

شبکه های عصبی گراف محور آفلودینگ (واگذاری) تسک و تخصیص منابع در شبکه های خودرویی مقیاس پذیر

Menghan Shao¹, Rongqing Zhang², Liuqing Yang³

¹ Internet of Things Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China

² School of Computer Science and Technology, Tongji University, Shanghai, China

³ Internet of Things Thrust and Intelligent Transportation Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China

دریافت: ۲۳ آوریل ۲۰۲۵ — بازبینی: ۱۲ ژوئن ۲۰۲۵ — پذیرش: ۹ ژوئیه ۲۰۲۵

حمایت مالی: چند پروژه ً استانی و بنیاد ملی چین

چکیده

وسایل نقلیه هوشمند^۱ (ITS) برای افزایش ایمنی و آسایش راننده نیازمند پردازش حجیم داده‌ها هستند. با وجود محدودیت منابع محاسباتی در واحدهای درون خودرویی، معمولاً بخشی از محاسبات به خودروهای مجاور یا سرورها واگذار می‌شود تا نیازهای پاسخ در لحظه برآورده شود. با این حال، پیچیدگی و دینامیک بالای محیط خودرویی، طراحی راهبردهای واگذاری مؤثر را دشوار می‌سازد. روش‌های فعلی گرچه می‌توانند به تغییر پارامترهای محیطی سازگار شوند، اما اساساً در پردازش اطلاعات با ابعاد متغیر و تصمیم‌گیری مقیاس‌پذیر با اندازه شبکه محدود هستند. روش‌های سنتی معمولاً به نمایش‌های ورودی ثابت و چارچوب‌های محاسباتی ایستا متکی‌اند که برای طبیعت پویا و مقیاس‌پذیر شبکه‌های خودرویی مناسب نیستند. در این مقاله سازوکاری نوین مبتنی بر شبکه عصبی گرافی^۲ (GNN) برای مسئله آفلودینگ تسک و تخصیص منابع در اندازه متغیر توسعه می‌دهیم. با بهره‌گیری از مکانیزم تجمیع همسایگان در GNN، این رویکرد به طور مؤثری به توپولوژی‌های با مقیاس متغیر در شبکه‌های خودرویی پویا سازگار می‌شود و عملکردی مقاوم نسبت به اندازه شبکه تضمین می‌کند. برای ارزیابی، شبیه‌سازی‌های گسترده‌ای انجام شده که نشان می‌دهد روش پیشنهادی مقیاس‌پذیری و قابلیت‌پذیری زمان-واقعی بهتری ارائه می‌دهد و از روش‌های موجود در مواجهه با تغییر اندازه شبکه فراتر می‌رود.

^۱ intelligent transport system

^۲ Graph neural network

مقدمه

در دهه‌های اخیر، خودروها به واسطه یکپارچه‌سازی حسگرها و الگوریتم‌های پیشرفته از یک وسیله که صرفاً برای حمل و نقل استفاده می‌شدند؛ به سامانه‌های حرکت هوشمند تبدیل شده‌اند. این نوآوری‌ها قابلیت‌های خود-ادراک، تصمیم‌گیری خودکار و یادگیری خودکار را فراهم ساخته و هوشمندی خودرو را به‌طور چشمگیری افزایش می‌دهند. خدمات خودرویی به دلیل پردازش حجم عظیمی از داده، حسگرها و اجرای الگوریتم‌های پیچیده، تصمیم‌گیری و کنترل، به محاسبات سنگین و تأخیر بسیار کم نیاز دارند؛ همچنین پاسخ فوری به سناریوهای ترافیکی مختلف برای جلوگیری از تصادفات ضروری است. [۲۱]

واحد محاسباتی درون خودرویی^۳ (OBU) به دلیلی محدودیت‌های اندازه و هزینه، منابع محاسباتی محدودی دارد و معمولاً برای برآورده کردن تقاضای محاسباتی سنگین کافی نیست. برای رفع این محدودیت، خودروها می‌توانند تسک‌ها را به پلتفرم‌های محاسباتی دیگر مانند ابر مرکزی، سرورهای لبه (edge) یا خودروهای مجاور واگذار کنند. با این وجود، اجرای آفلودینگ با چالش‌هایی همراه است؛ به‌ویژه به دلیل طبیعت بسیار پویا در شبکه‌های خودرویی که با تغییر توپولوژی سریع و حرکت غیرقابل پیش‌بینی خودروها مشخص می‌شود. [۳] استفاده از محاسبات لبه (به‌ویژه در کنار ایستگاه‌های کنار جاده‌ای^۴ (RSU) راهکار امیدبخشی برای فراهم کردن پشتیبانی محاسباتی است [۴، ۵]؛ اما پوشش جغرافیایی RSU ها محدود و استقرار متراکشان به دلیل هزینه زیرساختی و نگهداری مقرون به صرفه نیست. برای غلبه بر این محدودیت‌ها، پژوهشگران پیشنهاد داده‌اند محاسبات را به خودروهای پیرامون با منابع بیکار واگذار کنند تا استفاده منابع و تأخیر محاسباتی کاهش یابد [۶، ۷]. هرچند خودروها متحرکند، خودروهایی که در یک جهت حرکت می‌کنند معمولاً تأخیرهای ارتباطی کمی دارند که آن‌ها را برای آفلودینگ مناسب می‌سازد.

طراحی تصمیم‌گیری در شبکه‌های خودرویی پویا دشوار است؛ باید دستگاه مناسب برای اجرای تسک انتخاب شود و منابع محاسباتی به‌صورت پویا تخصیص یابد. این تصمیم‌ها تحت تأثیر فاکتورهای متغیر متعدد از جمله تغییر توپولوژی، کیفیت ارتباط وابسته به زمان و ناپایداری منابع قرار دارند. در میان این‌ها، تغییر عددی گره‌ها (خودروها) و نوسان در تعداد گره‌های فعال اهمیت ویژه‌ای دارد و به راهبردی نیاز است که هم سریع و هم مقیاس‌پذیر باشد. روش‌های موجود محدودیت‌هایی دارند: الگوریتم‌های دقیق از نظر محاسباتی سنگین و برای زمان-واقعی نامناسب‌اند؛ روش‌های هوریستیک قابل اجرا ولی فاقد سازگاری با محیط بسیار پویا هستند [۸].

^۳ on board unit

^۴ road side unit

روش‌های مبتنی بر یادگیری^۵ (ML) و بخصوص شبکه‌های عصبی عمیق^۶ (DNN) محبوب شده‌اند [۹-۱۱]، اما DNNها به دلیل ساختار ثابتشان توان پردازش توپولوژی متغیر را ندارند. رویکردهایی مانند پدینگ با صفر یا mask کردن خروجی‌ها ناکارآمدی و افت عملکرد ایجاد می‌کنند. برای رفع این محدودیت‌ها، ما طرحی مبتنی بر GNN پیشنهاد می‌کنیم که به خوبی برای مدل‌سازی توپولوژی پویا مناسب است. ما یک سناریوی شبکه خودرویی مقیاس‌پذیر ارائه می‌دهیم که در آن چندین خودروی وظیفه‌دار (task vehicles) و خودروی سرویس‌دهنده (service vehicles) همکاری می‌کنند تا راهبردی بهینه برای آفلودینگ و تخصیص منابع بیابند با هدف کاهش تأخیر متوسط و افزایش پاسخگویی برنامه‌های خودرویی. نتایج شبیه‌سازی نشان می‌دهد روش پیشنهادی به تغییرات مقیاس شبکه سازگار است و تعادل مناسبی بین زمان استنتاج تصمیم و کارایی پردازش تسک برقرار می‌کند.

مشارکت‌های اصلی مقاله:

- مدل‌سازی شبکه خودرویی مقیاس‌پذیر: مدل مسئله آفلودینگ و تخصیص منابع در شبکه‌ای با توپولوژی پویا و اندازه متغیر؛
- نمایش مبتنی بر گراف: گره‌ها نمایانگر خودروها و یال‌ها نمایانگر لینک‌های ارتباطی متغیر زمانی؛
- چارچوب GNN برای آفلودینگ و تخصیص منابع: ^۷(G-TORA) بهره‌گیری از توانمندی GNN در ورودی‌های با اندازه متغیر و تعمیم‌پذیری به مقیاس‌های متفاوت شبکه.

