Tabriz Journal of Electrical Engineering (TJEE), vol. 55, no. 2, Summer 2025 Serial no. 112
DOI: 10.22034/tjee.2024.59109.4758

Event-Triggered Inverse Reinforcement Learning for Optimal Adaptive Leader-Follower Consensus of Unknown Multi-Agent Systems

Zahra Jahan[1], Abbas Dideban[1*], Farzaneh abdollahi[2]

[1] Electrical Engineering Department, Semnan University, Semnan, Iran
[2] Electrical Engineering Department, Amirkabir University, Tehran, Iran
E-mails: z.jahan@semnan.ac.ir; adideban@semnan.ac.ir; f_abdollahi@aut.ac.ir

Abstract
This paper introduces an event-triggered inverse reinforcement learning (IRL) approach for multi-agent discrete-time graphical games with unknown dynamics. In the IRL problem for these games, the expert and the learner systems are both leader-follower multi-agent systems. The optimal synchronization of the follower agents with the leader is the objective of the expert system. Learner agents intend to imitate the control inputs and states of the expert agents, while the expert value function is unknown to them. For the learner system, an IRL algorithm based on value iteration adaptive dynamic programming is proposed to recreate the unknown value function of the expert and to solve the event-triggered coupled Hamiltonian-Jacobi-Bellman equations without requiring the dynamics of either the expert or learner systems. To implement the proposed algorithm, an actor-critic-state penalty structure is used, and the unknown dynamics of the expert and learner multi-agent systems are approximated by neural network identifiers. Unlike traditional adaptive dynamic programming, where the control policies are periodically updated, in the presented method, the control policies and neural network weights are updated only at the triggered events. Therefore, the computational complexity is reduced. Finally, the efficiency of the proposed technique is demonstrated through simulation results.

Keywords
Inverse reinforcement learning, adaptive optimal control, event-triggering scheme, optimal leader-follower consensus, discrete-time graphical games, neural networks.

1- Short Paper

Many real-world multi-agent system applications require agents to follow a leader instead of reaching a common value. Therefore, solving the leader-follower consensus problem is important. In graphical games that solve the optimal leader-follower consensus, the interactions between agents are expressed by the communication graph structure, and the local error dynamics, control input, and each agent's performance index are determined by the local information of its neighbors. Most references in the field of reverse reinforcement learning do not address this issue, and the relationship between agents and their neighbors has not been considered. To solve these games, this study presents an adaptive, optimal, distributed method based on inverse reinforcement learning. The presented algorithm solves leader-follower graphical games without any knowledge of expert and learner system dynamics. The event-triggered method has been used, and the calculations are performed non-periodically, which leads to a reduction in calculations and execution time.

2- Proposed Work and Methodology

In this study, first, inverse reinforcement learning is introduced for leader-follower multi-agent discrete-time graphical games, where the expert system's purpose is to optimally synchronize the follower agents to the leader agent. The goal of the learner is to imitate the behavior of expert agents without knowing their performance. We propose an IRL algorithm in which learner agents reconstruct the unknown expert's value function by observing the expert's optimal control inputs and states. An actor-critic-state penalty structure approximates the value function, control policy, and expert's state-penalty weight to implement the suggested method. The simulation results confirm the presented algorithm synchronizes the follower agents with the leader and tracks the expert agents with the learner agents. In the following table, some references and the present study are compared.

Inverse reinforcement learning	Graphical games	Optimal leader-follower consensus	Unknown dynamics	Event-triggered control	Discrete-time or Continuous-time	References
-	✓	✓	✓	_	Discrete	[13]
✓	_	_	-	_	Continuous	[18]
✓	✓	-	✓	-	Continuous	[20]
_	✓	✓	✓	✓	Continuous	[25]
✓	✓	✓	✓	✓	Discrete	proposed method

3- Conclusion

We provide an event-triggered IRL method for multi-agent leader-follower discrete-time graphical games with unknown dynamics in this study. In the presented algorithm, the optimal synchronization of expert follower agents with the leader has been obtained. The learner agents also reconstruct the value functions of the expert according to the behavior of the expert agents to imitate the expert control inputs and their states. The approach is applied in the actor-critic-state penalty structure to determine the optimal value function and optimal control policies for the expert and learner agents and the expert reward weights for the learner. The provided approach does not need any prior understanding of system dynamics. For this purpose, neural network identities have been used to identify each agent's unknown dynamics. To reduce computational complexity, the control policies and weights for neural networks are updated only at the triggered event.

Tabriz Journal of Electrical Engineering (TJEE), vol. 55, no. 2, Summer 2025 Serial no. 112
DOI: 10.22034/tjee.2024.59109.4758

یادگیری تقویتی معکوس مبتنی بر رویداد برای اجماع رهبر-پیرو بهینه تطبیقی سیستم‌های چندعاملی ناشناخته

زهرا جهان
دانشجوی دکتری، دانشکده مهندسی برق و کامپیوتر، دانشگاه سمنان، سمنان، ایران

عباس دیدبان
دانشیار، دانشکده مهندسی برق و کامپیوتر، دانشگاه سمنان، سمنان، ایران

فرزانه عبداللهی
دانشیار، دانشکده مهندسی برق، دانشگاه امیرکبیر، تهران، ایران

چکیده
در این مقاله، یادگیری تقویتی معکوس مبتنی بر رویداد برای بازی‌های گرافی زمان گسسته چند عاملی با دینامیک ناشناخته معرفی می‌شود. در مساله یادگیری تقویتی معکوس برای این بازی‌ها، سیستم خبره و یادگیرنده هر دو یک سیستم چند عاملی رهبر-پیرو می‌باشند. هدف سیستم خبره هم زمانی بهینه عامل‌های پیرو به عامل رهبر است. عامل‌های یادگیرنده قصد دارند از حالت‌ها و ورودی‌های کنترلی عامل‌های خبره تقلید کنند بطوریکه تابع ارزش خبره برای آن‌ها ناشناخته است. یک الگوریتم یادگیری تقویتی معکوس بر مبنای برنامه‌ریزی پویای تطبیقی برای سیستم یادگیرنده توسعه داده شده است تا تابع عملکرد ناشناخته خبره را بازسازی کند و معادلات همیلتون-ژاکوبی-بلمن مبتنی بر رویداد را بدون نیاز به هیچ دانشی از دینامیک‌های سیستم خبره و یادگیرنده حل کند. برای اجرای الگوریتم ارائه شده، از ساختار شبکه عصبی نقاد-عملگر-پاداش‌حالت استفاده شده است و دینامیک‌های ناشناخته سیستم‌های چندعاملی خبره و یادگیرنده با شبکه‌های عصبی شناساگر تقریب زده می‌شوند. برخلاف برنامه‌ریزی پویای تطبیقی سنتی که قاعده کنترل بصورت دوره‌ای به‌روز می‌شود، در روش ارائه شده قاعده کنترل و وزن‌های شبکه عصبی فقط در لحظات رویداد به‌روز می‌شوند. بنابراین پیچیدگی محاسباتی کاهش می‌یابد. در انتها، نتایج شبیه‌سازی برای توصیف کارایی روش پیشنهادی ارائه شده است.

کلمات کلیدی
یادگیری تقویتی معکوس، کنترل بهینه تطبیقی، روش مبتنی بر رویداد، اجماع رهبر-پیرو بهینه، بازی‌های گرافی زمان گسسته، شبکه‌های عصبی.

نام نویسنده مسئول: عباس دیدبان
ایمیل نویسنده مسئول: adideban@semnan.ac.ir

تاریخ ارسال مقاله: ۱۴۰۲/۰۹/۰۱
تاریخ(های) اصلاح مقاله: ۱۴۰۳/۰۵/۱۹
تاریخ پذیرش مقاله: ۱۴۰۳/۰۶/۱۷

۱- مقدمه

با توجه به کاربردهای متعدد کنترل توزیع‌شده سیستم‌های چندعاملی در زمینه‌های مختلف مهندسی، تحقیقات در این زمینه مورد توجه بسیاری از محققان قرار گرفته است[۱، ۲]. در دهه‌های اخیر، رویکردهای کنترل توزیع شده به مسائل اجماع[1] پرداخته‌اند که به اجماع بدون رهبر و اجماع رهبر-پیرو تقسیم می‌شوند. یک قانون کنترلی برای هر عامل فقط با استفاده از اطلاعات محلی بدست آمده از همسایگانش طراحی می‌شود که باعث می‌شود همه‌ی عامل‌ها به یک مقدار مشترک همگرا شوند (اجماع بدون رهبر) [۳] یا مسیر یک رهبر (اجماع رهبر-پیرو) را دنبال کنند [۴، ۵]. اجماع رهبر-پیرو در سیستم‌های چند عاملی زمان گسسته، که مساله مورد مطالعه این مقاله می‌باشد، موضوع مطالعات متعددی بوده است[۶]. روش‌های ذکر شده در بالا بهینه نیستند. به همین دلیل مسائل کنترل چند عاملی بهینه، که در آن هر عامل سعی می‌کند شاخص عملکرد خود را بهینه کند و سیاست‌های بهینه خود را به دست آورد، می‌توانند در چارچوب تئوری بازی‌ها [۷] استفاده شوند.

در بسیاری از موارد، برای یافتن پاسخ‌های بهینه در بازی‌های چندعاملی، نیاز به حل معادلات همیلتون-ژاکوبی-بلمن کوبل[2] شده‌است. اما این معادلات غیرخطی هستند و حل آن‌ها سخت یا غیرممکن است. بنابراین، اخیرا از روش‌های تقریبی برای حل این معادلات استفاده می‌شود.

در دهه‌های اخیر، الگوریتم‌های یادگیری تقویتی[3] [۸] به عنوان روش‌های تقریبی موثر و مدرن برای حل بازی‌های چند عاملی توسعه داده شده اند [۹]. یکی از روش‌های یادگیری تقویتی، برنامه‌ریزی پویای تطبیقی است که شامل تکرار سیاست و تکرار ارزش می‌باشد. این روش‌ها می‌توانند با استفاده از ساختار نقاد-عملگر[4] به روش برخط اجرا شوند. نقاد شبکه عصبی است که تابع ارزش را تخمین می‌زند در حالی که عملگر کنترل بهینه را تخمین می‌زند.

[1] Consensus
[2] Coupled Hamiltonian-Jacobi-Bellman
[3] Reinforcement learning
[4] Actor-critic

بازی‌های گرافی[5] [۱۰] اولین بار برای سیستم‌های زمان پیوسته معرفی شدند که مسائل کنترل اجماع رهبر-پیرو توزیع شده بهینه را برای سیستم‌های خطی [۱۱] حل می‌کنند. این بازی‌ها برای سیستم‌های زمان گسسته خطی در [۱۲] معرفی شد که در آن یک الگوریتم تکرار ارزش برای حل بازی توسعه داده شده‌است. در [۱۳] یک روش تکرار سیاست بدون مدل برای حل بازی‌های گرافی زمان گسسته خطی تحت دینامیک ناشناخته توسعه داده شده است.

در تمام روش‌های ذکر شده در بالا به دانش قبلی از توابع هزینه نیاز است. این نیاز می‌تواند کاربرد کنترل بهینه و یادگیری تقویتی را محدود کند. بنابراین محققان به روش‌های یادگیری تقویتی معکوس که در آن توابع هزینه ناشناخته هستند، علاقه‌مند شده اند [۱۴].

در یادگیری تقویتی معکوس دو سیستم خبره و یادگیرنده وجود دارند که می‌توانند بصورت تک عاملی و یا چندعاملی باشند. عامل‌های خبره، ورودی‌های کنترلی و حالت‌های بهینه را بر اساس تابع عملکرد بهینه خود تولید می‌کنند. عامل‌های یادگیرنده تابع عملکرد بهینه خبره را نمی‌دانند. هدف آن‌ها تقلید از حالت‌ها و ورودی‌های کنترلی عامل‌های خبره است. یادگیرنده تابع عملکرد ناشناخته خبره را با مشاهده مسیر رفتار خبره بازسازی می‌کند به طوری‌که مسیرهای رفتاری مشابه با خبره داشته باشد.

