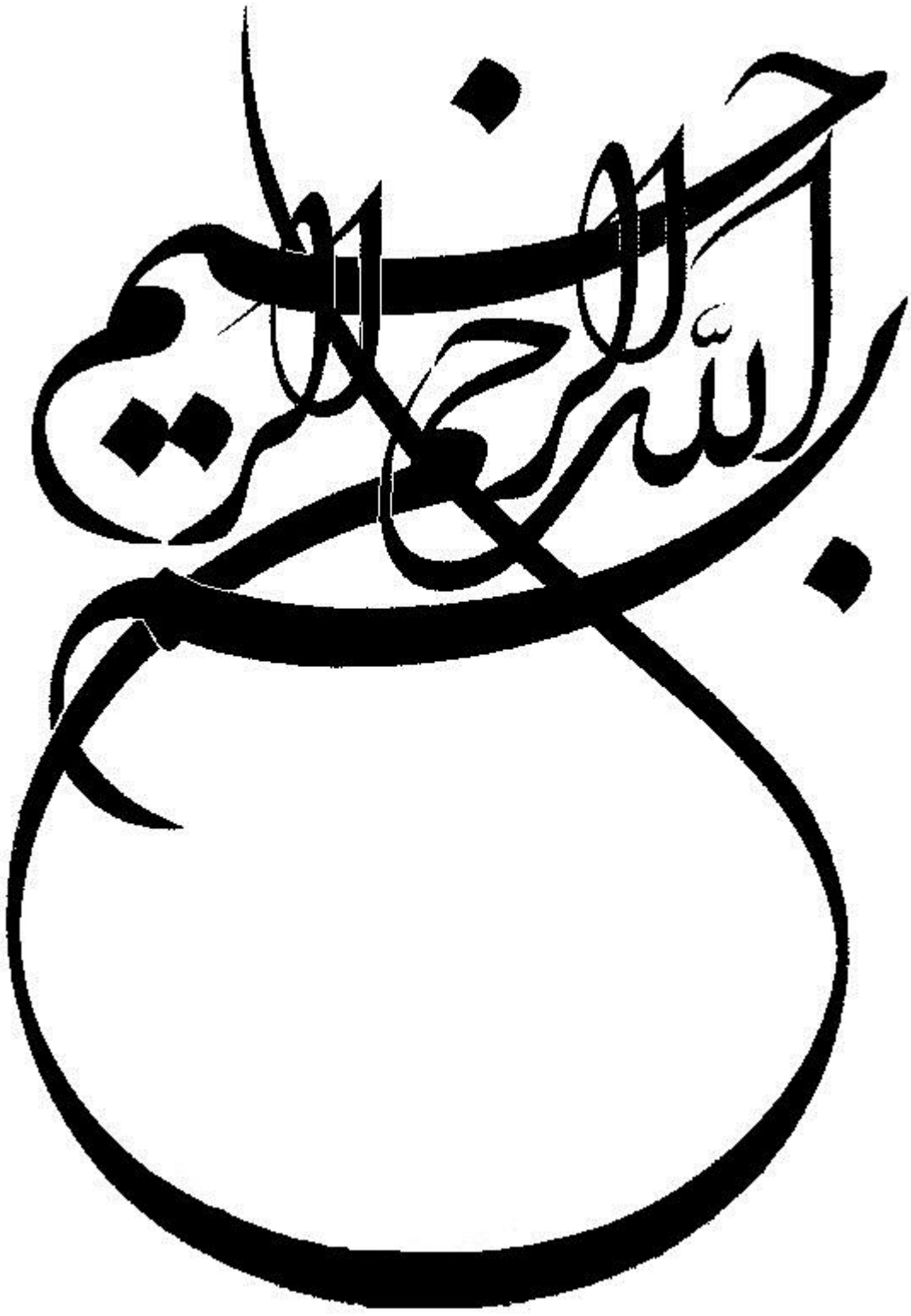

بسم الله الرحمن الرحيم

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

گروه هوش مصنوعی

پایان نامه کارشناسی ارشد

اکتساب مهارت در یادگیری تقویتی رباتیک توسط عاملهای خودمختار

فاطمه تلگردی

استاد راهنما :

دکتر علی اکبر پویان

اساتید مشاور:

مهندس علیرضا خلیلیان

دکتر سعید شیری

زمستان **۱۳۹۳**

تقدیم به...

خدایی که آفرید

جهان را، انسان را، عقل را، علم را، معرفت را، عشق را

و...

پدرم به استواری کوه،

مادرم به زلالی چشمه،

همسرم که نشانه لطف الهی در زندگی من است.

تشکر و قدردانی

حمد و سپاس یکتای بی همتا را که لطفش بر ما عیان است، ادای شکرش را هیچ زبان و دریای فضلش را هیچ کران نیست و اگر در این وادی هستیم، همه محبت اوست. نهال را "باران" باید، تا سیرابش کند از آب حیات و "آفتاب" باید تا بتاباند نیرو را و محکم کند شاخه های تازه روییده را؛ بسی شایسته است از اساتید فرهیخته و فرزانه‌ام:

جناب آقای دکتر پویان، به عنوان استاد راهنما، و جناب آقای دکتر خلیلیان و دکتر شیری، اساتید مشاور بنده که مرا در پیشبرد این پایان نامه صمیمانه و مشفقانه یاری نمودند کمال تشکر را دارم.

با تقدیر و تشکر شایسته از اساتید عالی قدر جناب آقای دکتر حسن پور، جناب آقای دکتر زاهدی و خانم دکتر مشایخی که از محضر پرفیض تدریسشان، بهره ها برده ام.

سپاس آخر را به مهربانترین همراهان زندگیم، به پدر، مادر و همسر عزیزم تقدیم می کنم که حضورشان در فضای زندگیم مصداق بی ریای سخاوت بوده است.

پروردگارا حسن عاقبت، سلامت و سعادت را برای آنان مقدر نما

حال، این برگ سبزی است تحفه درویش تقدیم آنان...

تعهد نامه

اینجانب فاطمه تلگردی دانشجوی دوره کارشناسی ارشد رشته هوش مصنوعی دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی شاهرود نویسنده پایان نامه اکتساب مهارت در یادگیری تقویتی رباتیک توسط عاملهای خودمختار تحت راهنمائی دکتر علی اکبر پویان متعهد می شوم.

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است .
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است .
- مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است .
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید .
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است .
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج ، کتاب ، برنامه های رایانه ای ، نرم افزار ها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می باشد . این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود .

* متن این صفحه نیز باید در ابتدای نسخه های تکثیر شده پایان نامه وجود

چکیده

یادگیری تقویتی یکی از حوزه های یادگیری ماشین است که هدف آن بهبود رفتار عامل بر اساس سیگنال‌های تقویتی است که از محیط دریافت می‌کند. مشکل اینجاست که در بسیاری از کاربردهای واقعی، پاداش محیط با تاخیر بسیار زیادی به عامل داده می‌شود. در نتیجه، عامل برای رسیدن به رفتار بهینه نیازمند صرف زمان بسیار است. راه‌کارهای مختلفی همچون شکل‌دهی پاداش تاکنون برای غلبه بر این مشکل پیشنهاد شده است. اما هیچ‌کدام نتوانستند تأثیر به‌سزایی در افزایش سرعت یادگیری، به‌خصوص در محیط‌های بزرگ و واقعی داشته باشند.

مشکل دیگر این است که تا زمانی که عامل به یک سطح قابل قبول از یادگیری برسد، تمام حرکات آن تصادفی خواهد بود. ضمناً با پیچیده‌تر شدن محیط، تعداد وضعیت‌های مورد اکتشاف و پارامترهای تصمیم‌گیری افزایش پیدا می‌کند. تمامی این مسائل، اکتشاف را رویکردی زمان‌بر، با هزینه بسیار بالا و گاهی بسیار پرخطر کرده است. یک راه‌کار مورد پژوهش محققان در این حوزه، یادگیری کیفی است.

در این پایان نامه، چارچوبی کلی برای یادگیری کیفی ارائه می‌شود و خصوصیات و اجزا آن معرفی می‌گردد. این چارچوب بر اساس یادگیری کیفی و تخمین پاداش ساختگی می باشد تا از فواید هر دو روش استفاده کند. چارچوب پیشنهادی آن‌چنان است که قابل تنظیم و انطباق با الگوریتم‌های مختلف، محیط‌های گسسته و پیوسته، ناوبری و غیر ناوبری باشد. سپس از چارچوب پیشنهادی یک نمونه ساخته شده، و روی محیط های محک ارزیابی گردیده است. آزمایش‌های صورت گرفته، مؤثر بودن چارچوب پیشنهادی را در تسریع رسیدن به سیاست بهینه نشان می‌دهد.

کلمات کلیدی: یادگیری تقویتی ، یادگیری کیو، یادگیری کیفی، تحلیل گراف، انتزاع

فهرست مطالب

عنوان **صفحه**

فهرست شکلها

عنوان **صفحه**

فصل اول: مقدمه

۱-۱ انگیزه تحقیق

در بسیاری از موارد برای حل یک مسئله جهان واقعی، تعیین آنچه برنامه باید انجام دهد دشوار است. اگر کامپیوتر بتواند از طریق سعی و خطا یاد بگیرد، ارزش عملی بسیاری دارد [۱]. در یادگیری تقویتی[1] عامل برای رسیدن به هدف، از طریق سعی و خطا با محیط تعامل می کند. آموزش عامل از طریق بازخوردی[2] که دریافت می کند انجام می شود، بدون آنکه نحوه انجام عمل را بداند. این ویژگی، یادگیری تقویتی را مورد توجه قرار داده است [۲].

زمانی که دینامیک محیط شناخته شده باشد، کسب سیاست بهینه بدون تعامل با محیط ممکن است. اما وقتی چنین اطلاعاتی در دسترس نیست، از طریق اکتشاف[3] می توان به یادگیری محیط پرداخت. انگیزه اصلی برای استفاده از یادگیری تقویتی برای آموزش مهارتهای جدید به عامل، دستیابی به سه توانایی مهم می باشد: ۱) یادگیری کارهایی که حتی معلم نمی تواند بطور فیزیکی نمایش دهد یا نمی تواند مستقیما برنامه نویسی کند. ۲) یادگیری کسب اهداف بهینه سازی مسائل مشکل که هیچ فرمول تحلیلی یا راه حل به شکل بسته شناخته شده ندارد. ۳) یادگیری برای تطبیق یک مهارت با یک نسخه قبلا دیده نشده جدید از کار [۳].

جهان باید به عنوان نیمه مشاهده پذیر[4] دیده شود. در یادگیری ماشین ، ادراک جهان توسط سنسورها فراهم می شود که نمی توانند کل جهان را شامل همه جنبه های آن نمایش دهند. همچنین، جمع آوری همه اطلاعات مرتبط با رفتار سیستم غیر ممکن است [۴]. یادگیری تقویتی وقتی ویژگی های سیستم مربوطه ناشناخته است، و یا توصیف آن مشکل است، یا وقتی محیط عامل عمل کننده نیمه

[1] Reinforcement Learning
[2] Feedback
[3] exploration
[4] Partially observable

مشاهده پذیر یا بطور کامل ناشناخته است، خیلی با ارزش است. بهرحال، بکارگیری الگوریتم های کلاسیک یادگیری تقویتی اغلب پرهزینه، زمان گیر، و گاهی خطرناک است.

در این الگوریتم ها، نگهداری و بروزرسانی پاداشهای دریافتی محیط در یک جدول به نام جدول کیو[1] انجام می شود. یکی از مشکلات اصلی این الگوریتم ها این است که با بزرگ شدن محیط، نیاز به حافظه زیاد برای نگهداری جداول دارد. همچنین، نیاز به زمان بسیار زیادی برای بروزرسانی مقادیر جدول کیو و رسیدن به مقادیر بهینه وجود دارد. مشکل دیگر این دسته از الگوریتم ها، همگرایی دیرهنگام به رفتار بهینه به خاطر پاداش های تاخیری می باشد [۵]. در حال حاضر، چندین روش در ادبیات برای غلبه بر مسائل مطرح شده وجود دارد.

روش شکل دهی پاداش[2][۶]، یک راه حل موثر برای مشکل پاداش های تاخیری[3] می باشد. از این روش، به عنوان یک فاکتور تسریع در فرآیند یادگیری و نرخ همگرایی استفاده می شود. یکی از مشکلات این روش این است که تعیین مقادیر پاداش ساختگی عملاً برای محیطهای واقعی و بزرگ مشکل است. برای مواجهه با این مساله، یک روش مناسب استفاده از روش های هیوریستیک خودکار مانند آنالیز گراف است ([۷]) تا مقادیر مناسب تخمین زده شود و مقداری پاداش برای عامل فراهم شود. این تخمین ممکن است مشکلاتی مانند پیچیدگی محاسباتی بالا و محاسبات نادرست و گمراه کننده داشته باشد. بهرحال، تخمین تابع پاداش ساختگی از طریق برخی تحلیل های پیش از اکتشاف [۸] نشان داده است که روشی موثر و امیدوارکننده می باشد. مخصوصا، اگر طراح تحلیل های کم هزینه و هیوریستیک های هوشمند را روی محیط استفاده کند.

[1] Q-Table

[2] Reward shaping

[3] Delayed Reward

همچنین، برای محیط های بزرگ و پیچیده می توان از روش های مبتنی بر تقریب تابع ارزش[1] [۹] استفاده کرد. در این روش ها، از یک تقریب زننده مانند شبکه عصبی استفاده می شود. یکی از مشکلات این روشها این است که شبکه های عصبی جزو روشهای یادگیری با ناظر هستند. در نتیجه، برای بهبود رفتار عامل، نیاز به مقدار واقعی و بهینه برای هر حالت -کنش[2] وجود دارد. روش دیگری که در این رابطه قابل استفاده است، روش های تجزیه مسئله می باشد. در رویکرد تجزیه مسئله، شکستن مسائل بزرگ به مسائل کوچکتر باعث کاهش فضای حالت می شود. عامل در هر زیر مسئله با مجموعه کوچکتری از حالات روبرو خواهد شد. یادگیری هر زیر مسئله، بسیار ساده تر و سریعتر از مسئله بزرگتر خواهد بود. همچنین، هر زیر مسئله می تواند در مسائل دیگر هم به کار گرفته شود.

تجزیه مسائل و کاهش فضای حالت می تواند به شیوه های مختلفی انجام گردد. یکی از این روشها، انتزاع زمانی[3] [۱۰] است. در انتزاع زمانی، گروهی از اعمال ترتیبی و پشت سرهم به عنوان یک عمل مجرد (فراکنش)[4] در نظر گرفته می شوند. به جای انتخاب اقدام در هر نقطه در زمان، تصمیم گیری ها با تکرار کمتر انجام می شود [۴]. تعیین این فراکنشها و بازنمایی آنها به دو شیوه دستی و خودکار قابل انجام است. تعیین دستی مهارت ها، توسط طراح انجام می پذیرد که نیازمند شناخت طراح نسبت به جزییات محیط است. در نتیجه، برای محیطهای پیچیده و ناشناخته کاری غیرممکن خواهد بود. لذا کسب خودکار این فراکنش ها [۱۱]، یکی از مسائل اساسی در یادگیری تقویتی سلسله مراتبی است. روش متداول برای بازنمایی فراکنش ها در یادگیری تقویتی، استفاده از چارچوبهای سلسله مراتبی [۱۲] است. استفاده از این چارچوب ها، در محیط های در مقیاس بزرگ هنوز چالش برانگیز است. یک اشکال اساسی این روش ها نیاز به مشاهده کامل فضای حالت است. چالش دیگر، ایجاد خودکار و انطباق سلسله مراتب ها می باشد.

[1] Value Function Approximation
[2] State-Acion
[3] temporal abstraction
[4] Macro-Action

یکی دیگر از راه حلهای کاهش فضای مسئله، انتزاع حالات[1] [۱۳] است. در انتزاع حالات، مجموعه ای از حالتهای مشابه با هم در مسئله به یک حالت مجرد بازنمایی می شوند. در نتیجه، فضای حالت انتزاعی را بوجود می آورد. اندازه فضای حالت به نحو موثری کاهش یافته است، بطوریکه چندین حالت در فضای حالت اصلی دیگر قابل تشخیص نیستند. همچنین در محیط های بزرگ و پیچیده، تعمیم[2] و استفاده مجدد دانش کسب شده اهمیت زیادی دارد. تعمیم اجازه می دهد تا از تجربه های کسب شده استفاده کنیم و دانش یاد گرفته را در شرایط کمی متغیر بکار ببریم. این امر، منجر به تسریع فرآیند یادگیری می شود [۴].

یک روش جایگزین برای مشکلات پاداش ساختگی، استفاده از یادگیری کیفی [۱۴] است. این روش به دنبال تعریف ویژگی های کیفی یک محیط است و آنها را به پارامترهای فرآیند تصمیم گیری مارکوف تبدیل می کند. این باعث استفاده از دانش اولیه ای برای مشخص کردن مقادیر کیفی برای کاوش حالات می شود. برای چنین توصیف فضای حالتی، پارادایم انتزاع کیفی استفاده می شود. نمایش انتزاعی کیفی، جزئیات متریک سنسورها را نادیده می گیرد و کلاس هایی از مفاهیم انتزاعی را می سازد.

وقتی دانش دقیقی از محیط وجود ندارد فراهم کردن دانش مبهم و نادقیقی از احتمالات گذر، می تواند منجر به کاهش مقدار کاوش شود. هدف یادگیری کیفی، پیدا کردن سیاستهای بهینه برای مسائلی است که بطور کیفی توصیف شده اند، یا وقتی یک راه حل برای مسئله وجود ندارد [۱۴]. همچنین، یادگیری کیفی[3][۵]، امکان تعمیم و استفاده مجدد از دانش را فراهم می کند. انتزاع حالات، حتی اگر به راه حل های کمتر از حد مطلوب منجر شود، برای حل وظایف پیچیده بسیار پذیرفته شده اند. بکارگیری انتزاع کیفی، باعث بهبود سرعت یادگیری می شود که مطمئنا توجیه قابل قبولی برای استفاده اش می

[1] spatial abstraction
[2] generalization
[3] Qualitative Reinforcement Learning

باشد. در این تحقیق، برای مواجهه با مباحث مهمی مانند فضاهای حالت پیچیده، تعمیم و استفاده مجـدد از دانش، یادگیری تقویتی کیفی به عنوان راهکاری مناسب بکار گرفته شده است. به منظور بهره گیری از فواید هر دو روش، تخمین پاداش ساختگی و یادگیری کیفی، در این تحقیق چارچوب کلی جدیدی بـرای یادگیری تقویتی کیفی پیشنهاد می شود.

۱-۲ هدف

در بخش قبل راهکارهای مختلفی برای مقابله با مشکلات و چالش های مطرح در یادگیری تقویتی مطرح گردید. یادگیری تقویتی کیفی، به عنوان راهکارمناسـبی بـرای مواجهـه بـا مباحـث مهمـی ماننـد فضاهای حالت پیچیده، تعمیم و استفاده مجـدد از دانـش معرفـی گردیـد. هـدف ایـن تحقیـق، طراحـی الگوریتمی است که از انتزاع حالات برای کاهش فضای حالت استفاده می کند، تا در محـیط هـای بـزرگ قابل استفاده باشد. ادغام حالات، باید به نحوی باشد که ویژگی های محیط را حفظ کند.

برای ادغام حالات محیط، از فرآیند انتزاع استفاده می شود. انتزاع، نمایش زمینه را به یک نمـایش انتزاعی فشرده تر نگاشت می کند. مسئله جدید، یک مسئله یادگیری با فضای حالت کاهش یافتـه اسـت که می تواند با روش های معمول حل شود. به جای کار کردن در فضـای حالـت زمینـه، تصـمیم گیرنـده معمولا راه حل هایی را در فضای حالت انتزاعی خیلی سریعتر پیدا می کند. این منجر به یک محیط کیفی جدید با حالات کمتر و دانش اولیه فراهم شده برای حالات انتزاعی جدید می شود. بدین ترتیب، منجر به تسریع فرآیند یادگیری و پیدا کردن سیاست بهینه توسط عامل می گردد.

همچنین، به منظور بهره گیری از فواید روش پاداش ساختگی سـعی داریـم تـا یـادگیری کیفـی و پاداش ساختگی را با هم ترکیب کنیم تـا بهبـود بیشـتری را در نتـایج داشـته باشـیم. در ایـن رابطـه، از

الگوریتمهای تحلیل گراف برای ساخت خودکار پاداش ساختگی [۸] استفاده می شود. در این راستا، در این پژوهش چارچوب کلی جدیدی برای یادگیری تقویتی کیفی ارائه می‌شود و خصوصیات و اجزا آن معرفی می‌گردد. سپس برای عینیت بخشیدن به این چارچوب، نمونه‌ای از آن ساخته می‌شود و روی محیط های محک ناوبری[1] و غیرناوبری با تعداد حالات مختلف ارزیابی می‌گردد. هدف از این چارچوب پیشنهادی، اکتشاف هوشمندانه‌تر محیط، کاهش زمان مورد نیاز جهت اکتشاف محیط و یادگیری محیط با سرعت بیشتر است.

۱-۳ راهکار ارائه شده

در این تحقیق، چارچوب جدیدی بر اساس یادگیری کیفی ارائه شده است که شامل بخش های مختلفی است. هر بخش از این چارچوب را می توان بر حسب نیازهای مسئله دلخواه انطباق داد. می توان راهکارهای مختلف پیشنهادی را در بخش های مختلف آن بکار برد. از این طریق، می توان از روی چارچوب انتزاعی الگوریتم‌های متعددی ساخت که هسته عملکردی همگی یکسان است. یک نمونه الگوریتم از روی چارچوب پیشنهادی با روش یادگیری کیو ساخته شده است [۵] که کاربردپذیری چارچوب پیشنهادی را نشان دهد.

در الگوریتم نمونه، با فرض یک محیط برای عامل خودمختار، (۱) محیط عامل به یک گراف نگاشت می شود. (۲) با استفاده از الگوریتم های تحلیل گراف، پاداش ساختگی ایجاد می شود و به عنوان دانش اولیه به عامل تزریق می شود. (۳) عامل برای تعدادی اپیزود محدود شروع به کاوش محیط با استفاده از یک روش مانند یادگیری کیو می کند، (۴) حالات بر اساس موقعیت زیرمحیط های مشابه دسته بندی

[1] navigational environments

می شوند، (۵) هر دسته مجددا بر اساس اقدامات قانونی و غیرقانونی دسته بندی می شود، (۶) علاوه براین، دسته های حاصل بر اساس شباهت Q-valueها دسته بندی می شوند، که کلاسترهایی با Q-valueهای مشابه را منجر می شود، (۷) حالات هر کلاستر به یک حالت انتزاعی مجرد ادغام می شوند که به ما یک محیط کوچکتر جدید را می دهد، و (۸) یادگیری در محیط جدید ادامه می یابد تا یک سیاست بهینه حاصل شود.

در این الگوریتم، محیط به یک گراف نگاشت می شود و بر اساس تحلیل این ساختار داده عمل می شود. نگاشت فضای حالت عامل به یک گراف سودمند است. در این صورت، می‌توان از طیف گسترده الگوریتم‌های گراف‌کاوی[1] و نظریه گراف‌ها[2] در ادبیات برای دستیابی به مقداری دانش اولیه از محیط استفاده کرد. در نتیجه، به تسریع کاوش محیط و شتاب فرآیند یادگیری کمک می کند. این اطلاعات، می تواند مستقل از دانش متخصص حوزه یا کاوش محیط توسط عامل بدست آید.

در این تحقیق، از تحلیل گراف برای تخمین تابع پاداش استفاده می شود. از این طریق، مقداری دانش اولیه از فضای حالت برای عامل خودمختار فراهم می شود. تاثیر نهایی آن در تسریع کاوش محیط است که باعث سرعت در فرآیند یادگیری و پیدا کردن یک سیاست بهینه می‌گردد. در روش پیشنهادی، فرض می شود که محیط پویا است، یعنی، حالت هدف و تعداد حالات در محیط تغییر خواهد کرد. دانش قرار گرفته در Q، بر اساس یادگیری قبلی عامل از محیط به‌دست آمده است. توجه کنید که، نمایش گراف فضای حالت به نمایش بهتر در محیط‌های پویا که در آنها مکان هدف و تعداد حالات در طی زمان تغییر می‌کند، کمک می‌کند.

بطور خلاصه، دستاوردهای این پایان نامه به صورت زیر است:

[1] graph mining
[2] graph theoretic

- یک چارچوب انتزاعی جدید برای یادگیری تقویتی کیفی، با عناصر مختلف که می تواند با توجه به نیازمندی های خاص هر مسئله تطبیق پیدا کند.
- یک الگوریتم نمونه از چارچوب پیشنهادی با استفاده از روش یادگیری کیو و تحلیل گراف.
- نتایج و تحلیل ارزیابی چارچوب جدید و یکی از الگوریتم های آن روی محیطهای محک شش اتاقه، برج هانوی و Complex Maze.

