ارائه روشی جدید برای کسب مهارت در یادگیری تقویتی با کمک خوشه‌بندی گراف

مرضیه داودآبادی فراهانی و ناصر مزینی

چکیده: یادگیری تقویتی، یکی از انواع یادگیری ماشین است که در آن عامل با استفاده از تراکنش با محیط، به شناخت محیط و بهبود رفتار خود می‌پردازد. یکی از مشکلات اصلی الگوریتم‌های استاندارد یادگیری تقویتی مانند یادگیری Q این است که نمی‌توانند مسایل بزرگ را در زمان قابل قبولی حل کنند. کسب خودکار مهارت‌ها می‌تواند به شکستن مسأله به زیرمسأله‌های کوچک‌تر و حل سلسله‌مراتبی آن کمک کند. با وجود نتایج امیدوارکننده استفاده از مهارت‌ها در یادگیری تقویتی سلسله‌مراتبی، در برخی تحقیقات دیگر نشان داده شد که بر اساس وظیفه مورد نظر، اثر مهارت‌ها بر کارایی یادگیری می‌تواند کاملاً مثبت یا منفی باشد و اگر به درستی انتخاب نشوند می‌توانند پیچیدگی حل مسأله را افزایش دهند. از این رو یکی از نقاط ضعف روش‌های قبلی کسب خودکار مهارت‌ها، عدم ارزیابی هر یک از مهارت‌های کسب‌شده می‌باشد. در این مقاله روش‌های جدیدی مبتنی بر خوشه‌بندی گراف برای استخراج زیرهدف‌ها و کسب مهارت‌ها ارائه می‌گردد. همچنین معیارهای جدید برای ارزیابی مهارت‌ها مطرح می‌شود که با کمک آنها، مهارتهای نامناسب برای حل مسأله حذف می‌گردند. استفاده از این روش‌ها در چندین محیط آزمایشگاهی افزایش سرعت یادگیری را به شکل قابل ملاحظه‌ای نشان می‌دهد.

کلیدواژه: **یادگیری تقویتی سلسله‌مراتبی، گزینه، انتزاع زمانی، مهارت، ارزیابی مهارت‌ها، خوشه‌بندی گراف.**

۱- مقدمه

یادگیری تقویتی (RL)[1] یکی از حوزه‌های فعال تحقیقات یادگیری ماشین است که توجه زیادی در زمینه‌های نظریه تصمیم، پژوهش در عملیات و مهندسی کنترل به آن شده است [۱]. الگوریتم‌های یادگیری تقویتی این مسأله را که چگونه یک عامل در حالی که با محیط خود ارتباط مستقیم دارد، می‌تواند سیاست بهینه رفتار خود را تقریب بزند، مورد توجه قرار داده‌اند. این در شرایطی است که معمولاً اطلاعات کاملی از محیط در دسترس نیست [۲]. روش‌های استاندارد یادگیری تقویتی مسایلی را به همراه دارند که استفاده از آنها را محدود می‌سازد. یکی از این مسایل، مسأله نفرین بعد[2] است، یعنی در بسیاری از موارد واقعی قادر نیستند مسایل بزرگ با ابعاد زیاد را در زمان معقولی حل کنند. دومین مسأله مطرح‌شده در این زمینه، امکان انتقال دانش[3] یاد گرفته شده است. اگر بخشی از فضای حالت مشابه بخش دیگری باشد بدون استفاده از تقریب تابع[4]، راهی برای انتقال دانش از بخشی از فضای حالت به بخش دیگر یا استفاده مجدد از این دانش در مسایل دیگر نیست. مسأله سوم کندبودن اکتشاف به دلیل راهپیمایی تصادفی[5] در مراحل اولیه یادگیری است. اگر اندازه فضای حالت بزرگ باشد و پاداش به جز در حالت هدف اعطا نگردد (همان گونه که بسیاری از مسایل یادگیری تقویتی این گونه هستند)، عامل در شروع یادگیری و تا قبل از این که به هدف برسد فقط راهپیمایی تصادفی می‌کند [۳].

در تلاش‌های اخیر برای حل مسایل با فضای حالت بسیار بزرگ در حوزه یادگیری تقویتی روش‌های تجزیه سلسله‌مراتبی ارائه شده‌اند. این روش‌ها با استفاده از ساختار یک مسأله، آن را به صورت سلسله‌مراتبی به زیرمسایل کوچک‌تر تجزیه می‌کنند و هر زیرمسأله جداگانه و با استفاده از زیرمجموعه کوچکی از مجموعه حالت‌ها حل می‌شود. در نتیجه یادگیری هر زیرمسأله بسیار ساده‌تر و سریع‌تر از مسأله اصلی خواهد بود. همچنین هر زیرمسأله می‌تواند در مسایل دیگر هم به کار گرفته شود.

به دو روش می‌توان مسایل بزرگ را به مجموعه‌ای از مسایل کوچک تجزیه کرد و فضای مسأله را کاهش داد. یکی از این روش‌ها، انتزاع حالت[6] است. در انتزاع حالت، مجموعه‌ای از حالت‌های مشابه با هم در مسأله به یک حالت مجرد بازنمایی می‌شود. یکی دیگر از راه حل‌های کاهش فضای مسأله، انتزاع زمانی[7] است. در انتزاع زمانی، مجموعه‌ای از کنش‌ها، تشکیل یک فراکنش[8] می‌دهند. انتزاع‌های زمانی نیاز به تصمیم‌گیری در هر گام را از بین می‌برند و با اجرای فعالیت‌های گسترش‌یافته زمانی[9] که سیاست خودشان را تا زمان توقف دنبال می‌کنند، منجر به معماری سلسله‌مراتبی می‌شوند [۴]. همچنین به کمک انتزاع‌های زمانی یا مهارت‌ها[10] اکتشاف سریع‌تر انجام خواهد شد و به الگوریتم اجازه داده می‌شود که پاداش‌ها را زودتر منتشر کند [۵].

با وجود مزایای گفته‌شده برای کسب خودکار مهارت‌ها این مسأله هنوز به عنوان یک مسأله باز باقی مانده است. علاوه بر این در مطالعات قبلی ارزیابی دقیقی از اثر آنها بر کارایی یادگیری ارائه نشده است. در [۶] نشان داده شده که مهارت‌ها گاهی سرعت یادگیری را افزایش و گاهی کاهش می‌دهند و این بستگی به این دارد که به وظیفه داده‌شده مرتبط هستند یا نه. در [۷] ذکر شده که مفیدبودن انتزاع‌های زمانی در یادگیری تقویتی به فاکتورهای متفاوتی بستگی دارد. این فاکتورها شامل الگوریتم یادگیری

این مقاله در تاریخ ۱۴ اسفند ماه ۱۳۹۵ دریافت و در تاریخ ۲۹ بهمن ماه ۱۳۹۶ بازنگری شد.

مرضیه داودآبادی فراهانی، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، (email: davoodabadi@comp.iust.ac.ir).

ناصر مزینی، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، (email: mozayani@iust.ac.ir).

1. Reinforcement Learning
2. Curse of Dimensionality
3. Knowledge Transfer
4. Function Approximation
5. Random Walk
6. State Abstraction
7. Temporal Abstraction
8. Macro-Action
9. Temporally Extended Activities
10. Skill

دقیقاً به کار گرفته شده، زمان و مکان به کارگیری انتزاع‌های زمانی و در دسترس بودن یا نبودن کنش‌های پایه می‌باشد. نویسندگان این مقاله همچنین مطرح کردند که بسیاری از بهبودهای نشان داده شده در سرعت یادگیری پس از استفاده از انتزاع‌های زمانی، مربوط به استفاده از روش تکرار بازنمایی تجربه[1] است که معمولاً برای یادگیری سیاست انتزاع‌های زمانی استفاده می‌شوند. بنا بر تجربیات به دست آمده در این دو مقاله ارزیابی هر مهارت به صورت مستقل ضروری به نظر می‌رسد.

در این مقاله روش‌های جدیدی برای استخراج و یادگیری زیرهدف‌ها و مهارت‌ها و ارزیابی آنها ارائه می‌گردد. بخش‌های بعدی این مقاله به صورت زیر سازماندهی شده‌اند. در بخش ۲ مقدمه‌ای بر یادگیری تقویتی ارائه شده و برخی روش‌های موجود برای کسب مهارت‌ها معرفی می‌گردند. در بخش ۳ روش‌های جدیدی برای کسب مهارت، کشف زیرهدف‌ها و ارزیابی مهارت‌ها ارائه می‌گردد. در بخش ۴ به آزمایشات انجام‌شده برای نشان‌دادن نتایج استفاده از الگوریتم‌های ارائه‌شده پرداخته می‌شود و در بخش ۵ جمع‌بندی بیان می‌گردد.

۲- یادگیری تقویتی و روش‌های موجود برای کسب مهارت‌ها

در یادگیری تقویتی عامل با محیط در دنباله‌ای از گام‌های زمانی گسسته تعامل برقرار می‌کند. در زمان t، عامل حالت سیستم را که با s_t نشان داده می‌شود دریافت نموده و کنش a_t را از بین کنش‌های مجاز که با $A(s_t)$ نشان داده می‌شود انتخاب و به محیط اعمال می‌کند. عامل پاداش فوری r_{t+1} را دریافت می‌کند و به حالت s_{t+1} می‌رود. هر سیاست، یک نگاشت میان هر حالت و احتمال انتخاب هر کنش ممکن ایجاد می‌کند. در یادگیری تقویتی معمولاً هدف بیشینه‌کردن تابعی از پاداش در آینده است که به آن تابع ارزش می‌گوییم. تابع ارزش ممکن است با متوسط پاداش، پاداش تخفیف‌یافته و غیره مشخص شود. مقدار تابع ارزش با پاداش تخفیف‌یافته در حالت s هنگامی که سیاست π دنبال می‌شود و نرخ تخفیف γ است با رابطه زیر به دست می‌آید [۲]

$$V^{\pi}(s) = E\{r_{t+1} + \gamma r_{t+2} + \gamma^{2} r_{t+3} + \ldots \mid s_t = s, \pi\} \qquad (1)$$

تابع ارزش بهینه عبارت است از تابع ارزش سیاستی که بیشترین مقدار ارزش را در هر حالت دارد و با رابطه زیر به دست می‌آید که به رابطه بلمن مشهور است

$$V^{*}(s) = \max_{a}[R(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V^{*}(s')] \qquad (2)$$

که $0 < \gamma < 1$ نرخ تخفیف نامیده می‌شود و $V^{*}(s)$ مجموع پاداش تخفیف‌یافته حالت s با اجرای سیاست بهینه می‌باشد. اگر در حالت s باشیم و کنش a را انتخاب کنیم و سپس سیاست بهینه را پی بگیریم، می‌توانیم رابطه بلمن را برای یک تابع ارزش- کنش تعریف کنیم که در یادگیری تقویتی با Q^{*} نشان داده می‌شود

$$Q_{k+1}(s,a) = (1-\alpha_k) Q_k(s,a) + \alpha_k [r + \gamma \max_{a'} Q_k(s',a')] \qquad (3)$$

در تلاش‌های اخیر برای حل مسایل با فضای حالت بسیار بزرگ در حوزه یادگیری تقویتی روش‌های تجزیه سلسله‌مراتبی ارائه شده‌اند. چارچوب‌های مختلفی برای یادگیری تقویتی سلسله‌مراتبی ارائه شده است [۱]. چارچوب‌های معروف در این زمینه عبارتند از گزینه[2] [۸] و MAXQ [۹]. روش گزینه بیشتر از روش‌های دیگر برای استخراج ساختار سلسله‌مراتبی و مهارت‌ها استفاده شده و بنابراین در این پژوهش نیز از روش گزینه استفاده گردیده است.

