زمانبندی چراغ راهنمایی در حالت اشباع با استفاده از یادگیری تقویتی

مقاله علمی – پژوهشی

شهریار افندی زاده *، استاد، دانشکده مهندسی عمران، دانشگاه علم و صنعت ایران، تهران، ایران
محمود احمدی نژاد، استاد، دانشکده مهندسی عمران، دانشگاه علم و صنعت ایران، تهران، ایران
علیرضا موحدی، دانشجوی کارشناسی ارشد، دانشکده مهندسی عمران، دانشگاه علم و صنعت ایران، تهران، ایران
حمید بیگدلی‌راد، دانشجوی دکتری، دانشکده مهندسی عمران، دانشگاه علم و صنعت ایران، تهران، ایران
*پست الکترونیکی نویسنده مسئول: zargari@iust.ac.ir

دریافت: ۱٤۰۳/۱۰/۱۵– پذیرش: ۱٤۰٤/۰۳/۰۱
صفحه ٤٤-۲۷

چکیده

اگرچه مدلسازی شبکه‌ها به امری پیچیده و دشوار تبدیل شده است و مدل‌سازی را برای نزدیک‌تر شدن به شرایط محیط با مشکلاتی مواجه می‌کند، در این میان چارچوب یادگیری تقویتی به عنوان یک روش مستقل از مدل می‌تواند نقش بهتری را در کنترل و شبیه‌سازی ترافیک فراهم کند. در این مطالعه سعی بر آن است که با استفاده از الگوریتم‌های مختلف یادگیری تقویتی، همچون الگوریتم‌های DQN و DDPG، بتوان به شیوه‌ای سریع‌تر و منظم‌تر شبکه ترافیکی در نظر گرفته شده را شبیه‌سازی کرده و بتوان موارد تاثیرگذاری همچون طول صف تشکیل شده در خیابان‌ها و چراغ‌های راهنمایی را با بکارگیری الگوریتم‌ها و برنامه ریزی مناسب، به شیوه‌ای نوین در جهت کاهش میزان ترافیک و روان‌سازی آن، بهینه کرد و با توجه به نتایج به دست آمده از دو الگوریتم ذکر شده، الگوریتمی که عملکرد بهتری داشت به عنوان الگوریتم برتر از زیر مجموعه الگوریتم یادگیری تقویتی مطرح شود. در نهایت، شبکه خود را با کاهش طول صف و همچنین کاهش میزان زمان توقف در پشت چراغ راهنمایی در شبکه‌های شهری در حالت اشباع، موجب بهبود عبور، مرور و روان‌سازی جریان ترافیک می‌شود. روش مطرح شده بر روی تقاطعات چراغ دار لندن در جنوب غربی استان انتاریو۲۷، در کشور کانادا انجام شده است. نتایج پیاده سازی این روش نشان می‌دهد که با استفاده از DDPG معیارهای ترافیکی مانند میانگین زمان ایستادن خودروها و درصد خودروها ساکن در کل شهر کاهش ملموسی پیدا می‌کنند.

واژه‌های کلیدی: برنامه‌ریزی شهری، جریان ترافیک، شبکه‌های حمل و نقل، یادگیری تقویتی

۱-مقدمه

شبکه حمل و نقل شهری را معمولاً به دو بخش تقاطع‌ها و معابر تقسیم می‌کنند، در این بین ظرفیت تقاطع‌ها نقش بسزایی در کارایی شبکه دارد. تقاطع‌ها به عنوان گره‌های شبکه حمل و نقل، بیشترین تاخیر را به کاربران شبکه وارد می‌کنند. طی دهه‌های گذشته تلاش‌های بسیاری برای زمان‌بندی مناسب تقاطع‌ها صورت گرفته است که همچنان نیز ادامه دارند. تقاطع‌های چراغ‌دار اجزای بحرانی هر سیستم حمل و نقل شهری می‌باشند، از این رو حفظ و کنترل این تقاطع‌ها در حالت بهینه برای هر شرایطی و هر میزان تقاضایی از دغدغه‌های اصلی مهندسان ترافیک می‌باشد (Hajisoleimani et al., 2021). هماهنگی سیگنال‌ها (چراغ‌های راهنمایی) به تنهایی یا با یکدیگر توسط بسیاری از آژانس‌ها به عنوان یک بهبود سودمند برای جامعه یا کریدور در نظر گرفته می‌شود. در بسیاری از موارد، تکنیک‌های هماهنگی سیگنال در بهبود کیفیت زندگی و تحرک در منطقه موفق بوده است (Qadri et al., 2020; Afandizadeh and Bigdeli Rad, 2021). این هماهنگی نقش بسیار تعیین کننده‌ای در کنترل تراکم و همچنین بهینه‌سازی چراغ راهنمایی دارد و اگر به صورت مناسب به آن پرداخته

نشود، مشکلات فراوانی را همچون پس‌زدگی صف حتی در زمانی که چراغ راهنمایی مورد نظر در شبکه سبز می‌باشد را به وجود می‌آورد که متاسفانه امروزه شاهد بسیاری از این موارد می باشیم. از شبیه‌سازی مناسب می‌توان به راهکاری مفید برای حل این معضل آشکار و گسترده استفاده نمود (Afandizadeh et al., 2023). شرایط اشباع و فوق اشباع، در هر شبکه‌ای جزء شرایط بحرانی و خاص تلقی می‌گردد. از این رو مطالعه و تحقیق بر روی انواع روش‌های هماهنگ‌سازی و فازبندی چراغ های راهنمایی این شبکه‌ها می‌تواند گام مهمی در جهت رفاع مشکل ازدحام و پس زدگی صف در این شبکه‌ها باشد (Ameri et al., 2021; Afandizadeh Zargari et al., 2019).

شهرها معمولاً توسط یک شبکه ترافیکی پیچیده که مسئولیت پشتیبانی از نیازهای روزانه در منطقه را برعهده دارد، پوشش داده می‌شود. متاسفانه تقاضای ترافیک بالا، پویا و در حال افزایش است. بهبود زیرساخت روش اولیه برای پاسخگویی به این خواسته‌ها بوده است. با این حال به دلیل محدودیت‌هایی مانند منابع مالی و فضا، همیشه این امکان وجود ندارد (Abdi et al., 2020). این امر منجر به گزینه‌هایی با درنظر گرفتن بهبود زیرساخت‌های موجود، بهینه سازی استفاده از زیرساخت‌های موجود و کاهش هزینه‌های زمان سفر از طریق سیستم‌های حمل و نقل هوشمند[1] شده است(Touhbi, Babram, Nguyen-Huu et al., 2017). کنترل سیگنال ترافیک تطبیقی بیشتر از اوایل دهه هفتاد استفاده شده است و نشان داده است که یک رویکرد مناسب برای کاهش تراکم ترافیک در مقایسه با سیستم‌های کنترل از پیش تعیین شده و فعال برای تقاطع‌های علامت دار است. با توجه به ماهیت تصادفی ترافیک، رویکردی که بتواند خود را با تغییرات ترافیک تطبیق دهد نیازی به مدل مشخصی برای یک محیط خاص نداشته باشد، برای فرآیند کنترل ساده‌تر خواهد بود. از سوی دیگر روش‌های مدرن و فراابتکاری همانند یادگیری تقویتی توانایی سازگاری و خودآموزی از تجربیات گذشته را دارد و بنابراین پتانسیل بیشتری برای بهبود خدمات در طول زمان از طریق تعامل مستمر با محیط دارند. در همین راستا اقداماتی در جهت کاهش میزان ترافیک نظیر، کاهش طول صف و تاخیر و سایر موارد تاثیرگذار در روانسازی و کاهش حجم جریان ترافیک صورت گرفته که در ادامه به مهمترین موارد آن اشاره خواهیم کرد.

