یازدهمین کنفرانس سیستم‌های هوشمند ایران
۹ و ۱۰ اسفند ۱۳۹۱
(ICIS2013)
11th Iranian Conference on Intelligent Systems
February 27th & 28th, 2013

روشی یادگیر برای ترکیب وظایف در یادگیری تقویتی پیمانه‌ای

سیدمحمدحسین میرهاشمی[۱]، ناصر مزینی[۲] و محمدرضا جاهد مطلق[۳]
[۱]دانشگاه علم و صنعت ایران، mirhashemi@comp.iust.ac.ir
[۲]دانشگاه علم و صنعت ایران، mozayani@iust.ac.ir
[۳]دانشگاه علم و صنعت ایران، jahedmr@iust.ac.ir

چکیده – دسته‌ای از روش‌های یادگیری تقویتی سعی می‌کنند مسائل پیچیده را با تجزیه به مسائل کوچکتر حل کنند. به‌این‌صورت که هدف اصلی را به تعدادی زیرهدف یا وظیفه می‌شکنند و هریک را توسط یک یادگیر فرا می‌گیرند، سپس به ترکیب این وظایف یادگرفته شده می‌پردازند. بیشتر این روش‌ها به علت عدم استفاده مناسب از دانش موجود در این پیمانه‌های فراگرفته شده، در ترکیب آن‌ها با مشکل روبرو می‌شوند، و در نتیجه نمی‌توانند به خوبی به هدف اصلی دست بیابند. در این مقاله روشی یادگیر برای ترکیب وظایف ارائه شده است که به علت راهکار مناسبی که برای ترکیب پیمانه‌ها ارائه می‌کند، به جواب مناسبی برای مسئله‌ی اصلی می‌رسد. پیاده‌سازی این روش در یک مسئله که از منظر یادگیری تقویتی بسیار بزرگ و پیچیده است، عملکرد بسیار مناسبی از خود نشان داده و به کارایی بسیار بالایی می‌رسد.

کلید واژه- ترکیب پیمانه‌ها، یادگیری تقویتی پیمانه‌ای، یادگیری تقویتی.

۱- مقدمه

یادگیری تقویتی یکی از شاخه‌های یادگیری ماشین است که به علت مزایای متعدد، بسیار مورد توجه قرار گرفته است. با این وجود یادگیری تقویتی از معایبی نیز رنج می‌برد که شاید مهمترین آن را بتوان مشکل نفرین ابعاد دانست. روش‌های یادگیری تقویتی به ابعاد محیط بسیار حساس هستند، و از آن‌جایی که محیط بیشتر مسائل واقعی بسیار پیچیده و بزرگ است، به‌کارگیری یادگیری تقویتی در آن‌ها با چالش جدی روبرو می‌شود.

دسته‌ای از روش‌ها برای غلبه بر این مشکل مسئله را به زیرمسائلی آسان‌تر تجزیه می‌کنند، سپس این زیرمسائل را که ساده‌تر و مناسب یادگیری هستند با یادگیرهای تقویتی مرسوم حل کرده، و آنگاه سعی می‌کنند تا با ترکیب مناسب این پیمانه‌های یادگرفته شده به جواب قابل قبولی برای مسئله‌ی اصلی برسند. این سازوکار پیمانه‌ای باعث انتساب نام یادگیری تقویتی پیمانه‌ای[۱] به این روش‌ها شده است.

مشکل عمده‌ی روش‌های پیمانه‌ای در ترکیب پیمانه‌های یادگرفته شده است. روش‌هایی که تا کنون ارائه شده‌اند یا در مسائلی خاص بوده تا بتوانند با استفاده از ویژگی‌های مسئله به حل این مشکل بپردازند، و یا سعی کرده‌اند تا با روش‌هایی ریاضی ترکیب پیمانه‌ها را عملی کنند. عده‌ی قلیلی هم که استفاده از یادگیری را برای ترکیب پیمانه‌ها در پیش گرفته‌اند، به خوبی از دانش موجود در پیمانه‌ها استفاده نکرده‌اند، و به همین

[1] Modular Reinforcement Learning

دلیل موفق به غلبه بر مشکل بزرگی فضای حالت نشده‌اند.

