مجله کنترل
ISSN (print) 2008-8345
ISSN (online) 2538-3752
جلد۱۴، شماره ۴، زمستان ۱۳۹۹، صفحه ۶۶-۵۵

یادگیری تقویتی چندعاملی مشارکتی در محیط‌های پویا بر اساس انتقال دانش برای مسأله گله‌داری

امین نیک‌انجام[۱]، منیره عبدوس[۲]، ماهنوش مهدوی مقدم[۳]

[۱] استادیار، دانشکده‌ی مهندسی کامپیوتر، گروه هوش مصنوعی، دانشگاه صنعتی خواجه نصیرالدین طوسی، nikanjam@kntu.ac.ir

[۲] استادیار، دانشکده‌ی مهندسی و علوم کامپیوتر، گروه هوش مصنوعی، رباتیک و رایانش شناختی، دانشگاه شهید بهشتی، m_abdoos@sbu.ac.ir

[۳] فارغ‌التحصیل کارشناسی ارشد مهندسی کامپیوتر، گروه هوش مصنوعی، دانشگاه صنعتی خواجه نصیرالدین طوسی، mahnooshmahdavi@email.kntu.ac.ir

دریافت: ۱۳۹۷/۱۰/۳۰ ویرایش : ۱۳۹۸/۰۶/۱۶ پذیرش: ۱۳۹۸/۱۰/۰۵

چکیده: امروزه، برای حل بسیاری از مسائل، از سیستم‌های چندعاملی مشارکتی استفاده می‌شود که در آن گروهی از عامل‌ها برای رسیدن به یک هدف مشترک همکاری می‌کنند. همکاری میان عامل‌ها، فوایدی همچون کاهش هزینه‌های عملیاتی، مقیاس‌پذیری بالا و سازگاری قابل‌توجه را به ارمغان خواهد آورد. برای آموزش این عامل‌ها در رسیدن به یک سیاست بهینه، از یادگیری تقویتی بهره می‌جویند. یادگیری در محیط‌های چندعاملی مشارکتی پویا، غیرقطعی و با اندازه فضای حالت بزرگ به یک چالش بسیار مهم در برنامه‌های کاربردی تبدیل‌شده است. ازجمله این چالش‌ها می‌توان به تأثیر اندازه فضای حالت بر مدت زمان یادگیری و همچنین همکاری ناکارآمد میان عامل‌ها و عدم وجود هماهنگی مناسب در تصمیم‌گیری عامل‌ها اشاره کرد. همچنین هنگام استفاده از الگوریتم‌های یادگیری تقویتی نیز با چالش‌هایی نظیر دشواری تعیین هدف یادگیری مناسب و زمان طولانی همگرایی ناشی از یادگیری مبتنی بر آزمایش و خطا مواجه خواهیم بود. در این مقاله، با معرفی یک چارچوب ارتباطی برای سیستم‌های چندعاملی مشارکتی، تلاش شده چالش‌های فوق تا حدی برطرف شود. در راستای حل مشکلات مربوط به همگرایی، انتقال دانش به کار برده شده است که می‌تواند به شکل قابل‌توجهی در افزایش کارایی الگوریتم‌های یادگیری تقویتی موثر واقع شود. همکاری میان عامل‌ها با استفاده از عامل سرگروه و هماهنگی میان آنان توسط یک عامل هماهنگ‌کننده صورت می‌پذیرد. چارچوب پیشنهادی برای حل مسأله گله‌داری به کار رفته است و نتایج تجربی افزایش کارایی عامل‌ها را نشان می‌دهند.

کلمات کلیدی: سیستم‌های چندعامله مشارکتی، یادگیری تقویتی، انتقال دانش، مساله گله‌داری

Collaborative Multi-Agent Reinforcement Learning in Dynamic Environments using Knowledge Transfer for Herding Problem

Amin Nikanjam, Monireh Abdoos, Mahnoosh Mahdavi Moghadam

Abstract: Nowadays, collaborative multi-agent systems in which a group of agents work together to reach a common goal, are used to solve a wide range of problems. Cooperation between agents will bring benefits such as reduced operational costs, high scalability and significant adaptability. Usually, reinforcement learning is employed to achieve an optimal policy for these agents. Learning in collaborative multi-agent dynamic environments with large and stochastic state spaces has become a major challenge in many applications. These challenges include the effect of size of state space on learning time, ineffective collaboration between agents and the lack of appropriate coordination between decisions of agents. On the other hand, using reinforcement learning has challenges such as the difficulty of determination the appropriate learning goal or reward and the longtime of convergence due to the trial and error in learning. This paper, by introducing a communication framework for collaborative multi-agent systems, attempts to address some of these challenges in herding problem. To handle the problems of convergence, knowledge transfer has been utilized that can significantly increase the efficiency of reinforcement learning algorithms. Cooperation and Coordination and between the agents is carried out through the existence of a head agent in each group of agents and a coordinator agent respectively. This framework has been successfully applied to herding problem instances and experimental results have revealed a significant improvement in the performance of agents.

Keywords: Collaborative multi-agent system, Reinforcement learning, Knowledge transfer, Herding problem.

۱- مقدمه

سیستم چندعاملی به سیستمی گفته می‌شود که از تعدادی عامل خودگردان تشکیل‌شده است که هر یک به‌نوبه‌ی خود فعالیت داخلی داشته و در محیط خارج نیز با یکدیگر ارتباط دارند. برای داشتن ارتباطات داخلی موفق بین عامل‌ها، آن‌ها نیاز دارند که با یکدیگر همکاری کنند، هماهنگ باشند و مذاکره دوطرفه داشته باشند. سیستم‌های چندعاملی نسبت به سیستم‌های تک‌عاملی مزایای زیادی دارند: مقاوم بودن[1] و مقیاس‌پذیری[2]، مقاوم بودن به مفهوم این که اگر عاملی قادر به ادامه فعالیت نبود، عامل‌های دیگر وظایف وی را بر عهده بگیرند و مقیاس‌پذیری به مفهوم اینکه افزودن عامل جدید به محیط فرآیندی ساده باشد [۱]. همچنین سیستم‌های چندعاملی در اکثر شرایط کار می‌کنند، به این معنا که متمرکز نیستند و تصمیم‌گیری در آن‌ها توزیع‌شده است و چنانچه حتی بخشی از آن‌ها نیز از کار بیفتد بازهم به کار خود ادامه می‌دهند. همچنین این نوع سیستم‌ها برای محیط‌هایی با مقیاس وسیع[3] و محیط‌های ناشناخته نیز گزینه مناسبی نسبت به سیستم‌های تک عاملی به شمار می‌آیند.

یادگیری تقویتی یکی از روش‌هایی است که در سیستم‌های چندعاملی به کار می‌رود. ازنظر ریاضی یادگیری به‌صورت نگاشتی از فضای حالات به فضای اعمال بیان می‌شود. این روش، تعاملی است و یادگیری در آن با سعی و خطا بهبود می‌یابد، یعنی عامل یادگیرنده دانش خود را به‌صورت سعی و خطا از تعامل با محیط می‌گیرد. عامل در محیط قرارگرفته یک کنش انجام می‌دهد، پاسخی از محیط می‌گیرد و طبق آن پاسخ می‌فهمد عملی که انجام داده مناسب بوده است یا خیر. یادگیری تقویتی جهت حل مسائلی که در آن‌ها مدلی از محیط موجود نیست بسیار کارآمد عمل خواهد کرد زیرا از طریق آزمایش و خطا شناخت نسبی از محیط را به عامل می‌دهد. همگرایی در الگوریتم‌های یادگیری تقویتی پس از جستجوی گسترده محیط حاصل می‌شود که معمولاً زمان زیادی از این طریق هدر خواهد رفت. یکی از راه‌های کاهش این زمان استفاده از الگوریتم‌های انتقال دانش و همچنین الگوریتم‌های خلاقانه[4] است [۲].

در این مقاله، یک چارچوب برای ارتباط و همکاری در سیستم‌های چندعاملی مشارکتی مبتنی بر یادگیری تقویتی پیشنهاد شده است. در روش پیشنهادی عامل‌ها گروه‌بندی می‌شوند و تحت نظر سرگروه عمل می‌کنند. این چارچوب برای حل مسأله گله‌داری به کار گرفته شده است. در بخش دوم کارهای مرتبط با این حوزه مرور شده است. بخش سوم روش پیشنهادی را تشریح می‌نماید. بخش چهارم به جزئیات پیاده‌سازی و ارائه نتایج اختصاص دارد. جمع‌بندی و کارهای آینده در بخش پنجم ارائه شده‌اند.

۲- کارهای مرتبط

انتقال دانش یک نمونه از مفاهیم یادگیری ماشین است که از دانش جمع‌آوری‌شده درگذشته برای ارتقا یادگیری استفاده می‌کند [۲]. استدلال موردی[5] یکی از پراستفاده‌ترین روش‌ها در حوزه‌ی هوش مصنوعی است. در این روش دانش حاصل‌شده از موقعیت قبلی را جهت استفاده در موقعیت جدید بکار می‌گیرند به این شکل که موردهای مشابه با حالت فعلی در موقعیت‌های قبلی جستجو می‌شوند و دانش آن‌ها، در حالت فعلی استفاده می‌شود [۲]. استفاده از این انتقال دانش شامل مراحل زیر است [۲]: ۱) به دست آوردن تعریفی از حالت، ۲) اندازه‌گیری تشابه حالت جاری با حالت قبلی، ۳) بازیابی دانش مربوط به حالت‌های مشابه قبلی و ۴) استفاده مجدد از دانش حالت‌های قبلی در حالت مشابه فعلی. برای انتقال دانش به صورت کارا و مطمئن، ما به نگاشتی نیاز خواهیم داشت که دو بخش مبدأ و مقصد را که قرار است انتقال دانش مابین آن‌ها انجام شود، به هم مرتبط کند [۳].

