فصلنامه علمی - پژوهشی فرماندهی و کنترل، سال هفتم، شماره دو، تابستان ۱۴۰۲

بهبود جستجوی یادگیری تقویتی عمیق با بهینه سازی مستعمره مورچه

محمد حسن نتاج صلحدار[۱]

تاریخ دریافت: ۱۴۰۲/۰۲/۲۴ تاریخ پذیرش: ۱۴۰۲/۰۵/۲۱

چکیده

ظهور اکوسیستم عظیم اینترنت اشیا[۲] (**IoT**) در حال تغییر سبک زندگی انسان است. اینترنت اشیا هنوز هم متکی به کمک های انسانی است و زمان پاسخ دهی غیرقابل قبول برای بررسی داده های بزرگ دارد و همچنان با چالش های قابل توجهی روبرو هستند. بنابراین، ایجاد چارچوب و الگوریتم جدید برای حل مشکلات خاص اینترنت اشیا سریع، بسیار ضروری است. رویکردهای یادگیری تقویتی و یادگیری تقویتی عمیق[۳] (**DRL**) توانایی تصمیم گیری را دارند، اما روشهای مدلسازی و آموزش سنتی، وقت گیر بوده و کاربردهای آنها را محدود می کنند. این مقاله برای غلبه بر این معضل، روش یادگیری تقویتی متناسب با اینترنت اشیا را پیشنهاد می کند. به این صورت که یک روش انتخاب ویژگی مبتنی بر بهینه سازی مستعمره مورچه[۴] (**ACO**) پیشنهاد می کنیم. از آنجا که توابع اکتشافی بر روند تصمیم گیری **ACO** در طی فرآیند جستجو تأثیر می گذارد، استفاده از روش یادگیری ابتکاری می تواند به الگوریتم کمک کند تا در فضای جستجو بهتر جستجو کند. سرانجام، به عنوان مطالعه موردی اینترنت اشیا، روش پیشنهادی برای کنترل چراغ راهنمایی، با هدف کاهش ازدحام ترافیک در تقاطع های شهرهای هوشمند، اعمال می شود. نتایج تجربی نشان می دهد که روش پیشنهادی می تواند در مقایسه با رویکردهای سنتی، اقدامات بهتری را در زمان اجرای کوتاهتر بیاموزد.

واژگان کلیدی: یادگیری تقویتی عمیق، اینترنت اشیاء، بهینه سازی مستعمره مورچه

[۱] دانشگاه شهید چمران اهواز_پردیس صنعتی شهدای هویزه، اهواز، ایران. مربی. n.solhdar@scu.ac.ir
نویسنده مسئول : محمد حسن نتاج صلحدار

[۲] Internet of Things
[۳] Deep Reinforcement Learning
[۴] Ant Colony Optimization

[Downloaded from ic4i-journal.ir on 2026-02-17]

مقدمه

امروزه، ما شاهد شکل گیری یک اکوسیستم عظیم اینترنت اشیا (**IoT**) هستیم. اینترنت اشیا به اتصال انواع دستگاه های مجهز به بی سیم اعم از تلفن های هوشمند، پوشیدنی و امکانات واقعیت مجازی گرفته تا سنسورها، هواپیماهای بدون سرنشین و وسایل نقلیه گفته می شود [۱]. این مورد در بسیاری از حوزه ها مانند مراقبت های بهداشتی، نظارت، حمل و نقل، خدمات عمومی، خدمات دولتی مورد استفاده قرار گرفته و خواهد گرفت که انتظار می رود سبک زندگی هوشمندتر، ایمن تر و راحت تری را برای ما فراهم کنند[۲, ۳].

اگرچه اینترنت اشیا بر زندگی ما تأثیر گذاشته یا حتی بر آن تسلط خواهد داشت، اما سیستم های اینترنت اشیا فعلی وقتی برای اجرای برخی وظایف در سناریوهای عملی استفاده می شوند، هنوز با چالش های مهمی روبرو هستند. اولاً، سیستمهای **IoT** فعلی برای کارهای کنترلی و مدیریتی معمولاً دستگاههای سنجش از قبل مستقر شده و طرحهای کنترل هوشمند از پیش طراحی شده را در نظر می گیرند. عملکرد آنها هنگامی خراب خواهد شد که بدون دخالت انسان کار کنند و با حوادث غیرمجاز مواجه شوند. بنابراین، باید به اینترنت اشیایی با توانایی لازم که اتکا به کمک های انسانی را به حداقل برسانند، توجه کافی شود [٤]. ثانیاً، اگرچه میلیون ها دستگاه اینترنت اشیا حجم زیادی از داده ها را جمع آوری می کنند، اما سیستم های اینترنت اشیا ممکن است قادر به تجزیه و تحلیل و پردازش داده های بزرگ در مدت زمان کوتاه نباشند. به عبارت دیگر، مدل سازی، پاسخ و زمان تصمیم گیری اینترنت اشیا نمی تواند نیازهای تأخیر شدید چندین برنامه را برآورده کند.

چندین برنامه مهم، مانند کنترل هوشمند ترافیک در حمل و نقل، توانبخشی از راه دور در مراقبت های بهداشتی، رویداد خطرناک هشدار دهنده در صنعت و غیره، نیازهای دقیق تر و بالاتر برای اینترنت اشیا را مطرح می کنند. یعنی سیستم های **IoT** نوظهور نه تنها باید دارای توانایی کنترل و مدیریت خودمختار باشند بلکه زمان پاسخگویی کم و سرعت پردازش سریع را نیز تضمین می کنند. بنابراین، ایجاد چارچوب و الگوریتم جدید برای حل مشکلات خاص **IoT** در حال ظهور، بسیار فوری و معنی دار است. این امر به توسعه عمیق اینترنت اشیا و همچنین ارائه خدمات انسانی با کیفیت بهتر کمک می کند [۵].