۲-پیشینه تحقیق

۲-۱- بهینه سازی زمانبندی چراغ راهنمایی

کمینه‌سازی یا کاهش میزان تاخیر تا حد بسیاری پرکاربردترین تابع هدف برای بهینه‌سازی زمانبندی چراغ راهنمایی می‌باشد. کمینه سازی تاخیر توسط مطالعه (Eom and Kim, 2020) جیو و همکاران (Guo et al., 2019) و تجلی و همکاران (Tajalli et al., 2020)، جهت کاهش زمان سفر برای بهینه‌سازی چراغ راهنمایی و مسیرهای مورد استفاده برای وسایل نقلیه در شبکه مورد استفاده قرار گرفت. ایشان یک برنامه غیر خطی اعداد صحیح مختلط جدید را برای کنترل مسیر وسایل نقلیه متصل مختلط بصورت خودکار (CAVs[2]) و وسایل نقلیه متصل که توسط انسان هدایت می‌شود (CHVs[3]) را از طریق تقاطع‌های سیگنال‌دار توسعه دادند. مسیر CAVها به طور مداوم از طریق یک روش مرکزی بهینه می شود، در حالی که یک فاز "سفید" جدید معرفی می‌شود تا CHV ها را فورا وادار به دنبال کردن خودروهای جلویی خود کنند.

در طول فاز سفید CAVها از طریق تقاطع CHVها را رهبری می‌کنند. فرمول پیشنهادی سیگنال بهینه برای هر گروه خط عبوری در هر مرحله زمانی را مشخص و تعیین می‌کند. روش پیشنهادی برای یک تقاطع مجزا با هشت حرکت تحت هشت سطح CAV اعمال شد.

نتایج این مطالعه نشان می‌دهد که روش حل بصورت موفقیت‌آمیزی مسئله را حل می‌کند و تاخیر کل را در مقایسه با چراغ راهنمایی القایی به میزان ۹۲,۶٪–۱۹,۶٪ کاهش می‌دهد. ازدحام ترافیک در شبکه‌های شهری ممکن است منجر به تخریب شدید در استفاده از زیرساخت شبکه شود که می‌تواند از طریق استراتژی‌های کنترلی مناسب کاهش یابد. سیمون بلدی، لاکاووس میکایلیدیس، وسیلیکی اینتامپاسی و همکاران Baldi et al., 2019) عملکرد یک استراتژی پاسخگو به ترافیک تطبیقی را مورد مطالعه و تجزیه و تحلیل قرار می‌دهند که پارامترهای چراغ راهنمایی را در یک شبکه شهری برای کاهش تراکم ترافیک کنترل می‌کنند. در این مطالعه یک فرمول کنترل تقریباً بهینه در حل مسئله کنترل بهینه مربوطه همیلتون–جاکوبی–بلمن (HJB[4]) رخ می‌دهد، اتخاذ شده است. درابتدا، یک راه حل (تقریبی، تخمینی) HJB از طریق تابع Lyapunov بطورمناسب بصورت پارامتری در می‌آید. سپس، با استفاده از یک شاخص نزدیک به بهینه و اطلاعاتی که از مدل شبیه‌سازی شبکه (طراحی

مبتنی بر شبیه‌سازی) بدست می‌آید، راه‌حل در هر تکرار به‌روزرسانی می‌شود تا به راه‌حل تقریباً بهینه نزدیک شود. الگوریتم پیشنهادی یک شاخص عملکرد متشکل از میانگین سرعت و مسافت کل سفر را به حداکثر می‌رساند. مقیاس‌پذیری کنترل‌کننده، همراه با الگوریتم بکار گرفته شده برای حل معادله تقریبیHJB، روش پیشنهادی را قادر می‌سازد تا مشکلات کنترلی ناشی از شبکه‌های شهری بسیار بزرگ را مدیریت کند. مقایسه‌های عددی بهبودهای مربوطه را از نظر میانگین سرعت، مسافت کل سفر و کل زمان صرف شده در شبکه نشان می‌دهند: علاوه بر این، استراتژی پیشنهادی قادر است استراتژی‌های کنترلی موثری را تحت بسیاری از سناریوهای مختلف تقاضای ترافیک (تقاضای کم، متوسط و زیاد) ارائه دهد. نتایج شبیه‌سازی به‌دست‌آمده با استفاده از مدل شبیه‌سازی ترافیک شبکهChania، یونان، یک شبکه ترافیک شهری حاوی انواع مختلفی از مرحله‌بندی اتصالات، کارایی روش پیشنهادی را در مقایسه با استراتژی‌های ترافیک جایگزین براساس یک مدل خطی ساده‌شده شبکه ترافیک نشان می‌دهد.نشان داده شده است که استراتژی پیشنهادی می‌تواند با شرایط ترافیکی مختلف سازگار شود و پارامترسازی‌های کم پیچیدگی راه‌حل بهینه، یک استراتژی خطی تکه‌ای خطی و دو وجهی، به ترتیب، یک مبادله رضایت‌بخش بین پیچیدگی محاسباتی و عملکرد شبکه فراهم می‌کنند.