درگذشته فعالیت‌های زیادی در زمینه پیاده‌سازی سیستم‌های چندعاملی مشارکتی انجام شده است که به عنوان نمونه می‌توان به کارهای انجام شده در حوزه تخصیص منابع اشاره کرد. تخصیص تعدادی منابع پراکنده به کاربران توسط عامل‌هایی که با همکاری یکدیگر و گاهی به‌تنهایی منابع را اختصاص خواهند داد یکی از پژوهش‌هایی است که در این حوزه انجام شده است [۴]. تنها راه ارتباط میان عامل‌ها اشتراک جداول سود آنهاست تا از این طریق اطلاعات کاملی نسبت به وضعیت تمام منابع داشته باشند [۴]. همچنین در حوزه کنترل ترافیک هوشمند از سیستم‌های چندعاملی مشارکتی استفاده شده است. کنترل جریان ترافیک یک راهکار مناسب برای کاهش تعداد تصادفات، کاهش مصرف سوخت و کاهش زمان انتظار است. در راستای کنترل ترافیک می‌توان از ابزارهایی همچون ایجاد محدودیت سرعت، استفاده از ابزارهایی جهت اطلاع از ترافیک و کنترل سیگنال ترافیک استفاده نمود [۵]. در هنگامی‌که هدف، کنترل سیگنال ترافیک باشد، به عامل تصمیم‌گیرنده (در مورد میزان سیگنال ترافیک) و عامل جمع‌کننده اطلاعات (در مورد ازدحام) نیاز است [۵].

حوزه‌ای که در این مقاله به آن پرداخته‌ایم مسئله گله‌داری[6] است. مسئله گله‌داری عبارت است از همکاری گروهی از عامل‌های چوپان برای هدایت گله‌ای از موجودات به سوی آغل، این مسئله نمونه‌ای از مسائل حوزه‌ی سیستم‌های چندعاملی مشارکتی است. در سال ۲۰۰۱ این مسئله با در نظر گرفتن یک گوسفند و یک سگ و با تکیه بر برنامه‌نویسی پویا[7] بررسی شده است [۶].

در سال ۲۰۰۲ این مسئله با یک چوپان بررسی شد. این چوپان با استفاده از نقشه راه[8] (نمایشی انتزاعی از فضای ممکن در محیط یا راهنمای محیط که بر اساس دید محلی عامل در حال به‌روزرسانی است و محل موانع را نشان می‌دهد) گله را دنبال می‌کند. از کاستی‌های روش ارائه‌شده

[1] Robustness
[2] Scalability
[3] Large scale
[4] Heuristic algorithms
[5] Case based reasoning
[6] Herding problem
[7] Dynamic programming
[8] Roadmap

در این مقاله می‌توان به عدم تلاش عامل در راستای جلوگیری از پراکنده شدن اعضای گله اشاره کرد [۷].

در سال ۲۰۰۴ برای کاهش تعداد دفعات تفکیک گله و همچنین جایابی هوشمندانه عامل راهکارهایی ارائه شد. راهکار خط مستقیم[1] موجب حرکت مستقیم چوپان در راستای گله می‌شود، این راهکار موجب تفکیک گله خواهد شد و مناسب نیست. راهکار ناحیه امن[2] با تخمین یک ناحیه دایره شکل در اطراف گله، اجازه‌ی ورود عامل چوپان به داخل این ناحیه را نمی‌دهد و از این رو تعداد دفعات تفکیک گله را کاهش می‌دهد [۸]. راهکار نقشه راه پویا[3] با درنظرگرفتن زیرمجموعه‌های یک گله در داخل ناحیه امن گله اصلی، فرض می‌کند اگر عامل وارد ناحیه امن زیر گله‌ها شود، آسیب قابل‌توجهی به گله اصلی وارد نمی‌شود [۸]. همچنین در این مقاله راهکارهایی جهت هدایت گله نیز ارائه شده است. راهکار مستقیم پشت گله[4]، گله را بر روی خط مستقیم به سمت محل پیشنهادی هدایت می‌کند. راهکار حرکت از یک طرف به طرف دیگر[5]، با تغییر مسیر از طرف راست به‌طرف چپ گله، سعی می‌کند هدایت گله به محل پیشنهادی بعدی‌اش انجام شود. راهکار چرخش گله، اگر در طول مسیر نیاز به چرخش گله بود، این امکان را فراهم می‌سازد. به‌صورت خلاصه می‌توان گفت، در این پژوهش روش‌هایی مطرح شد که یک چوپان بتواند یک گله مشخص را به شکل کاراتر هدایت کند. چوپانی به دو بخش نزدیک شدن[6] و هدایت[7] تقسیم شده و همچنین رفتارهایی برای چوپان نیز تعریف شد [۸].

در سال ۲۰۰۵ نشان داده شد که یک عامل چوپان به‌تنهایی نمی‌تواند یک گله بزرگ را به شکل مناسب هدایت کند و چندین چوپان می‌توانند کنترل مناسب‌تری بر روی گله‌های بزرگ و پیچیده داشته باشند [۹]. در این مقاله هیچ ارتباطی میان چوپان‌ها وجود ندارد و آن‌ها مستقل از هم هستند. چوپانی همانند گذشته به دو بخش نزدیک شدن و هدایت تقسیم می‌شود. در بخش نزدیک شدن دو راهکار جهت پیش‌بینی مکان بعدی هریک از عامل‌های چوپان در اطراف گله ارائه شد. مکان بعدی که هر عامل چوپان برای تأثیرگذاری بر گله انتخاب می‌کند نقطه هدایت نام دارد [۹]. در راهکار تقسیم برداری[8]، اگر k-1 چوپان، سمت چپ یک عامل چوپان باشند، این عامل در k-1 امین نقطه هدایت از چپ قرار خواهد گرفت [۹]. در راهکار کاهش حریصانه فاصله[9]، عامل چوپان به نزدیک‌ترین نقطه هدایت موجود می‌رود. جهت انتخاب نزدیک‌ترین نقطه هدایت از گراف دوبخشی[10] استفاده شده است.

در سال ۲۰۰۹ آزمایشی طراحی شد که در آن کاربران با استفاده از نشانگرهای لیزری حرکت چوپان را کنترل کنند. در این آزمایش با استفاده از متدهای برنامه‌ریزی به شکل دیداری تذکراتی به کاربران جهت جایابی مناسب داده می‌شد. در حقیقت در این مقاله مسئله چوپانی به‌عنوان نمونه‌ای از مسائل برنامه‌ریزی حرکتی در نظر گرفته‌شده است، که هدف طراح در آن یافتن مسیری برای حرکت چوپان است که در انتهای مسیر گله نیز به مقصد رسیده باشد [۱۰]. در ابتدا، برنامه با استفاده از نقشه راه و متدهای حرکتی ارائه‌شده در دو مقاله قبلی مسیری را از میانه گله تا مکان هدف محلی می‌یابد. سپس عامل چوپان گله را در مسیر پیشنهادی طراح حرکت می‌دهد. این مسیر ممکن است به دلایلی همچون نبود فضای کافی برای ادامه حرکت عامل (ناشی از پویایی محیط) مناسب نباشد، از این رو کاربران با تصحیح مسیر پیشنهادی تلاش می‌کنند چوپان بطور مناسبی هدایت شود. تعامل مناسب میان کاربر و طراح موجب یافتن راهکار مناسبی خواهد شد [۱۰].

در سال ۲۰۱۰ تحقیقی انجام شد که در آن محیط از یک سگ و تعدادی گوسفند تشکیل‌شده بود. برای جلوگیری از تفکیک گله از کوچک‌ترین دایره ممکن در اطراف گله استفاده شد: در صورت ورود عامل به داخل آن ناحیه اعضای گله فرار می‌کردند (ناحیه متراکم). همچنین هدایت گله با حرکت از یک طرف به طرف دیگر انجام شد. برای مسیریابی از الگوریتم A* استفاده شد که در آن مسیریابی به صورتی انجام می‌شد که عامل وارد ناحیه متراکم نشود. با توجه به پویایی محیط، مسیر انتخابی هرچند ساعت یک‌بار به‌روزرسانی خواهد شد. همچنین در این تحقیق یک انسان یا یک عامل هوشمند دیگر از طریق دستوراتی عامل سگ را در راستای هدایت گله راهنمایی می‌کند. سپس مسیر پیموده شده با مسیر حاصل از الگوریتم A* مقایسه خواهد شد و عملیات یادگیری نیز انجام خواهد شد [۱۱]. یک چارچوب جدید توسط زبان JACK در سال ۲۰۱۰ پیاده‌سازی شده است که مسیریابی در آن باز هم توسط الگوریتم A* انجام می‌شود [۱۲].

در سال ۲۰۱۲، سه مفهوم عدم استفاده از دانش پیشین، استفاده از دانش پیشین مربوط به خود و استفاده از دانش پیشین مربوط به سایر عامل‌ها بررسی و مقایسه شدند. مسئله چوپانی به‌عنوان یک بستر آزمایش، جهت مقایسه این سه مفهوم در نظر گرفته‌شده است. به عنوان نتیجه این آزمایش می‌توان گفت استفاده از دانش خود و سایر عامل‌ها تأثیر بسزایی بر روی عملکرد عامل‌ها داشته است [۱۳].

