Journal of New Technologies in Distributed Systems and Algorithmic Computing
Islamic Azad University of Sabzevar
E-ISSN: 3115-705X
https://sanad.iau.ir/journal/ntds

Research Article

Multi-Agent Task Offloading Approach based on Software Defined-Networking in Vehicular Fog Networks

Kobra Behravan [1] | Seyed Amin Hosseini Seno [2] | Nazbanoo Farzaneh [*3] | Mohsen Jahanshahi [4]

[1] Department of Computer Engineering, CT.C., Islamic Azad University, Tehran, Iran, k.behravan@iau.ac.ir

[2] Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran, hosseini@um.ac.ir

[3] Department of Computer Engineering, Imam Reza International University, Mashhad, Iran, nazbanou.farzaneh@imamreza.ac.ir

[4] Department of Computer Engineering, CT.C., Islamic Azad University, Tehran, Iran, mjahanshahi@iauctb.ac.ir

Correspondence
*Nazbanoo Farzaneh, Assistant professor, Department of Computer Engineering, Imam Reza International University, Mashhad, Iran, nazbanou.farzaneh@imamreza.ac.ir

Main Subjects: Finding optimal task offloading policy
Received: 15 November 2025
Revised: 4 December 2025
Accepted: 19 December 2025

https://doi.org/10.82195/ntds.2025.1224422

Abstract

Vehicular fog computing (VFC) been recognized as an effective architecture to address rising demands in smart vehicles. Fog servers deployed on moving or parked vehices can provide spatio-temporal heterogeneity of computational resources that capable of accomplishing computation and deadline intensive-tasks beyond the capacity which is embedded inside the vehicles. In multi-tier VFC architecture, vehicles can share idle low cost resources to increases acceptance tasks. However, the main issue is choosing optimal destination fog server for executing hard deadline tasks at each time slot. Therefore, in this work proposed a federated multi-agent deep Q-learning task offloading approach to provide collaborative learning and fast convergence. This approach improves privacy of data for agents and reduces average response time, energy consumption and processing cost in vehicular networks. Software-defined networking (SDN)-based architecture can provide flexibility, scalability, programmability, and overall network knowledge. With the help of SDN control plane configuration, the network can not only adapt to dynamic network changes, but also respond to emergency situations. Therefore, SDN technology integrated with federated reinforcement learning (FRL) method increases programmability, centralized network management,and dynamic configuration. The results show that the proposed method reduces the average response time, average energy consumption, and average economic cost of performing tasks in the network and will increase the successful performance of high-priority tasks in vehicular fog networks.

Keywords: Vehicular Fog Computing (VFC), Task Offloading, Software-Defined Networking (SDN), Federated Reinforcement Learning (FRL)

فناوری های نوین در سیستم های توزیع شده و محاسبات الگوریتمی

دانشگاه آزاد اسلامی واحد سبزوار

E-ISSN: 3115-705X
https://sanad.iau.ir/journal/ntds

پژوهشی

رویکرد برون‌سپاری وظیفه چند عامله مبتنی بر شبکه نرم‌افزار محور در شبکه‌های مه خودرویی

کبری بهروان[۱] | سید امین حسینی سنو[۲] | نازبانو فرزانه*[۳] | محسن جهانشاهی[۴]

[۱] گروه مهندسی کامپیوتر، واحد تهران مرکزی، دانشگاه آزاد اسلامی، تهران، ایران،
k.behravan@iau.ac.ir

[۲] گروه مهندسی کامپیوتر، دانشگاه فردوسی مشهد، مشهد، ایران،
hosseini@um.ac.ir

[۳] گروه مهندسی کامپیوتر، دانشگاه بین المللی امام رضا، مشهد، ایران،
nazbanou.farzaneh@imamreza.ac.ir

[۴] گروه مهندسی کامپیوتر، واحد تهران مرکزی، دانشگاه آزاد اسلامی، تهران، ایران،
mjahanshahi@iauctb.ac.ir

نویسنده مسئول
* نازبانو فرزانه، استادیار، گروه مهندسی کامپیوتر، دانشگاه بین المللی امام رضا، مشهد، ایران ،
nazbanou.farzaneh@imamreza.ac.ir

موضوع اصلی: یافتن سیاست برون‌سپاری وظیفه بهینه
تاریخ دریافت: ۲۴ آبان ۱۴۰۴
تاریخ بازنگری: ۱۳ آذر ۱۴۰۴
تاریخ پذیرش: ۲۸ آذر ۱۴۰۴

https://doi.org/10.82195/ntds.2025.1224422

چکیده:

رایانش مه خودرویی (VFC) به عنوان یک معماری مؤثر برای رسیدگی به تقاضاهای رو به افزایش در خودروهای هوشمند شناخته شده است. سرورهای مه مستقر در خودروهای در حال حرکت یا پارک شده می‌توانند ناهمگونی فضایی-زمانی منابع محاسباتی را فراهم کنند که قادر به انجام محاسبات وظایف فشرده با مهلت فراتر از ظرفیت تعبیه شده در داخل خودروها هستند. در معماری VFC چند لایه، خودروها می‌توانند منابع کم هزینه و بیکار را برای افزایش پذیرش وظایف به اشتراک بگذارند. با این حال، مسئله اصلی انتخاب سرور مه مقصد بهینه برای اجرای وظایف با مهلت سخت در هر برش زمانی است. بنابراین، در این کار یک رویکرد برون‌سپاری وظیفه Q-learning عمیق چند عاملی فدرال برای ارائه یادگیری مشارکتی و همگرایی سریع پیشنهاد شده است. این رویکرد حریم خصوصی داده‌ها را برای عامل‌ها بهبود می‌بخشد و میانگین زمان پاسخ، مصرف انرژی و هزینه پردازش را در شبکه‌های خودرویی کاهش می‌دهد. معماری مبتنی بر شبکه نرم‌افزار محور (SDN) می‌تواند انعطاف‌پذیری، مقیاس‌پذیری، قابلیت برنامه‌ریزی و دانش کلی شبکه را فراهم نماید. با کمک پیکربندی صفحه‌ی کنترلی SDN، شبکه می‌تواند نه فقط با تغییرات پویای شبکه، وفق پذیر باشد بلکه می‌تواند به وضعیت‌های اورژانسی پاسخ دهد. بنابراین، SDN که با روش یادگیری تقویتی فدرال (FRL) ادغام شده است، قابلیت برنامه‌ریزی، مدیریت شبکه متمرکز و پیکربندی پویا را افزایش می‌دهد. نتایج، نشان می‌دهند که روش پیشنهادی میانگین زمان پاسخدهی، میانگین مصرف انرژی و میانگین هزینه اقتصادی جهت انجام وظایف در شبکه را کاهش می‌دهد و باعث افزایش انجام موفقیت آمیز وظایف با اولویت بالا در شبکه‌های مه خودرویی خواهد شد.

کلید واژه‌ها: برون‌سپاری وظیفه، رایانش مه خودرویی، شبکه نرم‌افزار محور، یادگیری تقویتی عمیق

۱-مقدمه

آینده همیشه با ماشین‌های مدرن و تخیلی تصویر می‌شود که در مقایسه با وسایل حمل و نقل کلاسیک بیشتر شبیه ربات هستند. این تصویر امروزه به نوعی با وسایل نقلیه خودران محقق شده است و ما بیش از هر زمان دیگری به سبک حمل و نقل آینده‌نگر نزدیک هستیم. بازار اینترنت خودرویی[۱] در سال ۲۰۲۱ معادل ۹۶ میلیارد دلار تخمین زده شده و تقریباً پیش‌بینی می‌شود که در سال ۲۰۲۸ به ۳۷۰ میلیارد دلار برسد[۱]. با تکامل هوش مصنوعی[۲]، سرویس‌های حمل و نقل هوشمندتر و بیشتری در وسایل نقلیه هوشمند، مانند موقعیت‌یابی با دقت بالا، رانندگی هوشمند و سرگرمی‌های چندرسانه‌ای، پدیدار می‌شوند که تجربه رانندگی کاربران را تا حد زیادی بهبود می‌بخشند. با این حال، این سرویس‌های هوش مصنوعی بلادرنگ اغلب محاسباتی و حساس به زمان هستند که به دلیل منابع محاسباتی محدود موجود در خودروها، مدیریت کامل آنها توسط خودروها دشوار است. با بهره‌گیری از توسعه سریع فناوری ارتباطات بی‌سیم، برون‌سپاری وظایف از خودروها به سرورها به یک الگوی امیدوارکننده برای کمک به خودروها با منابع محاسباتی خارجی تبدیل شده است[۲]. رویکرد سنتی برون‌سپاری وظیفه مبتنی بر ابر، از منابع محاسباتی عظیم روی یک ابر متمرکز استفاده می‌کند که معمولاً در یک مکان دورافتاده قرار دارد. با این حال، رویکرد رایانش ابری[۳] اغلب از تأخیرهای بالای انتقال و اتصالات بی‌سیم نوسان‌دار رنج می‌برد که نمی‌تواند الزامات وظایف حساس به زمان را برآورده کند[۳].

برای برآورده کردن نیاز به تأخیر کم در برون‌سپاری وظیفه، رایانش لبه موبایل[۴] در شبکه‌های خودرویی، یعنی رایانش لبه خودرویی[۵]، معرفی شده است که در آن واحد کنار جاده[۶] یا ایستگاه پایه[۷] به مقدار مشخصی از منابع محاسباتی و ذخیره‌سازی برای ارائه خدمات محاسباتی برای خودروها مجهز شده‌اند. با این حال، با افزایش تراکم ترافیک، تضمین کیفیت خدمات[۸] همه برنامه‌های خودرویی برای منابع محاسباتی محدود در واحد کنار جاده یا ایستگاه پایه دشوار است. علاوه بر این، سرورهای معمولاً به صورت پراکنده مستقر می‌شوند و عملکرد آنها توسط پوشش رادیویی محدود می‌شود[۴].

برای غلبه بر محدودیت‌های ذکر شده، رایانش مه[۹] به عنوان یک راه حل امیدوارکننده برای وظایف حساس به تأخیر در برنامه‌های کاربردی وسایل نقلیه در شبکه‌های وسایل نقلیه ظهور کرد[۵]. کنسرسیوم OpenFog معماری مرجع خود را برای ساخت یک سکوی رایانش مه سازگار و مقیاس‌پذیر در سال ۲۰۱۷ منتشر کرده است. با افزایش ترافیک سرویس‌های خودرویی، گره‌های مه احتمالاً با مشکل مواجه خواهند شد. یک سوال طبیعی این است که چگونه می‌توان منابع محاسباتی گره‌های مه را افزایش داد. رایانش مه خودرویی[۱۰] به‌عنوان یک مورد استفاده از رایانش مه، در زمینه سیستم‌های حمل و نقل هوشمند پدیدار شده است. رایانش مه خودرویی از منابع خودرویی برای ارتقای قابلیت محاسباتی و همچنین کاهش بیشتر تأخیر محاسبات مه استفاده می‌کند. بنابراین، منابع محاسباتی استفاده نشده خودروها می‌توانند به‌عنوان اجزای گره‌های مه، مانند وسایل نقلیه در پارکینگ‌ها یا مراکز خرید، تکمیل شوند[۶].

با این حال، فناوری اینترنت با بهترین تلاش فعلی، انعطاف‌پذیری تصمیم‌گیری پویا در مورد برون سپاری وظایف را محدود می‌کند. برای رفع این محدودیت‌ها، محققان طرح‌های برون‌سپاری وظیفه را در شبکه‌های نرم‌افزار محور[۱۱] پیشنهاد کردند که در آن یک کنترل‌کننده متمرکز تصمیمات برون‌سپاری وظیفه را می‌گیرد. علاوه بر این، رویکرد شبکه نرم‌افزار محور به

---

[1] Internet of Vehicles
[2] Artificial Intelligence
[3] Cloud Computing
[4] Mobile Edge Computing
[5] Vehicular Edge Computing
[6] Road Side Unit
[7] Base Station
[8] Quality of Service
[9] Fog Computing
[10] Vehicular Fog Computing
[11] Software Defined Networks

کنترل‌کننده اجازه می‌دهد تا شبکه را به شیوه‌ای انعطاف‌پذیر و ساده مدیریت کند[۷]. در نتیجه، در این کار، ما بر برون‌سپاری وظیفه مبتنی بر رویکرد نرم‌افزار محور در حوزه رایانش مه خودرویی تمرکز می‌کنیم.

شبکه نرم‌افزار محور برای رفع عدم انعطاف‌پذیری در شبکه قدیمی از طریق جداسازی صفحه ارسال از صفحه کنترل و فراهم کردن قابلیت برنامه‌ریزی در صفحه کنترل طراحی شده است. همچنین، کنترلر شبکه نرم‌افزار محور، مسئول جمع‌آوری تمام اطلاعات در صفحه داده به صورت دوره‌ای است. این اطلاعات مانند قابلیت محاسبه، تأخیر ارتباط، میانگین زمان انتظار و هزینه واحد برای حفظ یک نمای کلی از شبکه است. بنابراین، می‌تواند تمام منابع موجود در شبکه را به روشی کارآمد و سریع مدیریت، پیکربندی و هماهنگ کند. با آوردن این ویژگی به معماری پیشنهادی، یک سیاست برون‌سپاری وظیفه مبتنی بر شبکه نرم‌افزار محور برای تصمیم‌گیری صحیح برون‌سپاری به گره مه بهینه پیشنهاد شد[۸].

در نتیجه، اخیراً برون‌سپاری وظیفه در شبکه مه خودرویی توجه فزاینده‌ای را به خود جلب کرده است. با این حال، به دلیل ویژگی‌های منحصر به فرد شبکه‌های خودرویی، به ویژه تحرک بالای گره‌ها و ویژگی‌های کانال پویا، طراحی یک طرح برون‌سپاری وظیفه مبتنی بر لبه کارآمد بسیار چالش برانگیز است. بنابراین، چگونگی طراحی یک استراتژی کارآمد برای برون‌سپاری وظیفه که بتواند با محیط پویای شبکه سازگار شود، از منابع محاسباتی لبه به طور کامل استفاده کند و به تعادل بار بین دستگاه‌های مختلف دست یابد، به یک مسئله تحقیقاتی فوری تبدیل شده است.