چارچوب ارائه شده و الگوریتم نمونه آن، روی محیط های محک استاندارد ارزیابی شده است. برای تعیین اینکه آیا بهبود بدست آمده توسط الگوریتم پیشنهادی تصادفی نیست و از نظر آماری معنی دار است، از روش های آزمون آماری استفاده شده است. استفاده از آزمون های آماری، صحت نتایج را تایید کرده است.

فصل دوم: مفاهیم و پیشینه تحقیق

۲-۱ مقدمه

در این فصل، در ابتدا مروری بر مفاهیم اساسی در یادگیری تقویتی خواهیم داشت. ویژگی مارکوف، فرآیندهای تصمیم گیری مارکوف و روشهای ارائه شده برای حل این فرآیند ها بررسی شده است. سپس جزییات مربوط به یادگیری تقویتی، مساله کاوش و نمونه ای از الگوریتمهای کلاسیک یادگیری تقویتی به نام یادگیری کیو تشریح خواهند شد . در ادامه، چالش های مطرح در یادگیری تقویتی در محیطهای بزرگ و پیچیده و مسئله تعمیم مورد بررسی قرار می گیرد. در ادامه، انتزاع زمانی و انتزاع حالات به عنوان رویکردهایی برای حل مشکلات الگوریتم های کلاسیک یادگیری تقویتی در محیط های بزرگ و پیچیده ارائه می گردد. سپس، شکل دهی پاداش بطور مختصر معرفی خواهد شد. همچنین، یادگیری تقویتی کیفی به تفصیل مورد بررسی قرار خواهد گرفت. در نهایت، راهکارهای ارائه شده در ادبیات برای انتزاع و یادگیری تقویتی کیفی مورد بررسی قرار خواهد گرفت و مزایا و معایب این راهکارها تحلیل و بررسی خواهد شد.

۲-۲ ویژگی مارکوف[1]

در چارچوب یادگیری تقویتی، عامل بر اساس سیگنال حالت که از محیط دریافت می کند تصمیم گیری می کند. لزومی ندارد نمایش حالات به ادراکات فوری محدود شود، بلکه می توانند نسخه های پردازش شده ای از سیگنال حالت دریافتی از محیط باشند، یا ساختارهایی که شامل دنباله ای از ادراکات هستند. همچنین، این سیگنال ممکن است همه آنچه در محیط وجود دارد را نمایش ندهد. در این شرایط، اطلاعات حالت مخفی وجود دارد که اگر عامل آنها را بداند مفید است. بهرحال عامل چون هیچ

[1] The Markov Property

ادراک مرتبطی در آن زمینه دریافت نمی کند، نمی تواند آنها را بداند. مطلوب این است که سیگنال حالت شامل همه ادراکات گذشته بطور فشرده باشد و همه اطلاعات مرتبط را حفظ کند. سیگنال حالتی که همه اطلاعات مرتبط را شامل می شود مارکوف یا دارای ویژگی مارکوف نامیده می شود [۲].

پاسخ محیط به انتخاب یک اقدام، ممکن است به هر چیزی که قبل از آن اتفاق افتاده است بستگی داشته باشد. بنابراین، دینامیک با مشخص کردن توزیع احتمال کامل به صورت زیر تعریف می شود:

$$\Pr\{ s_{t+1} = s', r_{t+1} = r \mid s_t, a_t, r_t, s_{t-1}, a_{t-1}, \ldots, r_1, s_0, a_0\}, \quad \textbf{(۱-۲)}$$

برای s'، r و همه مقادیر ممکن رویدادهای گذشته : $s_t, a_t, r_t, \ldots, r_1, s_0, a_0$. در مقابل، اگر سیگنال حالت ویژگی مارکوف داشته باشد، پاسخ محیط در $t+1$ تنها بستگی به حالت و اقدام در t دارد. در این صورت دینامیک محیط به صورت زیر تعریف می شود:

$$\Pr\{s_{t+1}= s', r_{t+1}= r \mid s_t,a_t\} \quad \textbf{(۲-۲)}$$

برای s'، r، s_t و a_t. به عبارت دیگر، یک سیگنال حالت ویژگی مارکوف دارد، اگر و تنها اگر (۲-۱) برابر با (۲-۲) برای همه s'، r و تاریخچه $s_t, a_t, r_t, \ldots, r_1, s_0, a_0$ باشد. همچنین، محیط و وظیفه مربوطه به عنوان یک کل ویژگی مارکوف را دارند. دینامیک یک مرحله ای محیط های دارای ویژگی مارکوف، پیش بینی حالت و پاداش بعدی را ممکن می سازد. همچنین، با تکرار معادله (۲-۲) می توان همه حالات بعدی و پاداش های مورد انتظار را از حالت جاری پیش بینی کرد. همچنین، می توان تاریخچه کاملی را تا زمان جاری بدست آورد. بهترین سیاست برای انتخاب اقدامات به عنوان تابعی از یک حالت مارکوف، به خوبی بهترین سیاست حاصل از تاریخچه کامل آنهاست. در نتیجه، حالات مارکوف بهترین پایه را برای انتخاب اقدامات فراهم می کنند. همچنین در نظر گرفتن حالات غیرمارکوفی، به صورت تقریبی از حالات

مارکوف کارایی بهتری را خواهد داشت. با در نظر گرفتن ویژگی مارکوف در یادگیری تقویتی، فرض می شود که تصمیمات و مقادیر تنها تابعی از حالت جاری هستند.

۲-۳ فرآیندهای تصمیم گیری مارکوف

یک وظیفه یادگیری تقویتی که ویژگی مارکوف را برآورده می کند، فرآیند تصمیم گیری مارکوف[1] نامیده می شود. یک فرآیند تصمیم گیری مارکوف به صورت چندگانه <S, A, T, R> تعریف می شود. بطور دقیق آن شامل موارد زیر است:

- مجموعه ای از حالات S
- مجموعه ای از اقدامات A
- $T : S \times A \times S \rightarrow [0, 1]$، تابع احتمال گذر[2]. T(s,a,s') احتمال گذر به حالت s' را در حالی که عامل در حالت $s \in S$ است و اقدام $a \in A$ را انجام می دهد مشخص می کند.
- $R : S \times A \rightarrow R$، تابع پاداشی[3] است که پاداش فوری R(s,a) را برای اقدام $a \in A$ در حالت $s \in S$ نسبت می دهد.

حالات بعدی در یک MDP توسط توزیع احتمال مفروض T توصیف می شوند. بنابراین آنها تنها به حالت جاری و اقدام اجرا شده بستگی دارند [۱۵] و دارای ویژگی مارکوف هستند. فرآیند تصمیم گیری مارکوف بدین صورت عمل می کند که در هر مرحله، عامل با مشاهده حالت جاری (s)، اقدام a را از مجموعه اقدامات A_s برای اجرا انتخاب می کند. عامل در هر حالت از فضای حالت خود، مجموعه ای از اعمال را می تواند انجام دهد که این مجموعه به صورت A_s نشان داده می شود. با انجام اقدام a، از طرف محیط به عامل پاداش r(s,a) داده می شود و با احتمال p(s'| s,a) به حالت بعدی s' می رود. به روشی که

[1] MDP
[2] Transition Probability Function
[3] Reward

رفتار عامل را در فرآیند تصمیم گیری مارکوف مشخص می کند، سیاست عامل گفته می شود و با π نمایش داده می شود.

$$\pi : S \times A_s \rightarrow [0,1] \quad \textbf{(۳-۲)}$$

که $\pi(s,a)$ احتمال انجام اقدام a در حالی که در حالت s قرار دارد را بیان می کند. هدف اصلی در فرآیند تصمیم گیری مارکوف، پیدا کردن سیاست بهینه π^* است. سیاست بهینه مطلوب π^*، سیاستی است که پاداش های کسب شده در طی زمان را ماکزیمم می کند [۱۶].

مسائل یادگیری تقویتی را در دو مرحله به ترتیب دنبال می کنیم. اولین مرحله، مدل کردن مساله به صورت یکی از انواع فرآیندهای تصمیم گیری مارکوف است. دومین مرحله، پیدا کردن یک سیاست بهینه با استفاده از خاصیت فرآیندهای تصمیم گیری مارکوف است. اگر فضاهای حالت و اقدام متناهی باشند، بنابراین فرآیند تصمیم گیری مارکوف متناهی[1] نامیده می شود. یک MDP متناهی توسط مجموعه های اقدام و حالتش و توسط دینامیک محیط یک مرحله ای آن تعریف می شود. با فرض هر حالت و اقدام s و a ، احتمال حالت بعدی ممکن به صورت زیر است:

$$P^a_{ss'} = \Pr\{s_{t+1} = s' \mid s_t = s, a_t = a\}. \quad \textbf{(۴-۲)}$$

این کمیت ها احتمالات گذر[2] نامیده می شوند. بطور مشابه، با فرض هر حالت و اقدام جاری s و a، و حالت بعدی s'، ارزش مورد انتظار پاداش بعدی به صورت زیر است:

$$R^a_{ss'} = E\{r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'\}. \quad \textbf{(۵-۲)}$$

[1] Finite MDP
[2] Transition Probability

این کمیت ها، $P_{ss'}^{a}$، و $R_{ss'}^{a}$، مهمترین جنبه های دینامیک یک MDP متناهی را مشخص مـی کننـد [۲]. در چارچوب پیشنهادی در این پایان نامه، فرض می شود که محیط یک MDP متناهی است. عامل باید به هدفش، با انتخاب دنباله ای از اقدامات برسد. هر اقدام نه تنها بازخورد جاری که عامل دریافـت مـی کنـد تاثیر می گذارد، بلکه حالت جدید محیط را نیز تغییر می دهد.

برای پیدا کردن یک راه حل برای MDP، یعنی سیاست بهینه π^{*}، مساله کلیدی پیش بینی پاداش های آینده دریافتی است. همچنین، چه مقدار از پاداش های آینده در نظر گرفته شوند و چطور این مساله اداره می شود. یادگیری تقویتی در چرخه های تکراری که ما آنها را اپیزود[۱] می نامیم اتفاق می افتد. یک اپیزود موقعی خاتمه می یابد که عامل به حالت پایانی برسد، و یا تعـداد مشخصـی از مراحـل H، بـه نـام کران[۲] اجرا شود. اگر تعداد مراحل تا هدف N در نظر گرفته شود، مجموع پاداش هـایی کـه بـا پیـروی از سیاست بهینه بدست می آید به صورت $\sum_{t=0}^{N-1} r(s,a)$، ماکزیمم می شود. به ایـن روش، پـاداش بـدون نرخ تنزیل[۳] گفته می شود.

مشکل این روش این است که اگر تعداد مراحل نامتناهی باشد، مجموع پاداش هایی که بدست می آید نامتناهی خواهد شد. برای رفع این مشکل، روش نرخ تنزیل یا میانگین پاداش[۴] بکار برده می شود. در این روش، هدف ماکزیمم کردن مجموع پاداش با وزن نرخ تنزیل است. معمول ترین مدل مـورد اسـتفاده در این حالت، مدل بازه نامتناهی تنزیلی[۵] می باشد. در این مدل، همه پاداش های بعدی جمع می شـوند، اما توسط یک فاکتور تنزیل $\gamma \in R\ (0 < \gamma < 1)$ وزن داده می شوند. در نتیجه، پـاداش هـای بعـدی تاثیر کمتری روی نتیجه کلی دارند[۱۵]. تابع ارزش در مدل بازه نامتناهی تنزیلی به صورت زیـر تعریـف می شود:

[1] Episode
[2] horizon
[3] Undiscounted Factor
[4] Average Reward
[5] Discounted Infinite Horizon

$$V(s) = E\left(\sum_{t=0}^{\infty} \gamma^t r_t\right) \quad (۶-۲)$$

که E به ارزش مورد انتظار اشاره می کند. نرخ تنزیل، مجموع نامتناهی را محدود نگه می دارد. مدل بـازه نامتناهی تنزیلی عمومیتش را به خاطر این که ویژگی های ریاضی آن اثبات شده است، کسب کرده اسـت [۱۷]. بنابراین، در چارچوب پیشنهادی روی مدل بازه نامتناهی تنزیلی تمرکز می شود.

تابع ارزش[1] $V: S \rightarrow R$ به هر حالت سیستمی s، پاداش مورد انتظار کلی را با شـروع از s نسـبت می دهد، ارزش s. ارزش بهینه یک حالت، پاداش مورد انتظار موقع دنبال کردن سیاست بهینه می باشـد. تابع ارزش بهینه *V به صورت زیر است:

$$V^*(s) = \max_{\pi} E\left(\sum_{t=0}^{\infty} \gamma^t r_t\right) \quad (۷-۲)$$

ارزش حالت جاری s برابر پاداش فوری بعلاوه ارزش حالت بعدی s' است. بنابراین، ما یک ارزش بهینـه را با دنبال کردن معادله بازگشتی مربوط به احتمالات گذر بدست می آوریم. این معادلـه، **معادله بهینگـی بلمن**[2] نامیده می شود:

$$V^*(s) = \max_{a \epsilon A}\left(R(s,a) + \gamma \sum_{s' \epsilon S} T(s,a,s')V^*(s')\right) \quad (۸-۲)$$

با داشتن تابع ارزش بهینه *V، سیاست بهینه اجـرای اقـدامی اسـت کـه بـالاترین ارزش را برمـی گرداند و به صورت زیر تعریف می شود:

$$\pi^*(s) = \underset{a \epsilon A}{\operatorname{argmax}}(R(s,a) + \gamma \sum_{s' \epsilon S} T(s,a,s')V^*(s')) \quad (۹-۲)$$

[1] value function
[2] Bellman Equation

این نوع انتخاب اقدام، سیاست حریصانه نامیده می شود [۱۵].

۲-۴ روش های حل فرآیندهای تصمیم گیری مارکوف

روشهای زیادی برای پیدا کردن سیاست بهینه و یا تابع ارزش بهینه وجود دارد. از یک وجهه، این روش ها را می توان به دو دسته روش های برون خط[۱] و برخط[۲] تقسیم کرد. در روش های برون خط، با داشتن تابع گذر و تابع پاداش، تابع سیاست بهینه بدون تعامل با محیط بدست می آید. در این روش ها، سیاست بهینه با استفاده از روش های بهینه سازی مانند برنامه نویسی پویا[۳] قابل دستیابی است. این روش ها نیاز به دانش قبلی از مسئله دارند. در مقابل، روش های برخط- بدون تابع گذر و تابع پاداش- از طریق تعامل با محیط سیاست بهینه را پیدا می کنند.

دو نمونه از الگوریتم های معروف برون خط برای محاسبه سیاست بهینه، به نام تکرار ارزش[۴] [۱۸] و تکرار سیاست[۵] [۱۹] می باشد. در روش تکرار ارزش، در ابتدا ارزش دلخواهی برای هر حالت در نظر گرفته می شود. در هر مرحله، الگوریتم آن را به روز رسانی می کند تا به مقدار بهینه V^* همگرا شود [۱۷]. شرط خاتمه الگوریتم این است که مقدار V_{k+1} (s) - V_k (s) به ε برسد. تکرار روی همه اقدامات در همه حالات انجام می شود و تابع ارزش طبق رابطه زیر تطبیق می یابد:

$$Q(s,a) = R(s,a) + \gamma \sum_{s' \epsilon S} T\,(s,a,s') \quad V\,(s')] \qquad (۱۰-۲)$$

[1] Offline
[2] Online
[3] Dynamic Programming
[4] Value Iteration
[5] Policy Iteration

$$V(s) = \max_{a \in A} Q(s,a). \qquad (۱۱-۲)$$

ثابت شده است که این روش به تابع ارزش بهینه همگرا می شود [۲۰]. شبه کد الگوریتم تکرار ارزش در پیوست ۱ آورده شده است. روش دیگر برای حل MDPها توسط تکرار روی سیاست ها، به جای حالات و اقدامات است که تکرار سیاست نامیده می شود. این الگوریتم، با توجه به سیاست انتخابی تعیین می کند که آیا کنش جاری، ارزش حالت را بهبود می دهد یا نه؟ اگر منجر به بهبود ارزش جاری شود، سیاست جاری را به گونه ای تغییر می دهد که شامل آن کنش نیز باشد. شرط خاتمه در این روش این است که سیاست بدست آمده دیگر تغییری نکند. شبه کد الگوریتم تکرار سیاست در پیوست ۱ موجود است. روش تکرار سیاست در مقایسه با روش تکرار ارزش تکرارهای کمتری را دارد، اما اثبات شده است که روش تکرار ارزش در عمل سریعتر است [۲۱]. به همین خاطر، موقع حل مسائل برنامه ریزی روش های مبتنی بر ارزش بر تکرار سیاست ترجیح داده می شوند [۲۲]. در [۲۳]، بیان شده است که نحوه بازنمایی[1] نقش مهمی را در تعیین اینکه کدام روش برای مساله بهتر است بازی می کند.

همچنین، با توجه به اینکه تابع سیاست بر اساس یکی از توابع ارزش حالت (V) یا ارزش حالت-کنش (Q) تعریف می شود، دو گروه از الگوریتم ها مطرح هستند: الگوریتم های مبتنی بر تابع ارزش[2] و الگوریتم های جست و جوی سیاست[3]. در بخش های بعدی، هر کدام از این روش ها بطور جداگانه بررسی می شود.

[1] representation
[2] Value Function Algorithms
[3] Policy Search

۲-۴-۱ الگوریتم های مبتنی بر تابع ارزش

در این دسته از الگوریتم ها، ابتدا تابع ارزش بهینه (V*) و یا تابع ارزش-حالت (Q*) بهینه را پیدا کرده، سپس از آن جهت به دست آوردن سیاست بهینه استفاده می کنیم. به اینگونه الگوریتمها، الگوریتمهای مبتنی بر تابع ارزش می گوییم. در صورتی که مدل مشخص باشد، با استفاده از حل هر یک از دو معادله زیر، تابع ارزش بهینه بدست می آید:

$$V(s) = \max_{a \in A_s}[r(s,a) + \gamma \sum_{s'}^{\infty} p(s'|s,a)\, V(s')] \quad (۱۲-۲)$$

$$Q(s,a) = \max_{a \in A_s}[r(s,a) + \gamma \sum_{s'}^{\infty} p(s'|s,a) \max_{a' \in A} Q(s',a')] \quad (۱۳-۲)$$

در شرایطی که مدل مشخص نیست، دو رویکرد می تواند استفاده شود. در رویکرد اول، ابتدا با استفاده از تعامل با محیط مدل بدست می آید. سپس، با استفاده از مدل بدست آمده سیاست بهینه بدست می آید. این رویکرد، روش های مبتنی بر مدل[1] نامیده می شود ([۲]، [۲۴]). در رویکرد دوم با استفاده از تعامل با محیط و روش های یادگیری، سیاست بهینه مستقیما بدست می آید. به این روش ها، روش های مدل-آزاد[2] گفته می شود. دو نمونه از الگوریتم های معروف مدل-آزاد یادگیری کیو[3] [۲۵] و یادگیری سارسا[4] ([۲۶]، [۲]) می باشد که بسیار مورد توجه هستند و در ادامه به آنها اشاره می شود.

[1] Model Based
[2] Model Free
[3] Q-Learning
[4] SARSA

یکی از مهمترین پیشرفت ها در یادگیری تقویتی، توسعه یک الگوریتم کنترلی تفاضل زمانی[1] Off-policy به نام یادگیری کیو است. بروزرسانی تابع ارزش-حالت الگوریتم یادگیری کیو ، به صورت زیر می باشد:

$$Q(s,a) = (1-\propto)Q(s,a) + \propto [r(s,a) + \gamma max_{a' \epsilon A_s} Q(s',a')] \quad (۱۴-۲)$$

که در اینجا، α نرخ یادگیری است. در این مورد تابع ارزش-حالت یاد گرفته شده Q، مستقل از سیاستی که دنبال می شود، تابع ارزش-حالت بهینه Q^* را تقریب می زند. به همین دلیل به آن روش Off-policy گفته می شود. این بطور چشم گیری تحلیل الگوریتم را ساده می کند و اثبات های همگرایی را ممکن می سازد. نشان داده شده است که یادگیری کیو با احتمال ۱ به سیاست بهینه همگرا می شود [۲]. شبه کد مربوط به روش یادگیری کیو، در شکل ۲-۱ نمایش داده شده است.

Initialize Q(s,a) arbitrarily
Repeat (for each episode):
. Initialize s

. Repeat (for each step of episode):
. Choose a from s using policy derived from Q
. . . (e.g. , $\varepsilon - greedy$)
. . . Take action a, observe r, s'
. . . Q(s,a) ← Q(s,a) + α [r + $\gamma max_{a'}$Q(s',a') – Q(s,a)]
. . S ← s' ;
. Until s is terminal

شکل ۲-۱: شبه کد الگوریتم یادگیری کیو

[1] Temporal Difference

الگوریتم یادگیری سارسا به روش on-policy معروف است. به این معنا که، در مورد سیاستی که اجرا می کند عمل یادگیری را انجام می دهد. در همه روش های on-policy، بطور پیوسته Q^{π} برای سیاست π تخمین زده می شود، و در همان زمان، π حریصانه با توجه به Q^{π} تغییر داده می شود [۲]. در این الگوریتم، تابع Q به صورت زیر بروزرسانی می شود:

$$Q(s,a) = (1-\propto)Q(s,a)+\propto [r(s,a) + \gamma Q(s',a')] \quad (۱۵-۲)$$

شبه کد مربوط به روش یادگیری سارسا، در پیوست ۱ آمده است.

۲-۴-۲الگوریتم های جستجوی سیاست بهینه

یک روش دیگر برای حل مساله مارکوف برای به دست آوردن مقادیر بهینه V^* و Q^* این است که سیاست بهینه را مستقیما به دست آورد. به اینگونه روشها، روشهای جستجوی سیاست می گویند.

۲-۵ یادگیری تقویتی

یادگیری تقویتی، به مجموعه روش هایی گفته می شود که در آن عامل هوشمند با استفاده از تعامل با محیط، رفتار خود را بهبود می بخشد ([۲۱]، [۲]). عامل، حالتی از محیط را ادراک می کند و اقدامی را انتخاب می کند تا حالت محیط را تحت تاثیر قرار دهد. سپس، برای هر اقدامی که انتخاب می کند یک سیگنال عددی، به نام بازخورد[1]، از محیط دریافت می کند. این بازخورد در بهبود کارایی رفتار عامل هوشمند به کار برده می شود. هدف آن، ماکزیمم کردن مجموع بازخوردهایی است که در طی زمان دریافت می کند [۲۷]. این فرآیند یادگیری توسط آموزش یا از طریق نظارت انجام نمی شود، بلکه از

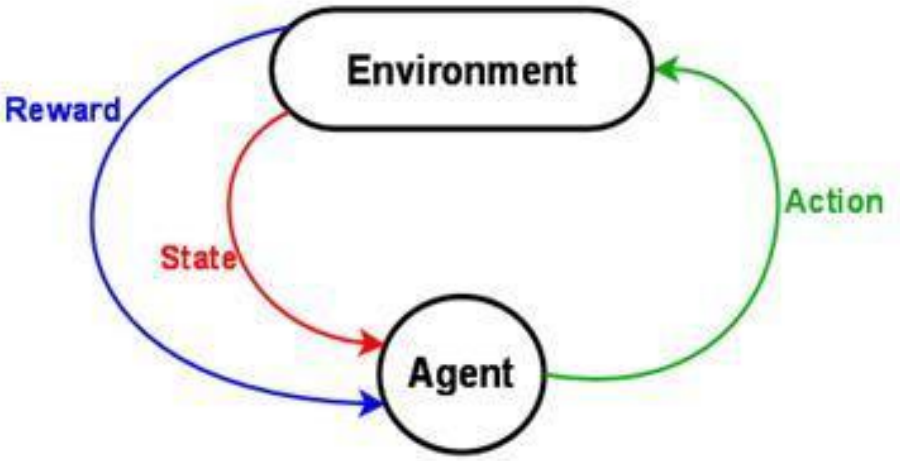
[1] reinforcement

طریق تعامل با یک محیط پویا و غیرقطعی انجام می شود [۲۸]. یادگیری تقویتی، یادگیری از طریق سعی و خطا می باشد. هدف این مکانیزم پیدا کردن روشی بهینه برای رسیدن به هدف است..