در سال ۲۰۱۳، با توجه به نیروهای جاذبه و دافعه وارد بر گله، راستای حرکت گله پیش‌بینی‌شده است. منظور از دافعه، دافعه میان گله و چوپان و دافعه میان اعضای دو گروه رقیب است و همچنین منظور از جاذبه، جاذبه میان اعضای یک گله است [۱۴].

در سال ۲۰۱۴، با استفاده از حرکت از یک طرف به طرف دیگر، مدل محلی جاذبه و دافعه میان اجزای مسئله، بررسی شد [۱۵]. اگر اعضای گله در فاصله‌ای دورتر از یک حد مشخص از چوپان قرار داشتند حرکت نمی‌کنند و یا تصادفی حرکت می‌کنند. اما اگر گوسفندان در فاصله مشخص

[1] Straight line
[2] Safe zone
[3] Dynamic roadmap
[4] Straight behind flock
[5] Side to side
[6] Approaching
[7] Steering
[8] Vector projection
[9] Greedy distance minimizing
[10] Bipartite graph

از چوپان قرار بگیرند، تحت دو نیروی زیر قرار خواهند گرفت: جاذبه میان اعضای گله نسبت به یکدیگر و دافعه میان اعضای گله و چوپان. چوپان با توجه به موقعیت عامل‌ها یکی از فعالیت‌های زیر را انجام خواهد داد [۱۵]: ۱) اگر تمام اعضای گله در راستای مناسبی نسبت به هدف محلی باشند، چوپان به سمت گله حرکت خواهد کرد و ۲) اگر حتی یکی از اعضای گله در راستای مناسبی نسبت به هدف محلی نهایی قرار نگرفته باشد، چوپان در راستای جمع‌آوری آن عامل و تشکیل یک گله واحد در راستای هدف محلی تلاش خواهد کرد.

در سال ۲۰۱۷، تعامل میان چوپان و هدف توسط یک تابع غیرخطی مدل شد [۱۶]. هنگامی‌که یک چوپان گله‌ای را انتخاب می‌کند، نزدیک‌ترین عضو گله به چوپان، دارای یک تابع حرکتی است که مجموع مقدار دافعه میان این عضو گله و چوپان و مقدار تمایل فرار این عضو از مقصد است. هنگامی‌که این عضو به مکان نزدیک به مقصد رسید و یا زمان طولانی از دنبال شدن‌اش گذشت و هنوز به نزدیکی مقصد نرسیده بود، چوپان به عضو دیگری از گله رجوع می‌کند. تعامل میان چوپان و دیگر اعضای گله نیز با مجموع مقادیر دافعه میان این عضو گله و چوپان (که شدت آن کمتر از دافعه میان نزدیک‌ترین عضو گله و چوپان است) و تمایل فرار این عضو از مقصد است [۱۶]. هدف اصلی از ارائه رابطه برای حرکت هدف‌های دور و نزدیک، طراحی یک کنترل‌کننده برای عامل‌های چوپان است که با استفاده از آن می‌توانند به‌صورت منظم اعضای دور و نزدیک گله را به مقصد برسانند [۱۶].

مسئله گله‌داری در حوزه شبکه‌های حسگر بی‌سیم هم کاربردهایی داشته است. استقرار حسگرها در یک محیط برای راه‌اندازی شبکه یکی از این کاربردها است [۱۸ و ۱۹]. یکی از نخستین کارها در این زمینه که در سال ۲۰۰۳ ارائه شده است رویکردی برای قرار دادن تیمی از حسگرها متحرک برای تشکیل یک شبکه حسگر است که از ربات‌های چوپان استفاده می‌کند [۱۸]. علت استفاده از مسئله گله‌داری آن است که ربات‌های متحرک توانایی مکان‌یابی و اجتناب از موانع را ندارند. بنابراین در این تیم غیرهمگن[1] دو نوع ربات با قابلیت‌های متفاوت به کار گرفته شده‌اند: رهبر و پیرو. ربات رهبر ربات‌های پیرو را بدون برخورد به موانع به مکان مناسب استقرار هدایت می‌کند. در این رویکرد رباتهای پیرو خط دید را تشکیل می‌دهند و با حفظ آن به کمک نشانه‌های بصری، مسیر طی شده توسط ربات پیرو را دنبال می‌کنند. نتایج شبیه‌سازی و پیاده‌سازی نشان می‌دهد که کار استقرار با دقت مناسبی انجام شده است. کاربردهای دیگری هم وجود دارد که صرفا از مفهوم تشکیل گروه یا گله استفاده کرده‌اند (هدایت ربات‌ها مطرح نبوده است) تا با سپردن وظایف و منابع بیشتر به برخی گره‌های شبکه تاخیر، مصرف انرژی و قابلیت اعتماد شبکه در مقایسه با نمونه‌های همگن ارتقا یابد [۲۰].

در این مقاله، با ارائه‌ی یک چارچوب ارتباطی مناسب میان عامل‌ها تلاش شده است روش جدیدی برای حل مسئله گله‌داری بصورت چندعاملی و با استفاده از امکان ارتباط بین عامل‌ها ارائه شود.

۳- روش پیشنهادی

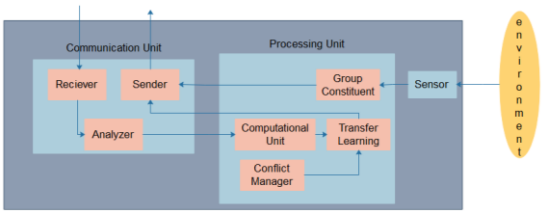
در این چارچوب عامل‌ها بر اساس نوع وظیفه به دو دسته‌ی عامل هماهنگ‌کننده و عامل اصلی تقسیم می‌شوند. وظیفه عامل هماهنگ‌کننده تشکیل گروه‌هایی از عامل‌های اصلی است (کار هدایت را انجام می‌دهند) که در ادامه به توضیح آن‌ها خواهیم پرداخت. اهداف اصلی این پژوهش عبارت‌اند از: ۱) بهبود یادگیری تقویتی از طریق انتقال دانش میان عامل‌ها و افزایش سرعت یادگیری تقویتی، ۲) تقویت همکاری میان اعضای یک گروه و رفع مشکل عدم همکاری مناسب عامل‌ها در سیستم‌های چندعاملی، ۳) تقویت هماهنگی میان گروه‌ها و رفع مشکل ناسازگاری میان تصمیمات گروه‌ها و ۴) یافتن پارامترهای یادگیری مناسب.

۴-۱ عامل هماهنگ‌کننده

برای پیاده‌سازی این محیط چندعاملی، ما به یک عامل هماهنگ‌کننده[2] نیاز داریم که بر اساس یک معیار مشخص، مانند مختصات محل قرارگیری عامل‌ها، گروه‌هایی را تشکیل دهد. پس از شکل‌گیری این گروه‌ها عضو اول هر گروه به‌عنوان سرگروه[3] شناخته می‌شود و بقیه‌ی اعضا، به عنوان زیرعضو آن گروه در نظر گرفته می‌شوند. عامل هماهنگ‌کننده، زیراعضای هر گروه را در قالب یک پیام متنی به سرگروه ارسال می‌کند. همچنین، به زیراعضا پیامی با مفهوم انتظار برای تصمیم‌گیری سرگروه ارسال خواهد شد. سرگروه به‌محض تصمیم‌گیری، به زیرعضوها اطلاع خواهد داد. شکل ۱ ساختار درونی عامل هماهنگ‌کننده را نشان می‌دهد.

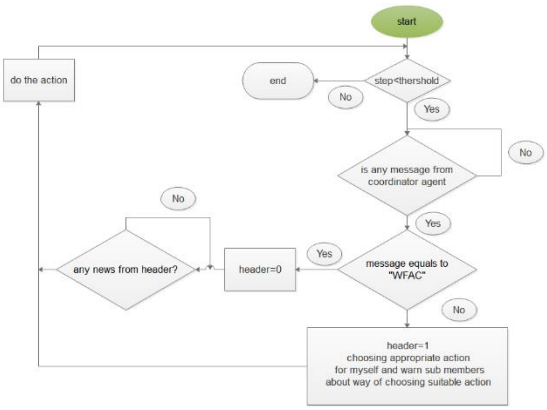
همان‌طور که در بالا اشاره کردیم، واحد تشکیل گروه[4] با کمک واحد محاسبات[5] و بر اساس یک معیار مشخص، گروه‌هایی را تشکیل می‌دهد. سپس واحد ارسال پیام، پیام‌های مربوط به سرگروه‌ها و همچنین زیر اعضا را ارسال خواهد کرد. پیام سرگروه‌ها، حاوی اطلاعات زیراعضا و پیام زیراعضا، یک‌رشته‌ی متنی به مفهوم انتظار برای تصمیم‌گیری سرگروه است. همه‌ی عامل‌های یک گروه، در پایان هر تکرار، جداول یادگیری حاصل از Q-learning خود را برای عامل هماهنگ‌کننده ارسال می‌کنند. واحد دریافت پیام، پس از دریافت جداول، آن‌ها را به واحد تحلیل پیام ارسال می‌کند. اگر پیام دریافتی از نوع جدول بود، واحد تحلیل پیام آن جدول را به واحد انتقال دانش ارسال خواهد کرد. در این واحد، دانش تمام جداول مشابه همه‌ی عامل‌ها، در یک جدول تجمیع می‌شود و این جدول تجمیعی جایگزین جداول محلی پیشین می‌شود و به تمام عامل‌ها ارسال می‌گردد.