یادگیری تقویتی معکوس برای سیستم‌های چندعاملی در مراجع [۱۵،۱۶] گسترش داده شده است که در آن دینامیک حالت، یک فرآیند تصمیم مارکوف است. در این مقالات، یادگیری تقویتی معکوس برای بازی‌های چند عاملی با معادلات دیفرانسیل در نظر گرفته نشده است. در اکثر سیستم‌های مهندسی عملی، مانند وسایل نقلیه خودران و سیستم‌های قدرت چند ناحیه‌ای [۱۷]، سیستم‌های دینامیکی با معادلات دیفرانسیل مدل‌سازی می‌شوند. مرجع [۱۸] به یادگیری تقویتی معکوس برای بازی‌های غیرهمکارانه چندبازیکنه می‌پردازد. در این مقاله، سیستم زمان پیوسته غیرخطی است که با معادلات دیفرانسیلی چندبازیکنه توصیف می‌شود. کنترل یادگیری تقویتی معکوس مبتنی بر داده برای بازی‌های چند نفره با معادلات دیفرانسیلی زمان پیوسته خطی در [۱۹] انجام شده است. [۲۰] به حل بازی‌های گرافی چندعاملی زمان پیوسته با یادگیری تقویتی معکوس پرداخته است. در این مقاله فرض می‌شود ارتباط بین عامل‌ها فقط تحت توپولوژی گراف است و مساله یادگیری تقویتی معکوس برای اجماع رهبر-پیرو هنوز انجام نشده است.

تمام مقالات ذکر شده در بالا از روش‌های مبتنی بر زمان استفاده می‌کنند که محاسبات را در تمام لحظات انجام می‌دهند. این فرآیند زمان بر است و پیچیدگی محاسبات را افزایش می‌دهد. با توجه به کاهش قابل توجه محاسبات، روش کنترل مبتنی بر رویداد اخیراً توجه بسیاری از محققان را به خود جلب کرده است [۲۱،۲۲]. روش‌های مبتنی بر رویداد برای سیستم‌های زمان گسسته بدون در نظر گرفتن کنترل بهینه در [۲۳] ارائه شده‌اند. در [۲۴] از یک روش کنترل تطبیقی مبتنی بر رویداد برای سیستم‌های زمان گسسته غیرخطی ناشناخته استفاده شده است. حل بازی‌های گرافی زمان پیوسته با روش‌های مبتنی بر رویداد در [۲۵] انجام شده است.

در بسیاری از کاربردهای عملی سیستم‌های چندعاملی، عامل‌ها لزوما نمی‌خواهند به یک مقدار مشترک همگرا بشوند و نیاز هست که یک رهبر را دنبال کنند. بنابراین حل اجماع رهبر-پیرو دارای اهمیت است. در بازی‌های گرافی که به حل اجماع رهبر-پیرو بهینه می‌پردازند، تعاملات بین عامل‌ها با ساختار گراف ارتباطی بیان می‌شود و دینامیک خطای محلی، ورودی کنترلی و شاخص عملکرد هر عامل به اطلاعات محلی همسایگانش بستگی دارد. در بسیاری از مراجع به این موضوع پرداخته نشده است و ارتباط عامل‌ها با همسایگان و موضوع هم زمانی رهبر-پیرو در نظر گرفته نشده است[۲۴-۱۵]. تاکنون حل این بازی‌ها با روش‌های یادگیری تقویتی معکوس و روش‌های مبتنی بر رویداد انجام نشده است. بنابراین این مطالعه به حل بازی‌های گرافی زمان گسسته چند عاملی ناشناخته با یادگیری تقویتی معکوس مبتنی بر رویداد می‌پردازد.

نوآوری‌های مقاله حاضر به شرح زیر می‌باشد:

• معرفی یادگیری تقویتی معکوس برای بازی‌های گرافی زمان گسسته چند عاملی رهبر-پیرو که در آن هدف سیستم خبره هم زمانی بهینه عامل‌های پیرو به عامل رهبر است. هدف عامل‌های یادگیرنده تقلید از رفتار عامل‌های خبره بدون دانستن تابع عملکرد آن‌ها می‌باشد.

• تاکنون هیچ مطالعه‌ای به حل بازی‌های گرافی زمان گسسته چند عاملی رهبر-پیرو با روش‌های یادگیری تقویتی معکوس نپرداخته است. بنابراین در این مقاله یک الگوریتم توزیع شده بهینه تطبیقی جدید بر مبنای یادگیری تقویتی معکوس برای حل این بازی‌ها ارائه شده است.

• الگوریتم ارائه شده بازی‌های گرافی رهبر-پیرو را بدون نیاز به هیچ دانشی از دینامیک‌های سیستم خبره و یادگیرنده حل می‌کند.

• استفاده از روش مبتنی بر رویداد و انجام محاسبات بصورت غیر دوره‌ای که منجر به کاهش محاسبات و زمان اجرا می‌شوند.

ادامه این مقاله شامل بخش‌های زیر می‌باشد. در بخش دوم به تئوری گراف‌ها، فرمول‌بندی سیستم‌های چندعاملی خبره و یادگیرنده با روش مبتنی بر رویداد پرداخته می‌شود. در بخش سوم، الگوریتم یادگیری تقویتی معکوس مبتنی بر رویداد برای بازی‌های گرافی رهبر-پیرو ارائه می‌شود. در بخش چهارم، الگوریتم یادگیری تقویتی معکوس مبتنی بر رویداد برای بازی‌های گرافی رهبر-پیرو ناشناخته تحت چارچوب نقاد-عملگر-پاداش‌حالت-شناساگر اجرا می‌شود. بخش پنجم نیز نتایج شبیه‌سازی را برای نمایش عملکرد و صحت روش معرفی شده نشان می‌دهد. در نهایت، جمع‌بندی در بخش ششم ارائه می‌شود.

۲- مقدمات

در این بخش ابتدا تئوری گراف‌ها بیان می‌شود. در ادامه به فرمول‌بندی سیستم‌های چندعاملی خبره و یادگیرنده برای هم زمانی بهینه رهبر-پیرو با مکانیزم مبتنی بر رویداد پرداخته می‌شود.

۲-۱- گراف‌ها

توپولوژی تعاملی برای تبادل اطلاعات بین N عامل پیرو و یک عامل رهبر توسط گراف جهت‌دار $Gr=(P,\Sigma)$ توصیف می‌شود که در آن $P=\{p_0,.....,p_N\}$ مجموعه گره‌های گراف است و $\Sigma\subset P\times P$ مجموعه شاخه‌های گراف است. ماتریس اتصال[6] C به صورت $C=[c_{ij}]\in\mathbb{R}^{N\times N}$ تعریف می‌شود به طوریکه اگر $(p_j,p_i)\in\Sigma$ آنگاه $c_{ij}>0$ در غیر اینصورت $c_{ij}=0$ است. مجموعه همسایه‌های هر گره p_i برابر با $N_i=\{p_j:(p_j,p_i)\in\Sigma\}$ است. ماتریس درجه وارد شونده[7]، یک ماتریس قطری $Q=diag\{q_i\}$ با $q_i=\sum_{j\in N_i}c_{ij}$ است. ماتریس لاپلاسین گراف بصورت $L=Q-C$ تعریف می‌شود. عامل رهبر با ۰ نشان داده می‌شود و اطلاعات می‌تواند از رهبر به همسایگان خود ارسال شود. ماتریس اتصال برای رهبر با $G=diag\{g_i\}\in\mathbb{R}^{N\times N}$ نشان داده می‌شود. $g_i\geq 0$ بهره اتصال عامل i است که اگر عامل i به رهبر متصل باشد، غیرصفر است. توجه داشته باشید که رهبر حداقل به یکی از گره‌های پیرو متصل است. درخت پوشا، درختی است با مجموعه‌ای از کمترین تعداد یال‌ها در حالی که تمام رئوس را شامل می‌شود. در این مقاله فرض می‌شود گراف دارای درخت پوشا است.

[5] Graphical games
[6] Connectivity matrix
[7] In-degree matrix

۲-۲- سیستم‌های خبره و یادگیرنده بازی‌های گرافی رهبر-پیرو

دو سیستم چندعاملی را بصورت بازی‌های گرافی رهبر-پیرو روی دو گراف مجزا از هم در نظر بگیرید. دینامیک‌های محلی هر عامل خبره و یادگیرنده به ترتیب بصورت زیر هستند:

$$s_{ei}(n+1)=As_{ei}(n)+B_iu_{ei}(n) \quad (۱)$$

$$s_i(n+1)=As_i(n)+B_iu_i(n) \quad (۲)$$

که $s_{ei}(n)\in\mathbb{R}^k$ و $u_{ei}(n)\in\mathbb{R}^m$ به ترتیب حالت و ورودی کنترلی عامل خبره i و $s_i(n)\in\mathbb{R}^k$ و $u_i(n)\in\mathbb{R}^m$ به ترتیب حالت و ورودی کنترلی عامل یادگیرنده i را برای $i=1,2,....,N$ نشان می‌دهند، A و B_i به ترتیب ماتریس‌های حالت و ورودی کنترل هستند. در این کار فرض می‌شود دینامیک‌های سیستم‌های چندعاملی ناشناخته هستند. دینامیک‌های رهبر خبره و یادگیرنده به‌ترتیب به‌صورت زیر تعریف می‌شوند:

$$s_{e0}(n+1)=As_{e0}(n) \quad (۳)$$

$$s_0(n+1)=As_0(n) \quad (۴)$$

خطای محلی $\varepsilon_{ei}(n)\in\mathbb{R}^k$ برای هر عامل خبره i به‌صورت زیر تعریف می‌شود:

$$\varepsilon_{ei}(n)=\sum_{j\in N_i}c_{ij}\left(s_{ej}(n)-s_{ei}(n)\right)+g_i\left(s_{e0}(n)-s_{ei}(n)\right) \quad (۵)$$

خطای محلی $\varepsilon_i(n)\in\mathbb{R}^k$ برای هر عامل یادگیرنده i نیز مشابه (۵) تعریف می‌شود.

بردار خطای محلی کلی برای همه عامل‌های خبره برابر است با:

$$\varepsilon_e(n)=-\left((L+G)\otimes I_k\right)\left(s_e(n)-s_{e0}(n)\right) \quad (۶)$$

که $s_e=\left[s_{e1}^T,\ldots,s_{eN}^T\right]^T\in\mathbb{R}^{kN}$ ، $\varepsilon_e=\left[\varepsilon_{e1}^T,\ldots,\varepsilon_{eN}^T\right]^T\in\mathbb{R}^{kN}$ و $s_{e0}=\left[s_{e0}^T,\ldots,s_{e0}^T\right]^T\in\mathbb{R}^{kN}$.

I_k ماتریس واحد $k\times k$ است و $\otimes$ ضرب کرونکر را نشان می‌دهد.

بردار خطای هم‌زمانی بصورت زیر می‌باشد:

$$\eta_e(n)=s_e(n)-s_{e0}(n) \quad (۷)$$

اگر گراف دارای یک درخت پوشا باشد و برای یک عامل ریشه $g_i\neq 0$ باشد، $(L+G)$ غیرتکین است[۲۶]. همانطور که در مرجع [۱۲] نشان داده شده است اگر $(L+G)$ غیرتکین باشد بردار خطای هم زمانی بصورت زیر کراندار است:

$$\left\|\eta_e(n)\right\|\le\left\|\varepsilon_e(n)\right\|/\underline{\sigma}(L+G) \quad (۸)$$

$\underline{\sigma}(.)$ مقدار تکین حداقل ماتریس است.