چالش‌های زیادی در رایانش مه خودرویی وجود دارند. اولین چالش، انرژی محدود خودروهای کاربران است. اگر دستگاه‌های کاربران، انرژی کافی برای اجرای برنامه نداشته باشند، برنامه می‌تواند به طور کامل[۱] و جزئی[۲] به سرورهای دیگر در شبکه منتقل شود. علاوه بر این، اکثر خودروهای الکتریکی با قابلیت‌های محاسباتی، ذخیره‌سازی و انرژی بالا بیش از ۹۰ درصد از وقت خود را در پارکینگ‌ها می‌گذرانند. بنابراین، یک ایده می‌تواند استفاده از منابع محاسباتی خودروهای الکتریکی پارک شده برای ارائه خدمات محاسباتی بر اساس تقاضا در سراسر شبکه باشد[۹].

چالش دیگر، حفظ کیفیت سرویس در سرورهای مه است. کیفیت سرویس را می‌توان با تقسیم بار محاسباتی به طور مساوی بین سرورهای مه بهبود بخشید. بار محاسباتی در یک سرور مه به نرخ‌های ورود درخواست مرتبط با آن بستگی دارد. سرورهای مه ممکن است نرخ‌های ورود درخواست متفاوتی داشته باشند، این ممکن است وضعیتی را ایجاد کند که یک سرور مه بیش از بقیه سرورها بارگذاری شود. از این رو، ارتباط بین سرورهای مه باید به گونه‌ای انجام شود که عدم تعادل بار در سرورهای مه به حداقل برسد.

امروزه، از تکنیک‌های یادگیری تقویتی[۳] برای یافتن یک استراتژی بهینه برون‌سپاری وظیفه استفاده می‌شود[۱۰]. یادگیری تقویتی به طور هوشمندانه سیاست بهینه را برای به حداکثر رساندن پاداش بر اساس بازخورد دائمی از محیط یاد می‌گیرد. با این حال، زمان مورد نیاز برای کاوش کامل محیط پیچیده و در نهایت همگرایی به یک سیاست بهینه، یادگیری تقویتی سنتی را غیرممکن می‌سازد. این امر به ویژه هنگام مقیاس‌بندی یادگیری تقویتی برای محیط‌های بزرگ و پویا با فضای حالت-عمل گسترده و پیوسته چالش برانگیز است. برای پرداختن به چنین فضاهایی در مقیاس بزرگ، یادگیری تقویتی عمیق[۴] معرفی شده است تا تعمیم و سازگاری بهتری با محیط‌های ناشناخته ارائه دهد[۱۱].

در این مقاله، ما یک تکنیک برون‌سپاری وظیفه فدرال بر خط[۵] مبتنی بر یادگیری عمیق[۶] را برای رایانش مه خودرویی پیشنهاد می‌کنیم که به طور مشترک مصرف انرژی و تعادل بار را در سراسر سرورهای مه بهینه می‌کند. ما از یک شبکه مه خودرویی مبتنی بر منطقه استفاده می‌کنیم که در آن هر منطقه دارای یک سرور مه ایستا، سرور مه پویا و چندین واحد کنار جاده ای است. در هر منطقه مه، یک شبکه یادگیری عمیق توسط یک عامل آموزش داده می‌شود که یک استراتژی برون‌سپاری وظیفه محلی بهینه برای سرورهای مه مربوط به خود تعیین می‌کند. یک استراتژی برون‌سپاری وظیفه بهینه، مصرف انرژی را

---

[1] Binary
[2] Partial
[3] Reinforcement learning
[4] Deep Reinforcement Learning
[5] Online federated task offloading
[6] Deep Learning

در سرورهای مه به حداقل می‌رساند و بار را به طور یکنواخت در سرورهای مه در یک منطقه مه توزیع می‌کند. یک رویکرد برون‌سپاری وظیفه سراسری با جمع کردن پارامترهای یادگیری محلی مناطق مختلف مه به دست می‌آید. دستاوردهای مقاله ما به شرح زیر است:

- ما مسئله برون‌سپاری وظیفه را به عنوان یک فرآیند تصمیم‌گیری مارکوف[1] مدل‌سازی می‌کنیم و در عین حال ظرفیت محاسباتی متغیر با زمان یک سرور مه پویا و صف‌های اولویت دار برای زمانبندی وظایف در وسایل نقلیه را در نظر می‌گیریم. مدل مارکوف هم بهینه‌سازی انرژی و هم تعادل بار در سرورهای مه را در نظر می‌گیرد.
- ما یک تکنیک برون‌سپاری وظیفه فدرال بر خط چند عامله مبتنی بر یادگیری عمیق را برای رایانش مه خودرویی پیشنهاد می‌کنیم. یک تکنیک برون‌سپاری وظیفه فدرال[2]، با تجمیع مدل‌های برون‌سپاری وظیفه محلی مناطق مختلف مه، یک مدل برون‌سپاری وظیفه سراسری توسعه می‌دهد.
- ما شبیه‌سازی‌های گسترده‌ای را برای ارزیابی عملکرد رویکرد پیشنهادی خود را با استفاده از داده‌های واقعی حرکت خودروها انجام می‌دهیم. نتایج شبیه‌سازی ما نشان می‌دهد که روش پیشنهادی با نرخ همگرایی مشابهی با رویکرد متمرکز همگرا می‌شود و در مقایسه با سایر تکنیک‌های پیشرفته برون‌سپاری وظیفه ، عملکرد بهتری دارد.

ساختار این مقاله به شرح زیر است. بخش ۲ کارهای مرتبط را ارائه می‌دهد. بخش ۳ مدل سیستم و فرمول‌بندی مسئله را شرح می‌دهد. در بخش ۴، تکنیک پیشنهادی ارائه می‌شود. ارزیابی عملکرد در بخش ۵ نشان داده شده است. در نهایت، نتیجه‌گیری در بخش ۶ ارائه شده است.

۲-مروری بر کارهای انجام شده

ماهیت پویای رایانش مه خودرویی، تحقیقات گسترده‌ای را در مورد استراتژی‌های برون‌سپاری وظیفه، با تمرکز اصلی بر کاهش زمان پاسخ و مصرف انرژی، برانگیخته است. محدودیت‌های مهم مطالعات موجود شامل عدم قطعیت در تحرک و تقاضاهای درخواست، پوشش محدود سرورهای مه و پیچیدگی محاسباتی وظایف مختلف است. در این بخش، ما یک دسته بندی برای کارهای مرتبط بر اساس رویکردهای برون‌سپاری وظیفه بهینه مبتنی بر یادگیری تقویتی تک عامله و رویکردهای برون‌سپاری وظیفه بهینه مبتنی بر یادگیری تقویتی چند عامله در محیط رایانش مه خودرویی را ارائه می‌دهیم. جدول ۱ تحقیقات مرتبط و روش پیشنهادی را با توجه به پارامترهای مختلف و روش‌های یادگیری ماشین مقایسه می‌کند.

۲-۱- رویکردهای برون‌سپاری وظیفه بهینه مبتنی بر یادگیری تقویتی تک عامله

با پیشرفت یادگیری ماشین، بسیاری از تحقیقات موجود یادگیری تقویتی تک عاملی را برای حل مشکلات تخصیص منابع محاسباتی در رایانش مه خودرویی به کار گرفته اند. در [۴]، یک سیاست برون‌سپاری وظیفه ارائه شده است که پویایی شبکه خودرو، اولویت وظیفه و در دسترس بودن سرویس را در نظر می‌گیرد. مسئله برون‌سپاری وظیفه با استفاده از فرآیند تصمیم‌گیری مارکوف فرموله شده و با استفاده از الگوریتم یادگیری تقویتی عمیق مبتنی بر مدل منتقد بازیگر نرم (SAC) [3]حل شده است. هدف این مقاله به حداکثر رساندن میانگین مطلوبیت آگاه از تأخیر است. یادگیری تقویتی عمیق در [۱۲] برای یافتن بهترین راه‌حل بلندمدت در یک چارچوب تصمیم‌گیری برون‌سپاری وظیفه مبتنی بر دانش، برای اینترنت خودرویی استفاده می‌شود. هنگامی که سرویس در حال کار است، خودروها یک مدل جدید را در گره محاسبات لبه به‌روزرسانی می‌کنند و یادگیری برخط ناهمزمان را انجام می‌دهند. در [۱۳]، یک معماری شبکه رایانش لبه خودرویی با استفاده از فناوری شبکه نرم‌افزار محور پیشنهاد شد. سپس، یک طرح برون‌سپاری وظیفه مبتنی بر یادگیری عمیق برای سازگاری با نرخ بالای تغییر شبکه ارائه گردید. در [۱۴]، بهینه‌سازی تحویل مبتنی بر یادگیری ارائه شده است. با استفاده از شبکه عصبی پیش‌خور، گره مه در زمان و مکان داده شده پیش‌بینی می‌شود و شبکه عصبی بازگشتی نیز برای پیش‌بینی کارایی و هزینه پیشنهاد شده است. برون‌سپاری وظیفه جزئی

[1] Markov Decision Process

[2] Federated

[3] Soft Actor-Critic

دودویی خودرو به خودرو (V2V) [1]در [۱۵] برای افزایش دسترسی به سرویس و استفاده از منابع پیشنهاد شده است. مدل‌سازی ریاضی توسط الگوریتم یادگیری عمیق حل شده است تا تأخیر و هزینه اجرای وظیفه را کاهش دهد. بوبکر و همکاران [۱۶] پیشنهاد دادند که یک سیاست بهینه برای برون‌سپاری جزئی وظایف گردش کار و مهاجرت پویا در محیط رایانش مه خودرویی ناهمگن و محاسبات ابری پیدا کنند. این مقاله بر رعایت محدودیت‌های مهلت و وابستگی وظایف تمرکز دارد. گردش کار به عنوان یک گراف جهتدار غیر مدور (DAG) [2]برای نشان دادن مجموعه وابستگی‌های متوالی بین جفت وظایف نشان داده می‌شود. در [۱۷]، یک مکانیسم تشویقی مبتنی بر نظریه قرارداد پیشنهاد شده است که از روش یادگیری تقویتی عمیق به صورت توزیع‌شده برای کاهش پیچیدگی پیاده‌سازی استفاده می‌کند و تصمیمات برون‌سپاری وظیفه می‌تواند از طریق شبکه عصبی عمیق سریع‌تر انجام شود. خودروها منابع خود را به اشتراک می‌گذارند و بر اساس میزان منابعی که ارائه می‌دهند، پاداش دریافت می‌کنند. همچنین، خودرو از پاداش انباشته خود برای تبادل پهنای باند انتقال بی‌سیم استفاده می‌کند. این روش مصرف انرژی و نسبت تکمیل وظیفه را بهبود می‌بخشد.

۲-۲- رویکردهای برون‌سپاری وظیفه بهینه مبتنی بر یادگیری تقویتی چند عامله

برون‌سپاری وظیفه و تخصیص منابع، یک نقطه کانونی تحقیقاتی در محاسبات مشارکتی رایانش مه خودرویی است. یادگیری تقویتی تک عاملی، دارای برخی نقص‌ها مانند استحکام پایین[3] و نادیده گرفتن مقیاس‌پذیری[4] است. مرجع [۱۸] یک یادگیری تقویتی عمیق چندعاملی فدرال را برای تصمیمات بهتر برون‌سپاری وظیفه پیشنهاد می‌دهد. این روش با یادگیری تصمیمات برون‌سپاری در چندین لایه، حریم خصوصی داده‌ها و همگرایی را در مدل‌های یادگیری مشارکتی بهبود می‌بخشد. نتایج نشان می‌دهد که رویکرد پیشنهادی زمان اقامت، تأخیر انتها به انتها و هزینه کلی سیستم را کاهش می‌دهد. در مرجع [۱۹]، یک استراتژی مهاجرت و همکاری برای انتقال نتایج محاسبات بین واحدهای کنار جاده ای پیشنهاد شده است که می‌تواند میانگین شکست در تحویل نتایج محاسبات، تأخیر و مصرف انرژی را کاهش دهد. رویکرد یادگیری تقویتی عمیق چندعاملی، تصمیمات برون‌سپاری وظیفه و تخصیص منابع را از طریق آموزش متمرکز بهینه می‌کند. دای و همکارانش [۲۰]، سناریوهای مختلف برون‌سپاری چند وظیفه‌ای (MTO) [5]را با یادگیری تقویتی متا[6] برای پیش‌بینی زمان اجرای وظیفه و سازگاری با سناریوهای پویا پیشنهاد کردند. وظایف کاربر به صورت گراف مبتنی بر گراف جهتدار غیر مدور مدل‌سازی می‌شوند تا زمان اجرا به حداقل برسد. مرجع [۲۱] یک چارچوب مدیریت لبه ارائه می‌دهد که کارایی یادگیری چندین عامل را بهبود می‌بخشد. طرح یادگیری چندعاملی توزیع‌شده می‌تواند هزینه برون‌سپاری وظیفه خودرو را تحت محدودیت تأخیر کاهش داده و استفاده از منابع را افزایش دهد. همچنین رایانش لبه‌ای نرم‌افزار محور ارائه شد که از روش یادگیری ماشین توزیع‌شده برای ایجاد یک معماری مقیاس‌پذیر، مدیریت منابع و هماهنگی استفاده می‌شود[۲۲]. مرجع [۲۳] معماری سلسله مراتبی رایانش مه خودرویی را در نظر می‌گیرد و وظایف را بر اساس یادگیری تقویتی چندعاملی، هم در داخل و هم در داخل ناحیه توزیع می‌کند. رویکرد یادگیری تقویتی چندعاملی به نام COMA [7] یک مبنای ارزیابی عملکرد برای هر عامل ارائه می‌دهد. نتایج نشان می‌دهد که این روش می‌تواند از قابلیت محاسباتی سیستم بهتر استفاده کند. در [۱۰]، یک تکنیک برون‌سپاری وظیفه مبتنی بر یادگیری Q عمیق فدرال برای رایانش مه خودرویی برای بهبود مصرف انرژی و تعادل بار در واحدهای کنار جاده پیشنهاد شده است.