پژوهش‌های مرتبط

آفلودینگ تسک و تخصیص منابع در شبکه‌های خودرویی به فرایند انتقال تسک‌های محاسباتی از خودروهای دارای منابع محدود به خودروهای سرویس‌دهنده یا سرورهای لبه برای بهبود توان محاسباتی و کاهش تأخیر اطلاق می‌شود. دو دسته^۸ اصلی روش‌ها در ادبیات عبارت‌اند از: روش‌های مبتنی بر متاهیوریستیک و روش‌های مبتنی بر یادگیری. در ادامه مروری بر این دو دسته و چالش‌های کلیدی ارائه می‌شود.

^۵machine learning

^۶deep neural network

^۷GNN-based task offloading and resource allocation

۲.۱ روش‌های متاهیوریستیک

مطالعات متعددی از الگوریتم‌های تکاملی و الهام گرفته از طبیعت برای زمان‌بندی تسک و تخصیص منابع استفاده کرده‌اند مثلاً ترکیب الگوریتم‌های بهینه‌سازی مانند PSO، الگوریتم‌های مبتنی بر کلونی، و غیره [۱۲-۱۵] در مجموع، این روش‌ها راه‌حل‌های تقریباً بهینه^۸ کارآمدی ارائه می‌کنند اما در مواجهه با تغییرات سریع محیط خودرویی نیاز به اجرای مکرر و هزینه محاسباتی دارند که برای تصمیم‌گیری در زمان-واقعی مشکل‌ساز است.

۲.۲ روش‌های مبتنی بر یادگیری

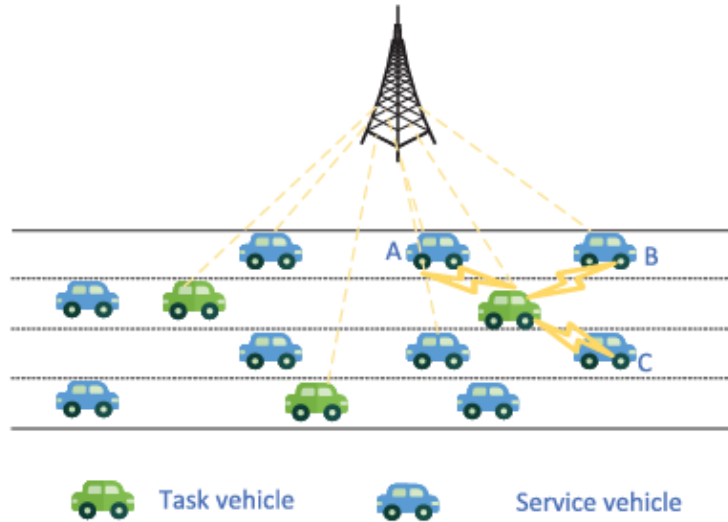
روش‌های یادگیری از جمله^۸ RL،^۹ DQN،^{۱۰} DDPG،^{۱۱} MARL و غیره (انعطاف‌پذیری بیشتری در محیط‌های پویا دارند و در ادبیات وسیعاً مورد استفاده‌اند [۱۶-۲۷] Q-learning و نسخه‌های عمیق آن، روش‌های DDPG برای مسائل با فضای عملیاتی پیوسته، و MARL برای چارچوب‌های غیرمتمرکز نمونه‌هایی از این رویکردها هستند. با این حال، بیشتر این روش‌ها فرض ثبات اندازه شبکه یا استفاده از DNN‌های با ابعاد ورودی ثابت را دارند که مقیاس‌پذیری را محدود می‌کند. به همین دلیل، ما GNN را برای بهره‌گیری از قابلیت‌های استخراج ویژگی‌های توپولوژیک و انعطاف‌پذیری ابعاد ورودی به کار می‌بریم.

^۸ reinforcement learning

^۹ deep Q network

^{۱۰} deep deterministic policy gradient

^{۱۱} multi-agent reinforcement learning



شکل ۱) معماری شبکه متصل خودرویی

مدل سیستم و فرمول بندی مسئله ۳

مدل سیستم ۳.۱

معماری شبکه مورد نظر در شکل ۱ نشان داده شده است. یک ایستگاه پایه 12 (BS) که یک مسیر یک طرفه را پوشش می‌دهد وجود دارد و خودروها می‌توانند با هم ارتباط برقرار کنند. زمان بندی تسک به بازه‌های زمانی (time slots) تقسیم شده است. در هر بازه، وضعیت شبکه تقریباً ثابت در نظر گرفته می‌شود اما بین بازه‌ها تغییر می‌کند [۲۸]. ایستگاه پایه (BS) مسئول گرفتن تصمیمات آفلودینگ است و به دلیل برد ارتباطی بیشتر قادر است وضعیت خودروها در ناحیه پوشش خود را گردآوری کند. وقتی خودرویی وارد محدوده BS می‌شود، پیام حاوی وضعیت تسک، موقعیت و توان محاسباتی محلی خود را ارسال می‌کند. بنابراین عامل (agent) مستقر در BS می‌تواند اطلاعات خودروهای داخل پوشش را جمع‌آوری و تصمیم‌گیری کند. در هر بازه زمانی، مجموعه‌ای از خودروها قادر به اجرای برخی برنامه‌ها نیستند — این‌ها را task vehicles می‌نامیم (M_t عدد از آن‌ها) و مجموعه‌ای از خودروها منابع بیکار دارند — این‌ها را service vehicles می‌نامیم (N_t عدد از آن‌ها). هر خودروی تسک‌دار در همان بازه یک تسک دارد و پروفایل تسک با سه تایی $\langle d_i, c_i, f_i \rangle$ تعریف می‌شود: d_i اندازه داده ورودی، c_i نیاز محاسباتی (به تعداد سیکل‌های CPU)، و f_i ظرفیت محاسباتی محلی (به سیکل CPU). ظرفیت

¹² base station

محاسباتی سرویس و هیکل ها با f نشان داده می شود و پهنای باند B است. شبکه خودرویی در هر بازه به صورت گراف $G_t = (V_t, E_t)$ مدل می شود که گره ها خودروها و یال ها اطلاعات کانال بین آنها هستند.

$$R_{i,j} = B \log \left(1 + \frac{p_{iH_{i,j}^2}}{\sigma^2} \right) \quad 1-1$$

که در آن B پهنای باند بین وسایل نقلیه است، $P_{i,j}$ توان انتقال وسیله نقلیه مأموریت دهنده i است، $H_{i,j}$ ضریب کانال بین وسیله نقلیه مأموریت دهنده i و وسیله نقلیه خدمات دهنده j است، و σ^2 نمایانگر نویز زمینه ای با توزیع گاوسی می باشد.

۳.۳ مدل محاسباتی

دو روش برای انجام وظایف محاسباتی وجود دارد:

۱. **محاسبات محلی:** (Local computing) در این روش، خود وسیله نقلیه ای که وظیفه را ایجاد کرده است، وظایف محاسباتی را انجام می دهد.

۲. **واگذاری محاسبات به وسیله نقلیه خدمات دهنده:** (Offloading computing) در این روش، وظایف به وسیله نقلیه های خدمات دهنده نزدیک واگذار می شوند تا با استفاده از توان محاسباتی بالاتر آنها، انجام شوند.

۱. محاسبات محلی

برای محاسبات محلی، وظیفه تولید شده توسط وسیله نقلیه روی پردازنده محلی اجرا می شود و تاخیر فقط به فرکانس پردازنده محلی f_i بستگی دارد، که به صورت زیر بیان می شود:

$$T_{loc}^i = \frac{c_i}{f_i} \quad 1-2$$

۲. واگذاری محاسبات

اگر وظیفه به وسیله نقلیه خدمات‌دهنده واگذار شود، سه مرحله برای تکمیل وظیفه وجود دارد:

۱. وسیله نقلیه مأموریت‌دهنده ابتدا وظیفه را از طریق کانال بی‌سیم به وسیله نقلیه خدمات‌دهنده ارسال می‌کند.

۲. وسیله نقلیه خدمات‌دهنده محاسبات را انجام می‌دهد.