۲-۲- هماهنگ‌سازی چراغ‌های راهنمایی با یکدیگر

استفاده از یک مدل کنترل‌کننده پیش‌بینی کننده (MPC[۵]) در یک شبکه ترافیک شهری امکان کنترل زیرساخت شبکه ترافیک و خطاهای موجود در عملیات را فراهم می‌کند. صدیقه جعفری، زینب شهبازی و یونگ چئول بیون (Jafari et al., 2021) یک کنترل کننده پیش‌بینی کننده جدید و پایدار را برای ترافیک شهری پیشنهاد داده‌اند و از دینامیک فضای حالت برای تخمین تعداد وسایل نقلیه در یک تقاطع جدا شده و طول صف آن استفاده می‌شود که این خود یک استراتژی کنترلی جدید براساس نوع چراغ راهنمایی و مدت زمان فاز چراغ سبز است و هدف آن دستیابی به تعادل بهینه در تقاطع‌ها است. روش پیشنهادی با کنترل ترافیک در یک جاده شهری با استفاده از کنترل پیش‌بینی مدل، حجم ترافیک و تعداد تصادفات مربوط به خودروها را کاهش می‌دهد. منطقه‌ای از تهران در این مطالعه برای کنترل زمانبندی سیگنال ترافیک در نظرگرفته شد و مدلی از سیستم‌های ترافیکی در تقاطع استخراج شد و پارامترهای فضای حالت تقاطع با استفاده از کنترل کننده پیش‌بینی مدل برای کنترل علایم ترافیکی براساس طول صف خودرو و تعداد وسایل نقلیه ورودی و خروجی که به عنوان ورودی استفاده می‌شوند، طراحی شد. این فرآیند نشان می‌دهد که این روش می‌تواند حجم ترافیک را در هر سمت یک تقاطع کاهش دهد و جریان را در یک شبکه جاده‌ای در مقایسه با روش زمان ثابت بهینه کند. پایداری تقاطع را با استفاده از کنترل پیش‌بینی مدل بررسی گردید و برای اثبات این پایداری از معادله همیلتونی استفاده شد. با اعمال یک کنترل کننده براساس معادلات فضای حالت گسسته زمانی، طول صف خودرو در مقایسه با حالت زمان ثابت به حداقل رسید و حجم ایجاد شده در هر سمت نیز نتایج مشابهی به همراه داشت. روش پیشنهادی نسبت به کنترل‌کننده زمان ثابت در تمامی خطوط تقاطع عملکرد بهتری داشت. دون فنگ ما، جیو ونگ سیو و سیالونگ ما (Ma et al., 2021) با یک روش کنترل سیگنال پیش‌بینی کننده مدل غیرمتمرکز با توالی فاز ثابت و با استفاده از سیاست فشار برگشتی[۶] پیشنهاد کردند. ایده اصلی روش جدید تشکیل یک حلقه کنترل[۷] با استفاده از کنترل پیش بینی مدل است که سیستم را قادر می سازد تا بازخورد بلادرنگ را از شبکه ترافیک به دست آورد و برنامه‌های زمان بندی سیگنال را به صورت پویا در ابتدای هر فاز تنظیم کند. از آنجایی که پیوندها در یک منطقه خاص دارای طول های مختلفی هستند، طول صف یکسان می‌تواند شرایط ترافیکی متفاوتی را بیان کند و در نتیجه روشی برای عادی سازی طول صف پیشنهاد شده است. بنابراین وقتی طول واقعی صف به ظرفیت پیوند نزدیک می شود، طول صف نرمال شده به شدت کاهش می‌یابد و در نتیجه از سرریز شدن صف جلوگیری می‌شود. روش پیشنهادی می‌تواند با استفاده از پارامترجریان ترافیک زمان واقعی بهبود بیشتری یابد. در همین راستا مدیر ناظر بر پروژه می‌تواند شبکه ترافیک شهری مقیاس بزرگ را به شبکه‌های فرعی تقیسم کند و استراتژی فشار برگشتی را با در نظر گرفتن ویژگی‌های مختلف شبکه‌های فرعی، تنظیم کند.

۲-۳- یادگیری تقویتی

یادگیری تقویتی یک رویکرد داده محور امیدوارکننده برای کنترل سیگنال ترافیک تطبیقی (ATSC[۸]) در شبکه‌های ترافیک شهری پیچیده است و شبکه‌های عصبی عمیق قدرت یادگیری آن را بیشتر می‌کنند(Moerland et al., 2023; Laskin et al., 2020). بیشتر مطالعات موجود در زمینه یادگیری تقویتی چند عاملی (MARL[۹]) بر طراحی ارتباطات و هماهنگی کارآمد در میان عوامل یادگیری Q[۱۰] متمرکز است. تیانشو چو، جی ونگ، لاراکدکا و همکاران (Chu et al., 2019) برای اولین بار یک الگوریتم MARL کاملا مقیاس‌پذیر و غیرمتمرکز را برای عامل RL عمیق[۱۱]، A2C[۱۲] را در چارچوب کنترل سیگنال ترافیک تطبیقی ارائه می‌کند. A2C چند عاملی پیشنهادی با الگوریتم‌های یادگیری مستقل A2C وQ، هم در یک شبکه ترافیک مصنوعی بزرگ و هم در یک شبکه بزرگ ترافیک دنیای واقعی شهر موناکو، تحت دینامیک ترافیک ساعت اوج شبیه‌سازی شده، مقایسه می‌شوند (Kiran et al., 2020). نتایج بهینه بودن، استحکام و کارایی نمونه را نسبت به سایر الگوریتم‌های غیر متمرکز MARL نشان می‌دهد. کنترل ناکارآمد ترافیک ممکن است باعث مشکلات متعددی مانند تراکم ترافیک و اتلاف انرژی شود. ژنینگ لی، هائو یو، شانگ چیا دونگ و همکاران (Li et al., 2021) در این مطالعه یک روش جدید یادگیری تقویتی چند عاملی تحت عنوان KS-DDPG[۱۳] را برای دستیابی به کنترل بهینه با افزایش همکاری بین سیگنال های ترافیکی پیشنهاد کردند. روش پیشنهادی از طریق دو آزمایش به ترتیب با استفاده از مجموعه داده‌های مصنوعی و دنیای واقعی ارزیابی می‌شود. در این مطالعه، KS-DDPG ، یک چارچوب یادگیری تقویتی چند عامله جدید برای بهینه سازی گسترده شبکه معرفی شد. یک مکانیسم اشتراک دانش به عنوان پروتکل ارتباطی درون عاملی برای بهبود مهارت‌های هماهنگی بین عوامل استفاده می شود (Kumar et al., 2021). الگوریتم پیشنهادی با هر دو روش حمل و نقل مرسوم و مبتنی بر یادگیری تقویتی با ارزیابی دقیق در سناریوهای مصنوعی و دنیای واقعی مقایسه می‌شود. مقایسه با روش‌های حمل و نقل مبتنی بر یادگیری تقویتی پیشرفته و روش‌های حمل و نقل مرسوم نشان می دهد که KS-DDPF پیشنهادی، کارایی قابل توجهی در کنترل شبکه های حمل و نقل در مقیاس بزرگ و مقابله با نوسانات در جریان ترافیک دارد (Farazi et al., 2021). علاوه براین، مکانیسم ارتباطی معرفی شده نیز ثابت شده است که همگرایی مدل را بدون افزایش قابل توجهی در بار محاسباتی، سرعت می‌بخشد. با بررسی‌های اولیه صورت گرفته، مقالات در زمینه‌های مختلفی از جمله بهینه سازی زمانبندی چراغ راهنمایی، هماهنگ‌سازی چراغ های راهنمایی با یکدیگر و بطور کلی تسهیل عبور و مرور در شرایط اشباع و بحرانی، در راستای بهینه سازی زمانبندی چراغ راهنمایی و کاهش حجم ترافیک و تلاش در جهت روانسازی جریان ترافیک و همچنین در موارد مورد نیاز، انتخاب تابع هدف مناسب برای انجام این اقدامات، مطالعاتی انجام شده است و باید سعی شود خلا این اقدامات از جمله مقایسه دو الگوریتم از الگوریتم‌های یادگیری تقویتی که در کار حاضر به آن پرداخته شده است و باید سعی شود با اصلاحات بهتر در تحقیقات پیش رو با بکارگیری روش‌های به روز و کارآمد، به صورت مناسب جبران گردد.