برخلاف کارهای تحقیقاتی مذکور، روش پیشنهادی ما یک تکنیک برون‌سپاری وظیفه پویا است که جنبه‌های بهبود کیفیت سرویس در شبکه‌های مه خودرویی مانند کاهش هزینه، تاخیر و مصرف انرژی، متعادل سازی بار در سرورهای مه و استفاده از اطلاعات سراسری در توسعه یک مدل برون‌سپاری وظیفه را پوشش می‌دهد.

---

[1] Vehicle to Vehicle
[2] Directed Acyclic Graph
[3] Low robustness
[4] Scalability
[5] Multi Task Offloading
[6] Meta reinforcement learning
[7] Counterfactual multi-agent

جدول ۱: مقایسه کارهای انجام شده براساس پارامترهای تاثیر گذار روی برون‌سپاری وظیفه
Table 1. Comparison of completed tasks based on parameters affecting task offloading

محیط	هزینه	اولویت وظایف	مهلت زمانی	انرژی	زمان انتظار در صف	مرجع
خودرو-مه	×	✓	✓	×	✓	[۴]
خودرو-مه	×	×	✓	✓	×	[۱۰]
خودرو-لبه	×	×	×	×	×	[۱۲]
خودرو-MEC-ابر	×	×	×	×	×	[۱۳]
خودرو-مه-ابر	×	×	×	×	×	[۱۴]
خودرو-مه	✓	×	✓	×	×	[۱۵]
خودرو-مه-ابر	×	✓	✓	✓	×	[۱۶]
خودرو-مه	×	×	✓	×	×	[۱۷]
خودرو-مه	×	✓	×	×	×	[۱۸]
خودرو-لبه-ابر	×	✓	✓	✓	×	[۱۹]
خودرو-MEC-ابر	×	×	×	×	✓	[۲۰]
خودرو-لبه	✓	×	✓	×	×	[۲۱]
خودرو-مه-ابر	×	×	×	×	×	[۲۲]
خودرو-مه	×	✓	✓	×	×	[۲۳]
خودرو-مه-ابر	✓	✓	✓	✓	✓	روش پیشنهادی

یادداشت: ✓ =پارامترهایی که در نظر گرفته شده اند , ×=پارامترهایی که در نظر گرفته نشده اند

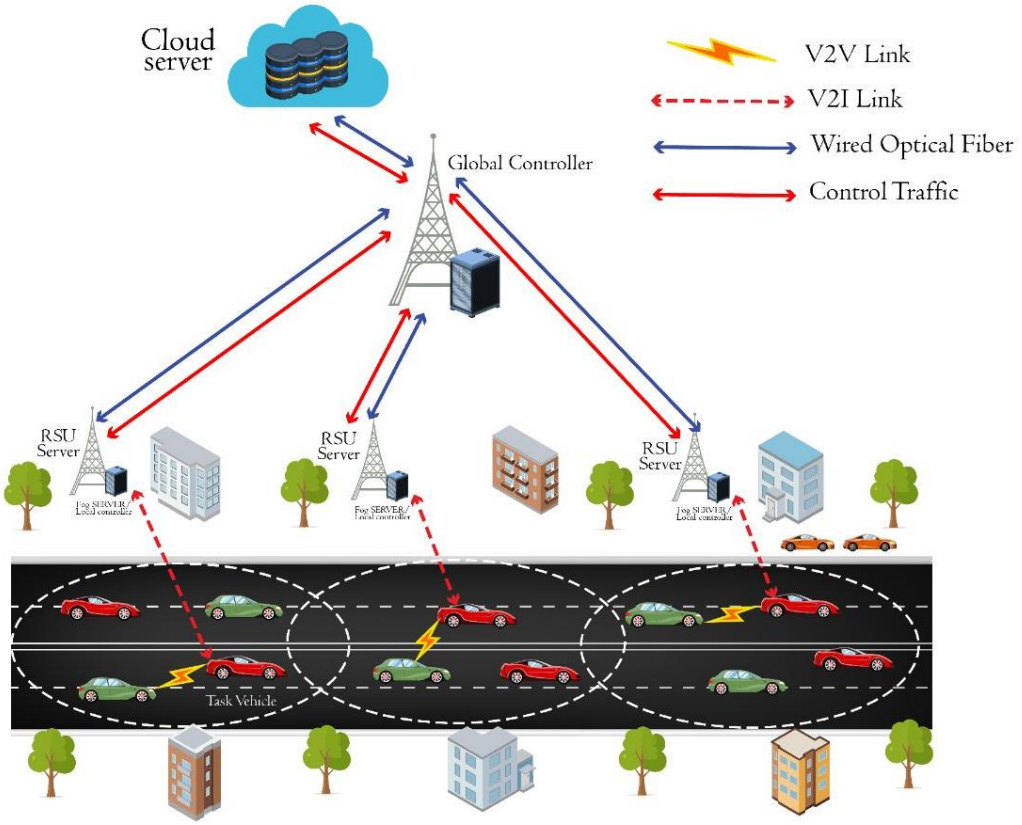
۲-روش پیشنهادی

معماری سلسله مراتبی مبتنی بر شبکه نرم‌افزار محور ، برای محیط رایانش مه خودرویی در شکل ۱ نشان داده شده است. هر منطقه شامل چند خوشه است. در هر خوشه، یک واحد کنارجاده ای، خوذروهای پارک شده کنار جاده(RSPVs) [1]و یک خودروی در حال حرکت با قابلیت محاسباتی بالا مانند اتوبوس به عنوان گره‌های مه عمل می‌کنند. ما فرض می‌کنیم که یک کنترل‌کننده محلی که عامل روی آن قرار دارد، یک خوشه را مدیریت می‌کند. کنترل محلی روی هر واحد کنار جاده مستقر شده است. در لایه اول کنترل‌کننده‌های محلی قرار دارند و در لایه دوم کنترل‌کننده سراسری قرار دارد. همانطور که قبلا گفته شد، ما از یک تکنیک برون‌سپاری وظیفه فدرال بر خط چند عامله مبتنی بر یادگیری عمیق برای پیاده‌سازی برون‌سپاری وظیفه استفاده می‌کنیم. اعمال رویکردهای یادگیری فدرال بر روی شبکه‌های نرم‌افزار محور می‌تواند مزایایی را داشته باشد. یک صفحه کنترل شبکه نرم‌افزار محور می‌تواند چندین کنترل‌کننده داشته باشد که در لایه‌ها سازماندهی شده‌اند؛ کنترل‌کننده‌های لایه پایینی می‌توانند در لبه شبکه قرار گیرند تا با عدم تقارن در سوئیچ‌ها و میزبان‌های متصل مقابله کنند و کنترل‌کننده در لایه بالایی می‌تواند کل شبکه را به صورت مرکزی و سراسری نظارت کند. اعمال یادگیری فدرال در شبکه‌های نرم‌افزار محور با یک صفحه کنترل توزیع‌شده لایه‌ای، می‌تواند ضمن بهبود همکاری بین شرکت‌کنندگان برای تولید مدل‌های با کیفیت بالاتر بر روی شبکه‌های نامتقارن، از حریم خصوصی داده‌های هر شرکت‌کننده محافظت کند[۲۴].

هنگامی که یک خودروی در حال حرکت نمی‌تواند وظایف را به صورت محلی اجرا کند، مشخصات وظایف را به کنترلر محلی در خوشه خود ارسال می‌کند. پس از آن، کنترل‌کننده محلی سیاست بهینه را برای اختصاص برنامه‌ها به گره‌های مه در خوشه خود انتخاب می‌کند تا مهلت انجام وظایف رعایت شود. یک کنترل‌کننده سراسری شبکه نرم‌افزار محور از تمامی خوشه‌های آن

[1] Road Side Parked vehicles

منطقه آگاهی دارد. در پایان هر قسمت[1] از یادگیری، وزن‌های مدل برون‌سپاری وظیفه که به صورت محلی آموخته شده است، با کنترلر سراسری به اشتراک گذاشته می‌شوند. کنترلر سراسری، وزن‌های مدل محلی را تجمیع کرده و یک مدل سراسری بروز شده را برای کنترل‌های محلی به اشتراک می‌گذارد. در نتیجه، مسئله بهینه‌سازی در لایه پایین‌تر حل می‌شود که در آن کنترلرها یا عامل‌های محلی، مدل برون‌سپاری وظیفه را به صورت توزیع‌شده یاد می‌گیرند.



شکل ۱: معماری توزیع شده پیشنهادی شامل کنترلر سراسری و کنترلرهای محلی است. کنترلرهای محلی منابع محلی را تخصیص می‌دهند و کنترلر سراسری دانش کلّی شبکه را دارد.

Figure 1. The proposed distributed architecture consists of a global controller and local controllers. Local controllers allocate local resources and the global controller has global knowledge of the network.

۲-۱-مدل وظیفه

وظایف در خودروها را می توان به سه دسته وظایف حیاتی، وظایف با اولویت بالا و اولویت پایین تقسیم‌بندی کرد. وظایف حیاتی، وظایف هسته‌ای و مرتبط با ایمنی هستند و باید به صورت محلی در خودروها اجرا شوند. بنابراین، خودروها باید مقداری از منابع محاسباتی را برای وظایف حیاتی محلی رزرو کنند. وظایف با اولویت بالا نشان‌دهنده دسته‌ای از وظایف هستند که محدودیت تأخیر شدیدی دارند، مانند ناوبری، حسگر جاده و غیره. اگر یک وظیفه با اولویت بالا نتواند در حداکثر تأخیر قابل تحمل به پایان برسد، وظیفه با شکست مواجه می‌شود و ممکن است ضررهایی را برای خودرو به همراه داشته باشد. در مقایسه با وظایف با اولویت پایین، وسایل نقلیه باید اطمینان حاصل کنند که وظایف با اولویت بالا ابتدا اجرا می‌شوند. وظایف با اولویت پایین دسته‌ای از وظایف مقاوم در برابر تأخیر هستند، مانند برنامه‌های سرگرمی خودرویی، خدمات ارزش افزوده و غیره. اگر زمان تکمیل یک وظیفه با اولویت پایین از تأخیر مرجع بیشتر شود، نتیجه همچنان قابل استفاده خواهد بود، اما با افزایش زمان اجرا، در دسترس بودن نتیجه کاهش می‌یابد[۴]. در روش پیشنهادی فرض می‌شود که وظایف با اولویت بالا و اولویت پایین

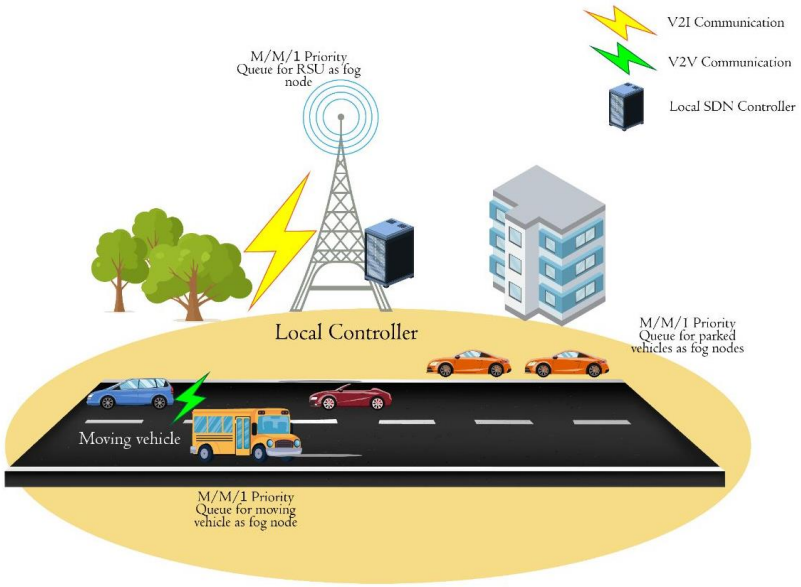
[1] Episode

به‌طور کامل برون‌سپاری می‌شوند. سعی بر این است که وظایف با اولویت بالا در خوشه محلی انجام شوند و تا حد ممکن به ابر منتقل نشوند.

خودروها وظایف محاسباتی برنامه‌ها را تولید می‌کنند که هر وظیفه i مربوط به خودروی v به صورت چندگانه $(x_{i,v},y_{i,v},\lambda_{i,v},d_{i,v},\tau_{i,v},L_{i,v},b_{i,v},B_{i,v},K_{i,v})$ تعریف می‌شوند. که در آن $x_{i,v}$ و $y_{i,v}$ مختصات خودرو v شامل وظیفه i را نشان می‌دهند، $\lambda_{i,v}$ ، نرخ تولید وظیفه محاسباتی، $d_{i,v}$ ، اندازه وظیفه (کد برنامه و پارامترهای مربوطه) برای انتقال، $\tau_{i,v}$ ، حداکثر تأخیر قابل تحمل وظیفه، $L_{i,v}$، مقدار منبع محاسباتی مورد نیاز برای اجرای وظیفه، $b_{i,v}$، بودجه وظیفه، $B_{i,v}$، پهنای باند مورد استفاده وظیفه و $K_{i,v}$ ، اولویت وظیفه را نشان می‌دهد و مقدار آن صفر یا یک است.

۲-۲-مدلسازی سیستم

این بخش، مدل ارتباطی، مدل محاسباتی، مدلسازی ریاضی و الگوریتم پیشنهادی در سیستم مه خودرویی را معرفی می‌کند. از سیستم صف جهت مدلسازی گره‌های مه مختلف استفاده شده است که در شکل ۲ نشان داده شده است.



شکل ۲: گره‌های مه در خوشه محلی شامل خودروی در حال حرکت مانند اتوبوس با ظرفیت محاسباتی بالا، خودروهای پارک شده در کنار خیابان و واحد کنار جاده‌ای با منابع محاسباتی بیکار که همگی به صورت صف اولویت دار M/M/1 مدل شده‌اند.

Figure 2. The fog nodes in the local cluster include moving vehicles such as buses with high computational capacity, cars parked on the side of the street, and roadside units with idle computational resources, all of which are modeled as M/M/1 priority queues.

