انتقال دانش تنظیم‌شده برای یادگیری تقویتی چندعاملی

نیلوفر علوی و جعفر طهمورث نژاد*
دانشگاه صنعتی ارومیه دانشکده مهندسی فناوری اطلاعات و کامپیوتر،
دانشگاه صنعتی ارومیه، ارومیه، ایران

چکیده

یادگیری تقویتی به آموزش مدل‌های یادگیری ماشین برای اتخاذ تصمیمات متوالی اشاره می‌کند که در آن یک عامل از طریق تعامل با محیط، آموزش دیده، نتایج این تعامل را مشاهده کرده و بر این اساس، پاداش مثبت یا منفی دریافت می‌کند. یادگیری تقویتی کاربردهای زیادی برای سیستم‌های چندعاملی به خصوص در محیط‌های پویا و ناشناخته دارد. با این حال، بیش‌تر الگوریتم‌های یادگیری تقویتی چندعاملی با مشکلاتی همچون پیچیدگی محاسباتی نمایی برای محاسبه فضای حالت مشترک مواجه هستند که منجر به عدم مقیاس‌پذیری الگوریتم‌ها درمسائل چندعاملی واقعی می‌شود. کاربردهای یادگیری تقویتی چندعاملی را می‌توان از فوتبال ربات‌ها، شبکه‌ها، محاسبات ابری، زمانبندی شغل تا اعزام نیروی واکنشی دسته‌بندی کرد. در این مقاله یک الگوریتم جدید به نام انتقال دانش تنظیم‌شده برای یادگیری تقویتی چندعاملی (RKT-MARL) معرفی می‌شود که براساس مدل تصمیم‌گیری مارکوف کار می‌کند. این الگوریتم برخلاف روش‌های یادگیری تقویتی سنتی، مفاهیم تعاملات پراکنده و انتقال دانش را برای رسیدن به تعادل بین عامل‌ها استفاده می‌کند. علاوه‌بر این، RKT-MARL از سازوکار مذاکره برای یافتن مجموعه تعادل و از روش کمینه واریانس برای انتخاب بهترین عمل در مجموعه تعادل به‌دست‌آمده استفاده می‌کند. همچنین الگوریتم پیشنهادی، دانش مقادیر حالت-عمل را در میان عامل‌های مختلف انتقال می‌دهد. از طرفی، الگوریتم RKT-MARL مقادیر Q را در حالت‌های هماهنگی به عنوان ضریبی از اطلاعات محیطی جاری و دانش قبلی مقداردهی می‌کند. به‌منظور ارزیابی عملکرد روش پیشنهادی، یک گروه از آزمایش‌ها بر روی پنج بازی جهانی انجام‌شده و نتایج حاصل بیانگر همگرایی سریع و مقیاس‌پذیری بالا در RKT-MARL است.

واژگان کلیدی: یادگیری تقویتی چندعاملی ، انتقال دانش، تعادل‌های متا و نش، تنظیم‌پذیری، تعاملات پراکنده، مذاکره عامل‌ها.

Regularized Knowledge Transfer for Multi-Agent Reinforcement Learning

Niloofar Alavi & Jafar Tahmoresnezhad*
Faculty of IT & Computer Engineering, Urmia University of Technology, Urmia, Iran

Abstract

Reinforcement learning (RL) refers to the training of machine learning models to make a sequence of decisions on which an agent learns by interacting with its environment, observing the results of interactions and receiving a positive or negative reward, accordingly. RL has many applications for multi-agent systems, especially in dynamic and unknown environments. However, most multi-agent reinforcement learning (MARL) algorithms suffer from some problems specifically the exponential computational complexity to calculate the joint state-action space, which leads to the lack of scalability of algorithms in realistic multi-agent problems. Applications of MARL can be categorized from robot soccer, networks, cloud computing, job scheduling, and to optimal reactive power dispatch.

In the area of reinforcement learning algorithms, there are serious challenges such as the lack of application of equilibrium-based algorithms in practice and high computational complexity to find

* نویسندهٔ عهده‌دار مکاتبات
* Corresponding author

• تاریخ ارسال مقاله: ۱۳۹۸/۵/۱۱ • تاریخ پذیرش: ۱۴۰۲/۹/۲۰ • تاریخ انتشار: ۱۴۰۲/۱۲/۲۹ • نوع مطالعه: پژوهشی

equilibrium. On the other hand, since agents have no concept of equilibrium policies, they tend to act aggressively toward their goals, which it results the high probability of collisions.

Consequently, in this paper, a novel algorithm called Regularized Knowledge Transfer for Multi-Agent Reinforcement Learning (RKT-MARL) is presented that relies on Markov decision process (MDP) model. RKT-MARL unlike the traditional reinforcement learning methods exploits the sparse interactions and knowledge transfer to achieve an equilibrium across agents. Moreover, RKT-MARL benefits from negotiation to find the equilibrium set. RKT-MARL uses the minimum variance method to select the best action in the equilibrium set, and transfers the knowledge of state-action values across various agents. Also, RKT-MARL initializes the Q-values in coordinate states as coefficients of current environmental information and previous knowledge. In order to evaluate the performance of our proposed method, groups of experiments are conducted on five grid world games and the results show the fast convergence and high scalability of RKT-MARL. Therefore, the fast convergence of our proposed method indicates that the agents quickly solve the problem of reinforcement learning and approach to their goal.

Keyword: Multi-agent reinforcement learning, Knowledge transfer, Meta and Nash equilibriums, Regularization, Sparse interactions, Agents negotiations.

۱- مقدمه

دانشمندان و پژوهش‌گران در حوزه سیستم‌های چندعاملی و یادگیری ماشین علاقه بسیاری به یادگیری‌های چندعاملی نشان داده‌اند. یادگیری چندعاملی یک تکنیک کاربردی برای عامل‌ها است تا بتوانند رفتارهای هماهنگ و مؤثری را در سیستم‌های چندعاملی بیاموزند. در یادگیری چندعاملی، فرآیندهای یادگیری توزیع‌شده و هم‌زمان، محیط یادگیری را برای عامل‌ها پویامی‌کنند. توسعه یک رویکرد یادگیری مؤثر برای هماهنگ‌کردن رفتارهای عامل‌ها در یک محیط پویا، به‌خصوص زمانی که عامل‌ها ساختار محیط را نمی‌دانند و فقط مشاهدات محلی از محیط دارند، مسئله دشواری است[۱].

یکی از پرکاربردترین روش‌های یادگیری در سیستم‌های چندعاملی، یادگیری تقویتی‌است. در فرآیند یادگیری تقویتی، یک عامل، فضای استراتژی‌های ممکن را بررسی کرده و بازخوردی را در مورد نتیجه انتخاب‌هایی که دارد، دریافت می‌کند [۲]. بنابراین عامل یاد می‌گیرد که برای دریافت پاداش بیشتر از محیط، چه عمل‌هایی را باید انتخاب کند. نکته قابل‌توجه در یادگیری تقویتی این است که انجام یک عمل توسط عامل علاوه‌بر دریافت پاداش فوری، بر حالت‌های بعدی و همچنین مجموع پاداش‌ها نیز تأثیر می‌گذارد [۳، ۴و۵].

یکی از تکنیک‌های کلیدی در یادگیری چندعاملی، یادگیری تقویتی چندعاملی[1] (MARL) است که توسعه‌ای از یادگیری تقویتی در محیط‌های چندعاملی‌است. یادگیری تقویتی چندعاملی، روشی برای عامل‌های خودمختار فراهم می‌کند تا بتوانند مسائل تصمیم‌گیری متوالی را در سیستم‌های چندعاملی حل کنند [۵ و ۶].

چندین مدل ریاضی به‌عنوان چارچوب یادگیری تقویتی چندعاملی ساخته‌شده‌اند، که می‌توان به بازی‌های مارکوف [2] (**MG**) [۷] و فرآیندهای تصمیم‌گیری مارکوف با تعاملات پراکنده غیرمتمرکز [3] (**Dec-SIMDP**) [۸] اشاره کرد. فرض بازی‌های مارکوف این است که عامل‌ها در تمام فضای مشترک حالت-عمل[4]، کاملاً مشاهده‌پذیر باشند. درحالی‌که فرض الگوریتم‌های مبتنی بر فرآیند تصمیم‌گیری مارکوف با تعاملات پراکنده غیرمتمرکز، برپایه مشاهدات محلی عامل‌ها است [۹]. به‌دلیل اینکه عامل‌ها در فرآیند تصمیم‌گیری مارکوف با تعاملات پراکنده غیرمتمرکز، خارج از مناطق تعاملی قرار دارند، با فرآیند تصمیم‌گیری مارکوف تک‌عاملی مدل‌سازی می‌شوند، ولی درصورتی‌که عامل‌ها در داخل مناطق تعاملی قرار داشته باشند، از بازی مارکوف برای مدل‌سازی آن‌ها استفاده می‌شود [۱۰].

چالش‌های موجود در حوزه یادگیری تقویتی شامل موارد زیر می‌باشند: ۱) برخلاف پیشرفت سریع در تئوری‌ها و الگوریتم‌های یادگیری تقویتی چندعاملی، کاربردهای عملی این یادگیری با توجه به محدودیت‌های روش‌های موجود به تلاش‌های بسیاری نیاز دارد، ۲) الگوریتم‌های یادگیری تقویتی چندعاملی مبتنی بر تعادل، متکی بر فرآیندهای یادگیری با اتصالات محکم[5] هستند که مانع از کاربرد آن‌ها درعمل می‌شوند، ۳) به‌دست‌آوردن تعادل (به‌عنوان نمونه، تعادل نش) در هر گام زمانی، حتی برای محیط‌های نسبتاً کوچک با دو یا سه عامل، پیچیدگی محاسباتی زیادی دارد، ۴) از طرفی در الگوریتم‌های یادگیری تقویتی چندعاملی با تعاملات

[1] Multi-agent reinforcement learning

[2] Markov games
[3] Decentralized sparse interaction markov decision processes
[4] State - action
[5] Tightly coupled learning process

پراکنده[1]، عامل‌ها هیچ مفهومی از سیاست تعادلی ندارند و به‌صورت مهاجمانه به سمت هدف خود حرکت می‌کنند که به احتمال زیاد منجر به برخورد می‌شود [۱۱]. درنتیجه برای مواجهه با چالش‌های ذکرشده، در این مقاله تمرکز اصلی بر این است که چگونه سازوکار تعادل و انتقال دانش، در الگوریتم‌های مبتنی بر تعاملات پراکنده و الگوریتم‌های یادگیری تقویتی چندعاملی مبتنی بر مذاکرات و تعاملات پراکنده استفاده شود [۱۲].

الگوریتم پیشنهادی در این مقاله با عنوان RKT-MARL [2] شامل چهار مرحله اصلی زیر است: ۱) ایجاد یک چارچوب مبتنی بر مذاکره برای تعاملات پراکنده، ۲) مذاکره برای به‌دست‌آوردن مجموعه تعادل، ۳) روش کمینه واریانس[3] برای انتخاب یک عمل مشترک، و ۴) انتقال دانش مقادیر Q محلی[4] به‌صورت تنظیم‌شده. الگوریتم RKT-MARL ابتدا بر پایه مدل فرآیند تصمیم‌گیری مارکوف شروع‌شده و فرض اولیه بر این است که عامل‌ها سیاست بهینه تک‌عاملی را قبل از یادگیری در محیط چندعاملی به‌دست می‌آورند؛ سپس عامل‌ها برای به‌دست‌آوردن مجموعه تعادل در حالت هماهنگی، باهم مذاکره کرده و از روش کمینه واریانس برای انتخاب بهترین عمل مشترک استفاده‌می‌کنند. در مرحله بعد، عامل‌ها طبق این عمل مشترک انتخاب‌شده به حالت بعدی حرکت و پاداش فوری خود را دریافت‌می‌کنند. درنهایت، مقدار Q با توجه به سود تعادلی و پاداش عامل‌ها، به‌روز می‌شود؛ همچنین نکته قابل توجه این است که در مرحله چهارم که انتقال دانش صورت می‌گیرد، Q در حالت هماهنگی مشترک به صورت تنظیم‌شده، توسط مجموعی از اطلاعات محیطی جاری و دانش آموزش داده‌شده قبلی، مقداردهی اولیه می‌شود. این ضرایب تنظیم باعث می‌شوند که بتوان درصد اثرگذاری اطلاعات جاری و دانش قبلی را در مقداردهی اولیه Q کنترل کرد؛ بنابراین، مقدار بالاتر ضرایب نشان‌دهنده تاثیر بیشتر اطلاعات جاری و دانش قبلی‌است.