هنگام رسیدگی به مسئله کنترل و مدیریت، معمولاً یادگیری تقویتی مورد توجه قرار گرفته است [٦]. یادگیری تقویتی یک رویکرد محاسباتی است که در آن یک عامل با یک محیط نامشخص و پیچیده ارتباط برقرار می کند، با تلاش برای به حداکثر رساندن کل مبلغ پاداش دریافت شده، یک سیاست مطلوب را می آموزد. اگرچه رویکردهای سنتی یادگیری تقویتی، مانند **Q-learning**، در چندین سناریوی عملی به موفقیت رسیده اند، اما کاربرد آنها محدود به حوزه هایی با ورودی‌های با بعد کم محدود شده است. برای عبور از این معضل، یادگیری تقویتی عمیق، ترکیبی از فناوری یادگیری عمیق و معماری یادگیری تقویتی، طراحی شده است [۷]. به طور خاص، در **DRL**، شبکه های عصبی عمیق می توانند برای تقریبی غیر خطی تابع ارزش_عمل[۱] از ارزش عمل‌ها هنگامی که تعداد حالات (بعد ورودی ها) زیاد است، اتخاذ شوند.

به عبارت دیگر، روش یادگیری عمیق عوامل مبتنی بر یادگیری تقویتی را قادر می سازد تا از مشکلات تصمیم گیری با فضاهای حالت و عمل[۲] با ابعاد بالا استفاده کنند. **DRL** به طور گسترده در زمینه های بینایی ماشین، رباتیک، ارتباطات و موارد دیگر مورد استفاده قرار گرفته است [۸]. اخیراً روش‌های **DRL**، برای حل مشکلات

[۱] action-value function
[۲] action

[Downloaded from ic4i-journal.ir on 2026-02-17]

خاص سیستم‌های نوظهور اینترنت اشیا[۹, ۱۰] آغاز شده است.

خواسته دیگر اینترنت اشیا، زمان پاسخگویی پایین و سرعت پردازش سریع است. اگرچه **DRL** توانایی یادگیری بازنمایی[۱] و تصمیم‌گیری[۲] دارد، اما در صورت استفاده از برنامه های اینترنت اشیا، کارایی آن تضمین نمی شود. دو دلیل وجود دارد، در مرحله اول، **DRL** برای مدل سازی مشخصه غیرخطی تابع ارزش_عمل به لایه های زیادی نیاز دارد. این ساختار پیچیده با تعداد زیادی پارامتر ممکن است به یک روش برآورد وقت گیر منجر شود. ثانیا، برای دستیابی به دقت بالاتر، باید لایه های زیادی برای شبکه های عصبی عمیق در **DRL** روی هم قرار بگیرند.

معمولا داده های ورودی از تعداد زیادی ویژگی تشکیل شده است که شامل ویژگی های بی ربط و زائد است. این ویژگی ها الگوریتم های یادگیری را کند کرده و بعضاً عملکرد آنها را کاهش می دهد. در نتیجه، انتخاب ویژگی۳ **(FS)** [۱۱, ۱۲] به مرحله اساسی پیش پردازش برای الگوریتم های یادگیری ماشین و داده کاوی تبدیل می شود. هدف اصلی روشهای انتخاب ویژگی، کنار آمدن با نفرین بعد٤ است. یعنی تلاش می کند زیرمجموعه کوچکی از ویژگی ها را با افزونگی کم و ارتباط زیاد پیدا کند که نشان دهنده مجموعه داده و همچنین مجموعه اصلی ویژگی ها باشد.

در این مقاله، ما از یادگیری تقویتی برای پاسخگویی به نیازهای **IoT** استفاده کردیم.این چارچوب به عنوان یادگیری تقویتی عمیق تعریف می‌شود که از پنج مولفه اصلی تشکیل شده است: محیط ، پایگاه تجربه، مجموعه آموزش، ارزیابی و هدف. اصل طراحی **DRL** پیشنهادی، کاهش بیشتر پیچیدگی ساختار وکاهش زمان آموزش است و همچنین تضمین صحت مدل سازی و تصمیم-گیری و تأمین نیازهای **IoT** است. به طور خلاصه، بخش‌های این مقاله به شرح زیر است:

i) چارچوب **DRL** پیشنهادی را ارائه می دهیم، که یک روش جایگزین برای **DRL** است.

ii) چارچوب آماده سازی نمونه آموزشی و سازوکار انتخاب ویژگی برای **DRL** پیشنهادی را مشخص میکنیم. این سازوکارها می تواند باعث افزایش کارایی پردازش و دقت تصمیم گیری مستقل و همچنین کاهش هزینه محاسبات شود.

iii) ما **DRL** پیشنهادی را برای رسیدگی به موضوعات خاص **IoT**، یعنی کنترل چراغ راهنمایی در شهرهای هوشمند، اتخاذ می کنیم. نتایج شبیه سازی نشان می دهد که روش **DRL** پیشنهادی می تواند در مقایسه با رویکردهای دیگر، اقدام بهتری را در زمان اجرای کوتاهتر بیاموزد.

کارهای مرتبط

DRL آماده است تا انقلابی در زمینه هوش مصنوعی ایجاد کند [۷]. این نشان دهنده گامی به سوی ایجاد سیستم های خودمختار است که با محیط خود در تعامل هستند تا رفتارهای بهینه را یاد بگیرند و با گذشت زمان از طریق آزمون و خطا بهبود می یابند. در حال حاضر، **DRL** در دامنه های مختلف برنامه اعمال شده است.

در سناریوی رایانش ابری یا لبه، **DRL** برای مدیریت منابع و انرژی استفاده می شود. در [۱۳]، یک الگوریتم **DRL** بدون مدل برای مدیریت منابع در سناریوی محاسبات لبه ای پیشنهاد شد. هدف این بود که الگوی تحرک کاربر به طور خودکار کشف شود و بر این اساس خدمات بین سرورهای لبه را منتقل کنیم. در [١٤]، یک معماری مدیریت منابع ابری هوشمند مبتنی بر **DRL** ارائه شد. با استفاده از **DRL**، ابرها می توانند مناسب ترین پیکربندی را از یک محیط پیچیده ابر پیاده سازی کنند. محلی سازی و ناوبری با دقت بالا سناریوی کاربردی

[۱] representation
[۲] decision-making
[۳] Feature Selection
[٤] curse of dimensionality

[Downloaded from ic4i-journal.ir on 2026-02-17]

دیگر است. در [۱۵]، مسئله محلی سازی مشارکتی وسایل نقلیه به عنوان یک فرآیند تصمیم گیری تا حدودی قابل مشاهده در مارکوف مورد استفاده قرار گرفت و با استفاده از الگوریتم های برنامه ریزی غیرمتمرکز با **DRL** حل شد. در [۱٦]، یک روش مبتنی بر **DRL** ارائه شد، که به وسایل نقلیه هوایی بدون سرنشین امکان می دهد به طور خودکار در یک فضای پیچیده در مقیاس بزرگ مجازی حرکت کنند. علاوه بر این، **DRL** با موفقیت برای پایان دادن به کارهایی مانند دستکاری رباتیک، پردازش ویدیو و غیره مورد استفاده قرار گرفت.