[1] Heterogeneous
[2] Coordinator agent
[3] Header
[4] Group constituent
[5] Computational unit



شکل ۱: ساختار درونی عامل هماهنگ‌کننده

در هنگام تجمیع جداول، واحد مدیریت تعارض[1] وظیفه دارد اگر به عنوان مثال سطری از دو جدول مشابه مربوط به دو عامل مجزا دارای کلید یکسان بود، یکی از این دو سطر را بر اساس یک سیاست مشخص انتخاب کند و در جدول نهایی قرار دهد. در هر تکرار، الگوریتم بالا، به همین ترتیب ذکرشده اجرا می‌گردد. شکل ۲، نحوه‌ی عملکرد عامل را با وجود عامل هماهنگ‌کننده نشان می‌دهد. وظایف عامل هماهنگ‌کننده عبارت است از: ۱) دریافت مشاهدات تمام اعضای یک گروه، ۲) گروه‌بندی بندی اعضای گروه بر اساس یک معیار مشخص، ۳) انتساب عضو اول هر گروه به‌عنوان سرگروه، ۴) ارسال پیامی حاوی اطلاعات زیراعضا به سرگروه هر گروه و ۵) تجمیع جداول Q مربوط به هر وضعیت تمام اعضای گروه و سپس تجمیع اطلاعات تمام جداول در یک جدول واحد و ارسال این جدول به تمام عامل‌های گروه (انتقال دانش)، فرآیند انتقال دانش موجب افزایش سرعت یادگیری تقویتی خواهد شد.



شکل ۲: نحوه عملکرد عامل اصلی با حضور عامل هماهنگ‌کننده

۴-۲ عامل اصلی

همان‌طور که گفتیم، عامل هماهنگ‌کننده جزئی از گروه اصلی نیست پس لازم است راجع به ساختار عامل‌های گروه نیز صحبت کنیم. در شکل ۳ ساختار عامل‌های گروه نشان داده شده است. مشاهدات دریافتی از محیط به واحد تحلیل مشاهدات تحویل داده‌شده تا بخشی از خروجی گرافیکی شکل گیرد. سپس عامل اصلی طبق مراحل زیر رفتار خواهد کرد.

۴-۲/۱ تعیین هدف

در بخش تعیین هدف، هدف عامل مشخص می‌گردد. سپس عامل منتظر دریافت پیام از عامل هماهنگ‌کننده می‌ماند و پس از دریافت پیام، سرگروه و زیراعضا از یکدیگر تفکیک می‌شوند. زیراعضا منتظر می‌مانند تا سرگروه تصمیم‌گیری‌های لازم را انجام دهد.

۴-۲٫۲ تعیین وضعیت

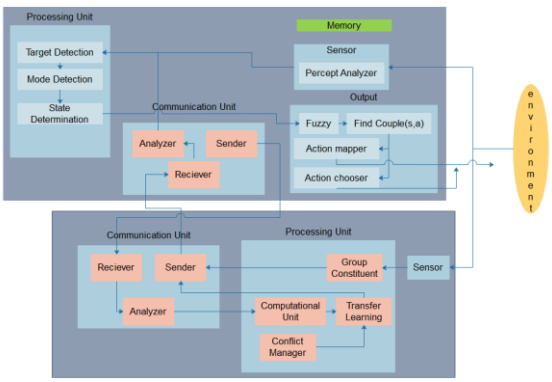
سرگروه با توجه به وضعیت خود و گروه‌اش نسبت به مقصد نهایی و همچنین نسبت به اهداف تجمیع‌شده (با دیگر زیراعضای گروه) وضعیتی را برای خود و گروهش انتخاب می‌کند و این وضعیت را به اطلاع زیراعضای گروه‌اش نیز می‌رساند. این کار توسط واحد تعیین وضعیت[2] انجام می‌شود.

۴-۲٫۳ تعیین حالت

حال لازم است وضعیت به شکل پارامتری ذخیره شود. عامل سرگروه وضعیت گروهش را نسبت به مقصد نهایی و هدف فعلی‌شان در واحد تعیین حالت[3]، به مقادیر عددی تبدیل می‌کند و این عدد یا مجموعه‌ای از اعداد حالت عامل را نمایش خواهند داد. عامل سرگروه حالت گروه را به دیگر اعضای گروه اطلاع می‌دهد.

۴-۲/۴ فازی‌سازی حالت

اگر محیط پیوسته باشد، لازم است عدد یا اعداد بیانگر حالت، وارد واحد فازی‌ساز شوند و به عدد یا اعدادی گسسته تبدیل شوند. در واحد فازی‌ساز عدد یا اعداد نشانگر حالت، هرکدام در بازه‌ای قرار خواهند گرفت که از این پس با عدد سردسته شناخته می‌شوند. به عنوان مثال فرض کنید عدد ۱.۵ نشانگر حالت باشد، وقتی این عدد وارد واحد فازی‌ساز شود چون در بازه‌ی ۰ تا ۲ قرار دارد با عدد ۰ فازی می‌شود و حالا دیگر ۰ نشانگر حالت است.



شکل ۳: ساختار درونی عامل اصلی

[1] Conflict manager
[2] Mode detection
[3] State determination

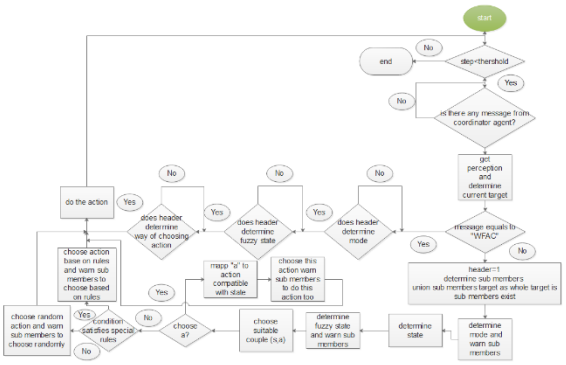
۲,۵-٤ انتخاب کنش

پس از مشخص شدن حالت، عامل سرگروه با رجوع به جدول Q مربوط به وضعیتش، زوج (حالت، کنش) را که دارای حالت مشابه با حالت آن‌ها است و بیشترین پاداش را دریافت کرده انتخاب می‌کند. حالا عامل می‌تواند با یک احتمال مشخص کنشی را که واحد انتخاب زوج مناسب توصیه کرده انتخاب کند و یا دست به انتخاب کنش‌های دیگر بزند. اگر عامل بخواهد از کنشی که واحد انتخاب زوج مناسب توصیه کرده پیروی کند، لازم است از طریق واحد نگاشت کنش، کنش موجود در جدول را به کنشی مناسب برای حالت خودش نگاشت کند. توجه کنید که از آنجایی که ما قصد داریم در ادامه تعریفی از حالت ارائه دهیم که به مختصات عامل بستگی نداشته باشد، پس تعداد زیادی حالت مشابه خواهیم داشت که به عنوان مثال کنش "حرکت به غرب" در یکی از این حالات معادل کنش "حرکت به شرق" در حالت دیگر باشد.

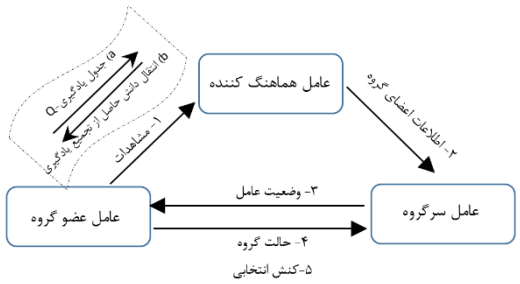
در واحد انتخاب کنش، اگر شرایط خاصی مهیا باشد، کنش، طبق قواعد و در غیر این صورت به صورت تصادفی انتخاب خواهد شد. عامل سرگروه اگر کنش را طبق قواعد انتخاب کرد به زیرعضوها خبر می‌دهد که آن‌ها نیز طبق قواعد کنش انتخاب کنند و اگر تصادفی انتخاب کرد به آن‌ها نیز خبر می‌دهد که کنش را تصادفی انتخاب کنند.

حالت و کنش همه‌ی عامل‌های گروه و همچنین مکان هدف فعلی در حافظه‌ی عامل سرگروه نگهداری می‌شود تا در تکرار بعدی، با توجه به تغییرات محیط پاداش تخصیص یابد. پس از به دست آوردن پاداش، زوج (حالت، کنش) در جدول مربوط به وضعیت عامل درج خواهد شد (برای هر وضعیت، عامل یک جدول جداگانه خواهد داشت این کار موجب می‌شود در هر بار رجوع به جدول Q، با جدول کوچک‌تری روبرو باشیم).

در یک جمله می‌توان گفت عامل هماهنگ‌کننده وظیفه دارد هماهنگی را میان گروه‌ها برقرار سازد و عامل سرگروه وظیفه دارد همکاری را میان اعضای یک گروه برقرار سازد. یکی از راه‌های برقراری هماهنگی مناسب، استفاده از زبان ارتباط عامل‌هاست. این زبان، همان روش انتقال پیام است [۱]. نوعی از هماهنگی را هم در تعامل درون گروه‌ها مشاهده می‌کنیم زیرا آن‌ها هم از پروتکل انتقال پیام به یکدیگر استفاده می‌کنند. منظور از همکاری همان تلاش سرگروه برای ایجاد یک روش مناسب انتخاب کنش برای همه‌ی زیراعضا است. در شکل ۴ نحوه‌ی عملکرد عامل اصلی را مشاهده خواهید کرد. شکل ۵ ارتباط بین عامل‌ها و ترتیب تبادل پیام‌ها را نشان می‌دهد.