هر گزینه از سه جزء تشکیل شده است. یک سیاست به صورت $\pi : S \times A \to [0,1]$، یک دامنه به صورت $I \subseteq S$ و یک شرط خاتمه به صورت $\beta : S^{+} \to [0,1]$. I مجموعه حالت‌هایی است که گزینه می‌تواند از آنها آغاز شود و ادامه پیدا کند و S مجموعه حالت‌های ممکن در محیط می‌باشد. گزینه $\langle I, \pi, \beta \rangle$ در حالت s قابل دسترسی است اگر و فقط اگر $s \in I$. اگر گزینه اجرا شود کنش‌ها بر اساس π انتخاب می‌شوند تا زمانی که گزینه به صورت تصادفی بر اساس β خاتمه یابد [۷].

برای ایجاد مهارت‌ها در پژوهش‌های پیشین روش‌های مختلفی استفاده شده است. در ادامه این بخش به معرفی برخی از این روش‌ها که به روش‌های کسب خودکار مهارت نیز معروف هستند می‌پردازیم. به طور کلی روش‌های کسب خودکار مهارت در یادگیری تقویتی را می‌توان به دو گروه اصلی تقسیم‌بندی کرد. در گروه نخست، عامل برای شکستن مسأله به مجموعه‌ای از زیرمسایل و ایجاد ساختار سلسله‌مراتبی فراکنش‌ها، زیرهدف‌ها را شناسایی می‌کند. تعاریف مختلفی از زیرهدف‌ها در مقالات متفاوت ارائه شده است. به عنوان مثال حالت‌هایی که فرکانس ملاقات بالایی دارند [۱۰] و [۱۱] یا حالت‌هایی که میان نواحی چگال در محیط واقع شده‌اند [۱۲] و [۱۳]. در گروه دوم، عامل بدون شناسایی زیرهدف‌ها و به صورت مستقیم ساختار سلسله‌مراتبی را ایجاد می‌کند. برخی روش‌های مبتنی بر گروه دوم مستقیماً مسأله اصلی را به چند زیرمسأله کوچک‌تر و ساده‌تر تقسیم می‌کنند و برای حل هر یک از آنها مهارت‌هایی تعریف می‌نمایند [۱۴] و [۱۵]. در برخی دیگر از این روش‌ها، زنجیره‌هایی از کنش‌ها در یک وظیفه یا وظایف مختلفی که عامل انجام داده استخراج شده و ویژگی‌های مشترک آنها به صورت یک مهارت برای عامل هوشمند تعریف می‌شود. در این گونه روش‌ها تأکید بر قابلیت استفاده مجدد مهارت و انتقال مهارت از یک عامل به عامل دیگر است [۸] و [۱۶].

بسیاری از روش‌های مبتنی بر زیرهدف از الگوریتم‌های گراف برای تشخیص زیرهدف‌ها استفاده می‌کنند. در این روش‌ها، ابتدا کنش‌های انجام‌شده توسط عامل و حالت‌های بازدیدشده در محیط به یک گراف نگاشت شده و سپس گره‌های گلوگاه در گراف متناظر به عنوان زیرهدف یا هدف میانی شناسایی می‌شود. هر حالت بازدیدشده متناظر یک گره در گراف و هر کنش که باعث تغییر وضعیت عامل از یک حالت به حالت دیگری می‌شود، متناظر یک یال از گراف در نظر گرفته می‌شود. این روش‌ها را می‌توان به سه دسته روش‌های مبتنی بر برش گراف، روش‌های مبتنی بر خوشه‌بندی گراف و روش‌های مبتنی بر مرکزیت میانگی تقسیم‌بندی کرد.

در روش‌های مبتنی بر برش، یال‌هایی از گراف که بیشترین جریان از آنها می‌گذرد به عنوان زیرهدف‌ها شناسایی می‌شود. الگوریتم برش Q [۱۷] یک الگوریتم مبتنی بر برش است که در آن از الگوریتم بیشترین جریان/ کمترین برش[3] برای شناسایی گلوگاه‌ها یا زیرهدف‌ها در گراف استفاده شده است. عامل در حالی که با محیط ارتباط برقرار می‌کند تاریخچه انتقال حالت‌ها را ذخیره می‌نماید. هر وقت شرایط برش مهیا

1. Experience Replay

2. Option

3. Maxflow/Mincut

باشد دو حالت را انتخاب و یک افراز برش مینیمم میان آنها پیدا می‌کند. اگر کیفیت برش خوب بود، یک گزینه برای رسیدن به محل افراز برش‌ها (یا گلوگاه‌های به دست آمده) یاد می‌گیرد. در الگوریتم NCut که توسط شیمشک[1] ارائه شد، برش‌هایی شناسایی می‌شود که گراف را به دو بخش که احتمال انتقال بین دو خوشه کمی دارند، تقسیم می‌کنند. حالت‌هایی که در مرز برش‌های انجام‌شده هستند به عنوان حالت‌های دستیابی[2] یا همان حالت‌های زیرهدف برچسب می‌خورند [۱۸].

در روش‌های مبتنی بر خوشه‌بندی، نواحی همبند گراف با استفاده از روش‌های خوشه‌بندی گراف شناسایی می‌شوند و سپس یال‌هایی از گراف را که یک گره آنها در یک خوشه گراف و گره دیگر آن در ناحیه دیگر قرار دارد را به عنوان گلوگاه شناسایی کرده و گره‌های متناظر آن به عنوان هدف میانی شناسایی می‌شوند. مانور[3] روشی مبتنی بر خوشه‌بندی برای تولید خودکار گزینه ارائه داده است. در این الگوریتم در ابتدا هر حالت را در یک خوشه قرار می‌دهد. سپس همه زوج خوشه‌های همسایه را در نظر گرفته و زوجی که الحاق آنها بیشترین بهبود را در معیار مکاشفه‌ای ایجاد کنند ادغام می‌کند [۱۲]. در روش دیگری که توسط متاو[4] ارائه شده است ارتباط عامل با محیط با یک سیستم دینامیکی مدل می‌شود و شناسایی نواحی با پایداری پایین منجر به بخش‌بندی فضای حالت می‌گردد. انتقال میان حالت‌های ناپایدار به عنوان مهارت‌ها شناخته می‌شوند [۱۹]. چیو[5] و سو[6] نظریه گراف طیفی[7] را برای تعیین زیرهدف‌ها به کار گرفتند و با کمک ویژگی همواری[8] طیفی، یال‌هایی که اختلاف تحلیل طیفی آنها ماکسیمم محلی را دارند به عنوان یال‌های گلوگاه و گره‌های اتصال‌دهنده آنها را حالت‌های گلوگاه معرفی کردند [۲۰]. متزن[9] یک روش افزایشی برای کسب مهارت‌ها با استفاده از خوشه‌بندی سلسله‌مراتبی گراف انتقال تخمین زده شده ارائه کرد و آن را به محیط پیوسته توسعه داد [۲۱]. بیکن[10] و همکاران از الگوریتم انتشار برچسب برای تشخیص خوشه‌ها به صورت افزایشی استفاده کردند. در این روش خوشه‌ها بعد از تعاملات زیاد عامل با محیط (۱۰۰ سناریو برای یک محیط کوچک ۴اتاقه) از گراف انتقال ساخته می‌شوند. همچنین خوشه‌های به دست آمده حتی برای محیط‌های کوچک دقیق نیستند (۹ خوشه در محیط ۴اتاقه). این روش به فضای حالت پیوسته تعمیم داده شده است [۲۲]. داودآبادی و همکاران یک الگوریتم جدید تشخیص خوشه‌ها در گراف ارائه دادند که با کمک آن زیرهدف‌ها در یادگیری تقویتی تشخیص داده می‌شوند. در این مقاله، الگوریتم انتشار برچسب توسعه داده شده است به گونه‌ای که خوشه‌های کوچک‌تر برای تعیین خوشه‌های واقعی با همدیگر ادغام می‌گردند [۲۳].

در روش‌های مبتنی بر مرکزیت، گره‌های گراف بر اساس میزان اهمیت آنها در گراف رتبه‌بندی می‌شوند و زیرهدف‌ها، گره‌های با رتبه‌بندی بالاتر هستند. شیمشک از معیاری به نام مرکزیت میانگی[11] برای پیداکردن گره‌های مرکزی گراف استفاده کرد [۲۴]. راد و همکاران در [۲۵] از معیار مرکزیت پایداری برای استخراج زیرهدف‌ها استفاده کردند.