در سال‌های اخیر، از سازوکار انتقال دانش برای بهبود سرعت در مسائل یادگیری تقویتی چندعاملی استفاده‌شده‌است. انتقال دانش به عامل‌ها اجازه می‌دهد که ابتدا بر روی یک محیط منبع ساده، یادگیری انجام داده و سپس اطلاعات به‌دست‌آمده از آن را در صورت لزوم به محیط‌های پیچیده‌تری انتقال دهند، تا بتوانند با عامل‌های دیگر کار کرده و برای جلوگیری از برخورد هماهنگی برقرار کنند [۱۳، ۱۴]. به‌کارگیری سازوکار انتقال دانش در الگوریتم‌های یادگیری تقویتی، سرعت یادگیری عامل‌ها را بالا برده [۱۵] و عملکرد الگوریتم‌ها را بهبود می‌بخشد و عامل‌ها را در برقراری هماهنگی با یکدیگر توانمندتر می‌کند. نکته قابل‌توجه این است که انتقال چه نوع دانشی می‌تواند برای یادگیری سیاست‌های بهتر در سیستم‌های چندعاملی، سودمند باشد. در بعضی سیستم‌های چندعاملی، عامل‌ها دانش تک‌عاملی (به‌عنوان نمونه، تابع مقدار محلی[5] یا سیاست محلی) را قبل از فرآیند یادگیری چندعاملی می‌آموزند و استفاده از چنین دانشی در طول یادگیری ممکن است، مفید باشد.

به‌منظور ارزیابی کارایی الگوریتم پیشنهادی در این مقاله، چندین مجموعه داده برای نشان‌دادن عملکرد الگوریتم از لحاظ همگرایی، مقیاس‌پذیری و قابلیت هماهنگی مورد آزمایش قرارگرفته و نتایج آن‌ها، با دیگر الگوریتم‌های یادگیری تقویتی چندعاملی مقایسه شده‌است. نتایج حاصل، نشان‌دهنده برتری قابل ملاحظه الگوریتم پیشنهادی نسبت به الگوریتم‌های دیگر در این حوزه‌است.

ساختار ادامه مقاله به‌صورت زیراست. در بخش ۲ پژوهش‌های مرتبط مورد بررسی قرارمی‌گیرند. در بخش ۳ تعاریف پایه‌ای و روش پیشنهادی گنجانده‌شده‌است. بخش ۴ شامل مجموعه داده‌ها و الگوریتم‌های مورد ارزیابی و فرضیات آزمایش‌هااست. در بخش ۵ نتایج حاصل از عملکرد الگوریتم پیشنهادی و مقایسه آن با دیگر الگوریتم‌های موجود، قرار دارد؛ و درنهایت در بخش ۶ نتیجه‌گیری و کارهای پیشنهادی آینده ارائه‌شده‌است.

۲- کارهای پیشین

در حوزه یادگیری ماشین، روش‌های بسیاری در زمینه یادگیری تقویتی پیشنهادشده، که اکثر آن‌ها از الگوریتم یادگیری Q [6] [۱۶] برای برقراری هماهنگی بین عامل‌ها استفاده‌می‌کنند. به‌طورکلی، روش‌های پیشنهادشده در زمینه یادگیری تقویتی چندعاملی، به سه دسته تقسیم می‌شوند: روش‌های مبتنی بر مفهوم تعادل[7]، روش‌های مبتنی بر سازوکار انتقال دانش[8] و روش‌های مبتنی بر تعاملات پراکنده[9]. به‌طور خلاصه، انواع روش‌های یادگیری تقویتی چندعاملی در جدول (۱) نشان داده‌شده‌است.

[1] Sparse interaction
[2] Regularized knowledge transfer for multi-agent reinforcement learning
[3] Minimum variance method
[4] Local Q-value
[5] Local value function
[6] Q-Learning
[7] Equilibrium-based methods
[8] Knowledge transfer-based methods
[9] Sparse-interaction-based methods

(جدول - ۱).: انواع روش‌های یادگیری تقویتی چندعاملی

(Table-1).Types of multi-agent reinforcement learning methods

	اساس کار هر روش	نمونه الگوریتم	ویژگی هر الگوریتم
۱) روش‌های مبتنی بر تعادل	• استفاده از مدل بازی مارکوف به عنوان چارچوب یادگیری تقویتی • استفاده از مفاهیم تعادلی به عنوان سیاست بهینه	الگوریتم NegoQ	• استفاده از سه نوع استراتژی پروفایل غیراحتمالی به جای استراتژی پروفایل احتمالی، به عنوان مفهوم تعادل • استفاده از یک فرآیند مذاکره چندمرحله‌ای برای یافتن سه استراتژی پروفایل
۲) روش‌های مبتنی بر انتقال دانش	• بهره گرفتن از دانش تک‌عاملی از قبل آموخته‌شده در طول یادگیری چندعاملی • انتقال دانش و تجربیات مورد نیاز به عامل از یک محیط ساده به یک محیط پیچیده	الگوریتم NegoQVFT	• استفاده از سه سازوکار انتقال دانش (۱) انتقال تابع مقدار، ۲) انتقال تابع مقدار انتخابی، و ۳) انتزاع بازی مبتنی بر انتقال مدل برای یادگیری سیاست‌های بهتر
۳) روش‌های مبتنی بر تعاملات پراکنده	• استفاده از تعاملات پراکنده در سیستم‌های چندعاملی به منظور بهبود عملکرد یادگیری تقویتی	الگوریتم NegoQVFT	• بهره بردن از سه نوع سازوکار انتقال دانش • استفاده از تعاملات پراکنده

ازجمله کارهای قبلی در حوزه یادگیری تقویتی می‌توان به الگوریتم یادگیری CQ [۱] [۱۷] اشاره کرد که در آن عامل‌ها می‌توانند به‌طور هم‌زمان از رشد نمایی فضای حالت جلوگیری کرده و همچنین سیاست‌های کارآمدی را در محیط‌های چندعاملی بیاموزند. ضعف این الگوریتم این است که جزو روش‌های مبتنی بر تعادل نیست و در آن عامل‌ها در یک حالت مشترک خاص به جای استراتژی‌های تعادلی، به‌صورت حریصانه عمل می‌کنند که درنهایت ممکن است، منجر به برخورد شوند.

همان‌طورکه در جدول (۱) نشان داده‌شده، روش‌های مبتنی بر تعادل NegoQ [۲][۱۸]، یکی از مهم‌ترین روش‌های یادگیری تقویتی چندعاملی هستند که مفاهیم تعادلی را در تئوری بازی‌ها در نظر می‌گیرند، به‌طوری‌که عامل‌ها در هر حالت، استراتژی‌های تعادلی را انتخاب‌می‌کنند. این روش‌ها از مدل بازی مارکوف، به‌عنوان چارچوب یادگیری تقویتی و از مفاهیم تعادلی مختلف برای یادگیری سیاست بهینه استفاده‌می‌کنند. در روش‌های مبتنی بر تعادل، مفهوم تعادل هم به یافتن سیاست بهینه توسط عامل‌ها و هم به جلوگیری از برخورد بین عامل‌ها کمک می‌کند؛ بااین‌حال، در بیشتر الگوریتم‌های یادگیری تقویتی چندعاملی مبتنی بر تعادل، عامل‌ها باید توابع مقدار عامل‌های دیگر را رونوشت کرده و از استراتژی پروفایل احتمالی[۳] استفاده کنند. با توجه به اینکه تابع مقدار، جزو اطلاعات خصوصی یک عامل است، به اشتراک‌گذاری این تابع غیرمنطقی بوده و همچنین، به‌دلیل بالابودن پیچیدگی محاسباتی استراتژی احتمالی، این الگوریتم‌ها کاربرد گسترده‌ای درعمل ندارند. الگوریتم NegoQ ازجمله روش‌های مبتنی بر تعادل است، که برای حل دو محدودیت بالا پیشنهادشده و دو ایده کلی دارد: ۱) به جای استراتژی پروفایل احتمالی، از سه نوع استراتژی پروفایل غیر احتمالی[۴] به‌عنوان مفهوم تعادل استفاده می‌کند، و ۲) به‌دلیل اینکه عامل‌ها نمی‌توانند تابع مقدار عامل‌های دیگر را رونوشت کنند، یک فرایند مذاکره چندمرحله‌ای برای یافتن سه استراتژی پروفایل غیر احتمالی استفاده می‌کند.

در روش‌های مبتنی بر انتقال دانش NegoSI [۵] [۱۲، ۱۵، ۱۹] و NegoQ with VFT [۶] [۲۰]، عامل‌ها دانش تک‌عاملی (به‌عنوان نمونه، تابع مقدار محلی یا سیاست محلی) را قبل از فرآیند یادگیری چندعاملی می‌آموزند و استفاده از چنین دانشی در طول یادگیری، می‌تواند مفید باشد. در این روش‌ها می‌توان انواع دانش و تجربیات موردنیاز، مثل تابع مقدار، تابع مقدار محلی، سیاست [۱۹] و تعادل را به عامل‌ها انتقال داد که ایده کلیدی انتقال تعادل [۱۵]، استفاده مجدد از تعادل‌هایی است که درقبل محاسبه‌شده‌اند. محاسبه تعادل و استفاده مجدد از آن برای انتخاب عمل و به‌روزرسانی توابع مقدار، از مهم‌ترین مراحل روش‌های مبتنی بر تعادل نیز محسوب

[1] CQ-Learning
[2] Negotiation-based Q-learning
[3] Mixed strategy profile
[4] Pure strategy profile
[5] Negotiation-based multi-agent reinforcement learning with sparse interaction
[6] Negotiation-based Q-learning with value function transfer

می‌شود. به‌طور کلی، مفهوم انتقال دانش، بدین معنی است که عامل پس از حل یک مسئله یادگیری تقویتی ساده، دانش آن را به یک مسئله پیچیده‌تر منتقل می‌کند [۲۱]. الگوریتم NegoQ with VFT، به جای قوانین یادگیری Q، از تئوری بازی‌ها به‌عنوان رویکرد پایه یادگیری استفاده می‌کند. این الگوریتم، جزو روش‌های مبتنی بر انتقال دانش و روش‌های مبتنی بر تعاملات پراکنده است که از سه سازوکار انتقال دانش (انتقال تابع مقدار، انتقال تابع مقدار انتخابی و انتزاع بازی مبتنی بر انتقال مدل) برای یادگیری سیاست‌های بهتر در سیستم‌های چندعاملی با تعاملات پراکنده استفاده می‌کند.

نوع سوم از روش‌های یادگیری تقویتی چندعاملی، روش‌های مبتنی بر تعاملات پراکنده [۱۱، ۲۰، ۲۲ و ۲۳] هستند. در این روش‌ها، به دلیل اینکه امکان دارد در همه حالت‌ها، عامل‌ها با یکدیگر تعامل نداشته باشند و همیشه این تعاملات، شامل همه عامل‌ها نباشد، تعاملات بین عامل‌ها پراکنده‌است. استفاده از تعاملات پراکنده در سیستم‌های چندعاملی، می‌تواند عملکرد یادگیری تقویتی را بهبود بیشتری ببخشد. در الگوریتم NegoQ with VFT، به دلیل پراکنده بودن تعاملات بین عامل‌ها، هر عامل می‌تواند با اندک دانشی که در اختیار دارد، خود را با کمترین تغییرات محیط تطبیق دهد.

در این مقاله، برای تمرکز بر حل مسائل یادگیری در سیستم‌های پیچیده، الگوریتم RKT-MARL به‌صورت ترکیبی از روش‌های مبتنی بر تعادل، روش‌های مبتنی بر تعاملات پراکنده و روش‌های مبتنی بر انتقال دانش پیشنهاد می‌شود. با توجه به اینکه الگوریتم‌های مبتنی بر تعادل با اتصالات محکم، در سیستم‌های پیچیده غیرعملی بوده و همچنین، الگوریتم‌های مبتنی بر تعاملات پراکنده، موجب برخوردهای زیادی می‌شوند،در این مقاله یک روش بهینه برای حل این مشکلات پیشنهادشده‌است. بدین ترتیب در الگوریتم پیشنهادی، یک چارچوب یادگیری مبتنی بر تعاملات پراکنده در نظر گرفته‌شده، به‌طوری‌که هر عامل، عمل مشترک تعادلی را در مناطق هماهنگی انتخاب می‌کند. بنابراین در مقایسه با الگوریتم‌های مبتنی بر تعاملات پراکنده، در این الگوریتم، عامل‌ها هنگام انتخاب عمل مشترک، تعادل را نیز در نظرمی‌گیرند. همچنین برخلاف الگوریتم‌های مبتنی بر انتقال دانش قبلی، در این الگوریتم برای مقداردهی اولیه Q از ضریب تنظیم‌پذیری بین اطلاعات آموزش داده‌شده قبلی و اطلاعات محیطی جاری استفاده‌شده که بین دو دانش، تنظیم ایجاد می‌کند و همچنین در محیط‌های پیچیده‌تر برای بالابردن سرعت یادگیری، دانش مناسب‌تری به عامل‌ها انتقال داده می‌شود.