برای طراحی یک وضعیت کنترل مبتنی بر رویداد، یک رشته افزایشی یکنواخت از لحظات زمان $\{n_m^i\}_{m=0}^{\infty}$ را برای هر عامل i تعریف می‌کنیم که n_m^i، mامین لحظه نمونه برداری با $n_0^i=0$ است و برای هر عامل i رابطه $n_m^i<n_{m+1}^i$ برقرار است. برای کاهش پیچیدگی محاسباتی، ورودی کنترل و وزن‌های شبکه عصبی فقط در لحظات نمونه برداری $n_0^i,n_1^i,....$ به‌روزرسانی می‌شوند. زمانی که شرط مبتنی بر رویداد زیر برای هر عامل خبره i در $n=n_m^i$ نقض شود یک رویداد اتفاق افتاده‌است و لحظات نمونه برداری بدست می‌آیند:

$$\left\|e_{ei}(n)\right\|\le e_T \quad (۹)$$

که e_T آستانه است و $e_{ei}(n)$ خطای مبتنی بر رویداد خبره بصورت زیر است:

$$e_{ei}(n)=\varepsilon_{ei}(n_m^i)-\varepsilon_{ei}(n),\forall n\in[n_m^i,n_{m+1}^i) \quad (۱۰)$$

که $\varepsilon_{ei}(n)$ و $\varepsilon_{ei}(n_m^i)$ به ترتیب بردار خطای محلی خبره در لحظات جاری و لحظات نمونه برداری هستند. این خطا برای سیستم یادگیرنده نیز به همین ترتیب تعریف می‌شود.

فرض ۱) ثابت مثبت D_i وجود دارد به‌طوری‌که:

$$\left\|e_{ei}(n+1)\right\|\le D_i\left\|e_{ei}(n)\right\|+D_i\left\|\varepsilon_{ei}(n)\right\| \quad (۱۱)$$

قضیه ۱. مشابه با مرجع [۲۴]، برای سیستم‌های خطی زمان گسسته، خطای مبتنی بر رویداد شرط زیر را برآورده می‌کند:

$$\left\|e_{ei}(n)\right\|\le e_T=\frac{1-(2D_i)^{n-n_m^i}}{1-2D_i}D_i\left\|\varepsilon_{ei}(n_m^i)\right\| \quad (۱۲)$$

یادآوری ۱. در لحظات رویداد n_m^i، خطای مبتنی بر رویداد خبره و یادگیرنده صفر می‌شوند. در حالی که سیاست‌های کنترلی عامل i فقط در $n=n_m^i$ به‌روز می‌شوند و در لحظات $[n_m^i,n_{m+1}^i)$ ثابت هستند یعنی $u_{ei}(n)=u_{ei}(n_m^i),u_i(n)=u_i(n_m^i),\forall n\in[n_m^i,n_{m+1}^i)$

با استفاده از (۱) و (۵) دینامیک‌های خطای محلی مبتنی بر رویداد عامل خبره i برای $n\in[n_m^i,n_{m+1}^i)$ بصورت زیر بدست می‌آید:

$$\varepsilon_{ei}(n+1)=A\varepsilon_{ei}(n)-(q_i+g_i)B_iu_{ei}(n_m^i)+\sum_{j\in N_i}c_{ij}B_ju_{ej}(n_m^j) \quad (۱۳)$$

دینامیک‌های خطای محلی مبتنی بر رویداد عامل‌های یادگیرنده $\varepsilon_i(n+1)$ نیز مشابه با (۱۳) بدست می‌آیند.

تابع ارزش هر عامل خبره بصورت زیر تعریف می‌شود:

$$V_{ei}(\varepsilon_{ei}(n))=\sum_{\cup_m[n_m^i,n_{m+1}^i)=[0,\infty)}\sum_{n=n_m^i}^{n_{m+1}^i}U_{ei}(\varepsilon_{ei},u_{ei},u_{-ei})$$
$$=\sum_{\cup_m[n_m^i,n_{m+1}^i)=[0,\infty)}\sum_{n=n_m^i}^{n_{m+1}^i}\left\{\begin{matrix}O_{ei}(n)\\+u_{ei}^T(n_m^i)R_{eii}u_{ei}(n_m^i)\\+\sum_{j\in N_i}u_{ej}^T(n_m^j)R_{eij}u_{ej}(n_m^j)\end{matrix}\right\} \quad (۱۴)$$

که $O_{ei}(n)=\varepsilon_{ei}^T(n)o_{ei}\varepsilon_{ei}(n)\in\mathbb{R}>0$ پاداش حالت خبره با وزن پاداش حالت $o_{ei}=o_{ei}^T\in\mathbb{R}^{k\times k}$ است. $R_{eii}>0\in\mathbb{R}^{m\times m}$ و $R_{eij}>0\in\mathbb{R}^{m\times m}$ وزن‌های ورودی کنترلی خبره هستند. $u_{-ei}=\{u_{ej}\mid j\in N_i\}$ مجموعه‌ای از ورودی‌های کنترلی برای همسایه‌های عامل خبره i است. تابع ارزش عامل‌های یادگیرنده $V_i(\varepsilon_i(n))$ نیز مشابه (۱۴) تعریف می‌شود.

با در نظر گرفتن اختلاف اول معادله (۱۴)، معادله بلمن مبتنی بر رویداد برای عامل خبره i و $n\in[n_m^i,n_{m+1}^i)$ بصورت زیر به‌دست می‌آید:

$$V_{ei}(\varepsilon_{ei}(n)) = U_{ei}(\varepsilon_{ei}(n), u_{ei}(n_m^i), u_{-ei}(n_m^i)) + V_{ei}(\varepsilon_{ei}(n+1)) \quad (۱۵)$$

هدف از مسئله بهینه‌سازی بازی‌های گرافی، یافتن تابع ارزش بهینه V_{ei}^* و سیاست کنترلی بهینه u_{ei}^* است. بر اساس اصل بهینه سازی بلمن داریم:

$$V_{ei}^*(\varepsilon_{ei}(n)) = \min_{u_{ei}} \left(U_{ei} + V_{ei}^*(\varepsilon_{ei}(n+1)) \right) \quad (۱۶)$$

در نتیجه، سیاست کنترل بهینه برای هر عامل خبره i برابر است با:

$$u_{ei}^*(n) = u_{ei}^*(n_m^i) = (q_i + g_i) R_{eii}^{-1} B_i^T \nabla V_{ei}^* \left(\varepsilon_{ei} \left(n_m^i + 1 \right) \right) \quad (۱۷)$$

که $\nabla V_{ei}^*(\varepsilon_{ei}(n_m^i+1)) = \partial V_{ei}^*(\varepsilon_{ei}(n_m^i+1)) / \partial \varepsilon_{ei}(n_m^i+1)$.

در روش کنترل مبتنی بر رویداد ما، سیاست‌های کنترل بهینه تنها در لحظات رویداد به روز می‌شوند. در نتیجه، آن‌ها با استفاده از حالت‌های نمونه برداری شده $\varepsilon_{ei}(n_m^i)$ به جای حالت‌های فعلی $\varepsilon_{ei}(n)$ برای $n \in [n_m^i, n_{m+1}^i)$ به‌روز می‌شوند.

با جایگزینی معادله (۱۷) در معادله بلمن مبتنی بر رویداد خبره (۱۶)، معادلات بهینه بلمن مبتنی بر رویداد خبره زیر به‌دست می‌آیند:

$$\begin{aligned} V_{ei}^*(\varepsilon_{ei}(n)) &= V_{ei}^*(\varepsilon_{ei}(n+1)) + U_{ei}^*(\varepsilon_{ei}, u_{ei}^*, u_{-ei}^*) \\ &= V_{et}^*(\varepsilon_{ei}(n+1)) \\ &+ \left(\begin{array}{l} \varepsilon_{ei}^T(n) o_{ei} \varepsilon_{ei}(n) + \left(\begin{array}{l} (q_i + g_i)^2 \times \nabla V_{ei}^* \left(\varepsilon_{ei} \left(n_m^i + 1 \right) \right)^T \\ \times B_{ei} R_{eii}^{-1} B_{ei}^T \nabla V_{ei}^* \left(\varepsilon_{ei} \left(n_m^i + 1 \right) \right) \end{array} \right) \\ + \sum_{j \in N_i} \left(\begin{array}{l} (q_j + g_j)^2 \times \nabla V_{ej}^* \left(\varepsilon_{ej} \left(n_m^j + 1 \right) \right)^T \\ \times B_{ej} R_{ejj}^{-1} R_{eij} R_{ejj}^{-1} B_{ej}^T \nabla V_{ej}^* \left(\varepsilon_{ej} \left(n_m^j + 1 \right) \right) \end{array} \right) \end{array} \right) \end{aligned} \quad (۱۸)$$

برای بدست آوردن سیاست و ارزش بهینه نیاز به حل معادله (۱۸) است که حل آن سخت یا غیرممکن است. بنابراین در ادامه الگوریتمی ارایه شده است که بطور تقریبی پاسخ‌های بهینه را بدست می‌آورد.

قضیه ۲. فرض کنید V_{ei}^* پاسخ معادلات بهینه بلمن مبتنی بر رویداد خبره (۱۸) است. برای $n \in [n_m^i, n_{m+1}^i)$ سیاست کنترل u_{ei}^* (۱۷) وجود دارد. فرض کنید گراف ارتباطی دارای یک درخت پوشا است و برای حداقل یک عامل $g_i \neq 0$ است. سپس دینامیک‌های خطای محلی (۱۳) پایدار مجانبی هستند و حالت همه عامل‌ها به حالت رهبر هم زمان می‌شوند.

اثبات: $V_{ei}^*(\varepsilon_{ei}) > 0$ را به عنوان تابع لیاپانف برای (۱۳) در نظر می‌گیریم و از (۱۸) داریم:

$$V_{ei}^* \left(\varepsilon_{ei} \left(n_m^i + 1 \right) \right) - V_{ei}^* \left(\varepsilon_{ei} (n_m^i) \right) = -U_i^* \left(\varepsilon_{ei}, u_{ei}^*, u_{-ei}^* \right) < 0 \quad (۱۹)$$

بنابراین دینامیک‌های خطای محلی (۱۳) پایدار مجانبی هستند. از آن‌جایی‌که گراف دارای یک درخت پوشا است و برای حداقل یک عامل $g_i \neq 0$ است، بنابراین با توجه به معادله (۸)، حالت همه‌ی عامل‌ها به حالت رهبر هم زمان می‌شوند.

فرض ۲) همه‌ی عامل‌ها در سیستم چندعاملی یادگیرنده، وزن‌های پاداش حالت و وزن‌های کنترل خود را می‌دانند، اما اطلاعاتی درباره O_{ei} ، R_{eii} و R_{eij} سیستم خبره ندارند.

فرض ۳) هر عامل یادگیرنده i می‌تواند مسیر ورودی کنترلی عامل خبره i یعنی u_{ei}^* را مشاهده کند. توجه کنید که سیستم چندعاملی یادگیرنده و خبره، بازی‌های گرافی مجزا از هم دارند و عامل‌های آن‌ها روی گراف مشابه همسایه نیستند.

مساله یادگیری تقویتی معکوس: دو سیستم چندعاملی رهبر-پیرو خبره و یادگیرنده وجود دارند. هدف سیستم خبره هم زمانی بهینه عامل‌های پیرو به رهبر است. عامل‌های یادگیرنده می‌خواهند رفتارهای مشابه با عامل‌های خبره داشته باشند، یعنی حالت‌ها و ورودی‌های کنترلی آن‌ها مشابه حالت‌ها و ورودی‌های کنترلی خبره باشند. اما شاخص عملکرد خبره (۱۴) برای یادگیرنده ناشناخته است. در نتیجه، یادگیرنده باید با مشاهده مسیرهای ورودی کنترلی خبره u_{ei}^* پاداش حالت ناشناخته O_{ei} را برای تابع ارزش خود یاد بگیرد به‌طوری‌که حالت و ورودی کنترلی خبره را دنبال کند.