1. Simsek
2. Access State
3. Mannor
4. Mathew
5. Chiu
6. Soo
7. Spectral Graph Theory
8. Smoothness
9. Metzen
10. Bacon
11. Betweenness Centrality

۳- روش پیشنهادی

در بخش قبل مطرح گردید که در برخی روش‌های کسب مهارت مستقیماً مسأله اصلی به چند زیرمسأله کوچک‌تر و ساده‌تر تقسیم می‌شود و در این روش‌ها فرض بر این است که مسأله دارای ساختار مشخصی می‌باشد. اما این موضوع در مورد همه مسایل ممکن است صادق نباشد. در برخی دیگر از روش‌ها، زنجیره‌هایی از کنش‌ها در یک وظیفه یا وظایف مختلفی که عامل انجام داده استخراج شده و ویژگی‌های مشترک آنها به صورت یک مهارت برای عامل هوشمند تعریف می‌شود. در این گونه روش‌ها نیاز به انجام سناریوهای قبلی زیاد تراکنش با محیط وجود دارد. این اشکال در روش‌های مبتنی بر تکرار و مبتنی بر برش نیز وجود دارد. روش‌های مبتنی بر مرکزیت، عملکرد بهتری در پیداکردن زیرهدف‌ها داشته‌اند ولی برای پیداکردن دامنه مهارت‌ها دچار مشکل هستند. با کمک خوشه‌بندی گراف، پیداکردن دامنه مهارت‌ها به راحتی می‌تواند انجام شود [۲۳]. همچنین این روش‌ها دقت بالایی در پیداکردن زیرهدف‌ها داشته‌اند. بنابراین در این مقاله، روشی مبتنی برخوشه‌بندی گراف برای پیداکردن زیرهدف‌ها ارائه می‌گردد. اشکال روش‌های پیشین استفاده از خوشه‌بندی گراف برای تعیین زیرهدف‌ها، پیچیدگی زمانی زیاد آنها می‌باشد [۲۱]. بنابراین در این مقاله تلاش کرده‌ایم روشی برای خوشه‌بندی گراف با پیچیدگی کم (خطی) ارائه نماییم.

در این بخش یک الگوریتم جدید برای کسب مهارت‌ها ارائه می‌گردد. در این روش، عامل پس از چند دوره اجرای یک الگوریتم یادگیری تقویتی در محیط به ساخت گراف انتقال متناظر با تاریخچه انتقالات می‌پردازد و با کمک روش‌های مبتنی بر خوشه‌بندی، زیرهدف‌ها را در گراف گذر عامل استخراج می‌نماید و بدین منظور یک الگوریتم جدید خوشه‌بندی گراف ارائه گردیده است. در این مقاله مهارت‌ها با کمک چارچوب گزینه ایجاد می‌گردند و روش‌هایی برای ارزیابی آنها ارائه شده است. در این بخش درباره نحوه تولید مهارت‌ها، استخراج زیرهدف‌ها و ارزیابی مهارت‌ها بحث خواهد شد.

۳-۱ ساخت مهارت‌ها

شکل ۱ الگوریتم جدید پیشنهادی برای کسب مهارت‌ها را نشان می‌دهد. نخست عامل طی k دوره به اکتشاف در محیط و یادگیری با کمک یادگیری Q می‌پردازد. از تراکنش‌های عامل با محیط یک گراف ساخته می‌شود و هر حالت ملاقات‌شده به یک گره و هر انتقال میان حالت‌ها به یک یال تبدیل می‌شود. سپس یک الگوریتم خوشه‌بندی بر روی گره‌های گراف اعمال می‌گردد و زیرهدف‌ها پیدا می‌شوند. گره‌های موجود در مرز خوشه‌ها، زیرهدف‌ها را تشکیل می‌دهند. پس از خوشه‌بندی گراف تراکنش عامل با محیط، برای رفتن از هر خوشه به هر خوشه همسایه یک گزینه ساخته می‌شود. همه گره‌های درون یک خوشه در دامنه گزینه مربوط به رسیدن به نواحی مرزی خوشه‌ها جای داده می‌شوند. احتمال خاتمه گزینه در دامنه صفر و در بقیه حالت‌ها یک است. با کمک روش بازنمایی تجربه [۲۶] از تجارب به دست آمده در k دوره تراکنش عامل با محیط برای یادگیری اولیه سیاست‌های گزینه‌ها استفاده می‌گردد. با دادن یک پاداش مجازی[12] به زیرهدف‌ها، هر گزینه می‌تواند

12. Pseudo Reward

1. *Initialization:*
   $s_t \leftarrow$ initial state, $Q(s,a) \leftarrow 0$ for all state s and action a, Graph $\leftarrow \{\}$, $O \leftarrow \{\}$, $A \leftarrow$ action set
2. *Repeat for k episodes: (each episode started from initial state and stopped at goal state or after performing Maximum steps)*
   a. Choose $a_t \in A$ using ε-greedy policy w.r.to Q ($a_t = \text{argmax}_{a \in A} Q(s_t,a)$ (ε-greedy)).
   b. Apply a_t (nondeterministic) and set r_t to the reward for transition $s_t, a_t \rightarrow s_{t+1}$.
   c. Add s_t and s_{t+1} to Graph Vertices and $s_t \rightarrow s_{t+1}$ to Graph Edges or Update the weight for $s_t \rightarrow s_{t+1}$ (If $s_t \rightarrow s_{t+1}$ exists in Graph)
   d. Update $Q(s_t, a_t) \leftarrow (1-\alpha_k)\ Q(s_t, a_t) + \alpha_k\ (r_t + \gamma \max_{a \in A} Q(s_{t+1}, a))$
   e. Set $s_t \leftarrow s_{t+1}$
3. *Run a graph clustering algorithm on Graph and create a set of clusters C .*
4. *For each pair of neighbor clusters c_i and $c_j \in C$:*
   a. Create an option o_{ij} for reaching from cluster c_i to c_j .
      $I_{oij} \leftarrow$ all states in c_i , β_{oij} (nodes belong to c_i and connected to a node in c_j)=1
   b. Create an option o_{ji} for reaching from cluster c_j to c_i .
      $I_{oji} \leftarrow$ all nodes in c_j , β_{oji} (nodes belong to c_j and connected to a node in c_i)=1
   c. Add o_{ij} and o_{ji} to O.
5. *Define a QTable, Q^o, for each option o. Use Experience Replay for learning option's policies.*
6. *Evaluate each option $o \in O$. Remove wrong options from O.*
7. *Repeat for m episodes: (each episode started from initial state and stopped at goal state)*
   a. Choose $a_t \in A \cup O$ using ε-greedy policy w.r.to Q ($a_t \leftarrow \text{argmax}_{a \in A \cup O} Q(s_t,a)$ (ε-greedy)).
   b. If a_t is an option
      $O \leftarrow a_t$
      Select an action $a_t \in A$ using ε-greedy according to Q^o ($a_t = \text{argmax}_{a \in A} Q^o(s_t,a)$ (ε-greedy))
      inOption=true
      NumActionsDoneInOption=0
      cumulativeReward=0
   c. Apply a_t (nondeterministic) and set r_t to the reward for transition $s_t, a_t \rightarrow s_{t+1}$.
   d. if (inOption==true)
      NumActionsDoneInOption++
      cumulativeReward+=reward* $\gamma^{\text{NumActionsDoneInOption-1}}$
      Add (s_t, a_t, s_{t+1}, r_t) to OptionExperience
   e. Add s_t and s_{t+1} to Graph Vertices and $s_t \rightarrow s_{t+1}$ to Graph Edges or Update the weight for $s_t \rightarrow s_{t+1}$ (If $s_t \rightarrow s_{t+1}$ exists in Graph)
   f. For each option o in O
      If $s_t \in I^o$ and $s_{t+1} \in I^o$
         $Q^o(s_t, a_t) \leftarrow (1-\alpha_k)\ Q^o(s_t, a_t) + \alpha_k\ (\text{PseudoReward}_t + \gamma \max_{a \in A} Q^o(s_{t+1}, a))$
         // PseudoReward_t is equal to a positive value for option's subgoal and 0 otherwise.
   g. if (inOption==true & $\beta_o(s_t)$==1) //end of option
      $Q(s_t, o) \leftarrow (1-\alpha_k)\ Q(s_t, o) + \alpha_k\ (\text{cumulativeReward} + \gamma^{\text{NumActionsDoneInOption}} \max_{a \in A \cup O} Q(s_{t+1}, a))$
      Update Q(s , o) for each state ,s, in OptionExperience
      //if (!inOption || (inOption && intraOptionLearning)
   h. Update $Q(s_t, a_t) \leftarrow (1-\alpha_k)\ Q(s_t, a_t) + \alpha_k\ (r_t + \gamma \max_{a \in A \cup O} Q(s_{t+1}, a))$
   i. Update $Q(s_t, o)$ using IntraOptionLearning
   j. Set $s_t \leftarrow s_{t+1}$

شکل ۱: الگوریتم ساخت گزینه‌ها.

به شکل یک زیرمسأله دیده شود. پس از استخراج گزینه‌ها، عامل به ارزیابی آنها می‌پردازد و گزینه‌های مناسب برای حل مسأله را انتخاب خواهد نمود. روش‌هایی برای ارزیابی گزینه‌ها در فصل ۳ ارائه خواهد شد. در دوره‌های بعدی تراکنش عامل با محیط، عامل می‌تواند از گزینه‌های مناسب برای رسیدن به هدف استفاده کند و البته سیاست گزینه‌ها نیز هم‌زمان با سیاست وظیفه اصلی یاد گرفته شود. همچنین از یادگیری درون‌گزینه‌ای[1] [۲۷] برای یادگیری مقادیر ارزش گزینه‌ها استفاده می‌شود.

در این مقاله از چند الگوریتم خوشه‌بندی سلسله‌مراتبی برای تشخیص زیرهدف‌ها استفاده شده است. ابتدا به بررسی دو روش خوشه‌بندی با کمک معیار مرکزیت یال و خوشه‌بندی با کمک انتشار برچسب‌ها می‌پردازیم. روش اول یک الگوریتم خوشه‌بندی بالا به پایین و روش دوم یک الگوریتم خوشه‌بندی پایین به بالا است. سپس الگوریتم پیشنهادی جدیدی برای تشخیص خوشه‌ها در گراف ارائه خواهیم کرد که با الگوریتم‌های قبلی مقایسه خواهد شد.

1. Intra Option Learning

1. Give a unique label to each node of graph.
2. Update each node's label by choosing the label which most of its neighbors have.
3. If every node has a label that the maximum number of their neighbors have, go to 4 Else go to step 2
4. For each pair of clusters calculate ΔM(i,j) which is indicated the change in M upon joining clusters i and j.
5. Find the pair of clusters with maximum ΔM value and called them C1, C2. If ΔM(C1,C2) < 0 stop the algorithm.
6. Merge (C1;C2).
7. Go to step 4.

شکل ۲: الگوریتم پیشنهادی خوشه‌بندی گراف (MBLP).