۳- روش شناسی پژوهش

در این مطالعه یک شبکه شهری فرض شده در نظر گرفته می‌شود و از نظر جغرافیایی اهمیت چندانی نداشت که این شبیه‌سازی و بهینه‌سازی برای چه منطقه‌ای از شهرهای دنیا صورت بپذیرد، مهم منظم بودن فواصل چراغ‌های راهنمایی در نزدیکی یکدیگر و همچنین مرتب و منظم بودن شبکه بود که بتوان شرایط اشباع را برای آن شبکه اعمال کرد و با دادن حجم معینی از وسایل نقلیه و رساندن شرایط به حالت اشباع و همچنین وارد کردن ورودی‌های مورد نظر و بدست آوردن خروجی از الگوریتم مورد نظر از طریق روش یادگیری تقویتی و مقایسه با نتایج شبیه‌سازی صورت گرفته، بهینه‌شدن زمان چرخه و کاهش طول صف در تقاطع‌های مورد نظر بررسی و مشاهده شود. در این تحقیق قصد بر آن است با استفاده از مدل‌ها و روش‌های یادگیری تقویتی و استفاده از دو نوع الگوریتم و مقایسه آنها با یکدیگر، طول صف تشکیل شده در پشت هر چراغ راهنمایی کاهش داده شود و زمانبندی هر چراغ راهنمایی از طریق کاهش طول صف که به آن اشاره شد و همچنین علاوه بر آن، میزان زمان توقف و معطل شدن وسایل‌نقلیه در پشت چراغ راهنمایی بهبود داده شود تا شبکه‌ای کارآمد با جریان روان و بهینه در دسترس باشد. برای تحقق این امر از نرم‌افزار شبیه ساز ترافیکی خردنگر سومو[۱۴] استفاده شد که در ادامه با کارایی های مفید این نرم افزار و همچنین مراحلی که

برای مدل سازی طی شد تا شبکه برای ایجاد الگوریتم یادگیری تقویتی موردنظر و اجرا روی آن، آماده شود. ابتدا داده‌های فرضی، به دو مدل DQN و مدل شبکه عصبی DDPG داده شد و این مدل‌ها وظیفه تعلیم در شبکه[۱۵] و بهبود آن را برعهده گرفتند. در ادامه این بار ترافیکی به مدل آموزش داده شده یادگیری تقویتی[۱۶] داده شد و این مدل عمل‌های بهینه را براساس خروجی‌های شبیه‌سازی انتخاب خواهد نمود. نتایج عملکرد در ادامه قابل مشاهده خواهد بود که باعث بهبود در هر مرحله از شبکه به صورت کلی و بهبود دو پارامتر زمان توقف در هر گام[۱۷] و میانگین زمان معطلی وسایل‌نقلیه[۱۸] به صورت محسوس در هر گام[۱۹] شد. که روند انجام کار به صورت دقیق‌تر در ادامه قابل مشاهده می‌باشد.

۳-۱- نرم افزارهای مورد استفاده

۳-۱-۱- نرم افزار سومو (SUMO)

به منظور ارزیابی سیستم‌های پیشنهادی سه روش وجود دارد. **روش اول**، مدل‌سازی فیزیکی است که یک نمونه واقعی از سیستم در مقیاس کوچکتر یا بزرگتر ساخته می‌شود، استفاده از این روش در سیستم‌های ترافیکی بسیار مشکل و غیرممکن می‌باشد. **روش دوم**، ترجمه سیستم پیشنهادی به زبان ریاضی می‌باشد. از معایب این روش می‌توان به پیچیدگی بالای آن در سیستم‌هایی با جزئیات زیاد اشاره کرد. **روش سوم**، استفاده از مدل شبیه‌سازی کامپیوتری می‌باشد که اجازه می‌دهد، مدیران در شبکه حمل و نقل ترافیک شهری، هر راه حل قابل قبولی را قبل از پیاده‌سازی، با پارامترهای متعدد ارزیابی کنند. شبیه‌ساز متحرک شهری سومو یکی از شبیه سازهای میکروسکوپیک است که به منظور مدل‌سازی، تحلیل و ارزیابی عملکرد و مدیریت شبکه‌های ترافیکی شهری استفاده می‌شود. SUMO یک مجموعه رایگان و متن باز شبیه‌سازی ترافیک است.

از سال ۲۰۰۱ در دسترس است و امکان مدل‌سازی سیستم‌های ترافیکی چند وجهی از جمله وسایل نقلیه جاده‌ای، حمل و نقل عمومی و عابران پیاده را فراهم می‌کند. همراه با SUMO مجموعه‌ای از ابزارهای پشتیبانی است که وظایف اصلی را برای ایجاد، اجرا و ارزیابی شبیه‌سازی‌های ترافیک، مانند واردات شبکه، محاسبات مسیر، تجسم و محاسبه انتشار آلودگی هوا می‌کند. اولین نسخه در سال ۲۰۰۱ در تعداد زیادی از پروژه‌های این موسسه مورد استفاده قرار گرفت. سپس در سال ۲۰۰۲ تکمیل و در دسترس برنامه نویسان و متخصصان در زمینه حمل و نقل قرار گرفت. در دو نسخه ۳۲ و ٦٤ بیتی بسته به نوع سیستم عامل عرضه می شود. لازم به ذکر است که حداکثر طول یک سناریوی شبیه‌سازی ۲۹۲ میلیون سال می‌باشد (https://sumo.dlr.de/docs/index.html , (۲۰۲۰)).

شکل ۱. صفحه اصلی سایت SUMO

۳-۱-۲- زبان برنامه نویسی پایتون (Python)