۳- تعریف مسأله

در این بخش، مفاهیم و تعاریف اصلی در یادگیری تقویتی چندعاملی ارائه‌شده و در ادامه تعریف مسئله به‌طور کامل شرح داده می‌شود.

۳- ۱- مدل فرآیند تصمیم‌گیری مارکوف [1] (MDP)

این مدل یکی از مدل‌های استاندارد تصمیم‌گیری در یادگیری تقویتی‌است. فرآیند تصمیم‌گیری مارکوف، یک مسئله تصمیم‌گیری متوالی را مطرح می‌کند که در آن عامل در هر گام زمانی، عملی را برای افزایش پاداش انتخاب می‌کند. MDP، یک چندتایی $< S.A.R.T >$ است که S فضای حالت‌ها و A فضای عمل‌های یک عامل، $R: S \times A \to \mathbb{R}$ تابع نگاشت پاداش به زوج حالت-عمل و $T: S \times A \times S \to [0.1]$ تابع انتقال را مشخص می‌کند. هدف مدل MDP این است که عامل‌ها بتوانند سیاست بهینه‌ای را پیدا کنند تا بعضی معیارهای بهینه‌سازی مبتنی بر پاداش را ماکزیمم کنند. ازجمله این معیارها، مجموع پاداش‌های مورد انتظار به صورت رابطه (۱) است [۸]:

$$V^*(s) = \max_\pi E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r^{t+k} \mid s^t = s \right\} \quad (۱)$$

که $V^*(s)$ مقدار حاصل در حالت s تحت سیاست بهینه را نشان می‌دهد. $\pi: S \times A \to [0.1]$ سیاست هر عامل، t هر گام زمانی، k گام زمانی بعدی، r^{t+k} پاداش حاصل در گام زمانی $(t+k)$ و $\gamma \in [0.1]$ پارامتر تنزیل[2] را مشخص‌می‌کنند.

هدف MDP را می‌توان با مقادیر Q برای هر زوج حالت-عمل به‌صورت رابطه (۲) بیان کرد [۸]:

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s'} T(s,a,s') max_{a'} Q^*(s',a') \quad (۲)$$

که $Q^*(s.a)$ مقدار بهینه در زوج حالت-عمل$(s.a)$ تحت سیاست بهینه، $\acute{s}$ حالت بعدی، $\acute{a}$ عمل بعدی، $r(s.a)$ پاداش فوری عامل با انتخاب عمل a در حالت s و $T(s.a.\acute{s})$ احتمال انتقال از حالت s به حالت $\acute{s}$ با انجام عمل a را نشان می‌دهند.

از الگوریتم یادگیری Q می‌توان برای تخمین $Q^*(s.a)$ استفاده کرد که قانون به‌روزرسانی یک مرحله‌ای آن به‌صورت رابطه (۳) است [۸]:

$$Q(s,a) \leftarrow (1-\alpha) Q(s,a) + \alpha \left[r(s,a) + \gamma max_{a'} Q(s',a') \right] \quad (۳)$$

[1] Markov decision process
[2] Discount factor

که $Q(s.a)$ مقدار زوج حالت-عمل $(s.a)$ بوده و $\alpha \in [0.1]$ پارامتری به نام نرخ یادگیری[1] را نشان می‌دهد.

۳- ۲- مدل بازی مارکوف (MG)

مدل بازی مارکوف به‌عنوان MDP چندگانه محسوب می‌شود که در آن احتمال انتقال و پاداش به زوج حالت عمل مشترک همه عامل‌ها بستگی دارد. در یک حالت مشخص، مجموعه عمل‌های فردی عامل‌ها، یک بازی تکرارشونده را ایجاد می‌کند که با روش نظریه بازی‌ها حل می‌شود؛ بنابراین MG یک چارچوب قوی‌تری به شمار می‌آید که می‌تواند هم MDP و هم بازی تکرارشونده را عمومیت بخشد.

بازی مارکوف n عاملی، یک چندتایی $\langle n, \{S_i\}_{i=1,\ldots,n}, \{A_i\}_{i=1,\ldots,n}, \{R_i\}_{i=1,\ldots,n}, T\rangle$ است که در آن n تعداد عامل‌ها، $\boldsymbol{S_i}$ مجموعه فضای حالت‌های عامل i ام، $S=\{S_i\}_{i=1,\ldots,n}$ مجموعه فضای حالت‌های تمام عامل‌ها، $\boldsymbol{A_i}$ مجموعه فضای عمل‌های عامل i ام، $A=\{A_i\}_{i=1,\ldots,n}$ مجموعه فضای عمل‌های تمام عامل‌ها و $\pi=(\pi_1,\ldots,\pi_n)$ سیاست مشترک تمام عامل‌ها را مشخص‌می‌کنند. مقادیر Q زوج حالت-عمل مشترک برای عامل i تحت سیاست مشترک $\boldsymbol{\pi}$ به‌صورت رابطه (۴) تعریف می‌شود [۱۱]:

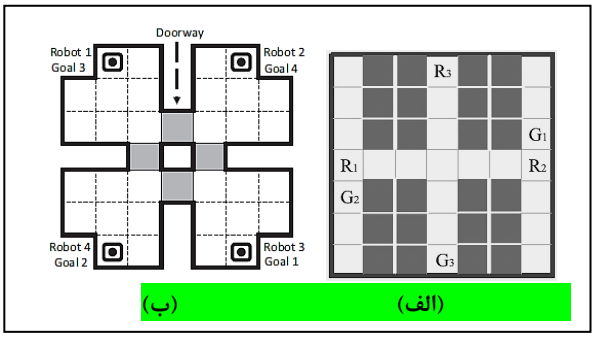
$$Q_i^{\pi}(\vec{s},\vec{a}) = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^k r_i^{t+k} \mid \vec{s}^t=\vec{s}, \vec{a}^t=\vec{a}\right\} \qquad (۴)$$

که، $\vec{s} \in S$ حالت مشترک، $\vec{a} \in A$ عمل مشترک و r_i^{t+k} پاداش دریافتی در گام زمانی $(t+k)$است. برخلاف هدف بهینه‌سازی MDP، هدف مدل MG پیداکردن سیاست مشترک تعادلی π به جای سیاست مشترک بهینه برای همه عامل‌هااست. بنابراین مقدار Q در زوج حالت-عمل مشترک بر اساس تعادل به‌صورت رابطه (۵) به‌روز می‌شود که در آن $\emptyset_i(\vec{s}')$ مقدار مورد انتظار تعادل در حالت مشترک بعدی برای عامل i است [۱۱]:

$$Q_i(\vec{s},\vec{a}) \leftarrow (1-\alpha)Q_i(\vec{s},\vec{a}) + \alpha\left(r_i(\vec{s},\vec{a}) + \gamma \emptyset_i(\vec{s}')\right) \qquad (۵)$$

۳- ۳- تعاملات پراکنده

بر اساس تعریفی که در مدل MG ذکر شد، ابتدا همه عامل‌ها باید سیاست خود را در تمام فضای حالت-عمل مشترک بیاموزند و سپس بتوانند با یکدیگر هماهنگی برقرار کنند؛ اما این فرضیه همیشه درعمل برقرار نیست و حقیقت این است که در بیش‌تر سیستم‌های چندعاملی، عامل‌های یادگیرنده با توجه به تعاملات محدود در بعضی مناطق، هماهنگی ضعیف‌تری با یکدیگر دارند، زیرا ممکن است، تعاملات بین عامل‌ها در تمام حالت‌ها رخ ندهد و یا همیشه شامل همه عامل‌ها نباشد. به‌عنوان مثالی از سیستم‌های چندعاملی با تعاملات پراکنده، سیستم انباره هوشمند[2] در شکل (۱) (الف) در نظر گرفته می‌شود که در آن، ربات‌های خودمختار تنها زمانی به ربات‌های دیگر توجه می‌کنند که به‌اندازه کافی به یکدیگر نزدیک باشند، در غیر این صورت به‌طور مستقل عمل‌می‌کنند.



(شکل-۱): مثال هایی از سیستم‌های چندعاملی با تعاملات پراکنده. الف) بازی انباره هوشمند که در آن نماد R و G به ترتیب نشان‌دهنده ربات‌های خودمختار و هدف هر یک از عامل‌هااست. ب) بازی چندعاملی **مبتنی بر تعاملات پراکنده** [۲۰].

(Figure-1).: Examples of multi-agent systems with sparse interactions. a) Intelligent warehouse game, in which the R and G symbols represent the autonomous robots and the goal of each agent, respectively. b) Multi-agent game based on the sparse interactions [20].

شکل (۱) (ب) را نیز می‌توان به‌عنوان مثالی دیگر از تعاملات پراکنده در نظر گرفت که در آن هر چهار ربات باید راهی برای رسیدن به اهداف خود پیدا کنند. اکثر اوقات، ربات‌ها می‌توانند آزادانه حرکت کنند ولی در دروازه باریک، هر زمان فقط یک ربات می‌تواند عبور کند و این بدین معنی است که ربات‌ها باید در اطراف دروازه هماهنگی برقرار کنند.

۴- روش پیشنهادی

در این بخش، الگوریتم RKT-MARL برای حل مسائل یادگیری در سیستم‌های پیچیده، با جزئیات بیشتر توضیح داده می‌شود. هدف روش پیشنهادی در این مقاله ارائه الگوریتمی است که با بهره‌گیری از راه‌حل‌های تعادلی، از

[1] Learning rate

[2] Intelligent warehouse system

برخوردهای احتمالی بین عامل‌ها در سیستم‌های چندعاملی مبتنی بر تعاملات پراکنده جلوگیری کند. در همین راستا، در این الگوریتم، عامل‌ها در زمان انتخاب عمل مشترک، برای به‌دست‌آوردن مجموعه راه‌حل‌های تعادلی، از استراتژی پروفایل تعادلی غالب غیرقطعی[1] (Non-strict EDSP) یا تعادل متا [2] استفاده‌می‌کنند.

در الگوریتم RKT-MARL، ابتدا عامل‌ها در یک محیط ساده و بدون توجه به حضور عامل‌های دیگر، سیاست بهینه تک‌عاملی خود را به کمک مدل فرآیند تصمیم‌گیری مارکوف به‌دست می‌آورند. سپس زمانی که در حالت هماهنگی قرار بگیرند، برای به‌دست‌آوردن مجموعه تعادل با عامل‌های دیگر مذاکره می‌کنند و برای انتخاب بهترین عمل مشترک در مجموعه تعادل حاصل، از روش کمینه واریانس استفاده‌می‌کنند. پس از آن، عامل‌ها بعد از انتخاب بهترین عمل مشترک به حالت بعدی حرکت کرده و پاداش فوری خود را دریافت‌می‌کنند. درنهایت، مقدار Q با توجه به سود تعادلی و پاداش عامل‌ها به‌روز می‌شود.

۴- ۱- چارچوب کلی

زمانی که افراد در محیط‌هایی با برخوردهای احتمالی کار می‌کنند، ابتدا یاد می‌گیرند که چگونه وظایف فردی خود را به پایان برسانند؛ سپس در این محیط چگونه با دیگران هماهنگ شوند. با الهام از این فرضیه رایج، در الگوریتم RKT-MARL فرآیند یادگیری را در دو زیر فرآیند به‌صورت زیر تجزیه می‌کنیم: ۱) هر عامل بدون توجه به حضور عامل‌های دیگر، سیاست بهینه تک‌عاملی خود را در یک محیط ایستا یاد می‌گیرد، و ۲) در مواقعی که حضور عامل‌ها برهم تأثیرگذار باشد و برخوردی بین آن‌ها رخ دهد، پاداش فوری[3] دریافتی توسط عامل‌ها تغییر می‌یابد؛ بنابراین در این شرایط لازم است که هر عامل بتواند بر اساس تغییرات پاداش فوری با عامل‌های دیگر هماهنگی برقرار کند. همچنین وجود سازوکار انتقال دانش در این الگوریتم، به یادگیری عامل‌ها سرعت بخشیده و آن‌ها را در برقراری هماهنگی با یکدیگر توانمندتر می‌کند. در ادامه، چهار مرحله اصلی الگوریتم RKT-MARL به‌طور کامل شرح داده می‌شود.