۳- الگوریتم یادگیری تقویتی معکوس مبتنی بر رویداد برای بازی‌های گرافی رهبر-پیرو

در این قسمت برای حل مساله یادگیری تقویتی معکوس مبتنی بر رویداد بازی‌های گرافی رهبر-پیرو چند بازیکنه یک الگوریتم ارایه شده است که ترکیبی از کنترل بهینه و کنترل بهینه معکوس است. در این الگوریتم ابتدا برنامه ریزی پویای مبتنی بر رویداد برای سیستم چندعاملی خبره ارائه شده است تا بازی‌های گرافی رهبر-پیرو چند بازیکنه را حل نماید. این قسمت از الگوریتم ارائه شده معادلات بهینه بلمن مبتنی بر رویداد خبره (۱۸) را برای ارزش بهینه و سیاست بهینه حل می‌کند. در ادامه ی الگوریتم از دو حلقه ی تکراری استفاده شده است. ابتدا عامل‌های یادگیرنده پاسخ‌های بهینه را برای هم زمانی عامل‌های پیرو به رهبر با یک تخمین جاری پاداش حالت یادگیرنده از پاداش حالت خبره O_{ei} و استفاده از روش کنترل بهینه برمبنای یادگیری تقویتی بدست می‌آورند. در ادامه سیستم چندعاملی یادگیرنده با مشاهده ورودی کنترلی خبره u_{ei}^* و استفاده از کنترل بهینه معکوس، پاداش حالت ناشناخته O_{ei} را تعیین می‌کند.

۳-۱- هم زمانی بهینه رهبر-پیرو سیستم چند عاملی یادگیرنده برمبنای پاداش حالت داده شده O_i

در این قسمت، مساله هم زمانی بهینه رهبر-پیرو برای عامل‌های یادگیرنده با پاداش دلخواه داده شده حل می‌شود. با در نظر گرفتن اختلاف اول تابع ارزش یادگیرنده، معادله بلمن مبتنی بر رویداد عامل یادگیرنده i برای $n \in [n_m^i, n_{m+1}^i)$ بصورت زیر بدست می‌آید:

$$V_i(\varepsilon_i(n)) = U_i(\varepsilon_i(n), u_i(n_m^i), u_{-i}(n_m^i)) + V_i(\varepsilon_i(n+1)) \quad (۲۰)$$

هدف بازی سیستم چندعاملی یادگیرنده یافتن u_i^* برای هر عامل است به‌طوری‌که

$$\begin{aligned} u_i^*(n) = u_i^*(n_m^i) &= \min_{u_{in}} \left(U_i + V_i^* \left(\varepsilon_i(n+1) \right) \right) \\ &= \left(q_i + g_i \right) R_{ii}^{-1} B_i^T \nabla V_i^* \left(\varepsilon_i \left(n_m^i + 1 \right) \right) \end{aligned} \quad (۲۱)$$

که $\nabla V_i^*(\varepsilon_i(n_m^i+1)) = \partial V_i^*(\varepsilon_i(n_m^i+1)) / \partial \varepsilon_i(n_m^i+1)$.

همانند سیستم چندعاملی خبره، سیاست‌های کنترل بهینه یادگیرنده تنها در لحظات رویداد به‌روز می‌شوند.

با جایگزینی معادله (۲۱) در معادله بلمن مبتنی بر رویداد یادگیرنده (۲۰)، معادلات بهینه بلمن مبتنی بر رویداد یادگیرنده زیر به‌دست می‌آیند:

$$V_i^*(\varepsilon_i(n)) = V_i^*(\varepsilon_i(n+1)) + \left[\begin{array}{l} \varepsilon_i^T(n) o_i \varepsilon_i(n) + \left(\begin{array}{l}(q_i+g_i)^2 \times \nabla V_i^*\left(\varepsilon_i\left(n_m^i+1\right)\right)^T \\ \times B_i R_{ii}^{-1} B_i^T \nabla V_i^*\left(\varepsilon_i\left(n_m^i+1\right)\right)\end{array}\right) \\ + \sum_{j\in N_i}\left(\begin{array}{l}(q_j+g_j)^2 \\ \times \nabla V_j^*\left(\varepsilon_j\left(n_m^j+1\right)\right)^T B_j R_{jj}^{-1} \\ \times R_{ij} R_{jj}^{-1} B_j^T \nabla V_j^*\left(\varepsilon_j\left(n_m^j+1\right)\right)\end{array}\right) \end{array}\right] \quad (۲۲)$$

۳-۲- یادگیری وزن‌های پاداش با کنترل بهینه معکوس

حلقه بیرونی در الگوریتم ارائه شده بر مبنای کنترل بهینه معکوس است. زمانی که $(s_i,u_i^*) \neq (s_{ei},u_{ei}^*)$، یادگیرنده با مشاهده مسیر رفتار ورودی کنترلی خبره u_{ei}^*، تخمین O_i نسبت به O_{ei} را برای تابع ارزش یادگیرنده بهبود می‌دهد. با تکرار این دو مرحله یادگیری، در نهایت همگرایی $(s_i,u_i^*) = (s_{ei},u_{ei}^*)$ توسط یادگیرنده حاصل می‌شود.

بر اساس کنترل بهینه معکوس، یادگیرنده با مشاهده مسیرهای ورودی کنترل بهینه خبره u_{ei}^* وزن پاداش حالت $\overline{o}_i$ را به‌روز می‌کند.

$$\varepsilon_i^T(n)\overline{o}_i\varepsilon_i(n) = u_{ei}^{*T}(n)R_{ii}u_{ei}^*(n) - 2u_{ei}^{*T}(n)R_{ii}u_i(n) - \sum_{j\in N_i} u_j^T(n)R_{ij}u_j(n) + V_i(\varepsilon_i(n)) - V_i(\varepsilon_i(n+1)) \quad (۲۳)$$

الگوریتم ۱: الگوریتم یادگیری تقویتی معکوس مبتنی بر رویداد برای بازی‌های گرافی رهبر-پیرو چند عاملی

۱. (مقداردهی اولیه) سیاست‌های کنترل، ارزش‌ها و پارامترهای شاخص عملکردی برای همه عامل‌ها بطور دلخواه مقدار دهی می‌شوند.

۲. اجرای حلقه خبره (l تکرار)

- (ارزیابی سیاست خبره) حل برای V_{ei}^{l+1} برای $n \in [n_m^i, n_{m+1}^i)$ با معادله زیر

$$V_{ei}^{l+1}(\varepsilon_{ei}(n)) = U_{ei}(\varepsilon_{ei}(n),u_{ei}^l(n_m^i),u_{-ei}^l(n_m^i)) + V_{ei}^l(\varepsilon_{ei}(n+1)) \quad (۲۴)$$

- (بهبود سیاست خبره) به‌روز رسانی سیاست‌های کنترل با معادله زیر

$$u_{ei}^{l+1}(n) = (q_i+g_i)R_{eii}^{-1}B_i^T\nabla V_{ei}(\varepsilon_{ei}\left(n_m^i+1\right))^{l+1} \quad (۲۵)$$

- برای همه‌ی عامل‌های خبره، زمانی که $\left\|V_{ei}^{l+1}(\varepsilon_{ei}) - V_{ei}^l(\varepsilon_{ei})\right\| \le \sigma$ پایان می‌یابد. σ یک ثابت کوچک است و l شاخص تکرار را نشان می‌دهد.

۳. اجرای حلقه خارجی(h تکرار)

۴.اجرای حلقه داخلی(t تکرار)

برای h داده شده و $t=0$ و با استفاده از u_i^{h0}

- (ارزیابی سیاست یادگیرنده) حل برای $V_i^{h(t+1)}$ برای $n \in [n_m^i, n_{m+1}^i)$ با معادله زیر

$$V_i^{h(t+1)}(\varepsilon_i(n)) = V_i^{ht}(\varepsilon_i(n+1)) + \left[\begin{array}{l}\varepsilon_i^T(n)o_i^h\varepsilon_i(n) + \left(u_i^{ht}(n_m^i)\right)^T R_{ii}u_i^{ht}(n_m^i) \\ + \sum_{j\in N_i}\left(u_j^{ht}(n_m^j)\right)^T R_{ij}u_j^{ht}(n_m^j)\end{array}\right] \quad (۲۶)$$

- (بهبود سیاست یادگیرنده) به‌روزرسانی سیاست‌های کنترل با معادله زیر

$$u_i^{h(t+1)}(n) = (q_i+g_i)R_i^{-1}B_i^T\nabla V_i(\varepsilon_i\left(n_m^i+1\right))^{h(t+1)} \quad (۲۷)$$

- برای همه‌ی عامل‌های یادگیرنده، زمانی که $\left\|V_i^{h(t+1)}(\varepsilon_i) - V_i^{ht}(\varepsilon_i)\right\| \le \sigma$ الگوریتم پایان می‌یابد. σ یک ثابت کوچک است و t و h شاخص تکرار را نشان می‌دهند.

به‌روز رسانی وزن‌های پاداش با کنترل بهینه معکوس: به‌روزرسانی وزن پاداش o_i^h با مشاهده مسیر ورودی کنترلی خبره

$$\varepsilon_i^T(n)o_i^{h+1}\varepsilon_i(n) = u_{ei}^{*T}(n)R_{ii}u_{ei}^*(n) - 2u_{ei}^{*T}(n)R_{ii}u_i^h(n) - \sum_{j\in N_i}\left(u_j^h(n)\right)^T R_{ij}u_j^h(n) + V_i^h(\varepsilon_i(n)) - V_i^h(\varepsilon_i(n+1)) \quad (۲۸)$$

برای همه‌ی عامل‌های یادگیرنده، زمانی که $\left\|o_i^{h+1} - o_i^h\right\| \le \delta$ الگوریتم پایان می‌یابد، در غیراینصورت $u_i^{(h+1)0}(n) = u_i^h(n)$ ، $h \leftarrow h+1$ و رفتن به مرحله ۴

قضیه ۳. (همگرایی الگوریتم ۱) با توجه به فرض‌های ۲ و ۳، الگوریتم ۱ را برای حل مسئله بازی‌های گرافی رهبر-پیرو چندعاملی در نظر بگیرید. $o_i^0 > 0$ ، $R_{ii} > 0$ و $R_{ij} > 0$ را انتخاب کنید. سپس، عامل یادگیرنده i در تعداد محدودی از تکرار h همگرا می‌شود، یعنی $(s_i,u_i^h) \to (s_{ei},u_{ei}^*)$ که u_i^h و u_{ei}^* به ترتیب با (۲۷) و (۱۷) داده شده است. علاوه بر این، o_i^h به $\overline{o}_i$ همگرا می‌شود که $\overline{o}_i$ معادل با پاداش حالت بهینه خبره o_{ie}^* است.

اثبات. در پیوست آورده شده است.

۴- اجرای الگوریتم یادگیری تقویتی معکوس مبتنی بر رویداد برای بازی‌های گرافی رهبر-پیرو ناشناخته

در این قسمت یک چارچوب نقاد-عملگر-پاداش حالت-شناساگر برای اجرای الگوریتم ۱ توسعه داده شده‌است. شبکه‌های نقاد برای اجرای ارزیابی سیاست (۲۴) و (۲۶) و تقریب تابع ارزش‌های بهینه برای عامل‌های خبره و یادگیرنده ساخته شده‌اند. تقریب زننده‌های عملگر برای اجرای بهبود سیاست (۲۵) و (۲۷) ساخته شده‌اند که سیاست کنترل بهینه عامل‌های خبره و یادگیرنده را تخمین می‌زنند. یادگیری وزن‌های ناشناخته پاداش حالت خبره و تخمین آن‌ها برای تابع ارزش یادگیرنده با شبکه عصبی پاداش حالت انجام می‌شود. علاوه بر این، با استفاده از دو شبکه عصبی، دینامیک‌های ناشناخته عامل‌های خبره و یادگیرنده تقریب زده می‌شود.

۴-۱- تقریب زننده‌های نقاد-عملگر-پاداش حالت-شناساگر و تنظیم آن‌ها

در این مطالعه، دینامیک عامل‌های خبره و یادگیرنده ناشناخته فرض شده‌است و دو شبکه عصبی شناساگر برای تقریب دینامیک‌های ناشناخته سیستم‌های چند عاملی (۱) و (۲) بصورت زیر اعمال شده است:

$$\hat{s}_i(n+1) = \hat{W}_{is}^T\varphi_{is}(n) \quad (۲۹)$$

که $\hat{s}_i$ تخمین حالت سیستم و $\hat{W}_{is}$ وزن شناساگر است.