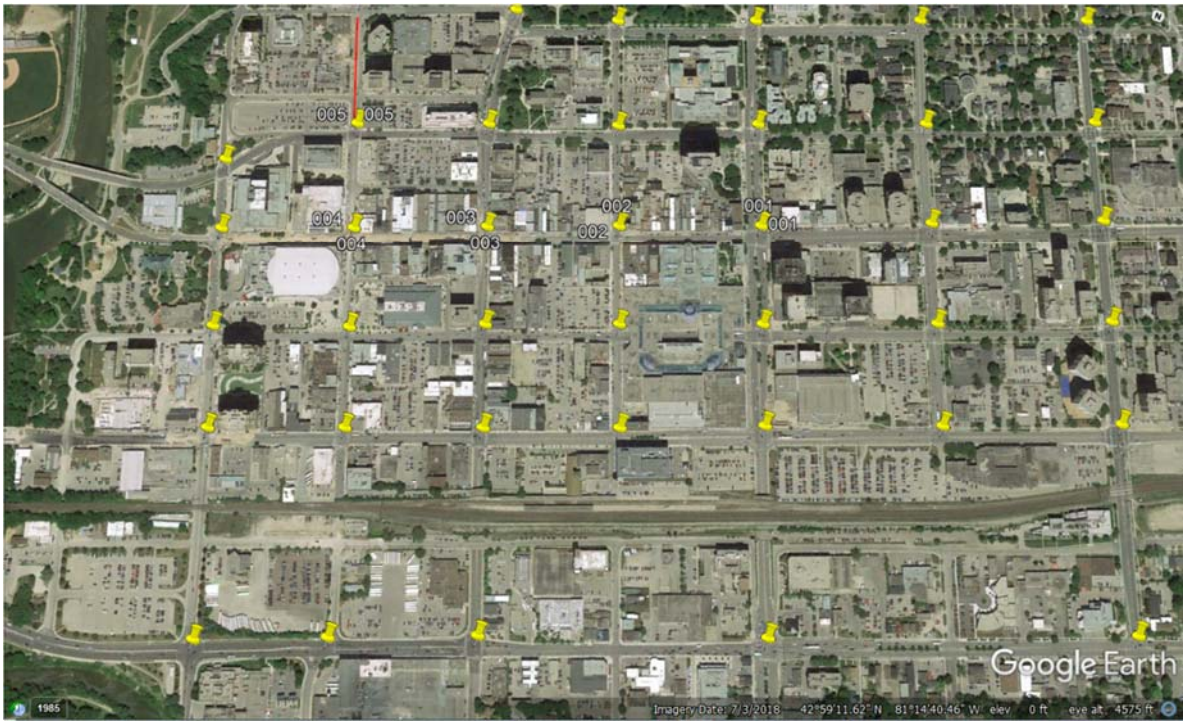
زبان برنامه‌نویسی پایتون، زبانی با یادگیری آسان محسوب می‌شود و از همین رو بسیاری از برنامه نویس‌های تازه کار آن را به عنوان اولین زبان برنامه‌نویسی خود بر می‌گزینند، زیرا پایتون به عنوان یک زبان همه منظوره ساخته و توسعه داده شده و محدود به توسعه نوع خاصی از نرم افزارها نیست. به بیان فنی، پایتون یک زبان برنامه نویسی شی گرا[۲۰] و سطح بالا با معناشناسی[۲۱] پویای یکپارچه شده برای وب و ساخت و توسعه نرم افزارهای کاربردی است. علاوه بر این، زبان برنامه نویسی پایتون از ماژول‌ها[۲۲] و بسته‌ها[۲۳] استفاده می‌کند، بدین معنا که برنامه‌های این زبان قابل طراحی به سبک ماژولار[۲۴] هستند و کدهای نوشته شده در یک پروژه در پروژه‌های گوناگون دیگر نیز قابل استفاده مجدد محسوب می‌شوند. یکی از قابل توجه‌ترین مزایای زبان برنامه نویسی پایتون آن است که کتابخانه و مفسر استاندارد[۲۵] آن، هم به صورت دودویی[۲۶] و هم منبع به رایگان در دسترس همگان قرار دارند. در پایتون هیچ انحصاری وجود ندارد، زیرا همه‌ی ابزارهای لازم برای آن در کلیه پلتفرم‌های اصلی موجود هستند.

۳-۲- منطقه مورد مطالعه

لندن شهری در جنوب غربی استان انتاریو[۲۷]، در کشور کانادا در قاره آمریکا است. این شهر بر روی دالان کبک سیتی – ویندزور[۲۸]، پرجمعیت‌ترین منطقه کانادا که با ۱۶ میلیون جمعیت نیمی از جمعیت کانادا را شامل می‌شود، واقع شده‌است. بر اساس آمار، تا سال ۲۰۰۱ جمعیت این ناحیه تقریباً هجده میلیون نفر بوده که ۵۱ درصد جمعیت کشور را در آن سال شامل می شده و سه منطقه از چهار منطقه شهری بزرگ کانادا را در برداشته است. لندن در بین ویندزور و تورنتو واقع شده است و در مرکز جنوب غربی انتاریو قرار دارد. لندن به عنوان ششمین شهر بزرگ در انتاریو و دهمین شهر بزرگ کانادا شناخته می‌شود. این شهر سریع‌ترین رشد جمعیت را در بین شهرهای کانادا داشته است. لندن بیش از ۴۰۰ هزار نفر جمعیت دارد و بیش از ۵۵۰ هزار نفر در کلان شهر ساکن هستند (https://adabvisa.com , (2021)).

شکل ۲. نمایی ازدالان کبک سیتی(Quebec) –ویندزور (Windsor) پرجمعیت‌ترین منطقه کانادا

شکل ۳. تصاویر شبکه شهری مورد مطالعه، نمای دور



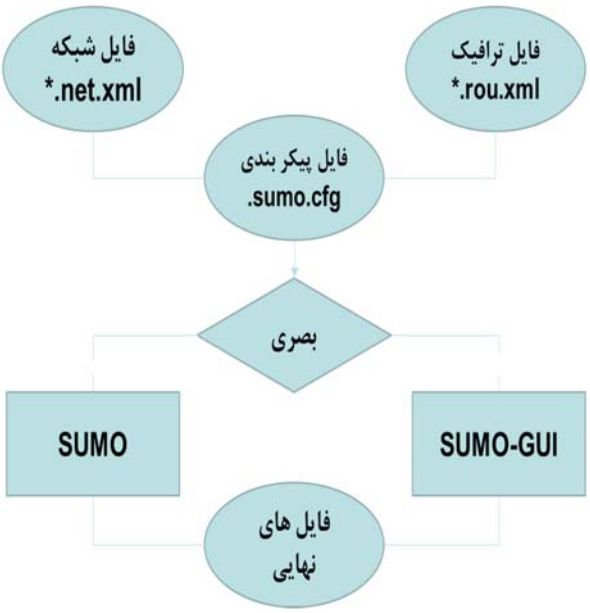
شکل ٤. تصاویر شبکه شهری مورد مطالعه، نمای نزدیک

۳-۳- آماده‌سازی داده‌ها در شبیه‌ساز SUMO

برای اینکه این نرم‌افزار بتواند سناریوی مورد نظر ما را اجرا کند باید فایل‌هایی را به عنوان ورودی دریافت کند. برای این منظور، باید یک شبکه‌ای از جاده که حاوی ترافیک می‌باشد را به عنوان ورودی به شبیه‌ساز بدهیم. این کار به طور معمول به صورت زیر صورت می‌گیرد.

