

۱. تخصیص منابع آگاه از معناشناصی با DRL در HetNet های 5G-V2X

۱. جزئیات مسئله و نوآوری کلیدی:

چالش: در V2X، همه پیام‌ها ارزش یکسان ندارند. یک پیام "تصادف فوری" حیاتی‌تر از یک پیام "اطلاعات سرگرمی" است. روش‌های سنتی (مانند ماکریم کردن Throughput کل) این تفاوت را در نظر نمی‌گیرند.

راه حل هسته: معرفی "شاخص اهمیت معنایی" (Semantic Importance Score - SIS). این شاخص می‌تواند بر اساس:

نوع پیام (استاندارد GPP^۳) vs. Decentralized Environmental Notification Message (DENM)

محتوای پیام: تحلیل اولیه متن/داده (مثلاً با NLP سبک) برای استخراج فوریت.

موقعیت و زمینه: نزدیکی به تقاطع خطرناک، شرایط جوی.

نوآوری: ادغام SIS در تابع پاداش (Reward) الگوریتم DRL. پاداش نه تنها بر اساس معیارهای فیزیکی (نرخ داده، SINR) بلکه بر اساس "انتقال موفق پیام‌های با اهمیت بالا" تعریف می‌شود.

۲. متدولوژی و روش تحقیق:

مدل سیستم:

شبکه: یک HetNet شامل ماکروسل (eNB)، پیکو سل (Pico-GNB) و ارتباطات (PC5).

کاربران: وسایل نقلیه با ارتباطات V2V (برای اینمی) و V2I (برای اینفوتیمنت).

منابع: باند فرکانسی ۹.۵ گیگاهرتز با باندهای فرعی (RBs) و سطوح توان قابل تخصیص.

فرمول‌سازی مسئله به عنوان MDP:

State (وضعیت): $s_t = \text{Channel Quality Indicator (CQI)} = \text{Buffer status}$ هر Buffer status (CQI)] برای تمام لینک‌ها، هر [

وسیله، Semantic Importance Score پیام‌های منتظر، موقعیت/سرعت نسبی وسایل، تاریخچه تداخل] .

Action (عمل): $a_t = \text{[تخصیص RB خاص به هر لینک V2V/V2I, سطح توان ارسال برای هر لینک, برای وسایل کم اهمیت'`Cycle]$

$r_t = \sum (\text{ 成功传输? } \text{ SIS_of_Message}) - \alpha \sum (\text{Interference}) - \beta \sum (\text{Power_Consumed})$

الگوریتم DRL:

انتخاب PPO: به دلیل پایداری، کارایی در فضای عمل پیوسته-گسسته و مناسب برای محیط‌های غیرایستا.

معماری شبکه عصبی: یک Actor-Critic با لایه های تمام متصل (FC) و احتمالاً لایه های LSTM/Attention برای درک و استنگی زمانی.

روند آموزش:

۱. شبیه ساز: پلتفرم شبیه سازی شبکه مبتنی بر Simu5G یا Python با 自定義 (OpenAI Gym).

۲. اکتشاف: استفاده از نویز گاوی در خروجی Actor برای کشف سیاست های جدید.

۳. آموزش: به روز رسانی پارامتر ها با بهینه سازی clipping surrogate objective در PPO.

۴. دیتاست و ارزیابی:

دیتاست ترافیک و ارتباطات:

شبیه سازی ترافیک: SUMO (Simulation of Urban MObility) برای تولید حرکت واقع گرایانه وسایل در یک سناریو شهری (مثلًا خیابان های منهتن).

شبیه سازی کanal: مدل ثلفات مسیر (Fading) و محوشگی (Path Loss) استاندارد GPP TR 38.901^۳ برای mmWave/Sub-6GHz فرکانس های.

تولید پیام های V2X: تولید دوره ای پیام های CAM (۱۰ هرتز) و تصادفی پیام های DENM بر اساس رویدادهای شبیه سازی شده در SUMO.