شکل ۴: نحوه عملکرد عامل اصلی



شکل ۵: ارتباط میان عامل‌های مختلف و ترتیب ارسال پیام‌ها. ارتباط بین عامل هماهنگ‌کننده و عامل عضو گروه در مرحله انتقال تجربه ذکر شده است.

٤- پیاده‌سازی

محیط آزمایشی یک شبکه[1] ۱۰۰×۱۰۰ است و از نظر دسته‌بندی پاره‌ای قابل‌مشاهده و همچنین جز محیط‌های غیرقطعی و ترتیبی است. همچنین محیط ما محیطی پویا و گسسته و ناشناخته نیز هست. اطلاعات دریافتی هر عامل حاوی وضعیت چهار مربع ۸×۸ اطراف عامل هست. هر خانه‌ی این مربع ۸×۸ می‌تواند شامل عناصر زیر باشد [۱۷]: مانع[2]، گاو، عامل، آغل[3]. پویایی محیط ناشی از حرکت تصادفی گاوها و عامل‌های حریف است. در مجموع می‌توان اطلاعات دریافتی هر عامل را به بخش‌های زیر تقسیم نمود [۱۷]: ۱) مکان دقیق عامل در شبکه، ۲) محتوای مربع ۸×۸ اطراف عامل، ۳) شناسه‌ی گاوهای اطراف عامل به همراه مکان آن‌ها و ۴) مکان آغل خودی (اطلاعی راجع به مکان آغل گروه حریف ندارد).

در هر یک از خانه‌های این شبکه تنها یک عنصر می‌تواند در زمان مشخص قرار داشته باشد. کنش‌های قابل‌قبول برای هر عامل عبارت‌اند از حرکت به یکی از جهت‌های شمال، جنوب، شرق، غرب، شمال شرقی، شمال غربی، جنوب شرقی و جنوب غربی. مجموع تعداد گوسفندان قرارگرفته در آغل خودی امتیاز هر گروه از عامل‌ها محسوب می‌شود.

[1] Grid
[2] Obstacle
[3] Corral

۴-۱ پیاده‌سازی عامل هماهنگ‌کننده

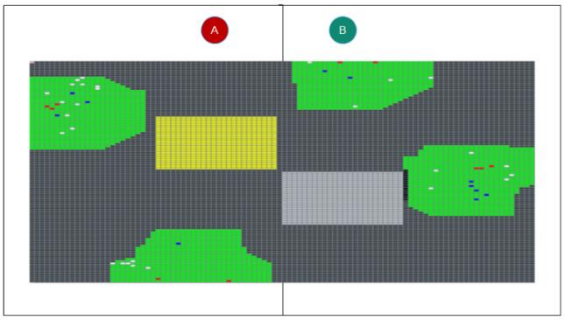
عامل هماهنگ‌کننده در هر تکرار از بازی با دریافت مختصات محل قرارگیری همه‌ی عامل‌های یک گروه و همچنین داشتن یک حد آستانه اقدام به تشکیل گروه‌ها می‌کند. عضو اول هر گروه به‌عنوان سرگروه شناخته می‌شود و بقیه‌ی اعضا زیرعضو این گروه هستند. سپس عامل هماهنگ‌کننده اطلاعاتی راجع به زیراعضا به هریک از سرگروه‌ها ارسال خواهد کرد و همچنین به زیراعضا پیامی حاوی یک رشته متنی جهت انتظار برای تصمیم‌گیری سرگروه ارسال خواهد شد.

۴-۲ تعیین هدف

هر عامل پس از دریافت اطلاعات چهار مربع ۸×۸ اطرافش وظایف زیر را بر عهده دارد: ۱) خوشه‌بندی گوسفندان بر اساس آستانه نزدیکی و تشکیل گله و ۲) انتخاب گله پرجمعیت به‌عنوان هدف فعلی. تا این مرحله هر عامل مشاهدات دریافتی خود را دسته‌بندی کرده و همچنین هدف فعلی‌اش یعنی دسته مشخص از گوسفندان را انتخاب کرده. حال لازم است صبر کند و منتظر دریافت پیامی از عامل هماهنگ‌کننده بماند. پس از دریافت پیام، اگر متن پیام "WFAC" بود، یعنی عامل زیر عضو یک گروه است و باید منتظر تصمیمات سرگروهش بماند و اگر رشته متنی جز این بود یعنی عامل سرگروه است. اگر رشته متنی خالی بود، یعنی عامل سرگروه، هیچ عامل همکاری ندارد و باید به تنهایی فعالیت کند، در غیراینصورت این رشته حاوی شناسه‌ی زیرعضوهایش همراه با یک جداکننده[1] است. اگر عامل سرگروه همکاران دیگری نیز داشته باشد لازم است اهداف فعلی همه آن‌ها به‌عنوان هدف فعلی تک‌تک اعضا در نظر گرفته شود پس لازم است از یک تابع تجمیع اهداف استفاده کنیم و اهداف کلیه‌ی اعضای یک گروه را به‌عنوان هدف فعلی تک‌تک آن‌ها در نظر بگیریم.

۴-۳ تعیین وضعیت

حال لازم است به سراغ تعیین وضعیت یک گروه برویم. همان‌طور که قبلاً هم گفته شد، منظور از وضعیت، کلمه‌ای است که بتواند وضعیت این گروه را نسبت به ورودی آغل و همچنین نسبت به دسته گوسفندان هدف توصیف کند. در شکل ۶ نمایی کلی از محیط را مشاهده خواهید کرد. در ناحیه زرد محل آغل را مشاهده می‌کنید. عامل‌های دوتیم به رنگ قرمز و آبی قابل‌مشاهده هستند. گوسفندان نیز به رنگ سفید در محیط قرار دارند. ناحیه‌ی خاکستری مربوط به بخش ناشناخته‌ی محیط است. همان‌طور که در شکل ۵ مشاهده می‌کنید، ما محیط را به دو بخش A و B تقسیم می‌نماییم. در بخش B ورودی آغل قرار دارد و اگر گروهی بخواهد در حالت چوپانی قرار گیرد حتماً باید در ناحیه B قرار گیرد. در ناحیه A زیرگروه‌ها تلاش می‌کنند گله را به ناحیه B برسانند تا بتوانند چوپانی کنند. پرسه‌زنی فقط در ناحیه‌ی A ممکن خواهد بود. منظور از پرسه‌زنی همان تلاش برای انتقال گله گوسفندان موردنظر به بخش B است.

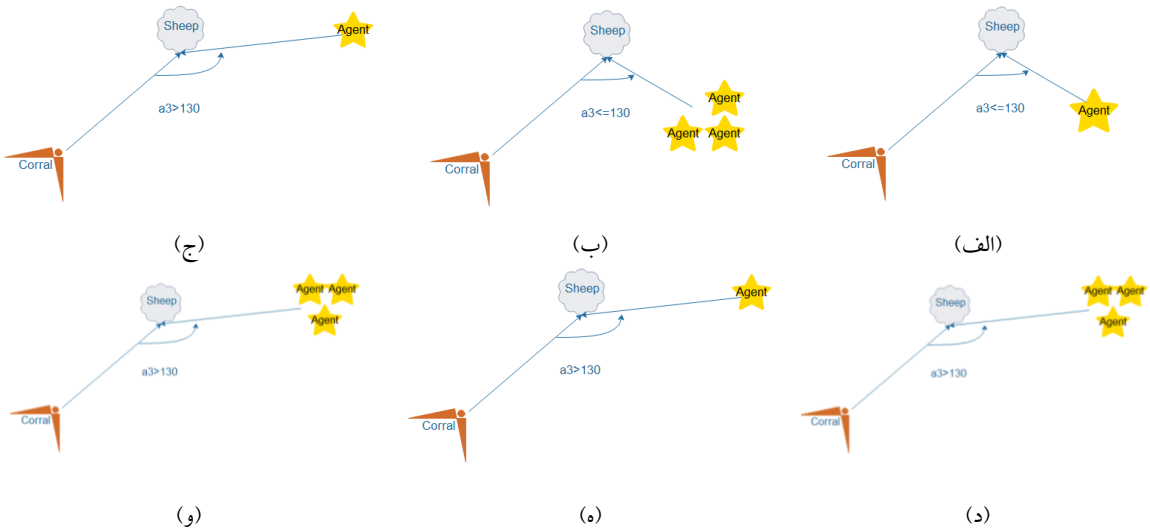


شکل ۶: نمایی کلی از محیط و نواحی مختلف آن.