اقدامات عامل در ابتدا، با توجه به فقدان دانش تصادفی هستند. عامل با انجام اقـدامات، رفتـارش را بر اساس بازخورد محیط تطبیق می دهد. این فرآیند یک حلقه کنترلی از مدل یادگیری تقویتی ایجاد می کند. همانطور که توسط [۲۱] تعریف شده، این مدل شامل:

- مجموعه ای از حالات محیطی S، فضای حالت
- مجموعه ای از اقدامات A که عامل می تواند انجام دهد، فضای اقدام.
- مجموعه ای از سیگنال های بازخورد، معمولا R یا زیرمجموعه ای از آن.

S و A حوزه مساله را تعریف می کنند. حلقه کنترلی مربوطه در شکل ۲-۲ به تصویر کشیده شده است: بر اساس حالت جاری s و پاداش r، عامل یک اقدام $a \in A$ را انتخاب می کند که حالـت سیسـتم s را تغییـر می دهد، و چرخه به همین ترتیب ادامه دارد.



شکل ۲-۲: مدل یادگیری تقویتی

راه حل برای مسئله یادگیری تقویتی، تعیین اقدام مناسب در هر حالت سیستمی است. این یک مسئله بهینه سازی است: عامل باید یک سیاست π را ایجاد کند که یک اقدام را به یک حالت سیستم نگاشت کند، $\pi: S \rightarrow A$ [۱۵].

۲-۶ کاوش

در وضعیت هایی که مدل MDP در ابتدا ناشناخته است، عامل تنها از طریق کاوش[1] در محیط می تواند یادگیری را انجام دهد. بنابراین، یک بخش مهم از الگوریتم یادگیری تقویتی طرح اکتشاف است. نقش اکتشاف، بدست آوردن اطلاعات جدید با انتخاب اقدام مناسب است. از این طریق، عامل را به سوی حالات ناشناخته MDP هدایت می کند [۲۹]. یک استراتژی کاوش خوب باید اطمینان بدهد که فضای حالت-اقدام به خوبی کاوش شده است. در مقابل، انتخاب اقدامات بر اساس سیاست مفروض، بهره برداری[2] نامیده می شود. یک مسئله مهم در هنگام انتخاب اقدامات، مساله توازن بین کاوش و بهره برداری می باشد. ارزیابی سیاست های گوناگون از طریق کاوش، یا بهره گیری از فواید دانش جاری؟.

یک روش اکتشاف ساده اما موثر، روش e-greedy است. $\varepsilon \in [0,1]$ احتمال اکتشاف می باشد و در هر مرحله زمانی، عامل سیاست π را با احتمال 1-e دنبال می کند. در موارد دیگر، یک اقدام تصادفی اجرا می شود. سیاست های e-greedy، تنها یک مرحله اکتشافی را با احتمال e فراهم می کنند. برای دو مرحله اکتشافی در یک سطر، احتمال آن بسیار ناچیز، e^2، است. بنابراین، سیاست های e-greedy عامل را نزدیک به مسیرهای ناشی از π حفظ می کنند و رفتار آموزش دیده را تنها اندکی بهبود می دهند. در نتیجه، استراتژی هایی که نیاز به محدوده های دورتری از π دارند ممکن است کشف نشده باقی بمانند.

[1] exploration
[2] Exploitation

ایده این روش آن است که، بهبود در نزدیکی سیاستی که موفقیت قابل قبولی را نشان داده است امیدوارکننده است.

رویکرد جدیدی که در زمینه اکتشاف محیط وجود دارد، کاوش توسط هیوریستیک می باشد ([۳۰]، [۳۱]). الگوریتم های اخیر یادگیری شامل E^3 [۳۲]، MBIE [۳۳]، R-Max [۳۴]، GCB [۳۵]، UCRL [۳۶] و OLP [۳۷] تکنیک های کاوش کارا را استفاده می کنند. این الگوریتم ها، اغلب بر اساس اصل به اصطلاح "خوش بینی در مواجهه با عدم قطعیت" هستند. ایده آنها، کشف حرکاتی است که عامل در گذشته کمتر تجربه کرده است. بهرحال، این الگوریتم ها در مواجهه با فضاهای حالت و یا اقدام خیلی بزرگ، غیرعملی هستند.

۲-۷ چالش های مطرح در یادگیری تقویتی

یک مسئله مهم کاربرد یادگیری تقویتی در فضاهای حالت پیچیده، که در مسائل جهان واقعی با آنها مواجه هستیم، می باشد. یادگیری تقویتی، به خوبی برای مسائل بزرگ مقیاس پذیر نیست. اگرچه تکنیک های یادگیری تقویتی در حل مسائل بزرگ موفق هستند، اما زمان آموزش آنها، به خاطر کاوش زیاد، اغلب خیلی طولانی است. مساله دیگر مربوط به نمایش داخلی تابع ارزش است. استفاده از جداول ارجاع[1] برای ذخیره سازی، به خاطر مصرف زیاد حافظه، جز برای محیط های کوچک مناسب نیست [۲۱]. برای فضاهای حالت پیوسته، که در مسائل واقعی با آنها مواجه هستیم، نمایش همه حالات سیستم غیر ممکن است. بنابراین، نیاز به استفاده از نمایش های تقریبی برای ذخیره سازی تابع ارزش می باشد [۴].

[1] lookup table

مسئله دیگر، استفاده مجدد دانش بدست آمده برای فرآیند یادگیری جاری می باشد. حل یک MDP، به معنای پیدا کردن یک راه حل بهینه برای مسئله مشخص شده توسط حوزه خاصی است. به عبارت دیگر سیاست آموزش دیده π، به MDP که سیاست بر اساس آن هست قابل اعمال می باشد، و قابل اعمال به مسائل تصمیم گیری دیگر نیست. در نتیجه، هیچ دانش عمومی درباره وظیفه نمی تواند از سیاست استنتاج شود. در نتیجه اگر ما هدف وظیفه را تغییر دهیم، یا اگر محیط عامل را کمی تغییر دهیم، سیاست آموزش دیده ممکن است کاملا بی ارزش باشد. در چنین مواردی، نیاز است تا وظیفه اصلاح شده مجددا از ابتدا یاد گرفته شود. از طریق تعمیم[1]، دانش کسب شده در زیرمجموعه حالات S می تواند در خارج از این زیرمجموعه قابل استفاده باشد.

با استفاده از تعمیم، سیستم قادر به استنتاج حالات مشابه که مقادیر مشابهی دارند می باشد. پس از آن، انتخاب اقدام در حالاتی که خیلی تفاوت ندارند مشابه است. بنابراین، تعمیم امکان استفاده از تجربه آموزش دیده را در شرایط کمی متغیر فراهم می کند. همچنین، ممکن است حالات ساختار خاصی را در حوزه مربوطه به اشتراک بگذارند. ساختارهای یک حوزه، به عنوان ویژگی های تکرارپذیر که انتخاب اقدام را تحت تاثیر قرار می دهند تعریف می شوند. مثلا، ساختار نقشه خیابان ها توسط جاده ها مشخص می شود، و محیط های اداری توسط دیوارها مشخص می شوند. در این محیط ها، مکان های زیادی ممکن است وجود داشته باشد که ساختار محلی را به اشتراک بگذارند. اینها موقعیت هایی هستند که عامل می تواند از دانشش مجددا استفاده کند.

مساله استفاده مجدد از دانش، با مساله بکارگیری یادگیری تقویتی در فضاهای حالت پیچیده مرتبط است. اگر سیستم یادگیری درکی را از ساختار فضای حالت کسب کند، می توان از این درک در مکان ها و موقعیت های گوناگون در کل فضای حالت استفاده کرد. سیستمی که قادر به تعمیم است،

[1] Generalization

وقتی با یک وضعیت جدید مواجه شود نیاز به یادگیری همه چیز از ابتدا نـدارد. در مقابـل، تجربـه جمـع شده ممکن است در این وضعیت ها قابل استفاده باشد، بنابراین تلاش های یادگیری را کاهش مـی دهـد [۴]. همچنین بخشی از فضای حالت را که هیچ دانشی در مورد آن کسب نشـده کـاهش مـی دهـد. یـک راهکار مفید در رابطه با چالش های مطرح شده و مسـئله کـاوش هـای طـولانی مـدت [۳۸]، اسـتفاده از انتزاع[۱] در یادگیری تقویتی کیفی[۲] می باشد که در ادامه بررسی خواهد شد.

۲-۸ انتزاع

انتزاع، کاهش داده ای است که اطلاعات غیر ضروری را حذف می کند. با انتزاع مسئله، جزییات غیر مرتبط در نظر گرفته نمی شود و به مساله در سطح بالا نگاه می شود. بنابراین از طریق انتزاع، چالش هایی که در بخش قبل مطرح گردید تا حدودی کاهش می یابد. در ادامه، انتزاع در دو سطح انتزاع حالات[۳] [۴] و انتزاع زمانی[۴] [۱۰] بررسی می شود.

۲-۸-۱ انتزاع زمانی

در انتزاع زمانی، دنباله ای از اقدامات تشکیل یک فراکنش[۵] را می دهند. نمونه هایی از فراکنش هـا شامل برداشتن یک شی، رفتن به نهار، و سفر کردن به شهر دیگر می باشد. این فراکنشها می تواند شامل فراکنش های دیگر، و یا اقدامات اولیه باشد. مثلا، یک ربات خانگی را در نظر بگیرید که تصمیم دارد عمل گرفتن آب میوه را انجام دهد. این عمل شامل مجموعه ای از فراکنش های پایه مانند برداشتن میوه، قـرار

[1] Abstraction
[2] Qualitative reinforcement learning
[3] Spatial Abstraction
[4] Temporal Abstraction
[5] Macro Action

دادن آن داخل دستگاه آبمیوه گیری، روشن کردن دستگاه و ریختن آب میوه در ظرف می باشد. از اقدامات پایه انجام این مهارت، می توان به تغییر مختصات و زاویه مفاصل اشاره کرد. این اعمال پایه به ترتیب خاصی اجرا می شوند تا عمل گرفتن آب میوه انجام شود [۱۰]. این ساختار، یک معماری سلسله مراتبی[1] را ایجاد می کند که در آن اقدامات اولیه در سطوح پایین تر، و اقدامات توسعه یافته تر در سطوح بالاتر قرار دارند [۳۹]. انتزاع زمانی، بطور ویژه توسط ساتن داخل چارچوب یادگیری تقویتی و فرآیندهای تصمیم گیری مارکوف بررسی شده است [۱۰].

۲-۸-۲انتزاع در سطح حالات

در مقابل انتزاع در سطح اقدام، همچنین می توانیم حالات سیستم را در نظر بگیریم. انتزاع حالات، به معنای ادغام مجموعه ای از حالات متفاوت در یک حالت انتزاعی غیر قابل تمایز می باشد [۴۰]. در این شرایط، فضای حالت انتزاعی را داریم که می تواند بطور قابل توجهی اندازه فضای حالت را کاهش دهد. همچنین، ورودی که یک عامل مصنوعی از سیستم حسی خود دریافت می کند انتزاعی از جهان واقعی است: آن اطلاعات غنی از جهان اطراف را، به یک تصویر دوربین یا تعدادی مقادیر فاصله کاهش می دهد. هیچ سنسوری، از جمله حواس انسان، نمی تواند همیشه تصویر کاملی از جهان اطراف را فراهم کند. بهرحال این اطلاعات برای حل موفق مسائل، توسط انسان ها و عامل های هوشمند کافی است. هدف باید تشدید این انتزاع، به نحوی باشد که کیفیت راه حل حفظ شود.

وقتی ادغام حالات انجام می شود، مساله کاهش یافته جدید می تواند از طریق روش های معمول حل شود. این ویژگی ها منجر به شتاب فرآیند یادگیری و پیدا کردن سیاست بهینه توسط عامل می شود.

[1] Hierarchical Reinforcement Learning

تعداد بیشتر حالاتی که می تواند شامل یک مفهوم با یک عمل متعاقب باشد، فضای حالت کوچکتری را می سازد. تحقیقات زیادی انجام شده که نشان دهد، تجرید حالات تاثیر زیادی برروی بهینگی راه حل ندارد. در واقع راه حلی که بدست می آید، یک حد واسط بین بهینگی و میزان کامل بودن راه حل است.

از طریق انتزاع، زیرمجموعه ای از حالات ادغام می شوند. بنابراین، حالات اصلی دیگر قابل تمایز نیستند و مساله نیمه مشاهده پذیر می شود. مساله ما دیگر یک MDP نیست، بلکه یک POMDP[1] است. عمل کردن در فضای حالت انتزاعی، باعث می شود سیستم ویژگی مارکوف را از دست بدهد. بنابراین روش های حل MDP معمول، برای بدست آوردن یک راه حل تضمین شده قابل اعمال نیست. بهرحال، بدست آوردن روش های حل برای یک POMDP خیلی سخت است.

افزودن این نوع پیچیدگی، با هدف آسان سازی فرآیند یادگیری مغایرت دارد. یک ایده ساده این است که، ویژگی نیمه مشاهده پذیری نادیده گرفته شود و از روش های معمول حل MDP استفاده گردد. با توجه به نیمه مشاهده پذیری، سیاست حاصل ممکن است بهینه نباشد. نمی توان انتظار داشت که بکارگیری دقت پایین تری از فضای حالت، بدون هیچ هزینه ای باشد. هنگام دور ریختن اطلاعات، همیشه باید از دست دادن کیفیت در نظر گرفته شود. اگر ما می خواهیم که کارایی یادگیری را در کنار سرعت، تسهیل استفاده و قدرت افزایش دهیم، راه حل کمی بدتر در اکثر موارد قابل قبول است. در چارچوب پیشنهادی، از همان روش های حل MDP معمول برای فضای انتزاعی استفاده شده است.

[1] Partial Observable Markov Decision Process

۲-۹ شکل دهی پاداش

در یادگیری تقویتی، هدف عامل رسیدن به سیاستی با بیشترین مجموع پاداش دریافتی است. بنابراین، تابع پاداش عامل را به سوی سیاست بهینه هدایت می کند. روش شکل دهی پاداش [۶]، یک راه حل موثر برای مشکل پاداش های تاخیری می باشد. ایده اصلی نهفته در شکل دهی پاداش، در نظر گرفتن یک تابع پاداش علاوه بر بازخورد دریافتی از محیط است. در نتیجه، به بهبود نرخ همگرایی و افزایش سرعت فرآیند یادگیری کمک می کند. تاثیر این تابع پاداش مجازی که توسط طراح فراهم شده است می تواند به شرح زیر توصیف شود: عامل در اپیزودهای اولیه یادگیری پاداشی را دریافت نمی کند. در این شرایط برای بهبود رفتار عامل، پاداش مجازی به آن داده می شود. در اپیزودهای بعدی، عامل رفتارش را بر اساس بازخوردهای محیط بهبود می دهد و اثر پاداش ساختگی را به تدریج می توان کاهش داد [۵]. بنابراین، پاداش ساختگی به عنوان یک فاکتور تسریع در فرآیند یادگیری و نرخ همگرایی عمل میکند. در الگوریتم ارائه شده از چارچوب پیشنهادی، پاداش ساختگی بدست آمده از تحلیل گراف استفاده شده است که در فصل سوم بررسی خواهد شد.

۲-۱۰ یادگیری تقویتی کیفی

همانطور که بیان شد، نمایش انتزاعی فرایندهای کنترلی در محیط های واقعی ضروری است. موقع استفاده از انتزاع در وظایف کنترلی عامل نمایش های کیفی انتخاب مناسبی هستند، زیرا آنها برای مدل کردن آنچه برای وظیفه در دست ضروری است طراحی شده اند [۴۱]. انتزاع، به عنوان یک تابع غیر یک به یک تعریف می شود. به این معنی که، چندین موجودیت در فضای حالت اصلی می تواند یک نمایش

انتزاعی را به اشتراک بگذارد، یک واقعیت که بر طبق [۴۲]، "در ماهیت یک سیستم نمایش کیفی" است. بنابراین، نمایش های کیفی به خوبی مناسب برای نمایش مطلوب انتزاعات در فرآیندهای کنترلی هستند.

نمایش های کیفی روی دقت اطلاعات تکیه نمی کنند. در عوض، آنها روی مفاهیمی مانند "سمت چپ"، "پشت"، "دور" یا نزدیک تکیه می کنند. آنها با نادیده گرفتن جزئیات متریک جهان، روی توصیف موجودیت های جهان و ارتباطشان با هم تمرکز دارند. با بکارگیری انتزاع کیفی، بهبودهایی در سرعت یادگیری و قدرت آن می تواند بدست آید، که مطمئنا توجیه قابل قبولی برای استفاده اش می باشد. همچنین استفاده از انتزاع کیفی می تواند منجر به رفتار تعمیم یافته عامل، تقریبا بدون هیچ تلاش اضافی شود.

۲-۱۱ مروری بر ادبیات

در بخش های قبلی بیان شد که در وضعیت هایی که مدل MDP در ابتدا ناشناخته است، عامل تنها از طریق اکتشاف می تواند مسئله را حل کند. نقش اکتشاف، بدست آوردن اطلاعات جدید با انتخاب اقدام مناسب است. از این طریق، عامل را به سوی حالات ناشناخته هدایت می کند. الگوریتم های اخیر یادگیری تقویتی، روش های کاوش کارایی را استفاده می کنند [۲۹]. بهرحال با توجه به اینکه کاوش فرآیندی نامطلوب است، این الگوریتم ها در مواردی که فضاهای حالت یا اقدام خیلی بزرگ یا نامتناهی هستند، غیرعملی می باشند. در نتیجه، باید سعی شود تا میزان کاوش را کاهش دهیم. در بین طرح های تقریب پیشنهاد شده مانند [۴۳]، [۴۴] و [۴۵] یادگیری کیفی با استفاده از انتزاع به عنوان راهکاری برای تولید MDPهای خیلی فشرده می تواند استفاده شود. در ادامه، مروری بر برخی روش های ارائه شده در این رابطه خواهیم داشت

۲-۱۱-۱ یادگیری کیفی

در عمل، نباید انتظار داشته باشیم که فرد خبره اطلاعات کاملی درباره مدل MDP داشته باشد. در نتیجه اگر بتوان دانش نادقیق و مبهمی را در رابطه با احتمالات انتقال کسب کرد، برای کاهش مقدار کاوش راه حل خوبی است. این اطلاعات برای ساده سازی استدلال مورد نیاز، برای تعمیم و تحلیل تجربه یادگیری عامل می تواند کافی باشد. به این ترتیب، امکان حل مسئله را بدون کاوش گسترده فراهم می کند [۴۶]. در این زمینه، چندین فعالیت انجام گردیده است که در ادامه به برخی از آنها اشاره می شود.

دین و کانازاوا [۴۷]، مدلی از استدلال زمانی[1] را ارائه کردند که در شرایطی که دانش کاملی از محیط وجود ندارد مناسب است. در این مقاله، مدلی از استدلال علی معرفی شده است که دانش مربوط به ارتباطات علت و معلولی، و دانش مربوط به گرایش را در رابطه با استمرار گزاره ها توصیف می کند. در این مدل رمزگذاری برای مدت زمان استمرار گزاره ها، به شکل نمایش شبکه مدل های احتمالاتی انجام می شود. در نتیجه، قادر خواهیم بود درباره پدیده هایی که مدل علت و معلولی دقیقی را نداریم استدلال کنیم. این نوع شبکه، دارای محددویت هایی است و باعث گردید که این روش کارایی خوبی نداشته باشد.

دیردن و بویتلیر [۴۸]، روش انتزاع خودکار را برای MDPها پیشنهاد کردند. در این روش مرتبط ترین ویژگی های مساله تعیین می شود. سپس، مسئله کاهش یافته بر اساس این ویژگی های مرتبط حل می شود. راه حل بدست آمده در این روش، مستقیما به مسئله اصلی قابل اعمال است. در ابتدا نمایش ساخت یافته خاصی از MDP، با استفاده از شبکه های بیزین دومرحله ای برای اقدامات و پاداش ها استفاده می شود. این نحوه نمایش باعث می شود تا تابع انتقال برای مسائل بزرگ، به شیوه ای مختصر نمایش داده شود. مدل پیشنهادی در این کار، دارای این نقطه ضعف است که روش ادغام استفاده شده در اینجا نیاز به حذف یکنواخت لیترال ها از توصیف مسئله دارد. بنابراین ادغام در این روش، ثابت و

[1] temporal reasoning

یکنواخت است. روش های ادغام مفیدتری نیز وجود دارد که می توان از آنها برای بهبود این کار استفاده کرد. در بخش های بعدی بطور مفصل به انواع روش های ادغام اشاره خواهد شد.

دین و گیوان [۴۹]، روشی را برای فشرده سازی و گروه بندی حالات مشابه بر طبق ویژگی های خاص مسئله بیان کردند. این مقاله در درجه ی اول، به معرفی روش مینیمم سازی مدل برای MDPها می پردازد. در این روش، فضای حالت اصلی به بخش هایی گروه بندی می شود که هر یک از آنها پایدار می باشد و احتمال انتقال مدل اصلی را حفظ می کند. آنها برای تحلیل مسائل برنامه ریزی ارائه شده به عنوان MDP، مفهوم همگنی bisimulation[1] را استفاده کردند. به عنوان یک نمونه از افراز فضای حالت همگن می توان به حذف متغیرهای نامرتبط در توصیف حالت اشاره کرد. الگوریتم ارائه شده، درشت ترین بهسازی همگن از هر افراز فضای حالت یک MDP را ایجاد می کند. الگوریتم ارائه شده در این مقاله، افراز دقیقی را تولید می کند و می تواند خیلی پیچیده باشد. در بسیاری از کاربردها می توان با استفاده از یک مدل تقریبی، سیاست های نزدیک به بهینه را بدست آورد.

بویتلیر و همکارانش [۴۴] مقاله ای را ارائه دادند، که مروری بر روش های مرتبط با MDP می کند. سپس، چارچوبی را ارائه می دهد تا بسیاری از کلاس های مسائل برنامه ریزی در هوش مصنوعی را مدل کند. آنها بطور گسترده، روی تکنیک های انتزاع و ادغام تمرکز کردند. این تکنیکها گروه بندی صریح یا ضمنی حالات را با توجه به ویژگی خاصی ممکن می سازند. در این مقاله، آنها تکنیک های ادغام و خلاصه سازی محیط را برای نمایش تابع انتقال و تابع ارزش به شکل درخت تصمیم استفاده کردند. یک نمایش درخت تصمیم از یک سیاست یا یک تابع ارزش، فضای حالت را به یک خوشه مجزا برای هر برگ درخت افراز بندی می کند.

[1] bisimulation homogeneity

همچنین، در این کار به بررسی تکنیک های تجزیه مسئله پرداخته شده است که در آن یک MDP به بخش هایی شکسته می شود. سپس، هر یک از این بخش ها بطور مستقل حل می شود. در مرحله بعدی، راه حل این بخش ها با هم ترکیب می شوند، یا برای راهنمایی جست و جو برای یک راه حل عمومی استفاده می شوند. برای استفاده از تکنیک های تجزیه مسئله، باید هر بخش از MDP قابلیت حل جداگانه را داشته باشد. نحوه انجام تقسیم و حفظ این شرط، کمی مشکل بوده و راه حل مشخصی برای آن وجود ندارد.

فرآیندهای تصمیم گیری مارکوف کیفی، توسط بونت و پیرل [۵۰] و سبدین [۵۱] مطالعه شدند. در نظریه ارائه شده در [۵۱]، یک همتای کیفی برای مدل فرآیندهای تصمیم گیری مارکوف نیمه مشاهده پذیر[1] پیشنهاد شده است. دراین نظریه، عدم قطعیت با استفاده از توزیع های احتمال کیفی مدل شده است. این روش، روی نظریه احتمالات تصمیم تحت عدم قطعیت تکیه می کند. در فرآیندهای تصمیم گیری مارکوف نیمه مشاهده پذیر با یک فضای حالت محدود، فضای حالت باور بدست آمده آن نامحدود است. اما در چارچوب احتمالی، فضای حالت باور محدود باقی می ماند. بهرحال، فرآیندهای تصمیم گیری مارکوف نیمه مشاهده پذیر باعث پیچیدگی بیشتر الگوریتم می شوند.

بونت و پیرل [۵۰]، نظریه ای کیفی از MDPها و MDPهای نیمه مشاهده پذیر توسعه دادند. از این نظریه برای مدل کردن وظایف ترتیبی تصمیم گیری، وقتی تنها اطلاعات کیفی در دسترس است، می توان استفاده کرد. سپس، نشان دادند که چطور الگوریتم های استاندارد می تواند برای حل مدل های حاصل استفاده شود. نظریه حاصل می تواند به عنوان تعمیمی بر نظریه MDP استاندارد دیده شود. مطالعه بونت و پیرل [۵۰] کاملا نظری است. در حالی که سبدین [۵۱]، یک کاربرد از الگوریتمش را به یک gridewor 3*3 توصیف می کند.. هیچ ارتباط واضحی بین نمایش های کیفی پیشنهاد شده در دو

[1] POMDP

مقاله [۵۰] و [۵۱]، و احتمالات کمی حاصل از تخمین تعاملات تجربی وجود ندارد. به همین دلیل، هیچ یک از دو مطالعه تلاشی برای ترکیب توصیف مسئله کیفی با کاوش کمی نمی کند.