۳. پس از انجام محاسبات، نتیجه به وسیله نقلیه مأموریت‌دهنده بازگردانده می‌شود.

تاخیر انتقال از وسیله نقلیه مأموریت‌دهنده به وسیله نقلیه خدمات‌دهنده به اندازه داده‌های وظیفه و نرخ انتقال بستگی دارد:

پس از دریافت وظیفه، وسیله نقلیه خدمات‌دهنده منابع محاسباتی باقی‌مانده خود را برای انجام محاسبات اختصاص می‌دهد. تاخیر محاسباتی وظیفه روی وسیله نقلیه خدمات‌دهنده j به صورت زیر محاسبه می‌شود:

$$T_{edge}^{i,j} = \frac{c_i}{f_{i,j}} \quad ۱-۳$$

که در آن $f_{i,j}$ فرکانس محاسباتی است که وسیله خدمات‌دهنده j به وسیله مأموریت‌دهنده i اختصاص داده است. پس از اجرای محاسبات، وسیله خدمات‌دهنده نتیجه پردازش را به وسیله مأموریت‌دهنده مربوطه بازمی‌گرداند. بر اساس پژوهش‌های [۲۹، ۳۰]، به دلیل اینکه حجم داده‌های نتایج محاسباتی بسیار کمتر از داده‌های ورودی است، می‌توان تاخیر بازگشت نتایج را نادیده گرفت. به طور خلاصه، برای شبکه‌های خودرویی، زمان پاسخ مجموعه وظایف تولیدشده در بازه زمانی t می‌تواند به صورت زیر بیان شود:

برای شبکه خودرویی، زمان پاسخ مجموعه وظایف تولیدشده در بازه زمانی t می‌تواند به صورت زیر بیان شود:

$$T_t = \sum_{i \in \mathcal{M}_t} (x_{i,0} T_{loc}^i + \sum_{j \in \mathcal{N}_t} (x_{i,j} (T_{trans}^{i,j} + T_{edge}^{i,j}))) \quad ۱-۴$$

۳.۴ فرموله‌سازی مسئله

در سیستم شبکه خودرویی پیشنهادی، فرض می‌کنیم که ایستگاه پایه (BS) تصمیمات واگذاری (offloading) را برای چندین وسیله نقلیه مأموریت‌دهنده در محدوده پوشش خود در بازه‌های زمانی مختلف می‌گیرد. برخلاف کارهای موجود که فرض می‌کنند اندازه توپولوژی شبکه همیشه ثابت است، ما معتقدیم که تعداد وسایل نقلیه مأموریت‌دهنده و خدمات‌دهنده در شبکه در بازه‌های مختلف تغییر خواهد کرد، مانند اضافه یا حذف گره‌ها.

در هر بازه زمانی، وسایل نقلیه مأموریت‌دهنده ممکن است وظایف محاسباتی خود را به وسایل نقلیه خدمات‌دهنده واگذار کنند و وسایل خدمات‌دهنده نیز منابع بلااستفاده خود را به این وظایف اختصاص می‌دهند. هدف ما کاهش میانگین تاخیر پاسخ در این دوره با طراحی یک استراتژی مؤثر برای واگذاری وظایف و تخصیص منابع محاسباتی است.

زمان پاسخ کل وظایف در این دوره به صورت زیر است:

$$T(\mathcal{G}_t, \mathcal{X}) = \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{M}_t} (x_{i,0}^t T_{loc}^i) + \sum_{j \in \mathcal{N}_t} (x_{i,j}^t (T_{trans}^{i,j} + T_{edge}^{i,j})) \quad ۱-۵$$

مسئله بهینه‌سازی به شکل زیر فرموله می‌شود:

$$\min T(\mathcal{G}_t, \mathcal{X}) \quad ۱-۶$$

با محدودیت‌ها:

$$C_1: x_{i,0}^t + \sum_{j \in \mathcal{N}_t} x_{i,j}^t = 1, \forall i \in \mathcal{M}_t \quad ۱-۷-۱$$

توضیح: این محدودیت اطمینان می‌دهد که هر وظیفه باید یا به صورت محلی اجرا شود یا به وسیله نقلیه خدمات‌دهنده واگذار گردد.

$$C_2: f_{i,j}^t \leq f_j, \forall j \in \mathcal{N}_t, \forall i \in \mathcal{M}_t \quad ۱-۷-۲$$

توضیح: مجموع منابع محاسباتی اختصاص داده شده توسط هر وسیله خدمات‌دهنده به وسایل مأموریت‌دهنده نباید از حد ماکسیمم منابع محاسباتی آن تجاوز کند.

$$C_3: x_{i,j}^t \in \{0,1\}, \forall i \in \mathcal{M}_t, \forall j \in \mathcal{N}_t \quad \bullet$$

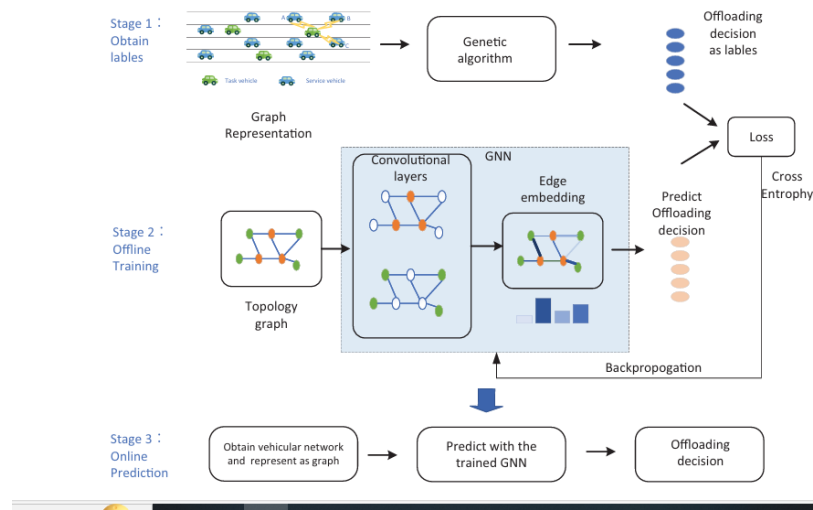
توضیح: اگر $x_{i,j}^t = 1$ باشد، یعنی وسیله نقلیه مأموریت‌دهنده i وظیفه خود را به وسیله j واگذار می‌کند، و اگر $x_{i,j}^t = 0$ باشد، واگذاری صورت نمی‌گیرد.

محدودیت‌ها:

- **C1** تضمین می‌کند که هر وظیفه باید یا به صورت محلی اجرا شود یا روی وسیله نقلیه خدمات‌دهنده.
- **C2** به محدودیت اشاره دارد که مجموع منابع محاسباتی اختصاص داده شده توسط هر وسیله خدمات‌دهنده به وسایل مأموریت‌دهنده نباید از حداکثر منابع محاسباتی آن وسیله فراتر رود.
- **C3** بیان می‌کند که اگر $x_{i,j}^t = 1$ باشد، وسیله نقلیه مأموریت‌دهنده i وظیفه خود را به وسیله j واگذار می‌کند و اگر $x_{i,j}^t = 0$ باشد، واگذاری صورت نمی‌گیرد.

مسئله بهینه‌سازی به دلیل ماهیت ترکیبی تصمیمات واگذاری وظایف و متغیرهای پیوسته تخصیص منابع، NP-hard است. زمانی که اقدام واگذاری محاسبات در بازه زمانی t مشخص شود (یعنی $x_{i,j}^t$ به عنوان یک مقدار قطعی تعیین شود)، مسئله اصلی به یک مسئله بهینه‌سازی محدب ساده می‌شود. بنابراین، در بخش بعدی تمرکز ما بر متغیرهای واگذاری وظایف خواهد بود.

الگوریتم‌های دقیق به دلیل فضای جستجوی بزرگ کند هستند و شبکه‌های عصبی عمیق سنتی (DNN) قابلیت سازگاری با ماهیت پویا و مقیاس‌پذیر شبکه‌های خودرویی را ندارند. برای غلبه بر این محدودیت‌ها، ما یک روش مبتنی بر شبکه عصبی گرافی (GNN) پیشنهاد می‌کنیم که ساختار گراف شبکه‌های خودرویی را مورد استفاده قرار می‌دهد. شبکه‌های GNN به طور ذاتی روابط توپولوژیکی بین وسایل نقلیه را به دست می‌آورند و برای اندازه‌ها و ساختارهای پویا و متغیر شبکه مناسب هستند.