برای مسئله مستقل در اینترنت اشیا، در [٤]، سیستمی ساخته شد که می تواند نوع دستگاه را در شبکه اینترنت اشیا به طور موثر شناسایی کند. این ارتباطات شبکه را توسط یک الگوریتم یادگیری بدون نظارت تجزیه و تحلیل می کند، پس از راه اندازی اولیه به طور مستقل عمل می کند. در [۱۷]، نحوه دستیابی به کنترل و پیش بینی باتری برای یک سیستم اینترنت اشیا کوچک مهم بود. برای حل این مشکل، یک شبکه یادگیری تقویتی دو لایه طراحی شد تا همزمان میزان مجموع را به حداکثر برساند و ضرر پیش بینی را به حداقل برساند. در [۱۸]، از **DRL** برای دستیابی بهینه سازی مشترک ذخیره سازی منابع رادیویی و محاسبات برای اینترنت اشیا خودمختار مجهز به **fog** به دلیل فضای بسیار زیاد حالت و عمل استفاده شد.

در [۱۹]، برای استفاده کامل از تعداد زیادی داده سنسوری **IoT** بدون برچسب، یک الگوریتم **DRL** نیمه نظارت شده برای خدمات هوشمند در شهرها طراحی شد. به ویژه، نویسندگان از روش پیشنهادی برای ارتقاء دقت در محلی سازی محیط داخلی در ساختمانهای هوشمند استفاده کردند.

برای انتشار سریع در اینترنت اشیا، در [۲۰]، یک سیستم اینترنت اشیا برای نظارت، بومی سازی و هشدار دادن به کارگرانی که وارد مناطق خطرناک شده اند، طراحی شده است. خواسته های این سیستم به صورت بلادرنگ[۱] با در نظر گرفتن تکنیک های آنتن های جهت دار، فرکانس رادیویی بالا و امواج اولتراسوند[۲] قابل درک است. در [۲۱]، یک چارچوب توزیع شده برای مشبک[۳] های هوشمند پیشنهاد شد، که مدیریت پاسخ و تقاضای بی درنگ را امکان پذیر می کند. در [۲۲]، به منظور رفع موثر نیازهای تأخیر چندین برنامه اینترنت اشیا، گره های مه معرفی شدند و مکانیسم اشتراک زمینه[٤] مجاز بود. در [۲۳]، برای رسیدگی به نیازهای تأخیر دقیق برنامه های **IoT** در زمان واقعی، یک طرح کنترل توزیع شده پیشنهاد شد.

مطالعه‌ای که در سال ۲۰۲٤ منتشر شده است [۲٤]، به بررسی استفاده از **DRL** برای تقویت امنیت و حفظ حریم خصوصی در دستگاه‌های **IoT** پرداخته است. در این مطالعه، از **DRL** برای تشخیص و جلوگیری از تهدیدات امنیتی و حملات سایبری استفاده شده است، که به‌طور خاص در سناریوهای **IoT** با تعداد زیادی دستگاه متصل مفید است.

در [۲۵]، یک روش **DRL** برای مدیریت هوشمند مصرف انرژی در وسایل نقلیه خودمختار معرفی شده است. این روش با در نظر گرفتن وضعیت بلادرنگ شبکه، بهینه‌ترین مسیرها و تنظیمات رانندگی را برای کاهش مصرف انرژی پیشنهاد می‌کند.

بهینه سازی کلنی مورچه ها

بهینه سازی مستعمره مورچه **(ACO)** یک الگوریتم مربوط به هوش انبوه است که توسط **M. Dorigo** در دهه ۱۹۹۰ پیشنهاد شده است. **ACO** به طور گسترده ای برای حل

[۱] real-time
[۲] ultrasound
[۳] grid
[٤] context sharing

[Downloaded from ic4i-journal.ir on 2026-02-17]

مشکلات بهینه سازی ترکیبی استفاده شده است. عوامل مصنوعی در **ACO** بر اساس ایده های ذکر شده در بالا، به ویژه "تقویت کننده های فرمون" و "انتقال احتمالی" کار می کنند. علاوه بر مورچه های واقعی، الگوریتم های **ACO** توانایی های بیشتری به مورچه های مصنوعی اضافه می کنند. آنها حالت های درونی دارند که اقدامات گذشته مورچه ها را برای افزایش کیفیت راه حل ها به خاطر می سپارند. آنها مسیر فرمون را به صورت محلی و جهانی به روز می کنند تا راه حل های بهینه بهتری پیدا کنند. آنها با معرفی توابع اکتشافی برخی از اطلاعات خاص مسئله را اضافه می کنند [۲۶].

الگوریتم های **ACO** معمولاً سه جز[۱] اصلی اطلاعات فرمون، ساخت راه حل[۲] و به روزرسانی فرمون را برای بیان مسئله ارائه می دهند. اطلاعات فرمون مشخص می کند که چگونه اطلاعات مشکل باید از نظر بردار دنباله فرمون نشان داده شود، که کیفیت یک راه حل را اندازه گیری می کند. ساخت راه حل در مورد تعریف قانون تصمیم احتمالی (انتقال وضعیت)، اطلاعات اکتشافی خاص مسئله و محدودیت های مسئله است.