۲-۲-۱-برون‌سپاری وظیفه خودرو به خودرو[1]

به منظور کاهش بار محاسباتی واحدهای کنار جاده ای، هنگامی که دستگاه‌های اطراف خودرو منابع محاسباتی آزادی دارند که می‌توانند مورد استفاده قرار گیرند، خودروها می‌توانند از طریق ارتباط خودرو به خودرو با یکدیگر ارتباط برقرار کنند تا وظایف را واگذار کرده و فشار محاسباتی را کاهش دهند. این مقاله استفاده از خودروهای در حال حرکت و پارک شده را پیشنهاد می‌دهد. به‌طوری که آنها می‌توانند به عنوان گره مه عمل کنند. نرخ انتقال داده بین خودروی v و خودروی v' برای وظیفه i را می‌توان به صورت رابطه ۱ بیان کرد:

$$tr_{v,v'}=B_{i,v}\log_2\left(1+\frac{|h_{v,v'}|^2 p_{v,v'}}{\sigma(dis_{v,v'})^{\bar{\omega}}}\right) \quad (۱)$$

[1] Vehicle-to-Vehicle task offloading

که در آن B_i، نشان دهنده پهنای باند کانال، $h_{v,v'}$، ضریب محوشدگی کانال، $p_{v,v'}$، نشان دهنده توان انتقال خودروی v برای ارسال به خودروی v' ، σ ، نشان دهنده توان نویز سفید گاوسی، $\overline{\omega}$، نشان دهنده توان تلفات مسیر و $dis_{v,v'}$ ، فاصله بین خودروی v و خودروی مه در حال حرکت v' است. این فاصله را می‌توان با رابطه ۲ بیان کرد:

$$dis_{v,v'}=\sqrt{(x_v-x_{v'})^2-(y_v-y_{v'})^2} \quad (۲)$$

که در آن x_v و y_v مختصات خودروی v را نشان می‌دهند در حالی که $x_{v'}$و$y_{v'}$ مختصات خودروی مه در حال حرکت v' را نشان می‌دهند. زمان انتقال وظیفه i (زمان بارگذاری) به گره v' را می‌توان طبق رابطه۳ محاسبه کرد:

$$T^u_{v,v'}=\frac{d_{i,v}}{tr_{v,v'}} \quad (۳)$$

در روش پیشنهادی، خودروهای پارک شده و در حال حرکت توسط صف‌های اولویت دار M/M/1 مدل‌سازی می‌شوند. کنترل‌کننده محلی در خوشه، نرخ رسیدن دستورالعمل‌های $\Delta^1_{v'}$ و $\Delta^2_{v'}$ را به ترتیب برای کلاس ۱ و کلاس ۲ به خودروی مه اختصاص می‌دهد. برای یک وظیفه با نرخ ورود $\lambda_{i,v}$ و میزان پردازش $L_{i,v}$، آنگاه $\Delta^1_{v'}$ و $\Delta^2_{v'}$ را می‌توان به ترتیب به صورت رابطه ۴ و رابطه ۵ محاسبه کرد:

$$\Delta^1_{v'}=\sum_i L_{i,v}\lambda_{i,v}\ K_{i,v}x_{i,v'} \quad (۴)$$

$$\Delta^2_{v'}=\sum_i L_{i,v}\lambda_{i,v}\ (1-K_{i,v})x_{i,v'} \quad (۵)$$

مجموع کل نرخ رسیدن دستورالعمل‌ها برای خودروی مه v' در خوشه به صورت رابطه ۶ محاسبه می‌شود:

$$\Delta_{v'}=\Delta^1_{v'}+\Delta^2_{v'} \quad (۶)$$

میانگین زمان انتظار در صف برای وظیفه با اولویت بالا و اولویت پایین را می‌توان به ترتیب به صورت معادلات ۷ و ۸ محاسبه کرد:

$$W^1_{i,v'}=\frac{\Delta_{v'}}{\mu_{v'}(\mu_{v'}-\Delta^1_{v'})} \quad (۷)$$

$$W^2_{i,v'}=\frac{\Delta_{v'}}{(\mu_{v'}-\Delta^1_{v'})(\mu_{v'}-\Delta^1_{v'}-\Delta^2_{v'})} \quad (۸)$$

$T^w_{i,v'}$، را می‌توان به صورت رابطه ۹ محاسبه کرد:

$$T^w_{i,v'}=\begin{cases}W^1_{i,v'} & \text{if } K_{i,v}=1\\ W^2_{i,v'} & \text{if } K_{i,v}=0\end{cases} \quad (۹)$$

بنابراین، میانگین زمان مورد نیاز خودروی مه v' برای تکمیل وظیفه i را می‌توان به صورت رابطه ۱۰ بیان کرد:

$$T^p_{i,v'}=\frac{1}{\mu_{v'}} \quad (۱۰)$$

$\mu_{v'}$، ظرفیت محاسباتی گره v' است. ما فرض می‌کنیم هزینه واحد به ازای هر MIPS[1] برای گره مه v' برابر با $C_{v'}$ است. بنابراین، میانگین هزینه برون‌سپاری محاسباتی را می‌توان با رابطه 11 محاسبه کرد:

$$C_{i,v'}=C_{v'}\Delta_{v'} \quad (۱۱)$$

میانگین مصرف انرژی مورد نیاز ارتباطی و محاسباتی برای تکمیل وظیفه با استفاده از رابطه ۱۲ محاسبه می‌شود:

$$E_{i,v'}=E_{v'}\Delta_{v'}+p_{v,v'}T^u_{v,v'} \quad (۱۲)$$

[1] Million Instruction Per second

که در آن $E_{v'}$ نشان دهنده مصرف انرژی به ازای هر MIPS برای گره مه v' است. میانگین زمان برون‌سپاری یا میانگین زمان پاسخ وظیفه i که توسط گره مه خودرویی v' پردازش می‌شود، می‌تواند در کنترل‌کننده محلی محاسبه شود که عموماً به صورت رابطه ۱۳ تعریف می‌شود:

$$tf_{i,v'}=T^{u}_{i,v'}+T^{w}_{i,v'}+T^{p}_{i,v'}+T^{d}_{i,v'} \quad (۱۳)$$

$T^{u}_{i,v'}$ و $T^{d}_{i,v'}$ به ترتیب زمان‌های بارگذاری و بازگشت پاسخ در خوشه محلی توسط گره مه خودرویی v' هستند. برای سادگی، زمان بارگذاری و بازگشت پاسخ از گره مه با یکدیگر برابر در نظر گرفته می‌شوند.

۲-۲-۲-برون‌سپاری وظیفه خودرو به زیرساخت[1]

در این روش، خودروها می‌توانند وظایف خود را به واحدهای کنار جاده‌ای(RSUs) [2]یا ابر برون‌سپاری نمایند. ارتباط بین خودروها به واحد کنار جاده‌ای از طریق ارتباط بیسیم و از واحد کنار جاده‌ای به ابر از طریق فیبر نوری است. نرخ انتقال داده بین خودروی v و واحد کنار جاده‌ای طبق معادله ۱۴ محاسبه می‌شود:

$$tr_{v,r}=B_{i,v}\log_2\left(1+\frac{|h_{v,r}|^2 p_{v,r}}{\sigma(dis_{v,r})^{\bar{\omega}}}\right) \quad (۱۴)$$

که در آن $h_{v,r}$، ضریب محوشدگی کانال، $p_{v,r}$ ، نشان دهنده توان انتقال خودرو ، σ، نشان دهنده توان نویز سفید گاوسی، ω ، نشان دهنده توان تلفات مسیر و $dis_{v,r}$، فاصله بین خودروی v و واحد کنار جاده‌ای r است. این فاصله را می‌توان با رابطه ۱۵ بیان کرد:

$$dis_{v,r}=\sqrt{(x_v-x_r)^2-(y_v-y_r)^2} \quad (۱۵)$$

زمان انتقال وظیفه i در گره مه خودرویی v به واحد کنار جاده‌ای r را می‌توان بر طبق رابطه ۱۶ محاسبه کرد:

$$T^{u}_{v,r}=\frac{d_{i,v}}{tr_{v,r}} \quad (۱۶)$$

مشابه با خودروهای مه، واحدهای کنار جاده‌ای نیز به‌صورت صف M/M/1 در نظر گرفته می‌شود. بنابراین، نرخ رسیدن دستورالعمل‌های Δ^1_r و Δ^2_r ، به ترتیب برای کلاس ۱ و کلاس ۲ به واحد کنار جاده‌ای به صورت زیر محاسبه می‌شوند:

$$\Delta^1_r=\sum_i L_{i,v}\lambda_{i,v}\ K_{i,v}x_{i,r} \quad (۱۷)$$

$$\Delta^2_r=\sum_i L_{i,v}\lambda_{i,v}\ (1-K_{i,v})x_{i,r} \quad (۱۸)$$

که در این صورت کل نرخ رسیدن دستورالعمل‌ها به واحد کنار جاده ای مجموع نرخ دستورالعمل‌های کلاس ۱ و کلاس ۲ است که می‌توان مشابه با فرمول‌های ۷و۸، میانگین زمان انتظار با اولویت بالا و پایین را بدست آورد که به صورت $W^1_{i,r}$ و $W^2_{i,r}$ نمادگذاری شده اند. همچنین میانگین زمان مورد نیاز برای پردازش وظیفه i ، به‌صورت رابطه ۱۹ محاسبه می‌گردد:

$$T^{p}_{i,r'}=\frac{1}{\mu_r} \quad (۱۹)$$

μ_r، ظرفیت محاسباتی گره r است. همچنین مشابه خودروهای پارک شده و در حال حرکت، میانگین مصرف انرژی، میانگین هزینه و میانگین زمان برون‌سپاری وظیفه از خودرو به واحد کنار جاده ای به ترتیب با استفاده از رابطه‌های ۲۰ تا ۲۲ بدست می‌آیند:

$$E_{i,r}=E_r\ \Delta_r+p_{v,r}T^{u}_{v,r} \quad (۲۰)$$

$$C_{i,r}=C_r\ \Delta_r \quad (۲۱)$$

[1] Vehicle-to-Infrastructure task offloading

[2] Road Side Units(RSUs)

$$tf_{i,r}=T^{u}_{i,r}+T^{w}_{i,r}+T^{p}_{i,r}+T^{d}_{i,r} \quad (۲۲)$$

$T^{u}_{i,r}$ و $T^{d}_{i,r}$ به ترتیب زمان‌های بارگذاری و بازگشت پاسخ در خوشه محلی توسط واحد کنار جاده‌ای r هستند. $E_{i,r}$، $C_{i,r}$ و $tf_{i,r}$ به ترتیب میانگین مصرف انرژی، میانگین هزینه و میانگین زمان برون‌سپاری وظیفه خودرو به واحد کنار جاده‌ای هستند.

در صورتیکه منابع مه محلی پاسخگوی نیازمندی‌های وظیفه نباشند، کنترلر محلی وظیفه را به ابر منتقل می‌کند. از آنجا که ارتباط بین واحدهای کنار جاده‌ای با مرکز ابر از طریق فیبر نوری است، بنابراین سرعت ارسال داده‌ای ثابت در نظر گرفته می‌شود. بنابراین زمان انتقال داده‌ها از واحد کنار جاده‌ای به ابر از طریق رابطه زیر بدست می‌آید:

$$T^{u}_{r,c}=\frac{d_{i,v}}{tr_{r,c}} \quad (۲۳)$$

در این مقاله، فرض می‌شود که مرکز ابر با c ماشین مجازی به‌صورت صف M/M/c مدل‌سازی می‌شوند که میانگین زمان انتظار در صف می‌تواند بر اساس فرمول‌های زیر محاسبه گردد:

$$T^{w}_{i,c}=\frac{C(c,\rho)}{(c\mu_c-\Delta_c)} \quad (۲۴)$$

$$C(c,\rho)=\frac{\left(\frac{c\rho^{c}}{c!}\right)\left(\frac{1}{1-\rho}\right)}{\sum_{k=0}^{c-1}\left(\frac{k\rho^{k}}{k!}\right)+\left(\frac{c\rho^{c}}{c!}\right)\left(\frac{1}{1-\rho}\right)} \quad (۲۵)$$

$C(c,\rho)$ ، فرمول C ارلنگ [۲۵] و $\rho=\frac{\Delta_c}{c\mu_c}$ نرخ شدّت ترافیک در سرور مجازی c است. همچنین میانگین زمان مورد نیاز برای پردازش وظیفه i در ابر توسط سرور مجازی c، به‌صورت معادله ۲۶ محاسبه می‌گردد:

$$T^{p}_{i,c}=\frac{1}{\mu_c} \quad (۲۶)$$

μ_r، ظرفیت محاسباتی گره r است. با توجه به اینکه داده‌ها ابتدا باید به واحد کنار جاده‌ای و سپس به ابر منتقل شوند، بنابراین میانگین مصرف انرژی، میانگین هزینه پردازش و میانگین زمان برون‌سپاری وظیفه از خودرو به ابر به ترتیب با استفاده از رابطه‌های ۲۷ تا ۲۹ بدست می‌آیند:

$$E_{i,c}=p_{v,r}T^{u}_{v,r}+p_{r,c}T^{u}_{r,c}+E_c\,\Delta_c \quad (۲۷)$$

$$C_{i,c}=C_c\,\Delta_c \quad (۲۸)$$

$$tf_{i,c}=T^{u}_{i,r}+T^{u}_{r,c}+T^{w}_{i,r}+T^{p}_{i,c}+T^{d}_{i,c} \quad (۲۹)$$

$T^{u}_{i,c}$ و $T^{d}_{i,c}$ به ترتیب زمان‌های بارگذاری و بازگشت پاسخ از ابر هستند. $E_{i,c}$، $C_{i,c}$ و $tf_{i,c}$ به ترتیب میانگین مصرف انرژی، میانگین هزینه و میانگین زمان برون‌سپاری وظیفه خودرو به ابر هستند. در روش پیشنهادی، فرض می‌شود که محدودیتی برای مصرف انرژی و همچنین پهنای باند، برای ارسال وظیفه به ابر وجود ندارد.