برچسب اهمیت معنایی: اختصاص دستی یا قاعده بندی شده SIS به هر نوع پیام (مثلًا CAM 10, SIS 7, DENM 3) اینمی: CAM معمولی: ۱، دیتای اینفوگریونت: ۳.

معیارهای ارزیابی (KPI):

Primary (اصلی): Packet Reception Rate (PRR) وزندار با SIS برای پیام های حیاتی.
Secondary (فرعی): میانگین تأخیر سرتاسری، توان مصرفی کل، طیف کارایی (bps/Hz)، عادلانه بودن Fairness (Benevolence) بین کاربران.

روش های مقایسه ای (Baseline):

۱. تصادفی (Random): تخصیص منابع به طور تصادفی.

۲. حریصانه (Greedy): تخصیص منابع به بهترین کanal بدون در نظر گرفتن تداخل.

۳. DRL سنتی: همان الگوریتم اما بدون SIS در Reward (فقط ماکسیمم کردن PRR کل).

۴. الگوریتم مبتنی بر نظریه بازی (Game Theory).

۴. نتایج کلیدی (به طور فرضی):

روش پیشنهادی ۳۵٪ بهبود در PRR پیام های با اهمیت بالا (SIS>7) نسبت به DRL سنتی نشان می دهد.

در عین حال، افزایش ۱۵٪ در بهره وری طیف کلی به دلیل مدیریت هوشمندتر تداخل.

کاهش ۲۰٪ در تأخیر پیام های حیاتی.

تحلیل Sensitivity: نشان می دهد وزن پارامتر 'a' در تابع Reward چگونه موازنی بین کیفیت سرویس و مصرف انرژی را کنترل می کند.

۲. UniTSA: کنترل چراغ راهنمایی جهان شمل با RL

۱. جزئیات معماری UniTSA:

مشکل روش های RL خاص تقطیع: نیاز به جمع آوری داده و آموزش مجدد برای هر تقطیع جدید. غیر عملی در مقیاس شهر.

هسته راه حل UniTSA: Standardized Observation (مشاهده استاندارد).

Junction Matrix دقيقاً چیست؟ یک تانسور ۳ بعدی با ابعاد '[N_Approaches, M_Lanes, K_Features]'.

'N_Approaches': تعداد ورودی های تقطیع (مثلًا ۴).

'M_Lanes': تعداد خطوط هر ورودی (مثلًا ۲: چپ+مستقیم/راست).

'K_Features': شامل [تعداد وسائل نقلیه در صفر، سرعت متوسط، تعداد عابران پیاده منتظر، فاز چراغ فعلی، زمان مانده فاز فعلی، اولویت وسائل نقلیه اضطراری (آمبولانس)، اطلاعات V2X از وسائل نزدیک].

این استانداردسازی به عامل اجازه می دهد الگوهای مشترک کنترل ترافیک (مثل مدیریت صفر، اولویت دهنده به جریان اصلی) را بیاموزد، فارغ از هندسه دقیق تقطیع.

۲. متدولوژی و معماری RL:

فرمول سازی MDP:

State: `s_t = Junction_Matrix_Flattened` (تبديل به بردار).

Action: `a_t = [انتخاب فاز بعدی از مجموعه فاز های پیش تعریف شده، تعیین مدت زمان این فاز (در یک بازه گسته مثلًا ۱۰-۶۰ ثانیه)]`.

Reward: `r_t = a - (δ Throughput + β تعداد توقف ها + γ تأخیر عابران پیاده - مجموع طول صفحات)`

الگوریتم RL:

استفاده از Rainbow DQN (ترکیبی از قابلیت های پیشرفته DQN) به دلیل فضای عمل گستره و نیاز به تخمین ارزش دقیق.

شبکه: یک CNN (برای درک ساختار مکانی ماتریس) متصل به لایه های FC.

روند آموزش جهانی:

۱. محیط آموزش متتنوع: یک شبیه ساز (مثلًا SUMO یا CityFlow با تقاطع های بسیار متتنوع) با هزاران تقاطع با هندسه ها، جریان های ترافیک و الگوهای تقاضای مختلف.
۲. آموزش: عامل در این محیط غنی شده، یک سیاست تعمیم یافته (Generalized Policy) می آموزد.
۳. استقرار: برای یک تقاطع کاملاً جدید، عامل تنها نیاز به دریافت 'Junction Matrix' آن دارد و بلافاصله می تواند تصمیم گیری بهینه را آغاز کند (Zero-Shot Transfer).

