

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

۱. جزئیات مسئله و نوآوری کلیدی:

چالش: در V2X، همه پیام ها ارزش یکسان ندارند. یک پیام "تصادف فوری" حیاتی تر از یک پیام "اطلاعات سرگرمی" است. روش های سنتی (مانند ماکزیم کردن Throughput کل) این تفاوت را در نظر نمی گیرند.

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

نوع پیام (استاندارد GPP^۳): Cooperative Awareness Message (CAM) vs. Decentralized Environmental Notification Message (DENM) با درجه فوریت.

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 پیام های با اهمیت بالا" تعریف می شود.

۲. متدولوژی و روش تحقیق:

مدل سیستم:

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

کاربران: وسایل نقلیه با ارتباطات V2V (برای ایمنی) و V2I (برای اینفو تیمنت).

منابع: باند فرکانسی ۵.۹ گیگاهرتز با باندهای فرعی (RBs) و سطوح توان قابل تخصیص.

فرمول سازی مسئله به عنوان MDP:

State (وضعیت): $s_t = [CQI]$ Channel Quality Indicator برای تمام لینک ها، Buffer status هر وسیله، Semantic Importance Score پیام های منتظر، موقعیت/سرعت نسبی وسایل، تاریخچه تداخل

Action (عمل): $a_t = [RB]$ تخصیص RB خاص به هر لینک V2V/V2I، سطح توان ارسال برای هر لینک، Duty Cycle برای وسایل کم اهمیت

Reward (پاداش): $r_t = \sum (成功传输? SIS_of_Message) - \alpha \sum (Interference) - \beta \sum (Power_Consumed)$

الگوریتم DRL:

انتخاب PPO: به دلیل پایداری، کارایی در فضای عمل پیوسته-گسسته و مناسب برای محیط های غیر ایستا.

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

روند آموزش:

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

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

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

۳. 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 (مثلاً خیابان های منتهن).

شبیه سازی کانال: مدل تلفات مسیر (Path Loss) و محوشدگی (Fading) استاندارد $GPP\ TR\ 38.901^3$ برای فرکانس های mmWave/Sub-6GHz.

تولید پیام های V2X: تولید دوره های 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 (مثلاً CAM: 10, DENM: 10, ایمنی: ۷, CAM معمولی: ۳, دیتای اینفوتیمنت: ۱).

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) بین کاربران.

روش های مقایسه ای (Baseline):

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

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

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

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

۴. نتایج کلیدی (به طور فرضی):

روش پیشنهادی ۳۵٪ بهبود در PRR پیام‌های با اهمیت بالا ($SIS > 7$) نسبت به DRL سنتی نشان می‌دهد.

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

کاهش ۲۰٪ در تأخیر پیام‌های حیاتی.

تحلیل Sensitivity: نشان می‌دهد وزن پارامتر α در تابع Reward چگونه موازنه بین کیفیت سرویس و مصرف انرژی را کنترل می‌کند.

۲. UniTSA: کنترل چراغ راهنمایی جهان‌شمل با RL

۱. جزئیات معماری UniTSA:

مشکل روش‌های RL خاص-تقاطع: نیاز به جمع‌آوری داده و آموزش مجدد برای هر تقاطع جدید. غیرعملی در مقیاس شهر.

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

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

$N_Approaches$: تعداد ورودی‌های تقاطع (مثلاً ۴).

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

$K_Features$: شامل [تعداد وسایل نقلیه در صف, سرعت متوسط, تعداد عابران پیاده منتظر, فاز چراغ فعلی, زمان مانده فاز فعلی, اولویت وسایل نقلیه اضطراری (آمبولانس), اطلاعات V2X از وسایل نزدیک].

این استانداردسازی به عامل اجازه می‌دهد الگوهای مشترک کنترل ترافیک (مثل مدیریت صف, اولویت‌دهی به جریان اصلی) را بیاموزد, فارغ از هندسه دقیق تقاطع.

۲. متدولوژی و معماری RL:

فرمول‌سازی MDP:

State: $s_t = \text{Junction_Matrix_Flattened}$ (تبدیل به بردار).

Action: a_t = [انتخاب فاز بعدی از مجموعه فازهای پیش‌تعریف شده, تعیین مدت زمان این فاز (در یک بازه گسسته مثلاً ۱۰-۶۰ ثانیه)].