در این مقاله روشی یادگیر برای ترکیب پیمانه‌های یادگرفته شده ارائه می‌شود که از یادگیری انجام شده توسط پیمانه‌ها حداکثر استفاده را می‌برد. به‌علاوه سعی می‌کند وابسته به خواص محیط نباشد تا جامعیت خود را برای حل مسائل مشابه از دست ندهد. روش پیشنهادی در یک محیط که برای یادگیرهای تقویتی ساده بسیار بزرگ و پیچیده است، پیاده‌سازی شده و نتایج بسیار ارزنده‌ای کسب کرده است.

۲- فرمول‌بندی مسئله

مسئله به صورت یک فرآیند تصمیم مارکوف[۲] (MDP) درنظر گرفته شده است. یک MDP توسط چهارتایی (S, A, P, R) تعریف می‌شود که در آن S فضای حالت و A فضای اعمال است. $P(s'|s,a)$ تابع توزیع احتمال رسیدن به حالت 's با انجام عمل a در حالت s است و تابع انتقال نامیده می‌شود. $R(s,a)$ امید ریاضی پاداش دریافتی با انجام عمل a در حالت s است و تابع پاداش نامیده می‌شود. یک سیاست[۳] $\pi(s)$ تابعی است که برای هر حالت یک عمل را پیشنهاد می‌کند. هدف نهایی یافتن سیاست بهینه (π^*) است که با انجام اعمال پیشنهادی آن، عامل به حداکثر مجموع پاداش در طی فعالیتش برسد.

برای عامل n زیرهدف تعریف می‌شود، برای هر زیرهدف یک سیگنال پاداش جداگانه درنظر گرفته می‌شود، بنابراین n سیگنال پاداش $R_1, R_2, \ldots, R_n$ داده شده است. اگر تنها اجزائی از محیط که به یک زیرهدف مربوط است را در نظر بگیریم، زیرفضایی از محیط به‌وجود می‌آید که مخصوص آن زیرهدف است، به‌عبارت دیگر پاداش مربوط به آن زیرهدف تنها به حالت عامل در آن زیرفضا مربوط است. بنابراین n زیرفضای $S_1, S_2, \ldots, S_n$ که هرکدام مربوط به یکی از اهداف است نیز داده شده است.

با کمک آنچه بیان شد برای هر زیرهدف یک MDP جداگانه درنظر گرفته می‌شود که فضای حالت آن، زیرفضای مربوط به آن زیرهدف، و پاداش آن همان پاداش مربوط به آن زیرهدف است: (S_i, A, P_i, R_i). فضای اعمال همه‌ی این MDPها یکسان و همان فضای اعمال MDP کلی است. تابع انتقال هرکدام (P_i) نیز به راحتی از روی تابع انتقال متعلق به MDP کلی قابل استخراج است، چراکه فضای حالت هرکدام، زیر فضایی از فضای حالت MDP کلی، و فضای اعمالش، یکسان با فضای اعمال MDP کلی است. هرکدام از این MDPها که مربوط به یک زیرهدف عامل است را یک وظیفه، و یادگیری که آن را یاد می‌گیرد پیمانه می‌نامیم، بنابراین عامل n وظیفه (پیمانه) $M_1, M_2, \ldots, M_n$ دارد.

۳- مروری بر کارهای گذشته

اولین کار در یادگیری تقویتی پیمانه‌ای را می‌توان [۱] دانست. در این کار ابتدا پیمانه‌ها که از نوع یادگیر-Q [۲] هستند یادگرفته می‌شوند؛ سپس یک یادگیر دیگر، که آن هم از نوع یادگیر-Q است، به‌کار گرفته می‌شود که می‌بایست در هر حالت یکی از پیمانه‌ها را انتخاب کند. انتخاب یک پیمانه به معنی انتخاب عمل پیشنهادی آن پیمانه و اجرا شدن آن توسط عامل است. به بیان دیگر حالت این یادگیر همان حالت مسئله، ولی عمل آن انتخاب یک پیمانه است. این روش یادگیری-Q سلسله‌مراتبی[۴] نامیده شده است.

مشکل آشکار روش فوق همان مشکل همیشگی بزرگی فضای حالت است. هرچند اگر تعداد وظایف کمتر از تعداد اعمال باشد، این روش تا حدی فضای حالت-عمل را کاهش می‌دهد، اما معمولا خود فضای حالت به تنهایی چنان بزرگ است که کاهش اعمال تاثیر عملی در نتیجه کار نخواهد داشت.