وضعیت‌هایی که ما با آن روبرو هستیم عبارت‌اند از:

- دنبال کردن انفرادی[2]: این وضعیت در هر دو ناحیه A و B قابل وقوع است. در این وضعیت عامل‌ها نسبت به مقصد نهایی و همچنین گله موردنظرشان در وضعیتی مطابق شکل ۷-الف قرار خواهند داشت.
- دنبال کردن گروهی[3]: این وضعیت در هر دو ناحیه‌ی A و B ، قابل وقوع است. در این وضعیت بیش از یک عامل در وضعیتی مشابه وضعیت بالا قرار دارند، این وضعیت مشابه شکل ۷-ب است.
- چوپانی انفرادی[4]: این وضعیت تنها در ناحیه‌ی B قابل وقوع است. در این وضعیت عامل‌ها نسبت به مقصد نهایی و همچنین گله موردنظرشان در وضعیتی مطابق شکل ۷-ج قرار خواهند داشت.
- چوپانی گروهی[5]: این وضعیت تنها در ناحیه‌ی B قابل وقوع است. در این وضعیت بیش از یک عامل در وضعیتی مشابه وضعیت بالا قرار دارند، این وضعیت مشابه شکل ۷-د است.
- پرسه‌زنی انفرادی[6]: این وضعیت در ناحیه‌ی A قابل وقوع است. این وضعیت زمانی دیده می‌شود که عامل در ناحیه‌ی A در وضعیتی مطابق شکل ۷-ه قرار بگیرد.
- پرسه‌زنی گروهی[7]: این وضعیت در ناحیه‌ی A قابل وقوع است. این وضعیت زمانی رخ می‌دهد که بیش از یک عامل در وضعیتی مشابه وضعیت پرسه‌زنی انفرادی قرار بگیرند. در شکل ۷-و مشاهده می‌شود.

[1] Delimiter
[2] Single Going-after
[3] Group Going-after
[4] Single Herding
[5] Group Herding
[6] Single Prowling
[7] Group Prowling

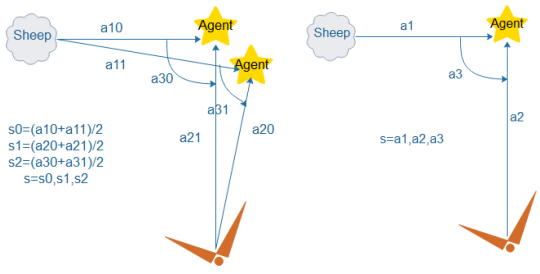


شکل ۷: ارتباط میان عامل‌های مختلف و ترتیب ارسال پیام‌ها. ارتباط بین عامل هماهنگ‌کننده و عامل عضو گروه در مرحله انتقال تجربه ذکر شده است.

- معلق[1]: این وضعیت در هر دو ناحیه‌ی A و B قابل وقوع است. در این وضعیت هیچ گله‌ای جهت هدایت در نزدیکی عامل و یا عامل‌ها قرار نخواهد داشت.

۴-۴ تعیین حالت

پس از تعیین وضعیت، عامل سرگروه وضعیت را به سایر زیراعضا اطلاع می‌دهد. اکنون لازم است حالت را تعیین کنیم. باید حالت را به‌گونه‌ای تعریف کنیم که وابسته به مختصات عامل نباشد و وضعیت عامل را منعکس کند. اگر عامل تنها باشد، تعیین حالت مطابق شکل ۸-الف است. همان‌طور که مشاهده می‌کنید فاصله عامل تا گله گوسفندان و فاصله عامل تا ورودی آغل و زاویه‌ی بین این سه عنصر، به همراه یک جداکننده، حالت عامل را تشکیل می‌دهند. اگر گروهی از عامل‌ها باشند حالت مطابق شکل ۸-ب محاسبه می‌شود. همان‌طور که مشاهده می‌کنید میانگین فاصله‌ی همه‌ی عامل‌های زیرگروه تا گله گوسفندان و میانگین فاصله همه‌ی عامل‌ها تا ورودی آغل و میانگین زاویه‌ی بین همه‌ی این عامل‌ها تا دو عنصر گله و ورودی آغل، به همراه یک جداکننده، حالت عامل را تشکیل می‌دهند.



شکل ۸: الف) حالت عامل تنها. ب) حالت عامل‌های گروهی

۴-۵ فازی‌سازی حالت

با توجه به پیوستگی محیط تعداد زیادی حالت خواهیم داشت، در این شرایط فازی‌سازی امر ضروری است. فازی‌سازی به مفهوم اینکه ما تمام اعداد قرارگرفته در یک بازه‌ی مشخص را با سردسته‌ی آن بازه معرفی خواهیم کرد. به عنوان مثال فاصله عامل تا گله را که یک عدد پیوسته است، به‌عنوان ورودی به تابع فازی‌ساز می‌دهیم و از خروجی یک عدد به‌عنوان شاخص بازه‌ای که عدد فاصله در آن قرار داشته دریافت می‌کنیم. پس ما سه عدد نشانگر حالت را به تابع فازی‌ساز خواهیم داد و به‌عنوان خروجی سه شاخص بازه‌ای که این اعداد در آن قرار داشتند را دریافت خواهیم کرد. این کار، باعث می‌شود تعداد حالات ما به شکل قابل‌توجهی کاهش یابد.

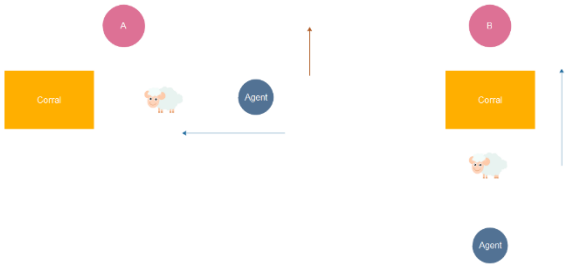
۴-۶ انتخاب کنش

اکنون با داشتن حالت می‌توانیم از جدول یادگیری مربوط به وضعیت گروه، تمام سطرهایی را که حالتی مشابه حالت این گروه را دارند استخراج کنیم و از بین همه‌ی آن‌ها، کنش سطری را انتخاب کنیم که مناسب‌ترین پاداش را دریافت کرده است. احتمال انتخاب این کنش با گذشت زمان افزایش می‌یابد. درصورتی‌که این کنش را انتخاب کردیم لازم است آن‌ها، به کنشی سازگار با وضعیت‌مان نگاشت کنیم (شکل ۹).

فرض کنید عامل ما در هر دو حالت A و B دارای حالت فازی‌شده‌ی یکسانی باشد و ما فعلاً در حالت A باشیم. فرض کنید با توجه به جدول یادگیری مربوط به این وضعیت، کنشی که بهترین پاداش را به ما داده، مطابق شکل B، کنش "حرکت به شمال" باشد. پس در وضعیت A هم ما باید کنش "حرکت به شمال" را انتخاب کنیم که انتخابی نامناسب است. در اینجا لزوم وجود یک تابع نگاشت محسوس است. بنابراین ما به تابعی نیاز خواهیم داشت که بر اساس مفهوم بیشترین شباهت میان حالت‌ها عمل

[1] Suspended

کند؛ یعنی اختلاف میان حالت جاری و حالتی که با انتخاب کنش به آن منتقل شدیم را به‌عنوان کنشی که باید در جدول یادگیری قرار بگیرد، ذخیره کند. پس هرگاه خواستیم از جدول یادگیری استفاده کنیم، کنشی را انتخاب می‌کنیم که تغییراتی مشابه آنچه در جدول یادگیری ذخیره‌شده را فراهم کند. سرگروه پس از یافتن کنش نگاشت‌شده آن را انتخاب می‌کند و به زیراعضا نیز اطلاع می‌دهد که آن کنش را انتخاب کنند؛ اما اگر سرگروه از کنش پیشنهادی جدول یادگیری استفاده نکرد، باید بررسی کند که آیا گروهش در شرایطی قرار دارند که بتوانیم از فنون خلاقانه استفاده کند یا خیر، اگر در چنین شرایطی قرار نداشتیم کنشی به‌صورت تصادفی انتخاب می‌کنیم و به زیراعضا اطلاع می‌دهیم که آن‌ها نیز کنشی تصادفی انتخاب کنند.



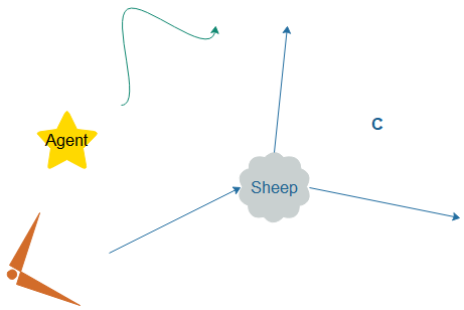
شکل ۹: الف) حالت عامل تنها. ب) حالت عامل‌های گروهی

۴-۷ فنون خلاقانه

همان‌طور که در بخش مقدمه گفتیم در هنگام استفاده از روش‌های یادگیری تقویتی، استفاده از الگوریتم‌های خلاقانه، موجب افزایش سرعت و بهبود کارایی خواهد شد. الگوریتم‌های خلاقانه‌ای که ما استفاده کردیم عبارت‌اند از: تعریف تابع چرخش و تعریف حالت هدف.

۴-۷/۱ تابع چرخش

همان‌طور که در شکل ۱۰ مشاهده می‌کنید، اگر در وضعیت دنبال کردن انفرادی و یا دنبال کردن گروهی بودیم، لازم است با انجام کنش‌هایی وارد ناحیه C شویم تا بتوانیم وارد وضعیت چوپانی انفرادی و یا چوپانی گروهی بشویم. دنباله کنش‌هایی که ما را وارد ناحیه‌ی C می‌کنند، از طریق تابع چرخش انتخاب خواهند شد.

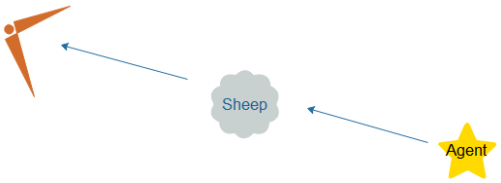


شکل ۱۰: نحوه عملکرد تابع چرخش

۴-۷/۲ حالت هدف

اگر در وضعیت چوپانی انفرادی و یا چوپانی گروهی بودیم و عامل و گله و ورودی آغل در یک راستا، مطابق شکل ۱۱ قرار گرفته بودند، کنش مطلوب کنشی است که در راستای میانه‌ی گله انجام پذیرد.