شکل ۲) چارچوب واگذاری وظایف مبتنی بر GNN در شبکه‌های خودرویی

۴. چارچوب واگذاری وظایف و تخصیص منابع مبتنی بر GNN

در این بخش، نمای کلی و جامع چارچوب مبتنی بر GNN برای واگذاری وظایف و تخصیص منابع ارائه می‌شود. ابتدا روند کاری کلی چارچوب بیان می‌شود، سپس توضیح داده می‌شود که چگونه الگوریتم ژنتیک (GA) برای تولید برجسب‌های واگذاری وظایف برای GNN استفاده می‌شود و در نهایت، فرآیند آموزش مدل GNN با استفاده از گراف توپولوژی شبکه خودرویی و برجسب‌های تولیدشده تشریح می‌گردد تا امکان استنتاج فراهم شود.

۴.۱ روند کاری چارچوب

ماهیت پویا شبکه‌های خودرویی به ویژه در تغییر تعداد وسایل نقلیه داخل محدوده پوشش ایستگاه پایه (BS) مشاهده می‌شود، زیرا وسایل نقلیه وارد یا خارج می‌شوند. این نوسان مداوم مقیاس مسئله بهینه‌سازی را تغییر می‌دهد، زیرا تعداد وسایل مأموریت‌دهنده و خدمات‌دهنده در طول زمان متغیر است و پیچیدگی فرآیند واگذاری

وظایف و تخصیص منابع را افزایش می‌دهد. بنابراین، طراحی یک طرح واگذاری وظایف که قادر به تصمیم‌گیری در زمان واقعی و سازگار با اندازه‌های مختلف توپولوژی شبکه باشد بسیار مهم است.

ما یک چارچوب مبتنی بر GNN پیشنهاد می‌کنیم که GA و GNN را برای حل مسئله واگذاری وظایف و تخصیص منابع مقیاس‌پذیر یکپارچه می‌کند. این چارچوب با بهره‌گیری از GA برای برچسب‌گذاری داده‌ها و GNN برای تصمیم‌گیری، واگذاری وظایف در شبکه‌های خودرویی را بهینه می‌کند.

روند کاری چارچوب پیشنهادی در شکل ۲ نشان داده شده است. چارچوب شامل سه مرحله اصلی است:

۱. مرحله آماده‌سازی داده‌ها:

- در این مرحله، الگوریتم ژنتیک (GA) برای حل مسئله واگذاری وظایف استفاده می‌شود و نتایج آن به عنوان برچسب برای آموزش GNN مورد استفاده قرار می‌گیرد.
- GA با استفاده از اصول تکاملی، برچسب‌های داده‌های آموزشی را به‌طور تکراری بهبود می‌بخشد و کیفیت داده‌ها برای آموزش GNN را افزایش می‌دهد.

۲. مرحله آموزش آفلاین:

- پس از دریافت نمونه‌های برچسب‌گذاری‌شده، شبکه GNN تحت یادگیری نظارت‌شده (supervised learning) آموزش داده می‌شود.
- این مرحله به GNN اجازه می‌دهد تا الگوها و ویژگی‌های پیچیده موجود در داده‌های برچسب‌گذاری‌شده را یاد بگیرد و توانایی تصمیم‌گیری دقیق برای واگذاری وظایف را به دست آورد.

۳. مرحله استنتاج آنلاین:

- پس از اتمام آموزش GNN، شبکه می‌تواند در ایستگاه پایه مستقر شود و به‌طور خودکار تصمیمات واگذاری وظایف را اتخاذ کند.
- با استفاده از این روش، GNN قادر است استراتژی‌های واگذاری بهینه را تعیین کند و عملکرد شبکه را بهبود دهد.

سپس هر ماژول چارچوب مبتنی بر یادگیری به تفصیل معرفی می‌شود:

- ابتدا توضیح داده می‌شود که چگونه از GA برای حل مسئله واگذاری وظایف استفاده شود.
- سپس نحوه تبدیل اطلاعات شبکه خودرویی به گراف قابل استفاده برای GNN و نحوه استفاده از برچسب‌های حاصل از GA برای آموزش GNN و کمک به تصمیم‌گیری برای واگذاری وظایف تشریح می‌گردد.

۴.۳ آموزش GNN با داده‌های شبکه خودرویی

در یک شبکه خودرویی پویا واقعی، حرکت مداوم وسایل نقلیه باعث تغییرات قابل توجه در توپولوژی شبکه می‌شود، از جمله تغییرات در لینک‌های ارتباطی بین وسایل نقلیه و تعداد وسایل نقلیه در محدوده پوشش یک ایستگاه پایه (BS).

در حالی که شبکه‌های مبتنی بر DNN، مانند RL، می‌توانند تغییرات پویا در وضعیت شبکه‌های خودرویی را پردازش کنند، عدم مقیاس‌پذیری آن‌ها باعث می‌شود نتوانند اطلاعات با ابعاد ثابت را پردازش کنند و تغییرات توپولوژی انعطاف‌پذیر شبکه را پوشش دهند. در مقابل، GNN قابلیت سازگاری و مقیاس‌پذیری بالایی دارد و برای این نوع شبکه‌ها بسیار مناسب است.

تبدیل اطلاعات شبکه خودرویی به گراف برای GNN

برای استفاده از GNN، اطلاعات شبکه خودرویی به یک گراف G_t تبدیل می‌شود:

- گره‌ها: (Nodes)
 - گره‌های وظیفه (Task nodes): ویژگی‌ها شامل اندازه داده d_i و نیاز محاسباتی c_i هستند.
 - گره‌های محاسباتی (Computing nodes): ویژگی‌ها شامل توان محاسباتی f_j و سیله خدمات‌دهنده می‌باشد.
- یال‌ها: (Edges)
 - یال‌ها ارتباط بین وسایل نقلیه را نشان می‌دهند.
 - ویژگی‌های یال‌ها شامل نرخ انتقال داده بین گره‌ها است که بر تأخیر کل شبکه تأثیرگذار است.

فرآیند آموزش GNN

۱. به روزرسانی ویژگی‌های گره‌ها:

- هر گره ویژگی‌های همسایگان خود را جمع‌آوری (aggregate) می‌کند:

$$m_{i,j}^k = \text{AGG}(h_i^k, h_j^k), A_{i,j} \neq 0 \quad 1-8$$

که در آن h_i^k و h_j^k وضعیت مخفی گره‌های i و j در لایه k هستند و $m_{i,j}^k$ پیام استخراج شده از گره همسایه i به گره j است.

۲. ترکیب ویژگی‌های گره و همسایگان:

$$h_i^{k+1} = \sigma(W^{k+1} \cdot C^k(h_i^k, \alpha_{i,j} m_{i,j}^k, j \in \mathcal{N}(i))) \quad 1-9$$

- C^k عمل concatenation است و ویژگی‌های گره و همسایگان را ترکیب می‌کند.

- σ تابع فعال‌سازی (Activation function) است.

- $\alpha_{i,j}$ وزن توجه (Attention weight) که تاثیر گره همسایه j روی گره i را ارزیابی می‌کند:

$$\alpha_{i,j} = \frac{\exp(\text{LeakyReLU}([W_1 \psi_i \parallel W_2 \psi_j]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}([W_1 \psi_i \parallel W_2 \psi_k]))} \quad 1-10$$

۳. ایجاد embedding برای یال‌ها و پیش‌بینی واگذاری وظایف:

$$x_{i,j} = \text{MLP}(C(h_i^k, h_j^k, h_{i,j})) \quad 1-11$$

- ترکیب ویژگی‌های گره‌ها و یال‌ها برای پیش‌بینی تصمیم واگذاری وظایف استفاده می‌شود.

- خروجی $x_{i,j}$ تعیین می‌کند که وظیفه گره i به گره j واگذار شود یا خیر.

