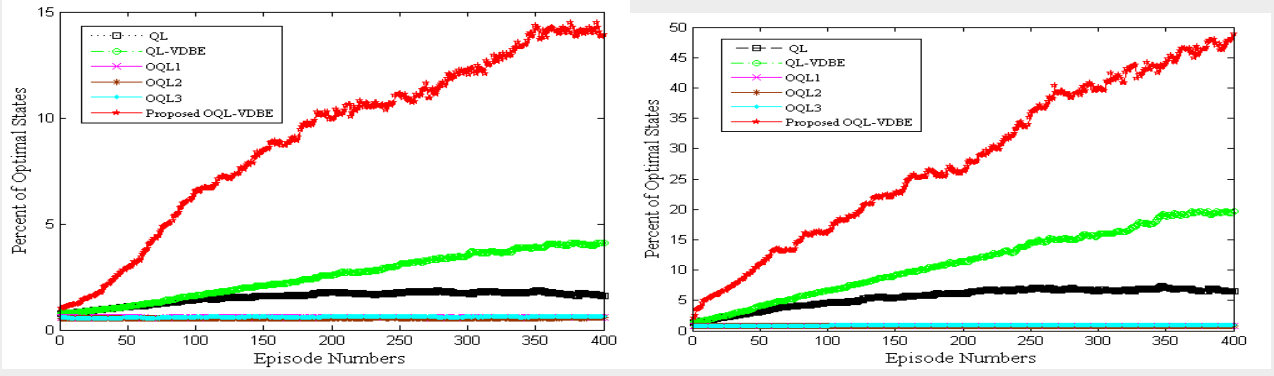
(ب) (أ)

شکل(٤): مقایسه عملکرد یادگیری با α=. /۱ و γ=. /۷. تعداد دوره‌ها به میانگین درصد حالت‌های بهینه در دو محیط (أ) و (ب)



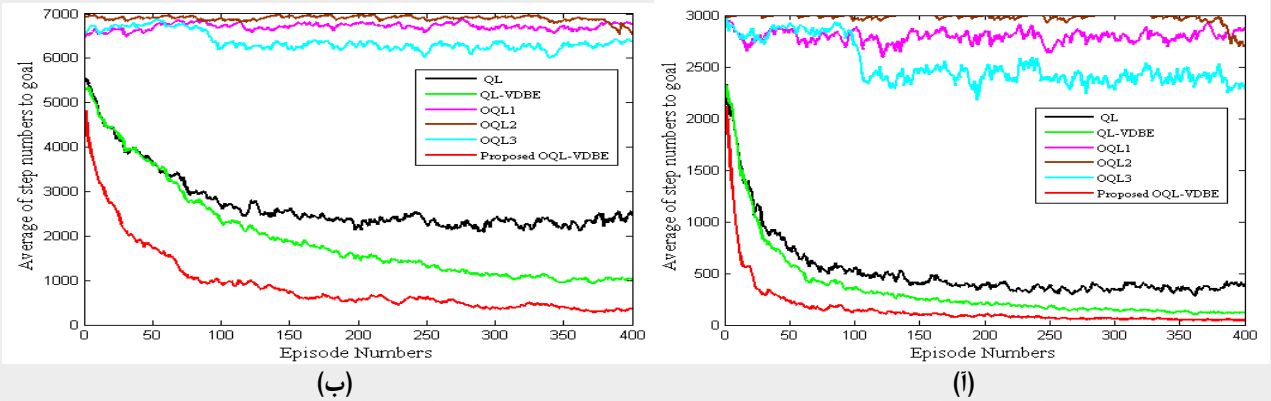
(ب) (أ)

شکل(٥): مقایسه عملکرد یادگیری با α=. /۱ و γ=. /۷. تعداد دوره‌ها به متوسط تعداد گام‌های عامل تا هدف در دو محیط (أ) و (ب)



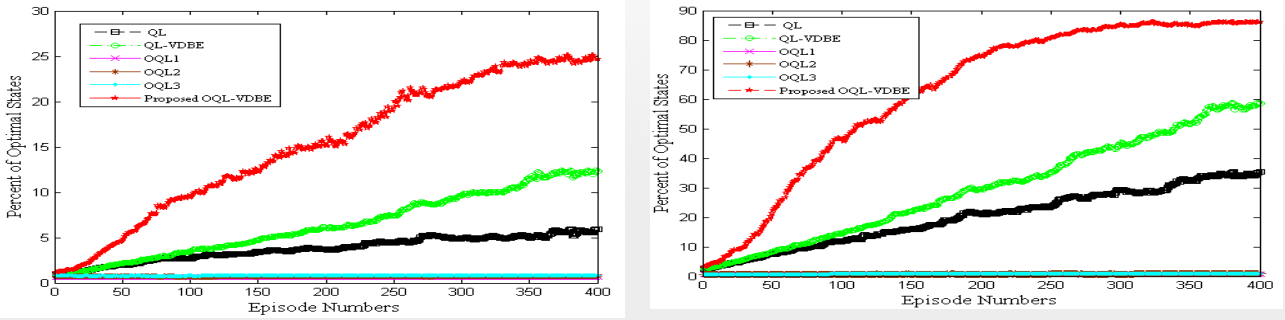
(ب) (أ)

شکل(٦): مقایسه عملکرد یادگیری با α=. /۳ و γ=. /۸. تعداد دوره‌ها به میانگین درصد حالت‌های بهینه در دو محیط (أ) و (ب)