Reward: $r_t = -(\alpha \times \text{مجموع طول صف‌ها} + \beta \times \text{تعداد توقف‌ها} + \gamma \times \text{تأخیر عابران پیاده} - \delta \times \text{Throughput})$

الگوریتم RL:

استفاده از Rainbow DQN (ترکیبی از قابلیت های پیشرفته DQN) به دلیل فضای عمل گسترده 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 متنوع: یک شبیه ساز (مثلاً CityFlow یا SUMO) با تقاطع های بسیار متنوع (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 (مثلاً پکن، شانگهای) و داده های تولید شده مصنوعی با الگوهای متفاوت (ساعت اوج، شب، آخر هفته).

آزمون: شهرها یا نواحی مشخصاً جداگانه که در داده آموزش دیده نشده اند (مثلاً داده های از برلین).

معیارهای ارزیابی:

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

نرخ گذر (Throughput) مجموع وسایل 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 لبه خودرویی تقویت شده با RIS و MADRL

۱. جزئیات فنی مسئله سه گانه:

سه تصمیم در هم تنیده (Joint Optimization Problem):

۱. تخصیص منابع محاسباتی (Computation Resource Allocation - CRA): یک سرور لبه 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 لبه 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 سیگنال های ارسالی از وسایل به سمت RSU را به طور سازنده جمع کند و نویز/تداخل را کاهش دهد.

چالش: این سه تصمیم عمیقاً وابسته به هم هستند. RIS خوب، کانال بهتری می سازد، که باعث می شود آفلوید کردن مقرون به صرفه تر شود و تخصیص منابع سرور را تحت تأثیر قرار می دهد.

۲. متدولوژی MADRL و BCD:

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

عاملان: هر وسیله نقلیه یک عامل مستقل و هر سرور لبه شبکه (یا کنترلر 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 می کند (Centralized Training). در اجرا، هر عامل فقط بر اساس مشاهده محلی خود عمل می کند (Decentralized Execution).

اصلاح در این مقاله: احتمالاً برای پایداری بیشتر، Priority Experience Replay یا Attention Mechanism برای وزن دهی 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 به‌دست‌آمده 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 تا تابع هدف (مثلاً مجموع نرخ داده) بهبود یابد. این دو گام متناوباً تکرار می‌شوند تا همگرایی حاصل شود.

۳. دیتاست و ارزیابی:

مدل کانال 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 ۰.۵ تا ۵ مگابایت) و نیاز محاسباتی متفاوت (از ۰.۵ تا ۳ گیگاسیکل) از دیتاست‌هایی مثل Google Cluster Trace یا TaskMeter 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.

Baselineها:

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

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

۳. الگوریتم مبتنی بر تشعشع (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 اشیاع می‌شود.

۴. کمّی‌سازی گرادیان تطبیقی با DRL توزیع‌شده برای FL در VEC

۱. جزئیات فنی کمّی‌سازی و ارتباط:

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

چالش اصلی: کمّی سازی زیاد (بیت کم) نویز را افزایش می دهد و ممکن است باعث واگرایی یا کندی همگرایی مدل فدرال شود. کمّی سازی کم (بیت زیاد) صرفه جویی کمی دارد.

نوآوری: تطبیقی و مبتنی بر شرایط شبکه/داده. عامل DRL در هر وسیله، در هر دور آموزش فدرال، سطح کمّی سازی (تعداد بیت) را به طور پویا انتخاب می کند.

۲. متدولوژی DRL توزیع شده:

فرمول سازی MDP محلی برای هر وسیله:

State (وضعیت محلی): s_t^i = [Battery Level, Channel Gain فعلی, اندازه و اهمیت (ناهمگنی) داده های محلی, تاریخچه خطای کمّی سازی در دورهای قبل, نرخ یادگیری جهانی سرور]

Action (عمل محلی): a_t^i = سطح کمّی سازی (تعداد بیت) برای این دور (مثلاً انتخاب از $\{۲, ۴, ۸, ۱۶\}$ بیت)

Reward (پاداش محلی): r_t^i = -(زمان انتقال + انرژی مصرفی برای ارسال) - λ (اختلاف بین گرادیان کمّی شده و اصلی)

آموزش عامل ها: هر عامل در محیط شبیه سازی محلی خود (که شامل یک مدل از کانال متغیر و داده های محلی است) به طور مستقل آموزش می بیند. هیچ اشتراک اطلاعات حساسی بین وسایل وجود ندارد. سپس مدل های آموزش دیده عامل ها به همه وسایل توزیع می شوند.

یکپارچه سازی با چرخه FL:

۱. سرور مدل جهانی w_t را پخش می کند.

۲. هر وسیله i ، بر اساس حالت s_t^i ، سطح کمّی سازی a_t^i را از عامل DRL خود می گیرد.

۳. با داده های محلی، $\nabla L_i(w_t)$ را محاسبه و با سطح بیت انتخاب شده کمّی سازی می کند: $Q(\nabla L_i(w_t); a_t^i)$.