۳-۱-۱ استفاده از خوشه‌بندی با کمک معیار مرکزیت برای ساخت گزینه‌ها

در الگوریتم مرکزیت یال [۲۸]، نخست یالی از گراف که بیشترین مقدار مرکزیت را دارد حذف می‌شود. در گراف حاصل این کار تکرار شده و یالی با بیشترین مرکزیت پیدا و حذف می‌گردد که این کار باعث شکسته‌شدن گراف به چند خوشه خواهد شد. مرکزیت پس از حذف هر یال، برای همه یال‌های گراف دوباره محاسبه می‌شود و هر بار از یک معیار استحکام خوشه‌بندی برای تعیین کیفیت خوشه‌بندی حاصل استفاده می‌گردد که در ادامه بیشتر توضیح داده می‌شود. مرکزیت یک یال برابر است با تعداد کوتاه‌ترین مسیرهای بین هر دو گره موجود در گراف که از آن یال عبور می‌کند.

در این الگوریتم از یک معیار ارزیابی به نام ماژولاریتی[1] برای انتخاب تعداد خوشه‌هایی که گراف باید به آن تقسیم گردد استفاده شده است. فرض کنید یک افراز خاص از گراف به k خوشه را داریم. برای محاسبه ماژولاریتی یک ماتریس متقارن e با ابعاد $k \times k$ تعریف می‌شود که المان‌های آن e_{ij} نسبت یال‌هایی است که گره‌های خوشه i را به خوشه j متصل می‌کند. $a_i = \sum_j e_{ij}$ نسبت یال‌هایی را نشان می‌دهد که گره‌ها در خوشه i را به گره‌ای در خودش یا سایر خوشه‌ها متصل می‌کند. با این متغیرها یک معیار ماژولاریتی به شکل زیر تعریف می‌گردد

$$M = \sum_i (e_{ii} - a_i^2) \qquad (۴)$$

اگر تعداد یال‌های درون خوشه بهتر از حالت تصادفی نباشد مقدار M برابر صفر می‌شود و اگر گراف ساختار خوشه‌بندی قوی داشته باشد مقدار M به یک نزدیک می‌شود. با افزایش تعداد یال‌های حذف‌شده تعداد خوشه‌های به دست آمده نیز بیشتر می‌گردد. در الگوریتم مرکزیت یال، M برای هر بخش‌بندی گراف محاسبه می‌شود و الگوریتم به دنبال بیشینه‌های محلی این مقدار می‌گردد. بخش‌بندی‌ای که مقدار M آن بیشینه محلی داشته باشد به تجزیه مورد نظر نزدیک خواهد بود. پیچیدگی این الگوریتم در بدترین حالت $O(m^2 n)$ یا در ماتریس خلوت $O(n^2)$ است [۲۸].

۳-۱-۲ استفاده از خوشه‌بندی با کمک روش انتشار برچسب‌ها

الگوریتم خوشه‌بندی با کمک انتشار برچسب‌ها [۲۹] را به صورت زیر می‌توان خلاصه کرد. هر گره گراف در ابتدا یک برچسب واحد می‌گیرد و برچسب هر گره با دیگری متفاوت است. در هر تکرار، هر گره با انتخاب برچسبی که بیشتر همسایه‌هایش آن را دارند به روز می‌شود. اگر چند برچسب تعداد تکرار یکسانی در همسایگی گره داشته باشند یکی از آنها به صورت تصادفی انتخاب می‌گردد. این کار تا زمانی تکرار می‌شود که هر گره برچسبی داشته باشد که بیشتر همسایه‌هایش آن را دارند. پیچیدگی این الگوریتم خطی است و نتایج خوبی از این روش گزارش شده است. در هر تکرار، همسایه‌های هر گره بررسی می‌گردد و برچسبی که در بین آنها بیشترین تکرار را داشته انتخاب می‌گردد. اگر k تعداد تکرارها، n تعداد گره‌ها، m تعداد یال‌ها و d درجه میانگین گره‌ها فرض شود، زمان اجرای این الگوریتم $O(knd)$ است که nd می‌تواند همان m باشد [۲۹].

۳-۱-۳ الگوریتم پیشنهادی برای کشف زیرهدف‌ها

در این بخش ابتدا درباره نقاط ضعف الگوریتم‌های بخش پیش بحث می‌شود، سپس یک الگوریتم جدید خوشه‌بندی گراف ارائه می‌دهیم و از آن برای پیداکردن زیرهدف‌ها و ساخت گزینه‌ها استفاده می‌کنیم. الگوریتم مرکزیت یال که پیش از این توضیح داده شد، توسط بسیاری از محققان مورد توجه قرار گرفته و کارایی خوبی در بسیاری از گراف‌ها برای خوشه‌بندی داشته است. همان طور که در بخش بعد نشان داده خواهد شد، این الگوریتم در آزمایشات انجام‌شده در این مقاله نیز نتایج خوبی نشان داده است اما این الگوریتم دارای مشکل پیچیدگی محاسباتی بالا است. پیچیدگی الگوریتم انتشار برچسب تقریباً خطی است ولی برخی نتایج تجربی نشان داد که این الگوریتم نمی‌تواند خوشه‌ها را در گراف‌های مربوط به محیط شبکه مشبک تشخیص دهد [۲۳]. شکل ۲ الگوریتم پیشنهادی برای خوشه‌بندی را نشان می‌دهد که آن را الگوریتم انتشار برچسب مبتنی بر ماژولاریتی نامیدیم (MBLP)[2]. در این الگوریتم ابتدا روش انتشار برچسب بر روی گراف اعمال می‌شود. پس از توقف انتشار برچسب، گره‌هایی که برچسب یکسانی دارند در یک خوشه قرار می‌گیرند. برای ساخت خوشه‌های واقعی نیاز به ادغام این خوشه‌ها وجود دارد و بدین منظور از معیار ماژولاریتی استفاده کردیم. مفهوم ماژولاریتی از چنان محبوبیتی برخوردار شده که نه تنها به عنوان معیاری برای خوشه‌بندی یک شبکه استفاده می‌شود بلکه به عنوان معیار کیفیت در الگوریتم‌های تشخیص خوشه مختلف نیز به کار می‌رود [۳۰].

در الگوریتم نشان داده شده در شکل ۲ خوشه‌ها مکرراً در هم ادغام می‌شوند. برای ادغام هر بار دو خوشه‌ای انتخاب می‌شود که بیشترین افزایش را در ماژولاریتی ایجاد کند. تغییرات در ماژولاریتی بعد از الحاق دو خوشه i و j با رابطه زیر قابل محاسبه است [۳۰]

$$\Delta M(i,j) = e_{ij} + e_{ji} - 2a_i a_j = 2(e_{ij} - a_i a_j) \qquad (۵)$$

بنابراین خوشه‌ها به طور حریصانه ترکیب می‌شوند تا ماژولاریتی بیشینه گردد. مراحل پیشرفت الگوریتم می‌تواند با یک درخت نشان داده شود که ترتیب ادغام را نشان می‌دهد. برش‌ها در سطوح مختلف این درخت، منجر به تقسیم گراف به تعدادی خوشه می‌شود و می‌توانیم بهترین برش را با مقدار بیشینه M انتخاب کنیم. پیچیدگی زمانی مراحل ۱ تا ۳ این الگوریتم مانند الگوریتم انتشار برچسب $O(mk)$ است که m تعداد یال‌ها و k تعداد مراحل مورد نیاز برای الگوریتم انتشار برچسب است. به منظور الحاق خوشه‌ها در مراحل ۴ تا ۷ فقط نیاز به در نظر گرفتن خوشه‌هایی که میان آنها یال هست وجود دارد که تعداد آنها در هر مرحله از ادغام کمتر از m است. تعداد مراحل ادغام کمتر از $L-1$ می‌باشد که

1. Modularity

2. Modularity Based Label Propagation

L تعداد خوشه‌های تشخیص داده شده در گام‌های ۱ تا ۳ است. روشن است که L خیلی کمتر از n است. مقدار ΔM می‌تواند در زمان ثابت محاسبه شود. بنابراین پیچیدگی زمانی الگوریتم $O(mk+mL)$ است. این الگوریتم می‌تواند در تشخیص خوشه‌های شبکه‌های نسبتاً بزرگ با سرعت بالایی استفاده شود.

۳-۲ ارزیابی مهارت‌ها

در [۶] و [۷] نشان داده شد که ایجادکردن مهارت‌ها و اضافه‌کردن آنها به مسأله گاهی ممکن است پیچیدگی یادگیری را افزایش دهد و بنابراین ارزیابی اثر هر مهارت بر کارایی عامل یادگیر ضروری است. انجام این کار یعنی اندازه‌گیری کارایی عامل یادگیر کار ساده‌ای نیست. در بسیاری از کارهای گذشته سرعت همگرایی به سیاست بهینه یا تقریباً بهینه معمولاً برای اندازه‌گیری کارایی الگوریتم یادگیری تقویتی استفاده می‌شود. روش‌های متفاوتی برای رسیدن به این هدف ارائه شده است. برای مثال برخی روش‌ها تعداد کنش‌های انجام‌شده توسط عامل برای رسیدن به حالت نهایی را رسم می‌نمایند و برخی روش‌های دیگر میزان پاداش به دست آمده بعد از تعدادی دوره یادگیری را محاسبه می‌کنند [۳۱]. این روش‌ها برای ارزیابی الگوریتم‌های یادگیری تقویتی سلسله‌مراتبی و الگوریتم‌های استخراج خودکار انتزاع‌های زمانی نیز به کار می‌روند. سرعت یادگیری این الگوریتم‌ها معمولاً با روش‌های یادگیری تقویتی مسطح مانند یادگیری Q مقایسه می‌شود. یکی از اشکالات این روش‌ها این است که سودمندی همه مهارت‌ها با هم در نظر گرفته می‌شود و کارایی هر مهارت به تنهایی روشن نیست. متأسفانه در پژوهش‌های پیشین به مسأله ارزیابی هر مهارت مستقل از مهارت‌های دیگر پرداخته نشده است.

در ادامه این بخش سعی می‌کنیم روش‌هایی برای تشخیص مهارت‌های نامفید ارائه کنیم. مهارت‌های نامفید به عامل برای پیداکردن سیاست بهینه کمک نمی‌کنند و بنابراین اضافه‌کردن آنها به انتخاب‌های عامل فضای حالت را بزرگ می‌کند. در نتیجه زمان بیشتری برای جستجو در آن صرف می‌گردد و پیداکردن سیاست بهینه سخت‌تر خواهد شد. از این رو حذف مهارت‌های نامفید از انتخاب‌های عامل سرعت یادگیری را افزایش می‌دهد. در این بخش، دو روش برای ارزیابی مهارت‌ها ارائه خواهیم کرد. در روش اول تابع ارزش هر مهارت با تابع ارزش کنش‌های پایه موجود در هر حالت مقایسه می‌شود. در روش دوم دامنه مهارت با حالت‌هایی که در سناریوهای پیشین، قبل از زیرهدف دیده شده‌اند مقایسه می‌گردد.