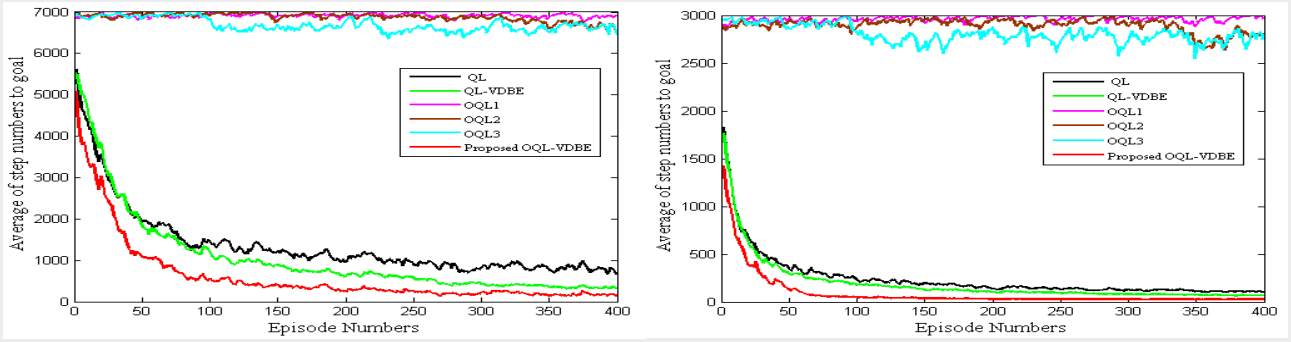
(ب) (أ)

شکل(۷): مقایسه عملکرد یادگیری با α=. /۳ و γ=. /۸. تعداد دوره‌ها به متوسط تعداد گام‌های عامل تا هدف در دو محیط (أ) و (ب)



(ب) (أ)

شکل(۸): مقایسه عملکرد یادگیری با α=۰/۸ و γ=۰/۹. تعداد دوره‌ها به میانگین درصد حالت‌های بهینه در دو محیط (أ) و (ب)



(ب) (أ)

شکل(۹): مقایسه عملکرد یادگیری با α=۰/۸ و γ=۰/۹. تعداد دوره‌ها به متوسط تعداد گام‌های عامل تا هدف در دو محیط (أ) و (ب)

مراجع

[1] Sutton, R.S., Barto A.G., *Reinforcement learning: An Introduction*, MIT Press, Cambridge, MA, 1998.

[2] Peng, J., Williams, R. J. "*Incremental multi-step Q-learning.*" Machine Learning, 22(1-3), 283-290, 1996.

[3] Ma, X., Xu, Y., Sun, G. Q., Deng, L. X., and Li, Y. B. "*State-chain sequential feedback reinforcement learning for path planning of autonomous mobile robots.*" Journal of Zhejiang University Science C, 14(3), 167-178, 2013.

[4] Tizhoosh, H. R. "*Reinforcement learning based on actions and opposite actions.*" In International Conference on Artificial Intelligence and Machine Learning (pp. 94-98), 2005.

[5] Tizhoosh, H. R. "*Opposition-based reinforcement learning*", Journal of Advanced Computational Intelligence and Intelligent Informatics 10 (4), 578–585, 2006.

[6] Senda, K., Mano, S., and Fujii, S. "*A Reinforcement Learning Accelerated by State Space Reduction*". SICE Annual Conf., pp:1992-1997, 2003.

[7] Hamagami, T., and Hirata, H. "*An Adjustment Method of the Number of States of Q-Learning Segmenting State Space Adaptively*". Proc. IEEE Int. Conf. on Systems, Man and Cybernetics, pp:3062-3067, 2003.

[8] Lampton, A., and Valasek, J. "*Multiresolution State-Space Discretization Method for Q-Learning.*" Proc. American Control Conf., p.1646-1651, 2009.

[9] Ribeiro, C. H. "*Embedding a priori knowledge in reinforcement learning*", Journal of Intelligent and Robotic Systems 21, pp:51-71. 1998.

[10] Terashima, K., and Murata, J. "*A study on Use of Prior Information for Acceleration of Reinforcement Learning*", SICE Annual Conf. 2011, pp. 537-543, 2011.

[11] Song, Y., Li, Y. B., Li, C. H., and Zhang, G. F. "*An effcient initialization approach of Q-learning for mobile robots*". International Journal of Control, Automation and Systems, 10:166–172, 2012.

[12] Pandey, P., Pandey, D., and Kumar, S. "*Reinforcement learning by comparing immediate reward*", IJCSIS, vol. 8, no. 5, pp. 1-5, August 2010.

[13] Manju, S., and Punithavalli, M. "*An analysis of Q-learning algorithms with strategies of reward function*", IJCSE, vol. 3, no. 2, pp. 814-820, February 2011.

[14] Mataric, M. J. "*Reward functions for accelerated learning*," Proc. of the International Conference on Machine Learning, pp. 181-189, 1994.

[15] Guo, M., Liu, Y., and Malec, J. "A new Q-learning algorithm based on the metropolis criterion." IEEE Trans. Syst. Man Cybern. B, 34(5):2140-2143, 2004.

[16] Tokic, M. "*Adaptive ε-greedy exploration in reinforcement learning based on value differences*". In: LNCS, vol. 6359, pp. 203–210. Springer, Heidelberg. 2010.

[17] Tokic, M. and Palm, G. "*Value-difference based exploration: Adaptive exploration between epsilon-greedy and softmax.*" In KI 2011: Advances in Artificial Intelligence, 335-346. Springer Berlin / Heidelberg. 2011.

[18] Watkins, C. J. C. H. , *Learning from Delayed Rewards*, PhD thesis, Cambridge University, Cambridge, England, 1989.

زیرنویس‌ها

[1] Opposite Q-learning

[2] Value Difference Based Exploration - Softmax