۳. دیتاست و ارزیابی:

محیط شبیه سازی: CityFlow (یک شبیه ساز کنترل ترافیک با کارایی بالا).

دیتاست های ترافیک:

آموزش: ترکیبی از داده های واقعی از چندین شهر (مثلًا پکن، شانگهای) و داده های تولید شده مصنوعی با الگوهای متفاوت (ساعت اوج، شب، آخر هفته).

آزمون: شهر ها یا نواحی مشخصاً جداگانه که در داده آموزش دیده نشده اند (مثلًا داده های از برلین).

معیار های ارزیابی:

میانگین زمان سفر، میانگین زمان انتظار، تعداد توقف ها در هر سفر.

نرخ گذر (Throughput) مجموع وسائل عبوری در ساعت.

Consistency: بررسی عملکرد یکسان عامل در تقاطع های مشابه.

: Baseline ها

۱. Fixed-Time: زمان بندی ثابت از پیش محاسبه شده.

۲. MaxPressure: الگوریتم کلاسیک مبتنی بر فشار ترافیک.

۳. IntelliLight (یا سایر RL های خاص- تقاطع): که برای هر تقاطع جداگانه آموزش دیده اند.

۴. FRAP (یک روش RL پیشین).

۴. نتایج کلیدی (به طور فرضی):

عامل UniTSA در تقاطع های جدید، بدون هیچ آموزش اضافی، عملکردی ۹۵٪ بهینه نسبت به یک عامل RL که به طور خاص بر روی همان تقاطع آموزش دیده دارد.

کاهش ۳۰-۴۰٪ در زمان سفر و زمان انتظار نسبت به روش Fixed-Time.

مقیاس پذیری: استقرار در ۱۰۰۰ تقاطع تنها نیازمند کپی یک مدل واحد، در مقابل آموزش و نگهداری ۱۰۰۰ مدل مجزا.

۳. محاسبات لبۀ خودرویی تقویت شده با RIS و MADRL

۱. جزئیات فنی مسئله سه‌گانه:

سه تصمیم در همتایه (Joint Optimization Problem):

۱. تخصیص منابع محاسباتی (Computation Resource Allocation - CRA): یک سرور لبۀ شبکه با منابع محدود (CPU سیکل/ثانیه) باید بین وظایف آفلوید شده از چندین وسیله تقسیم کند.

۲. تصمیم آفلوید کردن وظیفه (Task Offloading Decision - TOD): هر وسیله باید تصمیم بگیرد کدام وظیفه را به کدام سرور لبۀ شبکه بفرستد (یا به صورت محلی پردازش کند).

۳. تنظیم فاز (RIS Phase Shift Configuration - RPC): فاز هر عنصر RIS (مثلاً ۲۵۶ عنصر) باید طوری تنظیم شود که سیگنال‌های ارسالی از وسائل به سمت RSU را به طور سازنده جمع کند و نویز/تداخل را کاهش دهد.

جالش: این سه تصمیم عمیقاً وابسته به هم هستند. RIS خوب، کانال بهتری می‌سازد، که باعث می‌شود آفلوید کردن مقرن به صرفه‌تر شود و تخصیص منابع سرور را تحت تأثیر قرار می‌دهد.

۲. متولوژی BCD و MADRL

مدلسازی به عنوان بازی چندعامله (Multi-Agent Game):

عاملان: هر وسیله نقلیه یک عامل مستقل و هر سرور لبۀ شبکه (پاکنترلر مرکزی RIS) یک عامل است.

هدف: همکاری برای بیشینه کردن پاداش سیستم (کاهش تأخیر کل و مصرف انرژی).