راه‌کار دیگری که برای غلبه بر مشکل بزرگی فضای حالت در ادبیات موضوع مطرح شده است، استفاده از سازوکاری ساده و غیر یادگیر برای ترکیب اعمال پیشنهادی پیمانه‌ها است. مفهوم

[2] Markov Decision Process
[3] Policy
[4] Hierarchical Q-Learning

نهفته در این شیوه آنست که پیمانه‌ها عمده کار یادگیری را به انجام رسانده‌اند، حال کافی است که به نحوی ساده ترکیب شوند تا جواب کلی دلخواه حاصل شود.

کارلسون [۳] و هامفریس [۴] روش‌هایی از این نوع را ارائه می‌دهند. در هر دوی آن‌ها پیمانه‌ها از نوع یادگیر-Q هستند، و هر دو برای انتخاب عمل از ارزش اعمال استفاده می‌کنند. هامفریس بیان می‌کند که انتخاب عمل پیمانه‌ای که ارزش عمل پیشنهادی‌اش در جدول-Q خودش، از ارزش اعمال پیشنهادی دیگر پیمانه‌ها در جدول-Q خودشان بیشتر است، کار درستی نیست؛ چراکه ممکن است با این کار پیمانه‌ای دیگر کاهش ارزش بزرگتری را متحمل شود، و در نتیجه بیشینه پاداش مجموع حاصل نشود. بنابراین روشی را پیشنهاد می‌کند که در آن عمل پیشنهادی پیمانه‌ای انتخاب می‌شود که در صورت انتخاب نشدن عملش، بیشترین کاهش ارزش را متحمل خواهد شد.

کارلسون از روشی به نام بزرگترین جرم[5] برای انتخاب عمل استفاده می‌کند. در این روش در هر حالت عملی انتخاب می‌شود که مجموع ارزش‌های انتسابی به آن در جدول-Q پیمانه‌ها، بیشینه باشد. روش بزرگترین جرم اولین بار در [۵] پیشنهاد شده است.

در [۶] به روش‌هایی که عمل را بر اساس ارزششان در پیمانه‌ها انتخاب می‌کنند و پیمانه‌ها از نوع یادگیر-Q هستند اشکال گرفته شده است. یادگیر-Q یک یادگیر خارج-سیاست[6] است و بنابراین ارزش اعمال با این فرض بدست می‌آید که سیاست بهینه دنبال خواهد شد. درحالیکه در روش پیمانه‌ای هیچ‌کدام از پیمانه‌ها کنترل عامل را به تنهایی در دست ندارد تا همیشه اعمال پیشنهادی‌اش به اجرا درآید. بنابراین ارزش اعمال به‌دست آمده از طریق یادگیری-Q، برای مقایسه و انتخاب نامناسب هستند و تصمیم‌گیری براساس آن‌ها باعث کاهش کارایی خواهد شد.

برهمین اساس پیمانه‌ها را یادگیر سارسا[7] که یک یادگیر داخل-سیاست است، تعیین کرده است. یک یادگیر داخل-سیاست[8] سیاستی که در طول یادگیری به‌اجرا درمی‌آید را فرا می‌گیرد، بنابراین ارزش اعمال آن اشکال فوق را نخواهد داشت و انتظار می‌رود سامانه‌ی کلی نتیجه بهتری بدست آورد. با این روش کارایی عامل تا حدی افزایش می‌یابد، اما استفاده از ارزش اعمال که مشکل اصلی این روش‌هاست در این روش نیز وجود دارد. به‌علاوه همانطور که خودش بیان کرده همگرایی این روش اثبات نشده است.

سه روش بالا ([۳]، [۴] و [۶]) و روش‌های مانند آن‌ها که بر اساس ارزش اعمال به ترکیب پیمانه‌ها می‌پردازند، برای مسائل چند-هدفه ارائه شده‌اند. در مسائل چند-هدفه هر زیرهدف دارای اصالت است و هدف کلی عامل نیز چیزی جز ارضای حداکثری همه‌ی زیراهداف نیست. به همین دلیل ارزش اعمال از دید زیراهداف مختلف قابل مقایسه بوده و عامل می‌تواند بر اساس این مقایسه تصمیم‌گیری کرده و یک عمل را انتخاب کند. در مسائل چند-هدفه مجموع پاداش زیراهداف به عنوان پاداش به عامل تعلق می‌گیرد: $R = R_1 + R_2 + \ldots + R_n$.