۴. آموزش GNN با برچسب‌های GA:

- تصمیمات واگذاری حاصل از الگوریتم ژنتیک (GA) به عنوان برچسب‌های آموزشی استفاده می‌شوند.
- خطای پیش‌بینی با Cross-Entropy Loss محاسبه شده و پارامترهای GNN از طریق Backpropagation به‌روزرسانی می‌شوند.
- این فرآیند تکرار می‌شود تا الگوریتم به یک حالت پایدار برسد.

۵. استفاده آنلاین از GNN:

- پس از آموزش، GNN در BS مستقر شده و به‌طور خودکار تصمیمات واگذاری وظایف را اتخاذ می‌کند.
- این فرآیند شامل استنتاج از ویژگی‌های جدید شبکه و تعیین بهترین استراتژی‌های واگذاری در زمان واقعی است.

۵.۱ تنظیمات آزمایش

در این بخش، ارزیابی جامع (G-TORA چارچوب واگذاری وظایف و تخصیص منابع مبتنی بر GNN) از طریق شبیه‌سازی‌های عددی انجام شد.

۵.۱.۱ پیکربندی پارامترهای شبکه

پارامترهای شبکه خودرویی در جدول ۱ نشان داده شده‌اند. تنظیمات به شرح زیر است:

- اندازه وظایف 10: (Task size): {۲۰، ۵۰، ۱۰۰} مگابایت
- نیاز محاسباتی وظایف $10^9 \times [1, 2]$: (Computation requirements) سیکل
- توان محاسباتی وسایل خدمات‌دهنده [3، 5]: (Computational capacity) گیگاهرتز
- قدرت انتقال هر وسیله نقلیه 100: (Transmit power) میلی‌وات
- نویز کانال 100 dBm/Hz: (Channel noise power)
- پهنای باند کانال 10 V2V: (Bandwidth) مگاهرتز
- فاصله بین وسایل نقلیه: (Distance) بین ۲۰ تا ۵۰۰ متر

فرآیند آماده‌سازی داده‌ها و آموزش GNN

قبل از تبدیل پارامترهای شبکه به گراف و وارد کردن آن‌ها به: GNN

۱. نرمال‌سازی داده‌ها: (Normalization)

○ ویژگی‌های گره‌ها مانند توان محاسباتی و نیاز محاسباتی وظایف با min-max scaling به بازه $[0, 1]$ تبدیل شدند.

○ این کار باعث جلوگیری از غالب شدن یک ویژگی بر دیگر ویژگی‌ها و حفظ تفاوت نسبی بین آن‌ها می‌شود.

۲. تنظیم وزن‌های یال‌ها: (Edge weights)

○ وزن‌های یال‌ها بر اساس نسبت سیگنال به نویز (SNR) به صورت دینامیک تنظیم شدند.

○ این کار کیفیت ارتباطات و تأثیر آن بر تاخیر شبکه را به طور دقیق منعکس می‌کند.

الگوریتم آموزش GNN برای واگذاری وظایف

ورودی: نمونه‌های شبکه خودرویی و برچسب‌های واگذاری وظایف حاصل از GA

خروجی: یک مدل GNN آموزش‌دیده برای واگذاری وظایف

مراحل:

۱. تبدیل اطلاعات شبکه خودرویی به داده‌های گرافی شامل گره‌ها، یال‌ها و ویژگی‌ها و ذخیره در Train

Loader.

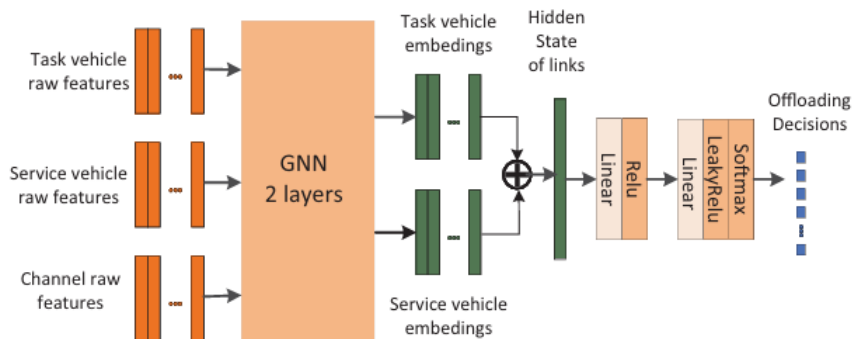
۲. مقداردهی اولیه مدل GNN با تعداد لایه‌ها، مکانیزم‌های attention و سایر هایپرپارامترها.

۳. مقداردهی اولیه بهینه‌ساز (Optimizer) و تابع خطا. (Loss Function)

۴. برای هر Epoch:

○ نمونه‌برداری یک Batch از Train Loader.

- وارد کردن داده‌های گراف به مدل.
 - اعمال مکانیزم attention برای تجمع اطلاعات از گره‌های همسایه و به‌روزرسانی embedding گره‌ها.
 - ترکیب ویژگی‌های گره‌ها و یال‌ها برای محاسبه تصمیمات واگذاری وظایف هر وسیله.
 - مقایسه پیش‌بینی‌های GNN با برچسب‌های GA.
 - محاسبه خطای کل batch با Cross-Entropy Loss.
 - انجام Backpropagation برای محاسبه گرادیان‌ها و به‌روزرسانی وزن‌های GNN با استفاده از Adam Optimizer.
۵. ذخیره پارامترهای آموزش‌دیده GNN و استقرار آن در BS برای اتخاذ تصمیمات واگذاری وظایف در زمان واقعی.



شکل ۳) معماری شبکه‌های مبتنی بر GNN

جدول (۱) تنظیمات شبیه سازی

مقدار	پارامتر
10، 20، 50، 100 MB	{اندازه داده وظایف
$10^9 \times [1, 2]$ سیکل	نیاز محاسباتی وظایف
[3, 5] GHz	توان محاسباتی وسایل خدمات دهنده
100 mW	قدرت انتقال
100 dBm/Hz	نویز کانال
10 MHz	پهنای باند
[20، 500] m	فاصله بین وسایل نقلیه

این تنظیمات و پیش پردازش ها باعث می شوند GNN بتواند با کیفیت بالا و دقت مناسب تصمیمات واگذاری وظایف را در شبکه های خودرویی پویا پیش بینی کند.

۵.۱.۲ پیکربندی پلتفرم شبیه سازی

در این مقاله، تنظیمات پلتفرم شبیه سازی به شرح زیر است:

- لایه های کانولوشن $L = 2$: GNN
- ابعاد embedding گره ها $d = 256$:
- ابعاد لایه های مخفی $d = 128$: MLP
- تعداد Epoch ها برای بهینه سازی 3000:
- بهینه ساز Adam Optimizer:

- نرخ یادگیری اولیه $lr = 1 \times 10^{-3}$:
- از Multi-step decay برای نرخ یادگیری استفاده شده است، با milestones در Epoch های ۱۰۰، ۲۰۰، ۵۰۰، ۱۰۰۰ و ۲۰۰۰ فاکتور $\gamma = 0.8$ decay
- محیط نرم افزاری و سخت افزاری:
- محیط توسعه Python 3.8 :
- فریم ورک یادگیری عمیق PyTorch :
- سخت افزار CPU Intel i7-12700F نسل ۱۲ و GPU Nvidia GeForce RTX 3060 Ti

۵.۲ الگوریتم های مرجع (Baseline Algorithms)

برای ارزیابی عملکرد G-TORA، آن را با چهار الگوریتم مرجع مقایسه کردیم:

۱. الگوریتم ژنتیک: (GA)

- یک الگوریتم هوریستیک مبتنی بر انتخاب طبیعی و وراثت ژنتیکی.
- جزئیات فرآیند در بخش ۴.۲ توضیح داده شده است.
- ۲. Hill Climbing (CH):

- یک الگوریتم جستجوی محلی که با بررسی راه حل های همسایه، راه حل فعلی را بهبود می دهد.
- سریع تر از GA است اما معمولاً در بهینه محلی گیر می کند.

۳. Multi-layer Perceptron (MLP):

- شبکه عصبی ساده با ابعاد ورودی و خروجی ثابت.
- برای مقایسه با روش های دیگر:
- اگر تعداد وسایل مأموریت دهنده بیشتر از ابعاد ورودی MLP باشد، وسایل اضافی محاسبات را به صورت محلی انجام می دهند.