به روزرسانی فرمون نحوه به روزرسانی اطلاعات فرمون را با توجه به راه حل های ارائه شده مشخص می کند [۲۷]. در **ACO**، همه مورچه ها درگیر یک کار تکراری هستند تا یک راه حل برای بهینه سازی مقادیر دنباله فرمون با حرکت در یک توالی متناسب تا رسیدن به معیارهای توقف، ایجاد کنند. بعد از اینکه مورچه ها محلول های خود را به طور کامل ساختند، کسری از فرمون به طور مساوی در همه حالت ها تبخیر می شود. سپس هر مورچه بردار فرمون را با توجه به راه حل های بنیادی خود به روز می کند. شبه کد **ACO** در الگوریتم 1 نشان داده شده است.

Algorithm 1 General pseudo-code of the Ant Colony Optimization.
1. Initializing the pheromone trails
2. Repeat
3. For each ant Do
4. Solution construction by modelling the problem in terms of pheromone vector;
5. Update the pheromone according to the founded solutions:
6. Evaporation
7. Reinforcement
8. Until stopping criteria
9. Output: final pheromone vector, which specifies the best-founded solution

روش پیشنهادی

مدل فرمون، مولفه اصلی الگوریتم های ACO است که برای ساختن راه حل های خوب استفاده می شود. در روش پیشنهادی، هر مورچه یک بردار d-بعدی به نام دنباله فرمون تشکیل می دهد. اطلاعات متقابل بین دو متغیر تصادفی X و Y با فرمول زیر محاسبه می شود:

$$MI(X,Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \quad (1)$$

اگر دو متغیر مقدار اطلاعات متقابل بالاتری داشته باشند، در این صورت بسیار وابسته هستند.

در روش پیشنهادی، در ابتدا، مورچه ها به طور تصادفی بر روی ویژگی ها قرار می گیرند و هر مورچه با بازدید از ویژگی های مختلف در فضای جستجو، با استفاده از هر دو قانون انتقال حالت حریصانه و احتمالی، یک راه حل می سازد. قانون انتقال حالات (state) با استفاده از اطلاعات ابتکاری و فرمونی[۳]، مقدار بین اکتشاف و بهره برداری[۴] را متعادل می کند. در هر مرحله، مورچه k با استفاده از یک قانون انتخاب اقدام احتمالی یا حریصانه از ویژگی بعدی i به شرح زیر بازدید می کند:

$$p_i^k(t) = \begin{cases} \frac{[\tau_i(t)][V_i(t)]^\beta}{\sum_{u\in N^k}[\tau_u(t)][V_u(t)]^\beta}, \forall i \in N^k, if\ q > q_0 \\ 0 \qquad\qquad , \text{otherwise} \end{cases}$$
(2)

$$i = \underset{u\in N^k}{argmax}\{[\tau_u][V_u]^\beta\}, {}_1^nY\ if\ q \le q_0$$
(3)

[۱] components
[۲] Solution
[۳] heuristic and pheromone
[۴] exploration and exploitation

[Downloaded from ic4i-journal.ir on 2026-02-17]

N^k مجموعه ای از ویژگی های عملی بازدید نشده است که مورچه k می تواند بازدید کند. τ مقدار فرمون مرتبط با ویژگی i است، و Vi مطلوبیت ابتکاری است، که در اینجا تابع مقدار حالت مربوط به ویژگی i است. پارامتر β بین [۰.۱] است و trade-of بین مقدار فرمون و اطلاعات ابتکاری را کنترل می کند. یعنی اگر $\beta = 0$ باشد، تأثیر اطلاعات ابتکاری نادیده گرفته می شود و تصمیم گیری فقط بر اساس تاریخچه اقدام قبلی اتفاق می افتد.

پارامتر q یک متغیر تصادفی است، که به صورت یکنواخت در [۰.۱] توزیع می شود. $q_0 \in [۰ ، ۱]$ یک عدد ثابت است. این دو پارامتر اهمیت نسبی بهره برداری در مقابل اکتشاف را مشخص می کنند. اگر $q > q_0$ باشد، هر ویژگی با توجه به احتمال زیاد توسط مورچه یکبار بازدید می شود ، که به معنای اکتشاف است و اگر $q < q_0$ باشد، مورچه با توجه به حداکثر مقدار فرمون و ارزش اکتشافی از بهترین ویژگی بازدید می کند، که منجر به بهره برداری بیشتر می شود.

در اینجا ضریب اکتشاف-بهره برداری q را با استفاده از نرخ تباهی[۱] چند جمله ای تنظیم می کنیم، که با در نظر گرفتن تعداد تکرار، مقدار بین اکتشاف و بهره برداری را کنترل می کند. یعنی مورچه ها در تکرارهای اولیه الگوریتم اکتشاف بیشتری انجام می دهند و در پایان تکرارها بهره برداری بیشتری انجام می دهند.

فرم ریاضی این نرخ فروپاشی چند جمله ای به شرح زیر است:

$$q = (start_{rate} - end_{rate}) \times (1 - k/_{NF})^{power} + end_rat \qquad (4)$$

که در آن $start_{rate}$ و end_{rate} حد پایین و بالایی نرخ تباهی را تعیین می کنند و power سرعت تباهی را کنترل می کند، NF تعداد ویژگی هایی است که هر مورچه باید در هر تکرار مشاهده کند و پارامتر k تعداد فعلی تکرار است.

الگوریتم های ACO معمولاً از اطلاعات ابتکاری ثابت استفاده می کنند. یعنی ابتکار عمل در طول فرآیند جستجو متفاوت نیست. اگر می توان در هر تکرار از تجربه همه مورچه ها، روش ابتکاری را یاد گرفت، به بهبود عملکرد الگوریتم کمک زیادی می کند. در اینجا عوامل (مورچه ها) در حین فرآیند جستجو در الگوریتم ACO ، تابع مقدار بهینه حالت V را به عنوان اطلاعات ابتکاری می آموزند. ما بردار مقدار حالت V را با حداکثر شباهت کسینوس بین هر یک از ویژگی ها و همه برچسب های کلاس مقدار دهی می کنیم و فضای جستجوی ویژگی ها را به صورت MDP مدلسازی می کنیم. خصوصیات مورد نیاز MDP را می توان به صورت زیر برآورده کرد: ویژگی ها نمایانگر حالات محیط هستند، عملکرد نماینده انتخاب ویژگی های بازدید نشده است و عملکرد انتقال براساس "نسبت شبه تصادفی" ACO است.