اپستین و دی جانگ [۳۸]، الگوریتمی را ارائه دادند که به خبره اجازه می دهد تا دانش غیر دقیقی را از احتمالات انتقال مشخص کند. الگوریتم پیشنهادی برای مسائلی که بطور کیفی مشخص شده است، یا وقتی هیچ راه حلی در دسترس نیست، می تواند منجر به کاهش میزان کاوش شود. بهرحال در این روش، مدل نیاز به دانش پیشینی برای مشخص کردن این که جهان چطور کار می کند، دارد. این روش در مواردی استفاده می شود که خبره درباره دینامیک جهان اطلاعات دارد، اما چگونگی حل مسئله را نمی داند.

ریس و همکارانش [۵۲]، یک نمایش انتزاعی بر اساس حالات کیفی برای MDPهای فضای پیوسته ارائه کردند. همچنین، آنها یک متدولوژی برای یادگیری خودکار مدل بر اساس این نمایش انتزاعی ارائه دادند. فضای حالت بر اساس ساختار پاداش، به مجموعه ای از حالات انتزاعی افراز بندی می شود. هر مجموعه از حالات انتزاعی، شامل حالاتی با مقادیر پاداش مشابه است. در این کار، این حالات انتزاعی حالات کیفی نامیده شده است. این افرازها، از طریق الگوریتم های یادگیری درخت تصمیم بطور خودکار ایجاد می شود. در نهایت از طریق کاوش تصادفی محیط، یک مدل انتقال روی حالات انتزاعی ایجاد می شود. خروجی این الگوریتم، یک MDP خیلی فشرده است که با روش های استاندارد قابل حل می باشد. در این روش، افرازبندی بر اساس ساختار پاداش فضای حالت انجام می گیرد. این کار می تواند با اصلاح افرازها توسط در نظر گرفتن سایر پارامترها و ویژگی های مدل بهبود یابد. با توجه به اهمیت انتزاع در یادگیری کیفی، در ادامه انواع روش های انتزاع در ادبیات موضوع بررسی خواهد شد.

۲-۱۱-۲ انتزاع

انتزاع توانایی مهمی در رابطه با عامل های هوشمند است. به همین دلیل، جنبه های متفاوت انتزاع در زمینه های علمی گوناگون بررسی شده است. در زمینه های علمی مختلف مانند علوم شناختی، هوش مصنوعی، معماری، زبان شناسی، جغرافیا، و غیره اصطلاحات گوناگونی برای مفهوم انتزاع بکار برده شده است. از بین آنها می توان به granularity ([۵۳]، [۵۴])، generalization ([۵۵])، schematization ([۵۶]، [۵۷])، idealization ([۵۶])، selection ([۵۶]، [۵۸])، amalgamation ([۵۸])، و aspectualization ([۵۹]، [۶۰]) اشاره کرد. تعدادی از اصطلاحات تعریف شده برای انتزاع دارای مفاهیم همپوشانی هستند. گاهی اصطلاحات متفاوت معنی مشابهی دارند، یا یک عبارت خاص برای مفاهیم متفاوتی استفاده می شود. در نتیجه، این اصطلاحات به شکلی دقیق از هم متمایز نمی شوند [۴۱].

از یک دیدگاه، انتزاع در دو سطح حالت و اقدام بررسی می شود. در انتزاع سطح حالات، فضای حالت به مجموعه متناهی از سلول ها گسسته می شود. هر سلول، حالاتی را که در آن قرار گرفته اند در خود ادغام می کند. هنگامی که ادغام انجام می شود مسئله جدید، یک مسئله یادگیری در فضای حالت کاهش یافته است که با تکنیک های معمول قابل حل است. در روش دیگر، انتزاع در سطح اقدام انجام می گیرد. گروهی از اعمال ترتیبی و پشت سرهم، به عنوان یک عمل مجرد در نظر گرفته می شوند و تشکیل یک فراکنش[1] را می دهند ([۱۰]، [۶۱]). در این رابطه، تعدادی از انتزاعات در ادبیات برنامه ریزی و یادگیری تقویتی پیشنهاد و بررسی شده اند. در ادامه، مروری بر تعدادی از این کارها خواهد شد.

[1] macro action

۲-۱۱-۲-۱ انتزاع زمانی

ساتن عبارات مختلفی را مانند "فراکنش"، "رفتار" و "اقدامات انتزاعی" برای اعمال مجرد در انتزاع زمانی معرفی کرده است. برای نمایش این فراکنش ها، چارچوب های مختلفی وجود دارد که از آن جمله می توان به چارچوب های گزینه [۱۰]، MAXQ [۱۲] و ماشین های انتزاعی [۶۲] اشاره کرد. فراکنش ها در این چارچوب ها، به صورت سلسله مراتبی در یک راستا قرار می گیرند. در این حالت، یک مسئله بزرگ به مسائل کوچکتر شکسته می شود. در ادامه، این مسائل کوچک به ترتیب خاصی اجرا می شوند که منجر به حل مسئله اصلی می شود. هر یک از این مسائل کوچک، خود می تواند شامل یکسری مسائل کوچکتر باشد که به صورت ساختار سلسله مراتبی قرار دارند.

طراحی ساختار سلسله مراتبی فراکنش ها به دو شیوه قابل انجام است: این عمل می تواند به صورت دستی و توسط طراح انجام شود، و یا به صورت خودکار انجام گیرد [۶۱]. روش طراحی دستی ساختار سلسله مراتبی، نیاز به شناخت طراح نسبت به جزئیات محیط دارد. این مسئله، در رابطه با محیط های بزرگ و پیچیده می تواند مشکل باشد. روش تعیین خودکار مهارت ها، این مشکل را ندارد و در روش های اخیر به آن توجه زیادی شده است. در روش های تعیین خودکار مهارت ها، معمولا مسئله به مجموعه ای از مسائل کوچکتر شکسته می شود. برای شکستن مسئله اصلی، نقاطی به عنوان نقاط شکست در نظر گرفته می شود. بر این اساس، دو روش اصلی برای کسب خودکار مهارت وجود دارد. در روش اول، عامل ابتدا حالات استراتژیک را در محیط شناسایی می کند. به این حالات استراتژیک، اهداف میانی گفته می شود. سپس در مرحله بعد، مهارتها بر اساس این اهداف میانی و با استفاده از چارچوب های نمایش سلسله مراتبی، مانند گزینه ایجاد می شوند ([۶۳]، [۶۴]).

از جمله این روش ها، می توان به روش های اکتساب مهارت مبتنی بر سیاست [۶۵]، روش های مبتنی بر گراف ([۶۴]، [۷]) و روش های مبتنی بر بسامد [۶۶] اشاره کرد. در روش دیگر، ایجاد ساختار سلسله مراتبی به طور مستقیم انجام می گیرد. در نتیجه، ایجاد ساختار سلسله مراتب در این روش بدون شناسایی اهداف میانی انجام می گیرد. این فرآیند، با استفاده از شیوه هایی مانند درخت تصمیم گیری ([۶۷]، [۶۸]،)، برنامه نویسی ژنتیک [۶۹] و مدل های گرافیکی ([۷۰]، [۷۱]) انجام می شود. بهرحال، استفاده از چارچوب های سلسله مراتبی در محیط های در مقیاس بزرگ هنوز چالش برانگیز است. چالش دیگر، ایجاد خودکار و انطباق سلسله مراتب ها می باشد.

۲-۱۱-۲-۲ انتزاع حالات

انتزاع حالات، بر اساس جنبه های متفاوت انتزاع، می تواند به شیوه های مختلفی انجام گیرد. مثلا ممکن است تنها زیرمجموعه ای از اطلاعات در دسترس در نظر گرفته شود [۷۲]، سطوحی از جزئیات اطلاعات کاهش یابد [۶۰]، و یا از اطلاعات در دسترس برای ساخت موجودیت های انتزاعی جدید استفاده شود [۷۳]. در این رویکرد، انتزاع حالت به دو شیوه انجام می شود. انتزاع حالات می تواند توسط افراز بندی فضای حالت که ساختاری را در طی یادگیری ایجاد می کند انجام شود. همچنین، می توان از طریق ادغام دانش حوزه در نمایش حالات به آن دست یافت.

در سال ۲۰۱۰ [۴۱]، یک معماری برای انتزاع حالات پیشنهاد کرده است. انتزاع نهایی توسط یک فرآیند یادگیری خودمختار است. در این کار، ادغام با بکارگیری تکنیک های انتزاعی، مستقیما به داده های ورودی سیستم سنسور، انجام می گیرد. بدین معنا که فضای حالت، انتزاعی از داده های سنسور ورودی می باشد. فرض می شود که عامل، به جای فضای حالت اصلی S روی مجموعه ای از موجودیت

های به تازگی ایجاد شده به نام مشاهدات[1] عمل می کند. در این روش، افرازی روی فضای حالات ایجاد می شود و حالات در دسته های مختلف قرار می گیرند. این رویکرد، نیاز به ملاقات همه حالات بطور تکراری دارد که فرآیندی وقت گیر است.

۲-۱۱-۲-۳ ترکیب انتزاع زمانی و انتزاع حالات

به خاطر مقیاس پذیری زیربهینه روش های انتزاع زمانی، انتزاع حالات بطور فزاینده در این روش ها گنجانیده شده است. در [۷۴] و [۷۵]، با استفاده از درخت تصمیم انتزاع حالات انجام گردیده و ساختارهای سلسله مراتبی شکل گرفته است. در این روش ها، عامل با تعامل با محیط اطرافش اطلاعاتی را بدست می آورد و آنها را به شکل درخت تصمیم ارائه می دهد. عامل مجموعه ای از ویژگی ها را، از تمام حرکات قبلی خود که در حافظه ذخیره کرده است استخراج می کند. سپس، درخت تصمیم بر مبنای این ویژگی ها ساخته می شود. در این درخت تصمیم، هر گره میانی نشان دهنده محل های تصمیم گیری عامل است. برگ های این درخت تصمیم، حالت های مجرد را نشان می دهند.

در [۶۷] درخت تصمیم U-Tree [۷۶] با چارچوب گزینه توسعه یافته است. در این کار الگوریتم U-tree برای ساخت خودکار نمایش های فضای حالت، با چارچوب گزینه تطبیق یافته است. الگوریتم حاصل، با استفاده از یک وظیفه سلسله مراتبی ساده مورد بررسی قرار گرفته است. در این روش، چارچوب گزینه [۱۰] استفاده می شود که یکی از چارچوب های یادگیری تقویتی برای انتزاع زمانی اقدامات است. در بسیاری از موارد، اجرای دقیق یک سیاست گزینه بستگی به همه ویژگی های حالت در دسترس ندارد. علاوه بر این، ویژگی های مرتبط اغلب از یک گزینه تا گزینه دیگر متفاوت هستند. در یک سیستم

[1] Observations

یادگیری سلسله مراتبی، انجام انتزاع حالات مختص گزینه امکان پذیر است. در این شرایط، ویژگی های نامرتبط مربوط به هر گزینه نادیده گرفته می شود

همچنین، در سال ۲۰۰۰ [۱۲] انتزاع حالات را در سلسله مراتب انتزاع زمانی بکار برد. وی توسعه ای از چارچوب MAXQ را از طریق انتزاع حالات ارائه داد و نتیجه گرفت که انتزاع حالات برای کاربرد موفق MAXQ اساسی است [۴]. بهرحال این روش نیاز دارد که توسعه دهنده قبل از یادگیری، مجموعه ای از ویژگی های حالت مرتبط را برای هر گزینه تعریف کند. همانطور که پیچیدگی مسئله رشد می کند، کد کردن چنین نمایش های حالتی بسیار مشکل می شود.

۲-۱۱-۳ ادغام از دیدگاه یکپارچگی، دقت و انطباق

روش های مختلفی در رابطه با ادغام در ادبیات وجود دارد و می تواند از ابعاد مختلف مقایسه شود. در این بخش، سه جنبه دیگر از انتزاع مطرح می شود و روش های مختلف در این رابطه بررسی می شود. جنبه اول، یکنواختی[1] است: بر حسب اینکه تمایزات در نظر گرفته شده، در سرتاسر فضای حالت یکسان هستند یا نه. در انتزاع یکنواخت، متغیرها در سراسر فضای حالت، بطور یکپارچه، مرتبط یا نامرتبط در نظر گرفته می شوند. در حالیکه یک انتزاع غیر یکنواخت این امکان را می دهد تا متغیرهای خاصی، تحت شرایط خاص نادیده گرفته شوند و تحت شرایط دیگر نه.

دومین جنبه مقایسه، دقت است. انتزاع دقیق، حالاتی را که ارزش نسبت داده شده به آنها توسط یک تابع ارزش یکسان است، با هم گروه بندی می کند. در حالی که انتزاع تقریبی، حالاتی را با هم گروه بندی می کند که ارزش های مشابه دارند و نه لزوما یکسان. جنبه سوم، تطبیق پذیری است. بر حسب

[1] uniformity

اینکه انتزاع در طول دوره محاسبه قابل تغییر است یانه، می تواند تطبیقی (پویا) یا ثابت باشد. این ویژگی مربوط به خود انتزاع نمی باشد، بلکه مربوط به نحوه استفاده از انتزاع توسط یک الگوریتم خاص است. در یک تکنیک انتزاع تطبیقی، انتزاع می تواند در طول دوره محاسبه تغییر کند و پویا باشد. در حالی کـه در یک انتزاع ثابت، گروه بندی حالات یک بار و برای همه انجام می شود. مثلا از طریق انتزاع تطبیقی، یـک انتزاع در نمایش تابع ارزش V^k انجام می شود، سپس این انتزاع بـرای نمـایش V^{k+1} اصـلاح مـی شـود [۴۴].

ویژگی های ثابت بودن و یکنواختی، در بیشتر موارد نقص هستند. بهرحال، نتیجه گیری کلی نمـی توان انجام داد و بر حسب شرایط مسئله ممکن است متفاوت باشد. مثلا، فرض کنید دو هـدف O1 و O2 برای تابع پاداش تعریف شده است. O1 از O2 مهمتر است اما، O2 نیز ارزشمند است و بهتر است در نظـر گرفته شود. ممکن است کسب هر دو هدف در زمان توسعه یک سیاست ممکن نباشد. در نتیجه، سیاست مربوطه O2 را بطور کامل نادیده می گیرد [۷۷]. بطور کلی، با توجه به سیاست تطبیق یافته ارتباط بـین متغیرها قابل تغییر است. در نتیجه، یک راه این است که ادغام بر اساس سیاست جاری تغییر کند. در این شرایط، روش های انتزاع تطبیقی بهتر عمل می کنند.

همچنین، حذف یکنواخت لیترال ها ممکن است مناسب نباشد. مثلا اگـر A درسـت باشـد، عامـل پاداشی را برای B دریافت می کند. اما در شرایطی که A نادرسـت اسـت، پـاداش دریـافتی از B مسـتقل است. بنابراین در انتزاع یکنواخت، می توان B را از همه جا حذف کرد و تفاوت موجـود در زمـانی کـه A درست است نادیده گرفته می شود. در روش دیگر، می توان تمایز بعد B را در همه جا ایجاد کرد، کـه در شرایطی که A نادرست است نامربوط است. در این شرایط، ادغام غیر یکنواخت می تواند راهکـار مناسـبی باشد [۷۸]. در ادامه، تعدادی از کارهـای انجـام شـده در بررسـی ابعـاد مختلـف انتـزاع از دیـدگاه دقـت، یکنواختی و تطبیق پذیری بررسی خواهد شد.

۲-۱۱-۳-۱ افراز بندی فضای حالت تطبیقی

یک روش برای پیدا کردن یک نمایش هوشمند از فضای حالت بر طبق ساختارش، افراز بندی فضای حالت تطبیقی است. این روش، در برخی شرایط مناسب است، در عین حال مزیت روش های انتزاع غیر تطبیقی این است که سربار کمی را دارند. اغلب روش های انتزاع تطبیقی دارای محدودیت هایی برای محیط های بزرگ و واقعی هستند. روش انتزاع تطبیقی، از یک افراز درشت شروع می شود و بطور تطبیقی، طبق ویژگی های ساختاری، آن را بهبود می دهد. چندین الگوریتم بر این اساس ارائه شده است.

الگوریتم Parti-game [۷۹]، به عنوان یکی از اولین روش های ایجاد خودکار فضای حالت انتزاعی معرفی شده است. در این روش، فضای حالات به سلول هایی تقسیم می شود. در حالی که عامل با محیط تعامل دارد، این سلول ها بطور تدریجی تقسیم می شوند. ضرورت افراز بندی سلول ها، توسط معیار Minimax تعیین می شود. این معیار تشخیص می دهد که آیا رزولوشن جاری حالات، در ناحیه مربوطه مناسب است یا نه. در نتیجه، رزولوشن به نحوی تنظیم می شود که در نزدیکی موانع بالا باشد، و در نواحی باز محیط پایین باشد. این الگوریتم بسیار معروف می باشد و چندین بار بهبود و تطبیق یافته است ([۸۰]، [۸۱]). بهرحال، این روش در کاربردهای عملی دارای محدودیت هایی می باشد. در این الگوریتم، اندازه محیط باید از قبل مشخص باشد.

روش دیگری توسط [۴۹] برای افراز بندی فضای حالت تطبیقی پیشنهاد شده است که سعی می کند تا مدل MDP را مینیمم کند. در این روش، حالتهایی که احتمال گذر مشابهی دارند در یک افراز قرار می گیرند. هر یک از این افراز ها، احتمال انتقال مدل اصلی را حفظ می کند. برای محیط های بزرگ که معمولا پیوسته هستند، بازنمایی توسط این روش پیچیده می شود. همچنین، باید دانش پیشینی درباره مدل سیستم داشته باشیم. در نتیجه، این روش را نمی توان برای الگوریتم های یادگیری مدل-آزاد مانند

یادگیری کیو بکار گرفت. با توجه به اینکه داشتن یک مدل برای کسب نتیجه بهینه مورد نیاز نیست، این مسئله محدود کننده است [۴].

همچنین، روش U-Tree پیوسته [۷۴] تعمیم یافته الگوریتم U-Tree [۷۶] برای فضاهای حالت پیوسته است. در این روش با شروع از یک حالت، حالات بر اساس آزمون های آماری تقسیم می شوند. در [۲۹]، یک الگوریتم ادغام تطبیقی مبتنی بر مدل برای حوزه های قطعی پیشنهاد شده است. الگوریتم ادغام تطبیقی AAA، برای حل آنلاین مسئله یادگیری در حوزه های قطعی با فضای حالت پیوسته عمل می کند. این الگوریتم با حرکت از شبکه های درشت به ریز روی فضای حالت، یک روش ادغام حالت تطبیقی را استفاده می کند. در نواحی مهم فضای حالت، رزولوشن بالا و در سایر نواحی رزولوشن پایین استفاده می شود. این الگوریتم، دارای نرخ های یادگیری چندجمله ای در رابطه با محدوده خطا می باشد.

در سال ۲۰۰۰، یک روش مدل-آزاد برای افراز بندی فضای حالت توسط [۸۲] پیشنهاد گردید. در این روش، برای تعیین اینکه کدام سلول ها تقسیم شوند، مرزهای تصمیم در نظر گرفته می شود. به این ترتیب که اگر دو زیرسلول در یک افراز اقدامات بهینه متفاوتی دارند، عملیات تقسیم انجام شود. بنابراین، نواحی بر اساس انتخاب اقدام بهینه مشابه ایجاد می شوند. روش مدل-آزاد دیگری که برای گسسته سازی فضای حالت، با استفاده از Kd-Treeها، پیشنهاد شده است، kd-Q-learning [۸۳] می باشد. در این روش، چندین رزولوشن متفاوت فضای حالت در طی یادگیری نگه داری می شود. این روش منجر به تسریع فرآیند یادگیری می شود، اما نیاز به نگه داری نمایش کاملی از بهترین سطح انتزاع در حافظه دارد. در نتیجه، باعث می شود که این الگوریتم برای حل مسائل خیلی پیچیده قابل اعمال نباشد.

۲-۱۱-۳-۲ دقت انتزاع

در رابطه با انتزاع حالات، قوانینی که در کارهای مربوطه وجود دارد دقت انتزاع را تحت تـاثیر قـرار می دهد [۱۳]. بویتلیر و همکارانش [۸۴]، الگوریتم برنامه نویسی پویای تصادفی را ارائـه کردنـد. در ایـن روش حالات با توابع پاداش و گذر مشابه، تحت یک سیاست ثابت در یک گروه قرار می گیرنـد. در ادامـه، گیوان و همکارانش [۸۵] شکلی از انتزاع را به نام bisimulation ارائه کردند کـه در آن حـالات بـا توابـع پاداش و گذر یکسان ادغام می شوند. آنها الگوریتم ارائه شده توسط بویتلیر و همکارانش را، به عنوان مورد خاصی از bisimulation بیان کردند. این روش افراز بندی، نمایش فشرده ای را از فضای حالات فراهم می کند. با این حال، bisimulation معیار بسیار سخت گیرانه ای برای ادغام می باشد. بر این اساس، چنـدین تطبیق در این زمینه پیشنهاد شده اند. رویندران و همکارانش [۸۶]، به جای تطابق اقدام سـخت در روش bisimulation، ادغام حالات بر اساس هوموورفیسم مدل را پیشنهاد کردند.

گیوان و همکارانش در [۸۷]، اجرای تقریبی bisimulation را مورد بررسی قرار دادنـد. ایـن روش، نیاز به هم ارزی دقیق را با تعریف یک محدوده کاهش مـی دهـد. بـدین ترتیـب، MDP بطـور کـران دار دقیقی[1] را فراهم می کند. سلسله مراتب MAXQ از Dietterich [۱۲]، در صورتی که توابع گذر و پاداش حالات برای هر سیاست سازگار با سلسله مراتب مشابه باشد، حالات را در زیر وظایف ادغـام مـی کنـد. در روش [۸۸]، حالات بر اساس آزمون های آماری گروه بندی می شود. اگـر حـالات، اقـدام بهینـه مشـابهی داشته باشند ادغام می شوند. با این حال، اگر MDPهای انتزاعی حاصل بـرای یـادگیری در محـیط هـای بزرگ استفاده شوند، ممکن است منجر به سیاست های بهینه نشود [۸۸].

روش های انتزاع با دقت کمتر، نتایج بهتری را در رابطه با کارایی محاسباتی و مسئله تعمیم دارنـد. بهرحال، روی تابع ارزش عملکرد ضعیف تر مرزهای اتلاف را دارند [۸۹]. با توجه بـه مطالـب بیـان شـده،

[1] BMDP

چندین توسعه در رابطه با دقت انتزاع می توان در نظر گرفت. انتزاع می تواند بر اساس هم ارزی دقیق کمیت های خاص انجام شود. توسعه ای دیگر از این کار می تواند با کاستن از دقت انتزاع، و فراهم کردن انتزاع نرم یا تقریب فراهم شود ([۸۷]، [۹۰]، [۹۱]).

۲-۱۱-۳-۳ یکنواختی انتزاع

در رابطه با بعد یکنواختی انتزاع در بخش های قبلی توضیحاتی داده شد و بیان گردید که در بیشتر موارد، بهتر است تا از انتزاعات غیریکنواخت استفاده شود. در این بخش، برخی روش های انتزاع یکنواخت و غیریکنواخت که در ادبیات وجود دارد معرفی می شود. یک روش انتزاع توسط [۹۲]، برای حل مدل های صفبندی تصادفی پیشنهاد شده است که به عنوان فرآیندهای ادغام-جداسازی مطرح شده اند. از این روش برای شتاب محاسبه تابع ارزش، برای یک سیاست ثابت استفاده می شود. در فرآیندهای ادغام-جداسازی، حالات در ابتدا در داخل کلاسترها ادغام می شوند. سپس سیستم معادلاتی حل می شود، یا یک مجموعه از خلاصه سازی ها انجام می شود. سپس، یک مرحله جداسازی برای هر کلاستر در نهایت انجام می شود. این روش انتزاع، یکنواخت و تطبیقی می باشد.

همچنین در [۹۳]، یک روش انتزاع یکنواخت ارائه شده است که در آن متغیرها، بر اساس اهمیت آنها رتبه بندی می شوند. متغیرهایی که دارای اهمیت کمتری در توصیف مسئله هستند، حذف می شوند. در راه حل برای این مسئله، عناصر هدف اصلی نباید حذف شوند از آنجا که گزاره ها در این روش بطور یکنواخت حذف می شوند، این نوع از انتزاع یکنواخت می باشد [۴۴].