- اگر تعداد وسایل کمتر باشد، ابعاد خالی با صفر پر می‌شوند.

۴. Online Multi-Armed Bandit (OMAB):

- از رویکرد ϵ -greedy برای تعادل بین کاوش (exploration) و بهره‌برداری (exploitation) استفاده می‌کند.
- تصمیمات را در زمان واقعی و تحت شرایط عدم قطعیت بهینه می‌کند.

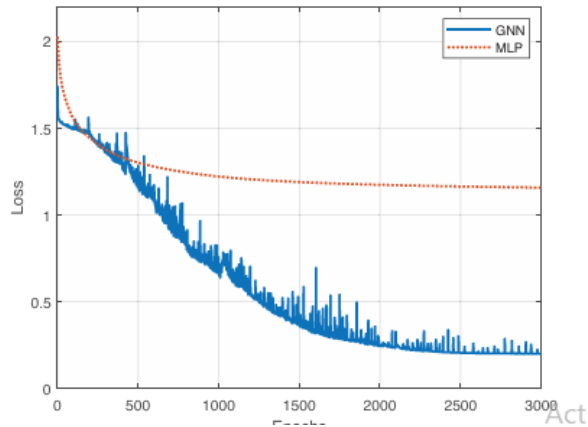
نمودارها و نتایج شبیه‌سازی

- شکل ۴: منحنی Loss برای GNN و MLP نشان می‌دهد که مدل GNN با توجه به داده‌های گراف شبکه توانسته به همگرایی مناسبی برسد.
- شکل ۵: تأخیر پردازش با افزایش تعداد وسایل مأموریت‌دهنده نمایش داده شده است.

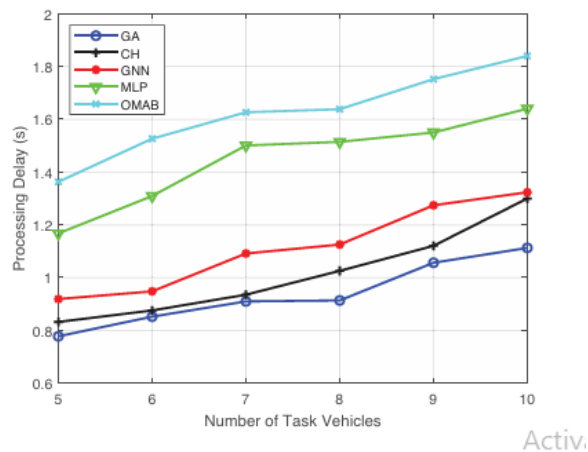
- (GNN-based) G-TORA عملکرد بهتری نسبت به GA، CH، MLP و OMAB نشان می‌دهد.

- این تأخیر شامل کل زمان از ورود وظیفه تا اجرای تصمیم و تکمیل وظیفه است.

این بخش نشان می‌دهد که چارچوب (G-TORA) GNN با در نظر گرفتن ویژگی‌های پویا و توپولوژیکی شبکه‌های خودرویی، می‌تواند تأخیر پردازش را به‌طور قابل توجهی کاهش دهد و نسبت به الگوریتم‌های مرجع عملکرد بهتری داشته باشد.



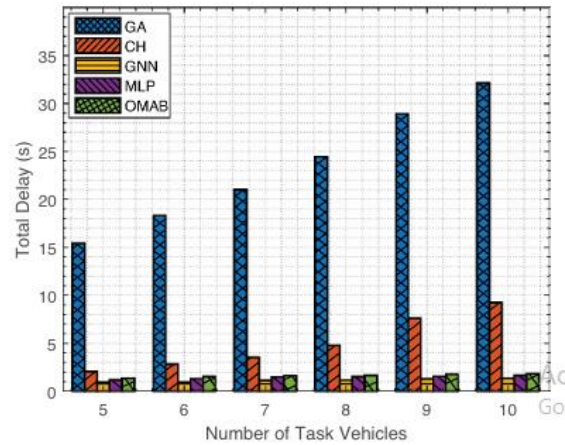
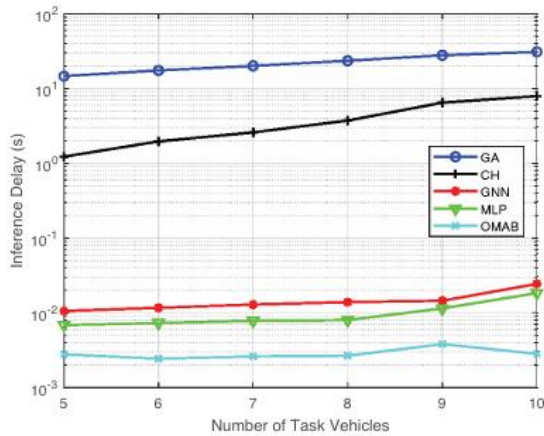
شکل ۴) نمودار اتلاف GNN و MLP



شکل ۵) پردازش تاخیر و افزایش تعداد تسک های خودروها

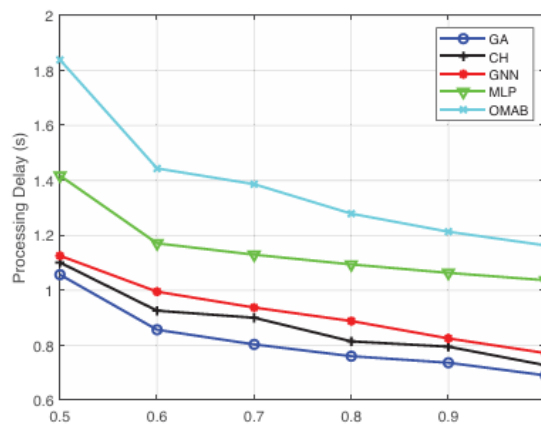
در یک آزمایش برای بررسی تأثیر تغییر تعداد وسایل مأموریت‌دهنده روی عملکرد الگوریتم‌ها، تعداد وسایل خدمات‌دهنده ثابت و برابر با ۴ و تعداد وسایل مأموریت‌دهنده بین ۵ تا ۱۰ تغییر یافت. هدف، مقایسه تاخیر شبکه از سه منظر بود: تاخیر پردازش وظایف (زمان لازم برای ارسال و پردازش وظایف)، زمان استنتاج تصمیم (زمان تصمیم‌گیری الگوریتم) و تاخیر کل (جمع دو مورد قبل). نتایج نشان داد با افزایش تعداد وسایل مأموریت‌دهنده، تمام روش‌ها به دلیل رقابت برای منابع محدود با تاخیر بیشتری مواجه می‌شوند. الگوریتم GA کمترین تاخیر را داشت و کارآمدی بالای خود در مسائل بهینه‌سازی پیچیده را نشان داد، CH عملکرد متوسط و مستعد گیرکردن در بهینه محلی بود، در حالی که GNN به دلیل توانایی مدل‌سازی توپولوژی‌های پویا عملکرد خوبی در شبکه‌های بزرگ داشت، اما به دلیل وابستگی به برچسب‌های آموزش نظارت‌شده کمی عقب‌تر از GA

قرار گرفت. به طور کلی، افزایش تعداد وسایل مأموریت‌دهنده باعث افزایش تأخیر کل می‌شود و GNN نسبت به روش‌های ساده مانند MLP عملکرد بهتری نشان می‌دهد، در حالی که GA بهترین عملکرد تأخیر را ارائه می‌دهد اما محاسبات آن پیچیده و زمان‌بر است.



شکل ۶) زمان استنتاج تصمیم با افزایش تعداد وسایل مأموریت‌دهنده

شکل ۷) افزایش زمان استنتاج و تأخیر کل با افزایش تعداد وسایل مأموریت‌دهنده



شکل ۸) توان انتقالی

۵.۴ تأثیر تغییر توان انتقال (Transmit Power)

در این مجموعه از آزمایش‌ها، اثر تغییر توان انتقال بر عملکرد الگوریتم‌ها بررسی شد. نتایج نشان داد که افزایش توان انتقال باعث کاهش تاخیر پردازش وظایف در تمام الگوریتم‌ها می‌شود.