خطای تقریبی شناسایی سیستم برای عامل‌های یادگیرنده بصورت زیر تعریف می‌شود:

$$e_{is} = \hat{s}_i(n+1) - s_i(n+1) = \hat{W}_{is}^T\varphi_{is}(n) - s_i(n+1) \quad (۳۰)$$

خطای تقریبی مربع برابر است با:

$$E_{is}=\frac{1}{2}(e_{is})^T e_{is} \quad (۳۱)$$

با استفاده از قاعده گرادیان نزولی و حداقل شدن (۳۱)، بروزرسانی وزن‌های شناساگر یادگیرنده بصورت زیر به‌دست می‌آیند:

$$\hat{W}_{is}^{(l+1)T}=\hat{W}_{is}^{lT}-\mu_{is}(\hat{W}_{is}^{lT}\phi_{is}(n)-s_i^l(n+1))\phi_{is}^T(n) \quad (۳۲)$$

که $0<\mu_{is}<1$ نرخ یادگیری شبکه شناساگر و l شاخص تکرار را نشان می‌دهند. شناسایی عامل‌های خبره نیز به همین ترتیب انجام می‌شود.

در ادامه، یک تقریب زننده عملگر $\hat{u}_i(.|\hat{W}_{ia})$ برای تخمین سیاست کنترلی و تقریب زننده نقاد $\hat{V}_i(.|\tilde{W}_{ic})$ برای تخمین تابع ارزش بهینه هر عامل استفاده شده‌است. همچنین، وزن‌های پاداش حالت خبره برای تابع ارزش یادگیرنده با شبکه عصبی پاداش حالت $\hat{O}_i(.|\tilde{W}_{iq})$ تقریب زده می‌شود.

$$\hat{u}_i(\hat{W}_{ia})=\hat{W}_{ia}^T\varphi_{iu}(n) \quad (۳۳)$$

$$\hat{V}_i(\hat{W}_{ic})=\varphi_{ic}^T(n)\hat{W}_{ic}^T\varphi_{ic}(n) \quad (۳۴)$$

$$\hat{O}_i(\hat{W}_{iq})=\hat{W}_{iq}^T\varphi_{iq}(n) \quad (۳۵)$$

$\hat{W}_{ia}\in\mathbb{R}^{kN_{i,j}\times m_i}$ ، $\hat{W}_{ic}\in\mathbb{R}^{kN_{i,j}\times nN_{i,j}}$ و $\hat{W}_{iq}\in\mathbb{R}^{kN_{i,j}}$ به ترتیب وزن‌های عملگر، نقاد و پاداش حالت هستند. φ_{iu} ، φ_{ic} و φ_{iq} به ترتیب توابع فعال‌سازی عملگر، نقاد و پاداش حالت هستند. فرض شده است که همه ی توابع فعال‌سازی کراندار هستند. تابع فعال‌سازی هر عامل برداری از خطای ردیابی خودش و عامل‌های همسایه آن می‌باشد. تعداد هر عامل i و همسایه‌های آن با $N_{i,j}$ نشان داده می‌شود.

خطای تقریبی شبکه عملگر برای عامل‌های یادگیرنده و $n\in[n_m^i,n_{m+1}^i)$ به‌صورت زیر تعریف می‌شود:

$$e_{ia}=\hat{u}_i(n_m^i)-\tilde{u}_{i(n_m^i)}=\hat{W}_{ia}^T\varphi_{iu}(n_m^i)-\tilde{u}_i(n_m^i) \quad (۳۶)$$

ورودی کنترلی بصورت زیر داده شده‌است:

$$\tilde{u}_i(n_m^i)=(q_i+g_i)R_{ii}^{-1}\hat{B}_i^T\nabla\hat{V}_i^*\left(\varepsilon_i\left(n_m^i+1\right)\right) \quad (۳۷)$$

با استفاده از (۳۴)، (۳۷) بصورت زیر نوشته می‌شود:

$$\tilde{u}_i(n_m^i)=(q_i+g_i)R_{ii}^{-1}\hat{B}_i^TQ_i\hat{W}_{ic}^T\varphi_{ic} \quad (۳۸)$$

که $Q_i=2\times\left[0....[I]_{ii}.....0\right]\in\mathbb{R}^{k\times kN_{i,j}}$

خطای تقریبی مربع شبکه عملگر یادگیرنده برای ورودی کنترلی بصورت زیر تعریف شده‌است:

$$E_{ai}=\frac{1}{2}(e_{ai})^T e_{ai} \quad (۳۹)$$

قاعده گرادیان نزولی برای بروزرسانی وزن‌های عملگر یادگیرنده اعمال شده است:

$$\hat{W}_{ia}^{(1+l)T}=\hat{W}_{ia}^{lT}-\mu_{ia}\left(\hat{W}_{ia}^{lT}\varphi_{iu}(n_m^i)-\tilde{u}_i^l(n_m^i)\right)(\varphi_{iu}(n_m^i))^T \quad (۴۰)$$

که $0<\mu_{ia}<1$ نرخ یادگیری شبکه عملگر یادگیرنده را نشان می‌دهد.

یک شبکه عملگر دیگر نیز برای تخمین ورودی‌های کنترلی عامل‌های خبره، مشابه با شبکه عملگر یادگیرنده در نظر می‌گیریم که به‌دلیل مشابه بودن در این‌جا آورده نشده‌است.

تابع ارزش هدف یادگیرنده برای $n\in[n_m^i,n_{m+1}^i)$ بصورت زیر می‌باشد.

$$\tilde{V}_i(n)=\hat{V}_i^l(n+1)+\frac{1}{2}\left(O_i(n)+\hat{u}_i^{lT}(n_m^i)R_{ii}\hat{u}_i^l(n_m^i)+\sum_{j\in N_i}\hat{u}_j^{lT}(n_m^j)R_{ij}\hat{u}_j^l(n_m^j)\right) \quad (۴۱)$$

و خطای تقریبی شبکه نقاد در مرحله l برای عامل‌های یادگیرنده برابر است با:

$$e_{ci}=\tilde{V}_i(n)-\hat{V}_i(n) \quad (۴۲)$$

خطای مربع شبکه نقاد بصورت زیر تعریف شده است:

$$E_{ic}=\frac{1}{2}(e_{ic})^T e_{ic} \quad (۴۳)$$

قاعده گرادیان نزولی برای بروزرسانی وزن‌های نقاد یادگیرنده اعمال شده‌است:

$$\hat{W}_{ic}^{(1+l)T}=\hat{W}_{ic}^{lT}-\mu_{ic}\left(\tilde{V}_i(n_m^i)-\varphi_{ic}^T(n_m^i)\hat{W}_{ic}^{lT}\varphi_{ic}(n_m^i)\right)\times\varphi_{ic}(n_m^i)\varphi_{ic}(n_m^i)^T \quad (۴۴)$$

که $0<\mu_{ic}<1$ نرخ یادگیری شبکه نقاد یادگیرنده را نشان می‌دهد.

یک شبکه نقاد دیگر نیز برای تخمین تابع ارزش بهینه عامل‌های خبره، مشابه با شبکه نقاد یادگیرنده در نظر می‌گیریم که بدلیل تشابه در اینجا آورده نشده است.

با درنظر گرفتن شبکه عصبی عملگر (۳۳) و شبکه عصبی نقاد (۳۴) برای یادگیرنده، پاداش حالت به صورت زیر به‌روز می‌شود:

$$\tilde{O}_i^h(n_m^i)=u_{ei}^{*T}(n_m^i)R_{eii}u_{ei}^*(n_m^i)-2u_{ei}^{*T}(n_m^i)R_{eii}\hat{u}_i^h(n_m^i)-\sum_{j\in N_i}\hat{u}_j^{hT}(n_m^i)R_{ij}\hat{u}_j^h(n_m^i)-\hat{V}_i^h(\varepsilon_i(n_m^i+1))+\hat{V}_i^h(\varepsilon_i(n_m^i)) \quad (۴۵)$$

خطای تقریبی پاداش حالت بصورت زیر تعریف می‌شود:

$$e_{iq}=\tilde{O}_i(n_m^i)-\hat{O}_i(n_m^i) \quad (۴۶)$$

خطای تقریبی مربع شبکه پاداش حالت بصورت زیر تعریف شده‌است:

$$E_{iq}=\frac{1}{2}\left(e_{iq}\right)^T e_{iq} \quad (۴۷)$$

قاعده گرادیان نزولی برای بروزرسانی وزن‌های پاداش حالت اعمال شده‌است:

$$\hat{W}_{iq}^{(h+1)T}=\hat{W}_{iq}^{hT}-\mu_{iq}\left(\hat{W}_{iq}^{hT}\varphi_{iq}(n_m^i)-\tilde{O}_{in_m^i}\right)\varphi_{iq}(n_m^i)^T \quad (۴۸)$$

که $0<\mu_{iq}<1$ نرخ یادگیری شبکه پاداش حالت را نشان می‌دهد.

در پایان الگوریتم ۲ برای تنظیم برخط وزن‌های شبکه عصبی نقاد- عملگر- پاداش حالت- شناساگر بازی‌های گرافی زمان گسسته ناشناخته ارائه شده‌است.

الگوریتم ۲: پیاده سازی الگوریتم یادگیری تقویتی معکوس برای بازی‌های گرافی رهبر-پیرو زمان گسسته ناشناخته

۱) مقدار دهی اولیه وزن‌های شبکه عملگر، پاداش حالت و شناساگر بطور تصادفی و وزن‌های شبکه نقاد با صفر برای همه ی عامل‌ها در سیستم خبره و یادگیرنده

۲) مقدار دهی اولیه حالت‌های $x_{ei}(0)$ و $x_i(0)$ برای همه‌ی عامل‌ها ، $x_{e0}(0)$ و $x_0(0)$ برای عامل‌های رهبر بطور تصادفی

۳) مقدار دهی اولیه o_{ei} و o_i

۴) مقدار دهی اولیه u_{ei}^0 و u_i^{00}

۵) اجرای حلقه (l تکرار) (محاسبه u_{ei}^* و V_{ei}^*)

- محاسبه خطای محلی خبره $\varepsilon_{ei}(0)$ روی مسیر سیستم
- محاسبه خطای مبتنی بر رویداد $e_{ei}(n)$ برای عامل‌های خبره با (۱۳)
- محاسبه آستانه e_T برای عامل‌های خبره با (۱۲)
- محاسبه حالت تخمین زده شده $\hat{s}_{ei}^l(n+1)$ با (۲۹)
- اگر برای عامل‌های خبره $\|e_i(n)\| \ge e_T$ آنگاه $l \leftarrow l+1$ و رفتن به مرحله ۵ در غیر اینصورت:
- محاسبه سیاست‌های کنترلی $\hat{u}_{ei}^l$ با (۳۳)
- محاسبه خطای محلی $\varepsilon_{ei}^l(n+1)$ با (۵) و استفاده از حالات تخمین زده شده
- محاسبه تابع ارزش $\hat{V}_{ei}^l(n+1)$ با (۳۴)
- بەروزرسانی وزن‌های نقاد خبره

$$\hat{W}_{ic}^{(1+l)T} = \hat{W}_{ic}^{lT} - \mu_{ic}\left(\tilde{V}_{ei}(n_m^i) - \varphi_{ic}^T(n_m^i)\hat{W}_{ic}^{lT}\varphi_{ic}(n_m^i)\right) \times \varphi_{ic}(n_m^i)\varphi_{ic}(n_m^i)^T$$

- بەروزرسانی وزن‌های عملگر خبره

$$\hat{W}_{ia}^{(1+l)T} = \hat{W}_{ia}^{lT} - \mu_{ia}\left(\hat{W}_{ia}^{lT}\varphi_{iu}(n_m^i) - \tilde{u}_{ei}^l(n_m^i)\right)(\varphi_{iu}(n_m^i))^T$$

- بەروزرسانی وزن‌های شناساگر خبره

$$\hat{W}_{is}^{(l+1)T} = \hat{W}_{is}^{lT} - \mu_{is}(\hat{W}_{is}^{lT}\phi_{is} - s_i^l(n_m^i+1))\phi_{is}^T$$

- برای همه عامل‌های خبره، زمانی که $\left\|\hat{V}_{ei}^{l+1}(\varepsilon_{ei}) - \hat{V}_{ei}^l(\varepsilon_{ei})\right\| \le \alpha$ پایان می‌یابد.