۲-۲-۳-مدلسازی ریاضی

این زیر بخش، تابع هدف و محدودیت‌های مسئله برون‌سپاری وظیفه را بیان می‌کند. تابع هدف براساس نرخ وظایف انجام شده با اولویت بالا و پایین را با توجه به محدودیت‌های تاخیر، انرژی، هزینه و پهنای باند خطوط ارتباطی است. با توجه به مدل‌های برون‌سپاری مختلف ذکر شده، سه متغیر تصمیم باینری نحوه انتخاب سرور مقصد را در برش زمانی t بیان می‌کنند که به‌صورت زیر است:

$$X_{i,v',t}, X_{i,r,t}, X_{i,c,t} \in \{0,1\} \quad (۳۰)$$

$X_{i,v',t}$، متغیر تصمیم باینری برای انتخاب خودروی مه محلی، $X_{i,r,t}$ ،متغیر تصمیم باینری برای انتخاب واحد کنار جاده‌ای و $X_{i,c,t}$، متغیر تصمیم باینری برای انتخاب ماشین مجازی در ابر است. مجموعه‌ی خودروها، خودروهای مه، ماشین‌های مجازی، و وظایف با استفاده از رابطه‌های ۳۱ تا ۳۴ بیان می‌شود:

$$V=\{vehicle_1, vehicle_2,..., vehicle_v\} \quad (۳۱)$$

$$V'=\{moving\ vehicle_1, parked\ vehicle_2,...,parked\ vehicle_n\} \quad (۳۲)$$

$$C=\{virtual\ machine_1, virtual\ machine_2,...,virtual\ machine_c\} \quad (۳۳)$$

$$I=\{task_1, task_2,...,task_m\} \quad (۳۴)$$

تابع هدف براساس رابطه ۳۵ بیان می‌شود که در هر برش زمانی t، ماکزیمم نرخ وظایف پذیرفته شده مربوط به خودروی v به عنوان تابع هدف در نظر گرفته می‌شود:

$$max \quad Acc_v^t=\frac{\sum_{i\in I(t)} \psi_{i,v,t}}{I(t)} \quad (۳۵)$$

s.t. (c1) $X_{i,v',t}, X_{i,r,t}, X_{i,c,t} \in \{0,1\}, \quad \forall i \in I, \forall v' \in V', \forall c \in C, \forall t \in T$

(c2) $X_{i,v',t}+X_{i,r,t}+X_{i,c,t} \le 1, \quad \forall i \in I, \forall v' \in V', \forall c \in C, \forall t \in T$

(c3) $\psi_{i,v,t} \in \{0,1\}, \quad \forall i \in I, \forall v \in V, \forall t \in T$

(c4) $$\sum_{v' \in V'} X_{i,v',t}\, tf_{i,v'} + X_{i,r,t}\, tf_{i,r} + \sum_{c\in C} X_{i,c,t}\, tf_{i,c} \le \tau_{i,v}, \quad \forall i \in I, \forall v \in V, \forall v' \in V', \forall c \in C, \forall t \in T$$

(c5) $$\sum_{v' \in V'} X_{i,v',t}\, C_{i,v'} + X_{i,r,t}\, C_{i,r} + \sum_{c\in C} X_{i,c,t}\, C_{i,c} \le b_{i,v}, \quad \forall i \in I, \forall v' \in V', \forall c \in C, \forall t \in T$$

(c6) $B_{i,v} \le \sum_{v' \in V'} X_{i,v',t}\, B_{i,v'} + X_{i,r,t}\, B_{i,r}, \ \forall i \in I, \forall v \in V, \forall v' \in V', \forall t \in T$

(c7) $\sum_{v' \in V'} X_{i,v',t} E_{v'}\, \Delta_{v'} + X_{i,r,t} E_r\, \Delta_r \le \sum_{v' \in V'} X_{i,v',t} E_{v'}^{total} + X_{i,r,t} E_r^{total}$,

$\forall i \in I, \forall v \in V, \forall v' \in V', \forall t \in T$

(c8) $\sum_{v' \in V'} X_{i,v',t}\, \Delta_{v'} + X_{i,r,t}\, \Delta_r + \sum_{c\in C} X_{i,c,t}\, \Delta_{i,c} \le \sum_{v' \in V'} X_{i,v',t}\, \mu_{v'} + X_{i,r,t} \mu_r + \sum_{c\in C} X_{i,c,t}\, \mu_{i,c}$,

$\forall i \in I, \forall v' \in V', \forall c \in C, \forall t \in T$

(c9) $\sum_{v' \in V'} X_{i,v',t} = 0, \quad \forall i \in I, \forall v' \in V', \forall t \in T$

(c10) $$\sum_{c\in C} X_{i,c,t} = 0, \quad \forall i \in I, \forall c \in C, \forall t \in T$$

محدودیت (c1)، نشان می‌دهد که سه نوع اجرای وظیفه در نظر گرفته می‌شوند. محدودیت (c2)، بیان می‌کند که تنها یک نوع اجرای وظیفه باید برای هر وظیفه در نظر گرفته شود. محدودیت (c3)، بیان می‌کند که متغیر تصمیم $\psi_{i,v,t}$، مقدار یک را می‌گیرد وقتی که وظیفه i در خودروی v به‌طور موفق آمیز اجرا شود و در غیر صورت مقدارش صفر می‌گردد. محدودیت (c4)، توضیح می‌دهد که زمان اجرای وظیفه باید کمتر یا مساوی از مقدار تعیین شده باشد. محدودیت (c5)، بیان می‌کند که هزینه انجام وظیفه باید کمتر یا مساوی از بودجه تعریف شده برای وظیفه باشد. محدودیت (c6)، نشان می‌دهد که پهنای باند مورد استفاده برای ارسال وظیفه به گره مه باید کمتر یا مساوی از پهنای باند تخصیص داده شده به گره مه باشد. محدودیت (c7)، بیان می‌کند که انرژی مصرف شده برای اجرای وظیفه باید کمتر یا مساوی از انرژی گره مه باشد. محدودیت (c8)، توضیح می‌دهد که منابع محاسباتی مورد نیاز وظایف باید کمتر یا مساوی ظرفیت محاسباتی گره مه یا سرور ابر باشد. محدودیت (c9)ومحدودیت (c10) به ترتیب بیان می‌کنند که یک و فقط یک خودروی مه و یا سرور ابر به وظیفه تخصیص داده می‌شود.

بر اساس جدول ۲، نمادگذاری‌های متداول در مدلسازی ریاضی را خلاصه می‌کند.

جدول ۲: نمادهای متداول استفاده شده در مدلسازی

Table 2. Common symbols used in modeling

تعریف	نماد
متغیر تصمیم‌گیری باینری برای انتخاب خودروی مه v' در برش زمانی t	$X_{i,v',t}$
متغیر تصمیم‌گیری باینری برای انتخاب واحد کنار جاده‌ای r در برش زمانی t	Xi,r,t
متغیر تصمیم‌گیری باینری برای انتخاب سرور ابر c در برش زمانی t	$X_{i,c,t}$
متغیر تصمیم‌گیری باینری برای پذیرش وظیفه i در خودروی v	$\psi_{i,v,t}$

زمان برون‌سپاری وظیفه i به خودروی مه v'(ثانیه)	$tf_{i,v'}$
زمان برون‌سپاری وظیفه i به واحد کنار جاده ای r (ثانیه)	tfi,r
زمان برون‌سپاری وظیفه i به سرور ابر c(ثانیه)	$tf_{i,c}$
مهلت زمانی وظیفه i در خودروی v(ثانیه)	τi,v
هزینه برون‌سپاری وظیفه i به خودروی مه v'(واحد هزینه)	$C_{i,v'}$
هزینه برون‌سپاری وظیفه i به واحد کنار جاده ای r(واحد هزینه)	Ci,r
هزینه برون‌سپاری وظیفه i به سرور ابر c(واحد هزینه)	$C_{i,c}$
بودجه وظیفه i در خودروی v(واحد هزینه)	$b_{i,v}$
پهنای باند مورد نیاز وظیفه i در خودروی v(مگا هرتز)	$B_{i,v}$
پهنای باند تخصیص داده شده به خودروی مه v'(مگا هرتز)	$B_{v'}$
پهنای باند تخصیص داده شده به واحد کنار جاده ای r(مگا هرتز)	B_r
انرژی مورد نیاز به ازای هر MIPS برای خودروی مه v'(ژول بر ثانیه)	$E_{v'}$
انرژی مورد نیاز به ازای هر MIPS برای واحد کنار جاده ای r(ژول بر ثانیه)	E_r
نرخ رسیدن کل دستورات به خودروی مه v'(MIPS)	$\Delta_{v'}$
نرخ رسیدن کل دستورات به واحد کنار جاده ای r(MIPS)	Δ_r
نرخ رسیدن کل دستورات به سرور ابر c(MIPS)	Δ_c
انرژی تخصیص داده شده به خودروی مه v'(ژول)	$E_{v'}^{total}$
انرژی تخصیص داده شده به واحد کنار جاده ای r(ژول)	E_r^{total}
ظرفیت محاسباتی خودروی مه v'(MIPS)	$\mu_{v'}$
ظرفیت محاسباتی واحد کنار جاده ای r(MIPS)	μ_r
ظرفیت محاسباتی سرور ابر c(MIPS)	μ_c

۳-فرمول‌بندی یادگیری و راه‌حل پیشنهادی

مسئله بهینه‌سازی مطرح شده، یک مسئله بهینه‌سازی عدد صحیح مختلط[1] است و چنین مسائلی معمولاً، مسائل NP سخت[2] در نظر گرفته می‌شوند[۲۶]. بنابراین، ما از یادگیری تقویتی برای حل مسئله بهینه‌سازی فرموله شده استفاده می‌کنیم. در این بخش، ابتدا پیاده‌سازی یادگیری تقویتی را شرح می‌دهیم و سپس راه‌حل یادگیری تقویتی عمیق را بررسی می‌کنیم.

۳-۱-پیاده‌سازی یادگیری تقویتی

ما از یک مدل فرآیند تصمیم‌گیری مارکوف متناهی (MDP) برای ثبت دینامیک انتقال حالت شبکه استفاده می‌کنیم. یک MDP را می‌توان با مجموعه‌ای از حالت‌های سیستم، مجموعه‌ای از اقدامات و مجموعه‌ای از توابع پاداش با مقدار حقیقی تعریف کرد[۲۷]. عامل‌های خودرویی با محیط خودرویی تعامل می‌کنند تا تصمیمات برون‌سپاری خود را بر اساس تجربیات قبلی بهبود بخشند. ما ارتباط بین کنترلر های محلی و سرورهای مه و ابر را به عنوان یک فرآیند MDP به‌صورت $(S,A,R,\Pi_{SS'})$ فرموله می‌کنیم، که در آن S، نشان دهنده فضای حالت، A، نشان دهنده فضای اقدام، R، نشان دهنده فضای پاداش و $\Pi_{SS'}$ ، نشان دهنده احتمال انتقال است. نمادگذاری‌ها و توابع مختلف مورد استفاده به شرح زیر است:

- **فضای حالت:** یک فضای حالت شامل تمام پیکربندی‌های شبکه در یک سیستم است. مجموعه وظایف در برش زمانی t به‌صورت $\Phi^t=\{\phi_1^t,\phi_2^t,...,\phi_I^t\}$ نمادگذاری می‌شوند. مشخصات وظیفه i، در برش زمانی t، را به‌صورت $\phi_i^t=\{x_{i,v}^t,y_{i,v}^t,\lambda_{i,v}^t,d_{i,v}^t,\tau_{i,v}^t,L_{i,v}^t,b_{i,v}^t,B_{i,v}^t,K_{i,v}^t\}$ نمادگذاری می‌کنیم. گره‌های مه و سرور ابری تنظیم‌شده در بازه زمانی t به صورت $F^t=\{F_1^t,F_2^t,...,F_{V'+1}^t\}$ و $C^t=\{C_1^t,C_2^t,...,C_C^t\}$ نشان داده می‌شوند. مشخصات گره مه با اندیس j به‌صورت $F_j^t=\{x_j^t,y_j^t,\mu_j^t,E_j^t,C_j^t,B_j^t\}$ تعریف می‌شود که به ترتیب x_j^t و y_j^t ، مختصات گره مه، μ_j^t ، ظرفیت محاسباتی گره مه، E_j^t ، مقدار انرژی گره مه، C_j^t، هزینه پردازش وظیفه و B_j^t، پهنای باند تخصیص داده شده به گره مه می‌باشند. همچنین مشخصات سرورابر با اندیس c به‌صورت $C_C^t=\{\mu_c,E_c,C_c\}$ نمادگذاری شود که به ترتیب μ_c، مشخص کننده ظرفیت

[1] Mixed-integer linear optimization problem

[2] NP-hard problems

محاسباتی ماشین مجازی، E_c، مقدار انرژی ماشین مجازی و C_c، هزینه استفاده از ماشین مجازی است. بنابراین، بردار حالت سیستم در بازه زمانی t را می‌توان به صورت زیر تعریف کرد:

$$S^{(t)}=[\phi^t,F^t,C^t] \quad (۳۶)$$

- **فضای عمل:** در رویکرد پیشنهادی، یک عامل در هر لحظه زمانی t، استراتژی برون‌سپاری وظیفه را تعیین می‌کند. بنابراین، یک عمل در لحظه زمانی t به صورت زیر تعریف می‌شود:

$$a^{(t)}=[X_{i,1,t},X_{i,2,t},\ldots,X_{i,v',t},X_{i,v'+1,t},\ldots,X_{i,v'+1+c,t}] \quad (۳۷)$$

- **تابع پاداش:** تابع پاداش بر اساس نرخ پذیرش وظایف، مصرف انرژی، تاخیر و هزینه است. به طور خاص، پاداش $r^{(t)}$ برای برنامه i که از سیاست $\pi(a^{(t)}|r^{(t)})$ پیروی می‌کند، می‌تواند به صورت معادله ۳۸ تا ۴۰ بیان شود:

$$r^{(t)} = Acc_v^t + penalty_v^t \quad (۳۸)$$

$$penalty_v^t = \sum_{i\in I(t)} \text{penalty}_{i,v,t} \quad (۳۹)$$

$$\text{penalty}_{i,v,t} = \begin{cases} 0 & \text{if } \sum_{v'\in V'} X_{i,v',t}\, tf_{i,v'} + X_{i,r,t}\, tf_{i,r} + \sum_{c\in C} X_{i,c,t}\, tf_{i,c} \le \tau_{i,v} \\ -\omega & \text{if } \sum_{v'\in V'} X_{i,v',t}\, tf_{i,v'} + X_{i,r,t}\, tf_{i,r} + \sum_{c\in C} X_{i,c,t}\, tf_{i,c} > \tau_{i,v} \text{ and } \kappa_{i,v}=1 \end{cases} \quad (۴۰)$$

در اینجا $-\omega$ ، یک عدد صحیح منفی است. در واقع در روش پیشنهادی سعی شده که وظایف با اولویت بالا بیشتر پذیرش شوند.