الگوریتم MADDPG اصلاح شده:

MADDPG استاندارد: هر عامل یک Actor-Critic دارد، اما Critic هر عامل حالت و عمل تمام عاملان را مشاهده می‌کند (Centralized Training). در اجرا، هر عامل فقط بر اساس مشاهده محلی خود عمل می‌کند (Decentralized Execution).

اصلاح در این مقاله: احتمالاً برای پایداری بیشتر، Priority Experience Replay با Attention Mechanism را برای وزن‌دهی به اطلاعات عاملان دیگر در Critic اضافه شده است.

روش ترکیبی MADDPG-BCD:

گام ۱ (بهروزرسانی عوامل): با استفاده از MADDPG، سیاست‌های TOD (برای وسایل) و CRA (برای سرور) بهروزرسانی می‌شوند.

گام ۲ (بهینه‌سازی RIS): با ثابت گرفتن سیاست‌های بهست‌آمده در گام ۱، مسئله تنظیم فاز RIS (که یک مسئله بهینه‌سازی غیرمحدب است) با الگوریتم Block Coordinate Descent (BCD) حل می‌شود. در BCD، فاز عناصر RIS یکی‌یکی یا گروهی بهروزرسانی می‌شوند تاتابع هدف (مثلًا مجموع نرخ داده) بهبود یابد.

این دو گام متناوبًا تکرار می‌شوند تا همگرایی حاصل شود.

۳. دیتاست و ارزیابی:

مدل کanal RIS: مدل تلفات مسیر دقیق با در نظر گرفتن لینک دوگانه (وسیله به RIS، RIS به RSU) بر اساس هندسه.

پروفایل وظایف (Tasks): وظایف با اندازه‌های متفاوت (از ۵.۰ تا ۵ مگابایت) و نیاز محاسباتی متفاوت (از ۵.۰ تا ۳ کیگاسیکل) از دیتاست‌هایی مثل TaskMeter یا Google Cluster Trace گرفته شده‌اند.

معیارهای ارزیابی: تأخیر سرتاسری وظایف، نرخ موفقیت انجام وظیفه در مهلت زمانی (Deadline)، انرژی کل مصرفی (وسیله + سرور)، نرخ داده بهبودیافته به لطف RIS.

: Baseline ها

۱. بدون RIS: فقط MADDPG برای TOD و CRA.

۲. تصادفی/حریصانه محلی: هر وسیله به صورت حریصانه تصمیم می‌گیرد.

۳. الگوریتم مبتنی بر تشبع (Optimization-Based).

۴. نتایج کلیدی (به طور فرضی):

سیستم پیشنهادی کاهش ۴۵٪ در تأخیر متوسط نسبت به حالت بدون RIS نشان می‌دهد.

افزایش ۶۰٪ در نرخ موفقیت انجام وظایف با مهلت زمانی سخت.

تحلیل کارایی-هزینه: نشان می‌دهد با افزایش تعداد عناصر RIS از ۶۴ به ۲۵۶، بهبود عملکرد به تدریج اشباع می‌شود.

۴. کمی‌سازی گرادیان تطبیقی با DRL توزیع شده برای FL در VEC

۱. جزئیات فنی کمی‌سازی و ارتباط:

کمی‌سازی (Quantization): تبدیل مقادیر ممیز شناور (۳۲ بیتی) گرادیان به مقادیر صحیح با بیت کمتر (مثلًا ۴ بیت، ۲ بیت). این کار حجم داده ارسالی را به شدت کاهش می‌دهد.

چالش اصلی: کمی‌سازی زیاد (بیت کم) نویز را افزایش می‌دهد و ممکن است باعث و اگرایی یا کندی همگرایی مدل فرال شود. کمی‌سازی کم (بیت زیاد) صرفهجویی کمی دارد.

نوآوری: تطبیقی و مبتنی بر شرایط شبکه/داده. عامل DRL در هر وسیله، در هر دور آموزش فرال، سطح کمی‌سازی (تعداد بیت) را به طور پویا انتخاب می‌کند.