۶) اجرای حلقه خارجی(h تکرار)

۷) اجرای حلقه داخلی(t تکرار) (محاسبه u_i^* و V_i^*)

برای h داده شده و $t=0$ و با استفاده از u_i^{h0}

- محاسبه خطای محلی یادگیرنده $\varepsilon_i(0)$ روی مسیر سیستم
- محاسبه خطای مبتنی بر رویداد $e_i(n)$ برای عامل‌های یادگیرنده
- محاسبه آستانه e_T برای عامل‌های یادگیرنده با (۱۲)
- محاسبه حالت تخمین زده شده $\hat{s}_i^{ht}(n+1)$ با (۲۹)
- اگر برای عامل‌های یادگیرنده $\|e_i(n)\| \ge e_T$ آنگاه $t \leftarrow t+1$ و رفتن به مرحله ۷ در غیر این‌صورت:
- محاسبه سیاستهای کنترلی $\hat{u}_i^{ht}$ با (۳۳)
- محاسبه خطای محلی $\varepsilon_i^{ht}(n+1)$ با استفاده از حالات تخمین زده شده
- محاسبه تابع ارزش $\hat{V}_i^{ht}(n+1)$ با (۳۴)
- بەروزرسانی وزن‌های نقاد یادگیرنده

$$\hat{W}_{ic}^{h(t+1)T} = \hat{W}_{ic}^{htT} - \mu_{ic}\left(\tilde{V}_i(n_m^i) - \varphi_{ic}^T(n_m^i)\hat{W}_{ic}^{htT}\varphi_{ic}(n_m^i)\right) \times \varphi_{ic}(n_m^i)\varphi_{ic}(n_m^i)^T$$

- بەروزرسانی وزن‌های عملگر یادگیرنده

$$\hat{W}_{ia}^{h(t+1)T} = \hat{W}_{ia}^{htT} - \mu_{ia}\left(\hat{W}_{ia}^{htT}\varphi_{iu}(n_m^i) - \tilde{u}_i^{ht}(n_m^i)\right)(\varphi_{iu}(n_m^i))^T$$

- بەروز رسانی وزن‌های شناساگر یادگیرنده

$$\hat{W}_{is}^{h(t+1)T} = \hat{W}_{is}^{htT} - \mu_{is}(\hat{W}_{is}^{htT}\phi_{is} - s_i^{ht}(n_m^i+1))\phi_{is}^T$$

- برای همه عامل‌های یادگیرنده،
- زمانی که $\left\|\hat{V}_i^{h(t+1)}(\varepsilon_i) - \hat{V}_i^{ht}(\varepsilon_i)\right\| \le \alpha$ پایان می‌یابد.

۸) بەروزرسانی وزن‌های پاداش خبره برای یادگیرنده

- بەروزرسانی وزن‌های پاداش حالت

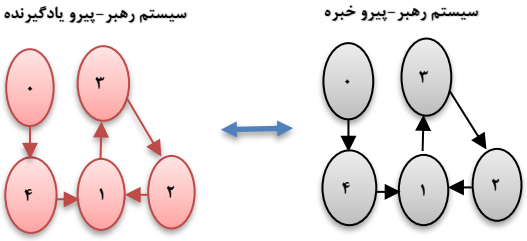
$$\hat{W}_{iq}^{(h+1)T} = \hat{W}_{iq}^{hT} - \mu_{iq}\left(\hat{W}_{iq}^{hT}\varphi_{iq}(n_m^i) - \tilde{O}_i(n_m^i)\right)\varphi_{iq}(n_m^i)^T$$

برای همه‌ی عامل‌های یادگیرنده، زمانی که $\left\|o_i^{h+1} - o_i^h\right\| \le \delta$ الگوریتم پایان می‌یابد، در غیراین‌صورت $u_i^{(h+1)0} = u_i^h$ ، $h \leftarrow h+1$ و رفتن به ۷

در پایان، فلوچارت الگوریتم ۲ در پیوست و در شکل ۱۰ ارائه شده است.

۵- نتایج شبیه سازی

در این بخش برای نمایش عملکرد و صحت روش معرفی شده یک مثال ارایه می‌گردد. ساختار گراف‌های شکل ۱ را در نظر بگیرید.



شکل ۱: توپولوژی تعاملی سیستم‌های چندعاملی خبره و یادگیرنده

دینامیک عامل‌های پیرو و رهبر برای سیستم چندعاملی خبره و یادگیرنده به‌صورت زیر می‌باشد:

$$s_i(n+1) = As_i(n) + B_iu_i(n)\ , \quad s_0(n+1) = As_0$$

$$, s_i = \begin{bmatrix} s_{i1} \\ s_{i2} \end{bmatrix}, i = 0,1,2,3,4$$

$$A=\begin{pmatrix} 0.995 & 0.09983 \\ -0.9983 & 0.995 \end{pmatrix}$$

$$B_1=\begin{bmatrix}0.2047\\0.8984\end{bmatrix}, B_2=\begin{bmatrix}0.2147\\0.2895\end{bmatrix}, B_3=\begin{bmatrix}0.02097\\-0.1897\end{bmatrix}, B_4=\begin{bmatrix}0.2\\0.01\end{bmatrix}$$

وزن‌های اتصال رهبر بصورت $g_1=g_2=g_3=0, g_4=1$ و وزن لبه‌ها بصورت $c_{12}=0.8, c_{14}=0.4, c_{23}=0.6, c_{31}=0.8$ انتخاب شده‌اند. نرخ‌های یادگیری بصورت $\hat{\mu}_{ic}=0.1, \hat{\mu}_{ia}=0.1$ ، $\hat{\mu}_{iq}=0.1$ و $\hat{\mu}_{is}=0.1$ در نظر گرفته شده‌اند.

بر اساس نتایج به‌دست آمده از شبیه سازی، وزن‌های شبکه‌های عصبی همه‌ی عامل‌های خبره و یادگیرنده همگرا می‌شوند، اما برای نمونه فقط نتایج عامل‌های اول آورده شده‌است. شکل ۲ همگرایی وزن‌های شبکه نقاد و عملگر را برای عامل اول خبره و یادگیرنده نشان می‌دهد. شکل ۳ همگرایی وزن‌های شبکه شناساگر برای عامل اول خبره و یادگیرنده و همچنین همگرایی شبکه عصبی پاداش حالت برای عامل اول یادگیرنده را نشان می‌دهد. همان‌طور که مشاهده می‌شود وزن‌های همه‌ی شبکه‌های عصبی همگرا شده‌اند.

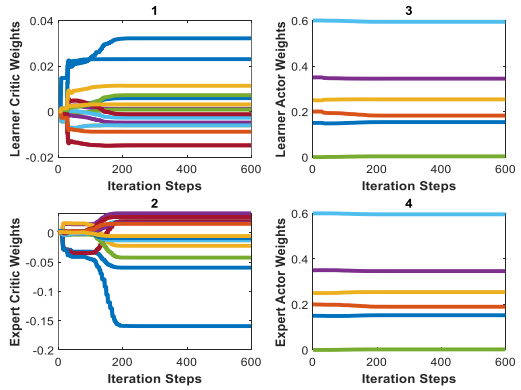
محاسبات برای همه‌ی شبکه‌های عصبی فقط در لحظات رویداد انجام شده‌است و در سایر لحظات ثابت می‌مانند. شکل ۴ خطای محلی برای همه‌ی عامل‌های خبره و یادگیرنده را نشان می‌دهد که همگی به صفر همگرا شده‌اند. همگرایی عامل‌های پیرو به رهبر برای هر دو سیستم خبره و یادگیرنده در شکل ۵ نشان داده شده‌است. با توجه به صفر شدن خطای محلی، هم‌زمانی عامل‌های پیرو به رهبر به‌دست می‌آید. عامل‌های یادگیرنده قصد داشتند که مسیر حالت عامل‌های خبره را دنبال کنند. در شکل ۶، ردیابی حالت‌های عامل اول خبره بوسیله عامل اول یادگیرنده نشان داده شده‌است. سایر عامل‌های یادگیرنده نیز مسیر حالت عامل‌های خبره را دنبال می‌کنند. نتایج نشان می‌دهد که الگوریتم کنترل بهینه توزیع شده پیشنهادی برای حل بازی‌های گرافی زمان گسسته خطی به حل تقریبی بهینه همگرا شده‌است.

ورودی کنترلی عامل اول یادگیرنده با تغییرات وزن‌های ورودی کنترلی در شکل ۷ نشان داده شده است. اثر وزن‌های ورودی کنترلی در نتایج بررسی شد و مشاهده شد که با بزرگ‌تر شدن R ، ورودی کنترلی کوچک‌تر می‌شود ولی زمان بیشتری طول می‌کشد تا به پاسخ بهینه برسیم.

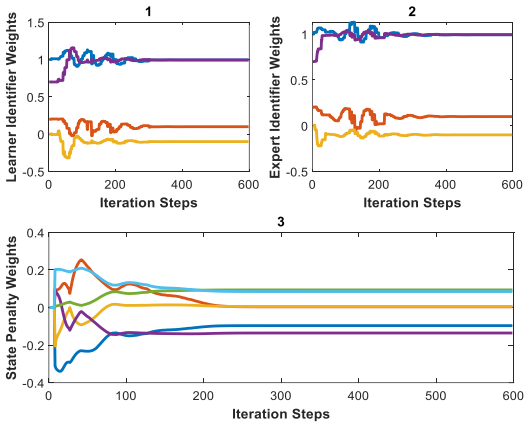
شکل ۸ نشان می‌دهد که روش مبتنی بر رویداد پیشنهادی، نیاز به ۱۵۰ حالت نمونه برداری شده و انجام محاسبات دارد درحالی که در روش مبتنی بر زمان نیاز به ۶۰۰ بار انجام محاسبات است تا به پاسخ‌های بهینه برسیم. بنابراین در روش پیشنهادی بار محاسباتی کمتر است.

در آخر برای مقایسه و تاثیر روش پیشنهادی در عملکرد سیستم، شبیه‌سازی‌ها تحت شرایط اولیه یکسان برای روش مبتنی بر رویداد و مبتنی بر زمان انجام شد. پس از بررسی نتایج، مشاهده شد که سرعت همگرایی وزن‌های شبکه عصبی نقاد و عملگر برای هر دو سیستم خبره و یادگیرنده با هر دو روش تقریبا یکسان است. اما سرعت همگرایی وزن‌های شبکه عصبی شناساگر در روش مبتنی بر رویداد و زمان متفاوت و متاثر از نرخ یادگیری شبکه عصبی شناساگر است. در برخی از نرخ‌های یادگیری، همگرایی وزن‌های شناساگر در روش مبتنی بر رویداد سریعتر انجام می‌شود و در برخی نرخ‌های دیگر بالعکس می‌باشد.