تابع پاداش از دو معیار نظارت شده و بدون نظارت تشکیل شده است. اولی، ماتریس $I \times d$، به نام flCorr است که با شباهت کسینوس بین ویژگی ها و تمام برچسب های کلاس (ارتباط ویژگی ها) محاسبه می شود. مورد دوم یک ماتریس $d \times d$، به نام fCorr است که با همبستگی بین ویژگی ها (افزونگی ویژگی ها) محاسبه می شود. هم برای شباهت کسینوس و هم برای همبستگی، اگر مقدار نزدیک به صفر باشد، دو بردار مستقل هستند و اگر مقدار به یکی نزدیکتر باشد، بردارها به یکدیگر وابسته هستند.

هنگامی که یک عامل با انجام عمل a از حالت s_t به حالت s_{t+1} حرکت می کند، می توان پاداش r_{t+1} را به شرح زیر محاسبه کرد:

$$(s_t) = V(t) + \alpha\,[\, r_{t+1} + \gamma V\,(s_{t+1}) - V\,(s_t)]$$

$$r_{t+1} = \frac{\max_{s_{t+1}}\left(\mathrm{flCorr}_{s_{t+1},l}\right)}{1+fCorr_{s_t,s_{t+1}}} \qquad (5)$$

که $flCorr_{st+1,l}$، مقدار همبستگی برچسب ویژگی بعدی s_{t+1} و تمام برچسب های مربوط به آن است و $fCorr_{st,st+1}$، همبستگی ویژگی بین ویژگی فعلی s_t و ویژگی بعدی s_{t+1} است. این ترکیب به ویژگیهایی که همبستگی زیادی با برچسب کلاس l دارند و کمترین همبستگی را با ویژگی قبلاً انتخاب شده همزمان دارند، پاداش بیشتری می دهد. در هر تکرار، در حالی که مورچه ها راه حل خود را می سازند، آنها بردار حالت V را به صورت محلی بروز می کنند. سپس در پایان تکرار، بردار V محلی هر مورچه k با میانگین گیری همه آنها در سطح جهانی به روز می شود:

$$V_{S_t} = \frac{1}{nAnt}\sum_{k=1}^{nAnt} V_{S_t}^{k} \qquad (6)$$

که در آن پارامتر nAnt تعداد مورچه هایی است که راه حل ها را می سازند و زیرمجموعه های ویژگی را انتخاب می کنند.

بکارگیری الگوریتم پیشنهادی در کنترل چراغ راهنمایی و رانندگی

۱ decay rate

[Downloaded from ic4i-journal.ir on 2026-02-17]

به عنوان یک مورد استفاده از اینترنت اشیا، کنترل چراغ راهنمایی نقش مهمی در شهرهای هوشمند دارد. می توان آن را به عنوان یک روش موثر برای کاهش تراکم ترافیک در نظر گرفت [۲۸, ۲۹].

در اینجا ، ما روش پیشنهادی را در مورد مشکل کنترل چراغ راهنمایی به نمایش می گذاریم. به عبارت دیگر، DRL برای بهینه سازی چراغ های راهنمایی در تقاطع ها با توجه به جریان ترافیک در زمان واقعی وسایل نقلیه استفاده می شود. به طور خاص، آزمایشات بر روی سیستم عامل شبیه سازی تحرک شهری (SUMO) انجام می شود [۳۰].

SUMO یک شبیه ساز ترافیک منبع باز و چند حالته است که به کاربران امکان می دهد شرایط جاده و مسیرهای خودرو را شخصی سازی کنند. در اینجا، محیط سناریوی برنامه در مورد یک تقاطع چهار طرفه است. چهار خط در هر بازو در تقاطع جمع می شوند. از جهت قطب نما در حالی که چهار خط در هر بازو تقاطع را ترک می کند. هنگامی که وسایل نقلیه به تقاطع نزدیک می شوند، باید خط مورد نظر را با توجه به جهت از قبل انتخاب کنند:

- چرخش به چپ: فقط بیشترین سمت چپ را انتخاب کنید.
- مستقیم بروید: دو خط مرکزی یا بیشترین سمت راست را انتخاب کنید.
- گردش به راست: فقط بیشترین مسیر سمت راست را انتخاب کنید.

قبل از اتخاذ روش پیشنهادی، ما باید عناصر مربوط به محیط ، عامل ، حالات، عملکردها و عملکرد پاداش را تعریف کنیم.

محیط: محیط شرایط تراکم وسایل نقلیه در خطوط تقاطع چهار طرفه است. خطوط مسئول حرکت مستقیم و چرخش راست توسط همان چراغ راهنمایی کنترل می شوند. این بدان معناست که وسایل نقلیه در این سه خط به طور همزمان آزاد می شوند.چرخش سمت چپ اختصاص داده شده به یک خط به طور جداگانه کنترل می شود. بنابراین، هشت چراغ راهنمایی مختلف در محیط وجود دارد. علاوه بر این، هر خط به ده سلول با اندازه های مختلف تقسیم می شود. هرچه سلول از خط توقف دورتر باشد، طولانی تر است. اگر سلول نزدیک خط توقف بیش از حد طولانی تنظیم شده باشد، ممکن است برخی از وسایل نقلیه که به خط توقف نزدیک می شوند شناسایی نشوند و این در دقت مشاهده تأثیر می گذارد.

عامل: خود کنترل چراغ راهنمایی به عنوان عامل نشان داده می شود. عامل با گذشت زمان با محیط ارتباط برقرار می کند.

حالتها:حالت ها برای توصیف اطلاعات محیط در یک مرحله زمانی مشخص استفاده می شوند. برای اطمینان از یادگیری کارآمد عامل، حالت باید تا آنجا که ممکن است اطلاعات مربوط به توزیع وسایل نقلیه را پوشش دهد. با توجه به تعریف و طول ده سلول در هر خط ، اگر وسیله ای در سلول وجود داشته باشد، آن سلول به عنوان ۱، در غیر این صورت، ۰ مشخص می شود.

اقدامات: در این آزمایش، شرط می کنیم که عملکرد جهت مربوطه "برو" است. به عبارت دیگر، رنگ روشن خط سبز یا برای مدت زمان واحد سبز می شود.