در سال ۱۹۹۶[۹۴]، یک طرح تقریبی ارائه شد که از طبیعت ساختار درختی توابع ارزش استفاده می کند. توابع ارزش تقریبی، به شکل ساختار درختی که برگ های آنها دارای محدوده های بالا و پایین

روی تابع ارزش هستند مشخص می شود. در مواقعی که برگ های زیردرخت دارای ارزش خیلی نزدیکی هستند، یا وقتی محدودیت های محاسباتی داریم، حرص انجام می شود. این روش، به عنوان رویکردی غیریکنواخت، تطبیقی و تقریبی می باشد.

در [۹۵]، یک الگوریتم برنامه ریزی تقریبی برای حوزه های تصادفی بزرگ ارائه شده است. این روش، انتزاع غیریکنواخت و خودکار را انجام می دهد که ساختار فضای حالت را بکار می گیرد. در این روش، راه حل های تقریبی با تغییر سطوح انتزاع بدست می آید. انتزاع، با نادیده گرفتن برخی ابعاد در برخی بخش های فضای حالت انجام می شود. در این مقاله، تغییر انتزاع یکنواخت بر اساس سیاست جاری، و احتمال مواجهه با حالات خاص در آینده انجام می شود.

۲-۱۱-۴ تعمیم بر اساس دانش حوزه

روش انتزاع حالت تطبیقی، با در نظر گرفتن مدلی از سیستم یا یک سیاست زیربهینه انجام می گیرد. در عوض، می توان از دانش حوزه برای مشخص کردن یک نمایش حالات مناسب و ارتباطات بین حالات استفاده کرد. در روش ارائه شده در [۹۶]، ارتباطات همسایگی توپولوژیکی فضای حالت در نظر گرفته شده است. همچنین، سیگنال تقویتی از طریق حالات مجاور در محیط منتشر می شود. از این طریق، دانش روی حالات مرتبط به اشتراک گذاشته شود. بهرحال، این روش نیاز به نقشه توپولوژیکی محیطی که عامل در آن عمل می کند دارد. گلابیوس [۹۷] برای بهره گیری از تجربه در بخش های مشابه فضای حالت، نمایش تابع ارزش داخلی را در نظر می گیرد. این روش، نیاز به دانش پیشین در قالب کلاس های از پیش تعریف شده دارد تا نواحی مشابه جهان را تشخیص دهد [۴].

۲-۱۱-۵ انتزاع خاص-وظیفه

در سال ۲۰۰۲[۹۸] نمایش حالات انتزاعی سفارشی، برای یک وظیفه خاص، انجام گرفت. بدین ترتیب، روشی برای هماهنگ کردن اقدامات ربات ها در محیط های چندعامله [۹۹] ارائه گردید. همچنین [۱۰۰] بر اساس دانش مدل، ناوبری ربات را به عنوان نمونه ای از کلاس های POMDP بررسی می کند. الگوریتم ارائه شده، به نام HPOMDP، از انتزاع زمانی و حالات در داخل یک چارچوب سلسله مراتبی بهره می گیرد. انتزاع با استفاده از الگوریتم های EM [۱۰۱]، انجام می شود. روش آنها کارایی خوبی را در مقایسه با الگوریتم های دیگر POMDP ارائه کرد. بهرحال، نیاز به مدل اولیه خوبی از POMDP دارد.

۲-۱۲ نتیجه گیری

در این فصل، در ابتدا ویژگی مارکوف معرفی گردید. سپس، فرآیندهای تصمیم گیری مارکوف به عنوان وظایف یادگیری تقویتی که ویژگی مارکوف دارند معرفی گردید. در ادامه، مروری بر روش های مختلف حل فرآیندهای تصمیم گیری مارکوف انجام شد. سپس یادگیری تقویتی معرفی گردید و بیان شد که این روش یادگیری از طریق سعی و خطا می باشد. عامل خودمختار رفتارش را با استفاده از تعاملات با محیط بهبود می دهد. عامل حالتی از محیط را ادراک می کند، و اقدامی را انتخاب می کند تا حالت محیط را تحت تاثیر قرار دهد. سپس برای هر اقدامی که انتخاب می کند، بازخوردی از محیط دریافت می کند. هدف، ماکزیمم کردن مجموع بازخوردهایی است که در طی زمان دریافت می کند. هدف این مکانیزم، پیدا کردن روشی بهینه برای رسیدن به هدف می باشد. در ادامه بیان گردید که این فرآیند یادگیری، توسط آموزش یا از طریق نظارت انجام نمی شود، بلکه از طریق تعامل با یک محیط دینامیک و غیرقطعی انجام می شود.

یک بخش مهم از الگوریتم یادگیری تقویتی، طرح اکتشاف است که در ادامه بررسی گردید. در وضعیت هایی که مدل MDP در ابتدا ناشناخته است، تنها منبع عامل کاوش در محیط است. نقش اکتشاف، بدست آوردن اطلاعات جدید با انتخاب اقدام مناسب است. این فرآیند، عامل را به سوی حالات ناشناخته MDP هدایت می کند. در هنگام انتخاب اقدامات، مسئله توازن بین کاوش و بهره برداری مطرح می شود. یادگیری تقویتی شامل طیف گسترده ای از تکنیک ها برای حل این مسئله می باشد که به برخی از آنها در این بخش اشاره گردید. همچنین، به چالش های مطرح در یادگیری تقویتی اشاره شد و انتزاع به عنوان رویکردی برای رفع این چالش ها مطرح گردید. در ادامه، انتزاع زمانی، انتزاع حالات و شکل دهی پاداش بررسی گردید. سپس، یادگیری کیفی با استفاده از انتزاع به عنوان توسعه ای از یادگیری تقویتی برای بهبود نرخ همگرایی معرفی گردید. در نهایت، مروری بر کارهای انجام شده در این زمینه انجام شد و از جنبه های مختلف بررسی گردید.

فصل سوم: چارچوب پیشنهادی و بررسی نتایج

۳-۱ مقدمه

در فصل قبل، مروری بر روش های مختلف در رابطه با بهبود فرآیند یادگیری تقویتی انجام گردید. یادگیری تقویتی کیفی، به عنوان راهکاری برای چالش های مطرح در فرآیند یادگیری عنوان گردید. در این بخش چارچوب جدیدی مبتنی بر یادگیری تقویتی کیفی، با هدف رفع چالش های مطرح و تسریع فرایند یادگیری تقویتی پیشنهاد می شود. چارچوب پیشنهادی، به‌طور کاملاً انتزاعی و کلی معرفی می‌شود. هدف این چارچوب به‌طور اخص، اکتشاف هوشمندانه‌تر محیط، کاهش زمان مورد نیاز اکتشاف محیط و یادگیری محیط با سرعت بیشتر خواهد بود.

در ادامه الگوریتم نمونه ای از چارچوب پیشنهادی ارائه می شود. الگوریتم نمونه، روی محیط های محک مختلف ناوبری و غیرناوبری با تعداد حالات مختلف بررسی می شود. برای بررسی آنکه آیا بهبود بدست آمده در الگوریتم پیشنهادی از لحاظ آماری معنی دار است، صحت نتایج با آزمون های فرض آماری بررسی شده است. آزمون های آماری، بهبود بدست آمده با استفاده از الگوریتم پیشنهادی را تایید می کند.

۳-۲ نگاشت محیط به گراف

الگوریتم پیشنهادی در این تحقیق، محیط را به یک گراف نگاشت می کند و بر اساس تحلیل این ساختار داده عمل می کند. همانطور که قبلا اشاره شد، نگاشت فضای حالت عامل به یک گراف سودمند است. از این طریق، می‌توان از طیف گسترده الگوریتم‌های گراف‌کاوی و نظریه گراف‌ها برای دستیابی به مقداری دانش اولیه از محیط استفاده کرد. الگوریتم نمونه چارچوب پیشنهادی، از تحلیل گراف در بخش دوم چارچوب استفاده می کند، تا مقداری دانش اولیه از فضای حالت برای عامل فراهم شود [۸]. تاثیر

نهایی آن، در تسریع کاوش محیط است. در نتیجه، باعث سرعت در فرآیند یادگیری و پیدا کردن یک سیاست بهینه می‌گردد. همچنین گراف فضای حالت، نمایش بهتری را در محیط‌های پویا فراهم می‌کند.

قبل از اینکه بخش اصلی الگوریتم پیشنهادی شروع شود، محیط باید به گراف مربوطه نگاشت شود. فضای حالت عامل، به یک گراف بدون وزن و غیرجهت دار G(V,E) نگاشت می شود. در این گراف، V مجموعه ای از رئوس در گراف است (حالات سیستم)، و E مجموعه ای از لبه ها است (گذر بین حالات). هر گذر $v \to v'$ در تابع انتقال، یک لبه در گراف را نمایش می دهد [۱۱]. در نهایت، الگوریتم پیشنهادی حالات را به حالات انتزاعی گروه بندی می کند که اندازه محیط را کاهش می دهد.

۳-۳ خوشه بندی

خوشه بندی[1]، داده ها را به گروهی از اشیای مشابه دسته بندی می کند [۱۰۲]. هر گروه شامل مجموعه ای از اشیا است که بیشترین شباهت را به اشیای درون خوشه[2] دارند، و ناهمسان با اشیای خارج خوشه[3] هستند. در چارچوب پیشنهادی، خوشه بندی به عنوان یکی از مراحل برای انتزاع حالات استفاده شده است. روش خوشه بندی مورد استفاده، الگوریتم EM می باشد که روال انجام آن به صورت شبه کد در پیوست ۱ نمایش داده شده است.

[1] Clustering
[2] Intra-cluster
[3] Inter-cluster

۳-۴ چارچوب پیشنهادی

در این پایان نامه، چارچوب جدیدی پیشنهاد می‌شود که هدف آن تسریع فرایند یادگیری تقویتی است. در این بخش، چارچوب پیشنهادی به‌طور کاملاً انتزاعی و کلی معرفی می‌شود. این چارچوب بر اساس یادگیری کیفی [۱۴] و تخمین پاداش ساختگی [۶] می باشد تا از فواید هر دو روش استفاده کند. در نتیجه، تا آنجا که ممکن است نرخ همگرایی بالاتری را فراهم می کند. یادگیری کیفی یک روش بر اساس تزریق دانش اولیه برای تعیین مقادیر کیفی پایه برای کاوش حالات است. چارچوب پیشنهادی یادگیری تقویتی، با روش کیفیت دادن به محیط، پنج بخش دارد:

۱. ابتدا محیط ورودی به یک **مدل ریاضی** تبدیل می‌شود. این مدل ریاضی، به عامل کمک می‌کند تا تحلیل اولیه و ایستایی روی محیط انجام دهد. تحلیل ایستا، همچنین شناخت اولیه‌ای از محیط برای عامل فراهم می‌سازد.

۲. با کمک یکی از **انواع الگوریتم‌های** موجود روی مدل ریاضی منتخب، تحلیلی روی محیط صورت می‌گیرد. از این طریق، دانش اولیه ای فراهم می شود که با تزریق در وضعیت‌ها به اکتشاف محیط کمک می کند.

۳. در مرحله سوم، یکی از **الگوریتم‌های یادگیری تقویتی** که مناسب محیط ورودی است استفاده می شود. عامل در چند اپیزود اول به یادگیری محیط می‌پردازد و دانش اولیه‌ای کسب می‌شود.

۴. سپس، نیاز به **الگوریتمی برای انتزاع حالات** می باشد تا محیط کاهش یافته ای را بدست آوریم و از این طریق منجر به کاهش اکتشاف می شود. این الگوریتم، باید ویژگی‌های محیط اصلی را حفظ نماید. از انواع الگوریتم‌های یادگیری ماشینی و داده کاوی می‌توان در این بخش بهره برد.

۵. در بخش آخر بعد از انتزاع حالات، **الگوریتم** به ادامه یادگیری می‌پردازد. یادگیری تا رسیدن به سیاست بهینه ادامه پیدا می‌کند. الگوریتم این بخش لزوماً با الگوریتم مرحله سوم یکی نیست.

از دیدگاه نظری، بکارگیری همه بخش‌های این چارچوب ضرورت ندارد. بهرحال، برای افزایش سودمندی چارچوب پیشنهادی، بهتر است همه اجزای آن در هر الگوریتم نمونه وجود داشته باشند. چنانچه طراح بخواهد کارایی الگوریتم خودش را بالا نگه دارد، می‌تواند از الگوریتم‌هایی با مرتبه اجرایی کم در هر بخش استفاده نماید.

۳-۵نمونه‌سازی و الگوریتم پیشنهادی

به‌منظور عینیت بخشیدن به چارچوب پیشنهادی و نشان دادن سودمندی آن، الگوریتم جدیدی بر مبنای چارچوب پیشنهادی ارائه می‌شود. به‌عبارت بهتر، الگوریتم نمونه‌ای ساخته می‌شود. در بخش سوم و پنجم چارچوب پیشنهادی، از یادگیری کیو بهره برده شده است. مدل ریاضی بکار رفته در بخش اول چارچوب پیشنهادی گراف است. همچنین از طریق تحلیل گراف در بخش دوم چارچوب پیشنهادی، تابع پاداشی با روش [۸] تخمین زده می شود. خوشه‌بندی نیز برای خلاصه‌سازی وضعیت‌ها و انتزاع حالات مورد استفاده قرار گرفته است. این فرآیند منجر به کاهش تعداد حالات محیط و فراهم کردن مقداری دانش اولیه برای عامل می شود. الگوریتم مربوط به روش شکل دهی پاداش در پیوست ۱ بطور مفصل آمده است.

قبل از اینکه بخش اصلی الگوریتم پیشنهادی را شروع کنیم، محیط باید به گراف مربوطه نگاشت شود. علاوه بر این، به منظور فراهم کردن یک دانش اولیه برای تابع کیو، و بدست آوردن محیطی با تابع پاداش آگاه، الگوریتم‌های کاوش گراف استفاده می شود. این الگوریتم ها به گراف فضای حالت اعمال

میشود تا چنین اطلاعاتی بطور خودکار بدست آید. در نهایت، الگوریتم پیشنهادی حالات را به حالات انتزاعی گروه بندی می‌کند که اندازه محیط را کاهش می‌دهد. شبه کد الگوریتم پیشنهادی در شکل ۳-۱ نشان داده شده است که مراحل آن به صورت زیر است:

مرحله اول: برخی تحلیل های اولیه (پیش از اکتشاف) انجام می شود که اهداف میانی و حالات بد شناسایی می شوند و پاداش ساختگی مناسبی برای حالات در محیط تزریق می شود.

الگوریتم استفاده شده در این بخش، به حالات محیط بر اساس اهمیت آنها پاداشی اختصاص می دهد. اهداف میانی بر اساس الگوریتم های مرکزیت میانگی گراف [۱۰۳] شناسایی می شوند و پاداش بیشتری به آنها نسبت داده می شود. تعیین حالات بد بر اساس ماتریس مجاورت گراف صورت می گیرد. این حالات در ماتریس مجاورت گراف، دارای کمترین تعداد همسایگی هستند. برای این حالات، پاداش مجازی کمتری نسبت داده می شود. در مرحله سوم، با استفاده از روش های مرکزیت میانگی زیرمحیط های گراف شناسایی می شود و برای هر حالت بر اساس تراکم در زیرناحیه آن پاداشی اختصاص می یابد.

مرحله دوم: عامل یک الگوریتم ساده یادگیری تقویتی، مانند یادگیری Q، را در اپیزودهای اولیه یادگیری استفاده می‌کند. این فرآیند منجر به کسب مقادیر اکتشافی محیط می‌شود که در حال حاضر، مثلاً، در تابع Q قرار گرفته‌اند. دقت شود که معادله یادگیری شامل پاداش ساختگی نیز خواهد بود.

مرحله سوم: نگاشت گراف محیط برای دسته‌بندی فضای حالت، طبق مجاورت حالات و زیر محیط‌ها مورد استفاده قرار می‌گیرد. این‌کار، یک دسته‌بندی اولیه از حالات می‌دهد که در آن حالات با مجاورت‌های مشابه در دسته یکسانی قرار می‌گیرند.

توجه داشته باشید که در نظر گرفتن مجاورت‌های حالات، برای این محاسبه کافی نیست. برای درک بهتر، فرض کنید که هر عامل، واقع در یک حالت خاص، حداکثر چهار جهت ممکن (یا عمل) برای انتخاب داشته باشد. پس عامل می‌تواند به بالا، پایین، چپ و راست حرکت کند. در این محیط، ممکن است عمل‌های بسیاری غیر مجاز باشد. مثلا، عامل از حالت s قادر به حرکت به سمت بالا نخواهد بود، زیرا این عمل غیر مجاز است (مسدود شده است، یا یک دیوار در این جهت وجود دارد). در این موقعیت، تنها سه عمل ممکن برای انتخاب وجود دارد. اما، در حالت دیگر با سه عمل ممکن که در آن جهت راست مسدود شده است (و غیر مجاز)، وضعیت متفاوتی حاکم است. فرآیند دسته‌بندی اولیه، باید این وضعیت‌های متفاوت را در نظر بگیرد. علاوه بر این، زیرمحیط‌های متفاوتی در یک محیط وجود دارد (چنین وضعیت‌هایی در محیط‌های آزمایشی در بخش ارزیابی نشان داده شده‌اند). فرآیند دسته‌بندی اولیه، این نکته مهم را نیز باید در نظر بگیرد.

بعد از این طبقه‌بندی، هر گروه از حالات شامل حالاتی با عمل‌های مجاز مشابه است. در مرحله بعدی، از خوشه‌بندی برای شناسایی حالات مشابه در هر دسته - بر اساس مقادیر هر عمل- استفاده می شود. برای این منظور، هر خوشه به‌عنوان یک حالت انتزاعی جدید در نظر گرفته می‌شود. Q-value های این حالات، بدست آمده از اکتشافات اولیه، ممکن است میانگین‌گیری شود و در حالت انتزاعی جدید قرار گیرد. البته در آزمایشات مقادیر دیگری مانند میانگین کوچکترین و بزرگترین داده هر خوشه، بزرگترین و کوچکترین مقدار درون خوشه، داده ای با بالاترین فراوانی در خوشه و مانند آن هم بررسی گردید. بهرحال، میانگین مقادیر همه حالات درون خوشه نتایج بهتری را داشت. حالات اکنون بطور خودکار ادغام شده‌اند، که یک نمایش فشرده از محیط را نشان می‌دهد. کاهش در اندازه محیط، همراه با یک Q-value اولیه، باعث سرعت قابل توجه در فرآیند یادگیری و همگرایی به سیاست بهینه می‌گردد.

مرحله چهارم: یادگیری در محیط کاهش یافته جدید ادامه می‌یابد تا یک سیاست بهینه حاصل شود.

بطور خلاصه، با فرض یک محیط برای عامل خودمختار کیفی سازی محیط به این صورت انجام می شود (۱) عامل برای تعدادی اپیزود محدود شروع به کاوش محیط، با استفاده از یک روش مانند یادگیری کیو می‌کند، (۲) حالات بر اساس موقعیت زیر محیط‌های مشابه دسته‌بندی می‌شوند، (۳) هر دسته مجدداً بر اساس عمل‌های مجاز و غیر مجاز دسته‌بندی می‌شود، (۴) علاوه بر این، دسته‌های حاصل بر اساس شباهت Q-valueها خوشه‌بندی می‌شوند. (۵) حالات هر خوشه در یک حالت انتزاعی ادغام می‌شوند که یک محیط کوچکتر جدید را حاصل می‌کند و (۶) یادگیری در محیط کاهش یافته جدید ادامه می‌یابد تا یک سیاست بهینه حاصل شود.

```
input:
  Q(s, a): stores the agent's knowledge gained from exploring the environment. It is initialized according to
           legal and illegal actions in a given environment.
  reward(s, a): stores the rewards obtained from the environment when moving from a state and an action.
           It is initialized according to legal and illegal actions in a given environment.
  manualReward(s, a): stores the rewards estimated by some pre-exploration analyses on the environment.
  next(s, a): indicates the resulting state when going from state s and action a. it is initialized according to a
           given environment.
  goal: the final state in which simulation is stopped.
  episodeToSelectGreedily: the episode after which the next action is one with highest Q value; It ranges from
           0 to max episode number.
  episodeToMergeStates: a variable indicating the episode after which the states are merged.
  Modify(manualReward): A function that finds sub goals and bad states automatically, and then modifies
           the manualReward matrix according to these special states.

output:
  r: the final value of the reward obtained during execution of each episode of the simulation

algorithm QLearningWithQualityAndManualReward(Q, reward, next) begin
STEP 1:   Initialize manualReward(s, a) with reward(s, a);
  Modify(manualReward);
  episodeToMergeStates:= a value between 1 to max episode;
  for each episode e do begin
    r:= 0.0;
    if e < episodeToMergeStates then
STEP 2:       r:= RunSingleEpisodeForPrimaryEnvironment(e);
    else
STEP 3:       if e == episodeToMergeStates then begin
      categories:= {};
      categories:= CategorizeAccordingToLegalActions();
      categories:= CategorizeAccordingToAdjacencies(categories);
      clusters:= Perform cluster analysis on the states in each category of the categories using a clustering
           algorithm (e.g., EM, K-Means, etc.), and extract final clusters;
      maxStates:= Max cardinality of clusters;
      Generate newQ, newReward, and newNext of the size (maxStates, number of actions);
      newStates:= {};
      for each cluster in clusters do begin
        Consider the pair (newState, mergedstates), where newState is a unique number and mergedstates
        are similar state grouped in that cluster;
        Add the new pair to newStates;
      endfor
      FillNewNextForNewEnvironment(newStates, newNext);
      FillNewRewardAndNewQForNewEnvironment(newReward, newQ, newNext, newStates);
      SetNewGoal(newStates);
    endif
STEP 4:       r:= RunSingleEpisodeForNewEnvironment(e, newQ, newReward, newNext);
    endif
  endfor
end QLearningWithQualityAndManualReward
```

شکل ۳-۱: شبه کد الگوریتم پیشنهادی

۳-۶ تحلیل پیچیدگی محاسباتی الگوریتم

منظور از پیچیدگی زمانی، تعداد واحدهای زمانی است که الگوریتم نیاز دارد تا سیاست بهینه را پیدا کند. در مقاله [۱۰۴]، به تحلیل پیچیدگی الگوریتم های یادگیری تقویتی آنلاین، یادگیری کیو و تکرار ارزش پرداخته است. در کارهای قبلی، ثابت شده بود که در بسیاری از موارد یادگیری تقویتی به شکل لوح سفید برای چنین مسائلی نمایی است. تنها در صورتیکه، الگوریتم یادگیری تقویت شده بود قابل مهار است. این مقاله، نشان داده است که با تغییراتی در نمایش وظیفه یا مقداردهی اولیه می تواند قابل مهار باشد. در این مقاله بیان شده است که با در نظر گرفتن شرایطی، بدترین حالت پیچیدگی رسیدن به هدف یک مرز محکم $O(S^3)$ را برای Q-learning دارد. این مقدار، می تواند تا $O(S^2)$ کاهش یابد که S تعداد حالات محیط می باشد.

در این بخش، پیچیدگی زمانی بخش های مختلف الگوریتم در چارچوب پیشنهادی بررسی می شود. پیچیدگی محاسباتی الگوریتم پاداش ساختگی برابر O(ES) می باشد که در آن E تعداد یالهای گراف و S تعداد گره ها می باشد. با توجه به اینکه گراف ها در یادگیری تقویتی اغلب تنک هستند، این پیچیدگی برابر $O(S^2)$ می باشد. پیچیدگی الگوریتم دسته بندی حالات بر اساس اقدامات قانونی برابر $O(S^2a)$ می باشد. پس از آن، دسته بندی حالات بر اساس مجاورت حالات با مرتبه $O(S^2)$ و خوشه بندی حالات با مرتبه $O(S^2a^2i)$ انجام می شود. در اینجا، a به تعداد اقدامات اشاره می کند و i تعداد تکرارهای الگوریتم خوشه بندی است. پس از آن، عملیات تعریف محیط جدید بعد از ادغام حالات از مرتبه $O(S^2a)$ انجام می شود. همچنین، اجرای اپیزودهای یادگیری از مرتبه O(eta) می باشد. پارامتر e تعداد اپیزودها و t آستانه در هر اپیزود می باشد. به این ترتیب، پیچیدگی زمانی الگوریتم پیشنهادی از مرتبه $O(S^2)$ است.