در دیگر مسائل که زیر اهداف دارای اصالت نیستند و تنها برای رسیدن به هدفی کلی‌تر طراحی شده‌اند، استفاده از این روش‌ها ناکارآمد خواهد بود، چراکه ارضای خود زیراهداف مدنظر نیست و بنابراین ارزش اعمال از منظر آن‌ها معیار مناسبی برای یافتن عمل منتخب نیست، بلکه تنها هدف کلی دارای اصالت بوده و زیراهداف تنها وسیله‌ای برای رسیدن به آن هستند. روش یادگیری-Q سلسله‌مراتبی [۱] و روشی که در این مقاله ارائه شده است برای این نوع مسائل نیز قابل به‌کارگیری هستند.

۴- روش پیشنهادی

برای حل مسئله‌ی کلی با استفاده از پیمانه‌هایی که هریک

[5] Greatest Mass
[6] Off-Policy
[7] Sarsa
[8] On-Policy

وظیفه‌ای را فرا گرفته‌اند، می‌بایست به نحوی نتیجه‌ی یادگیری آن‌ها با هم ترکیب شود تا جواب کلی حاصل شود. سازوکاری که از یادگیری برای این ترکیب استفاده کند مطلوب به‌نظر می‌رسد، اما این سازوکار چگونه می‌تواند از پیمانه‌ها سود ببرد، و از آن مهمتر چگونه می‌تواند بر مشکل همیشگی بزرگی فضای حالت غلبه کند؟

روش‌هایی که در مسائل چند-هدفه ارائه شده‌اند و از ارزش اعمال برای ترکیب پیمانه‌ها استفاده می‌کنند، درواقع مشکل را با استفاده از ویژگی‌های خاص مسائل مورد نظرشان دور زده‌اند. روش یادگیری-Q سلسله‌مراتبی [۱] نیز درواقع استفاده‌ی چندانی از نتیجه‌ی یادگیری پیمانه‌ها نمی‌کند و به همین دلیل پیشرفتی در مورد مشکل بزرگی فضای حالت بدست نمی‌آورد.

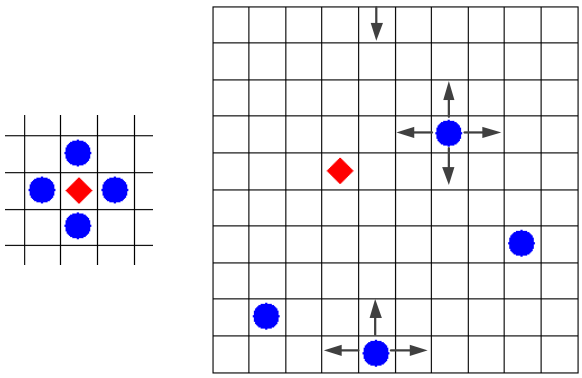
برای استفاده از پیمانه‌های یادگیری شده، می‌بایست به نحوی از دانش و تجربه‌ی ذخیره شده در آن‌ها استفاده کرد. اگر از مقادیر ارزش اعمال به عنوان ابزاری برای مقایسه استفاده شود، می‌بایست زیراهداف دارای اصالت و مقایسه پذیر باشند، که همانطور که پیش از این بیان شد دراین صورت روش به مسائل چند-هدفه محدود شده است. بنابراین هم می‌بایست از ارزش اعمال استفاده کرد و هم نباید آن‌ها را به عنوان ابزاری برای مقایسه به‌کار برد.

در روش پیشنهادی این مقاله هر پیمانه بر اساس ارزشی که به اعمال مختلف در هر حالت داده است، فضای حالت را به فضای حالتی جدید نگاشت می‌کند. این فضای حالت جدید در واقع فضای حالت از دیدگاه زیرهدف آن پیمانه است. این فضاهای حالت به‌وجود آمده از دیدگاه زیراهداف مختلف، در کنار یکدیگر قرار داده شده و یک فضای حالت کلی جدید به‌وجود می‌آید. این فضای جدید در اختیار یادگیر دیگری قرار می‌گیرد تا در آن به یادگیری بپردازد و عمل خروجی عامل را تولید کند. ازآنجاییکه این فضای جدید از ترکیب نتایج یادگیری زیراهداف بدست آمده است، انتظار می‌رود که فضایی کوچکتر، چگال‌تر از منظر دانش حل مسئله، و آسان‌تر برای یادگیری باشد. یادگیر مذکور را یادگیر ترکیب‌کننده می‌نامیم.