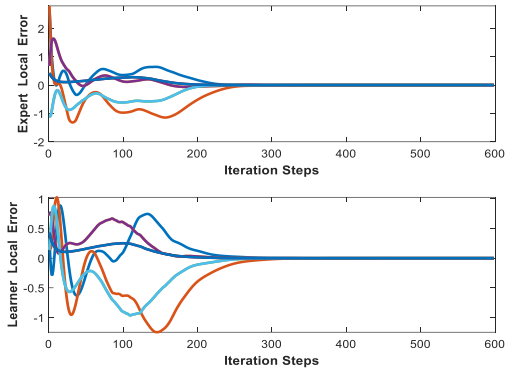
در نرخ‌های یادگیری که شناسایی سیستم روش زمان محور نسبت به رویداد محور دیرتر انجام می‌شود، صفر شدن خطا و همگرایی عامل‌ها به رهبر نیز دیرتر اتفاق می‌افتد. در این حالت ممکن است پاسخ‌ها از مقدار حقیقی دور شوند که عدم محاسبات در برخی تکرارها موجب بهتر شدن نتایج در روش مبتنی بر رویداد می‌شود. شکل ۹ همگرایی وزن‌های شبکه عصبی شناساگر برای سیستم یادگیرنده در نرخ یادگیری ۰.۱ و با دو روش بیان شده را نشان می‌دهد. همگرایی وزن‌های شناساگر در روش مبتنی بر زمان خیلی دیرتر از روش مبتنی بر رویداد اتفاق می‌افتد. بنابراین وقتی شناسایی سیستم دیرتر انجام می‌شود، صفر شدن خطا و همگرایی عامل‌ها به رهبر نیز دیرتر اتفاق می‌افتد. در شکل ۱۰ خطای ردیابی و حالت عامل‌های یادگیرنده نشان داده شده است. همان‌طور که مشاهده می‌شود، در روش مبتنی بر رویداد، خطاهای ردیابی زودتر صفر می‌شوند و در نتیجه عامل‌ها به رهبر در تکرار کمتری همگرا می‌شوند. در روش پیشهادی در تکرار حدود ۲۰۰ و در روش زمان محور در تکرار حدود ۴۰۰ هم‌زمانی به رهبر انجام می‌شود. بنابراین در برخی از نرخ‌های یادگیری شناسایی، سرعت همگرایی روش پیشنهادی بیشتر است.



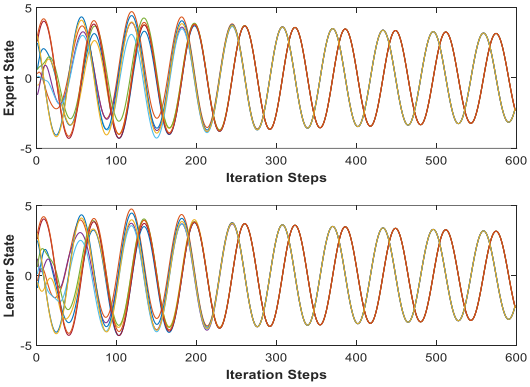
شکل ۲: به‌روزرسانی وزن‌های شبکه نقاد یادگیرنده (۱)، نقاد خبره (۲)، شبکه عملگر یادگیرنده (۳) و عملگر خبره (۴)



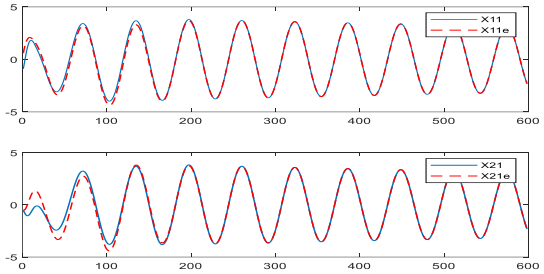
شکل ۳: به‌روزرسانی وزن‌های شبکه شناساگر یادگیرنده (۱) و شناساگر خبره (۲)، به‌روزرسانی وزن‌های شبکه عصبی پاداش حالت برای عامل اول یادگیرنده (۳)



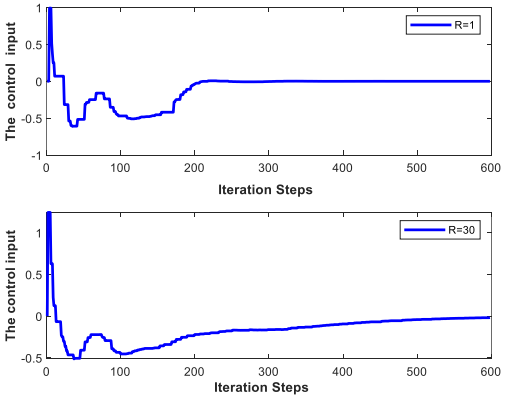
شکل۴ : خطای محلی همه ی عامل‌ها در سیستم‌های چندعاملی خبره و یادگیرنده



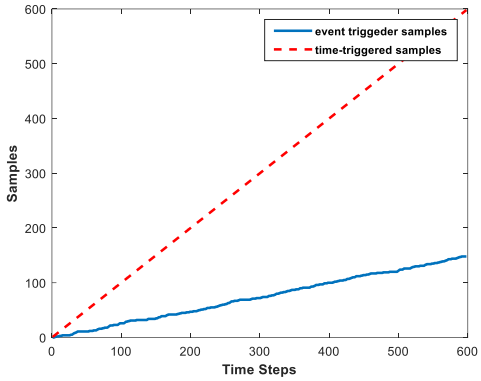
شکل ۵ : هم زمانی عامل‌های خبره و یادگیرنده به رهبر



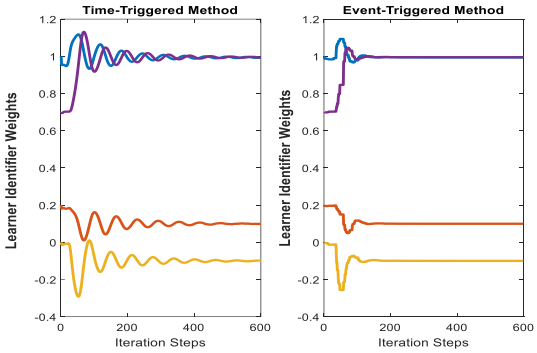
شکل ۶ : ردیابی حالت‌های عامل اول خبره بوسیله عامل اول یادگیرنده



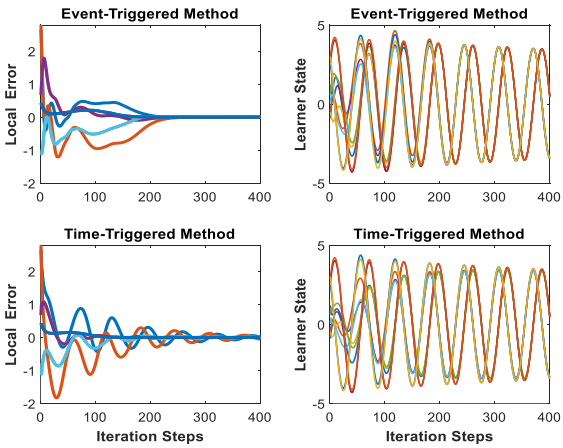
شکل ۷: ورودی کنترلی عامل اول یادگیرنده با تغییرات R



شکل ۸: تعداد نمونه‌ها برای انجام محاسبات در روش‌های مبتنی بر رویداد و زمان



شکل ۹: مقایسه سرعت همگرایی وزن‌های شبکه عصبی شناساگر در روش رویداد محور و زمان محور



شکل ۱۰: مقایسه روش رویداد محور و زمان محور

۶- جمع بندی

در این مقاله، ما یک الگوریتم یادگیری تقویتی معکوس مبتنی بر رویداد برای بازی‌های گرافی زمان گسسته چند عاملی رهبر-پیرو با دینامیک ناشناخته توسعه داده ایم. در الگوریتم ارایه شده، هم‌زمانی بهینه عامل‌های پیرو به رهبر بدست آمده‌است. عامل‌های یادگیرنده نیز با توجه به رفتار عامل‌های خبره، توابع هزینه ناشناخته خبره را بازسازی می‌کنند تا حالت‌های خبره و ورودی‌های کنترلی آن‌ها را تقلید کنند. الگوریتم در ساختار نقاد-عملگر-پاداش حالت اجرا شده‌است تا تابع ارزش بهینه و سیاست‌های کنترل بهینه را برای عامل‌های خبره و یادگیرنده و وزن‌های پاداش خبره را برای یادگیرنده تقریب بزنند.

الگوریتم ارائه شده به هیچ دانشی از دینامیک‌های سیستم نیاز ندارد. برای این هدف شناساگرهای شبکه عصبی اعمال شده‌است تا دینامیک‌های ناشناخته هر عامل را شناسایی کنند. برای کاهش پیچیدگی محاسباتی، سیاست کنترلی و وزن‌های شبکه عصبی فقط در لحظات رویداد به روز شده اند. همچنین پایداری حلقه بسته سیستم خبره نشان داده شده است. نتایج شبیه‌سازی، تایید الگوریتم ارائه شده در هم‌زمانی عامل‌ها به رهبر و ردیابی عامل‌های خبره با یادگیرنده را نشان می‌دهد.

پیوست

اثبات قضیه ۳: مشابه با قضیه ۶ و ۷ در [۱۲]، اثبات همگرایی حلقه خبره در گام ۲ الگوریتم ۱ قابل انجام می‌باشد، که شامل بهبود و ارزیابی سیاست خبره است. در ادامه همگرایی حلقه تکرار داخلی را نشان می‌دهیم، که شامل بهبود و ارزیابی سیاست یادگیرنده در گام ۴ الگوریتم ۱ است. همانطور که در [۲۷]، [۲۸] نشان داده شده است، برای یک تکرار خارجی ثابت h، اگر $o_i^h \geq 0$ و ورودی کنترل پایدار اولیه u_i^{h0} برای هر عامل یادگیرنده i باشد، حلقه تکرار داخلی مجموعه پاسخ همگرای (V_i^h, u_i^h) را برای $n \to \infty$ می‌دهد که V_i^h بهینه است.

زمانی که حلقه تکرار داخلی همگرا می‌شود، حلقه تکرار خارجی شروع می‌شود، بنابراین همگرایی هر حلقه تکرار خارجی h بر مبنای حلقه تکرار داخلی تحلیل شده است. حلقه تکرار داخلی با وزن پاداش حالت o_i^h را که دارای پاسخ‌های همگرای u_i^h و V_i^h است در نظر بگیرید. سپس:

$$0 = V_i^h(\varepsilon_i(n+1)) - V_i^h(\varepsilon_i(n)) + \left[\begin{matrix} \varepsilon_i^T(n) o_i^h \varepsilon_i(n) + u_i^{hT}(n) R_{ii} u_i^h(n) \\ + \sum_{j \in N_i} u_j^{hT}(n) R_{ij} u_j^h(n) \end{matrix}\right] \quad (۴۹)$$

از معادله (۲۸) وزن پاداش حالت o_i^{h+1} هر عامل یادگیرنده i برابر است با:

$$\begin{aligned} \varepsilon_i^T(n) o_i^{h+1} \varepsilon_i(n) = {} & u_{ei}^{*T}(n) R_{ii} u_{ei}^*(n) - 2u_{ei}^{*T}(n) R_{ii} u_i^h(n) \\ & - \sum_{j \in N_i} u_j^{hT}(n) R_{ij} u_j^h(n) \\ & + V_i^h(\varepsilon_i(n)) - V_i^h(\varepsilon_i(n+1)) \end{aligned} \quad (۵۰)$$

با اضافه کردن معادله (۵۰) به (۴۹) داریم:

$$\begin{aligned} \varepsilon_i^T(n) o_i^{h+1} \varepsilon_i(n) = {} & u_{ei}^{*T}(n) R_{ii} u_{ei}^*(n) - 2u_{ei}^{*T}(n) R_{ii} u_i^h(n) \\ & + \varepsilon_i^T(n) o_i^h \varepsilon_i(n) + u_i^{hT}(n) R_{ii} u_i^h(n) \\ = {} & \left(u_{ei}^*(n) - u_i^h(n)\right)^T R_{ii} \left(u_{ei}^*(n) - u_i^h(n)\right) \\ & + \varepsilon_i^T(n) o_i^h \varepsilon_i(n) \end{aligned} \quad (۵۱)$$