۲. متولوژی DRL توزیع شده:

فرمول‌سازی MDP محلی برای هر وسیله:

State (وضعیت محلی): s_{t^i} = [Channel Gain] فعلی, Battery Level, اندازه و اهمیت (ناهمگنی) داده‌های محلی, تاریخچه خطای کمی‌سازی در دورهای قبل, نرخ یادگیری جهانی سرور]

Action (عمل محطی): a_{t^i} = سطح کمی‌سازی (تعداد بیت) برای این دور (مثلاً انتخاب از {۱۶, ۸, ۴, ۲, ۱} بیت)

Reward (پاداش محلی): $r_{t^i} = -(زمان انتقال + انرژی مصرفی برای ارسال) - \lambda$ (اختلاف بین گرادیان کمی‌شده و اصلی)

آموزش عامل‌ها: هر عامل در محیط شبیه‌سازی محلی خود (که شامل یک مدل از کانال متغیر و داده‌های محلی است) به طور مستقل آموزش می‌بیند. هیچ اشتراک اطلاعات حساسی بین وسایل وجود ندارد. سپس مدل‌های آموزش‌دیده عامل‌ها به همه وسایل توزیع می‌شوند.

یکپارچه‌سازی با چرخه FL:

۱. سرور مدل جهانی w_t را پختن می‌کند.
۲. هر وسیله i ، بر اساس حالت s_{t^i} ، سطح کمی‌سازی a_{t^i} را از عامل DRL خود می‌گیرد.
۳. با داده‌های محلی، $\nabla L_i(w_t)$ را محاسبه و با سطح بیت انتخاب شده کمی‌سازی می‌کند: $Q(\nabla L_i(w_t); a_{t^i})$
۴. گرادیان کمی‌شده را به سرور می‌فرستد.
۵. سرور گرادیان‌ها را جمع می‌کند و مدل را بهروز می‌فرستد: $w_{t+1} = w_t - \eta \sum_i Q(\nabla L_i(w_t); a_{t^i})$.

۳. دیتاست و ارزیابی:

aset یادگیری: یک کار واقعی در VEC، مثلاً تشخیص عابر پیاده (با دیتاستی مثل COCO) یا پیش‌بینی ترافیک (با داده‌های GPS).

ناهمگنی داده: تقسیم غیریکنواخت دیتاست بین وسایل (مثلاً برخی وسایل فقط عکس‌های روز، برخی فقط شب دارند) تا شرایط واقعی VEC شبیه‌سازی شود.

معیارهای ارزیابی:

دقت (Accuracy) مدل جهانی نهایی بر روی یک تست است یکپارچه.

سرعت همگرایی: تعداد دورهای لازم برای رسیدن به دقت هدف.

هزینه ارتباطی کل: مجموع بیت‌های ارسالی از همه وسائل در طول آموزش.

انرژی مصرفی کل.

: Baseline ها

۱. کمی‌سازی ثابت (Fixed Quantization): مثلاً همیشه ۴ بیت.

۲. SignSGD: یک روش کمی‌سازی افراطی (۱ بیت).

۳. AdaQuant: روش‌های تطبیقی کلاسیک (غیر DRL).

۴. نتایج کلیدی (به طور فرضی):

روش DRL-تطبیقی به همان دقت نهایی روش بدون کمی‌سازی می‌رسد اما با ۷۰٪ کاهش ترافیک ارتباطی.

در مقایسه با کمی‌سازی ثابت ۴ بیت، ۲۰٪ سریع‌تر همگرا می‌شود زیرا در دورهای اولیه که گرادیان‌ها بزرگ‌ترند، از دقت بیشتری استفاده می‌کند.

وسائل با کanal ضعیف یا باتری کم، به طور خودکار سطح کمی‌سازی پایین‌تری (صرف کمتر) انتخاب می‌کنند.

۵. تخصیص منابع با GNN+DRL برای V2X

۱. جزئیات معماری GNN:

چرا گراف؟ شبکه V2X ذائقاً یک گراف است:

گره‌ها: وسائل نقلیه (V) و نقاط دسترسی زیرساخت (I).

یال‌ها: لینک‌های مطلوب (ارتباطات مورد نظر) و لینک‌های تداخل (اگر دو لینک از یک RB مشابه استفاده کنند و در محدوده هم باشند).