اگر پیمانه‌ها یادگیر-Q فرض شوند و هر یادگیر-Q وظیفه مربوط به زیرهدف خود را فرا گرفته باشد، آنگاه ارزش اعمال مختلف یک حالت در جدول-Q یک پیمانه، نشان دهنده‌ی وضعیت آن حالت از منظر زیرهدف آن پیمانه خواهد بود. اگر این ارزش اعمال در یک حالت، از همه‌ی پیمانه‌ها دریافت شده و در کنار هم قرار گیرند، آن‌گاه وضعیت حالت از منظر همه‌ی زیراهداف مشخص خواهد شد. با این روش هر حالت از محیط به حالت دیگری نگاشت می‌شود، و فضای حالت جدیدی به‌وجود می‌آید که به‌نوعی می‌توان گفت که در آن هر حالت از نظر زیراهداف ارزش‌گذاری شده است. یادگیر ترکیب‌کننده در این فضای جدید به یادگیری می‌پردازد.

۵- پیاده‌سازی

برای پیاده‌سازی روش پیشنهادی، مسئله‌ی تعقیب[۹] انتخاب شده است. در این مسئله که شکار و شکارچی نیز نامیده شده است، هدف چهار شکارچی گرفتن یک شکار در یک محیط گسسته دوبعدی است. شکارچی‌ها تنها زمانی می‌توانند شکار را بگیرند که او را از چهار طرف محاصره کنند. شکل ۱.



(ب) (الف)

شکل ۱: (الف) محیط مسئله تعقیب (ب) وضعیت گرفتن شکار

هر موجود (شامل شکارچی‌ها و شکار) در هر گام می‌تواند به یکی از چهار خانه‌ی همسایه‌ی خود حرکت کرده و یا در جای

[9] Pursuit

خود ثابت بماند. محیط چنبره‌ای[10] است، یعنی حرکت به سمت بیرون در یک طرف محیط، باعث ورود از طرف دیگر می‌شود. بنابراین محیط دیوار و گوشه ندارد که باعث شود کمتر از چهار شکارچی بتوانند شکار را محاصره کنند. موجودات محیط در هر گام به‌طور همزمان یک عمل (شامل حرکت به چهار جهت و ثابت ایستادن) انجام می‌دهند. هیچ دو موجودی نمی‌توانند در یک خانه قرار بگیرند، و تصادم بین آن‌ها از طریق اولویت در حرکت حل می‌شود که این اولویت در هر گام بصورت تصادفی به موجودات محیط تعلق می‌گیرد.

شکار در هر گام یکی از پنج عمل را به صورت تصادفی انتخاب می‌کند. عامل‌های یادگیر شکارچی‌ها هستند که می‌بایست گرفتن شکار را فرا بگیرند. این مسئله یک مسئله‌ی چند-عامله[11] است، اما چون عامل‌های آن به دیگر عامل‌ها تنها به عنوان یک جزء از محیط نگاه می‌کنند و به مدل کردن یکدیگر نمی‌پردازند، مسئله از دید هر عامل تبدیل به مسئله‌ای تک-عامله با محیطی بسیار پویا می‌شود. بنابراین بر طبق دسته‌بندی [۷]، این مسئله یک مسئله‌ی چند-عامله متجانس فاقد ارتباطات[12] است، به‌علاوه در آن عامل‌ها یکدیگر را مدل نمی‌کنند.

اگر ابعاد محیط ۱۰×۱۰ باشد، فضای حالت برای عامل $۱۰^{۸}$ حالت خواهد داشت. این فضای حالت بسیار بزرگ‌تر از آن است که یک یادگیر تقویتی ساده بتواند در زمان معقول آن را فرا بگیرد.

هریک از زیراهداف عامل در رابطه با یکی از موجودات محیط تعیین شده است، بنابراین عامل چهار زیرهدف دارد که سه تای آن مربوط به سه شکارچی دیگر، و یکی مربوط به شکار است. پاداش زیرهدف مربوط به یک شکارچی، هنگامیکه عامل باعث تصادم با آن می‌شود مقداری منفی، و در گام‌های دیگر صفر تعیین شده است. پاداش زیرهدف مربوط به شکار نیز هنگامیکه عامل به شکار می‌رسد یک مقدار مثبت، و در گام‌های دیگر صفر تعیین شده است. یادگیر ترکیب‌کننده هر عامل نیز هنگامی که شکارچی‌ها شکار را بگیرند، یک پاداش مثبت بزرگ، و در هر گام دیگر یک پاداش منفی کوچک دریافت می‌کند.