۳-۲-۱ مقایسه مقادیر ارزش مهارت با کنش‌های پایه

با فرض این که چارچوب گزینه برای نمایش مهارت‌ها به کار گرفته شود، اگر از یک سیاست SMDP استفاده شود، در هر حالت، عامل می‌تواند از میان گزینه‌ها و کنش‌های پایه موجود در آن حالت بر اساس سیاستش انتخاب کند. بر اساس معادله بلمن برای Q^* تابع ارزش- کنش برای هر کنش پایه یا گزینه با معادله زیر محاسبه می‌گردد

$$Q^*(s,o) = R(s,o) + \sum_{s',k} P(s',k|s,o)\gamma^k \max_{a'} Q^*(s',a') \qquad (۶)$$

که k یک متغیر تصادفی است که زمان انتظار را نشان می‌دهد وقتی کنش o از حالت s شروع می‌شود. اگر k برابر یک باشد، o کنش پایه است. مقادیر بالاتر k نشان می‌دهد که o گزینه است. $P(s',k|s,o)$ احتمال انتقال از حالت s به حالت s' بعد از k گام است وقتی کنش o اجرا می‌شود. $\gamma \in [0,1]$ پارامتر نرخ تخفیف است و $Q^*(s,o)$ مقدار تابع ارزش- کنش بهینه برای حالت s و کنش o می‌باشد. $R(s,o)$ پاداش مورد انتظار تخفیف‌یافته در زمان اجرای کنش o در حالت s است.

اگر در هر حالت از دامنه گزینه، مقدار ارزش گزینه بزرگ‌تر از همه کنش‌های پایه باشد نتیجه گرفته می‌شود که گزینه در آن حالت مفید است. در این حالت، معیار مورد نظر برای مفیدبودن گزینه o در حالت‌های D_O برابر است با

$$\forall s \in D_O \ , a \in A_s : Q^*(s,o) \geq Q^*(s,a) \qquad (۷)$$

که o، D_O و A_s گزینه ارزیابی‌شده، مجموعه دامنه این گزینه و کنش‌های پایه موجود در حالت s است. $Q^*(s,o)$ تابع ارزش- کنش برای گزینه o در حالت s و $Q^*(s,a)$ تابع ارزش- کنش برای کنش a در حالت s می‌باشد.

از آنجایی که سیاست بهینه می‌تواند با انتخاب میان کنش‌های پایه و گزینه‌ها که تابع ارزش- کنش را بهینه می‌سازد به دست آید، (۷) می‌تواند به درستی اهمیت یک گزینه را مشخص نماید. از آنجایی که تابع ارزش- کنش تخمین زده شده و مقدار واقعی آن ممکن است هنوز یاد گرفته نشده باشد، نمی‌توانیم انتظار داشته باشیم که در تراکنش‌های اولیه یادگیری بعد از اضافه‌شدن گزینه این معادله برای همه حالت‌های دامنه درست باشد. بنابراین می‌توان ارزیابی را به بعد از دوره‌های یادگیری کافی به تعویق انداخت یا با ساده‌ترکردن شرط معادله انتظار داشته باشیم که معادله برای حداقل نیمی از حالت‌های دامنه گزینه (به جای همه آنها) درست باشد.

۳-۲-۲ استفاده از مسیرهای قبلی برای رسیدن به زیرهدف

در این بخش روشی برای شناسایی مهارت‌ها یا گزینه‌هایی که عامل را از هدف دور می‌کنند ارائه می‌کنیم. زیرهدف‌ها را به عنوان حالت‌هایی تعریف می‌کنیم که عامل برای رسیدن به هدف می‌بایست از آنها عبور کند. بنابراین اگر حالت‌های آغازین یک گزینه معمولاً قبل از زیرهدف در مسیر رسیدن به هدف دیده شوند گزینه برای رسیدن به آن هدف مناسب است.

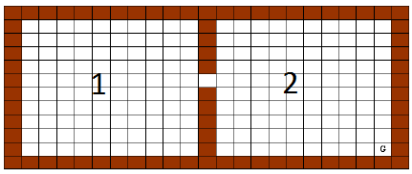
برای استفاده از این ایده ارزیابی، دوره‌های یادگیری قبلی که در آن زیرهدف دیده شده است بررسی می‌گردند و همه حالت‌هایی که حداکثر فاصله‌شان تا زیرهدف d است جمع‌آوری می‌گردند. به طور کلی اگر یک دوره یادگیری را به شکل زیر فرض کنیم

$$S_1, S_2, S_3, \ldots, S_k, S_{k+1}, \ldots, S_m, \ldots, S_G$$

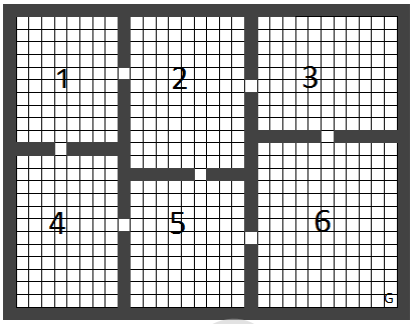
که در آن S_t حالتی که در زمان t دیده شده است، S_m زیرهدف و $m-k=d$ باشد، همه حالت‌های میان S_k و S_m جمع‌آوری می‌گردند و در یک مجموعه قرار می‌گیرند که آن را مجموعه حالت‌های مرتبط (RSS)[1] می‌نمایم. اگر تعداد حالت‌های مشترک میان دامنه یک گزینه و مجموعه حالت‌های مرتبط (که با تعداد حالت‌های دامنه گزینه نیز نرمال شده باشند) از یک آستانه بزرگ‌تر باشد، گزینه احتمالاً در مسیر هدف قرار دارد. در نتیجه ما معیار زیر را برای ارزیابی گزینه به کار می‌گیریم

$$\frac{N(s \in D_O \text{ and } s \in RSS_O)}{N(s \in D_O)} > t_o \qquad (۸)$$

1. Related States Set



شکل ۳: محیط مشبک دواتاقه.



شکل ۴: محیط مشبک شش‌اتاقه [۲۴].

که $N(s \in D_O)$ اندازه دامنه گزینه و $N(s \in D_O \text{ and } s \in RSS_O)$ تعداد حالت‌های دامنه گزینه است که در RSS هم وجود دارند و t_o یک آستانه است. مقدار پارامتر d می‌تواند با قطر خوشه مربوط به دامنه تخمین زده شود. قطر یک خوشه حداکثر فاصله (عدم شباهت) دو نود خوشه است [۳۲].

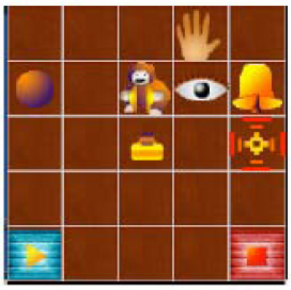
٤- نتایج آزمایش‌های تجربی برای ارزیابی روش‌های پیشنهادی

در این بخش نخست چند محیط آزمایشگاهی که در پژوهش‌های پیشین برای نمایش میزان کارایی روش‌های سلسله‌مراتبی استفاده شده‌اند معرفی می‌شوند. سپس نتایج آزمایشات انجام‌شده برای ارزیابی روش‌های پیشنهادی و تأثیر آنها بر سرعت یادگیری بررسی می‌گردند.

٤-١ محیط‌های آزمایشگاهی

برای ارزیابی الگوریتم‌های اکتساب مهارت در یادگیری تقویتی در پژوهش‌های پیشین معمولاً تعدادی محیط آزمایشی معرفی شده است. برخی از این محیط‌ها که نتیجه آزمایشات بر روی آنها نشان داده خواهد شد عبارتند از محیط اتاق‌های مشبک، محیط بازی و برج‌های هانوی. در هر کدام از این آزمایشات، عامل تعدادی وظیفه را در هر کدام از این محیط‌ها انجام می‌دهد و هر وظیفه را در دوره‌های مختلف تکرار می‌کند.

محیط‌های مشبک از تعدادی اتاق تشکیل شده‌اند که کف هر اتاق کاشی فرش شده است. بین برخی از اتاق‌ها در وجود دارد. هر کدام از کاشی‌های این اتاق‌ها یک حالت در نظر گرفته می‌شود. عامل در یکی از حالت‌های این محیط‌ها که به صورت تصادفی انتخاب شده است قرار داده می‌شود و سپس از او خواسته می‌شود تا به خانه هدف که یکی دیگر از حالت‌های این محیط‌ها است برسد. عامل می تواند چهار کنش پایه‌ای "حرکت به سمت شمال"، "حرکت به جنوب"، "حرکت به شرق" و "حرکت به غرب" را انجام دهد. در گراف متناظر این محیط هر خانه یا حالت نمایانگر یک گره است که به خانه‌های همسایه خود با یک یال متصل می‌گردد. محیط‌های مشبک دواتاقه و شش‌اتاقه در شکل‌های ۳ و ۴ نشان داده شده‌اند.



شکل ۵: محیط بازی [۳۳].

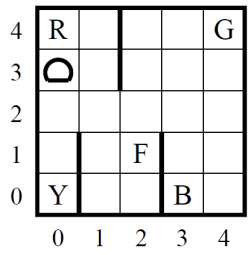
شکل ۶: برج هانوی [۱۸].

محیط بازی[1] که در شکل ۵ نشان داده شده است، توسط بارتو[2] و همکاران ارائه شد [۳۳]. در این محیط پیچیده، یک عامل با اشیای اطراف خود در تعامل است. اشیای این محیط عبارتند از ۱) یک سوئیچ برق برای خاموش و روشن‌کردن لامپ، ۲) توپ، ۳) زنگ، ۴) دو بلاک که در واقع کلیدهایی هستند برای خاموش و روشن‌کردن موسیقی و ۵) عروسک میمون. عامل نیز دارای چشم، دست و یک نشانه‌گذار است. برخی کنش‌هایی که عامل در محیط می‌تواند انجام می‌دهد شامل موارد زیر است: نگاه‌کردن به یک شیء که به صورت تصادفی انتخاب شده است، نگاه‌کردن به شیئی که در دست قرار دارد، نگهداری شیئی که به آن نگاه می‌کند، نگاه‌کردن به اشیایی که نشانه‌گذاری شده‌اند، نشانه‌گذاری شیئی که به آن نگاه می‌کند، قراردادن شیئی که در دست قرار دارد در جایی که به آن نگاه می‌کند، روشن و خاموش‌کردن چراغ برق، روشن و خاموش‌کردن موسیقی، شوت‌کردن توپ به سمت نشانه‌گذار.