عملکرد پاداش: زمان کل وزن وسایل نقلیه به عنوان جمع کل زمان انتظار وسایل نقلیه در محیط در یک مرحله زمان مشخص t تعریف می شود:

$$\text{WaitingTime}(t) = \sum_{v=1}^{N_t} \omega v\,(t) \qquad (7)$$

در اینجا، $\omega v\,(t)$ زمان انتظار وسیله نقلیه را نشان می دهد، که با مقدار زمانی که یک وسیله نقلیه با سرعت کمتر از ۰.۱ m/s طی میکند، اندازه گیری می شود. N_t تعداد کل وسایل نقلیه را در مرحله زمان پایان زمان t نشان می دهد.

بر اساس WaitingTime (t)، تابع پاداش به عنوان تفاوت بین زمان انتظار قبل و بعد از اجرای یک عمل تعریف شده است:

$$r_t = \text{WaitingTime}\,(t-1) - \text{WaitingTime}\,(t) \qquad (۸)$$

هنگامی که کل زمان انتظار بین مرحله $t-1$ و t افزایش یابد، عامل پاداش منفی دریافت می کند. به منظور بهینه سازی استراتژی کنترل چراغ راهنمایی، جمع پاداش منفی باید به حداقل برسد.

نتیجه

در این بخش، ما عملکرد روش پیشنهادی را برای کار کنترل چراغ راهنمایی ارزیابی می کنیم. در ابتدا تنظیمات آزمایشی را توصیف می کنیم و سپس نتیجه و تحلیل می کنیم.

در آزمایش، γ=۰.۵ تنظیم شده است. یک اپیزود شامل ۵۴۰۰ ثانیه است که برابر با ۱ ساعت و ۳۰ دقیقه است. K برابربا ۳ تنظیم شده است. در یک قسمت، ما به ترتیب فرض می کنیم ۵۰۰ ، ۱۰۰۰ و ۱۵۰۰ وسیله نقلیه از چهارراه چهار طرف عبور می کنند که سناریوهای ترافیک سبک، ترافیک عادی و ترافیک سنگین را نشان می دهد. در اینجا، در ابتدای هر شبیه سازی، برای به دست آوردن اطلاعات بیشتر

[Downloaded from ic4i-journal.ir on 2026-02-17]

state برای آموزش DRL، با اجتناب از سقوط رویکرد DRL در بهینه محلی، مقدار € بزرگ درنظر گرفته می شود. با افزایش اپیزود، € باید تا حد ممکن کوچک تنظیم شود. به طور خلاصه، مقدار € را به صورت زیر تنظیم می کنیم:

$$€_{episode} = 1- (episode/M) \qquad (9)$$

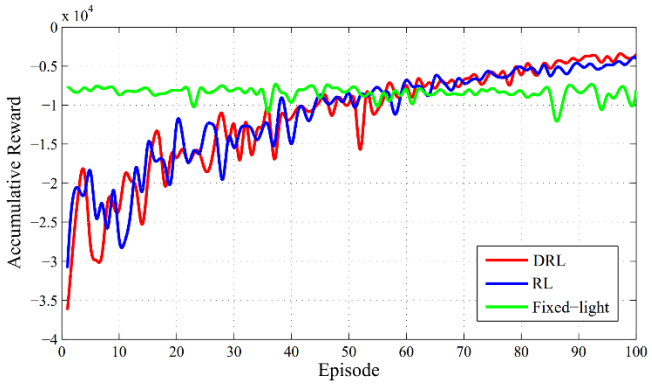
به منظور ارزیابی عملکرد روش پیشنهادی، روش کنترل چراغ راهنمایی مبتنی بر DRL پیشنهادی را با دو رویکرد دیگر کنترل fixed-time و کنترل مبتنی بر RL مقایسه می کنیم. عملکرد سیستم هر قسمت از طریق دو شاخص، یعنی پاداش انباشته و میانگین زمان انتظار، اندازه گیری می شود. به طور خاص، "پاداش انباشته" با جمع پاداش ها محاسبه می شود. "میانگین زمان انتظار" با متوسط زمان انتظار برای همه وسایل نقلیه بدست می آید.

در آزمایشات، هدف افزایش کارایی ترافیک در تقاطع ها یا به حداکثر رساندن پاداش انباشته، است. شکل ۱ و شکل ۲ به ترتیب پاداش تجمعی و میانگین زمان انتظار رویکردهای کنترل زمان ثابت، مبتنی بر RL و مبتنی بر DRL را در مورد سناریوی ترافیک عادی نشان می دهد.

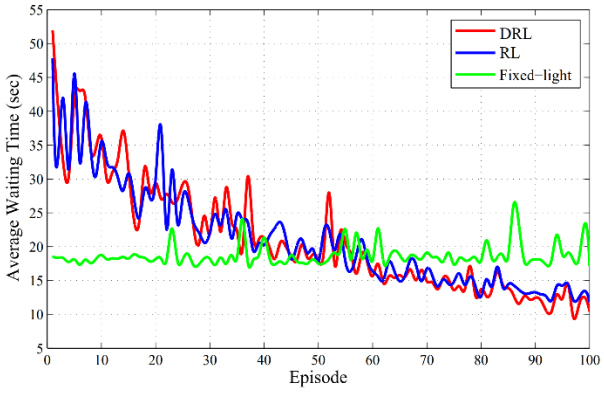
از شکل ۱، در اپیزودهای اولیه، منحنیهای روشهای کنترل مبتنی بر RL و مبتنی بر DRL پیشنهادی دارای نوسان زیادی از دامنه ها به دلیل نرخ اکتشاف بالا هستند. در اپیزودهای بعدی، منحنی هر دو روش پایدار است و دامنه پاداش های انباشته نسبتاً نزدیک است. در مقایسه با رویکرد سنتی زمان ثابت، رویکردهای کنترل مبتنی بر RL و مبتنی بر DRL پیشنهادی می توانند پاداش بالاتری کسب کنند، در نتیجه عملکرد بهتری دارند. میانگین زمان انتظار می تواند بصورت بصری توانایی تصمیم گیری رویکردها را منعکس کند.