اگر کنش سرگروه بر اساس یکی از دو قاعده‌ی بالا انتخاب شود، او به زیراعضا خبر می‌دهد که آن‌ها نیز مطابق قاعده‌ای که وی کنش انتخاب کرده، کنش انتخاب کنند. اگر سرگروه در هیچ‌یک از این مراحل کنش انتخاب نکرد، یک کنش به‌صورت تصادفی انتخاب خواهد کرد و به زیراعضا خبر می‌دهد تصادفی کنش انتخاب کنند. سپس حالت و وضعیت جاری در یک متغیر واسطه قرار می‌گیرد تا در تکرار بعدی، با توجه به شرایط محیط، پس از انجام کنش همه‌ی عامل‌ها، پاداش محاسبه شود و در جدول یادگیری مربوط به وضعیت سابق قرار گیرد.



شکل ۱۱: تعیین حالت هدف

۴-۸ تابع پاداش

برای هر وضعیت مطابق روابط زیر پاداش تعیین خواهد شد:

الف. دنبال کردن انفرادی

$$R = \alpha(P_{agent}.P_{cent}.P_{corral})_{prev} - \alpha(P_{agent}.P_{cent}.P_{corral})_{current} \quad (۱)$$

ب. دنبال کردن گروهی

$$R = avg_{\alpha(P_{agent}.P_{cent}.P_{corral})prev} - avg_{\alpha(P_{agent}.P_{cent}.P_{corral})current} \quad (۲)$$

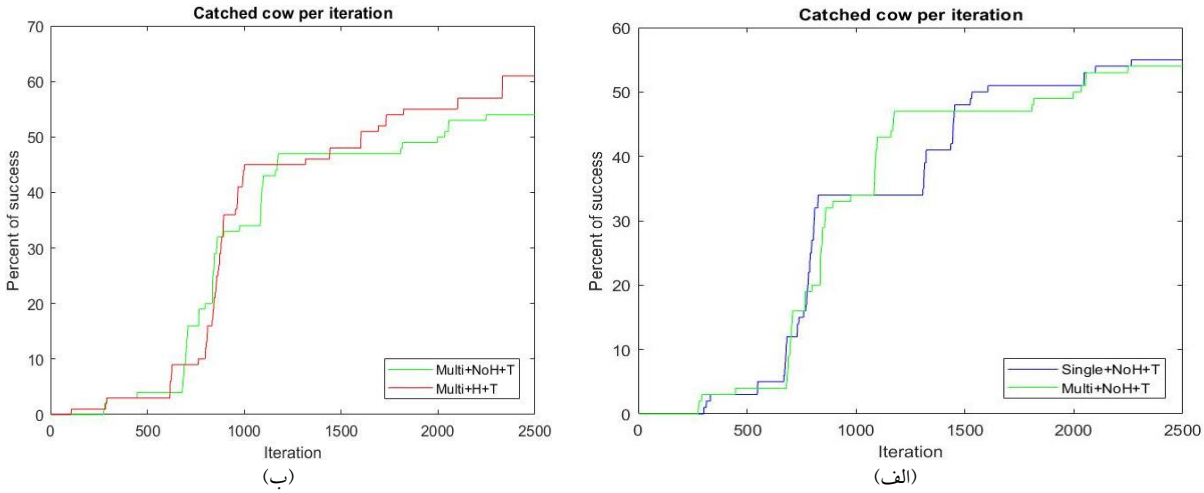
ج. چوپانی انفرادی

$$R = d(P_{agent}.P_{corral})_{current} - d(P_{agent}.P_{corral})_{prev} \quad (۳)$$

د. چوپانی گروهی

$$R = avg_{d(P_{agent}.P_{corral})current} - avg_{d(P_{agent}.P_{corral})prev} \quad (۴)$$

ه. پرسه‌زنی انفرادی



شکل ۱۲: مقایسه نسبت تعداد گوسفندان در آغل الف: تک‌عاملی و چندعاملی ب: با وجود توابع خلاقانه و بدون وجود توابع خلاقانه

$$R = \alpha(P_{agent}.P_{corral}.X^{+})_{current} - \alpha(P_{agent}.P_{corral}.X^{+})_{prev} \quad (۵)$$

و. پرسه‌زنی گروهی

$$R = avg_{\alpha(P_{agent}.P_{corral}.X^{+})_{current}} - avg_{\alpha(P_{agent}.P_{corral}.X^{+})_{prev}} \quad (۶)$$

در این روابط از نماد P_i برای نمایش موقعیت مکانی عنصر i، d(A,B) برای فاصله اقلیدسی بین A و B، از α برای زاویه بین عامل(ها)، گله و آغل (مطابق شکل‌های ۷ و ۸) و از X^{+} برای موقعیت گله در حالت پرسه‌زنی (وقتی در حالت چوپانی یا دنبال کردن یک گله یا گروه خاص نیستیم) استفاده شده است. در محاسبه پاداش اصول کلی رعایت شده است که به آنها اشاره می‌شود. در حالت چوپانی، فاصله عامل نسبت به آغل محاسبه می‌شود و هر چه عامل بیشتر به آغل نزدیک شود، پاداش بیشتری می‌گیرد. بنابراین تفاضل فاصله (یا میانگین آن) بین دو گام زمانی پی‌درپی (جاری و قبلی) ملاک پاداش است. در حالت پرسه‌زنی و دنبال کردن، بهبود زاویه بین عامل(ها)، گله و آغل (مطابق شکل‌های ۷ و ۸) بین دو گام زمانی پی‌درپی (جاری و قبلی) یا به تعبیر دیگر تلاش عامل برای قرار گرفتن در محل مناسب نسبت به آغل و گله ملاک پاداش است.

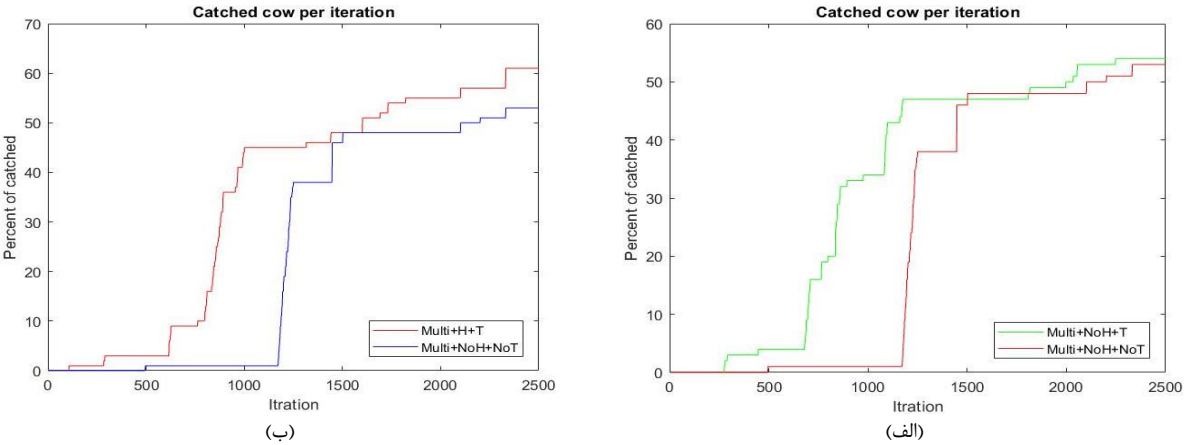
۴-۹ تحلیل نتایج

نتایج اجراها در چهار شرایط زیر در شکل ۱۲ نمایش داده شده است:

- تک‌عاملی بدون توابع خلاقانه و همراه با انتقال دانش
- چندعاملی بدون توابع خلاقانه و همراه با انتقال دانش
- چندعاملی همراه با توابع خلاقانه و همراه با انتقال دانش
- چندعاملی بدون توابع خلاقانه و بدون انتقال دانش

همان‌طور که در شکل ۱۲-الف مشاهده می‌کنید، در بعضی از تکرارها، وجود چندین عامل موجب هدایت درصد بیشتری از گوسفندان به سمت آغل شده است. درمجموع می‌توان گفت، سیستم‌های چندعاملی کنترل مناسب‌تری بر روی اهداف دارند. همان‌طور که در شکل ۱۲-ب مشاهده می‌کنید استفاده از الگوریتم‌های خلاقانه موجب افزایش سرعت یادگیری تقویتی خواهد شد و همچنین نتایج بهتری را نیز به ارمغان خواهد آورد.