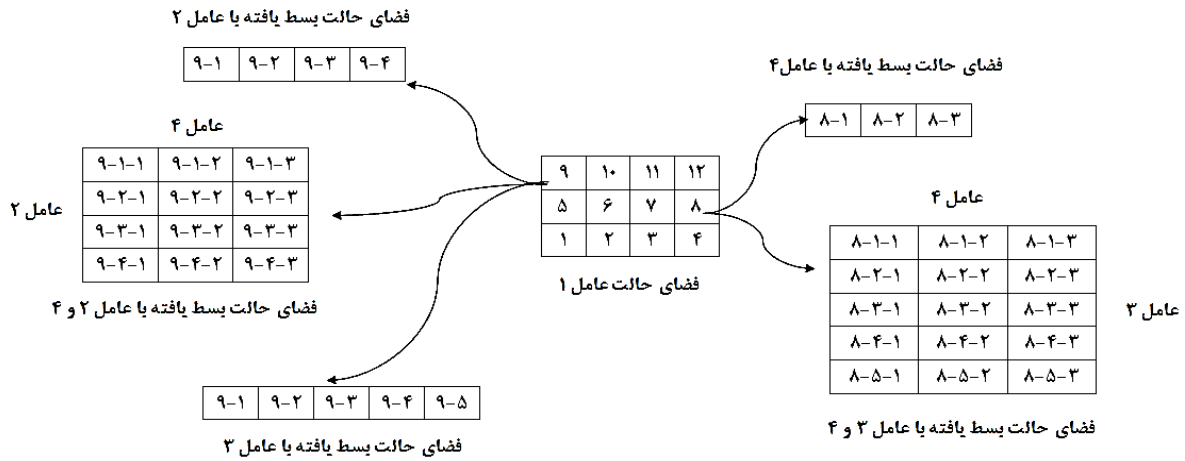
۴-۱- ۱- ایجاد یک چارچوب مبتنی بر مذاکره برای تعاملات پراکنده

همان‌طور که درقبل نیز ذکرشده، فرض اولیه بر این است که عامل‌ها سیاست بهینه تک‌عاملی و مدل پاداش خود را قبل از یادگیری در محیط چندعاملی به دست می‌آورند؛ ولی زمانی که عامل‌ها در محیط چندعاملی قرار می‌گیرند، دو حالت ممکن است، رخ دهد: اگر در یک زوج حالت-عمل، پاداش فوری دریافتی با مقدار مورد انتظار از مدل پاداش یکسان باشند، عامل به‌صورت مستقل عمل می‌کند. در غیر این صورت، به‌منظور هماهنگی بهتر، عامل‌ها باید زوج حالت-عمل فردی خود را با افزودن زوج‌های حالت-عمل عامل‌های دیگر به نوع مشترک بسط دهند. مراحل ایجاد چارچوب مبتنی بر مذاکره برای تعاملات پراکنده به‌صورت زیر است:

منتشرکردن حالت مشترک: اگر عامل‌ها در یک حالت خاص با انتخاب یک عمل، تغییری در پاداش فوری تشخیص دهند، این زوج حالت-عمل به‌عنوان "خطرناک" و این عامل‌ها به‌عنوان "عامل‌های هماهنگ‌کننده[4]" علامت‌گذاری می‌شوند؛ سپس این عامل‌ها اطلاعات حالت- عمل خود را به همه عامل‌های دیگر منتشر کرده و اطلاعات آن‌ها را دریافت‌می‌کنند. این زوج حالت- عمل به همراه تغییرات پاداش، یک زوج مشترک به نام "زوج هماهنگی[5]" تشکیل می‌دهند و حالتی که در آن هماهنگی نیاز است به‌عنوان "حالت هماهنگی[6]" علامت‌گذاری می‌شود که در فضای حالت همه عامل‌های هماهنگ‌کننده وجود دارد.

انجام مذاکره برای به‌دست‌آوردن سیاست تعادلی: عامل‌ها هنگام انتخاب یک زوج حالت-عمل خطرناک، با انتشار اطلاعات خود به یکدیگر، مشخص می‌کنند که آیا در زوج هماهنگی قرار دارند یا نه. اگر چنین باشد، برای جلوگیری از برخورد، عامل‌ها باید به جای سیاست حریصانه به دنبال پیداکردن سیاست تعادلی باشند. بدین ترتیب، عامل‌ها می‌توانند از سازوکار مذاکره برای پیدا کردن سیاست تعادلی استفاده کنند. در چنین شرایطی اگر تعداد عامل‌ها بیش از دو باشد، ممکن است چندین بسط متفاوت برای حالت‌های هماهنگی مختلف در یک حالت وجود داشته باشد. همان‌طور که در شکل (۲) مشاهده می‌شود، عامل‌های ۱ تا ۴ به ترتیب دارای ۱۲، ۴، ۵ و ۳ حالت هستند. به‌عنوان مثالی از بسط حالت مشترک، عامل ۱ تمام حالت‌های عامل‌های دیگر را حتی اگر به بخش کوچکی از آن‌ها نیاز داشته باشد، به حالت مشترک خود اضافه می‌کند. حالت هشتم از عامل ۱ به دو حالت مشترک، یکی با عامل ۴ و دیگری با عامل ۳ و ۴، بسط داده می‌شود. به‌طور مشابه، حالت نهم به سه حالت مشترک بسط داده‌شده‌است.

[1] Non-strict equilibrium dominating strategy profile
[2] Meta equilibrium
[3] Immediate reward
[4] Coordinating agents
[5] Coordination pair
[6] Coordination state



(شکل-۲).: مثالی از بسط فضای حالت عامل‌ها، زمانی‌که در محل برخورد باهم مذاکره می‌کنند . حالت هشتم از عامل ۱ به دو حالت مشترک، یکی با عامل ۴ و دیگری با عامل ۳ و ۴، بسط داده‌می‌شود. حالت نهم نیز به سه حالت مشترک با عامل ۲، عامل ۳ و عامل‌های ۲ و ۴ بسط داده می‌شود.

(Figure-2). Example of the expansion of an agent's state space, when they negotiate at the collision place. The 8th state of agent 1 is expanded to two joint states, the first one is with agent 4 and the second one is with agents 3 and 4. Similarly, the 9th state of agent 1 is expanded to three joint states, the first one is with agent 2, the second one is with agents 2 and 4, and the third one is with

۴-۱-۲- مذاکره برای به‌دست‌آوردن مجموعه تعادل

هدف اصلی الگوریتم RKT-MARL، این است که عامل‌ها بتوانند راه‌حل‌های تعادلی را در حالت‌هایی که هماهنگی موردنیاز است، پیدا کنند. بدین ترتیب، در این بخش دو استراتژی پروفایل غیر احتمالی، non-strict EDSP و تعادل متا به‌عنوان مفاهیم تعادلی به‌طور کامل شرح داده می‌شوند.

۱) non-strict EDSP : در یک بازی n عاملی ($n \geq ۲$) به شکل نرمال، اگر $\vec{e}_i \in A\ (i = 1.2.\dots.m)$ مجموعه تعادل نش‌های بازی باشد، عمل مشترک $\vec{a} \in A$ به شرط زیر (رابطه (۶)) یک non-strict EDSP است:

$$\forall j \leq n \ , \ U_j(\vec{a}) \geq min_i U_j(\vec{e}_i) \quad (۶)$$

که $U_j(\vec{a})$ و $U_j(\vec{e}_i)$ به ترتیب مقادیر سودهایی هستند که عامل j در عمل مشترک $\vec{a}$ و تعادل نش $\vec{e}_i$ به دست می‌آورد. در این رابطه، هدف این است که سود عمل مشترک $\vec{a}$ (سمت چپ نامساوی) از کمترین سود مجموعه تعادل نش‌ها، بیشتر باشد.

فرآیند مذاکره برای به‌دست‌آوردن مجموعه non-strict EDSP، شامل دو گام زیر است: ۱) هرکدام از عامل‌ها با توجه به سود فردی خود، مجموعه استراتژی پروفایل‌هایی که non-strict EDSP هستند را پیدا می‌کنند، و ۲) سپس اشتراک مجموعه‌های به‌دست‌آمده در گام قبلی را پیدا کرده و یک مجموعه non-strict EDSP را انتخاب‌می‌کنند.

با توجه به اینکه در بعضی موارد به‌دلیل عدم وجود تعادل نش، مجموعه non-strict EDSP تهی است، تعادل متا به‌عنوان مفهوم تعادل محاسبه می‌شود که همیشه غیر تهی است.

۲) شرایط لازم و کافی برای تعادل متا: در یک بازی n عاملی به شکل نرمال، عمل مشترک $\vec{a}$ در بازی متا $k_1 k_2 \dots k_r \Gamma$، به شرطی یک تعادل متا به شمار می‌آید که اگر و تنها اگر رابطه زیر برقرار باشد:

$$U_i(\vec{a}) \geq min_{\vec{a}_{p_i}} max_{a_i} min_{\vec{a}_{s_i}} U_i(\vec{a}_{p_i}, a_i, \vec{a}_{s_i}) \quad (۷)$$

که p_i مجموعه عامل‌هایی است که در پیشوند $k_1 k_2 \dots k_r$ قبل از علامت i و s_i عامل‌هایی که بعد از علامت i قرار دارند.

برای مثال در یک بازی متا سه عاملی $213\ \Gamma$، تعادل متای $\vec{a}$ باید شرایط رابطه (۸) را داشته باشد:

$$\begin{aligned} U_1(\vec{a}) &\geq min_{a_2} max_{a_1} min_{a_3} U_1(a_1, a_2, a_3) \\ U_2(\vec{a}) &\geq max_{a_2} min_{a_1} min_{a_3} U_2(a_1, a_2, a_3) \\ U_3(\vec{a}) &\geq min_{a_2} min_{a_1} max_3 U_3(a_1, a_2, a_3) \end{aligned} \quad (۸)$$

۴-۱- ۳- روش کمینه واریانس برای انتخاب عمل مشترک

در مرحله قبل، فرآیندی برای همه عامل‌های هماهنگ‌کننده ارائه‌شده‌است تا برای به‌دست‌آوردن مجموعه عمل مشترک تعادلی با یکدیگر مذاکره کنند. بااین‌حال، مجموعه حاصل به‌طورمعمول شامل چندین استراتژی پروفایل بوده و عامل‌ها توانایی انتخاب بهترین استراتژی را ندارند. بنابراین در الگوریتم RKT-MARL، عامل‌های هماهنگ‌کننده برای انتخاب عمل مشترک با

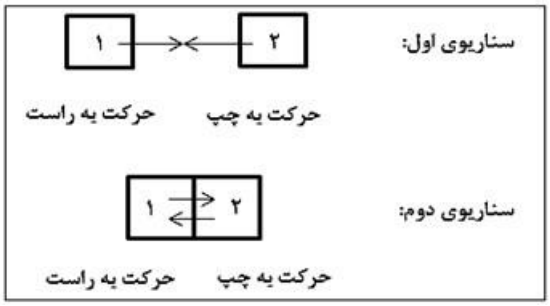
بالاترین مجموع سودمندی و پایین‌ترین واریانس سودمندی، از روش کمینه واریانس [۱۱] استفاده‌می‌کنند. این روش ویژگی همکاری بین عامل‌ها و عادلانه‌بودن فرآیند یادگیری را تضمین می‌کند. عامل‌ها بعد از استفاده از روش کمینه واریانس برای به‌دست‌آوردن یک راه‌حل تعادلی و انتخاب عمل مشترک، حالت‌های خود را به‌روز کرده و پاداش فوری را دریافت می‌کنند که برای به‌روز کردن مقادیر **Q** محلی استفاده می‌شود.

۴-۱- ۴- انتقال دانش مقادیر Q محلی

در ابتدای فرآیند یادگیری، الگوریتم RKT-MARL، دانشی درمورد سیاست بهینه تک‌عاملی به عامل‌ها برای سرعت‌بخشیدن به یادگیری انتقال داده می‌شود. علاوه بر این، با انتقال مقادیر Q محلی، عملکرد الگوریتم نیز بهبود می‌یابد. در الگوریتم یادگیری CQ [۱۷و ۲۴]، مقدار اولیه Q در حالت‌های مشترک بسط داده‌شده، صفر است، ولی در روش یادگیری انتقالی که توسط *ورانکس*[1] و همکارانش [۲۵] پیشنهادشده، Q با مقادیری که از قبل روی یک منبع آموزش داده‌شده، مقداردهی می‌شود. زمانی که افراد در مسیر رسیدن به مقصد با یکدیگر ملاقات می‌کنند، به‌طورمعمول دانش قبلی برای جلوگیری از برخورد دارند. بر اساس این فرضیه و داشتن اطلاعاتی درمورد به‌اتمام‌رساندن وظایف فردی، عامل‌ها یاد می‌گیرند که هنگام نزدیک‌شدن به همدیگر مذاکره کرده و استراتژی‌های مناسبی برای محیط‌های خاصی پیدا کنند. *ورانکس* تنها از دانش هماهنگی برای مقداردهی اولیه Q استفاده کرده و اطلاعات محیطی را نادیده گرفته است، ولی در الگوریتم RKT-MARL، برای مقداردهی اولیه Q، هم دانش هماهنگی که از قبل آموزش داده‌شده و هم اطلاعات محیطی در نظر گرفته‌شده‌است. بدین صورت که Q در حالت هماهنگی بسط داده‌شده ($Q_i^J\left(\left(s_i.s_{-i}^{\rightarrow}\right).\left(a_i.a_{-i}^{\rightarrow}\right)\right)$) توسط ضرایب تنظیم (a و b) بین دانش آموزش داده‌شده قبلی ($Q_i^{CT}(\vec{s}.\vec{a})$) و اطلاعات محیطی جاری ($Q_i(s_i.a_i)$)، به‌صورت زیر مقداردهی اولیه می‌شود:

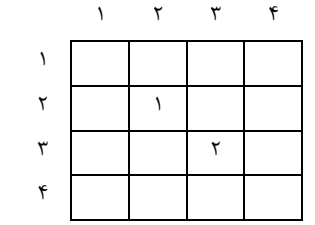
$$Q_i^J\left(\left(s_i,s_{-i}^{\rightarrow}\right),\left(a_i,a_{-i}^{\rightarrow}\right)\right)\leftarrow a*\left(Q_i\left(s_i,a_i\right)\right)+ b*\left(Q_i^{CT}\left(\vec{s},\vec{a}\right)\right) \quad (۹)$$

که $s_{-i}^{\rightarrow}$ مجموعه حالت‌های مشترک به‌جز حالت s_i، $a_{-i}^{\rightarrow}$ مجموعه‌عمل‌های مشترک به جز عمل a_i و $Q_i^{CT}(\vec{s}.\vec{a})$ مقدار Q انتقال‌یافته از یک کار منبع در حالت $\vec{s}$ یا همان دانش آموزش داده‌شده قبلی می‌باشند. همچنین در این رابطه، عبارت $Q_i^J\left(\left(s_i.s_{-i}^{\rightarrow}\right).\left(a_i.a_{-i}^{\rightarrow}\right)\right)$ با عبارت $Q_i^J(\vec{s}.\vec{a})$ یکسان‌است. مشابه روش *ورانکس*، حالت مشترک $\vec{s}$ در $Q_i^{CT}(\vec{s}.\vec{a})$ با موقعیت نسبی $(\Delta x.\Delta y)$ محاسبه می‌شود. همانطورکه در شکل (۳) مشاهده می‌شود، زمانی که عامل‌ها به یک موقعیت مکانی مشترک یا مکان همدیگر حرکت کنند، امتیاز ۱۰- را به‌عنوان جریمه دریافت می‌کنند و درغیراینصورت پاداش صفر می‌گیرند.

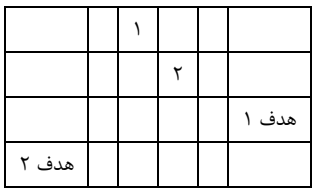


(شکل-۳). دو سناریو از برخوردهای احتمالی بین دو عامل. سناریوی اول: حرکت دو عامل به یک موقعیت مکانی مشترک. سناریوی دوم: حرکت عامل‌ها به مکان همدیگر.[11]

(Figure-3). Two scenarios of possible collisions between two agents. Scenario 1) moving two agents into the same grid location. Scenario 2) moving two agents into their previous locations [11].



(الف)



(ب)

(شکل-۴). انتقال مقادیر Q محلی در سیستم دو عاملی. الف) محیط ساده ۴×۴ با دو عامل، ب): یک محیط پیچیده.

(Figure-4). Example of the local Q-value transfer in a two-agent system. a) 4 × 4 simple environment with two agents. b) A complex environment.

[1] Vrancx

همچنین یادگیرنده‌های عمل مشترک در این محیط ساده، بدون توجه به اطلاعات محیطی، برقراری هماهنگی با عامل‌های دیگر را یادمی‌گیرند.

در این محیط ساده، مکان عامل اول $s_1 = (2.2)$ و مکان عامل دوم $s_2 = (3.3)$ است و دو عامل در خانه‌های (2.3) و (3.2) احتمال برخورد دارند؛ بنابراین $\vec{s} = (x_1 - x_2 . y_1 - y_2) = (1.1)$ یا $\vec{s} = (x_2 - x_1 . y_2 - y_1) = (-1.-1)$ بیانگر موقعیت نسبی عامل‌ها نسبت به موقعیت فعلی خودشان است. عمل مشترک دو عامل نیز به‌صورت رابطه (۱۰) تعریف می‌شود:

$$\vec{a} = ((a_1, a_2)) = \begin{pmatrix} (\uparrow,\uparrow) & (\uparrow,\downarrow) & (\uparrow,\leftarrow) & (\uparrow,\rightarrow) \\ (\downarrow,\uparrow) & (\downarrow,\downarrow) & (\downarrow,\leftarrow) & (\downarrow,\rightarrow) \\ (\leftarrow,\uparrow) & (\leftarrow,\downarrow) & (\leftarrow,\leftarrow) & (\leftarrow,\rightarrow) \\ (\rightarrow,\uparrow) & (\rightarrow,\downarrow) & (\rightarrow,\leftarrow) & (\rightarrow,\rightarrow) \end{pmatrix} \quad (۱۰)$$

در این ماتریس، $\{\uparrow.\downarrow.\leftarrow.\rightarrow\}$ مجموعه عمل‌های هر عامل است که به‌ترتیب نشان‌دهنده حرکت به بالا، پایین، چپ و راست هستند. بعد از چندین گام یادگیری، عامل‌ها ماتریس مقادیر Q خود را در حالت مشترک $\vec{s}$ طبق رابطه (۱۱) یاد می‌گیرند:

$$Q_1^{CT}(\vec{s},.) = Q_2^{CT}(\vec{s},.) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & 0 \\ -10 & 0 & 0 & 0 \end{pmatrix} \quad (۱۱)$$

در این ماتریس، جریمه ۱۰- برای هر عامل نشانگر این است که عامل‌های ۱ و ۲ با انتخاب عمل‌های $(\uparrow.\rightarrow)$ یا $(\leftarrow.\downarrow)$ منجر به برخورد می‌شوند و در بقیه موارد پاداش صفرمی‌گیرند. حال فرض کنید عامل‌ها در محیط پیچیده‌تری قرار داشته باشند (شکل (۴) (ب)). زمانی که عامل‌ها به‌صورت مستقل در محیط عمل می‌کنند، بردارهای مقادیر Q بهینه تک‌عاملی در حالت‌های s_1 و s_2 به‌صورت رابطه (۱۲) می‌باشند:

$$Q_1(s_1,.) = (-10,-1,-5,-1)$$
$$Q_2(s_2,.) = (-5,-1,-1,-5) \quad (۱۲)$$

در این بردارها، مقادیر ۵-، ۱۰- و ۱- به‌ترتیب نشان‌دهنده دورشدن از محیط، خروج از محیط شبکه‌شده و نزدیک‌شدن به هدف همراه با صرف انرژی هستند. این مقادیر به‌دست‌آمده همان اطلاعات محیطی است که عامل‌ها در محیط پیچیده به‌دست می‌آورند و سپس ماتریس مقادیر Q محلی در حالت هماهنگی برای هریک از عامل‌ها به صورت رابطه (۱۳) مقداردهی می‌شود:

$$Q_1^I(\vec{s},.) = Q_1(s_1,.)^T \times (1,1,1,1) + Q_1^{CT}(\vec{s},.) = \begin{pmatrix} -10 & -10 & -10 & -10 \\ -1 & -1 & -11 & -1 \\ -5 & -5 & -5 & -5 \\ -11 & -1 & -1 & -1 \end{pmatrix} \quad (۱۲)$$

$$Q_2^I(\vec{s},.) = (1,1,1,1)^T \times Q_2(s_2,.) + Q_2^{CT}(\vec{s},.) = \begin{pmatrix} -5 & -1 & -1 & -5 \\ -5 & -1 & -11 & -5 \\ -5 & -1 & -1 & -5 \\ -15 & -1 & -1 & -5 \end{pmatrix}$$

در این رابطه، $Q_1(s_1,.)$ و $Q_2(s_2,.)$ به‌ترتیب اطلاعات محیطی جاری برای عامل‌های ۱ و ۲ هستند که از رابطه (۱۲) به دست آمده اند. همچنین $Q_1^{CT}(\vec{s},.)$ و $Q_2^{CT}(\vec{s},.)$ بیان‌گر دانش انتقالی از محیط ساده‌تر می‌باشند که براساس رابطه (۱۱) مقداردهی شده‌اند. بدین ترتیب، مجموعه تعادل نش در این حالت مشترک {(چپ راست)، (پایین راست) و (پایین پایین)} است.

۵- تنظیمات اولیه محیط آزمایش

به منظور بررسی اهمیت الگوریتم ارائه شده، آزمایش‌های مختلفی برای مسائل یادگیری در سیستم‌های چندعاملی بر روی چندین نوع بازی جهانی انجام‌شده‌است. در ادامه جزئیات مجموعه داده‌های ارزیابی شده و الگوریتم‌های مورد مقایسه شرح داده می‌شود.

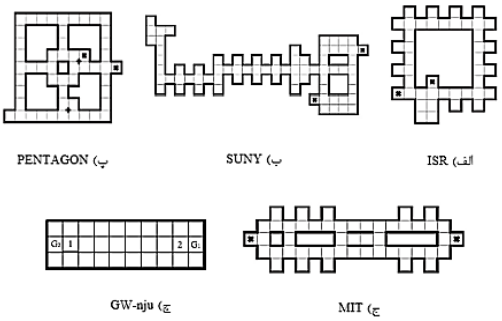
۵- ۱- معرفی مجموعه داده‌ها

کارایی الگوریتم پیشنهادشده در این مقاله بر روی پنج نوع بازی جهانی شبکه شده [1] [۲۴ و ۲۶] مختلف ارزیابی‌شده که در شکل (۵) نشان داده‌شده‌اند. همچنین این بازی‌ها در پژوهش‌های متعددی برای ارزیابی الگوریتم‌های مختلف [۲۰ و ۱۷، ۱۱] مورد استفاده قرار گرفته‌اند. برای چهار بازی ابتدایی (ISR، SUNY، PENTAGON و MIT) از جمله بازی‌های دو عاملی هستند که در هر کدام علامت ضربدر موقعیت اولیه هر عامل و هدف عامل دیگر را نشان می‌دهد. بازی پنجم (GW-nju) نیز شامل دو عامل است که موقعیت ابتدایی عامل i ام با i و هدف آن با G_i مشخص‌شده و از جمله بازی‌های با رقابت بالا به‌حساب می‌آید. در همه بازی‌ها، هر عامل در یک حالت می‌تواند در چهار جهت حرکت کرده و چهار عمل بالا، پایین، چپ و راست را انتخاب کند. همچنین در هر بازی عامل‌ها با طی تعدادی گام و جلوگیری از برخورد، سعی در رسیدن به هدف‌های خود دارند.

[1] Grid world games

۵- ۲- ارزیابی الگوریتم ها

الگوریتم‌هایی که RKT-MARL با آن‌ها مقایسه شده‌است، عبارتند از: یادگیری CQ [۱۷]، NegoQ with VFT [۲۰]، ILVFT و NegoSI[۱۱]. همه الگوریتم‌ها در محیط و شرایط آزمایشی یکسانی مورد بررسی قرار گرفته‌اند. عملکرد الگوریتم پیشنهادی با بهترین نتایج گزارش‌شده از الگوریتم‌های مورد مقایسه، ارزیابی‌شده و نتایج حاصل از اعمال الگوریتم‌ها بر روی پنج بازی جهانی، نشان‌دهنده برتری قابل‌ملاحظه الگوریتم RKT-MARL نسبت به سایر الگوریتم‌هااست. در این مقاله از دو معیار پاداش‌های هر اپیزود [۱] (REE) و تعداد گام‌های آخرین اپیزود [۲] (SFE) برای مقایسه این الگوریتم‌ها استفاده می‌شود. قوانین پاداش دریافتی هر عامل نیز به‌صورت زیر است:



(شکل-۵): بازی‌های جهانی شبکه‌شده.
(Figure-5). Grid world games.

۵- ۳- مفروضات پیاده‌سازی

روش پیشنهادی دارای پنج پارامتر شامل نرخ یادگیری α، پارامتر ε یا همان پارامتر اکتشافی که در انتخاب یک عمل به‌صورت رندوم به عنوان یک احتمال ثابت در نظر گرفته می‌شود، a و b (در رابطه ۹) پارامترهای تنظیم بین دانش آموزش داده‌شده قبلی و اطلاعات محیطی که با مقادیر مختلف مورد ارزیابی قرار می‌گیرند و پارامتر تنزیل γ است. مقادیر پارامترهای ε،γ و α در همه الگوریتم‌ها به‌ترتیب ۰/۰۱، ۰/۹و ۰/۱است. همچ[illegible]دیر بهینه دو پارامتر تنظیم b و a که در الگوریتم [illegible]RKT-MA برای تنظیم‌پذیری بین دانش آموزش داده[illegible]بلی و اطلاعات محیطی جاری استفاده می‌شوند، در [illegible]های مختلف در جدول (۲) آورده‌شده‌است. تمام الگو[illegible]ای مورد تست به تعداد دوهزار تکرار اجراشده‌اند. [illegible]وه، پیاده‌سازی الگوریتم پیشنهادی RKT-MARL [illegible]نرم‌افزار متلب انجام‌گرفته‌است.