نوع گراف: گراف هتروژن (Heterogeneous) با دو نوع گره (V و I) و احتمالاً دو نوع یال.

استخراج ویژگی گره اولیه: هر گره ویژگی‌های خودش را دارد: موقعیت، CQI به نقاط دسترسی، نیاز منابع (اندازه بسته)، اولویت.

پیامرسانی در GNN (با GraphSAGE):

۱. Aggregate: هر گره (مثلاً یک وسیله)، ویژگی‌های همسایگانش در گراف (وسائل دیگر که با آن تداخل دارند یا نقاط دسترسی نزدیک) را جمع‌آوری می‌کند. عمل Aggregate می‌تواند Max، Mean LSTM یا باشد.

۲. Update: ویژگی های جمع آوری شده با ویژگی های فعلی خود گره ترکیب شده و از طریق یک شبکه عصبی (مثلًا یک لایه FC) عبور داده می شود تا نمایه تعبیه شده (Embedding) جدید آن گره تولید شود: $h_{v^l} = \sigma(W^l h_{v^{l-1}} + b^l)$

۳. این فرآیند برای چند لایه (مثلًا ۲ یا ۳) تکرار می شود. در پایان، هر گره یک نمایه تعبیه شده غنی (h_{v_final}) دارد که نه تنها اطلاعات خودش، بلکه اطلاعات زمینه ای شبکه (تدالو های احتمالی، شلوغی محلی) را نیز در بر می گیرد.

۲. متدولوزی ادغام GNN+DRL:

معماری ترکیبی:

ماژول GNN: دریافت کننده گراف شبکه و تولیدکننده embeddings برای همه گره ها.

ماژول DRL (مثلًا DQN):

ورودی: embeddings تمام گره های درخواست کننده منبع (یا یک خلاصه از آن ها).

خروجی: برای هر لینک V2V/V2I فعال، احتمال تخصیص هر RB موجود.

انتخاب عمل: با توجه به خروجی DQN، یک مکانیزم (مثلًا گرسیلی یا نمونبرداری) RB ها را به لینک ها اختصاص می دهد.

تابع پاداش DQN: بر اساس معیار های سیستم پس از تخصیص، مانند: $\sum r_t = \text{نرخ داده لینک های موفق} - \sum \mu_i (\text{تدالو انداز مگری شده})$

۳. دیتاست و ارزیابی:

شبیه سازی: یک پلتفرم سفارشی یا گسترش یافته (مثلًا با Python و کتابخانه های Ray RLlib + DGL/PyG) برای (GNN).

سناریو: یک خیابان یا تقاطع با تراکم متغیر وسائل (از ۵۰ تا ۲۰۰ وسیله در کیلومتر مربع).

معیار های ارزیابی:

نرخ تحويل بسته (PDR) برای لینک های V2V.

نرخ داده برای لینک های V2I.

Latency برای پیام های V2V.

Spectral Efficiency کل سیستم.

Baseline ها:

۱. تصادفی (Random).

۲. حریصانه (Greedy) مبتنی بر CQI.

۳. الگوریتم مبتنی بر تطابق (Matching-Based).

۴. DRL خالص (بدون GNN): که حالت را به صورت یک بردار مسطح از همه ویژگی‌ها دریافت می‌کند.

۴. نتایج کلیدی (به طور فرضی):

DRL+GNN نسبت به DRL خالص، ۲۵٪ بهبود در PDR لینک‌های V2V در سناریوهای با تراکم بالا نشان می‌دهد، زیرا الگوی تداخل پیچیده را بهتر درک می‌کند.

قابلیت تعمیم (Generalization): مدل آموخته شده بر روی یک توپولوژی جاده‌ای، در یک توپولوژی متفاوت (مثلًاً از خط مستقیم به نقاطع) عملکرد بهتری نسبت به DRL خالص دارد.

تحلیل توجه (اگر از Attention استفاده شده باشد): نشان می‌دهد GNN به کدام همسایگان (تداخل‌گران اصلی) بیشتر "توجه" کرده است.