- GA کمترین تاخیر را داشت و از ۱.۰۵۶ ثانیه به ۰.۶۹۰ ثانیه کاهش یافت.
- CH نیز کاهش قابل توجهی از ۱.۱۰۱ به ۰.۷۲۷ ثانیه داشت.
- GNN تاخیر خود را از ۱.۱۲۶ به ۰.۷۷۱ ثانیه کاهش داد و نشان‌دهنده عملکرد رقابتی آن بود.
- در مقابل، MLP و OMAB تاخیر بالاتری داشتند، اگرچه افزایش توان انتقال برای آن‌ها نیز مفید بود، اما به سطح کارایی GA، CH یا GNN نرسید.

زمان استنتاج تصمیم (Inference Delay) تقریباً با افزایش توان انتقال ثابت باقی ماند، زیرا اندازه کل مسئله در طول شبیه‌سازی تغییر نکرده است. تاخیر کل نیز با افزایش توان انتقال برای GA و CH تقریباً ثابت بود، در حالی که برای GNN، MLP و OMAB کاهش قابل توجهی مشاهده شد. دلیل این تفاوت این است که تاخیر کل GA و CH بیشتر تحت تأثیر زمان استنتاج است، اما در GNN، MLP و OMAB، تاخیر پردازش غالب است و کاهش زمان ارتباطات باعث کاهش کل تاخیر می‌شود. GNN همچنان پایین‌ترین تاخیر کل را حفظ کرد و از ۱.۱۳۵ به ۰.۷۸۰ ثانیه کاهش یافت.

۵.۵ تأثیر تغییر توان محاسباتی (Computing Capacity)

برای بررسی تأثیر منابع محاسباتی، توان محاسباتی سرویس‌دهنده‌ها از ۳ تا ۷ گیگاهرتز تغییر یافت. نتایج نشان داد که با افزایش توان محاسباتی:

- تمام الگوریتم‌ها کاهش واضحی در تاخیر پردازش وظایف داشتند، که نشان‌دهنده اثر مثبت توان محاسباتی بیشتر بر کارایی اجرای وظایف است.
- تاخیر کل برای GA و CH تقریباً ثابت ماند، زیرا تحت تأثیر زمان استنتاج است.
- برای GNN، MLP و OMAB، کاهش واضحی در تاخیر کل مشاهده شد، زیرا کاهش زمان پردازش منجر به کاهش تاخیر کل می‌شود.

- GNN پایین‌ترین تأخیر کل را حفظ کرد و از ۰.۹۲۳ به ۰.۸۱۷ ثانیه کاهش یافت، که نشان‌دهنده تعادل مؤثر بین دقت بهینه‌سازی و استنتاج بلادرنگ است.

۶. جمع‌بندی

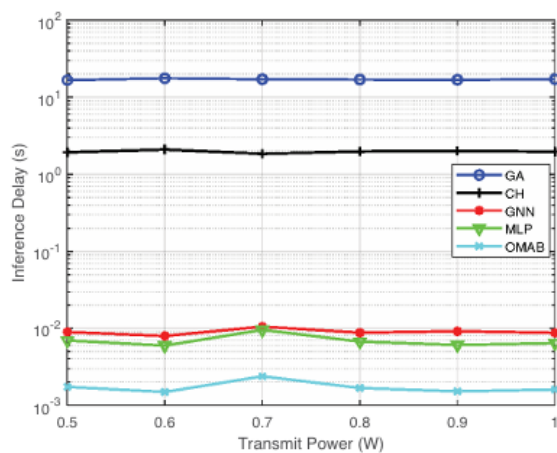
واگذاری وظایف و تخصیص منابع در شبکه‌های خودرویی برای افزایش کارایی محاسباتی و کاهش تأخیر حیاتی هستند و خدمات و اپلیکیشن‌های درون خودرو را پاسخگو و قابل اعتماد می‌کنند. با این حال، روش‌های موجود اغلب به شبکه‌های عصبی با معماری ثابت وابسته‌اند و قادر به مدیریت محیط‌های پویا و ورودی‌های متغیر نیستند. برای رفع این محدودیت‌ها، G-TORA، یک چارچوب مبتنی بر GNN ارائه شد که توانایی یادگیری استراتژی‌های واگذاری وظایف را در شبکه‌های پویا و با توپولوژی‌های متغیر دارد. در این روش، وظایف خودرو و وسایل خدمات‌دهنده به عنوان گره و لینک‌های ارتباطی به عنوان یال‌ها مدل می‌شوند. GNN با استخراج ویژگی‌های ساختاری و روابط شبکه، تصمیمات واگذاری بلادرنگ را پیش‌بینی می‌کند.

نتایج شبیه‌سازی نشان می‌دهد که G-TORA تحت شرایط شبکه متغیر و پویا عملکرد خوبی دارد، مقیاس‌پذیری بالایی دارد و کارایی پردازش وظایف را بهبود می‌بخشد. با این حال، محدودیت‌هایی نیز دارد:

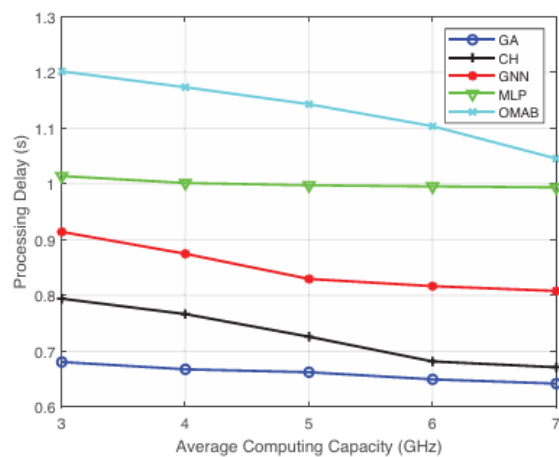
۱. آموزش به حجم بالایی از داده‌های توپولوژی و برجسب‌های به دست آمده از الگوریتم‌های بهینه‌سازی مانند GA وابسته است.

۲. چارچوب فرض می‌کند وضعیت ارتباطی در هر بازه زمانی ثابت است که با شبکه‌های خودرویی واقعی و بسیار پویا کاملاً همخوانی ندارد.

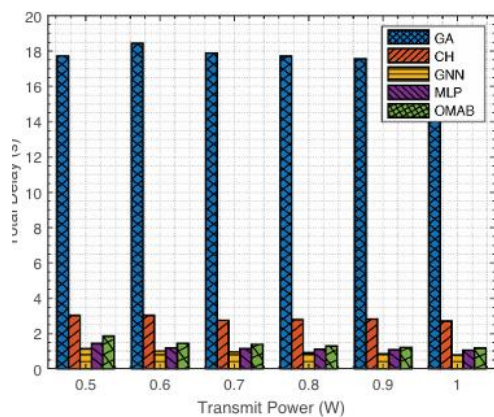
در آینده قصد داریم با استفاده از روش‌های یادگیری نیمه‌نظارتی یا بدون نظارت، مدل‌سازی بلادرنگ شبکه و توسعه چارچوب برای پشتیبانی از واگذاری وظایف با در نظر گرفتن QoS، مهلت‌ها، قابلیت اطمینان و انتقال چند مرحله‌ای، این محدودیت‌ها را برطرف کنیم. این بهبودها قابلیت تطبیق و عملی بودن راهکارهای مبتنی بر GNN در محیط‌های خودرویی پویا را افزایش خواهد داد.