(جدول-۲). مقادیر بهینه پارامترهای b وa برای هر پنج بازی جهانی شبکه‌بندی‌شده.

(Table-2). The optimal values of the parameters a and b for each five grid world games.

پارامترها	ISR	SUNY	MIT	PENTAGON	GW_nju
a	10	25	25	1	20
b	1	25	0.01	10	25

۶- نتایج و بحث ها

در این بخش، عملکرد الگوریتم RKT-MARL در مقایسه با الگوریتم‌های دیگر در حوزه یادگیری تقویتی چندعاملی مورد تحلیل و ارزیابی قرار می‌گیرد. نتایج به‌دست‌آمده برای یادگیری عامل‌ها در محیط‌های پیچیده‌تر روی بازی‌های معرفی‌شده نیز به‌صورت جداگانه ارزیابی می‌شود.

۶- ۱- ارزیابی نتایج

ابتدا عملکرد الگوریتم‌های مورد مقایسه با توجه به میانگین گام‌های یادگیری آخرین اپیزود (معیار SFE) بر روی هر پنج بازی ارزیابی می‌شود. معیار SFE برای هر الگوریتم بر روی هر پنج بازی در جدول (۳) نشان داده‌شده‌است. سطر آخر جدول (۳) مقادیر بهینه‌ای هستند که به‌وسیلهٔ سیاست بهینه به‌دست‌آمده‌اند. سازوکار انتقال تابع مقدار در الگوریتم‌های ILVFT و NegoQVFT به هم‌گرایی سریع در تمام بازی‌ها به جزء SUNY کمک می‌کند. سرعت همگرایی بالاتر نشان‌دهنده این است که عامل‌ها بهتر و سریع‌تر می‌توانند حالت‌های تعادل را یاد گرفته و مسئله یادگیری تقویتی را حل کرده و به هدف خود نزدیک شوند. در الگوریتم CQ-learning در بازی‌های PENTAGON و GW_nju، سیاست بهینه‌ای که عامل‌ها از قبل آموخته‌اند، همیشه دارای برخورد است و CQ-learning نمی‌تواند عمل مشترک تعادلی را برای برقراری هماهنگی بین عامل‌ها پیدا کند. زمانی که برخوردهای زیادی رخ می‌دهد، عامل‌ها به‌صورت متوالی به حالت‌های قبلی خود برمی‌گردند و مجبورند گام‌های زیادی قبل از رسیدن به هدف طی کنند [27,28]. در همه بازی‌ها الگوریتم RKT-MARL نزدیک‌ترین مقدار را به مقدار بهینه نشان می‌دهد؛ زیرا در این الگوریتم عامل‌ها هم دانش قبلی در مورد برخوردها دارند و هم با انتقال این دانش و انجام مذاکره در حالت‌های هماهنگی، از برخوردهای احتمالی جلوگیری می‌کنند. درنتیجه عامل‌ها

[1] Rewards of each episode

[2] Steps of final episode

برای رسیدن به هدف تعداد گام‌های کمتری را نسبت به زمانی که برخورد رخ دهد، طی‌می‌کنند.

(جدول-۳). میانگین گام‌های یادگیری آخرین اپیزود در هر بازی تست‌شده برای هر الگوریتم مورد مقایسه.

(Table-3). Average learning steps of the final episode in each tested game for each algorithm.

الگوریتم‌ها	ISR	SUNY	MIT	PENTAGON	GW_nju
CQ-learning	8.91	10.70	19.81	15.32	12.65
ILVFT	13.11	10.38	18.67	14.18	10.94
NegoQVFT	8.36	12.98	19.81	8.55	10.95
NegoSI	7.48	10.92	21.29	10.30	12.11
RKT-MARL	7.1	10	18.6	8.2	10.05
The optimal policy	6	10	18	8	10

سپس در شکل‌های (۶) تا (۱۰) ، پاداش‌های هر اپیزود (معیار REE) در اپیزودهای مختلف بر روی الگوریتم‌های مورد مقایسه بررسی‌شده و نتایج حاصل حاکی از آن است که مجموع پاداش به‌دست‌آمده برای عامل‌ها در هر بازی و برحسب الگوریتم مورد نظر متفاوت‌است.

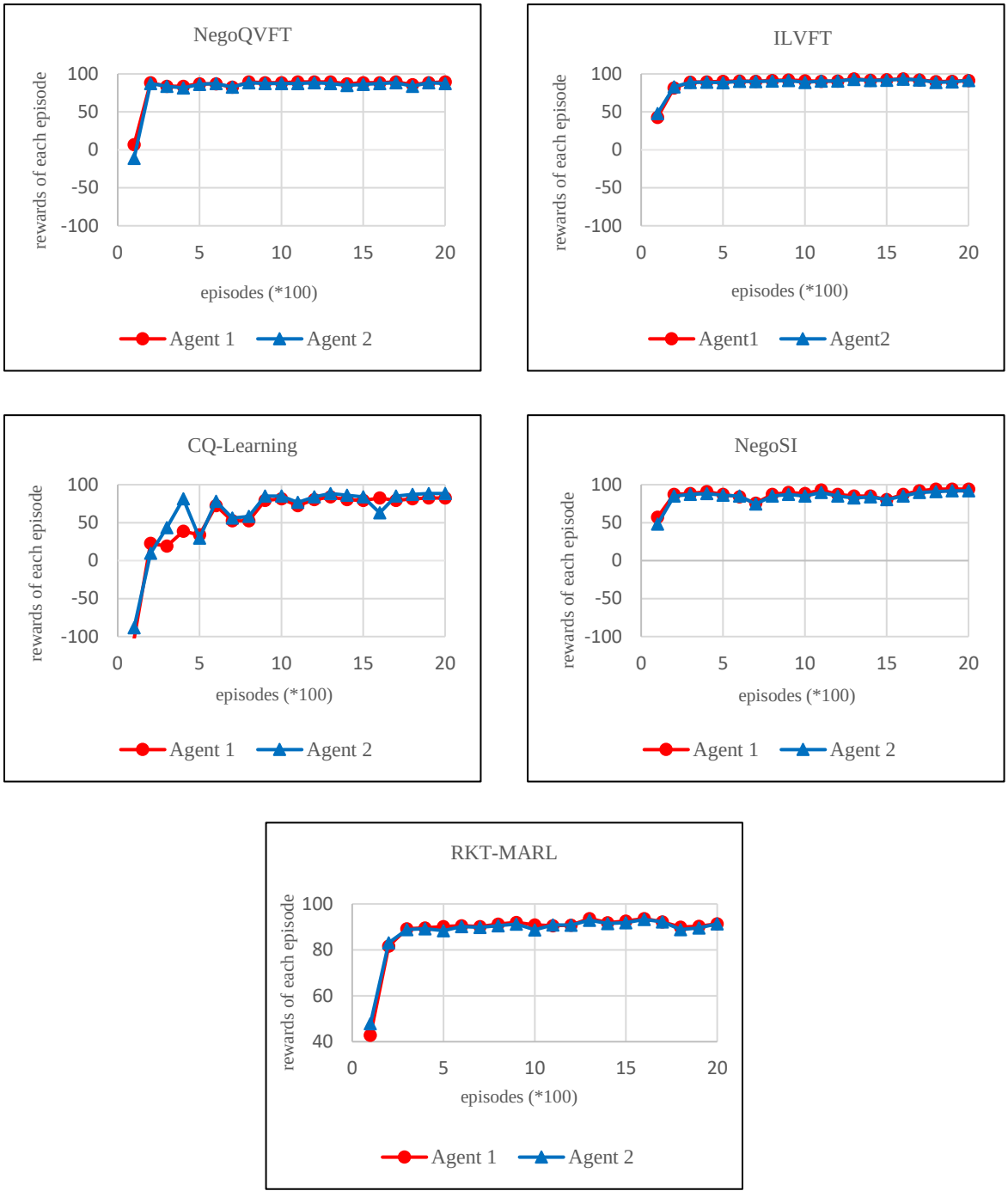
در بازی ISR شکل (۶) ، در هر دو الگوریتم NegoSI و RKT-MARL به دلیل انتقال مناسب دانش اولیه به عامل‌ها، شروع مجموع پاداش هر دو عامل بالاتر از چهل امتیاز است. در الگوریتم CQ-learning پاداش منفی بیان‌گر این است که عامل‌ها توانایی یافتن عمل مشترک تعادلی برای برقراری هماهنگی با یکدیگر را نداشته و بین آن‌ها تعداد زیادی برخورد رخ داده‌است. در اثر هر یک از این برخوردها، عامل‌ها یک گام به عقب حرکت کرده، تعداد گام‌های رسیدن به هدف افزایش و مجموع پاداش‌ها عددی منفی می‌شود. الگوریتم NegoSI در این بازی در تمام فرآیند یادگیری نسبت به الگوریتم‌های دیگر بالاترین پاداش را دارد که نشان‌دهنده برخوردهای کمتر و هماهنگی بیشتر بین عامل‌هااست. همچنین الگوریتم NegoSI نسبت به الگوریتم‌های دیگر عادلانه‌تر است؛ زیرا مقادیر پاداش دو عامل در نمودار بهم نزدیک‌تراست.

در بازی SUNY شکل (۷)، عامل‌ها مستقل از هم هستند و هرکدام سه انتخاب برای سیاست بهینه تک‌عاملی خود دارند. تفاوت بین مقادیر پاداش دو عامل در الگوریتم RKT-MARL کم است و این نشان‌دهنده عادلانه‌بودن فرآیند یادگیری این الگوریتم‌است. با اینکه در الگوریتم ILVFT یکی از عامل‌ها عملکرد بهتری دارد، ولی درکل نسبت به الگوریتم RKT-MARL عادلانه نیست. عامل‌ها در الگوریتم RKT-MARL دانش اولیه مناسب‌تری دریافت کرده و نسبت به سه الگوریتم NegoQVFT، CQ-learning و NegoSI بالاترین مقدار پاداش را دارند. در الگوریتم RKT-MARL از اپیزود ۲۰۰ام به بعد منحنی یادگیری آن نسبت به سایر الگوریتم‌ها حالت پایدارتری به خود می‌گیرد.

در بازی MIT شکل (۸) که عامل‌ها انتخاب بیشتری برای جلوگیری از برخورد دارند، همه الگوریتم‌های مورد تست، پاداش نهایی حدود ۸۰ امتیاز را کسب کرده‌اند. همان‌طورکه در شکل (۸) مشاهده می‌شود، دانش اولیه الگوریتم RKT-MARL و CQ-learning نسبت به بقیه روش‌ها بیشتر است، ولی مشکل الگوریتم CQ-learning این است که منحنی یادگیری آن به‌نسبه ناپایدار هست.

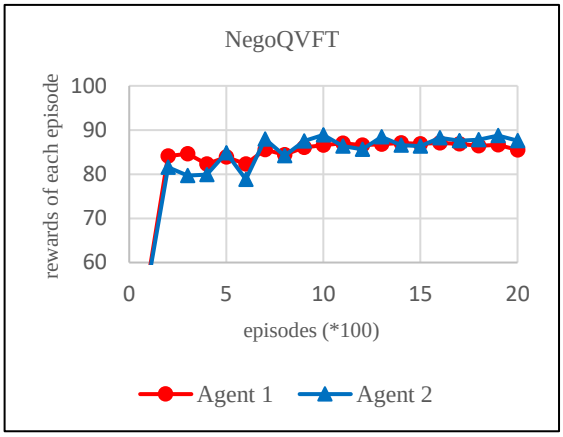
در بازی PENTAGON شکل (۹)، در الگوریتم RKT-MARL ویژگی عادلانه‌بودن فرآیند یادگیری بین دو عامل به‌وضوح مشخص است و از اپیزود ۸۰۰ ام به بعد منحنی یادگیری آن حالت پایداری به خود می‌گیرد. همچنین این الگوریتم به دلیل داشتن سازوکار انتقال دانش و دریافت دانش اولیه مناسب عملکرد بهتری دارد. الگوریتم CQ-learning نیز به‌دلیل برخوردهای زیاد بین عامل‌ها، منحنی یادگیری پایداری ندارد.