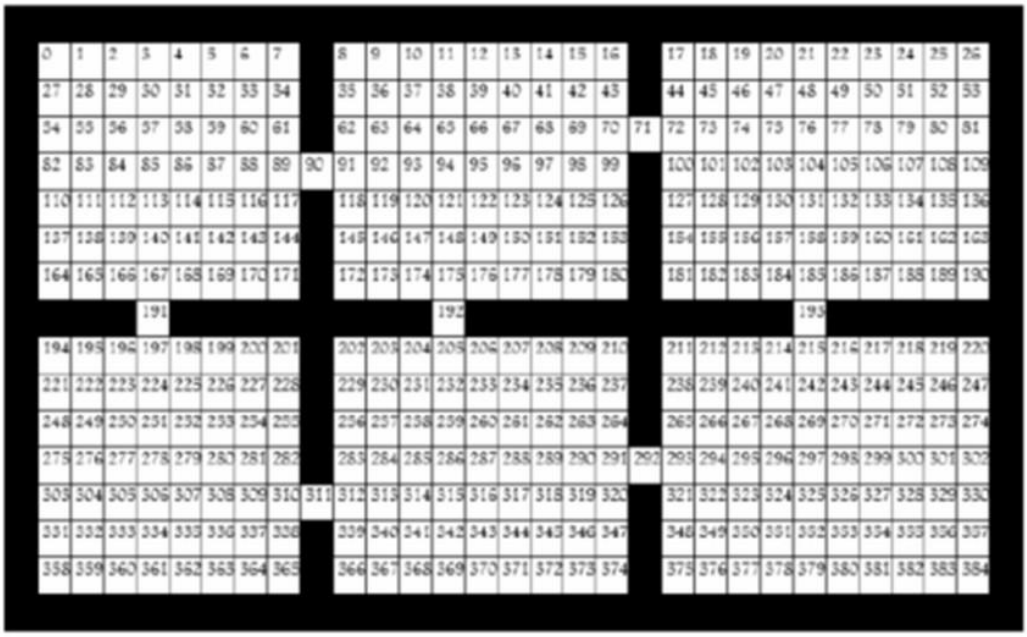
۳-۷محیط های آزمایشگاهی

به منظور ارزیابی چارچوب پیشنهادی، آزمایشاتی در سه محیط محک با تعداد حالات مختلف انجام گردید. این محیط ها شامل محیط مشبک شش اتاقه، Complex Maze و برج هانوی با سه میله و سه دیسک می باشد. محیط های مشبک شش اتاقه و Complex Maze محیط های ناوبری هستند درحالیکه محیط برج هانوی یک محیط غیرناوبری می باشد. در ادامه، هر یک از این محیط ها به تفصیل معرفی خواهد شد.

۳-۷-۱ محیط ۶ اتاقه

محیط مشبک[1]، یک محیط معیار برای ارزیابی الگوریتم های یادگیری تقویتی است. اولین بار، محیط مشبک چهار اتاقه توسط ساتن ارائه گردید [۱۰]. محیط مشبک ۶ اتاقه، در شکل ۳-۲ نشان داده شده است. این محیط، شامل ۶ اتاق با اندازه های تقریبا برابر است که توسط ۷ درب به هم متصل هستند که یک محیط با ۳۸۵ حالت را تشکیل می دهد. عامل، باید هدف را که در یکی از خانه های محیط بطور تصادفی قرار داده شده است پیدا کند. مکان شروع عامل نیز بطور تصادفی مشخص می شود. چهار کنش پایه ای در این محیط "حرکت به بالا"، "حرکت به پایین"، "حرکت به چپ" و "حرکت به راست" می باشد. در ازای انجام هر حرکت، عامل پاداش تعریف شده ای را از محیط دریافت می کند.

[1] grideworld



شکل ۳-۲: محیط مشبک ۶ اتاقه

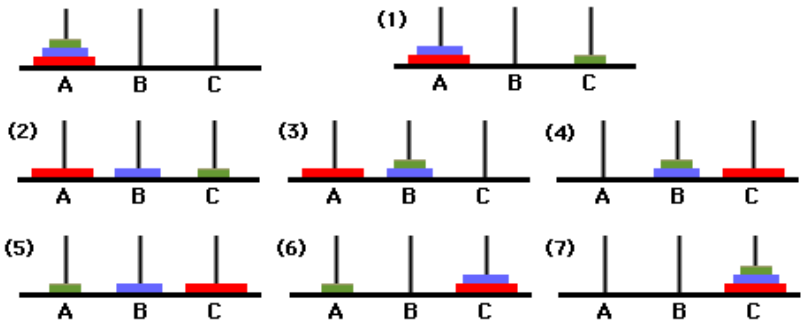
۳-۷-۲ برج های هانوی

محیط برج هانوی در شکل ۳-۳ نمایش داده شده است. این محیط، یک محیط غیر ناوبری است که از سه میله و تعدادی دیسک در اندازه‌های متفاوت تشکیل شده‌است. یکی از میله‌ها میله مبدا (A) ، دیگری میله کمکی (B) و سومی میله مقصد (C) است. هدف، انتقال تمام دیسک‌ها توسط عامل از میله مبدا به میله مقصد با رعایت شرایط زیر است:

- در هر زمان، فقط یک دیسک را می‌توان جابجا نمود.
- نباید در هیچ زمانی، دیسکی بر روی دیسک با اندازه کوچکتر قرار بگیرد.

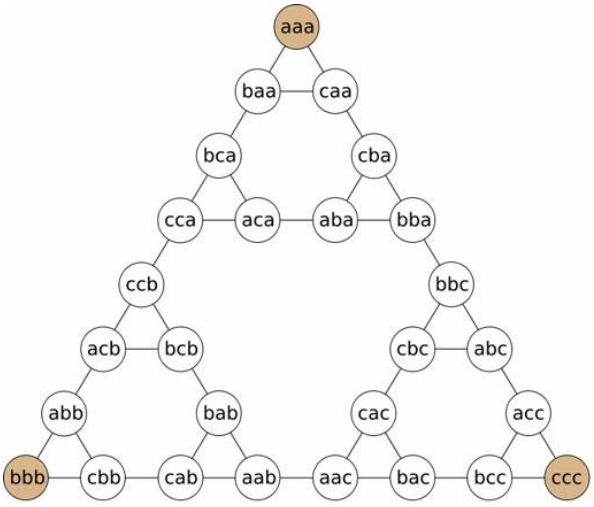
در این بررسی، محیط برج هانوی شامل سه دیسک می باشد که با توجه به قوانین در انتقال دیسک ها، تعداد ۲۷ حالت مجاز تعریف می شود. تعدادی از این وضعیت های مجاز در شکل ۳-۳ نمایش

داده شده است. در صورت رسیدن به هدف نهایی، انتقال همه دیسک ها، عامل پـاداش ۱۰۰۰ را دریافـت می کند و در سایر موارد یک واحد جریمه می شود.



شکل ۳-۳: حالت های مختلف در برج هانوی با سه میله و سه دیسک

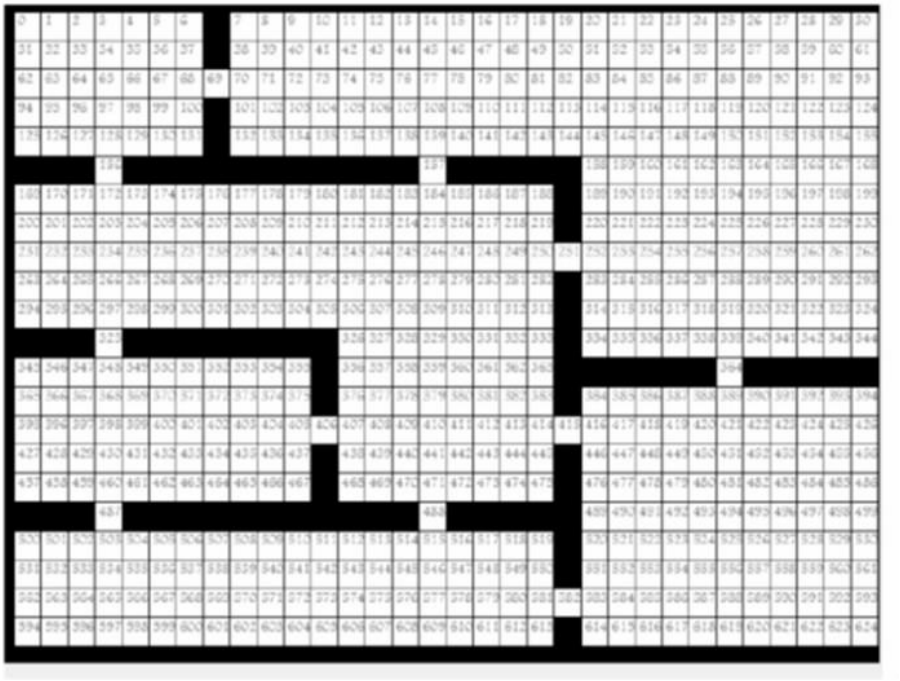
این محیط را می توان توسط یک گراف غیرجهتدار نمایش داد. در این گراف، گره ها توزیع دیسـک هـا را نشان می دهند و لبه ها نشان دهنده حرکات هستند. برای سه دیسک، گراف حاصل به صورت شکل ۳-۴ می باشد. در این شکل، A، B و C نشان دهنده میله ها هستند، و لیست دیسک ها از چپ بـه راسـت بـه ترتیب افزایش اندازه می باشد.



شکل ۳-۴: گراف متناظر حالات مختلف برج هانوی

۳-۷-۳ محیط Complex Maze

همانطور که در شکل ۳-۵ مشاهده می شود، Complex Maze محیطی ناوبری شامل ۶۲۵ حالت است. در اینجا هم مانند محیط مشبک شش اتاقه، در هر حالت چهار حرکت مجاز "حرکت به بالا"، "حرکت به پایین"، "حرکت به چپ" و "حرکت به راست" وجود دارد. اما، نسبت به محیط مشبک شش اتاقه محیط پیچیده تری می باشد.



شکل ۳-۵: محیط Complex Maze

۳-۸ مطالعه‌های تجربی و ارزیابی

در ابتدا الگوریتم کیفی پیشنهادی، بدون فراهم کردن دانش اولیه ناشی از تحلیل گراف، مورد بررسی قرار گرفته است تا تاثیر کیفی سازی محیط بررسی شود. این الگوریتم بر روی محیط های محک مختلف شبیه سازی شده است و نمودارهای نتایج مقایسه ای آن آورده شده است. در ادامه، الگوریتم

نمونه بطور کامل (به همراه استفاده از پاداش ساختگی) روی محیط محک شش اتاقه پیاده سازی شده که نتایج قبلی را بهبود بخشیده است. هدف از این آزمایش‌ها، نشان دادن بهبود الگوریتم پیشنهادی (و چارچوب پیشنهادی) نسبت به الگوریتم های کلاسیک یادگیری تقویتی مانند Q-Learning می باشد. همچنین تاثیر تزریق پاداش ساختگی، همراه با کیفی سازی محیط بررسی شده است.

در آزمایشات، مقادیر زیر استفاده شدند: به‌منظور ایجاد یک توازن، برای ۵۰ اپیزود اول، کاوش تصادفی استفاده شد و سپس، برای باقیمانده اپیزودها سیاست ε-greedy [۲۱] استفاده گردید. برای ε مقدار ۰/۱ در نظر گرفته شده است. نرخ یادگیری α هم ۰/۵ تنظیم شده است و برای فاکتور تنزیل γ مقدار ۰/۹ استفاده گردید. علاوه بر این، تعداد آستانه عمل‌های هر اپیزود برابر ۵۰۰ و σ نیز ۰/۹ مقداردهی شده‌اند. همچنین، هر آزمایش برای ۲۰۰۰ اپیزود اجرا شده است و تعداد دفعات اجرای الگوریتم روی محیط هم برابر با ۱۰ بار اجرا می باشد. عامل پاداش ۱۰۰۰ را در حالت هدف دریافت می‌کند. پاداش ۱- برای انتخاب عمل‌های غیر مجاز، و پاداش صفر برای حالات دیگر در نظر گرفته شده‌اند. مقدار پاداش و جریمه برای اهداف میانی و حالت های کم اهمیت پاداش ساختگی یک درصد پاداش هدف در نظر گرفته شده است. این پاداش برای اهداف میانی، مثبت و برای حالت های کم اهمیت منفی می باشد.

۳-۹ نتایج شبیه سازی

نتایج آزمایش‌ها برای الگوریتم کیفی در شکل ۳-۶، شکل ۳-۷ و شکل ۳-۹ به‌ترتیب برای محیط‌های مشبک ۶ اتاقه، Complex Maze و برج های هانوی نشان داده شده‌اند. بخش الف برای هر شکل، میانگین پاداش در هر اپیزود که در هر ۲۰ اپیزود متوالی میانگین گرفته شده است، مقایسه

می‌کند. همچنین برای وضوح بیشتر نمودارها، مقادیر نهایی پاداش ها بر ۲۰۰ تقسیم شده است. برای خوانایی بهتر نتایج، بخش ب از هر شکل میانگین پاداش تجمعی را نشان می‌دهد. نتایج نشان می دهد که استفاده از استراتژی یادگیری کیفی، بهبود عملکرد قابل توجهی با شتاب سرعت یادگیری و نرخ همگرایی، نسبت به روش یادگیری کیو دارد.

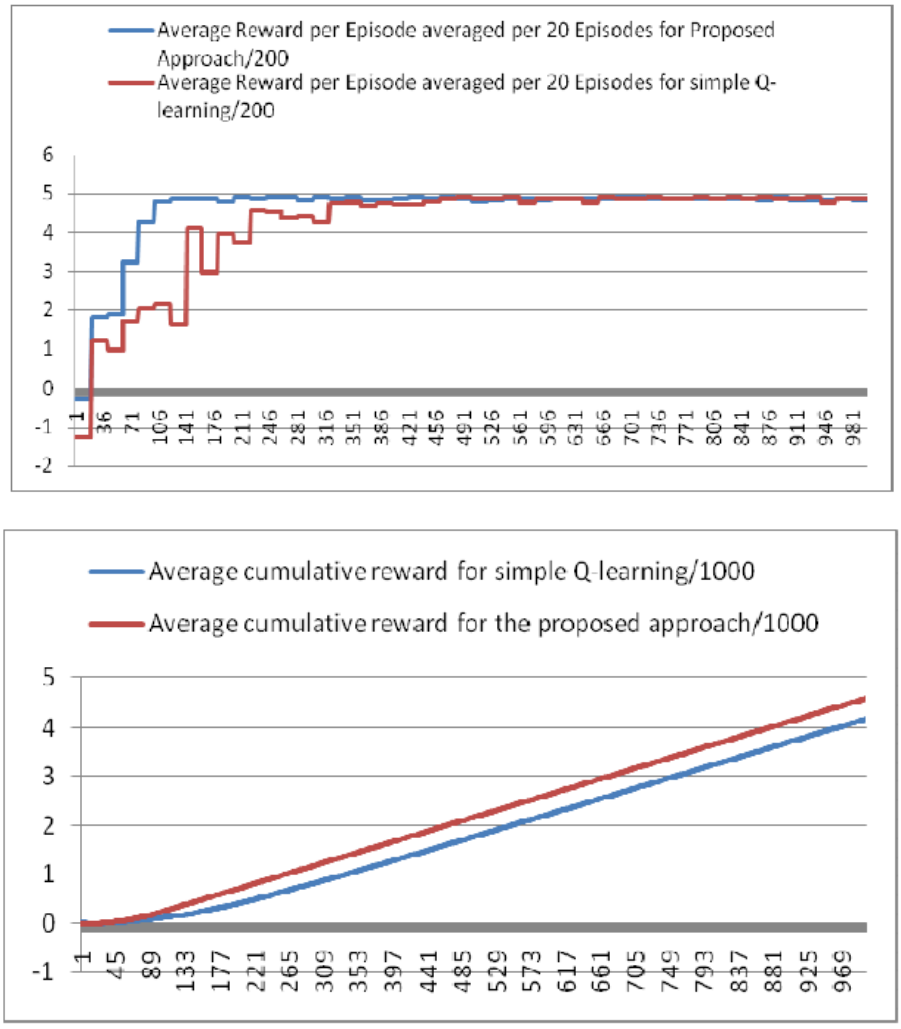
برای تعیین اینکه آیا بهبود بدست آمده توسط الگوریتم پیشنهادی تصادفی نیست و از نظر آماری معنی دار است، از یک روش آزمون آماری، به نام آزمون فرض برای تفاضل دو میانگین [۱۰۵]، استفاده شده است. این روش، آماره z را از نتایج بدست آمده محاسبه می‌کند. سپس، با مراجعه به یک جدول مقادیر بحرانی، مقدار احتمال و درصد اطمینانی که می‌توان فرض صفر را رد کرد، بدست می آید. آماره z با استفاده از معادله (۳-۱) محاسبه می‌شود.

$$\frac{\bar{x}_1 - \bar{x}_2 - \delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \qquad (۳-۱)$$

که در آن x_1 و x_2 مقادیر میانگین نتایج هستند، s_1 و s_2 انحراف معیار استاندارد مربوطه، و n_1 و n_2 تعداد داده‌ها در هر مجموعه از نتایج هستند. چون هر اپیزود یک مقدار نهایی به عنوان خروجی نتیجه می دهد، n_1 و n_2 برابر تعداد اپیزودها در نظر گرفته می‌شوند. δ هم با توجه به اینکه فرض صفر در اینجا این است که تفاوتی بین دو میانگین وجود ندارد، مقدار آن را صفر در نظر می گیریم.

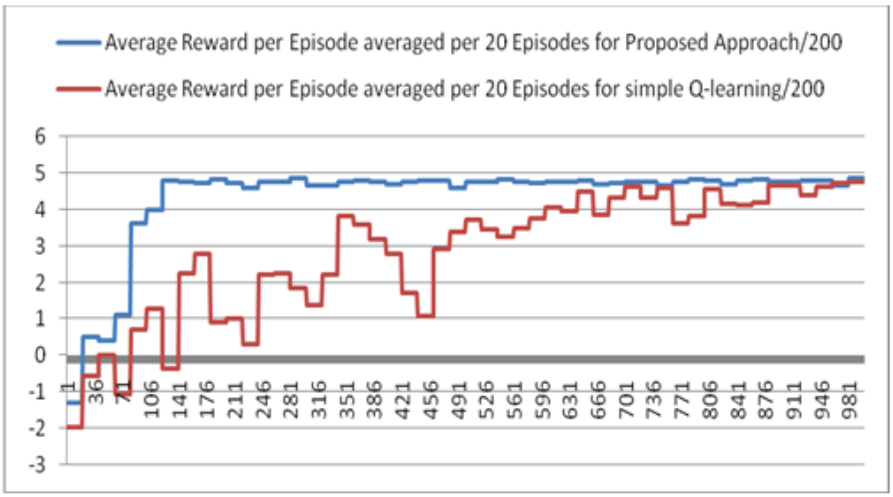
فرض صفر، این‌طور در نظر گرفته شده است که هیچ تفاوت آماری بین نتایج الگوریتم پیشنهادی و یادگیری کیو وجود ندارد. به‌عنوان مرجع، جدولی از مقادیر بحرانی ارائه شده در [۱۰۵] مورد استفاده قرار گرفته است. پس از محاسبه، ارزش z برای محیط‌های شش اتاقه و Complex Maze به ترتیب ۷/۷۲، ۲۱/۷۰ بدست آمد. برای این مقادیر، مشخص است که می‌توان فرض صفر را با اطمینان ۹۹/۹۹ درصد رد

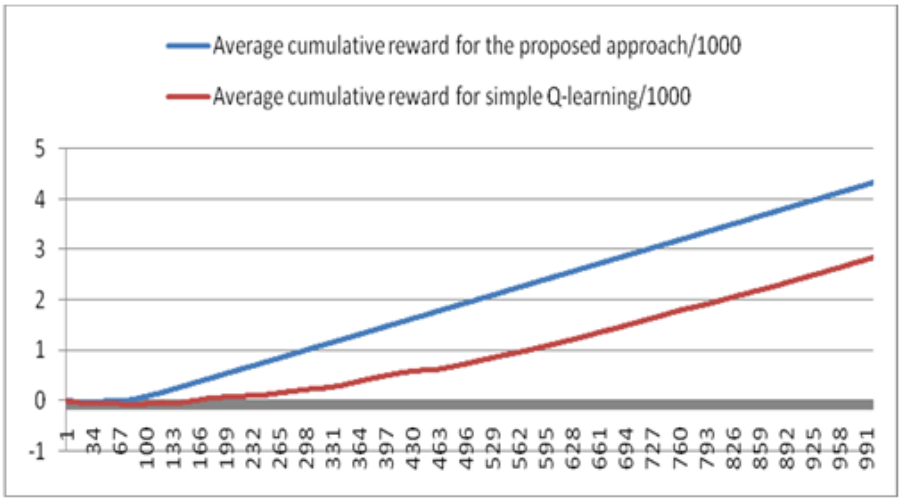
کرد. همچنین ارزش z برای محیط برج های هانوی، با احتمال ۹۲/۲ فرض صفر را رد می کند. رد فـرض صفر نشان می‌دهد که تفاوت در مقادیر میانگین هر دو الگوریتم تصادفی نیست و از لحاظ آماری معنی دار است.



شکل ۳-۶: نتایج آزمایش روی محیط ۶ اتاقه.

میانگین نتایج در هر اپیزود (بالا)، میانگین پاداش تجمعی (پایین)



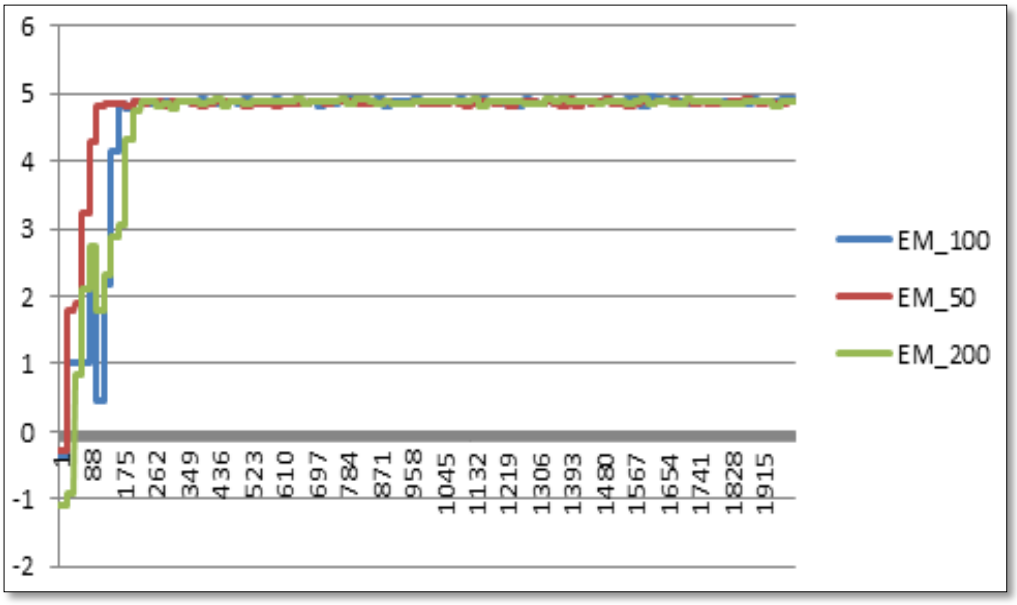


شکل ۳-۷: نتایج آزمایش روی محیط Complex Maze

میانگین نتایج در هر اپیزود (بالا)، میانگین پاداش تجمعی (پایین)

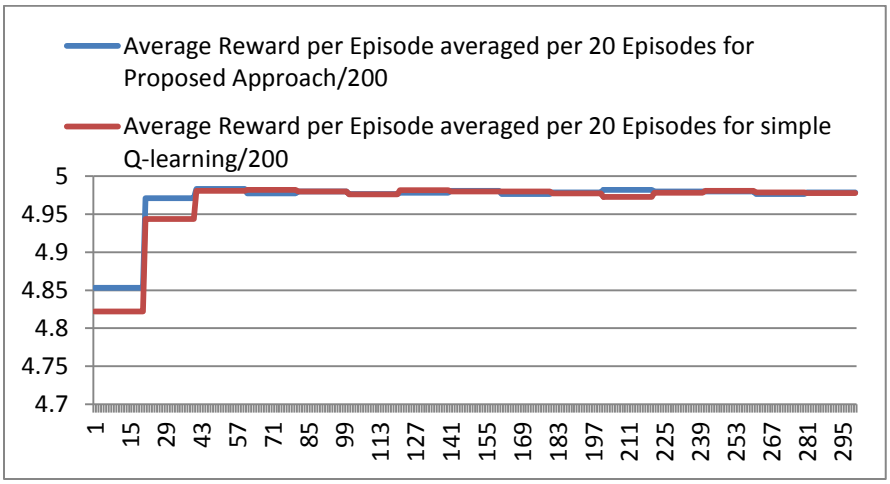
یکی از پارامترهایی که در نتایج تاثیر گذار است، زمانی است که خوشه بندی وضعیت ها انجام می گیرید. فرآیند خوشه بندی، نباید در اپیزودهای اولیه که هنوز عامل شناختی نسبت به وضعیت ها ندارد انجام گیرد. همچنین، انجام عمل خوشه بندی در اپیزودهای خیلی بالا منجر به کاهش تاثیر الگوریتم در تعداد اپیزود مناسب خواهد شد. بنابراین، در آزمایشات به بررسی این وجهه نیز پرداخته شده است. به

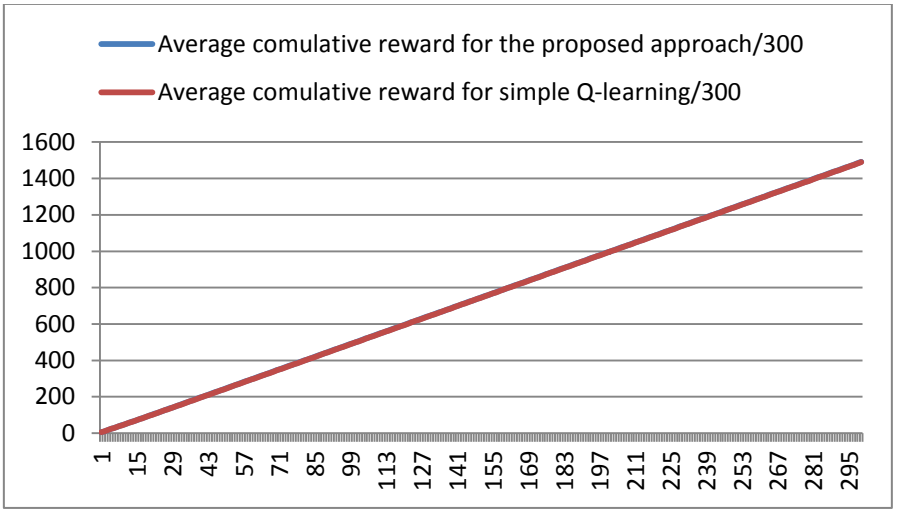
عنوان نمونه، برای محیط مشبک شش اتاقه شکل ۳-۸ نتایج اجـرای خوشـه بنـدی در تعـداد اپیزودهـای مختلف را نشان می دهد.