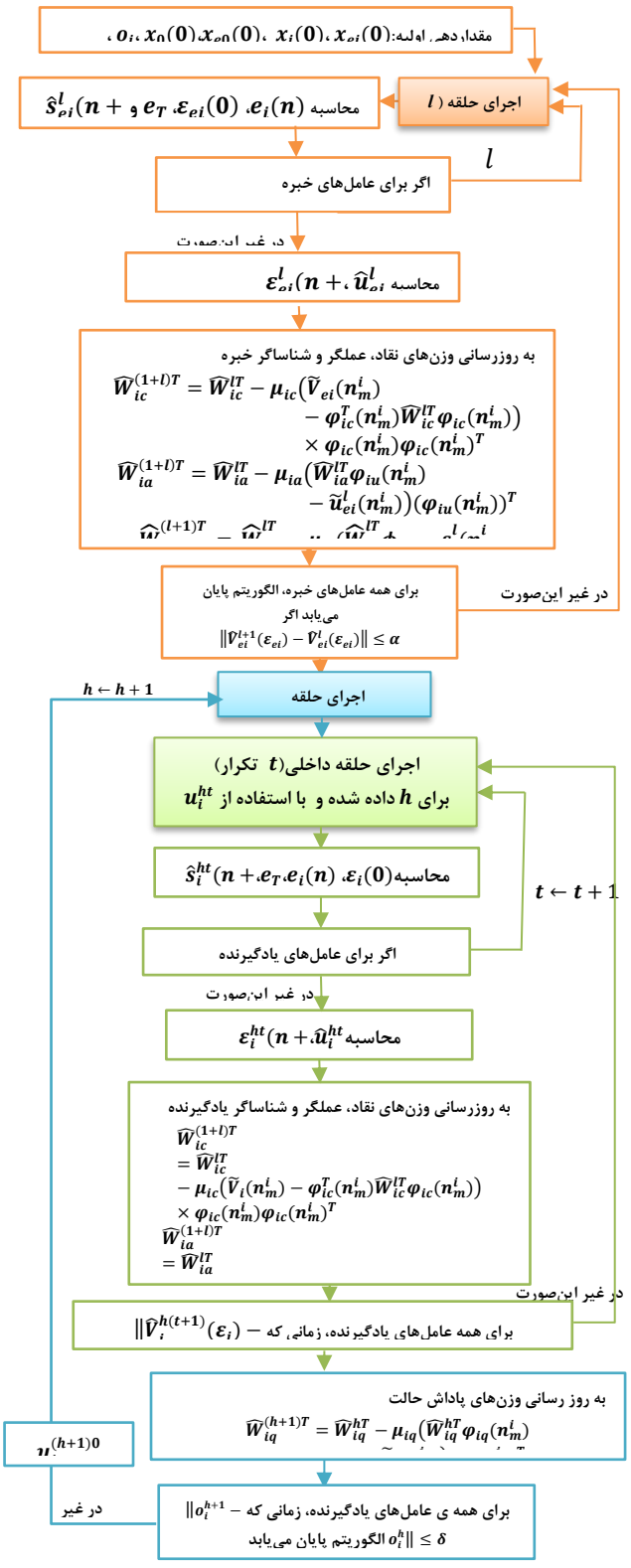
با توجه به $o_i^0 \geq 0$ اولیه، از (۵۱) به این نتیجه می‌رسیم که $o_i^h \geq 0$ برای هر $h = 1, 2, ..$.

زمانی که $h \to \infty$ از معادله (۵۱) داریم:

$$\varepsilon_i^T(n) o_i^\infty \varepsilon_i(n) = \left(u_{ei}^*(n) - u_i^\infty(n)\right)^T R_{ii} \left(u_{ei}^*(n) - u_i^\infty(n)\right) + \varepsilon_i^T(n) o_i^\infty \varepsilon_i(n) \quad (۵۲)$$

بنابراین داریم:

$$u_i^\infty(n) = u_{ei}^*(n), h \to \infty$$



شکل۱۱ : فلوچارت الگوریتم ۲

بنابراین، ورودی‌های کنترل بهینه یادگیرنده u_i^* به ورودی‌های کنترل بهینه خبره u_{ei}^* همگرا می‌شوند. علاوه بر این، از آنجایی که ماتریس‌های حالت و ورودی هر دو سیستم یکسان هستند، حالت‌های یادگیرنده s_i به حالت‌های خبره s_{ei} همگرا می‌شوند.

با توجه به $o_i^0 \geq 0$، از (۵۱) به این نتیجه می‌رسیم که $o_i^h \geq 0$ و $\|o_i^h\|$ برای هر حلقه تکرار خارجی $h = 1, 2, ..$ عامل یادگیرنده i افزایش می‌یابد. بنابراین حداقل یک $\bar{o}_i$ وجود دارد بطوریکه $o_i^0 \leq \bar{o}_i$ و o_i^h می‌تواند به $\bar{o}_i$

نزدیک شود. اگر $o_i^h \to \bar{o}_i$ بطوریکه $\left\| o_i^h - \bar{o}_i \right\| \le \varepsilon_i$ باشد الگوریتم متوقف می‌شود. آستانه ε_i برای پایان دادن به الگوریتم ۱ استفاده می‌شود. بنابراین، الگوریتم ۱ در h متوقف می‌شود. بنابراین اثبات کامل می‌شود.

در پایان فلوچارت الگوریتم ۲ در شکل ۱۱ ارائه شده است.

مراجع

[1] K. Deng, Y. Chen, and C. Belta, ‘‘An approximate dynamic programming approach to multiagent persistent monitoring in stochastic environments with temporal logic constraints’’, IEEE Trans. Autom. Control, vol. 62, no. 9, pp. 4549–4563, 2017.

[2] D. Panagou, D. M. Stipanović, and P. G. Voulgaris, ‘‘Distributed coordination control for multi-robot networks using Lyapunov-like barrier functions,’’ IEEE Trans. Autom. Control, vol. 61, no. 3, pp. 617–632, 2015.

[3] J. Long, W. Wang, J. Huang, J. Lu, and K. Liu, “Adaptive leaderless consensus for uncertain high-order nonlinear multiagent systems with event-triggered communication,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 52, no. 11, pp. 7101–7111, Nov. 2022.

[۴] عرفان شهامت خواه، محمد طباطبایی، «کنترل ردیابی رهبر-پیرو رخداد-تحریک سیستم‌های چندعاملی با دینامیک تک انتگرال‌گیر چندمتغیره»، مجله مهندسی برق دانشگاه تبریز، جلد ۵۰، شماره ۱، صفحات ۲۵۳-۲۶۷، ۱۳۹۹.

[۵] بابک عبدالملکی، علیرضا سیفی، محمدمهدی عارفی، «کنترل ردیابی رهبر-پیرو رخداد-تحریک سیستم‌های چندعاملی با دینامیک تک انتگرال‌گیر چندمتغیره»، مجله مهندسی برق دانشگاه تبریز، جلد ۴۸، شماره ۲، صفحات ۷۷۷-۷۸۴، ۱۳۹۷.

[6] X. Liu, J. Sun, L. Dou, and J. Chen, “Leader-following consensus for discrete-time multi-agent systems with parameter uncertainties based on the event-triggered strategy,” Journal of Systems Science and Complexity, vol. 30, no. 1, pp. 30–45, Feb. 2017.

[7] T. Basar and G.J. Olsder, “Dynamic noncooperative game theory,” Society for Industrial and Applied Mathematics, 1998.

[8] R.S. Sutton, and A.G. Barto, “Reinforcement learning: An introduction,” Robotica, vol. 17, no. 2, pp. 229-235, 1999.

[9] B. Kiumarsi, H. Modares, and F. Lewis, “Reinforcement learning for distributed control and multi-player games”, In Handbook of Reinforcement Learning and Control. Springer, Cham, pp. 7-27, 2021.

[10] K. G. Vamvoudakis, F. L. Lewis, and G. Hudas, “Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality,” Automatica, vol. 48, no. 8, pp. 1598–1611, Aug. 2012.

[11] F. Tatari, M.B. Naghibi-Sistani, and K.G. Vamvoudakis, “Distributed optimal synchronization control of linear networked systems under unknown dynamics”, American Control Conference(ACC), pp. 668-673, 2017.

[12] M. Abouheaf, F. L. Lewis, K. G. Vamvoudakis, Sofie Haesaert, and R. Babuska, “Multi-agent discrete-time graphical games and reinforcement learning solutions,” Automatica, vol. 50, no. 12, pp. 3038–3053, Dec. 2014.

[13] M. I. Abouheaf, F. L. Lewis, M. S. Mahmoud, and D. G. Mikulski, “Discrete-time dynamic graphical games: model-free reinforcement learning solution,” Control Theory and Technology, vol. 13, no. 1, pp. 55–69, Feb. 2015.

[14] S. Arora, & D. Prashant, “A survey of inverse reinforcement learning: Challenges, methods and progress”, Artificial Intelligence, vol. 297, 2021.

[15] X. Wang and D. Klabjan, “Competitive multi-agent inverse reinforcement learning with sub-optimal demonstrations,” In International Conference on Machine Learning, pp. 5143–5151, 2018.

[16] L. Yu, J. Song, and Stefano Ermon, “Multi-agent adversarial inverse reinforcement learning,” In International conference on machine learning , pp. 7194–7201, May 2019.

[17] C. Mu, K. Wang, Z. Ni, and C. Sun, “Cooperative differential game based optimal control and its application to power systems,” IEEE Trans. Ind. Informat., vol. 16, no. 8, pp. 5169–5179, Aug. 2020.

[18] B. Lian, W. Xue, F. L. Lewis, & T. Chai, “Inverse reinforcement learning for multi-player noncooperative apprentice games”, Automatica, vol. 145, pp. 110524, 2022.

[19] B. Lian, V. S. Donge, F. L. Lewis, T. Chai, and A. Davoudi, “Data-Driven Inverse Reinforcement Learning Control for Linear Multiplayer Games,” IEEE Transactions on Neural Networks and Learning Systems, pp. 1–14, 2022.

[20] V. S. Donge, B. Lian, F. L. Lewis, & A. Davoudi, “Multi-agent graphical games with inverse reinforcement learning”, IEEE Transactions on Control of Network Systems, vol. 10, no. 2, pp. 841 – 852, 2022.

[21] X. Li, Y. Tang, & H. R. Karimi, “Consensus of multi-agent systems via fully distributed event-triggered control”, Automatica, vol. 116, pp. 108898, 2020.

[22] X. Li, Z. Sun, Y. Tang, and Hamid Reza Karimi, “Adaptive event-triggered consensus of multiagent systems on directed graphs,” IEEE Transactions on Automatic Control, vol. 66, no. 4, pp. 1670–1685, Apr. 2021.

[23] S. Hu, D. Yue, X. Yin, X. Xie, and Y. Ma, “Adaptive event-triggered control for nonlinear discrete-time systems,” International Journal of Robust and Nonlinear Control, vol. 26, no. 18, pp. 4104–4125, Apr. 2016.

[24] L. Dong, X. Zhong, C. Sun, and H. He, “Adaptive event-triggered control based on heuristic dynamic programming for nonlinear discrete-time systems,” IEEE transactions on neural networks and learning systems , vol. 28, no. 7, pp. 1594–1605, Jul. 2017.

[25] W. Zhao, W. Yu, and H. Zhang, “Event-triggered optimal consensus tracking control for multi-agent systems with unknown internal states and disturbances,” Nonlinear Analysis: Hybrid Systems, vol. 33, pp. 227–248, Aug. 2019.

[26] S. Khoo, L. Xie, and Z. Man, “Robust finite-time consensus tracking algorithm for multirobot systems,” IEEE Transactions on Mechatronics, vol. 14, no. 2, pp. 219–228, 2009.

[27] H. Modares, F. L. Lewis and Z. Jiang, “H_∞ Tracking Control of Completely Unknown Continuous-Time Systems via Off-Policy Reinforcement Learning,” IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 10, pp. 2550-2562, Oct. 2015.

[28] Warren B. Powell, Approximate Dynamic Programming: Solving the Curses of Dimensionality. vol. 703. Hoboken, NJ: John Wiley & Sons, Mar. 2007.