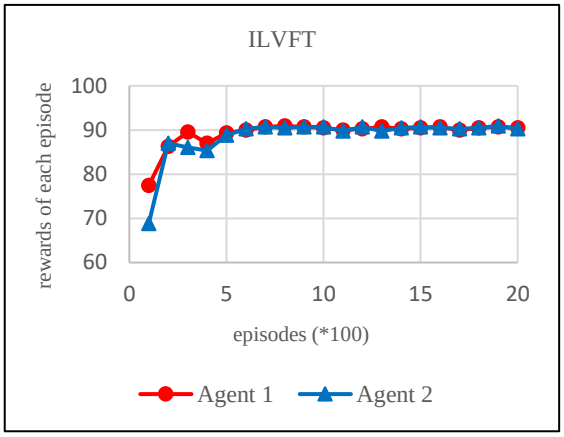
در بازی GW_nju شکل (۱۰)، سیاست‌های بهینه تک‌عاملی عامل‌ها با هم برخورد دارند و نیاز به برقراری هماهنگی مناسب‌تری دارند. در این بازی مقادیر بهینه برای پاداش نهایی عامل‌های ۱ و ۲ به‌ترتیب ۹۳ و ۹۱است. در الگوریتم‌های RKT-MARL، NegoSI و NegoQVFT منحنی یادگیری هر دو عامل درنهایت به مقادیری نزدیک مقدار بهینه همگرا می‌شود، با این تفاوت که برخلاف روش NegoQVFT، در روش‌های RKT-MARL و NegoSI، ابتدا عامل‌ها می‌توانند برای به‌دست‌آوردن سیاست امن و ثابت مذاکره کرده و سپس به‌صورت حریصانه حرکت کنند که این بیان‌گر توانایی برقراری هماهنگی بهتر در این روش‌هااست. درنهایت نکته قابل توجه این است که با وجود سازوکار مذاکره و انتقال دانش، عامل‌ها یاد می‌گیرند که هم سود بیشتری به خود و هم ضرر کمتری به عامل‌های دیگر برسانند و این شاهدی بر وجود همکاری بین عامل‌هااست. همچنین در الگوریتم RKT-MARL، عامل‌ها توانایی جلوگیری از برخورد را برای کسب پاداش بیشتر دارند، درحالی‌که در الگوریتم‌های یادگیری تقویتی قبلی، تمایل کمتری در برقراری هماهنگی داشتند و این موجب ازدست‌دادن پاداش توسط عامل‌ها می‌شد.

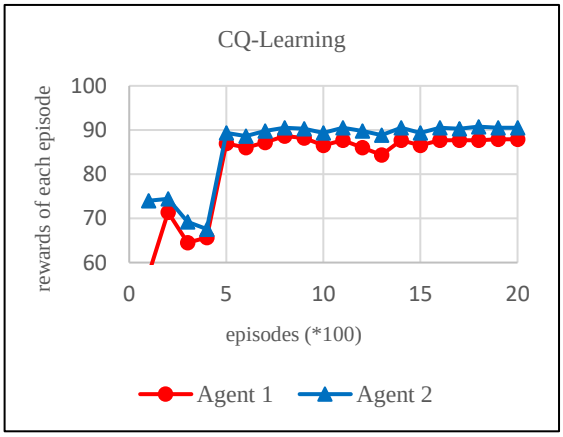


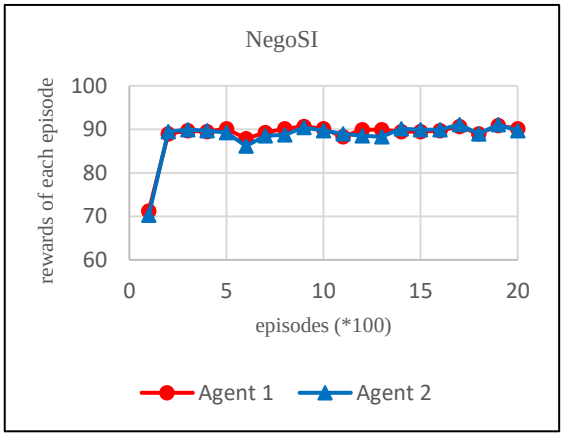
(شکل-۶): معیار REE برای هر الگوریتم در بازی ISR.

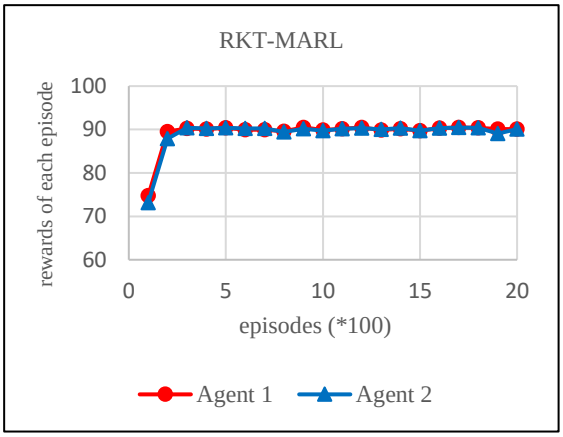
(Figure-6). REE criterion for each algorithm in ISR game.





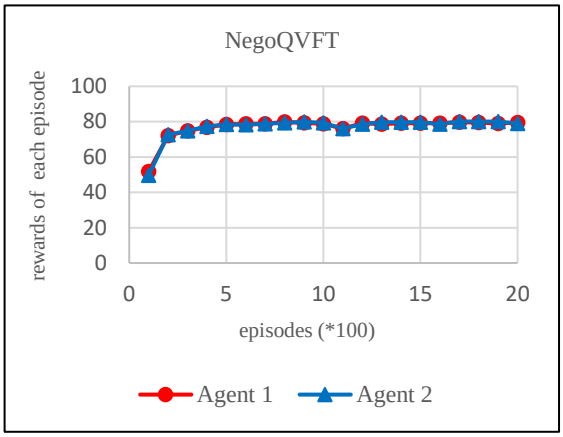


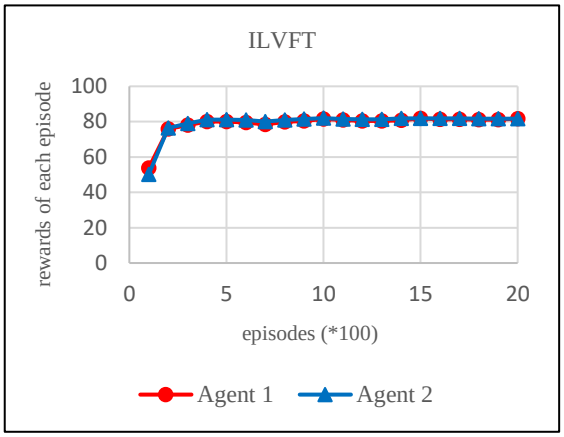


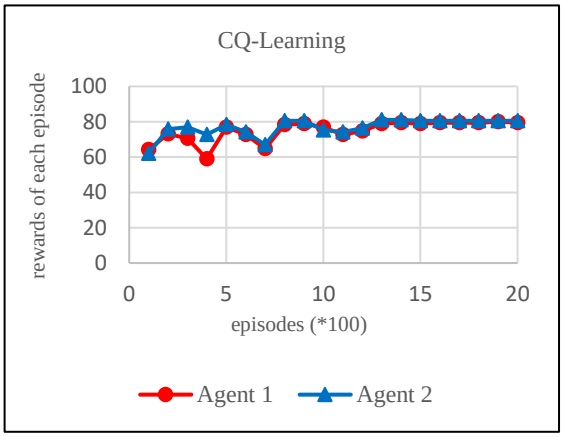


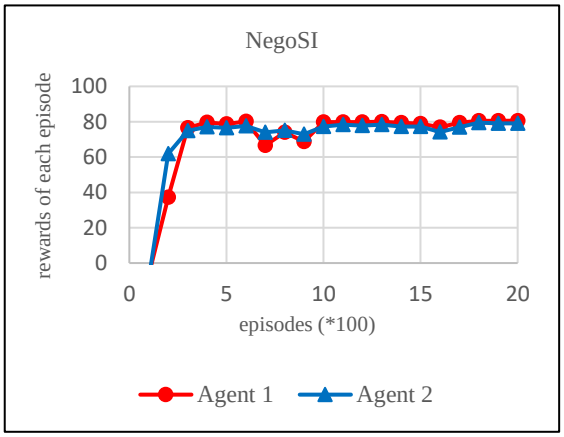
(شکل-۷). معیار REE برای هر الگوریتم در بازی SUNY.

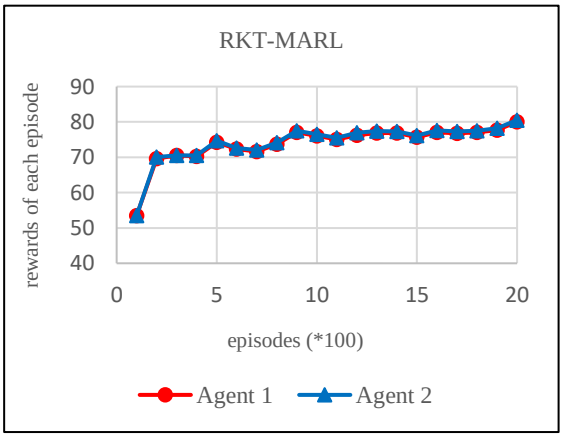
(Figure-7). REE criterion for each algorithm in SUNY game.





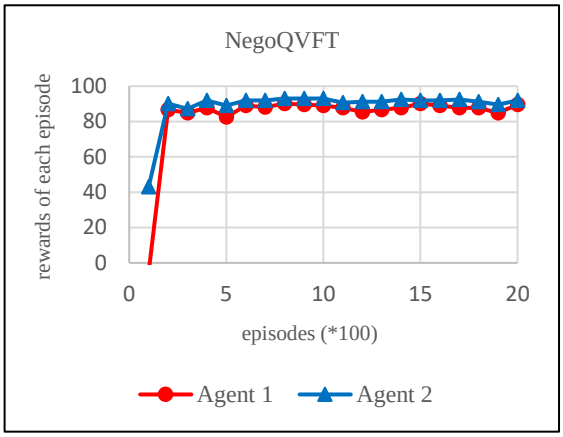


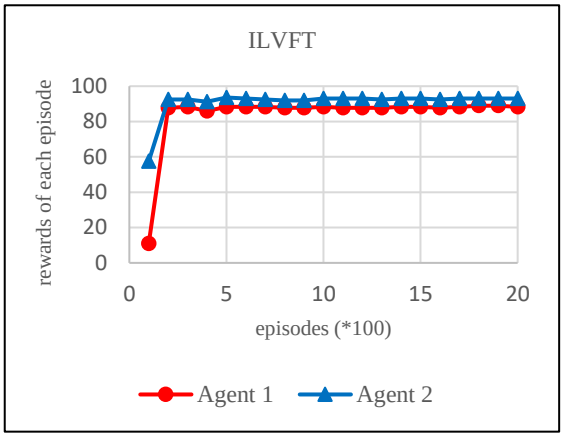


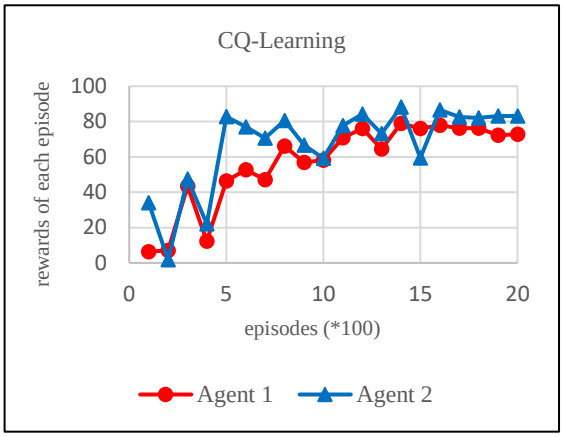


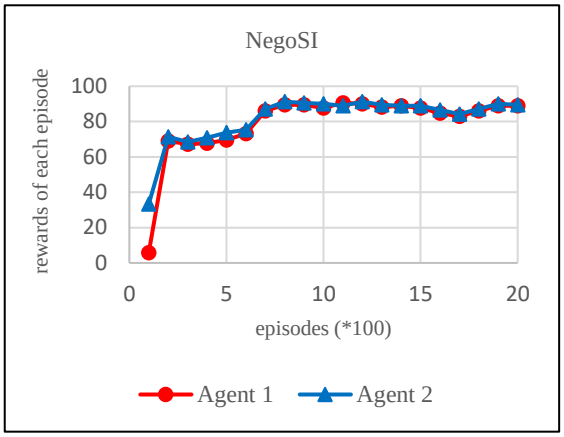
(شکل-۸). معیار REE برای هر الگوریتم در بازی MIT.

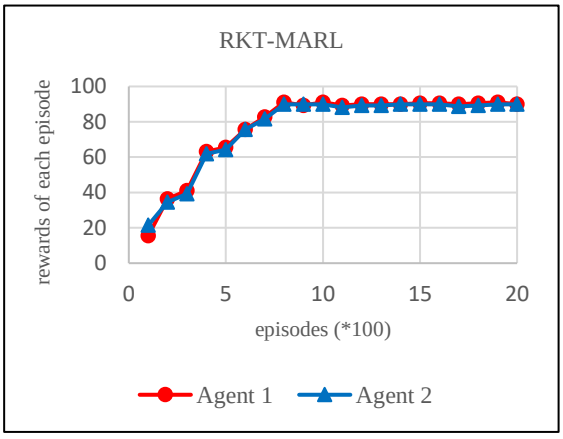
(Figure-8). REE criterion for each algorithm in MIT game.