از شکل ۲ می بینیم که، پس از چندین قسمت یادگیری، نشانگر رویکردهای کنترل مبتنی بر RL و DRL به طور قابل توجهی کمتر از رویکرد کنترل زمان ثابت است. این نشان می دهد که هر دو الگوریتم یادگیری تقویتی می توانند وضعیت فعلی تقاطع را با موفقیت درک کنند و می توانند با کنترل منطقی چراغ سیگنال، کارایی جریان ترافیک در تقاطع را افزایش دهند و باعث کاهش زمان انتظار وسایل نقلیه شوند.

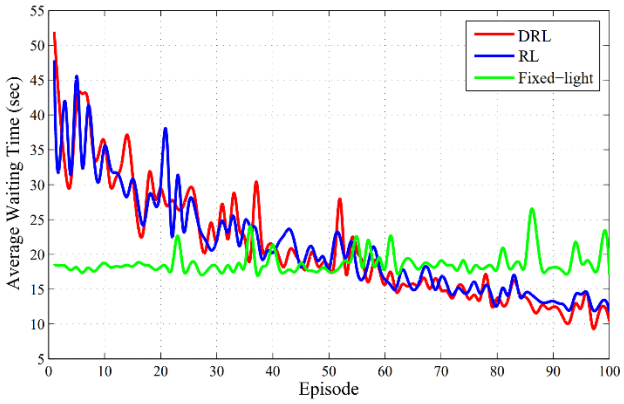
علاوه بر این، پاداش تجمعی و میانگین زمان انتظار رویکرد کنترل مبتنی بر DRL بهتر از رویکرد کنترل مبتنی بر RL است. دلیل این امر این است که با استفاده از الگوریتم یادگیری بهینه شده، در DRL می تواند ویژگی ها و پارامترهای مناسب برای مدل سازی تعیین کند، برعکس RL که ساختار مدل معمولاً از قبل تعیین شده است. تغییر ساختار مدل آن بسیار پیچیده و دشوار است و بر دقت مدل سازی تأثیر می گذارد.



شکل ۱: پاداش تجمعی در ترافیک عادی

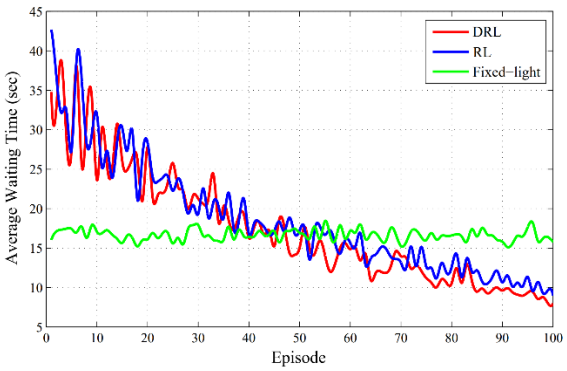


شکل ۲: میانگین زمان انتظار در ترافیک عادی



شکل ۳: میانگین زمان انتظار در ترافیک سبک

[Downloaded from ic4i-journal.ir on 2026-02-17]



شکل ۴: میانگین زمان انتظار در ترافیک سنگین

به منظور ارزیابی بیشتر عملکرد روشهای پیشنهادی مبتنی بر DRL تحت سناریوهای مختلف ترافیک. نتیجه تجربی سناریوی ترافیک سبک در شکل ۳ نشان داده شده است. دیده می شود که وقتی خطوط نسبتاً آزاد هستند، عامل می تواند به سرعت بیاموزد که چگونه یک استراتژی کنترل ترافیک موثرتر برای کاهش زمان انتظار خودرو تدوین کند. برعکس، تحت سناریوید ترافیک سنگین یا ترافیک در تقاطع نسبتاً شلوغ، محیط پیچیده می شود و عوامل نامطمئن تری را شامل می شود. از نتیجه ای که در شکل ٤ نشان داده شده است، تحت سناریوی ترافیک سنگین، روش کنترل زمان ثابت نمی تواند جریان ترافیک را از طریق تقاطع به خوبی کنترل کند و ازدحام جدی وسیله نقلیه اغلب رخ می دهد.

برای رویکردهای کنترل مبتنی بر DRL پیشنهادی و مبتنی بر RL، اگرچه در اپیزودهای بعدی نیز نوساناتی را تجربه می کنند، اما هنوز هم می توانند استراتژی های تصمیم گیری عالی برای کاهش متوسط زمان انتظار وسایل نقلیه را تحقق بخشند. به طور خلاصه، از شکل ۲ تا شکل ٤، روش کنترل مبتنی بر DRL می تواند به ترتیب در پایان سه حالت مختلف، ۱۰ ثانیه، ۸ ثانیه و ۱۵ ثانیه میانگین زمان انتظار را بدست آورد و بهترین عملکرد را داشته باشد.

در این مقاله، چارچوبی جدید به نام DRL برای پشتیبانی از اینترنت اشیا پیشنهاد شده است. به طور خاص، شبکه های عصبی عمیق در معماری های الگوریتم های تقویتی موجود با DRL پیشنهادی جایگزین می شوند، که ساختار آن ساده تر اما موثرتر است. علاوه بر این، آماده سازی نمونه آموزش و انتخاب ویژگی برای DRL به دقت طراحی شده است. سرانجام، به عنوان یک مورد استفاده، روش پیشنهادی DRL برای سناریوی کنترل چراغ راهنمایی اجرا می شود.

نتایج تجربی نشان می دهد که رویکرد کنترل مبتنی بر DRL پیشنهادی نه تنها می تواند توانایی های مدل سازی و تصمیم گیری بالاتر را بدست آورد، بلکه در مقایسه با رویکردهای کنترل مبتنی بر DRL نیز به زمان پیچیدگی محاسباتی کمتری دست پیدا می کند. بدون استفاده از کمک انسان و زمان پاسخگویی طولانی برای حل مشکل داده های بزرگ، برای اینترنت اشیا بسیار مناسب است.

مراجع

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE communications surveys & tutorials, vol. 17, no. 4, pp. 2347-2376, 2015.

[2] L. Zhou, D. Wu, J. Chen, and Z. Dong, "When computation hugs intelligence: Content-aware data processing for industrial IoT," IEEE Internet of Things Journal, vol. 5, no. 3, pp. 1657-1666, 2017.