شکل ۳-۸: نتایج اجرای خوشه بندی در محیط مشبک شش اتاقه با تعداد اپیزود مختلف

محیط برج هانوی نسبت به محیط های دیگر مورد بررسی، کوچکتر است و در تعداد اپیزود کمتری همگرا می شود. به همین دلیل، نتایج نمودارها تنها برای ۳۰۰ اپیزود اول رسم شده است. همچنین، ادغام حالات بعد از ۱۰ اپیزود اول انجام گرفته است. نتایج شبیه سازی ایـن محـیط در شـکل ۳-۹ نشـان داده شده است. با توجه به اینکه محیط برج های هانوی نسبت به دو محـیط دیگـر کـوچکتر اسـت، نتـایج آن بهبود کمتری را نسبت به الگوریتم های کلاسیک یادگیری تقویتی نشان می دهد. بهرحـال، نتـایج در دو محیط دیگر که بزرگتر هستند بطور قابل توجهی بهبود یافته است.

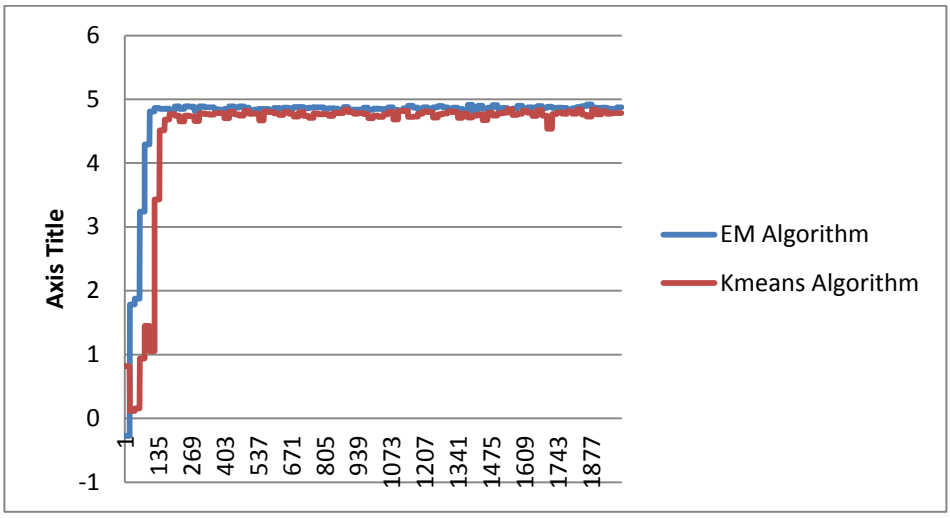




شکل ۳-۹: نتایج آزمایش روی محیط برج های هانوی.

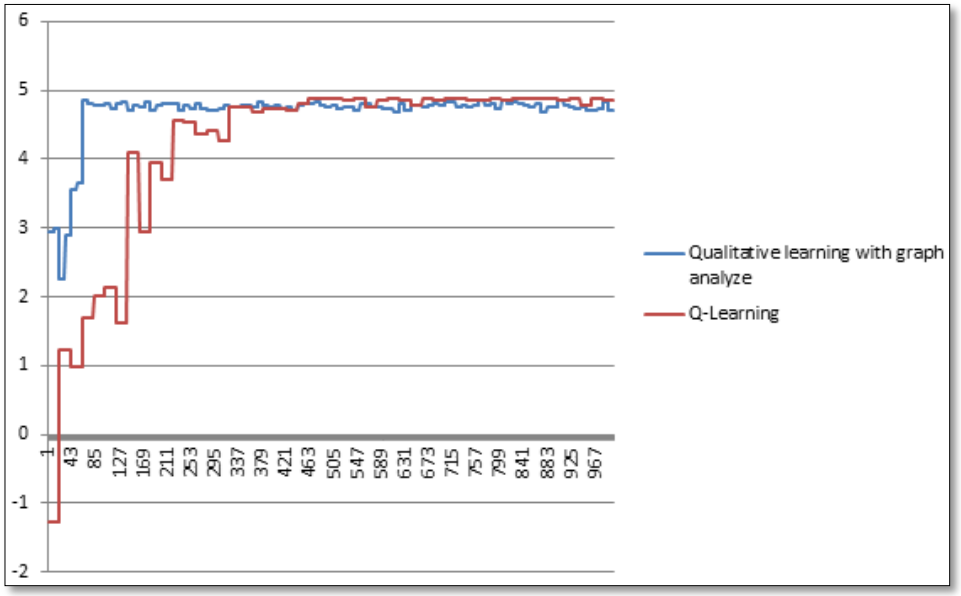
میانگین نتایج در هر اپیزود (بالا)، میانگین پاداش تجمعی (پایین)

علاوه بر این، الگوریتم های مختلفی برای خوشه بندی حالات بررسی گردیده است. یکی از الگوریتم های مورد بررسی K-means است. نتایج استفاده از آن برای محیط مشبک شش اتاقه در شکل ۳-۱۰ نمایش داده شده است. همانطور که مشاهده می شود، الگوریتم پیشنهادی با بکارگیری الگوریتم خوشه بندی EM اندکی سریعتر از K-means همگرا شده است. بررسی های مختلف و نتایج آن، به خوبی عمومیت چارچوب پیشنهادی را برای بکارگیری الگوریتم های مختلف نشان می دهد.

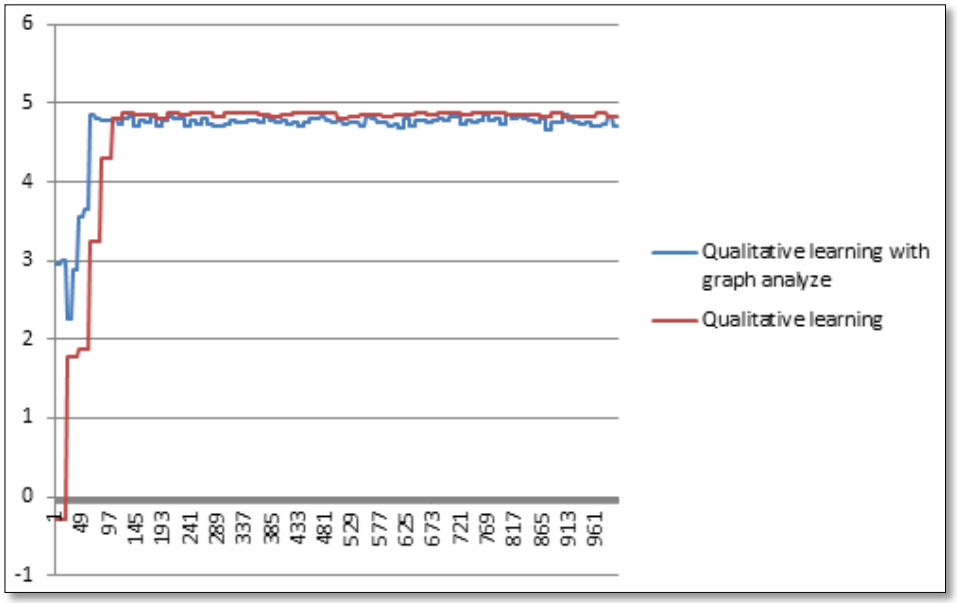


شکل ۳-۱۰: مقایسه نتایج بکارگیری الگوریتم خوشه بندی kmeans و EM

در ادامه، الگوریتم نمونه با ترکیب پاداش ساختگی و یادگیری کیفی در محیط شش اتاقه شبیه سازی شده است. نتیجه این بررسی در شکل ۳-۱۱ و شکل ۳-۱۲ آمده است. همانطور که مشاهده می شود، این نتایج بهبود عملکرد یادگیری را در الگوریتم حاصل نسبت به الگوریتم یادگیری کیو و همچنین الگوریتم کیفی (بدون پاداش ساختگی) نشان می دهد. همچنین، آزمون های آماری برای بررسی صحت نتایج در این بخش انجام گردیده است. ارزش z برای محیط شش اتاقه، برای مقایسه روش ترکیبی با روش یادگیری کیو و کیفی به ترتیب ۱۰/۴۶ و ۲/۰۱ بدست آمد. برای این مقادیر، مشخص است که می‌توان فرض صفر را به ترتیب با اطمینان ۹۹/۹۹ و ۹۷ درصد رد کرد. نتایج این شبیه سازی نشان می دهد که اگرچه استفاده از یادگیری کیفی تاثیر چشم گیری در شتاب سرعت یادگیری دارد، ترکیب آن با پاداش ساختگی، ناشی از تحلیل گراف، منجر به تسریع بیشتر یادگیری می شود.



شکل ۳-۱۱: مقایسه نتایج بکارگیری الگوریتم ترکیبی و یادگیری کیو



شکل ۳-۱۲: مقایسه نتایج بکارگیری الگوریتم ترکیبی و یادگیری کیفی

۳-۱۰ نتیجه گیری

در این تحقیق، چارچوب کیفی جدیدی پیشنهاد گردید و بخش های مختلف آن معرفی گردید. هدف این چارچوب رفع چالش های مطرح و تسریع فرایند یادگیری تقویتی است. این چارچوب به صورت کاملا کلی و انتزاعی معرفی شده است که قابلیت انطباق با الگوریتم های مختلف و محیط های مختلف را دارد. از دیدگاه نظری، بکارگیری همه بخش‌های این چارچوب ضرورت ندارد. بهرحال، برای افزایش سودمندی چارچوب پیشنهادی، بهتر است همه اجزای آن در هر الگوریتم نمونه وجود داشته باشند.

از روی این چارچوب پیشنهادی، الگوریتم نمونه ای معرفی شد که با کیفی سازی محیط و فراهم کردن دانش اولیه به شکل پاداش ساختگی می باشد. در ابتدا، این الگوریتم بدون در نظر گرفتن پاداش ساختگی برای محیط های مختلف آزمایش شبیه سازی شد. نتایج این بخش بهبود آنها را نسبت به روش یادگیری کیو نشان می دهد. در ادامه، با بهره گیری از پاداش ساختگی و کیفی سازی محیط الگوریتم نمونه روی محیط شش اتاقه پیاده سازی گردید. نتایج خروجی این الگوریتم نشان می دهد که اگرچه استفاده از یادگیری کیفی تاثیر چشم گیری در شتاب سرعت یادگیری دارد، ترکیب آن با روش های پاداش ساختگی منجر به تسریع بیشتر یادگیری می شود.

برای بررسی آنکه آیا بهبود بدست آمده در الگوریتم پیشنهادی از لحاظ آماری معنی دار است، از آزمون های فرض آماری صحت نتایج آن بررسی گردید. نتایج این آزمون ها، بهبود بدست آمده با استفاده از الگوریتم پیشنهادی را تایید می کند. بررسی های مختلف و نتایج آن، به خوبی عمومیت چارچوب پیشنهادی را برای بکارگیری الگوریتم های مختلف نشان می دهد. یکی از مهمترین پارامترهایی که در نتیجه فرآیند یادگیری کیفی موثر است، زمان ادغام حالات می باشد. این پارامتر بررسی شد و نتایج آن برای محیط مشبک شش اتاقه نیز ارائه گردید. اگر فرآیند ادغام در اپیزودهای اولیه که عامل شناختی

نسبت به وضعیت ها ندارد، یا در اپیزودهای بالا انجام شـود، نتیجـه آن مطلـوب نخواهـد بـود. همچنـین الگوریتم های مختلف خوشه بندی حالات بررسی شد، که نتایج پیاده سازی الگوریتم پیشنهادی بـا روش های خوشه بندی K-means و EM ارائه گردید.

با توجه به اینکه محیط برج های هانوی محیط کوچکی است و تعداد حالات زیادی ندارد، الگوریتم های یادگیری کیوی کلاسیک خیلی زود همگرا می شوند. در نتیجه بهبود نتایج برای استفاده از الگوریتم پیشنهادی در این محیط، چندان چشم گیر نیست. اما در محیط های بزرگتر، مانند محیط مشـبک شـش اتاقه و Complex Maze بطور قابل توجهی نتایج، بهبود در همگرایی به سیاست بهینه را نشان می دهـد. در نتیجه چارچوب پیشنهادی، در مقیاس پذیری الگوریتم های یادگیری تقویتی برای محیط های بـزرگ به خوبی عمل می کند.

فصل چهارم: نتیجه گیری و پیشنهادات

۴-۱ نتیجه گیری

کاربرد یادگیری تقویتی به محیط های بزرگ و پیچیده که در مسائل جهان واقعی با آنها مواجه هستیم، اغلب انگیزه ای برای توسعه الگوریتم های جدید بوده است. این الگوریتم ها، سعی دارند این فرآیند یادگیری را تا آنجا که ممکن است تسریع ببخشند. یک راهکار مفید در این رابطه، یادگیری تقویتی کیفی می باشد که در این تحقیق بررسی گردید. در این پایان نامه، چارچوب جدیدی بر مبنای یادگیری تقویتی کیفی معرفی گردید. هدف این چارچوب رفع چالش های مطرح در یادگیری تقویتی، مخصوصا در محیط های بزرگ و پیچیده، و تسریع فرآیند یادگیری تقویتی می باشد.

این چارچوب بر اساس یادگیری کیفی [۱۴] و تخمین پاداش ساختگی [۶] می باشد تا از فواید هر دو روش استفاده کند. در نتیجه، تا آنجا که ممکن است نرخ همگرایی بالاتری را فراهم می کند. اگرچه استفاده از یادگیری کیفی تاثیر چشم گیری در شتاب سرعت یادگیری دارد، ترکیب آن با پاداش ساختگی، ناشی از تحلیل گراف، منجر به تسریع بیشتر یادگیری می شود.

چارچوب پیشنهادی به صورت کاملا کلی و انتزاعی معرفی گردید و الگوریتم نمونه ای از آن ساخته شد. الگوریتم نمونه روی محیط های محک مختلف ناوبری و غیر ناوبری، با تعداد حالات مختلف بررسی گردید. نتایج آزمایش ها، بهبود در عملکرد یادگیری را نشان می دهد. صحت نتایج آزمایش ها، با استفاده از آزمون های فرض آماری بررسی شد که بهبود بدست آمده با استفاده از الگوریتم پیشنهادی را تایید می کند. هدف این چارچوب به‌طور اخص اکتشاف هوشمندانه‌تر محیط، کاهش زمان مورد نیاز جهت اکتشاف محیط و یادگیری محیط با سرعت بیشتر می باشد.

۲-۴ پیشنهادات برای کارهای آینده

چارچوب پیشنهادی در این تحقیق، دارای بخش های مختلفی است که می توان هر بخش از آن را با توجه به نیازهای مسئله دلخواه تطبیق داد. الگوریتم های مختلفی را می توان از روی این چارچوب انتزاعی ساخت که هسته عملکردی یکسانی دارند. کارهای مختلفی، می تواند از توسعه این چارچوب پیشنهادی در آینده انجام شود. به عنوان نمونه، برای بخش سوم چارچوب پیشنهادی الگوریتم دیگری استفاده شود. این الگوریتم می تواند به عنوان مثال، مبتنی بر مدل یا مدل آزاد باشد، تا نیاز های مسئله خاصی را برآورده کند. این الگوریتم می تواند با الگوریتم مورد استفاده بعد از کیفی سازی محیط و برای یادگیری محیط جدید متفاوت باشد. همچنین، روش های انتزاع زمانی می تواند در توسعه چارچوب بکار گرفته شود. چارچوب پیشنهادی در محیط های دیگر با ویژگی های مختلف می تواند در آینده بررسی شود تا از جنبه های مختلف کاملا بررسی گردد.

پیوست شماره ۱

پ-۱-۱ شبه کد الگوریتم دسته بندی حالات بر اساس اقدامات قانونی:

```
function CategorizeAccordingToLegalActions()
    categories := {};
    allStates := all states of the given environment;
    while ∃ state in allStates do begin
        list := {};
        for each state s in allStates do begin
            twoStatesHaveSameActions := TRUE;
            for each action a do begin
                if Q(s, a) != Q(state, a) then
                    twoStatesHaveSameActions := FALSE;
            endfor
            if twoStatesHaveSameActions == TRUE then
                list := list ∪ {s};
        endfor
        allStates := allStates – list;
         categories := categories ∪ list;
    endwhile
    return categories;
end CategorizeAccordingToLegalActions
```

پ-۱-۲ شبه کد الگوریتم دسته بندی حالات بر اساس مجاورت:

```
function CategorizeAccordingToAdjacencies(categories)
    newCategories := {};
    for each list in categories do begin
        tempCategories := {};
        while ∃ state in list do begin
           list2 := {};
            list2 := list2 ∪ {state};
           list = list – {state};
           newStateAdded := TRUE;
           while newStateAdded == TRUE do begin
              newStateAdded := FALSE;
```

```
            for each state s in list do begin
                tList := {};
                for each state s2 in list2 do begin
                    if adjMatrix(s, s2) == 1 then begin
                        tList := tList ∪ {s};
                        newStateAdded := TRUE;
                    endif
                endfor
                for each tState in tList do begin
                    list2 := list2 ∪ {tState};
                endfor
            endfor
            for each state2 in list2 do begin
                list := list – {state2};
            endfor
        endwhile
        tempCategories := tempCategories ∪ {list2};
    endwhile
    for each category in tempCategories do begin
        newCategories := newCategories ∪ category;
    endfor
endfor
return newCategories;
end CategorizeAccordingToAdjacencies
```

پ-۱-۳ شبه کد الگوریتم اجرای یک اپیزود در محیط اصلی:

```
function RunSingleEpisodeForPrimaryEnvironment(e)
    Initialize s, e.g. randomly, 1 ≤ s ≤ number of all states in environment, s.t. s is not goal
    state;
    r := 0.0;
    repeat (for each step, action, of current episode)
        if e > episodeToSelectGreedily then
            Choose a from s with highest Q value;
        else
            Choose a from s using policy derived from Q (e.g., ε-greedy);
        s' := next(s, a);
        r := r + reward(s, a);
        if useShaping == TRUE then
        Q(s, a) := Q(s, a) + α * [r + max_a' Q(s', a') – Q(s, a) + γ * max_a' Q(s', a') – Q(s, a)];
        else
```

```
        Q(s, a) := Q(s, a) + Beta * ManualReward(s,a) + α * [r + γ * max_a' Q(s', a') –
        Q(s, a)];
      endif
      s := s';
    until s is goal or the number of actions exceeds a threshold
    Store (e, r);
end RunSingleEpisodeForPrimaryEnvironment
```

پ-۱-۴ شبه کد الگوریتم اجرای یک اپیزود در محیط کاهش یافته:

```
function RunSingleEpisodeForNewEnvironment(e, newQ, newReward, newNext)
    Initialize s, e.g. randomly, 1 ≤ s ≤ number of all newStates in new environment, s.t. s
    is not goal state;
     r := 0.0;
    repeat (for each step, action, of current episode)
      if e > episodeToSelectGreedily then
        Choose a from s with highest newQ value;
      else
        Choose a from s using policy derived from newQ (e.g., ε-greedy);
      s' := Choose a new state from newNext(s, a) randomly;
      if s' is goal then
        r := r + the reward value when leading to goal;
      else
        r := r + newReward(s, a);
      endif
      if useShaping == TRUE then
        newQ(s, a) := newQ(s, a) +
          α * [r + max_a' newQ(s', a') – newQ(s, a) + γ * max_a' newQ(s', a') – newQ(s,
          a)];
      else
        newQ(s, a) := newQ(s, a) + Beta * ManualReward(s,a) +α * [r + γ * max_a'
        newQ(s', a') – newQ(s, a)];
      endif
      s := s';
    until s is goal or the number of actions exceeds a threshold
    Store (e, r);
end RunSingleEpisodeForNewEnvironment
```

پ-۱-۵- شبه کد الگوریتم مربوط به روش تکرار ارزش:

Input π,
Initialize V(s) = 0, for all s ϵ S
Repeat
  $\Delta \leftarrow 0$
  For *each s ϵ S*
    $v \leftarrow V(s)$
    $V(s) \leftarrow max a \sum_{s'} P^{a}_{ss'} \left[R^{a}_{ss'} + \gamma V(s')\right]$
    $\Delta \leftarrow max(\Delta, |v - V(s)|)$
Until $\Delta < \theta$ *(a small possible number)*
Output a deterministic policy, π, such that
  $\pi(s) = arg\, max_a \sum_{s'} P^{a}_{ss'} [R^{a}_{ss'} + \gamma V(s')]$

پ-۱-۶- شبه کد الگوریتم مربوط به روش تکرار سیاست:

Initialization
  V(s) ϵ R and π(s) ϵ A(s) arbitrary for all s ϵ S
Policy evaluation
  Repeat
    $\Delta \leftarrow$ **0**
    For each s ϵ Sv $\leftarrow$ **V(s)**
    $V(s) \leftarrow max a \sum_{s'} P^{a}_{ss'} \left[R^{a}_{ss'} + \gamma V(s')\right]$
    $\Delta \leftarrow$ **max**$(\Delta, |$**v** $-$ **V(s)**$|)$
  Until $\Delta <$ **θ (a small possible number)**
Policy improvement
  Policy-stable ← true
  For each s ϵ S
    h $\leftarrow$ **π(s)**
    π(s) $\leftarrow$ **arg max a** $\sum_{s'} \mathbf{P^{a}_{ss'}} [\mathbf{R^{a}_{ss'}} + \boldsymbol{\gamma}\, \mathbf{V(s')}]$
    If b $\neq$ **π(s), then policy-stable ← false**
  If *policy-stable,* **then** *stop,* **else** *go to 2*

پ-۱-۷- شبه کد الگوریتم یادگیری سارسا:

Initialize Q(s,a) arbitrarily
Repeat (for each episode):
. *Initialize s*
. *Choose a from s using policy derived from Q*
. . . *(e.g. , $\varepsilon - greedy$)*
. **Repeat** *(for each step of episode):*
. . *Take action a, observe r, s'*
. . *Choose a' from s' using policy derived from Q*
. . . *(e.g. , $\varepsilon - greedy$)*
. . $Q(s,a) \leftarrow Q(s,a) + \alpha\,[r + \gamma Q(s',a') - Q(s,a)]$
. . $S \leftarrow s' ; a \leftarrow a'$
. **Until** s is terminal

پ-۱-۸- شبه کد الگوریتم خوشه بندی EM:

Select an initial set of model parameters: can be done randomly or in a,variety of ways. (the algorithm's input are the data set (x), the total number of clusters (M),,the accepted error to converge (e) and the maximum number of iterations)
Repeat
Estimation Step*: estimates the probability of each point belongs to each cluster.*
Maximization Step: *re-estimate the parameter vector of the probability ,, distribution of each class.*
Until *the distribution parameters converges or reach the maximum number of iterations.*

پ-۱-۹- الگوریتم استفاده شده برای پاداش ساختگی

این روش در سه مرحله پاداش ساختگی را فراهم می کند. در مرحله اول، اهداف میانی شناسایی می شوند. برای این منظور از خاصیت مرکزیت میانگی گراف استفاده می شود [21]. استفاده از معیار مرکزیت میانگی در شناسایی اهداف میانی دو ویژگی دارد. این روش، حالاتی که در وظایف مختلف عامل

بطور مکرر ملاقات می شوند، و حالاتی که دارای مرکزیت بالایی هستند شناسایی می کند. بـدین ترتیـب میانگین زمان جست و جوی عامل در محیط کاهش می یابد. روش های مبتنی بر مرکزیـت گـراف دارای پیچیدگی محاسباتی بالایی هستند. همچنین این روش ها نیاز به حافظه زیادی برای نگـه داری اطلاعـات کل گراف دارند. از مزایای روش های مبتنی بر مرکزیت میانگی این است که پیچیدگی محاسباتی آنها کم است. همچنین می توان توسط این روش ها، تنها تعداد محدودی از گره ها را در حافظه نگه داری کرد.