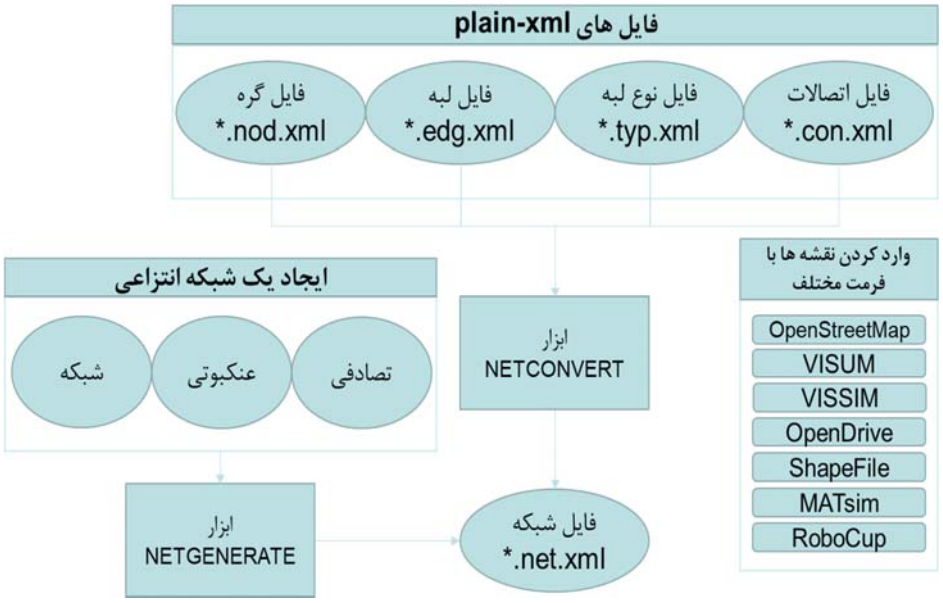
شکل ۵. نحوه تولید به فایل پیکربندی شبکه

بسته نرم‌افزاری SUMO به منظور ایجاد فایل شبکه جاده، دو ابزار به نام های Netconvert، Netgenerate و یک نرم‌افزار NETEDIT در پوشه bin قرار داده است.



شکل ٦. ورودی شبیه ساز sumo و sumo-gui

شکل (۷) یک دید کلی و اجمالی از فایل‌های ورودی مورد نیاز برای ابزارهای Netconvert و Netgenerate را نمایش می‌دهد.



شکل ۷. روش‌هایی برای تولید فایل شبکه

۳-٤- الگوریتم‌های مورد استفاده

۳-٤-۱- الگوریتم DQN

با فرض اینکه که پاداش مورد انتظار (فرض: جمع پاداش‌ها تا پایان) هر عمل را در هر مرحله بدانیم، این مانند یک برگه تقلب برای عامل خواهد بود، بطوریکه عامل[۲۹] دقیقاً می‌داند که کدام عمل را انجام دهد و دنباله‌ای از اقدام[۳۰]را انجام می‌دهد که در نهایت حداکثر پاداش کل را ایجاد کند. این پاداش کل نیز همانطور که پیشتر گفته شد، Q نامیده می‌شود که طبق معادله‌ی بلمن رابطه‌ی زیر درمورد آن صادق است:

$$Q(s.a) = r(s.a) + \gamma \max_a Q(s'.a) \qquad (۱)$$

همانطورکه متغیرهای رابطه بالا پیشتر ذکر گردید، در الگوریتم‌های Q-Learning به دنبال یادگیری تابع Q هستیم. در پیاده سازی این روش‌ها عامل، اقدام به سعی و خطا در محیط می‌کند و مقدار Q در تمام وضعیت‌ها و به ازای تمام اقدام‌ها مطابق فرمول بالا محاسبه شده و در یک جدول ذخیره می‌شود. نکته‌ی مهم این است که این استراتژی (ذخیره مقادیر Q بصورت جدولی دوبعدی از وضعیت و اقدام) تنها در محیط‌ها و مسائلی قابل انجام است که فضای وضعیت و فضای اقدام هر دو گسسته باشند. به عنوان مثال اگر محیط را بصورت یک جدول دو بعدی درنظر بگیریم، که مکان‌ها در این جدول بصورت سلول های مجزا در نظر گرفته شوند و یک عامل قصد حرکت در این جدول را داشته باشد، می‌توان از الگوریتم DQN برای حرکت دادن ایجنت استفاده کرد. اما اگر مکان‌ها بصورت سلول‌های مجزا تعریف نشوند بلکه موقعیت‌های x و y با مقادیر پیوسته باشند، این الگوریتم برای هدایت عامل، کارایی نخواهد داشت.

learningیک الگوریتم ساده و در عین حال بسیار قدرتمند جهت ایجاد یک برگه تقلب برای عامل است که در راستای کمک به انجام دقیق عمل مورد نظر، انجام می‌شود. اما اگر این برگه‌ی تقلب خیلی طولانی باشد استفاده از آن دشوار خواهد شد. محیطی با ۱۰۰۰۰ حالت و ۱۰۰۰ اقدام مثالی از یک محیط دشوار است که اگر بخواهیم مقادیر Q را در جدولی ذخیره کنیم، جدولی که از ۱۰ میلیون سلول ایجاد می شود و یادگیری تمام مقادیر آن در هر سلول بسیار دشوار خواهد بود. این دشواری از دو جهت حائز اهمیت است.

۱. اول، مقدار حافظه مورد نیاز برای ذخیره و به روزرسانی آن جدول با افزایش تعداد حالت‌ها افزایش می‌یابد.

۲. دوم، مقدار زمان مورد نیاز برای بررسی هر حالت برای ایجاد جدول Q مورد نیاز غیر واقعی خواهد بود.

در این شرایط راه‌حل، استفاده از روش‌های تقریب تابع و روش‌های یادگیری ماشین است. شبکه‌های عصبی رایج‌ترین

تکنیک برای تقریب زدن مقدار Q هستند و جایگزین جداول ذخیره‌ی Q می‌شوند. در صورتیکه در روش Q-Learning از یک شبکه‌ی عصبی که وضعیت سیستم را به عنوان ورودی بگیرد و برای تمام اقدام های ممکن مقدار ارزش (همان Q) اقدام را خروجی دهد، نام الگوریتم حاصل Deep Q-Learning است. بنابراین، مراحل یادگیری تقویتی با استفاده از شبکه‌های یادگیری عمیق Q (DQN) بصورت زیر می‌باشد.

-تمام تجربیات گذشته توسط ایجنت در حافظه ذخیره می‌شود.

-عمل بعدی با حداکثر خروجی شبکه Q تعیین می شود.

-تابع هزینه در اینجا میانگین مربعات خطای Q پیش بینی شده و Q هدف است. با این که مسئله یک مسئله‌ی رگرسیون است، ما در اینجا Q واقعی را نمی دانیم (اگر Q واقعی معلوم بود مسئله یک مسئله یادگیری نظارتی[۳۱] بود در حالیکه مسئله یادگیری تقویتی[۳۲] است).

با مرور معادله به روز رسانی Q حاصل از معادله بلمن داریم :

$$Q(S_t.A_t) \leftarrow Q(S_t.A_t) + a[R_{t+1} + \gamma \max_a Q(S_{t+1}.a) - Q(S_t.A_t)] \quad (۲)$$

بخش سبز رنگ نشان‌دهنده هدف است. می‌توان اینگونه استدلال کرد که شبکه مقدار خودش را در آینده پیش‌بینی می‌کند و سعی می‌کند مقادیر Q به گونه‌ای باشند که در معادله‌ی بلمن صدق کنند.