۴. گرادیان کمّی شده را به سرور می فرستد.

۵. سرور گرادیان ها را جمع می کند و مدل را به روز می کند: $w_{t+1} = w_t - \eta \sum_i Q(\nabla L_i(w_t); a_t^i)$.

۳. دیتاست و ارزیابی:

دیت aset یادگیری: یک کار واقعی در VEC، مثلاً تشخیص عابر پیاده (با دیتاستی مثل Cityscapes یا COCO) یا پیش بینی ترافیک (با داده های GPS).

ناهمگنی داده: تقسیم غیر یکنواخت دیتاست بین وسایل (مثلاً برخی وسایل فقط عکس های روز، برخی فقط شب دارند) تا شرایط واقعی VEC شبیه سازی شود.

معیار های ارزیابی:

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

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

هزینه ارتباطی کل: مجموع بیت های ارسالی از همه وسایل در طول آموزش.

انرژی مصرفی کل.

Baseline ها:

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

۲. SignSGD: یک روش کمی سازی افراطی (۱ بیت).

۳. AdaQuant: روش های تطبیقی کلاسیک (غیر DRL).

۴. نتایج کلیدی (به طور فرضی):

روش DRL-تطبیقی به همان دقت نهایی روش بدون کمی سازی می رسد اما با ۷۰٪ کاهش ترافیک ارتباطی.

در مقایسه با کمی سازی ثابت ۴ بیت، ۲۰٪ سریع تر همگرا می شود زیرا در دورهای اولیه که گرادین ها بزرگترند، از دقت بیشتری استفاده می کند.

وسایل با کانال ضعیف یا باتری کم، به طور خودکار سطح کمی سازی پایین تری (مصرف کمتر) انتخاب می کنند.

۵. تخصیص منابع با GNN+DRL برای V2X

۱. جزئیات معماری GNN:

چرا گراف؟ شبکه V2X ذاتاً یک گراف است:

گره ها: وسایل نقلیه (V) و نقاط دسترسی زیرساخت (I).

یال ها: لینک های مطلوب (ارتباطات مورد نظر) و لینک های 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 ویژگی های خودش را دارد: موقعیت، CQI به نقاط دسترسی، نیاز منابع (اندازه بسته)، اولویت.

پیام رسانی در GNN (با GraphSAGE):

۱. Aggregate: هر گره (مثلاً یک وسیله)، ویژگی های همسایگانش در گراف (وسایل دیگر که با آن تداخل دارند یا نقاط دسترسی نزدیک) را جمع آوری می کند. عمل Aggregate می تواند Mean، Max یا LSTM باشد.

۲. Update: ویژگی های جمع آوری شده با ویژگی های فعلی خود گره ترکیب شده و از طریق یک شبکه عصبی (مثلاً یک لایه FC) عبور داده می شود تا نمایه تعبیه شده (Embedding) جدید آن گره تولید شود: $h_v^{(l+1)} = \sigma(W^l \cdot \text{CONCAT}(h_v^l, \text{AGGREGATE}(\{h_u^l, \forall u \in N(v)\})))$

۳. این فرآیند برای چند لایه (مثلاً 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 (تداخل های احتمالی، شلوغی محلی) را نیز در بر می گیرد.

۲. متدولوژی ادغام GNN+DRL:

معماری ترکیبی:

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

ماژول DRL (مثلاً 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 (مثلاً گرسیلی یا نمونه برداری) RB ها را به لینک ها اختصاص می دهد.

تابع پاداش DQN: بر اساس معیارهای سیستم پس از تخصیص، مانند: $r_t = \sum$ (نرخ داده لینک های موفق) - $\mu \Sigma$ (تداخل اندازه گیری شده)

۳. دیتاست و ارزیابی:

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

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

معیارهای ارزیابی:

نرخ تحویل 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 ویژگی ها دریافت می کند.

۴. نتایج کلیدی (به طور فرضی):

GNN+DRL نسبت به DRL خالص، ۲۵٪ بهبود در PDR لینک های V2V در سناریوهای با تراکم بالا نشان می دهد، زیرا الگوی تداخل پیچیده را بهتر درک می کند.

قابلیت تعمیم (Generalization): مدل آموزش دیده بر روی یک توپولوژی جاده ای، در یک توپولوژی متفاوت (مثلاً از خط مستقیم به تقاطع) عملکرد به مراتب بهتری نسبت به DRL خالص دارد.

تحلیل توجه (اگر از Attention استفاده شده باشد): نشان می دهد GNN به کدام همسایگان (تداخل گران اصلی) بیشتر "توجه" کرده است.