۳-۲-مقدمات

با توجه به دانش محدود در مورد احتمال انتقال بین حالت‌ها و فضای قابل توجه حالت-عمل در شبکه، برنامه‌نویسی پویا[1]ی سنتی نمی‌تواند سیاست بهینه را به طور موثر پیدا کند. بنابراین، ما از یادگیری تقویتی عمیق (DRL) را برای حل مسئله انتخاب سرور پیشنهادی اتخاذ می‌کنیم[۲۸]. ما یادگیری Q [2]عمیق را به دلایل مختلف به عنوان یک رویکرد مبتنی بر ارزش اتخاذ می‌کنیم. اولاً، سیستم زمان گسسته‌ای دارد و در هر بازه زمانی در نظر گرفته می‌شود. دوما، تمام منابع، مانند قابلیت‌های محاسباتی، بر اساس مقادیر گسسته مدل‌سازی شده و به سطوح گسسته تقسیم می‌شوند. سوما، راه‌حل یادگیری ما باید فقط یک سرور را برای هر بازه زمانی به عنوان نتیجه یادگیری عمیق انتخاب کند. چهارما، یکی از اهداف استفاده از یک شبکه Q عمیق[3]، به حداکثر رساندن پاداش در مرحله‌ها است. رویکرد یادگیری-Q می‌تواند برای یافتن یک سیاست تصمیم‌گیری بهینه مورد استفاده قرار گیرد. در رویکرد یادگیری Q، یک تابع Q تعریف می‌شود که عامل را به سمت راه‌حل بهینه هدایت می‌کند. تابع Q به صورت بازگشتی با استفاده از معادله بلمن[4] محاسبه می‌شود:

$$Q_{(s^t,a^t)}=Q_{(s^t,a^t)}+\alpha\left[r^t+\gamma \max_{a^{t+1}} Q_{(s^{t+1},a^{t+1})}-Q_{(s^t,a^t)}\right] \quad (۴۱)$$

هر عامل می‌تواند یک جدول Q را نگهداری کند که توسط جدول $Q_{(s^t,a^t)}$ داده شده است، که در آن s^t وضعیت فعلی در بازه زمانی t و a^t عملی است که توسط عامل در بازه زمانی t انجام می‌شود. در فرمول بالا، α نرخ یادگیری، γ ضریب تخفیف، s^{t+1} حالت بعدی و a^{t+1} عملی در حالت s^{t+1} است که مقدار Q را به حداکثر می‌رساند. یادگیری Q یک راه حل امیدوارکننده است که می‌تواند به طور موثر به سیاست بهینه برسد، زمانی که فضاهای حالت و عمل کوچک هستند. با این حال، در اکثر محیط‌های دنیای واقعی، فضاهای حالت و عمل بسیار بزرگ هستند و بنابراین نگهداری جدول Q به فضای ذخیره‌سازی عظیمی

[1] Dynamic Programming
[2] Q-learning
[3] Q-network
[4] Bellman equation

نیاز دارد. رویکرد یادگیری Q در الگوریتم ۱ شرح داده شده است. در ابتدا، مقادیر Q برای تمام لحظات زمانی ۰ در نظر گرفته می‌شوند. یک عامل اطلاعات وضعیت را از واحدهای کنار جاده‌ای دریافت می‌کند و با استفاده از سیاست ε -حریصانه [۱] یک عمل را اجرا می‌کند. در سیاست ε -حریصانه، عامل یک اقدام بهینه با احتمال $\varepsilon - 1$ و یک اقدام تصادفی را با احتمال ε انجام می‌دهد. جدول مقدار Q برای T برش زمانی در هر مرحله به‌روزرسانی می‌شود. پاداش، به عنوان میانگین پاداش‌های فوری در داخل یک مرحله محاسبه می‌شود. بنابراین، رویکرد یادگیری Q عمدتاً برای MDPهایی که فضاهای حالت یا عمل کوچکی دارند، استفاده می‌شود.

Algorithm 1: Q-Learning Approach

Input: Set of all tasks and set of all possible associations between controllers and predefined servers.
Output: Efficient offloading decisions
1: Initialize $Q_{(s^t,a^t)}$ to 0 for all time instants ;
2**: for** each episode **do**
3**: for** t=1 to T **do**
4: At time instant t, an agent acquires the state s^t;
5: An agent chooses an action $a^t=\text{argmax}_{a^t}Q(s^t,a^t)$with probability 1-ε or randomly choose action from action space with probility ε;
6: An agent updates $Q_{(s^t,a^t)}$from Eq. 41 ;
7: **end for**
8: **end for**
9: return Q(S,A);

از این رو، از یک رویکرد شبکه یادگیری Q عمیق (DQN) [۲] برای بدست آوردن راه حل بهینه استفاده می‌شود. در DQN، یک تابع مقدار Q، با استفاده از یک شبکه عصبی عمیق (DNN) [۳] با مجموعه پارامتر θ تخمین زده می‌شود که حالت ورودی را به یک عمل نگاشت می‌دهد. جدا از شبکه عصبی اصلی، یک شبکه عصبی هدف برای تثبیت شبکه وجود دارد. پس از تعداد ثابتی از تکرارها، پارامترهای شبکه عصبی هدف از شبکه عصبی اصلی کپی می‌شوند. پارامترهای شبکه عصبی اصلی به طور مکرر برای به حداقل رساندن تابع اتلاف بروزرسانی می‌شوند. تابع اتلاف DQN به صورت زیر تعریف می‌شود:

$$L^t(\theta^t)=\left(y^t-Q(s^t,a^t|\theta^t)\right)^2 \quad (۴۲)$$

که در آن y^t مقدار هدف شبکه هدف برای بازه زمانی t است، و $Q(s^t,a^t|\theta^t)$ مقدار تخمینی شبکه اصلی با پارامتر وزنی θ^t است. مقدار هدف y^t به صورت معادله ۴۳ تعریف می‌شود:

$$y^t=r^t+\gamma\ \max Q'(s^t,a^t|\theta'^t) \quad (۴۳)$$

که در آن θ^t و θ'^t به ترتیب پارامترهای وزنی شبکه‌های اصلی و هدف هستند. یک رویکرد DQN در الگوریتم ۲ شرح داده شده است.

Algorithm 2: DQN Approach

Input: Set of all tasks and set of all possible associations between controllers and predefined servers.
Output: Efficient offloading decisions
1: Initialize Replay Buffer, Paramter set (θ *and* θ') of main and target networks, Update interval U;
2**: for** each episode **do**
3**: for** t=1 to T **do**
4: At time instant t, an DQN agent acquires the state s^t;
5: An agent chooses an action based on the ε –greedy policy;

[1] ϵ-greedy
[2] Deep Q-learning Network (DQN)
[3] Deep Neural Network (DNN)

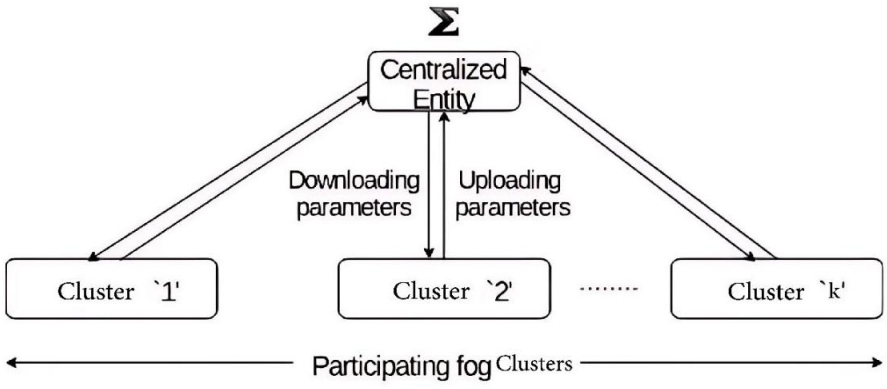
6: Execute action $a_i^{(t)}$, and observe the reward $r_i^{(t)}$ and the next state S^{t+1}
7: Process $S_i^{(t+1)}$ to be the next state;
8: Store transition ($S_i^{(t)}$,$a_i^{(t)}$,$r_i^{(t)}$,$S_i^{(t+1)}$) into replay memory;
9: Sample random mini-batch of transitions ($S_i^{(t)}$,$a_i^{(t)}$,$r_i^{(t)}$,$S_i^{(t+1)}$) from replay memory;
10: Compute y^t from Eq. 43;
11: The DQN agent uses gradient descent to minimize the loss function as shown in Eq. 42 ;
12: After every U steps, update the parameters of target network ;
7: **end for**
8: **end for**
9: return Q(S,A; θ);

در مرحله اول، پارامترهای DQN شبکه‌ی اصلی و هدف مقداردهی اولیه می شوند. هر قسمت دارای T مرحله است. در هر مرحله، عامل DQN وضعیت فعلی s^t را به دست می آورد. پس از به دست آوردن حالت، یک عامل DQN، براساس سیاست ε -حریصانه عملی را انتخاب می کند. چندگانه[1] انتقال در یک بافر پخش مجدد[2] ذخیره می شود. سپس از روش گرادیان نزولی برای بروز سانی وزن‌ها و کم کردن تابع اتلاف استفاده می شود. پس از تعداد ثابتی از مراحل U، پارامترهای 'θ شبکه هدف از طریق شبکه اصلی بروزرسانی می‌شود.

۳-۳-الگوریتم پیشنهادی مبتنی بر یادگیری فدرال

در این کار، ما یک چارچوب برون‌سپاری وظیفه فدرال مبتنی بر DRL چند عاملی ارائه می‌دهیم. راه‌حل پیشنهادی یک معماری چند لایه را ارائه می‌دهد که در آن مدل‌های محلی و سراسری به طور متقابل دانش را به اشتراک می‌گذارند تا تصمیمات برون‌سپاری وظیفه آگاهانه‌ای را بگیرند. مدل‌ها به صورت محلی یاد گرفته می‌گیرند و در کنترلرهای محلی نگهداری می‌شوند. یادگیری فدرال با آموزش محلی در کنترلرهای محلی به کاهش تأخیرهای ارتباطی کمک می‌کند. همچنین حریم خصوصی خودروها و داده‌ها را بهبود می‌بخشد، زیرا مقدار کمی از داده‌های حیاتی به اشتراک گذاشته می‌شود. همچنین احتمال تغییر داده‌ها کمتر است زیرا فقط گرادیان‌ها در شبکه‌ها به اشتراک گذاشته می‌شوند. گردش کار یادگیری فدرال در شکل ۳ نشان داده شده است که شامل دو لایه است: لایه‌های پایین و بالا. لایه بالایی شامل کنترلر سراسری و مبتنی بر فدراسیون کنترلرهای محلی هستند، در حالی که لایه پایینی شامل کنترلرهای محلی هستند. عامل‌ها در لایه پایینی، حالت را به عنوان ورودی خود در نظر می‌گیرند و استراتژی بهینه را با استفاده از تغییر وزن‌ها (پارامترها) در شبکه اصلی یاد می‌گیرند. سپس بروزرسانی‌ها به عامل در لایه بالایی به اشتراک گذاشته می‌شوند که بروزرسانی‌های پارامترهای همه عامل‌ها در محدوده انتقال را تجمیع می‌کند. این اشتراک‌گذاری متقابل بروزرسانی‌ها بین لایه پایینی و بالایی، مدل تصمیم‌گیری برون‌سپاری وظیفه را بهبود می‌بخشد. شبکه سراسری، وزن‌های مدل محلی را تجمیع کرده و یک مدل سراسری بروز شده را با عامل‌های لایه پایینی به اشتراک می‌گذارد.

[1] Tuple
[2] Replay buffer



شکل ۳: فرآیند یادگیری فدرال برای k خوشه و ارتباط بین کنترلرهای محلی و سراسری را نشان می‌دهد.
Figure 3. Shows the federated learning process for k clusters and the connection between local and global controllers.

مسئله بهینه‌سازی در لایه پایین‌تر حل می‌شود که در آن کنترلرهای محلی مدل بروزرسانی وظیفه را به صورت توزیع‌شده یاد می‌گیرند. هر کنترلر محلی با محیط تعامل دارد، به صورت بر خط تجربه جمع‌آوری می‌کند و تصمیمات مربوط به برون‌سپاری خود را بهینه می‌کند. لایه بالایی مسئول تجمیع پارامترهای دریافتی از لایه پایینی و بازگرداندن بروزرسانی‌ها به لایه پایینی است. لایه بالایی با جمع‌آوری بروزرسانی‌ها برای همه کنترلرها در یک صحنه شبیه‌سازی، به عامل‌ها کمک می‌کند تا تجربه آموزشی متنوع‌تری داشته باشند. رویکرد فدرال توسط الگوریتم ۳ بیان شده است.

Algorithm 3: DQN-based Federated Learning Approach

1:**Input:** The Initial set of local agents K;
2:**Output:** Reward $r^{(t)}$ and offloading action $a^{(t)}$;
3:**Repeat**
4: each local controller agent in cluster $k \in K$ updates its parameters set θ_k^t using Algorithm 2;
5: each local controller agent in cluster $k \in K$ sends its parameters set θ_k^t to the centralized controller agent;
6: The centralized controller agent aggregates the local parameters
7: sets of k number of fog clusters as $\theta_k^{t+1} = \sum_{k \in K} \frac{D_k}{D} * \theta_k^t$
8: $t \leftarrow t + 1$;
9:**until** $t > T$

در ابتدا، تعداد عامل های اولیه مقدار دهی می‌شود. یادگیری فدرال شامل تعداد T دور است. در هر دور، کنترلرهای محلی، مجموعه پارامترهای خود را بروزرسانی کرده و برای تجمیع به کنترلر متمرکز ارسال می‌کنند، همانطور که در مراحل ۴ و ۵ نشان داده شده است. در مرحله ۷، کنترلر متمرکز، مجموعه پارامترهای سراسری θ_k^{t+1} را در هر دور t با تجمیع مجموعه پارامترهای محلی θ_k^t هر کنترلر محلی شرکت‌کننده k ∈ K به شرح زیر پیدا می‌کند[۲۹]:

$$\theta_k^{t+1} = \sum_{k \in K} \frac{D_k}{D} * \theta_k^t \quad (۴۴)$$

در اینجا، D_k تعداد وظایف در k امین خوشه است و D اندازه کل وظایف است که به صورت $D = \sum_{k \in K} D_k$ محاسبه می‌شود.