۳-٤-۲- الگوریتم DDPG

DDPG الگوریتمی است که همزمان یک تابع Q و یک سیاست را یاد می‌گیرد. برای یادگیری تابع Q از معادله بلمن و برای یادگیری سیاست از تابع Q استفاده می‌کند. این رویکرد ارتباط نزدیک و انگیزه‌ی یکسانی با Q-Learning دارد: اگر تابع ارزش عمل بهینه Q* را بدانید، در هر حالت معین، اقدام بهینه a* را با حل رابطه‌ی زیر می‌توان پیدا کرد.

$$a^*(s) = \arg\max_a Q^*(s.a) \quad (۳)$$

DDPG برخلاف Deep Q-Learning برای محیط‌های با اقدام پیوسته مناسب است. در این روش به دنبال یافتن یک تقریب کننده (مثلا شبکه‌ی عصبی) برای سیاست بهینه هستیم که از آن با عنوان Actor نیز یاد می‌شود. اما در پروسه‌ی یادگیری، نیاز به داشتن تعدادی ورودی و جواب‌های متناظر صحیح برای هرکدام از ورودی‌ها وجود دارد. برای این منظور از مقدار Q در هر حالت و هر اقدامی که شبکه‌ی Actor اتخاذ می‌کند استفاده می‌کنیم. از آنجا که وضعیت‌ها و اقدام‌ها پیوسته هستند و می‌توانند هر مقداری را دارا باشند، نمی‌توان مانند الگوریتم‌های رایج، از جداول برای ذخیره‌ی مقادیر Q استفاده نمود. از این رو باید برای تخمین Q نیز از یک تقریب‌گر (شبکه‌ی عصبی) استفاده نمود که به آن شبکه‌ی Critic گفته می‌شود. بر این اساس تابع هزینه‌ای که در حین یادگیری شبکه‌ی Critic استفاده می‌شود مبتنی بر معادله‌ی بلمن است و با نام MSBE[۳۳] شناخته می‌شود و بصورت زیر است:

$$L(\phi.D) = E_{(s.a.r.s'.d)\sim D}[(Q_\phi(s.a) - (r + \gamma(1-d)\max_{a'} Q_\phi(s'.a')))^2] \quad (٤)$$

یادگیری شبکه‌ی Actor در DDPG نسبتاً ساده است.

ما می‌خواهیم یک سیاست قطعی[۳٤] یاد بگیریم که عملی را ارائه می‌دهد که Q را به حداکثر می‌رساند. از آنجایی که فضای عمل پیوسته است، و ما فرض می‌کنیم که تابع Q با توجه به عملکرد قابل تمایز است، فقط می‌توانیم الگوریتم صعود شیب[۳۵] (فقط نسبت به پارامترهای Actor) را برای حل انجام دهیم.

$$\max_\theta E_{s\sim D}[Q_\phi(s.\mu_\theta(s))] \quad (۵)$$

همه الگوریتم‌های استاندارد برای آموزش یک شبکه عصبی عمیق به روش DDPG از چیزی به نام بافر بازپخش[۳٦] استفاده می‌کنند. این مجموعه شامل تجربیات قبلی است که در طول سعی و خطاها ذخیره شده‌اند. این تجربیات بصورت یک مجموعه از وضعیت(s)، تصمیم شبکه (a)، وضعیت جدید ناشی از تصمیم شبکه (s') و جایزه‌ی دریافتی به خاطر این تغییر وضعیت (r) ذخیره می‌شوند و بافر باید شامل تعداد زیادی از این مجموعه‌ها باشد. برای اینکه الگوریتم رفتار پایداری داشته باشد، بافر بازپخش باید به اندازه کافی بزرگ باشد که طیف وسیعی از تجربیات را در خود داشته باشد، اما ممکن است همیشه حفظ همه چیز خوب نباشد. اگر فقط از جدیدترین داده‌ها استفاده شود، شبکه بیش از حد به آن تطبیق داده می‌شود. اگر بیش از حد از تجربه استفاده کنید، ممکن است یادگیری خود را کند کنید.

الگوریتم‌های DDPG از شبکه‌های هدف[۳۷] استفاده می‌کنند. عبارت زیر (رابطه(٦)) هدف نامیده می‌شود، زیرا زمانی که ما تابع هزینه‌ی MSBE را به حداقل می‌رسانیم، سعی می‌کنیم تابع Q را بیشتر شبیه به این هدف کنیم.

$$\phi_{targ} \leftarrow \rho\phi_{targ} + (1-\rho)\phi \quad (٦)$$

اما مشکل در اینجاست که هدف بستگی به پارامترهای همان شبکه‌ای دارد که ما می‌خواهیم آموزش دهیم. این باعث می‌شود که حداقل‌سازی MSBE ناپایدار باشد. راه حل این است که از مجموعه‌ای از پارامترها استفاده شود که نزدیک به پارامترهای شبکه‌ی ما هستند، اما با تاخیر زمانی، یعنی شبکه دوم به نام شبکه هدف، که از اولی عقب‌تر است به کار برده شود. در الگوریتم‌های DDPG، شبکه هدف یک بار در هر به روزرسانی شبکه اصلی با میانگین‌گیری (همانند رابطه (٦) که در بالا به آن اشاره شد) مطابق زیر به روز می‌شود.

$$\phi_{targ} \leftarrow \rho\phi_{targ} + (1-\rho)\phi$$

که در آن ρ یک هایپرپارامتر بین ۰ و ۱ است (معمولاً نزدیک به ۱).