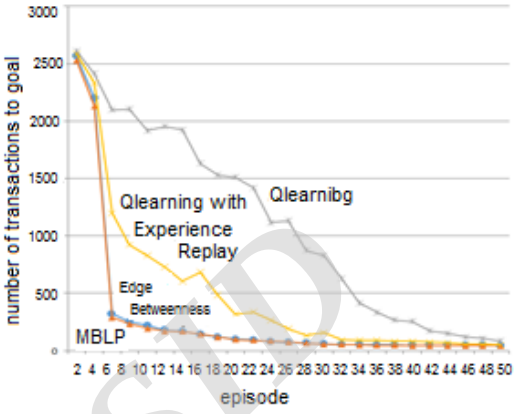
اگر هم چشم به شیئی نگاه کند و هم دست بر روی آن باشد، کنش‌های مخصوص آن شیء قابل انجام است. به عنوان مثال اگر تمرکز چشم و دست بر روی کلید برق باشند کنش زدن کلید برق قابل انجام است و اگر تمرکز چشم و دست بر روی توپ باشد کنش شوت‌کردن توپ قابل انجام می‌باشد. قبل از روشن‌کردن موسیقی، چراغ برق باید روشن شده باشد. اگر توپ به زنگ برخورد کرد، شروع به زنگ‌زدن می‌کند. در صورتی که موسیقی روشن باشد و زنگ به صدا در بیاید و اتاق تاریک باشد، میمون شروع به گریه می‌کند و در صورتی که که موسیقی خاموش شود، میمون ساکت خواهد شد. وظیفه عامل این است که میمون را وادار به گریه کند و سپس کاری کند که میمون ساکت شود. برای نمایش این محیط از مجموعه‌ای از متغیرهای حالت استفاده می‌شود. این متغیرها شامل اطلاعات مربوط به شیئی که عامل به آن نگاه می‌کند، شیئی که در دست عامل قرار دارد، شیئی که عامل با ماژیک آن را نشانه کرده است، وضعیت موسیقی، وضعیت چراغ و وضعیت زنگ است.

مسأله برج‌های هانوی که در شکل ۶ نمایش داده شده است، شامل سه میله و تعدادی دیسک با اندازه‌های مختلف است. وظیفه عامل این است که تمامی دیسک‌ها را از یکی از میله‌ها به میله دیگر منتقل کند (مثلاً میله اول به سوم). در هر حرکت، عامل می‌تواند یک دیسک را از

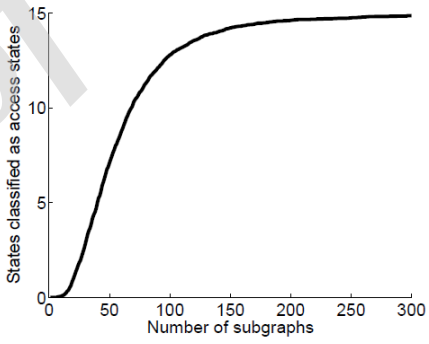
1. Playroom Domain
2. Barto



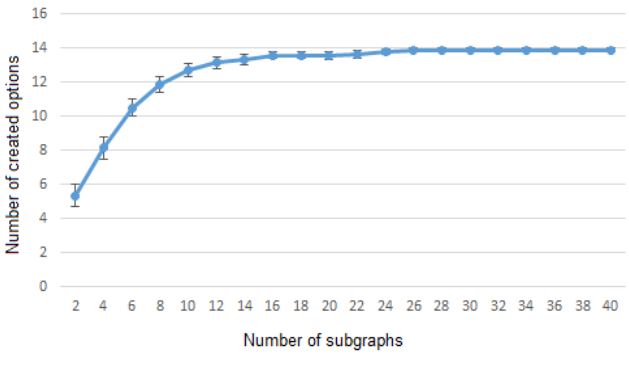
شکل ۷: محیط تاکسی [۹].



شکل ۸: مقایسه یادگیری Q و یادگیری Q به همراه بازنمایی تجربه و استفاده از روش مرکزیت یال و روش MBLP برای کسب مهارت در محیط دواتاقه.



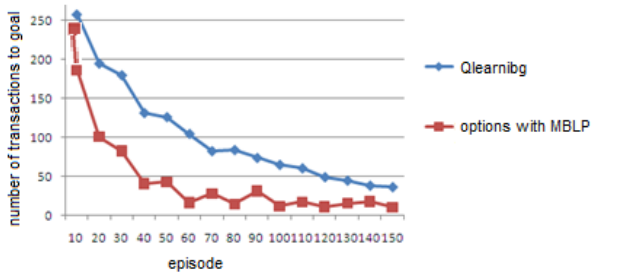
(الف)



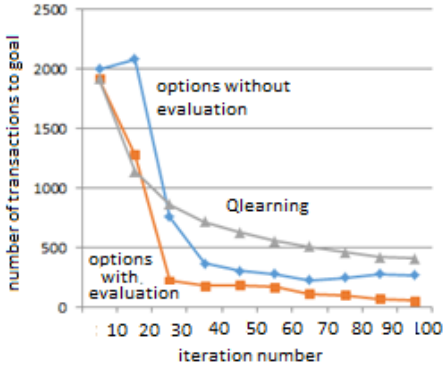
(ب)

شکل ۹: (الف) نتایج الگوریتم LoBet [۱۸] و (ب) نتایج الگوریتم MBLP در محیط شش‌اتاقه. نتایج بر روی ۲۰ تکرار میانگین‌گیری شده و خطای استاندارد بر روی میانگین نشان داده می‌شود.

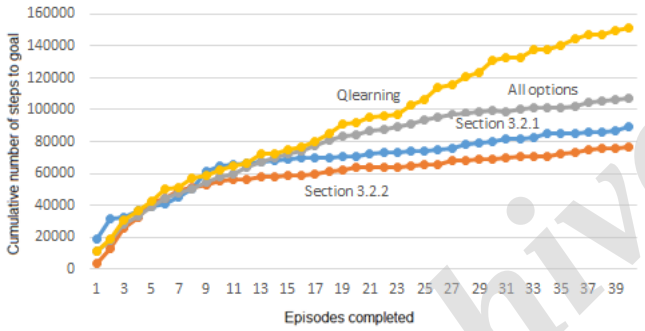
یک میله به یک میله دیگر انتقال دهد. در هر مرحله نباید دیسک بزرگ‌تر بر روی دیسک کوچک‌تر قرار گیرد. برای حل این مسأله در یادگیری تقویتی می‌توان به عامل در صورت انتقال کامل دیسک‌ها پاداش داد و برای جلوگیری از انجام بیهوده کارها برای انجام هر حرکت یک واحد جریمه برای آن مقرر کرد.



شکل ۱۰: مقایسه یادگیری Q و گزینه‌هایی که با روش MBLP در محیط تاکسی ساخته شده‌اند.



شکل ۱۱: مقایسه یادگیری Q و استفاده از همه گزینه‌های ایجادشده و استفاده از گزینه‌های هرس‌شده در محیط هانوی.



شکل ۱۲: مقایسه کارایی یادگیری در چهار عامل در محیط بازی. اولین عامل از روش یادگیری Q استفاده می‌کند، دومی همه گزینه‌ها را به کار می‌گیرد و بقیه عامل‌ها از یکی از روش‌های ارزیابی مهارت استفاده می‌کنند.

محیط تاکسی یکی دیگر از محیط‌های ارزیابی در حوزه یادگیری تقویتی سلسله‌مراتبی است. این محیط که در شکل ۷ نمایش داده شده است، یک محیط مشبک ۵×۵ است که یک مسافر در این محیط در یکی از چهار مکان R، G، Y و B قرار دارد. یک تاکسی هم وظیفه دارد که به مکان مسافر برود و سپس مسافر را سوار نموده و به مقصد برساند. مقصد مسافر هم یکی از چهار مکان R، G، Y و B خواهد بود. عامل محل قرارگیری مسافر را به صورت دقیق نمی‌داند و باید در محیط به جستجو بپردازد تا مسافر را پیدا کرده و سوار نماید. عامل در صورت رسیدن به محل قرارگیری مسافر، کنش "سوارکردن مسافر" و در صورت رسیدن به مقصد، کنش "پیاده‌کردن مسافر" را انجام می‌دهد. برای حرکت در محیط هم از چهار کنش "حرکت به بالا"، "حرکت به پایین"، "حرکت به چپ" و "حرکت به راست" استفاده می‌کند [۹].

۴-۲ نتایج آزمایشات انجام‌شده

در شکل‌های ۸ تا ۱۲ نتایج ساخت گزینه‌ها بر اساس زیرهدف‌های کشف‌شده با الگوریتم مرکزیت یال و با الگوریتم تشخیص خوشه

پیشنهادی نشان داده شده است. آزمایش‌ها در پنج محیط مشبک دواتاقه، محیط مشبک شش‌اتاقه، محیط تاکسی، برج هانوی و محیط بازی انجام گرفت. نخست عامل در k دوره به اکتشاف در محیط می‌پردازد. در آغاز هر دوره، عامل در یک حالت شروع قرار می‌گیرد و با انتخاب کنش‌ها بر اساس مقادیر Q و با روش ε–حریصانه[1] با $\varepsilon = ۰/۱$ به حالت بعدی می‌رود تا جایی که به هدف برسد.

در این شبیه‌سازی‌ها عامل از یادگیری Q به همراه یادگیری درون گزینه‌ای با نرخ یادگیری به صورت ثابت ۰/۰۵ و نرخ تخفیف $\gamma = ۰/۹$ استفاده می‌کند. مقادیر اولیه Q صفر است. در محیط مشبک عامل در ازای رسیدن به هدف پاداش ۱ و در ازای هر حرکت تنبیه ۰/۰۰۰۰۰۰۱− دریافت می‌نماید. هدف مربعی در گوشه جنوب غربی است. چهار کنش پایه شمال، جنوب، شرق و غرب، عامل را در جهت مشخص‌شده هدایت می‌کند. هر کنش عامل با احتمال ۰/۹ به درستی انجام می‌شود و با احتمال ۰/۱ عامل به یکی دیگر از جهت‌ها حرکت داده می‌شود. اگر جهت حرکت مسدود باشد عامل در همان قبلی می‌ماند.