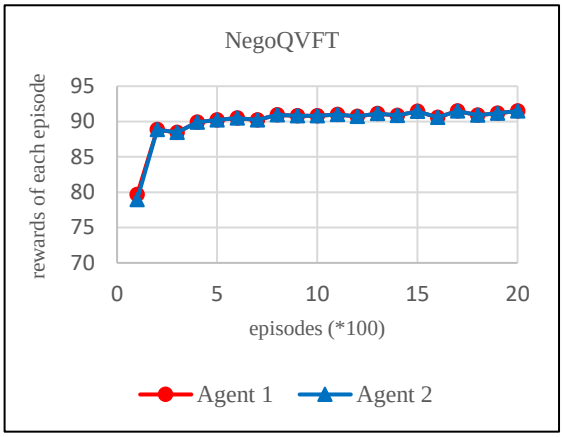


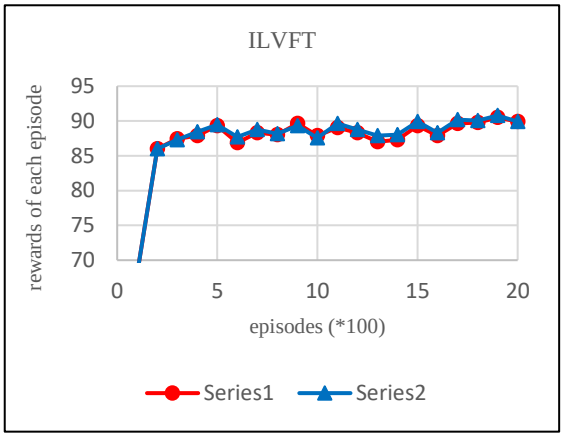


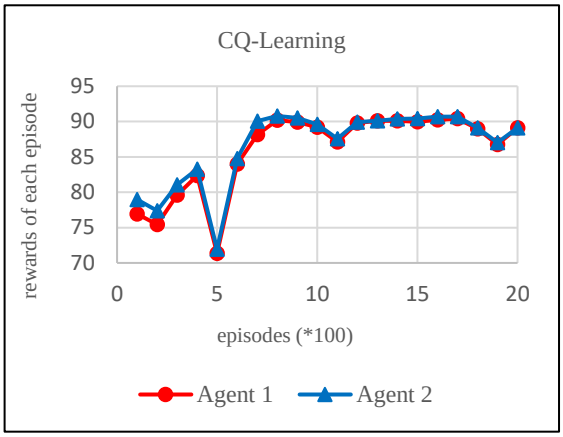


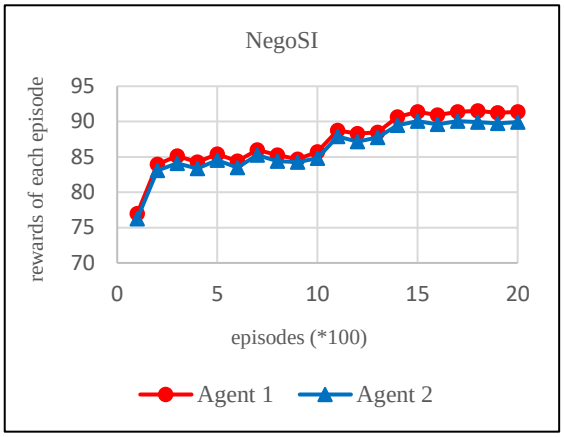
(شکل-۹). معیار REE برای هر الگوریتم در بازی PENTAGON.

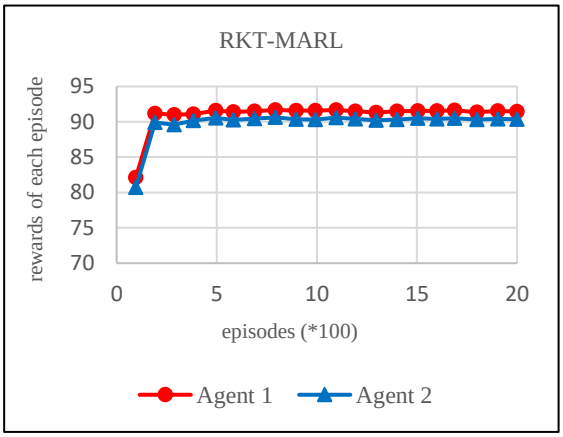
(Figure-9). REE criterion for each algorithm in PENTAGON game.











(شکل-۱۰). معیار REE برای هر الگوریتم در بازی GW_nju.

(Figure-10). REE criterion for each algorithm in GW_nju game.

۷- نتیجه‌گیری و کارهای آینده

یادگیری تقویتی چندعاملی یکی از مهم‌ترین روش‌های یادگیری در حوزه سیستم‌های چندعاملی و یادگیری ماشین است که در آن عامل‌ها می‌آموزند که برای دریافت پاداش بیشتر از محیط، چه عمل‌هایی را باید انجام دهند. در این مقاله یک الگوریتم یادگیری تقویتی چندعاملی برای حل مسائل یادگیری و هماهنگی در سیستم‌های چندعاملی تحت عنوان RKT-MARL معرفی شده که به صورت ترکیبی از روش‌های مبتنی بر تعاملات پراکنده، راه‌حل‌های تعادلی و انتقال دانش‌است. در این الگوریتم مفاهیم تعادلی و مذاکره برای جلوگیری از برخورد بین عامل‌ها و سازوکار انتقال دانش برای بهبود سرعت یادگیری عامل‌ها به کار می‌رود. همچنین از ضرایب تنظیم برای برقراری تعادل بین دانش انتقالی و اطلاعات محیطی استفاده می‌شود. نتایج حاصل، اهمیت و برتری الگوریتم پیشنهادی را از دیدگاه همگرایی سریع و پیچیدگی محاسباتی کم نسبت به الگوریتم‌های دیگر در این حوزه نشان می‌دهد.

برای ادامه کار، در حال برنامه‌ریزی جهت توسعه روش RKT-MARL برای سیستم‌های سه عاملی هستیم. همچنین به دنبال استفاده از یادگیری تقویتی چند هدفی می‌باشیم تا به صورت گسترده‌ای اطلاعات محیطی و دانش هماهنگی را برای یادگیری محلی ترکیب کنیم.

۸- مراجع 8-Refrence

[1] C. Yu, M. Zhang, F. Ren and G. Tan, "Multiagent learning of coordination in loosely coupled multiagent systems," *IEEE Transactions on Cybernetics*, vol. 45, no.12, pp. 2853-2867, 2015.

[2] J. Kober, J. A. Bagnell and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*,vol. 32, no. 11, pp. 1238-1274,2013.

[3] R. Babuška, L. Busoniu and B. D. Schutter, "Reinforcement learning for multi-agent systems," *IEEE International Conference on Emerging Technologies and Factory Automation*, 2006.

[4] K. Arulkumaran, M. P. Deisenroth, M. Brundage and A. A. Bharath, "A brief survey of deep reinforcement learning," arXiv preprint arXiv:1708.05866, 2017.

[5] Q. Zhang, P. Jiao, Q. Yin and L. Sun, "Coordinated Learning by Model Difference Identification in Multiagent Systems with Sparse Interactions," D*iscrete Dynamics in Nature and Society*, 2016.

[6] A. OroojlooyJadid and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning , arXiv preprint arXiv: 1908.03963, 2019.

[7] A. Nowé, P. Vrancx and Y. M. D. Hauwere, "Game theory and multi-agent reinforcement learning," *In Reinforcement Learning, Springer, Berlin, Heidelberg*, pp. 441-470, 2012.

[8] F. S. Melo and M. Velso, "Decentralized MDPs with sparse interactions," *Artificial Intelligence*, vol. 175, no.11, pp. 1757-1789, 2011.

[9] D. S. Bernstein, R. Givan, N. Immerman and S. Zilberstein, "The complexity of decentralized control of Markov decision processes," Mathematics *of operations research*, vol. 27, no. 4, pp. 819-840, 2002.

[10] A. M. Metelli, M. Mutti and M. Restelli, "Configurable Markov decision processes," *In International Conference on Machine Learning*, pp. 3491–3500, PMLR, 2018.

[11] L. Zhou, P. Yang, C. Chen and Y. Gao, "Multiagent reinforcement learning with sparse interactions by negotiation and knowledge transfer," *IEEE transactions on cybernetics*, vol. 47, no. 5, pp. 1238-1250, 2017.

[12] L. Canese, G.C. Cardarilli, L. Di Nunzio, R. Fazzolari, D. Giardino, M. Re and S. Spano, "Multi-Agent Reinforcement Learning: A Review of Challenges and Applications," Applied Sciences, p. 4948, 2021.

[13] J. Tahmoresnezhad and S. Hashemi, "Exploiting kernel-based feature weighting and instance clustering to transfer knowledge across domains," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 25, no. 1, pp. 292-307, 2017.

[14] J. Tahmoresnezhad and S. Hashemi, "Visual domain adaptation via transfer feature learning," *Knowledge and Information Systems*, vol. 50, no. 2, pp. 585-605, 2017.

[15] Y. Hu, Y. Gao and B. An, "Accelerating multiagent reinforcement learning by equilibrium transfer," *IEEE transactions on cybernetics*, vol. 45, no. 7, pp. 1289-1302, 2015.

[16] C. J. C. H. Watkins, "Learning from delayed rewards," (Doctoral dissertation, King's College, Cambridge), 1989.

[17] Y. M. D. Hauwere, P. Vrancx and A. Nowé, "Learning multi-agent state space representations," *In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, vol. 1, pp. 715-722, 2010.

[18] Y. Hu, Y. Gao and B. An, "Multiagent reinforcement learning with unshared value functions," *IEEE transactions on cybernetics*, vol. 45, no. 4, pp. 647-662, 2015.

[19] D. Abel, Y. Jinnai, S. Y. Guo, G. Konidaris and M. Littman, "Policy and Value Transfer in Lifelong Reinforcement Learning," *In International Conference on Machine Learning*, pp. 20-29, 2018.

[20] Y. Hu, Y. Gao and B. An, "Learning in multi-agent systems with sparse interactions by knowledge transfer and game abstraction," *In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 753-761, 2015.

[21] P. Mannion, "Knowledge-based multi-objective multi-agent reinforcement learning (Doctoral dissertation), 2017.

[22] T. Kujirai and T. Yokota, "Breaking Deadlocks in Multi-agent Reinforcement Learning with Sparse Interaction," *In Pacific Rim Internatioal Conference on Artifical Intelligence*, pp. 746-759, 2019.

[23] Y. Liu, Y. Hu, Y.Gao, Y.Chen and C.Fan, "Value Function Transfer for Deep Multi-Agent Reinforcement Learning Based on N-Step Returns," *In IJCAI*, pp. 457-463, 2019.

[24] F. S. Melo and M. Veloso, "Learning of coordination: Exploiting sparse interactions in multiagent systems," *In Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, vol. 2, pp. 773-780, 2009.

[25] P. Vrancx, Y. M. D. Hauwere and A. Nowé, "Transfer Learning for Multi-agent Coordination," *In ICAART (2)* , pp. 263-272, 2011.

[26] Melo, Francisco S., and Manuela Veloso. "Decentralized MDPs with sparse interactions," *Artificial Intelligence* 175.11 (2011).

[27] Sherafati F, Tahmoresnezhad J. Image Classification via Sparse Representation and Subspace Alignment. JSDP 2020; 17 (2) :58-47.

[28] Zandifar M, Tahmoresnezhad J. Sample-oriented Domain Adaptation for Image Classification. JSDP 2019; 16 (3) :148-129.

جعفر طهمورث نژاد مدرک دکترای مهندسی کامپیوتر-هوش مصنوعی خود را در سال ۱۳۹۴ از دانشگاه شیراز دریافت کردند. ایشان در راستای فعالیت‌های علمی خود، در حال حاضر به‌عنوان دانشیار دانشگاه صنعتی ارومیه در حال فعالیت هستند. علایق پژوهشی ایشان شامل حوزه‌های یادگیری ماشین، یادگیری انتقالی، داده‌کاوی و سامانه‌های امنیتی است.

نشانی رایانامه ایشان عبارت است از:

j.tahmores@it.uut.ac.ir

نیلوفر علوی مدرک کارشناسی خود را در رشته مهندسی فناوری اطلاعات در سال ۱۳۹۵ از دانشگاه شهید مدنی آذربایجان و مدرک کارشناسی ارشد رشته مهندسی فناوری اطلاعات را از دانشگاه صنعتی ارومیه در سال ۱۳۹۹ دریافت کرده‌است. علایق پژوهشی ایشان شامل حوزه‌های یادگیری انتقالی و یادگیری تقویتی است.

نشانی رایانامه ایشان عبارت است از:

nilooalavi504@it.uut.ac.ir