[3] D. Wang, W. Bao, C. Hu, Y. Qian, M. Zheng, and S. Wang, "sTube: An architecture for IoT communication sharing," IEEE Communications Magazine, vol. 56, no. 7, pp. 96-101, 2018.

[4] S. Marchal, M. Miettinen, T. D. Nguyen, A.-R. Sadeghi, and N. Asokan, "Audi: Toward autonomous iot device-type identification using periodic communication," IEEE Journal on Selected Areas in Communications, vol. 37, no. 6, pp. 1402-1412, 2019.

[5] L. Zhou, D. Wu, X. Wei, and Z. Dong, "Seeing isn't believing: QoE evaluation for privacy-aware users," IEEE Journal on Selected Areas in Communications, vol. 37, no. 7, pp. 1656-1665, 2019.

[6] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT press, 2018.

[7] V. Mnih et al., "Human-level control through deep reinforcement learning," nature, vol. 518, no. 7540, pp. 529-533, 2015.

[8] C. Lei, "Deep reinforcement learning," in Deep Learning and Practice with

[Downloaded from ic4i-journal.ir on 2026-02-17]

[17] M. Chu, H. Li, X. Liao, and S. Cui, "Reinforcement learning-based multiaccess control and battery prediction with energy harvesting in IoT systems," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2009-2020, 2018.

[18] Y. Wei, F. R. Yu, M. Song, and Z. Han, "Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural actor–critic deep reinforcement learning," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2061-2073, 2018.

[19] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J.-S. Oh, "Semisupervised deep reinforcement learning in support of IoT and smart city services," IEEE Internet of Things Journal, vol. 5, no. 2, pp. 624-635, 2017.

[20] R. Kanan, O. Elhassan, and R. Bensalem, "An IoT-based autonomous system for workers' safety in construction sites with real-time alarming, monitoring, and positioning strategies," Automation in Construction, vol. 88, pp. 73-86, 2018/04/01/ 2018, doi: https://doi.org/10.1016/j.autcon.2017.12.033.

[21] L. Barbierato et al., "A Distributed IoT Infrastructure to Test and Deploy Real-Time Demand Response in Smart Grids," IEEE Internet of Things Journal, vol. 6, no. 1, pp. 1136-1146, 2019, doi: 10.1109/JIOT.2018.2867511.

[22] D. S. Roy, R. K. Behera, K. H. K. Reddy, and R. Buyya, "A context-aware fog enabled scheme for real-time cross-vertical IoT applications," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2400-2412, 2018.

[23] B. V. Philip, T. Alpcan, J. Jin, and M. Palaniswami, "Distributed Real-Time IoT for Autonomous Vehicles," IEEE Transactions on Industrial Informatics, vol. 15, no. 2, pp. 1131-1140, 2019, doi: 10.1109/TII.2018.2877217.

[25] Patel, R., & Singh, V. (2024). Enhancing IoT security with deep reinforcement MindSpore: Springer, 2021, pp. 217-243.

[9] H. Sami, A. Mourad, H. Otrok, and J. Bentahar, "Demand-Driven Deep Reinforcement Learning for Scalable Fog and Service Placement," IEEE Transactions on Services Computing, 2021.

[10] L. Yu, S. Qin, M. Zhang, C. Shen, T. Jiang, and X. Guan, "A review of deep reinforcement learning for smart building energy management," IEEE Internet of Things Journal, 2021.

[11] M. Paniri, M. B. Dowlatshahi, and H. Nezamabadi-pour, "MLACO: A multi-label feature selection algorithm based on ant colony optimization," Knowledge-Based Systems, vol. 192, p. 105285, 2020.

[12] B. H. Nguyen, B. Xue, and M. Zhang, "A survey on swarm intelligence approaches to feature selection in data mining," Swarm and Evolutionary Computation, vol. 54, p. 100663, 2020.

[13] D. Zeng, L. Gu, S. Pan, J. Cai, and S. Guo, "Resource management at the network edge: A deep reinforcement learning approach," IEEE Network, vol. 33, no. 3, pp. 26-33, 2019.

[14] Y. Zhang, J. Yao, and H. Guan, "Intelligent cloud resource management with deep reinforcement learning," IEEE Cloud Computing, vol. 4, no. 6, pp. 60-69, 2017.

[15] B. Peng, G. Seco-Granados, E. Steinmetz, M. Fröhle, and H. Wymeersch, "Decentralized scheduling for cooperative localization with deep reinforcement learning," IEEE Transactions on Vehicular Technology, vol. 68, no. 5, pp. 4295-4305, 2019.

[16] C. Wang, J. Wang, Y. Shen, and X. Zhang, "Autonomous navigation of UAVs in large-scale complex environments: A deep reinforcement learning approach," IEEE Transactions on Vehicular Technology, vol. 68, no. 3, pp. 2124-2136, 2019.

[Downloaded from ic4i-journal.ir on 2026-02-17]

learning. *ACM Transactions on Internet Technology*.

[26] Kim, S., & Park, H. (2024). DRL-based energy management for autonomous vehicles. *IEEE Transactions on Vehicular Technology*.

[2٦] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," IEEE computational intelligence magazine, vol. 1, no. 4, pp. 28-39, 2006.

[2٧] E.-G. Talbi, Metaheuristics: from design to implementation. John Wiley & Sons, 2009.

[2٨] D. Zhao, Y. Dai, and Z. Zhang, "Computational Intelligence in Urban Traffic Signal Control: A Survey," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 4, pp. 485-494, 2012, doi: 10.1109/TSMCC.2011.2161577.

[2٩] Z. Li, R. A. Hassan, M. Shahidehpour, S. Bahramirad, and A. Khodaei, "A Hierarchical Framework for Intelligent Traffic Management in Smart Cities," IEEE Transactions on Smart Grid, vol. 10, no. 1, pp. 691-701, 2019, doi: 10.1109/TSG.2017.2750542.

[٣٠] "Simulation of Urban MObility." https://www.eclipse.org/sumo/ (accessed.

[Downloaded from ic4i-journal.ir on 2026-02-17]