پیمانه‌ها و همچنین یادگیر ترکیب‌کننده، یادگیر-Q تعیین شده‌اند. در هر حالت از محیط، ارزش چهار عمل حرکت به چهار جهت در جدول-Q چهار پیمانه، تشکیل یک حالت جدید با ۱۶ متغیر را می‌دهد. یادگیر ترکیب‌کننده در این فضای حالت ۱۶ متغیره به یادگیری می‌پردازد و عملی که عامل در هر حالت انجام می‌دهد را تعیین می‌کند، در واقع این یادگیر ترکیب‌کننده است که کنترل عامل را در دست دارد. شکل ۲.

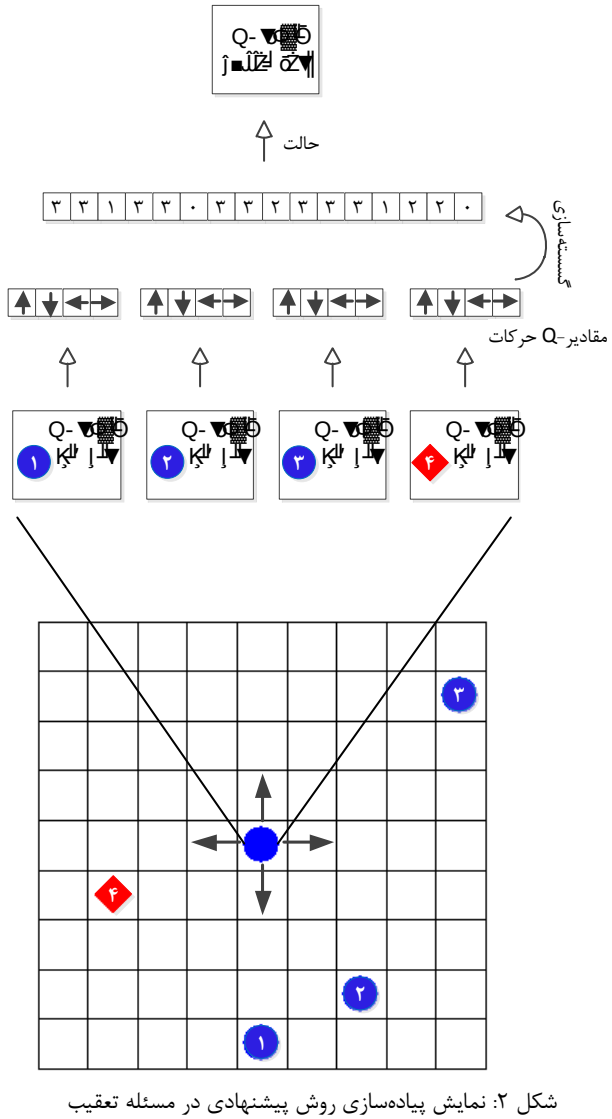
ازآنجاییکه ارزش اعمال در جدول-Q مقادیری پیوسته هستند، این ۱۶ متغیر حالت، مقادیری پیوسته خواهند داشت. اما یادگیر ترکیب کننده که یک یادگیر-Q است، می‌بایست در فضای گسسته به یادگیری بپردازد. بنابراین یک گسسته سازی ساده‌ی ۴-مقداره بر روی ارزش هر عمل انجام می‌شود، تا فضای حالت یادگیر ترکیب‌کننده گسسته شود.

با گسسته‌سازی ۴-مقداره هر متغیر می‌تواند ۴ مقدار متفاوت داشته باشد، بنابراین فضای حالت یادگیر ترکیب‌کننده $۴^{۱۶} \approx ۱۰^{۹}$ حالت می‌تواند داشته باشد. اما در عمل تعداد حالاتی که ترکیب‌کننده با آن‌ها برخورد می‌کند بسیار کمتر از این مقدار و حدود ۲×۱۰^{۴} حالت است. علت این امر را می‌توان در وابستگی متغیرهای حالت به یکدیگر دانست، به این معنی که برای هر مقدار از یک متغیر، یک یا تعدادی از متغیرهای دیگر تنها مقادیر خاصی را می‌توانند اتخاذ کنند. بنابراین همه‌ی ترکیب‌های مقادیر متغیرهای حالت قابل دستیابی نیستند و بنابراین فضای حالت بسیار کوچکتر از تعداد کل حالات قابل ساخت با متغیرهای حالت است.