پس از k دوره تراکنش عامل با محیط، الگوریتم خوشه‌بندی گراف اجرا می‌شود و زیرهدف‌ها شناسایی می‌گردند. بعد از این که خوشه‌ها انتخاب شدند همه حالت‌های مرزی متعلق به خوشه‌های همسایه به عنوان زیرهدف‌ها در نظر گرفته می‌شوند و گزینه‌ها از هر خوشه برای رسیدن به خوشه‌های همسایه ساخته می‌شوند. به هر زیرهدف یک پاداش مجازی مثبت داده می‌شود و یک سیاست محلی اولیه برای رسیدن به زیرهدف‌ها با روش بازنمایی تجربه یاد گرفته می‌شود. احتمال خاتمه گزینه در حالت s $(\beta(s))$ برای حالت‌های مرزی برابر یک و برای حالت‌های میانی برابر صفر است. مجموعه حالت‌های آغازین یا دامنه برای هر گزینه شامل همه حالت‌های درون خوشه می‌گردد. گزینه‌های تازه ایجاد شده به مجموعه انتخاب‌های عامل در حالی که یادگیری وظیفه کنونی ادامه دارد اضافه می‌شوند. سیاست گزینه‌ها در دوره‌های بعدی تراکنش عامل با محیط هم‌زمان با سیاست وظیفه اصلی یاد گرفته می‌شوند.

شکل ۸ نتایج استفاده از روش‌های مرکزیت یال و روش انتشار برچسب مبتنی بر ماژولاریتی را در محیط دواتاقه نشان می‌دهد. بعد از ساختن گزینه‌ها از بازنمایی تجربه برای یادگیری سیاست گزینه‌ها استفاده می‌شود. این نتایج با نتایج یادگیری Q استاندارد مقایسه شده است. همچنین برای نشان‌دادن اثر بازنمایی تجربه در سرعت یادگیری، در یک آزمایش دیگر بعد از این که در k دوره از یادگیری Q استفاده شده از بازنمایی تجربه استفاده می‌شود بدون این که گزینه‌ها ساخته شوند. همان طور که دیده می‌شود نتایج الگوریتم مرکزیت یال و الگوریتم MBLP مشابه می‌باشد. همچنین استفاده از بازنمایی تجربه اثر خوبی بر کارایی یادگیری دارد ولی استفاده از گزینه‌ها اثر بهتری در افزایش سرعت یادگیری دارد. در این آزمایش k برابر ۴ در نظر گرفته شده و روش ارزیابی ارائه‌شده در ۳–۲–۲ برای هرس‌کردن گزینه‌ها به کار گرفته شده است. در این شکل نتایج در ۱۰۰ اجرای مستقل میانگین‌گیری شده است. در بین کارهای پیشین کسب مهارت، معروف‌ترین الگوریتم، روش مرکزیت گره افزایشی است که توسط شیمشک [۱۸] ارائه شده است. در شکل ۹، مقایسه این روش با روشی که در این مقاله برای کسب مهارت ارائه کرده‌ایم دیده می‌شود. در روش مرکزیت ارائه‌شده توسط شیمشک، گزینه‌ها در ابتدای یادگیری در اختیار عامل قرار می‌گیرند و از پیش تعیین شده می‌باشند. ولی در روش ارائه‌شده در این مقاله گزینه‌ها به صورت خودکار پس از k دوره یادگیری اولیه تولید می‌شوند. بنابراین روش مرکزیت شیمشک قابل مقایسه با الگوریتم این مقاله نیست. در [۱۸] یک روش مرکزیت افزایشی نیز ارائه شده که در آن گزینه‌ها به صورت خودکار تولید می‌شوند. این الگوریتم که LoBet نامیده شده است را پیاده‌سازی کردیم و نتایج حاصل از آن را با نتایج الگوریتم ارائه‌شده توسط این پژوهش مقایسه نمودیم. برای مقایسه بهتر شرایط آزمایشات انجام‌شده در مطالعه شیمشک در آزمایش ما نیز پیاده‌سازی گردید. یکی از نقاط ضعف روش شیمشک وابستگی آن به پارامترهاست. پارامترهای این الگوریتم همانند [۱۸] مقداردهی شده‌اند.

شکل ۹ مقایسه بین الگوریتم ما و الگوریتم LoBet را در محیط شش‌اتاقه نشان می‌دهد. بیشینه محلی مرکزیت (که حالت‌های دسترسی تعریف شده‌اند) در این محیط، چهارده حالت همسایه هفت راهرو بین اتاق‌ها هستند. در این آزمایش پس از هر ۱۰۰۰ تراکنش با محیط، عامل یک زیرگراف انتقال می‌سازد. شکل ۹– الف (که در [۱۸] نشان داده شده است) تعداد حالت‌های دسترسی را که با الگوریتم LoBet در محیط شش‌اتاقه در طی زمان استخراج می‌شوند نشان می‌دهد. برای هر حالت دسترسی یک گزینه استخراج می‌گردد. شکل ۹– ب تعداد گزینه‌های استخراج‌شده با الگوریتم استخراج گزینه ارائه‌شده در این مقاله (MBLP) را نشان می‌دهد. برای این که نتایج به دست آمده توسط این دو الگوریتم با هم قابل مقایسه باشند الگوریتم خوشه‌بندی MBLP بعد از هر ۱۰۰۰ انتقال در مراحل $m \in \{۱۰۰۰, ۲۰۰۰, \ldots, ۳۰۰۰۰\}$ تکرار می‌گردد و گزینه‌ها در این گام‌ها ایجاد می‌گردند.

همان طور که در شکل ۹ دیده می‌شود، الگوریتم LoBet حالت‌های دستیابی را در محیط شش‌اتاقه بعد از تولید ۱۰۰ زیرگراف پیدا می‌کند ولی الگوریتم این مقاله، خوشه‌ها و گزینه‌ها را پس از تولید ۲۰ زیرگراف می‌تواند استخراج کند. بنابراین روش ما زیرهدف‌ها را زودتر از LoBet استخراج می‌نمایند. به نظر می‌رسد روش مرکزیت یال با روش ارائه‌شده توسط شیمشک در [۱۸] از جهت استفاده از مرکزیت مشابهت دارد اما تفاوت‌های این دو روش را می‌توان به شرح زیر بیان کرد. نخست این که در LoBet از مرکزیت میانگی گره استفاده می‌شود ولی در روش مرکزیت یال از میزان مرکزیت میانگی یال‌ها استفاده می‌شود و مرز خوشه‌ها به عنوان زیرهدف معین می‌گردند. تفاوت دوم این است که در روش مرکزیت یال همه گره‌های موجود در یک خوشه به عنوان حالت‌های آغازین یک گزینه در نظر گرفته می‌شوند و یک گزینه برای رفتن از هر خوشه به خوشه دیگر تعریف می‌شود ولی در [۱۸] تعدادی از حالت‌هایی که کمترین فاصله تا زیرهدف را دارند حالت‌های دامنه گزینه در نظر می‌گیرد. تعداد این حالت‌ها نیز از قبل مشخص نیستند و از یک پارامتر برای مشخص‌کردن تعداد آنها استفاده می‌گردد. بنابراین الگوریتم مرکزیت یال استفاده‌شده در این مقاله دامنه گزینه را با دقت بهتری پیدا می‌کند.

شکل ۱۰ نتایج استفاده از یادگیری Q و گزینه‌هایی را که با روش MBLP استخراج شده‌اند در محیط تاکسی نشان می‌دهد. در این محیط نیز گزینه‌ها تأثیر به‌سزایی در سرعت یادگیری داشته‌اند. شکل ۱۱ نتایج استفاده از یادگیری Q و گزینه‌هایی که با روش پیشنهادی در محیط هانوی ایجاد شده‌اند را نشان می‌دهد. همچنین نتیجه استفاده از همه گزینه‌ها و گزینه‌های هرس‌شده با روش ارزیابی بخش ۳–۲–۲ مقایسه می‌گردند. همان طور که دیده می‌شود گزینه‌های هرس‌شده نتیجه بهتری در کارایی یادگیری دارند. در آزمایش ۱۰ و ۱۱ مقدار k برابر ۵ در نظر گرفته شده است.

1. ε – Greedy

تحقیقات قبلی بر روی ارزیابی هر زیرهدف و گزینه بسیار ناچیز بوده است. آنها نمی‌توانند زیرهدف‌ها و گزینه‌ها را به صورت عمومی ارزیابی کنند. به عنوان مثال در [۳۴] روشی برای ارزیابی زیرهدف‌های از پیش تعیین شده ارائه گردیده است ولی در این مقاله زیرهدف‌ها و مهارت‌ها به صورت خودکار استخراج می‌گردند. در بخش ۳-۲ ما روش‌هایی برای ارزیابی گزینه‌ها در همان محیطی که استخراج می‌شوند و در همان مسأله یادگیری ارائه کردیم. این روش‌ها می‌توانند در مسایل مشابه آتی نیز استفاده شوند. در آزمایش بعدی که نتیجه آن در شکل ۱۲ نشان داده شده است روش‌های ارزیابی گزینه‌ها در محیط بازی با هم مقایسه می‌شوند.

در این آزمایش، تعداد گام‌های انباشته‌شده برای رسیدن به هدف در چهار عامل مقایسه شده است. اولین عامل تنها از یادگیری Q استفاده می‌کند، دومین عامل همه گزینه‌های ایجادشده با روش پیشنهادی را به کار می‌گیرد و دو عامل دیگر از یکی از روش‌های ارزیابی گزینه‌ها برای هرس‌کردن گزینه‌ها استفاده می‌کنند. همان طور که دیده می‌شود وقتی گزینه‌ها هرس می‌شوند کارایی یادگیری بهبود می‌یابد. روش ارزیابی ارائه‌شده در بخش ۳-۲-۲ نتیجه بهتری نسبت به روش ۳-۲-۱ در محیط بازی نشان می‌دهد.