این نقاط شناسایی می شود و پاداش مجازی بالاتری به آنها داده می شود. در مرحلـه دوم، حـالات کم اهمیت بر اساس ماتریس مجاورت گراف شناسایی می شوند. حالاتی که در مـاتریس مجـاورت گـراف کمترین همسایگی را دارند (به شرط هدف میانی نبودن)، به عنوان حالات کم اهمیت شناسایی می شوند. به این دسته از حالات، پاداش مجازی کمتری داده می شود. در مرحله آخر، گراف بر اساس اهداف میـانی شناسایی شده به زیرگراف هایی تجزیه می شود. در ادامه، برای هر حالت پاداشی در نظر گرفته می شـود که بر اساس نسبت تعداد گره ها در زیرگراف حاوی آن حالت، به تعداد کل گره ها می باشد. ایـن نسـبت، در مقدار پاداش حالات کم اهمیت ضرب می شود. ایده این پاداش مجازی این اسـت کـه احتمـال اینکـه هدف در زیرگراف بزرگتر باشد، بیشتر است. در نهایت، پاداش ساختگی حاصل برای هر حالـت در فرمـول یادگیری کیو در نظر گرفته می شود. تزریق این پاداش باعث فراهم شدن دانـش اولیـه و تسـریع فرآینـد یادگیری می شود.

پیوست شماره ۲: واژه نامه

Abstraction	انتزاع – تجرید
Abstract Action	فراکنش
Abstract State	حالت مجرد
Agent	عامل
Average Reward	میانگین پاداش
Behavior	رفتار
Bellman Equation	معادله بلمن
Betweenness Centrality	مرکزیت میانگی
Clustering	خوشه بندی
Decision Tree	درخت تصمیم
Deterministic	قطعی
Discount Factor	نرخ تنزیل
Dynamic Bayesian Network	شبکه دینامیکی بیزی
Environment	محیط
Episode	اپیزود
Framework	چارچوب
Function Approximation	تقریب توابع
Graph Analysis	تحلیل گراف
Hierarchical Reinforcement Learning	یادگیری تقویتی سلسله مراتبی
hypothesis test	آزمون فرض
Infinite Horizon	بازه نامتناهی
Iterative	تکراری
Knowledge Reuse	استفاده مجدد از دانش

Machine Learning	یادگیری ماشین
Macro Action	فراکنش
Markov Decision Process	فرایند تصمیم گیری مارکوف
Model Based	مبتنی بر مدل
Model Free	مدل آزاد
Neural Network	شبکه عصبی
Offline	برون خط
Online	برخط
Optimal Policy	سیاست بهینه
Option	گزینه
Partial Observable Markov Decision Process	فرایند تصمیم گیری مارکوف با اطلاعات ناقص
Policy	سیاست
Policy Search	جستجوی سیاست
Primitive Action	اقدام پایه
Q-Leaning	یادگیری کیو
Q-Table	جدول کیو
Qualitative Reinforcement Learning	یادگیری تقویتی کیفی
Reinforcement Learning	یادگیری تقویتی
Reward shaping	شکل دهی پاداش
SARSA	سارسا
Sensor	سنسور- حسگر
Sequential Decision Making	تصمیم گیری ترتیبی
Skill	مهارت
State Abstraction	تجرید حالت
State-Action	حالت – کنش
Stochastic	تصادفی
Subgoal	هدف میانی
Sub-environment	زیرمحیط
Task	وظیفه

Temporal Abstraction	انتزاع زمانی
The Curse of Dimensionality	نفرین ابعاد
Transition Function	تابع گذر
Transition Probability Function	تابع احتمال گذر
Value Function	تابع ارزش

فهرست مراجع

[1] M. E. Harmon and S. S. Harmon, “Reinforcement learning: a tutorial,” *WL/AAFC, WPAFB Ohio*, vol. 45433, 1996.

[2] R. S. Sutton and A. G. Barto, “Reinforcement learning: an introduction.,” *IEEE Trans. Neural Netw.*, vol. 9, no. 5, p. 1054, 1998.

[3] P. Kormushev, S. Calinon, and D. Caldwell, “Reinforcement Learning in Robotics: Applications and Real-World Challenges,” *Robotics*, vol. 2, pp. 122–148, 2013.

[4] L. Frommberger, “Qualitative Spatial Abstraction in Reinforcement Learning,” pp. 23–41, 2010.

[5] F. Telgerdi, A. Khalilian, and A. A. Pouyan, “Qualitative reinforcement learning to accelerate finding an optimal policy,” in *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2014, pp. 575–580.

[6] M. Grześ and D. Kudenko, “Online learning of shaping rewards in reinforcement learning,” *Neural Networks*, vol. 23, pp. 541–550, 2010.

[7] P. Moradi, M. E. Shiri, A. A. Rad, A. Khadivi, and M. Hasler, “Automatic skill acquisition in reinforcement learning using graph centrality measures,” in *Intelligent Data Analysis*, 2012, vol. 16, pp. 113–135.

[8] M. Marashi, A. Khalilian, and M. E. Shiri, “Automatic reward shaping in Reinforcement Learning using graph analysis,” in *2012 2nd International eConference on Computer and Knowledge Engineering (ICCKE)*, 2012, pp. 111–116.

[9] S. Whiteson and P. Stone, “Evolutionary Function Approximation for Reinforcement Learning,” *J. Mach. Learn. Res.*, vol. 7, pp. 877–917, 2006.

[10] R. S. Sutton, D. Precup, and S. Singh, “Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning,” *Artif. Intell.*, vol. 112, pp. 181–211, 1999.

[11] A. A. Rad, M. Hasler, and P. Moradi, “Automatic skill acquisition in Reinforcement Learning using connection graph stability centrality,” in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, 2010, pp. 697–700.

[12] T. G. Dietterich, “Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition,” *J. Artif. Intell. Res.*, vol. 13, pp. 227–303, 2000.

[13] L. Li, T. J. Walsh, and M. L. Littman, “Towards a Unified Theory of State Abstraction for MDPs,” 2006.

[14] A. Epshteyn and G. DeJong, “Qualitative reinforcement learning,” *Proc. 23rd Int. Conf. Mach. Learn. - ICML ’06*, pp. 305–312, 2006.

[15] L. Frommberger, *Qualitative Spatial Abstraction in Reinforcement Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[16] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1994.

[17] R. Bellman, *Dynamic Programming*, 1st ed. Princeton, NJ, USA: Princeton University Press, 1957.

[18] J. A. Boyan and A. W. Moore, “Generalization in Reinforcement Learning: Safely Approximating the Value Function,” in *Advances in Neural Information Processing Systems 7*, 1995, pp. 369–376.

[19] R. A. Howard, *Dynamic Programming and Markov Processes*, vol. 3. 1960, p. 120.

[20] D. Bertsekas, *D.P.: Dynamic Programming*. Prentice-Hall, 1987.

[21] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, 1996.

[22] S. M. LaValle, “Planning Algorithms,” *Methods*, vol. 2006, p. 842, 2006.

[23] U. T. Austin, C. Science, and T. Report, “Learning Methods for Sequential Decision Making with Imperfect Representations,” no. December, 2011.

[24] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement Learning : A Survey,” pp. 237–285, 1996.

[25] Watkin, “Watkin Proof of Q-Learning Convergence,” 1992.

[26] G. A. Rummery and M. Niranjan, “On-line Q-learning using connectionist systems,” no. September, 1994.

[27] B. Y. L. Li, “A UNIFYING FRAMEWORK FOR COMPUTATIONAL REINFORCEMENT LEARNING THEORY,” 2009.

[28] L. Frommberger, *Qualitative Spatial Abstraction in Reinforcement Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 1–8.

[29] A. Bernstein and N. Shimkin, “Adaptive Aggregation for Reinforcement Learning with Efficient Exploration : Deterministic Domains,” 2009.

[30] G. Z. G. Zhao, S. Tatsumi, and R. S. R. Sun, “A heuristic Q-learning architecture for fully exploring a world and deriving an optimal policy by model-based planning,” *Proc. 1999 IEEE Int. Conf. Robot. Autom. (Cat. No.99CH36288C)*, vol. 3, 1999.

[31] R. A. C. Bianchi, C. H. C. Ribeiro, and A. H. R. Costa, “Heuristic Selection of Actions in Multiagent Reinforcement Learning,” in *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, 2007, pp. 690–696.

[32] M. Kearns and S. Singh, “Near-optimal reinforcement learning in polynomial time,” *Mach. Learn.*, vol. 49, pp. 209–232, 2002.

[33] A. L. Strehl and M. L. Littman, “A theoretical analysis of model-based interval estimation,” in *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, 2005, pp. 857–864.

[34] R. I. Brafman and M. Tennenholtz, “R-MAX -- A General Polynomail Time Algorithm for Near-Optimal Reinforcement Learning,” *J. Mach. Learn. Res.*, vol. 3, pp. 213–231, 2002.

[35] H. Chapman, “Global confidence bound algorithms for the exploration-exploitation tradeoff in reinforcement learning,” Technion – Israel Institute of Technology, 2007.

[36] P. Auer and R. Ortner, “Logarithmic Online Regret Bounds for Undiscounted Reinforcement Learning.,” in *NIPS*, 2006, pp. 49–56.

[37] A. Tewari and P. Bartlett, “Optimistic Linear Programming gives Logarithmic Regret for Irreducible {MDPs},” in *Advances in Neural Information Processing Systems 20*, 2008, pp. 1505–1512.

[38] A. Epshteyn and G. DeJong, “Qualitative reinforcement learning,” in *Proceedings of the 23rd international conference on Machine learning - ICML ’06*, 2006, pp. 305–312.

[39] L. Saitta and J.-D. Zucker, *Abstraction in Artificial Intelligence and Complex Systems*. Springer Publishing Company, Incorporated, 2013.

[40] F. Giunchiglia, “A theory of abstraction,” *Artificial Intelligence*, vol. 57. pp. 323–389, 1992.

[41] L. Frommberger, *Qualitative Spatial Abstraction in Reinforcement Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 43–65.

[42] A. Galton and R. Meathrel, “Qualitative Outline Theory,” in *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, 1999, pp. 1061–1066.

[43] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. II*, vol. 2. 2007, p. 543.

[44] C. Boutilier, T. Dean, and S. Hanks, “Decision-Theoretic Planning: Structural Assumptions and Computational Leverage,” *J. Artif. Intell. Res.*, vol. 11, pp. 1–94, 1999.

[45] C. S. A. Antos, R. Munos, "Fitted Qiteration in continuous action-space MDPs," in *Neural Information Processing Systems Conference (NIPS)*, 2007.

[46] A.Epshteyn, "Combining Prior Knowledge And Data: Beyond The Bayesian Framework," University Of Illinois At Urbana-Champaign, 2006.

[47] T. Dean and K. Kanazawa, "A model for reasoning about persistence and causation," *Comput. Intell.*, vol. 5, no. 2, pp. 142–150, 1989.

[48] R. Dearden and C. Boutilier, "Artificial Intelligence Abstraction and approximate decision-theoretic planning *," vol. 3702, no. 96, 1997.

[49] T. Dean and R. Givan, "Model minimization in Markov decision processes," in *Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence*, 1997, pp. 106–111.

[50] B. Bonet and J. Pearl, "Qualitative MDPs and POMDPs: an order-of-magnitude approximation," in *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, 2002, pp. 61–68.

[51] R. Sabbadin, "Possibilistic Model for Qualitative Sequential Decision Problems under Uncertainty in Partially Observable Environments," in *Uncertainty in Artificial Intelligence - UAI*, 1999, pp. 567–574.

[52] A. Reyes, L. E. Sucar, E. Morales, and P. H. Ibarguengoytia, "Learning Qualitative Markov Decision Processes," *Neural Inf. Process. Syst. NIPS 2005 Work. Mach. Learn. Based Robot. Unstructured Environ.*, 2005.

[53] Zadeh L., "Fuzzy sets and information granularity," in *M. Gupta, R. Ragade and R. Yager (eds.) Advances in Fuzzy Systems-Applications and Theory*, vol. 6, North-Holland Publishing Co., 1997, pp. 3–18.

[54] T. Bittner and B. Smith, "A taxonomy of granular partitions," *Spat. Inf. Theory*, pp. 28–43, 2001.

[55] W. A. MACKANESS and O. CHAUDHRY, “Generalization and symbolization,” in *Encyclopedia of GIS*, S. Shekhar and H. Xiong, Eds. Boston, MA: Springer US, 2008, pp. 330–339.

[56] A. Herskovits, “Schematization,” in *Representation and Processing of Spatial Expressions*, P. Olivier and K. P. Gapp, Eds. Lawrence Erlbaum Associates, Mahwah, NJ, USA, 1998, pp. 149–162.

[57] A. Klippel, K. F. Richter, T. Barkowsky, and C. Freksa, “The cognitive reality of schematic maps,” in *Map-based Mobile Services: Theories, Methods and Implementations*, 2005, pp. 55–71.

[58] J. G. Stell and M. F. Worboys, “Generalizing graphs using amalgamation and selection,” in *Advances in Spatial Databases*, vol. 1651, 1999, pp. 19–32.

[59] S. Bertel, C. Freksa, and G. Vrachliotis, “Aspectualize and conquer in architectural design,” in *Visual and Spatial Reasoning in Design III*, 2004, pp. 255–279.

[60] S. Bertel, G. Vrachliotis, and C. Freksa, “Aspect-oriented building design: Towards computer-aided approaches to solving spatial constraints in architecture,” in *Applied spatial cognition: From research to cognitive technology*, G. L. Allen, Ed. Lawrence Erlbaum Associates; Mahwah, NJ, 2007, pp. 75–102.

[61] N. K. Jong, T. Hester, and P. Stone, “The Utility of Temporal Abstraction in Reinforcement Learning,” in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, 2008, pp. 299–306.

[62] R. Parr and S. Russell, “Reinforcement learning with hierarchies of machines,” *Adv. neural Inf. Process. …*, pp. 1043–1049, 1998.

[63] A. G. Barto and S. Mahadevan, “Recent advances in hierarchical reinforcement learning,” *Discret. Event Dyn. Syst.*, vol. 13, pp. 341–379, 2003.

[64] G. Kheradmandian and M. Rahmati, "Automatic abstraction in reinforcement learning using data mining techniques," *Rob. Auton. Syst.*, vol. 57, pp. 1119–1128, 2009.

[65] B. Bakker and J. Schmidhuber, "Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization," *Proc. 8-th Conf. Intell. ...*, 2004.

[66] R. M. Kretchmar, T. Feil, and R. Bansal, "Improved Automatic Discovery of Subgoals for Options in Hierarchical Reinforcement Learning," *October*, vol. 3, pp. 9–14, 2003.

[67] A. Jonsson and A. G. Barto, "Automated State Abstraction for Options using the U-Tree Algorithm," in *NIPS'00*, 2000, pp. 1054–1060.

[68] M. Asadpour, M. N. Ahmadabadi, and R. Siegwart, "Heterogeneous and hierarchical cooperative learning via combining decision trees," in *IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 2684–2690.

[69] S. Elfwing, E. Uchibe, K. Doya, and H. I. Christensen, "Evolutionary development of hierarchical learning structures," *IEEE Trans. Evol. Comput.*, vol. 11, pp. 249–264, 2007.

[70] A. Jonsson, "Causal graph based decomposition of factored MDPs," *J. Mach. Learn. Res.*, vol. 7, pp. 2259–2301, 2006.

[71] V. Manfredi and S. Mahadevan, "Hierarchical Reinforcement Learning using Graphical Models," *ICML*, 2005.

[72] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," *Proceedings. 1985 IEEE Int. Conf. Robot. Autom.*, vol. 2, 1985.

[73] J.-S. Gutmann, T. Weigel, and B. Nebel, "A fast, accurate and robust method for self-localization in polygonal environments using laser range finders," *Adv. Robot.*, vol. 14, no. 8, pp. 651–667, Jan. 2001.

[74] W. T. B. Uther and M. M. Veloso, "Tree based discretization for continuous state space reinforcement learning," pp. 769–774, Jul. 1998.

[75] M. Asadpour, M. N. Ahmadabadi, and R. Siegwart, "Reduction of Learning Time for Robots Using Automatic State Abstraction," in *Proc. of the First European Symposium on Robotics*, 2006, vol. 22, pp. 79–92.

[76] A. K. McCallum, "Reinforcement learning with selective perception and hidden state," 1996.

[77] C. Boutilier, R. Dearden, and M. Goldszmidt, "Exploiting structure in policy construction," in *International Joint Conference on Artificial Intelligence*, 1995, vol. 14, pp. 1104–1113.

[78] R. Dearden and C. Boutilier, "Artificial Intelligence Abstraction and approximate decision-theoretic planning *," vol. 3702, no. 96, 1997.

[79] A. W. Moore and C. G. Atkeson, "Parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces," *Mach. Learn.*, vol. 21, pp. 199–233, 1995.

[80] R. J. W. Mohammad A. Al-Ansari, "Robust, Efficient, Globally-Optimized Reinforcement Learning with the Parti-Game Algorithm," *Adv. Neural Inf. Process. Syst. Proc. 1998 Conf.*, 1999.

[81] M. Likhachev and S. Koenig, "Speeding up the Parti-Game Algorithm," *Adv. Neural Inf. Process. Syst. Proc. 2002 Conf.*, 2003.

[82] S. I. Reynolds, "Adaptive Resolution Model-Free Reinforcement Learning: Decision Boundary Partitioning," in *In Proc. 17th International Conf. on Machine Learning*, 2000, pp. 783–790.

[83] H. Vollbrecht, "Hierarchic function approximation in kd-Q-learning," *KES'2000. Fourth Int. Conf. Knowledge-Based Intell. Eng. Syst. Allied Technol. Proc. (Cat. No.00TH8516)*, vol. 2, 2000.

[84] C. Boutilier, R. Dearden, and M. Goldszmidt, "Stochastic dynamic programming with factored representations," *Artif. Intell.*, vol. 121, pp. 49–107, 2000.

[85] R. Givan, T. Dean, and M. Greig, "Equivalence notions and model minimization in Markov decision processes," *Artif. Intell.*, vol. 147, pp. 163–223, 2003.

[86] B. Ravindran and A. G. Barto, “SMDP homomorphisms: An algebraic approach to abstraction in semi-Markov decision processes,” in *IJCAI International Joint Conference on Artificial Intelligence*, 2003, pp. 1011–1016.

[87] S. L. Thomas Dean, Robert Givan, “Model Reduction Techniques for Computing Approximately Optimal Solutions for Markov Decision Processes,” 2013.

[88] N. K. Jong and P. Stone, “State abstraction discovery from irrelevant state variables,” in *IJCAI International Joint Conference on Artificial Intelligence*, 2005, pp. 752–757.

[89] B. Van Roy, “Performance Loss Bounds for Approximate Value Iteration with State Aggregation,” *Math. Oper. Res.*, vol. 31, no. 2, pp. 234–244, May 2006.

[90] E. Even-dar and Y. Mansour, “LNAI 2777 - Approximate Equivalence of Markov Decision Processes,” pp. 581–594, 2003.

[91] N. Ferns, P. Panangaden, and D. Precup, “Metrics for finite Markov decision processes,” *UAI ’04 Proc. 20th Conf. Uncertain. Artif. Intell.*, 2004.

[92] P. L. Schweitzer, M. L. Puterman, and K. W. Kin-dle, “Iterative aggregation-disaggregation procedures for discounted semi-Markov reward processes,” in *Operations Research*, 1985.

[93] C. A. Knoblock, *Generating Abstraction Hierarchies: An Automated Approach to Reducing Search in Planning*. Norwell, MA, USA: Kluwer Academic Publishers, 1993.

[94] C. Boutilier and R. Dearden, “Approximating value trees in structured dynamic programming,” in *In Proceedings of the Thirteenth International Conference on Machine Learning*, 1996, pp. 54–62.

[95] J. Baum and A. E. Nicholson, “Dynamic non-uniform abstractions for approximate planning in large structured stochastic domains,” in *PRICAI’98: Topics in Artificial Intelligence*, vol. 1531, 1998, pp. 587–598.

[96] A. P. S. Braga and A. F. R. Ara$\hat{}$jo, “A topological reinforcement learning agent for navigation,” *Neural Comput. Appl.*, vol. 12, no. 3–4, pp. 220–236, Dec. 2003.

[97] R. Glaubius, M. Namihira, and W. D. Smart, “Speeding up reinforcement learning using manifold representations: Preliminary results,” in *In Proceedings of the ?CAI Workshop $\hat{}$Reasoning with Uncertainty in Robotics*, 2005.

[98] D. Busquets, R. L. de Màntaras, C. Sierra, and T. G. Dietterich, “Reinforcement Learning for Landmark-based Robot Navigation,” in *First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002)*, 2002, pp. 841–842.

[99] C. Sierra, R. L. de Màntaras, and \textbfD’idac Busquets, “Multiagent Bidding Mechanisms for Robot Qualitative Navigation,” in *Lecture Notes in Artificial Intelligence*, 2000, vol. 1986, pp. 198–212.

[100] G. Theocharous and L. P. Kaelbling, “Spatial and Temporal Abstractions in POMDPS: Learning and Planning.” 2005.

[101] L. E. Baum and G. R. Sell, “Growth Transformations for Functions on Manifolds,” *Pacific J. Math.*, vol. 27, pp. 211–227, 1968.

[102] P. Berkhin, “A survey of clustering data mining techniques,” *Group. Multidimens. data*, pp. 25–71, 2006.

[103] P. Moradi, “Automatic Skill Acquisition In R.L. Using Autonomous Agent,” Amirkabir University Iran, 2011.

[104] S. Koenig and R. G. Simmons, “Complexity Analysis of Real-Time Reinforcement Learning $*$,” 1992.

[105] Freund, *Mathematical Statistics*, 5th ed. Prentice Hall College Div, 1992.

Abstract

Reinforcement learning is an interesting area of machine learning, which aims at improving the agent behavior based on Reinforcement signals received from the environment. However, in many real applications, the reward is given to the agent with much delay. As a result, the agent needs to spend much time to achieve optimal behavior. Different methods such as reward shaping have been proposed to overcome this problem. But none could have noticeable effect to increase the learning speed, especially in large and real environments. Another problem is that as long as the agent doen not reach to an acceptable level of learning, all of its movements are random. Moreover, further complicating the environment would lead to an increase in the number of explorations and decision parameters. These issues make exploration time consuming, costly and sometimes very dangerous. An interesting solution for researchers to address these issues is qualitative learning. In this thesis, a general framework for qualitative learning, along with its properties and components is presented. This framework is designed based on the qualitative learning and reward shaping to take the benefits from both worlds and provides as much possible convergence rate as possible. The proposed framework is adjustable and adaptable to different algorithms either discrete or continuous, and navigation and non-navigation environments. Then, a prototype is instantiated from the proposed framework and further evaluated on some benchmark environments. The results of the experiments demonstrate the effectiveness of the proposed framework to achieve the optimal policy and accelerating the learning process.

Keywords: reinforcement learning, Q-learning, qualitative learning, Graph Analysis, abstraction

Shahrood University of Technology

Faculty of Computer Engineering and IT

Skill Acquisition in robotic Reinforcement Learning

Using Autonomous Agents

Fateme Telgerdi

Supervisor:

Dr. Ali Akbar Pouyan

2014