یادگیری در دوره‌های[13] متوالی انجام می‌شود. هر دوره با محاصره شدن شکار توسط شکارچی‌ها، و یا با رسیدن تعداد گام

[10] Toroidal
[11] Multi-Agent
[12] Homogeneous Non-Communicating
[13] Episode

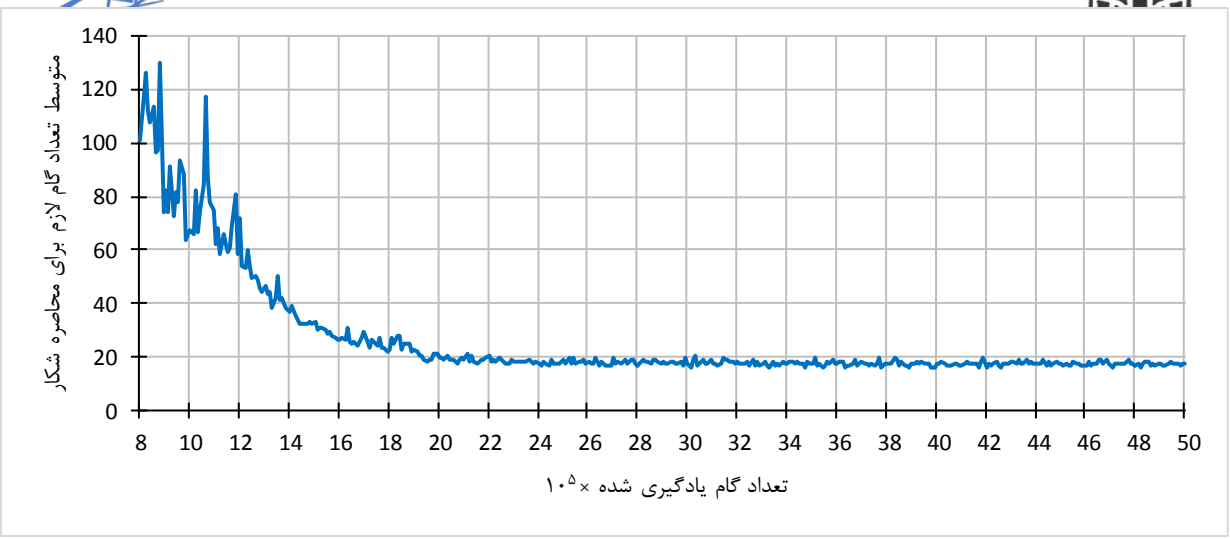
دوره به ۱۰۰ پایان می‌یابد. در ابتدای هر دوره موجودات به صورت تصادفی در خانه‌های محیط پخش می‌شوند. نرخ یادگیری ثابت و به مقدار ۰/۱، و ضریب کاهش نیز ثابت و به مقدار ۰/۹ تنظیم شده است. همه‌ی یادگیرها از یک سیاست e-greedy استفاده می‌کنند که مقدار پارامتر e در ۵۰ گام اول هر دوره از ۰/۴ به ۰ به صورت خطی تنزل پیدا می‌کند.



شکل ۲: نمایش پیاده‌سازی روش پیشنهادی در مسئله تعقیب

شکل ۳ نمودار کارایی روش پیشنهادی را نشان می‌دهد. با گذشت هر ۱۰ هزار گام از یادگیری، یادگیری بطور موقت متوقف شده و عامل‌ها هزار گام مورد تست قرار گرفته‌اند تا متوسط تعداد گام‌های لازم برای محاصره کردن شکار بدست آید. برای رسم نمودار، ۱۰ دفعه یادگیری و تست با روش ذکر شده به انجام رسیده و بین آن‌ها میانگین‌گیری انجام شده است.

پس از گذشت ۲ میلیون گام از یادگیری، عامل‌ها تقریبا به حداکثر کارایی خود نزدیک شده‌اند. برای مسئله‌ای با فضای حالت به اندازه 10^8، این تعداد گام برای یادگیری بسیار مناسب است. به‌علاوه پس از یادگیری، شکارچی‌ها در طی کمتر از ۲۰ گام شکار را محاصره می‌کنند، که در چنین محیطی، تقریبا نزدیک به بیشینه‌ی کارایی هوشمندترین عامل‌ها است.