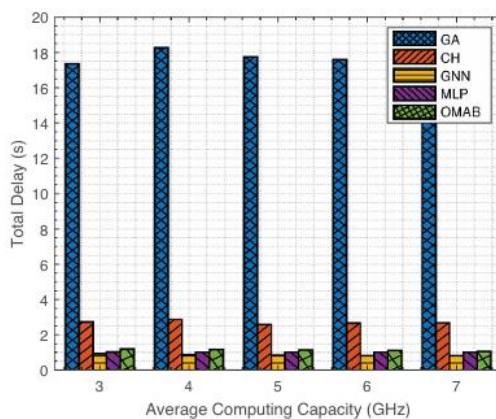
شکل ۹) تاخیر تداخل با افزایش توان انتقال



شکل ۱۱) تاخیر پردازش با افزایش ظرفیت محاسباتی



شکل ۱۰) تاخیر کل با افزایش توان انتقالی



شکل ۱۲) تاخیر کل با افزایش ظرفیت محاسباتی

منابع

1. E. S. Ali, M. K. Hasan, R. Hassan, et al., "Machine Learning Technologies for Secure Vehicular Communication in Internet of Vehicles: Recent Advances and Applications," *Security and Communication Networks* 2021 (2021): 1–23.
2. A. Waheed, M. A. Shah, S. M. Mohsin, et al., "A Comprehensive Review of Computing Paradigms, Enabling Computation Offloading and Task Execution in Vehicular Networks," *IEEE Access* 10 (2022): 3580–3600.
3. A. B. De Souza, P. A. L. Rego, T. Carneiro, J. D. C. Rodrigues, P. P. R. Filho, and J. N. De Souza, "Computation Offloading for Vehicular Environments: A Survey," *IEEE Access* 8 (2020): 198214–198243.
4. C. Feng, P. Han, X. Zhang, B. Yang, Y. Liu, and L. Guo, "Computation Offloading in Mobile Edge Computing Networks: A Survey," *Journal of Network and Computer Applications* 202 (2022): 202–216.
5. M. Talebkhan, A. Sali, V. Khodamoradi, T. Khodadadi, and M. Gordan, "Task Offloading for Edge-IoV Networks in the Industry 4.0 Era and Beyond: A High-Level View," *Engineering Science and Technology* 54 (2024): 1–40.
6. K. Luo, Y. Wang, Y. Liu, and K. Zhu, "Collaborative Integration of Vehicle and Roadside Infrastructure Sensor for Temporal Dependency Aware Task Offloading in the Internet of Vehicles," *International Journal of Intelligent Systems* 2025 (2025): 8064–8086.
7. R. Rauch, Z. Becvar, P. Mach, J. Gazda, "Cooperative Multi-Agent Deep Reinforcement Learning for Dynamic Task Execution and Resource Allocation in Vehicular Edge Computing," *IEEE Transactions on Vehicular Technology* 74, no. 4 (2025): 5741–5756.
8. Y. He, J. Xu, B. Zheng, J. Hu, and Y. Xie, "Timing-Oriented Task Offloading Algorithms for Internet-Of-Vehicles," *Journal of Circuits Systems and Computers* 31, no. 08 (2022).
9. A. Boukerche and V. Soto, "Computation Offloading and Retrieval for Vehicular Edge Computing: Algorithms, Models, and Classification," *ACM Computing Surveys* 53, no. 4 (2021): 1–35.
10. J. Liu, M. Ahmed, M. A. Mirza, et al., "RL/DRL Meets Vehicular Task Offloading Using Edge and Vehicular Cloudlet: A Survey," *IEEE Internet of Things Journal* 9, no. 11 (2022): 8315–8338.
11. T. Li, K. Zhu, N. C. Luong, et al., "Applications of Multi-Agent Reinforcement Learning in Future Internet: A Comprehensive Survey," *IEEE Communications Surveys Tutorials* 24, no. 2 (2022): 1240–1279.
12. S. Midya, A. Roy, K. Majumder, and S. Phadikar, "Multi-Objective Optimization Technique for Resource Allocation and Task Scheduling in Vehicular Cloud Architecture: A Hybrid Adaptive Nature Inspired Approach," *Journal of Network and Computer Applications* 103 (2018): 58–84.
13. A. R. Hameed, S. ul Islam, I. Ahmad, and K. Munir, "Energy and Performance-Aware Load-Balancing in Vehicular Fog Computing," *Sustainable Computing-Informatics & Systems* 30 (2021): 1–16.
14. A. Bozorgchenani, S. Maghsudi, D. Tarchi, and E. Hossain, "Computation Offloading in Heterogeneous Vehicular Edge Networks: On-Line and Off-Policy Bandit Solutions,"

IEEE Transactions on Mobile Computing 21, no. 12 (2022): 4233–4248.

15. A. Barbosa, P. A. L. Rego, V. Chamola, T. Carneiro, P. H. G. Rocha, and J. N. de Souza, "A Bee Colony-Based Algorithm for Task Offloading in Vehicular Edge Computing," IEEE Systems Journal 17, no. 3 (2023): 4165–4176. 16. A. Mekrache, A. Bradai, E. Moulay, and S. Dawaliby, "Deep Reinforcement Learning Techniques for Vehicular Networks: Recent Advances and Future Trends Towards 6G," Vehicular Communications 33 (2022): 1–23. 17. Y. Li and S. Xu, "Collaborative Optimization of Edge-Cloud Computation Offloading in Internet of Vehicles," in Proceedings of the 2021 12 of 13 IET Communications, 2025 International Conference on Computer Communications and Networks (ICCCN) (IEEE, 2021), 1–6. 18. A. Asheralieva and D. Niyato, "Hierarchical Game-Theoretic and Reinforcement Learning Framework for Computational Offloading in UAV-Enabled Mobile Edge Computing Networks With Multiple Service Providers," IEEE Internet of Things Journal 6, no. 5 (2019): 8753–8769. 19. H. Liang, X. Zhang, J. Zhang, Q. Li, S. Zhou, and L. Zhao, "A Novel Adaptive Resource Allocation Model Based on SMDP and Reinforcement Learning Algorithm in Vehicular Cloud System," IEEE Transactions on Vehicular Technology 68, no. 10 (2019): 10018–10029. 20. Z. Yan and H. Tabassum, "Reinforcement Learning for Joint V2I Network Selection and Autonomous Driving Policies," in 2022 IEEE Global Communications Conference (GLOBECOM 2022) (IEEE, 2022), 1241–1246. 21. H. Peng and X. Shen, "Deep Reinforcement Learning Based Resource Management for Multi-Access Edge Computing in Vehicular Networks," IEEE Transactions on Network Science and Engineering 7, no. 4 (2020): 2416–2428. 22. Y. Wang, W. Fang, Y. Ding, and N. Xiong, "Computation Offloading Optimization for UAV-Assisted Mobile Edge Computing: A Deep Deterministic Policy Gradient Approach," Wireless Networks 27, no. 4 (2021): 2991–3006. 23. I. Budhiraja, N. Kumar, H. Sharma, M. Elhoseny, Y. Lakys, and J. J. P. C. Rodrigues, "Latency-Energy Tradeoff in Connected Autonomous Vehicles: A Deep Reinforcement Learning Scheme," IEEE Transactions on Intelligent Transportation Systems 24, no. 11 (2022): 1–13. 24. J. Wu, J. Wang, Q. Chen, et al., "Resource Allocation for Delay Sensitive Vehicle-to-Multi-Edges (V2Es) Communications in Vehicular Networks: A Multi-Agent Deep Reinforcement Learning Approach," IEEE Transactions on Network Science and Engineering 8, no. 2 (2021): 1873–1886. 25. P. Tehrani, F. Restuccia, and M. Levorato, "Federated Deep Reinforcement Learning for the Distributed Control of NextG Wireless Networks," in 2021 IEEE International Symposium on Dynamic Spectrum Access Networks (IEEE, 2021), 248–253. 26. X. Ma, J. Zhao, Q. Li, et al., "Reinforcement Learning Based Task Offloading and Take-Back in Vehicle Platoon Networks," in 2019 IEEE International Conference on Communications Workshops (ICC Workshops) (IEEE, 2019), 1–6. 27. 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 23, no. 4 (2024): 3074–3085. 28. J. Wang, C. Jiang, K. Zhang, T. Q. S. Quek, Y. Ren, and L. Hanzo, "Vehicular Sensing Networks in a Smart City: Principles, Technologies and Applications," IEEE Wireless Communications 25, no. 1 (2018): 122–132. 29. X. Chen, L. Jiao, W.

Li, et al., "Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing," IEEE/ACM Transactions on Networking 24, no. 5 (2015): 2795–2808. 30. 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 69, no. 12 (2020): 16067–16081. 31. X. Wang, N. Cheng, L. Fu, et al., "Scalable Resource Management for Dynamic Mec: An Unsupervised Link-Output Graph Neural Network Approach," in 2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) (IEEE, 2023), 1–6