۴-نتایج شبیه‌سازی

این بخش با اتخاذ یک مجموعه داده دنیای واقعی، تنظیمات تجربی را ارائه می‌دهد و کارهای مورد مقایسه را معرفی می‌کند. سپس، نتایج تجربی و بحث‌ها را نشان می‌دهد.

۴-۱-تنظیمات شبیه‌سازی و کارهای مورد مقایسه

از جریان ترافیک واقعی خودروها برای ارزیابی عملکرد رویکرد پیشنهادی مدیریت منابع تصمیم‌گیری سرورهای مه در یک شبکه خودرویی استفاده می‌شود. بنابراین، از یک مجموعه داده واقعی خودروها با استفاده از ردیابی‌های حرکتی تاکسی‌ها در

شهر رم[1]، ایتالیا استفاده شد[۳۰]. این مجموعه داده، شامل مختصات GPS تقریباً ۳۲۰ تاکسی است که طی ۳۰ روز، از ۱ فوریه تا ۲ مارس ۲۰۱۴، جمع‌آوری شده‌اند که شامل موقعیت مکانی رانندگان است که هر ۷ ثانیه جمع‌آوری شده‌اند. مکان تاکسی‌ها از عرض جغرافیایی ۳۹/۳۶ تا عرض جغرافیایی ۴۵/۵۱ و از طول جغرافیایی ۰/۱۴- تا طول جغرافیایی ۴۱/۸۹ متغیر است. ما به طور تصادفی با استفاده از مجموعه داده‌ها، مسیر هر تاکسی در طول یک روز در نظر می‌گیریم. از آنجایی که سرورهای RSU در جاده نزدیک به خودرو قرار دارند، ما فاصله حداکثر و حداقل محدوده مکان خودروها را بر تعداد سرورهای RSU تقسیم می‌کنیم تا مکان هر RSU را پیدا کنیم. علاوه بر این، برای اختصاص یک موقعیت جغرافیایی برای سرور مرکزی، یک مقدار بالا به حداکثر مقدار عرض و طول جغرافیایی موقعیت تاکسی‌ها اضافه می‌کنیم. همچنین، برای ارائه بهینه الگوریتم و مقایسه آن، از سه مدل به عنوان مبنا استفاده می‌کنیم.

روش پایه اول، روش حریصانه[2] برای تخصیص هر وظیفه به یک سرور است. در راه حل حریصانه، وظایف به سرورهایی با قابلیت‌های منابع بالاتر اختصاص داده می‌شوند. روش پایه دوم، یادگیری تقویتی جدولی است، که RL [3] نامیده می‌شود و وظایف را به یک سرور مناسب اختصاص می‌دهد. در روش RL، یک جدول بزرگ، برای حفظ تمام شرایط حالت پیاده‌سازی می‌شود. روش پایه سوم، سیاست برون‌سپاری مبتنی بر SDN ارائه شده در [۸] است. این روش، یک سیاست بهینه سعی می‌کند که با دانش سراسری از شبکه خودوریی مه توسط SDN سراسری، تفاوت زمانی بین زمان پاسخ سرور مقصد و مهلت درخواست را به حداکثر برساند.

ما از کتابخانه PyTorch [۳۱]، برای پیاده‌سازی یادگیری تقویتی عمیق پیشنهادی خود استفاده می‌کنیم. علاوه بر این، ما از چهار لایه از یک شبکه عصبی کاملاً متصل با لایه‌های پنهان با اندازه ۱۲۸ نورون استفاده کرده‌ایم. همچنین از تابع ReLU [۳۲] که یک تابع فعال‌سازی است به لایه‌های مرتبط اضافه شده است که ویژگی‌های غیرخطی را به شبکه عصبی معرفی می‌کند و عملکرد شبکه عصبی عمیق را بهبود می‌بخشد. برخی از پارامترهای روش مدیریت منابع پیشنهادی و فرآیند یادگیری در جدول ۳ آمده است.

جدول ۳: پارامترهای شبیه‌سازی
Table 3. Simulation parameters

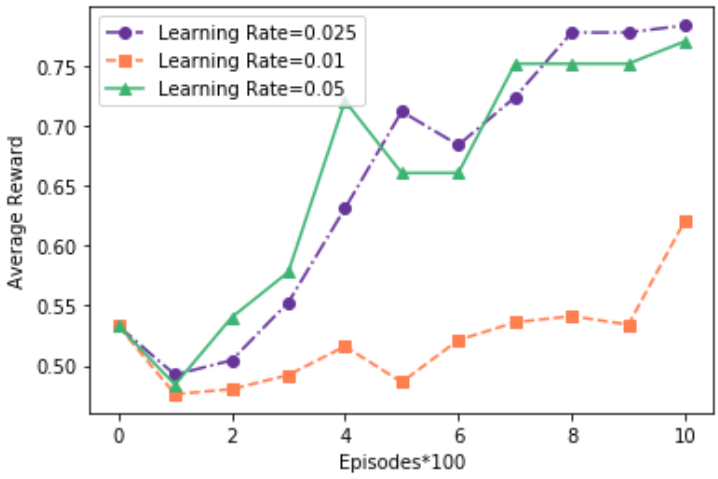
پارامترها	مقدارها
خوشه‌ها	۵
تعداد نورون ها	۱۲۸
تابع فعالسازی	Relu
اندازه بافر	٦۰۰۰
بهینه ساز	Adam
Mini-batch	۱۲۸
نرخ یادگیری	۰/۰۲۵
نرخ ورود دستورالعمل های وظایف	٥۰۰-٤۰۰۰(MIPS)
قدرت نویز	-۱۲۰dB
پهنای باند خودرو	۱-۲ (GHz)
پهنای باند RSU	۱۰۰-۲۰۰(MHz)
پهنای باند وظیفه	۱-۱۰۰(MHz)
نرخ ارسال داده ها از RSU به ابر	۱۰(GB/s)
ظرفیت محاسباتی خودرو/RSU/ابر	۲۰۰۰-۳۰۰۰/٥۰۰۰/۱۰۰۰۰(MIPS)
توان انتقال خودرو	۰/٥ watts
انرژی پردازش خودرو/RSU/ابر	۰/۰۰۰۱ /۰/۰۰۰٤ / ۰/۰۰۰۱۲-۰/۰۰۰۸ (joule/s)

[1] Rome
[2] Greedy
[3] Reinforcement Learning(RL)

هزینه پردازش خودرو/RSU/ابر	۰/۰۰۸۲ /۰/۰۰۶۱/ ۰/۰۰۳۳و۰/۰۰۲۲ (unit cost/s)
هزینه ارتباط بین خودرو و خودرو/RSU/ابر	۰/۰۰۱ (unit costs) /۰/۰۰۳۳/۰/۰۰۴
تعداد خودروها	۴۰۰-۵۰۰
تعداد گره‌های مه	۳۵

۴-۲-نتایج تجربی

با توجه به اینکه نرخ یادگیری یک پارامتر اساسی در یادگیری عمیق است که ممکن است بر عملکرد مدل تأثیر بگذارد، این مقاله نرخ یادگیری را روی ۰/۰۵، ۰/۰۱ و ۰/۰۲۵ تنظیم می‌کند. این مقاله بررسی می‌کند که چگونه آنها می‌توانند عملکرد الگوریتم پیشنهادی را افزایش دهند و شکل ۴ یافته‌ها را نشان می‌دهد. به خاطر نتایج زیاد، برای نشان دادن بهتر نتایج، هر ۱۰۰ قسمت[1]، میانگین گرفته می‌شود. همانطور که شکل نشان می‌دهد، الگوریتم پیشنهادی با سرعت همگرایی سریع‌تر و مقدار همگرایی بهتر در طول مرحله آموزش، زمانی که نرخ یادگیری روی ۰/۰۲۵ تنظیم شده است، عملکرد بهتری دارد. مقادیر همگرایی ضعیف مدل زمانی که نرخ یادگیری روی ۰/۰۵ و ۰/۰۱ تنظیم شده است، نشان داده شده است. نتایج بیان می‌کند که افزایش نرخ یادگیری می‌تواند منجر به همگرایی سریع‌تر شود. با این حال، افزایش بیشتر نرخ یادگیری هیچ بهبود قابل توجهی در مقادیر نشان نمی‌دهد که احتمال افتادن در راه‌حل بهینه محلی و شکست برای رسیدن به راه‌حل بهینه را افزایش می‌دهد. اما نرخ یادگیری پایین برای همگرایی زمان زیادی خواهد برد. نتیجه نشان می‌دهد که نرخ یادگیری بالاتر ممکن است منجر به یک مرحله به‌روزرسانی وزن کوچک شود که زمان آموزش را افزایش می‌دهد و برای دستیابی به حالت همگرایی، تکرارهای بیشتری را ضروری می‌کند. بنابراین، در تمام آزمایش‌ها، مقدار نرخ یادگیری ۰/۰۲۵ است.

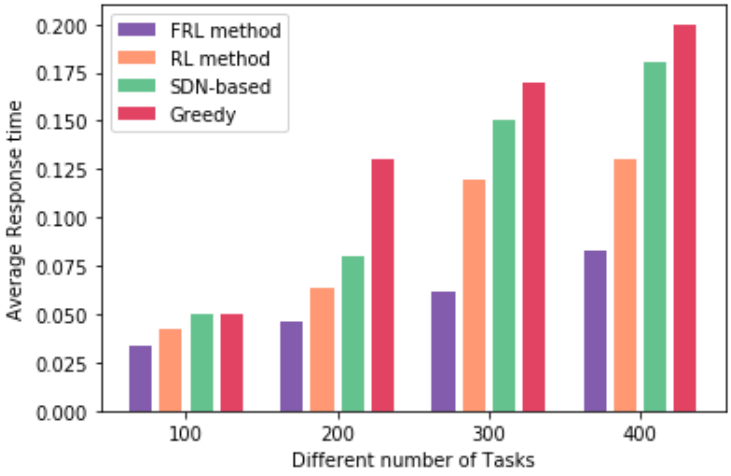


شکل ۴: مقایسه نرخ‌های یادگیری
Figure 4. Comparison of learning rates

نتایج آزمایش در شکل ۵ نشان داده شده است، که در آن محور طولی نشان‌دهنده تعداد وظایف و محور عرضی نشان‌دهنده مقدار میانگین زمان پاسخدهی (زمان برون‌سپاری وظیفه) است. زمان پاسخدهی شامل زمان انتقال وظیفه، زمان انتظار در صف، و زمان پردازش وظیفه است. همانطور که در شکل ۵ نشان داده شده است، زمان پاسخدهی در طرح پیشنهادی ما برای تعداد وظایف مختلف کمترین مقدار را دارد. تحلیل نتایج نشان می‌دهند که FRL[2] یک سکوی[3] مقیاس پذیر را برای یادگیری سیاست بهینه فراهم می‌کند. در بهینه‌سازی فدرال، چندین کنترلر توزیع شده برای بهینه‌سازی یک تابع هدف مشترک همکاری می‌کنند،

[1] Episode
[2] Federated Reinforcement Learning (FRL)
[3] Platform

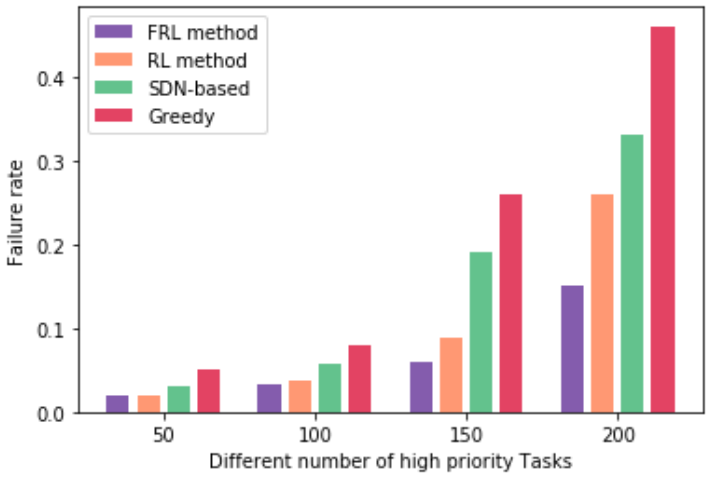
در حالی که عامل‌ها یاد می‌گیرند که در یک محیط غیرمتمرکز و بدون هماهنگی مرکزی عمل کنند. FRL با آموزش الگوریتم‌های مشارکتی بر روی داده‌های محلی، به اشتراک‌گذاری به‌روزرسانی‌های مدل سراسری به جای داده‌های خام، کار می‌کند و در نتیجه به ایجاد مدل با ایجاد وزن‌های بهینه کمک می‌کند. همچنین، از شکل ۵ می‌توانیم دریابیم که راه‌حل‌های RL و FRL در حال یادگیری نحوه تخصیص وظایف به سرورهای مناسب بر اساس منابع موجود هستند، در حالی که تقاضاهای تأخیر وظیفه را برآورده می‌کنند. روش RL در محیط‌های بزرگ خوب کار نمی‌کند و استراتژی پیشنهادی می‌تواند یک راه حل خوب باشد. یک جدول بزرگ در روش RL پیاده‌سازی شده است تا تمام شرایط حالت را حفظ کند و از کمبود مقیاس‌پذیری رنج می‌برد. رویکرد پیشنهادی از خودروهای پارک شده و در حال حرکت جهت انجام وظایف استفاده می‌کند بنابراین می‌تواند میانگین زمان پاسخدهی را کاهش دهد. روش SDN-based ارائه شده، در درون خوشه تنها از منابع RSU ها استفاده می‌کند. از آنجا که منابع RSUها محدود هستند، بنابراین وظیفه نمی‌تواند به صورت محلی اجرا شود و در نتیجه باید براساس تصمیم کنترلر سراسری و از طریق سوییچ ها به سرور مقصد برون‌سپاری گردد. در راه‌حل حریصانه، هر وظیفه به سروری با ظرفیت منابع بالا اختصاص داده می‌شود و از هیچ سیاست یادگیری پیروی نمی‌شودکه در نتیجه زمان پاسخدهی در روش حریصانه برای استراتژی بهینه در نظر گرفته نمی‌شود.