شکل ۳: نمودار کارایی عامل‌های شکارچی - متوسط تعداد گام‌های لازم برای محاصره‌ی شکار برحسب تعداد گام‌های یادگیری عامل‌های شکارچی

۶- جمع‌بندی و نتیجه‌گیری

در این مقاله روشی برای حل مشکل ترکیب وظایف، در روش‌های یادگیری تقویتی پیمانه‌ای ارائه شد. مزیت روش ارائه شده نه تنها در استفاده از یادگیری برای ترکیب پیمانه‌ها، بلکه در استفاده‌ی مناسب از تجربه‌ی یادگیری انجام شده در پیمانه‌ها برای ترکیب آن‌ها نیز هست؛ و به همین علت به نتایج قابل توجهی دست پیدا می‌کند.

مزیت دیگر روش پیشنهادی را می‌توان در عدم وابستگی آن به اندازه‌ی محیط مسئله دانست، چراکه پیمانه‌ها تنها ارزش اعمال را به یادگیر ترکیب‌کننده عرضه می‌کنند، و با افزایش اندازه محیط، تغییری در اندازه فضای حالت یادگیر ترکیب‌کننده ایجاد نخواهد شد. خود پیمانه‌ها نیز چون تنها یک یا تعدادی کمی از اجزاء محیط را درنظر دارند، حساسیت زیادی به اندازه‌ی محیط ندارند.

روش ارائه شده نیز مانند هر روشی علاوه بر فواید، با مسائلی نیز روبروست که می‌بایست برای آن‌ها چاره‌ای اندیشیده شود. یکی از این مشکلات افزایش نمایی اندازه فضای حالت یادگیر ترکیب‌کننده، با افزایش تعداد وظایف است. بنابراین در مسائلی که تعداد وظایف زیاد است، روش کارایی خود را از دست می‌دهد. برای رفع این مشکل می‌توان ابتدا وظایف را در دسته‌های کوچک‌تر با یکدیگر ترکیب کرد، سپس نتایج بدست آمده از دسته‌ها را نیز برای رسیدن به جواب نهایی ترکیب کرد. با این سازوکار انتظار می‌رود که این مشکل مرتفع گردد.

مسئله‌ی دیگر از بین رفتن بخشی از اطلاعاتی که توسط پیمانه‌ها یادگرفته شده، به دلیل گسسته‌سازی مقادیر ارزش اعمال است. برای مقابله با این مشکل می‌توان از یک یادگیر پیوسته برای ترکیب‌کننده استفاده کرد. بنابراین در صورت استفاده از یک یادگیر پیوسته مناسب، استفاده کامل از اطلاعات موجود در ارزش اعمال، می‌تواند باعث افزایش کارایی عامل شود.

با توجه به نتایج موفق بدست آمده، روش ارائه شده می‌تواند آغاز راهی باشد که روش‌های پیمانه‌ای را به عنوان راه‌کاری مناسب برای غلبه بر مشکل نفرین ابعاد در یادگیری تقویتی مطرح کند.

مراجع

[1] L. J. Lin, "Scaling up Reinforcement Learning for robot control," *Proc. 10th. Int. Conf. on Machine Learning*. pp. 182-196, 1993.

[2] C. J. C. H. Watkins, *Learning from Delayed Rewards*. PhD Thesis, University of Cambridge, Psychology Department, 1989.

[3] J. Karlsson, *Learning to Solve Multiple Goals*. PhD Thesis, University of Rochester, 1997.

[4] M. Humphrys, "Action Selection Methods using Reinforcement Learning," *From Animals to Animats 4: Proc. 4th. Int. Conf. on Simulation of Adaptive Behavior*. Cambridge, MA, pp. 134-144, 1996.

[5] S. Whitehead, J. Karlsson and J. Tenenberg, *Learning Multiple Goal Behavior via Task Decompostion and Dynamic Policy Merging*. Kiluwer Academic Press, pp. 45-78, 1993.

[6] N. Sprague and D. H. Ballard, "Multipe-Goal Reinforcement Leraning with Modular Sarsa(0)," *Proc. Int. Joint Conf. on Artificial Intelligence*. pp. 1445-1447, 2003.

[7] P. Stone, M. Veloso, "Multiagent systems: A survey from a machine learning perspective" *Autonomous Robots*, Vol. 8, pp. 348-383, 2000.