همان‌طور که در شکل ۱۳-الف مشاهده می‌کنید انتقال دانش تأثیر بسزایی بر روی نتایج نهایی خواهد داشت. نرخ انتقال دانش در این شکل برابر ۰/۳۲۶۳ است. پرش ابتدا و کارآیی مجانبی نیز در ابتدا و انتهای نمودارها مشاهده می‌شود. در شکل ۱۳-ب تأثیر همزمان انتقال دانش و توابع خلاقانه مشاهده می‌شود: استفاده از الگوریتم‌های خلاقانه و انتقال دانش تأثیر بسزایی در نتایج نهایی داشته است به ترتیبی که بیش از ۱۳ درصد بهبود را نشان می‌دهد. شکل ۱۲ و ۱۳ نتایج بدست آمده در ۲۵۰۰ تکرار اول را نشان می‌دهد که اختلاف و تغییرات عملکرد دو الگوریتم را بهتر نشان می‌دهد. نرخ انتقال دانش و تعداد تکرارها با سعی و خطا و براساس نتایج بدست آمده در حدود ۵۰ تلاش مختلف برای تعیین پارامترهای مناسب بدست آمده‌اند. توجه به سربار اجرای روش پیشنهادی هم مانند هر الگوریتم دیگر ضروری است. نتایج بدست آمده نشان می‌دهد که روش پیشنهادی (استفاده همزمان از چندعاملی، توابع خلاقانه و انتقال دانش) در مجموع کمتر از ۲۰ درصد به زمان اجرای الگوریتم گله‌داری ساده (تک‌عاملی بدون توابع خلاقانه و انتقال دانش) می‌افزاید. بهبود پیاده‌سازی با هدف کاهش سربار می‌تواند یکی از اهداف برای پژوهش‌های آتی باشد. برای مقایسه روش پیشنهادی با سایر روش‌ها در حوزه چندعاملی و مسئله گله‌داری، ایجاد شرایط آزمایشی یکسان برای مقایسه منصفانه ضروری است. متأسفانه در بررسی ما روشی که بتواند در شرایط مشابه آزمایش شود یافته نشد. بنابراین مقایسه و ارزیابی روش پیشنهادی صرفا در محیط پیاده‌سازی شده در این پژوهش و با درنظر گرفتن

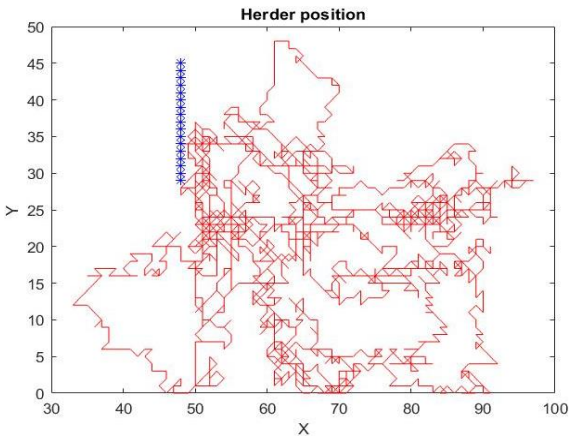


شکل ۱۳: مقایسه نسبت تعداد گوسفندان در آغل الف: با انتقال دانش و بدون انتقال دانش ب: با وجود همزمان توابع خلاقانه و انتقال دانش

شرایط مختلفی که عملکرد روش پیشنهادی را تا حد امکان مشخص کند انجام شد.

در شکل ۱۴ مختصات محل قرارگیری یکی از چوپان‌ها در شبیه‌سازی نمایش داده شده است. نقاط آبی‌رنگ، ورودی آغل را نمایش می‌دهند. اگر عامل در نقاط سمت راست ورودی آغل قرار گیرد و در حال کاهش x باشد آنگاه در وضعیت چوپانی قرار دارد و اگر در حال افزایش x باشد در وضعیت دنبال کردن قرار خواهد داشت. اگر چوپان در سمت چپ ورودی آغل قرار داشته باشد در وضعیت پرسه‌زنی قرار خواهد داشت.

بطور خلاصه می‌توان گفت حالت چندعاملی توانایی هدایت گله‌های بزرگتر و پیچیده‌تری را نسبت به تک‌عاملی دارد. همچنین در هنگام استفاده از انتقال دانش، پرش در ابتدا بسیار سریع‌تر از هنگامی که که از انتقال دانش استفاده نمی‌شود رخ خواهد داد. بهترین نتایج، هنگام استفاده از توابع خلاقانه همراه با انتقال دانش حاصل خواهد شد.



شکل ۱۴: نمونه‌ای از مکان قرار گرفتن عامل در محیط

۵- نتیجه‌گیری و کارهای آینده

در این مقاله به ارائه چارچوبی جهت یادگیری مشارکتی عامل‌ها در محیط‌های پویا پرداختیم. با توجه به اینکه در محیط‌های چندعاملی، اغلب با فضای حالت بزرگی مواجه هستیم و از طرفی دیگر اگر محیط پویا و غیرقطعی نیز باشد، یادگیری عامل‌ها یک مسئله چالش برانگیز خواهد شد. استفاده از راهکار انتقال دانش، همانطور که نشان داده شده است، می‌تواند در راستای افزایش کارایی موثر واقع شود. چارچوب ارائه شده در مسئله گله‌داری ارزیابی شده است. در این چارچوب، عامل‌ها به دو نوع عامل هماهنگ‌کننده و عامل اصلی تقسیم شده‌اند و معماری و چگونگی ارتباط آنها با یکدیگر مشخص شده است. نتایج بدست آمده نشان می‌دهد بهبود عملکرد حاصل از روش پیشنهادی قابل قبول است. کارهای آینده عبارت هستند از: ارزیابی دقیق و تلاش برای کاهش یا توزیع سربار محاسباتی، توسعه و ارزیابی این چارچوب در سایر حوزه‌های کاربردی که به صورت سیستم چندعاملی مشارکتی باشند، تعمیم چارچوب پیشنهادی جهت کاربرد در سیستم‌های ناهمگن که عامل‌های قابلیت‌های متفاوت دارند، استفاده از آنتولوژی برای ارتباط میان عامل‌ها است.

مراجع

[1] Glavic, M., “Agents and multi-agent systems: A short introduction for power engineers”, University of Liege-Electrical engineering and computer science department, 2006.

[2] Celiberto Jr, Luiz A., Jackson P. Matsuura, Ramón López De Màntaras, and Reinaldo AC Bianchi. "Using transfer learning to speed-up reinforcement learning: a cased-based approach." In *Robotics Symposium and Intelligent Robotic Meeting (LARS), 2010 Latin American*, pp. 55-60. IEEE, 2010.

[3] Taylor, Matthew E., and Peter Stone. "Transfer learning for reinforcement learning domains: A survey" *Journal of Machine Learning Research*, 10, 1633-1685, 2009.

[4] Wu, Jun, Xin Xu, Pengcheng Zhang, and Chunming Liu. "A novel multi-agent reinforcement learning approach for job scheduling in Grid computing." *Future Generation Computer Systems*, 27(5), 430-439, 2011.

[5] Khamis, Mohamed A., and Walid Gomaa. "Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework." *Engineering Applications of Artificial Intelligence*, 29, 134-151, 2014.

[6] Kachroo, Pushkin, Samy A. Shedied, John S. Bay, and Hugh Vanlandingham. "Dynamic programming solution for a class of pursuit evasion problems: the herding problem." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 31(1), 35-41, 2001.

[7] Bayazit, O. Burchan, Jyh-Ming Lien, and Nancy M. Amato. "Better group behaviors using rule-based roadmaps." In *Algorithmic Foundations of Robotics V*, pp. 95-111. Springer, Berlin, Heidelberg, 2004.

[8] Lien, Jyh-Ming, O. Burchan Bayazit, Ross T. Sowell, Samuel Rodriguez, and Nancy M. Amato. "Shepherding behaviors." In *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 4159-4164. IEEE, 2004.

[9] Lien, Jyh-Ming, Samuel Rodriguez, Jean-Phillipe Malric, and Nancy M. Amato. "Shepherding behaviors with multiple shepherds." In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2005)*, pp. 3402-3407. IEEE, 2005.

[10] Lien, Jyh-Ming, and Emlyn Pratt. "Interactive Planning for Shepherd Motion." In *AAAI Spring Symposium: Agents that Learn from Human Teachers*, pp. 95-102. 2009.

[11] Cowling, Peter I., and Christian Gmeinwieser. "AI for Herding Sheep." In Proceedings of the Sixth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2010), pages 2–7, 2010.

[12] Yadav, Nitin, Chenguang Zhou, Sebastian Sardina, and Ralph Rönnquist. "A BDI agent system for the cow herding domain." *Annals of mathematics and artificial intelligence*, 59(3-4), 313-333, 2010.

[13] Dow, Steven, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. "Shepherding the crowd yields better work." In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pp. 1013-1022. ACM, 2012.

[14] Strömbom, Daniel. "Attraction based models of collective motion." PhD dissertation, Uppsala university, Department of Mathematics, 2013.

[15] Strömbom, Daniel, Richard P. Mann, Alan M. Wilson, Stephen Hailes, A. Jennifer Morton, David JT Sumpter, and Andrew J. King. "Solving the shepherding problem: heuristics for herding autonomous, interacting agents." *Journal of the royal society interface*, 11, 2014.

[16] Licitra, Ryan A., Zachary D. Hutcheson, Emily A. Doucette, and Warren E. Dixon. "Single agent herding of n-agents: A switched systems approach." *IFAC-PapersOnLine*, 50(1), 14374-14379, 2017.

[17] https://multiagentcontest.org/2008/protocol.pdf, (last access on September 2018)

[18] Parker, Lynne E., Balajee Kannan, Xiaoquan Fu, and Yifan Tang. "Heterogeneous mobile sensor net deployment using robot herding and line-of-sight formations." In Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), vol. 3, pp. 2488-2493. IEEE, 2003.

[19] Strumberger, Ivana, Nebojsa Bacanin, Slavisa Tomic, Marko Beko, and Milan Tuba. "Static drone placement by elephant herding optimization algorithm." In *2017 25th Telecommunication Forum (Telfor)*, pp. 1-4. IEEE, 2017.

[20] Stathopoulos, Thanos, Lewis Girod, John Heidemann, and Deborah Estrin. "Mote herding for tiered wireless sensor networks.", Technical Report No. 58, Center for Embedded Networked Computing, University of California, Los Angeles, 2005.