شکل ۵: میانگین زمان پاسخدهی برای چهار الگوریتم با تعداد وظایف مختلف

Figure 5. Average response time for four algorithms with different numbers of tasks

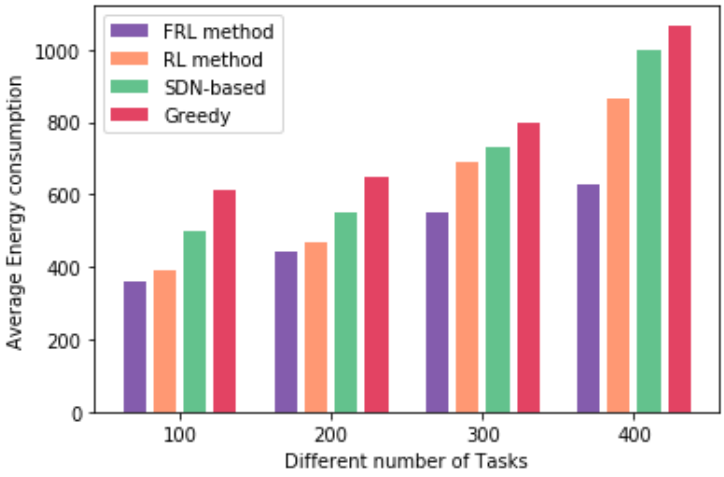
همانطور که قبلاً بیان شد، وظایف با اولویت بالا دارای مهلت زمانی از پیش تعریف شده اند و باید مهلت زمانی رعایت شود. روش حریصانه، رویکرد مناسبی برای انتخاب سیاست بهینه برای برنامه‌های بلادرنگ نیست. از آنجا که الگوریتم حریصانه معمولاً مانند برنامه‌نویسی پویا به طور کامل روی همه داده‌ها عمل نمی‌کند، معمولاً (اما نه همیشه) در کشف یک راه‌حل بهینه شکست می‌خورد. از طرف دیگر، تکنیک‌های یادگیری تقویتی می‌توانند با استفاده از قابلیت‌های شبکه‌های عصبی، سرورهای مقصد را برای برون‌سپاری وظایف به طور مؤثرتری انتخاب کنند. روش SDN-based، اولویت وظایف را در نظر نمی‌گیرد و در نتیجه نمی‌تواند اجرای وظایف بلادرنگ را تضمین نماید.



شکل ۶: نرخ شکست وظیفه برای چهار الگوریتم با تعداد وظایف اولویت بالای مختلف

Figure 6. Task failure rates for four algorithms with different numbers of high priority tasks

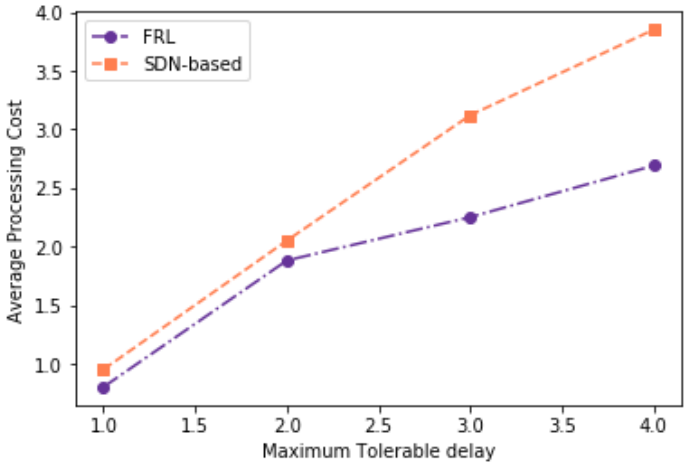
شکل ۷، میانگین مصرف انرژی برای چهار الگوریتم مختلف را نشان می‌دهد که الگوریتم‌های FRL و RL با یادگیری استراتژی بهینه می‌توانند تصمیمات بهتری را با توجه به در نظر گرفتن مصرف انرژی به عنوان محدودیت اتخاذ کنند. همانطور که شکل نشان می‌دهد الگوریتم RL نمی‌تواند در شبکه‌های با مقیاس بالا خوب کار کند. الگوریتم SDN-based و حریصانه مصرف انرژی را در نمی‌گیرند و تنها بر روی انجام وظیفه در مهلت زمانی خود تاکید دارند که با کوتاه‌ترین زمان وظیفه انجام شود و بنابراین کارایی مناسبی از لحاظ انرژی ندارند.



شکل ۷: میانگین مصرف انرژی برای چهار الگوریتم با تعداد وظایف اولویت بالای مختلف

Figure 7. Average energy consumption for four algorithms with different numbers of high priority tasks

همانطور که بیان شد، روش پیشنهادی از یادگیری فدرال همراه با فناوری SDN استفاده می‌کند که می‌تواند انعطاف‌پذیری و مقیاس‌پذیری را در شبکه افزایش دهد. در شکل ۸، دو الگوریتم مبتنی بر FRL و SDN-based با هم مقایسه شده اند. در الگوریتم FRL، تابع هدف افزایش نرخ پذیرش وظایف و همچنین در الگوریتم SDN-based، تابع هدف براساس کاهش زمان برون‌سپاری وظیفه است. بنابراین با افزایش مهلت زمانی، وظایف می‌توانند روی RSU و ابر اجرا شوند که می‌تواند، هزینه اجرا را افزایش دهد. می‌توان گفت که هزینه پردازش در مدل هر دو الگوریتم به عنوان محدودیت است که لزوما هزینه کمینه نمی‌شود. نتایج نشان می‌دهد که الگوریتم پیشنهادی با یادگیری استراتژی بهینه بر اساس الگوی ترافیکی خودروها می‌تواند هزینه را نیز کاهش دهد.



شکل ۸: میانگین هزینه در مقابل بیشینه مهلت زمانی برای وظایف مختلف
Figure 8. Average cost vs. maximum deadline for different tasks

۵-ارزیابی و نتیجه‌گیری

در این مقاله، ما یک تکنیک برون‌سپاری وظیفه مبتنی بر Q-learning عمیق فدرال را برای توسعه یک مدل برون‌سپاری سراسری برای محاسبات مه خودرویی پیشنهاد کرده‌ایم. روش پیشنهادی هم مصرف انرژی و هم عدم تعادل بار در سرورهای مه را به حداقل می‌رساند. نتایج تجربی نشان می‌دهند که تکنیک برون‌سپاری پیشنهادی ما از تکنیک‌های برون‌سپاری پایه از لحاظ میانگین زمان پاسخدهی و هم میانگین هزینه پردازش کارآمدتر است. در آینده، ما قصد داریم این کار را گسترش دهیم تا تکنیک‌های یادگیری تقویتی عمیق پیچیده‌تری را برای حل مسئله برون‌سپاری وظیفه همراه با دیگر مجموعه داده‌های محیط واقعی نظر بگیریم.

مراجع

[1] M. K. Farimani, S. Karimian-Aliabadi, R. Entezari-Maleki, B. Egger, and L. Sousa, "Deadline-aware task offloading in vehicular networks using deep reinforcement learning," *Expert Systems with Applications,* vol. 249, p. 123622, 2024.

[2] O. Akyıldız, F. Y. Okay, İ. Kök, and S. Özdemir, "Road to efficiency: Mobility-driven joint task offloading and resource utilization protocol for connected vehicle networks," *Future Generation Computer Systems,* vol. 156, pp. 157-167, 2024.

[3] X. Zhang, Y. Zhu, C. Wang, J. Cao, Y. Chen, and J. Wang, "Multi-agent reinforcement learning for vehicular task offloading with multi-step trajectory prediction," *CCF Transactions on Pervasive Computing and Interaction,* vol. 6, no. 2, pp. 101-114, 2024.

[4] J. Shi, J. Du, J. Wang, J. Wang, and J. Yuan, "Priority-aware task offloading in vehicular fog computing based on deep reinforcement learning," *IEEE Transactions on Vehicular Technology,* vol. 69, no. 12, pp. 16067-16081, 2020.

[5] S. Zhou, Y. Sun, Z. Jiang, and Z. Niu, "Exploiting moving intelligence: Delay-optimized computation offloading in vehicular fog networks," *IEEE Communications Magazine,* vol. 57, no. 5, pp. 49-55, 2019.

[6] Z. Zhou, H. Liao, X. Wang, S. Mumtaz, and J. Rodriguez, "When vehicular fog computing meets autonomous driving: Computational resource management and task offloading," *IEEE Network,* vol. 34, no. 6, pp. 70-76, 2020.

[7] S. Misra and S. Bera, "Soft-VAN: Mobility-aware task offloading in software-defined vehicular network," *IEEE Transactions on Vehicular Technology,* vol. 69, no. 2, pp. 2071-2078, 2019.

[8] A. A. Khadir and S. A. H. Seno, "SDN-based offloading policy to reduce the delay in fog-vehicular networks," *Peer-to-Peer Networking and Applications,* vol. 14, no. 3, pp. 1261-1275, 2021.

[9] H. M. Birhanie, M. A. Messous, S.-M. Senouci, E.-H. Aglzim, and A. M. Ahmed, "MDP-based resource allocation scheme towards a vehicular fog computing with energy constraints," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1-6: IEEE.

[10] V. Sethi and S. Pal, "FedDOVe: A Federated Deep Q-learning-based Offloading for Vehicular fog computing," *Future Generation Computer Systems,* vol. 141, pp. 96-105, 2023.

[11] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE circuits and systems magazine,* vol. 9, no. 3, pp. 32-50, 2009.

[12] Q. Qi *et al.*, "Knowledge-driven service offloading decision for vehicular edge computing: A deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology,* vol. 68, no. 5, pp. 4192-4203, 2019.

[13] H. Guo, J. Liu, J. Ren, and Y. Zhang, "Intelligent task offloading in vehicular edge computing networks," *IEEE Wireless Communications,* vol. 27, no. 4, pp. 126-132, 2020.

[14] S. Memon and M. Maheswaran, "Using machine learning for handover optimization in vehicular fog computing," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2019, pp. 182-190.

[15] J. Shi, J. Du, J. Wang, and J. Yuan, "Deep Reinforcement Learning-Based V2V Partial Computation Offloading in Vehicular Fog Computing," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1-6: IEEE.

[16] N. E. H. Boubaker, K. Zarour, N. Guermouche, and D. Benmerzoug, "Optimizing Workflow Offloading and Migration under Timed Constraints in Fog and Cloud Computing," *Journal of Grid Computing,* vol. 23, no. 1, p. 10, 2025.

[17] J. Zhao, M. Kong, Q. Li, and X. Sun, "Contract-based computing resource management via deep reinforcement learning in vehicular fog computing," *IEEE Access,* vol. 8, pp. 3319-3329, 2019.

[18] B. Shabir, A. U. Rahman, A. W. Malik, R. Buyya, and M. A. Khan, "A federated multi-agent deep reinforcement learning for vehicular fog computing," *The Journal of Supercomputing,* pp. 1-27, 2022.

[19] J. Xue, L. Wang, Q. Yu, and P. Mao, "Multi-agent deep reinforcement learning-based partial offloading and resource allocation in vehicular edge computing networks," *Computer Communications,* vol. 234, p. 108081, 2025.

[20] P. Dai, Y. Huang, K. Hu, X. Wu, H. Xing, and Z. Yu, "Meta Reinforcement Learning for Multi-task Offloading in Vehicular Edge Computing," *IEEE Transactions on Mobile Computing,* 2023.

[21] K. Zhang, J. Cao, and Y. Zhang, "Adaptive digital twin and multiagent deep reinforcement learning for vehicular edge computing and networks," *IEEE Transactions on Industrial Informatics,* vol. 18, no. 2, pp. 1405-1413, 2021.

[22] S.-C. Lin, K.-C. Chen, and A. Karimoddini, "SDVEC: software-defined vehicular edge computing with ultra-low latency," *IEEE Communications Magazine,* vol. 59, no. 12, pp. 66-72, 2021.

[23] Y. Hou, Z. Wei, R. Zhang, X. Cheng, and L. Yang, "Hierarchical task offloading for vehicular fog computing based on multi-agent deep reinforcement learning," *IEEE Transactions on Wireless Communications,* 2023.

[24] X. Ma, L. Liao, Z. Li, R. X. Lai, and M. Zhang, "Applying federated learning in software-defined networks: A survey," *Symmetry,* vol. 14, no. 2, p. 195, 2022.

[25] M. U. Thomas, "Queueing systems. volume 1: Theory (leonard kleinrock)," *SIAM Review,* vol. 18, no. 3, pp. 512-514, 1976.

[26] B. Yang *et al.*, "Edge intelligence for autonomous driving in 6G wireless system: Design challenges and solutions," *IEEE Wireless Communications,* vol. 28, no. 2, pp. 40-47, 2021.

[27] M. Li, J. Gao, L. Zhao, and X. Shen, "Deep reinforcement learning for collaborative edge computing in vehicular networks," *IEEE Transactions on Cognitive Communications and Networking,* vol. 6, no. 4, pp. 1122-1135, 2020.

[28] Z. Ning and L. Xie, "A survey on multi-agent reinforcement learning and its application," *Journal of Automation and Intelligence,* vol. 3, no. 2, pp. 73-91, 2024.

[29] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, 2017, pp. 1273-1282: PMLR.

[30] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi, "CRAWDAD dataset roma/taxi," ed: Version 2014-07-17, 2014.

[31] S. Imambi, K. B. Prakash, and G. Kanagachidambaresan, "PyTorch," in *Programming with TensorFlow: solution for edge computing applications*: Springer, 2021, pp. 87-104.

[32] I. Daubechies, R. DeVore, S. Foucart, B. Hanin, and G. Petrova, "Nonlinear approximation and (deep) ReLU networks," *Constructive Approximation,* vol. 55, no. 1, pp. 127-172, 2022.