۵- جمع‌بندی

در این مقاله روش جدیدی برای کسب مهارت‌ها و استفاده از آنها در یادگیری تقویتی ارائه شده است. در این روش پس از یک شناخت اولیه کوتاه‌مدت از محیط، با کمک یک الگوریتم پیشنهادی برای خوشه‌بندی گراف، زیرهدف‌ها استخراج می‌گردند و مهارت‌ها ایجاد می‌شوند. سپس این مهارت‌ها، ارزیابی و آنهایی که برای حل مسأله مفید هستند به انتخاب‌های عامل اضافه می‌گردند. با وجود اهمیت بررسی مناسب‌بودن یا نبودن تک‌تک مهارت‌های ایجادشده برای حل مسأله، در پژوهش‌های پیشین روش مناسبی در این زمینه ارائه نشده است. در این مقاله دو روش مکاشفه‌ای برای ارزیابی مهارت‌ها ارائه گردیده که با کمک آنها، مهارت‌های نامناسب برای حل مسأله حذف می‌گردند. استفاده از روش‌های پیشنهادی برای استخراج زیرهدف‌ها، ایجاد مهارت‌ها و ارزیابی آنها در چندین محیط آزمایشگاهی افزایش سرعت یادگیری به شکل قابل ملاحظه‌ای را نشان می‌دهد.

با وجود این که روش‌های خوشه‌بندی گراف از دقت بالایی برای کشف زیرهدف‌ها و پیداکردن دامنه مهارت‌ها برخوردار هستند اما این مسأله که خوشه‌بندی گراف گذر عامل می‌بایست بعد از چه میزان تراکنش با محیط انجام شود، مبهم است. از طرفی خوشه‌بندی زودهنگام گراف گذر ممکن است موجب خوشه‌بندی نادرست گردد و از طرف دیگر، به تعویق انداختن خوشه‌بندی به منظور شناخت بیشتر محیط باعث می‌شود که نتوانیم از مزایای مهارت‌ها در سرعت بخشیدن به حل مسأله استفاده کنیم. در کارهای آتی قصد داریم روشی افزایشی برای خوشه‌بندی گراف گذر عامل ارائه نماییم که امکان خوشه‌بندی و ایجاد مهارت‌ها در چند مرحله را ایجاد نماید.

مراجع

[1] A. G. Barto and S. Mahadevan, "Recent advances in hierarchical reinforcement learning," *Discrete Event Dynamic Systems*, vol. 13, no. 4, pp. 341-379, Jan. 2003.

[2] R. S. Sutton and A. G. Barto, "Reinforcement learning: an introduction," *IEEE Trans. on Neural Networks*, vol. 9, no. 5, pp. 1054-1054, Sep. 1998.

[3] W. Moerman, Hierarchical Reinforcement Learning: Assignment of Behaviours to Subpolicies by Self-Organization, Ph.D Thesis, Cognitive Artificial Intelligence, Utrecht University, 2009.

[4] J. Pfau, *Plans as a Means for Guiding Reinforcement Learner*, Master Thesis, Department of Information Systems, University of Melbourn, 2008.

[5] T. Mann and S. Mannor, "Scaling up approximate value iteration with options: better policies with fewer iterations," in *Proc. of the 31st Int. Conf. on Machine Learning*, vol. 1, pp. 127-137, Beijing, China, 21-26 Jun. 2014.

[6] A. McGovern and R. S. Sutton, *Macro-Actions in Reinforcement Learning: An Empirical Analysis*, University of Massachusetts, Department of Computer Science, Tech. Rep, pp. 98-70, 1998.

[7] N. K. Jong, T. Hester, and P. Stone, "The utility of temporal abstraction in reinforcement learning," in *Proc. of the 7th Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, vol. 1, pp. 299-306, Estoril, Portugal, 12-16 May 2008.

[8] R. S. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, no. 1-2, pp. 181-211, Aug. 1999.

[9] T. Dietterich, "An overview of MAXQ hierarchical reinforcement learning," *Abstraction, Reformulation, and Approximation*, pp. 26-44, 2000.

[10] M. Stolle, Automated Discovery of Options in Reinforcement Learning, M.Sc Thesis, McGill University, 2004.

[11] A. McGovern and A. G. Barto, "Automatic discovery of subgoals in reinforcement learning using diverse density," in *Proc. Int. Workshop Conf. Machine Learning*, pp. 361-368, 28 Jun-1 Jul. 2001.

[12] S. Mannor, I. Menache, A. Hoze, and U. Klein, "Dynamic abstraction in reinforcement learning via clustering," in *Proc. of the 21st Int. Conf. on Machine Learning*, p. 71, Banff, Alberta, Canada, 4-8 Jul. 2004.

[13] O. Simsek and A. Barto, "Identifying useful subgoals in reinforcement learning by local graph partitioning," in *Proc. of the 22nd Int. Conf. on Machine Learning*, pp. 816-823, Bonn, Germany, 7-11 Aug. 2005.

[14] A. Jonsson and A. Barto, "A causal approach to hierarchical decomposition of factored MDPs," in *Proc. of the 22nd Int. Conf. on Machine Learning*, pp. 401-408, Bonn, Germany, 7-11 Aug. 2005.

[15] N. Mehta, S. Ray, P. Tadepalli, and T. Dietterich, "Automatic discovery and transfer of MAXQ hierarchies," in *Proc. of the 25th Int. Conf. on Machine Learning*, pp. 648-655, Helsinki, Finland, 5-9 Jul. 2008.

[16] P. Zang, P. Zhou, D. Minnen, and C. Isbell, "Discovering options from example trajectories," in *Proc. of the 26th Annual Int. Conf. on Machine Learning*, pp. 1217-1224, Montreal, Quebec, Canada, 14-18 Jun. 2009.

[17] I. Menache, S. Mannor, and N. Shimkin, "Q-cut-dynamic discovery of sub-goals in reinforcement learning," in *Proc. 13th European Conf. on Machine Learning, ECML'02*, pp. 187-195, 19-23 Aug. 2002.

[18] O. Simsek, *Behavioral Building Blocks for Autonomous Agents: Description, Identification, and Learning*, Ph.D. Thesis, Department of Computer Science, University of Massachusetts Amherst, 2008.

[19] V. Mathew, K. Peeyush, and B. Ravindran, "Abstraction in reinforcement learning in terms of metastability," in *Proc. of the 10th European Workshop on Reinforcement Learning, EWRL'12*, pp. 1-14, 2012.

[20] C. C. Chiu and V. W. Soo, "Automatic complexity reduction in reinforcement learning," *Computational Intelligence*, vol. 26, no. 1, pp. 1-25, Feb. 2010.

[21] J. H. Metzen and F. Kirchner, "Incremental learning of skill collections based on intrinsic motivation," *Frontiers in Neurorobotics*, vol. 7, p. 11, Jan. 2013.

[22] P. Bacon and D. Precup, "Using label propagation for learning temporally abstract actions in reinforcement learning," in *Proc. of the Workshop on Multiagent Interaction Networks*, pp. 357-368, 2013.

[23] M. Davoodabadi and H. Beigy, "A new method for discovering subgoals and constructing options in reinforcement learning," in *Proc. IICAI*, pp. 441-450, 2011.

[24] O. Simsek and A. G. Barto, "Skill characterization based on betweenness," in *Proc. 21th Int. Conf. on Neural Information Processing Systems, NIPS'08*, pp. p.1497-1504, Vancouver, Canada, 8-10 Dec. 2008.

[25] A. A. Rad, M. Hasler, and P. Moradi, "Automatic skill acquisition in reinforcement learning using connection graph stability centrality," in *Proc. of 2010 IEEE Int. Symp. on Circuits and Systems, ISCAS'10*, pp. 697-700, Paris, France, 30 May-2 Jun. 2010.

[26] L. J. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Machine Learning*, vol. 8, no. 3-4, pp. 293-321, May 1992.
[27] R. S. Sutton, D. Precup, and S. Singh, "Intra-option learning about temporally abstract actions," in *Proc. of the Fifteenth Int. Conf. on Machine Learning*, pp. 556-564, 1998.
[28] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, p. 026113, Feb. 2004.
[29] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, p. 036106, Sep. 2007.
[30] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, p. 066133, Jun. 2004.
[31] K. Merrick, *Modelling Motivation for Experience-Based Attention Focus in Reinforcement Learning*, Ph.D. Thesis, School of Information Technologies, the University of Sydney, 2007.
[32] M. Davoodabadi and N. Mozayani, "Automatic construction and evaluation of options in reinforcement learning," Submitted in Artificial Intelligence, https://www.journals.elsevier.com/artificial-intelligence/
[33] A. G. Barto, S. Singh, and N. Chentanez, "Intrinsically motivated learning of hierarchical collections of skills," in *Proc. of the 3rd Int. Conf. on Development and Learning, ICDL'04,* pp. 112-119, Salk Institute, San Diego, USA, Oct. 2004.
[34] J. Murata, "Controlled use of subgoals in reinforcement learning," *Robotics, Automation and Control*, pp. 167-182, 2008.

مرضیه داودآبادی فراهانی در سال ۱۳۸۲ مدرک کارشناسی مهندسی کامپیوتر خود را در گرایش نرم‌افزار از دانشگاه صنعتی امیرکبیر و در سال ۱۳۸۴ مدرک کارشناسی ارشد مهندسی کامپیوتر خود را در گرایش هوش مصنوعی از دانشگاه صنعتی اصفهان دریافت نمود. از سال ۱۳۸۵ به عضویت هیأت علمی دانشگاه قم درآمد و هم‌اکنون دانشجوی دکتری مهندسی کامپیوتر گرایش هوش مصنوعی در دانشگاه علم و صنعت ایران می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: هوش مصنوعی، یادگیری ماشین، داده‌کاوی و تحلیل داده‌ها، امنیت اطلاعات.

ناصر مزینی در سال ۱۳۶۹ مدرک کارشناسی خود را از دانشگاه صنعتی شریف در مهندسی برق گرایش کامپیوتر سخت افزار اخذ نمود و سپس در سال ۱۳۷۲ مدرک کارشناسی ارشد را در رشته سیستم‌های اطلاعاتی و تله‌ماتیک از سوپلک فرانسه و همچنین در سال ۱۳۷۷ از دانشگاه رن یک فرانسه مدرک دکتری را در رشته انفورماتیک دریافت نمود. وی از سال ۱۳۷۹ در دانشکده مهندسی کامپیوتر دانشگاه علم و صنعت ایران مشغول به فعالیت گردید و اینک نیز عضو هیأت علمی این دانشکده می‌باشد. زمینه‌های علمی مورد علاقه نام‌برده عمدتا در زمینه رایانش نرم، فناوری اطلاعات و شبکه های کامپیوتری است.