۳-۵- گام‌ها و اپیزودها در کاربرد کنترل ترافیک

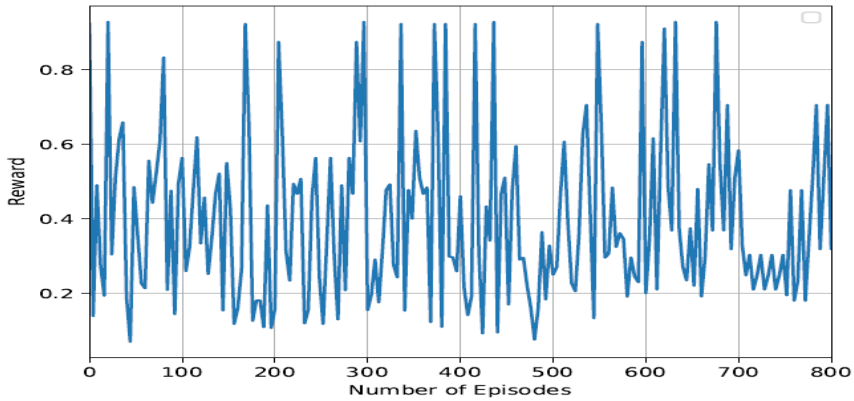
یادگیری تقویتی در اینجا طی گام‌های زمانی ثابت ۲۰ ثانیه‌ای صورت می‌گیرد. بدین صورت که تصمیم عامل به چراغ‌های شهر اعمال می‌شود و سپس ۲۰ ثانیه شبیه‌سازی بدون مداخله ادامه می‌یابد تا تاثیر تصمیمات برروی وضعیت ترافیک مشخص شود. سپس وضعیت سیستم (طول صف‌های پشت چراغ‌ها) ضبط شده و دوباره به شبکه و الگوریتم یادگیری سپرده می‌شوند تا برای تصمیم‌گیری در مورد گام بعدی مورد استفاده قرار گیرند. در صورتیکه عملکرد عامل باعث قفل شدن شهر شود (ترافیک بصورت غیرقابل حلی زیاد شود و عملاً شهر به حالت اشباع یا حتی فوق اشباع برسد)، نیاز است که شبیه‌سازی ریست شده و وضعیت صف‌ها به حالت اولیه برگردد. معیار قفل شدن ترافیک، نسبت کل خودروهای ساکن به خودروهای موجود در شهر است. هرگاه این نسبت از ۰/۹ فراتر رود، ترافیک غیرقابل حل خواهد بود و اصطلاحاً یک اپیزود از آموزش به پایان می‌رسد. بنابراین هر اپیزود به بازه‌ی شروع شبیه سازی تا ساکن شدن ۹۰ درصد خودرو اطلاق می‌شود.

٤- مدل‌ها و عملکرد و نتایج الگوریتم‌ها

٤-١ پیاده سازی DQN برای کنترل ترافیک

الگوریتم DQN مشابه توضیحات بخش قبل، پیاده‌سازی شده است. برای پیاده سازی این روش از کتابخانه و ماژول pytorch استفاده شده است. پیاده سازی الگوریتم مشابهت زیادی به پیاده سازی در مسائل رباتیکی مانند دارد. شبکه‌ی استفاده شده از چهار لایه تشکیل شده است. لایه‌ی ورودی دارای ابعاد ٦١٩ است که هم تعداد با طول صف‌های موجود در شهر است. در لایه‌های دوم و سوم و خروجی ابعاد ویژگی‌ها به ترتیب به ١٢٨، ٦٤ و در نهایت ٢ (تعداد اکشن‌های ممکن که برابر با حالت سبز یا قرمز برای شلوغ ترین چراغ شهر است) کاهش می‌یابد و خروجی شبکه نیز مقدار Q برای هرکدام از اکشن‌ها است. در لایه‌های میانی از تابع فعال‌سازی relu استفاده شده است و در لایه‌ی آخر نیز تابع softmax برای تبدیل خروجی به توزیع احتمال بکار رفته است. تابع پاداش[۳۸] در اینجا نیز تابع differential است که پیشتر معرفی شد. آموزش شبکه‌ی مذکور برای ۸۰۰ اپیزود است که نتیجه آن مطابق شکل (۸) که در زیر قابل مشاهده می‌باشد را دارد.

در اینجا میزان پاداش دریافت شده روند افزایشی به خود نگرفته است و از این رو نمی‌توان نتیجه گرفت که روش DQN برای کنترل یک چراغ سودمند بوده است. از آنجا که روش DQN تنها برای حالتی کاربرد دارد که اکشن‌ها گسسته باشند و تنها مقدار Q را برای هر کدام از اکشن‌های گسسته پیش‌بینی می‌کند، این روش در حالتی که تعداد اکشن‌های ممکن زیاد باشد کارایی ندارد. اگر بخواهیم تمام چراغ‌های شهر را با این روش کنترل کنیم، تعداد اکشن‌های ممکن برابر با 2^n خواهد بود که برای تمام چراغ‌های شهر عددی بسیار زیاد خواهد شد.

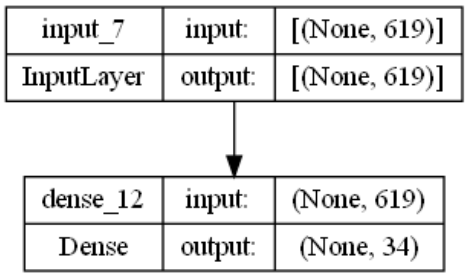


شکل ۸ آموزش شبکه DQN برای ۸۰۰ اپیزود و مقایسه پاداش‌های دریافتی در هر اپیزود

٤-٢- پیاده سازی DDPG برای کنترل ترافیک

الگوریتم DDPG مطابق با توضیحات فوق برای آموزش شبکه‌های عصبی Actor و Critic بکار رفته است. این شبکه‌های عصبی توسط کتابخانه و ماژول‌های keras در پایتون مدلسازی شده‌اند. در اینجا از دو کانفیگ برای شبکه‌های Actor و Critic استفاده شده است. در ساده‌ترین حالت از شبکه‌ای تک لایه برای Actor و یک شبکه با دو شاخه ورودی تک لایه برای Critic استفاده شده است.

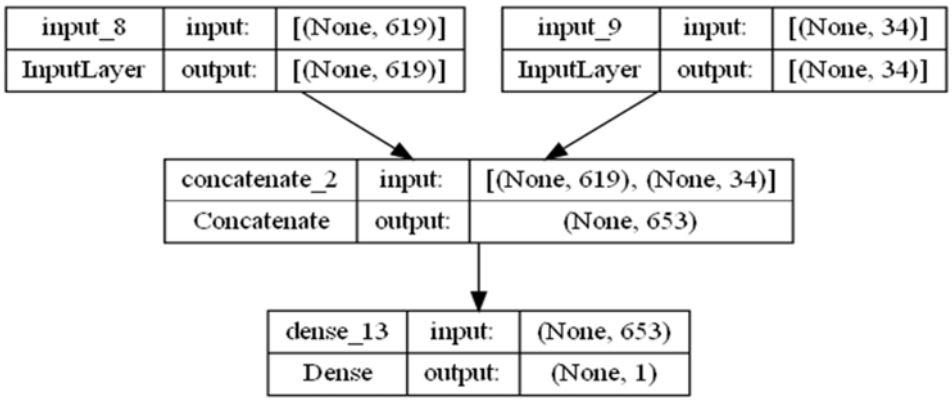
به صورت شهودی این کانفیگ ساده باید بتواند با محاسبه‌ی یک ترکیب خطی از تمام صف های موجود در شهر، برای هر چراغ تصمیم‌گیری کند. تصویر زیر ساختار شبکه‌ی Actor تک لایه بکار رفته را نشان می‌دهد. همانطور که مشهود است، در لایه‌ی اول ٦١٩ عدد ورودی توسط شبکه دریافت می‌شود که هم تعداد با تمام لاین‌های موجود در خیابان های شهر است. هر کدام از این اعداد تعداد خودروهای موجود در لاین مربوطه می‌باشد. این عدد همچنین برابر با تعداد وضعیت‌های مسئله‌ی یادگیری تقویتی (تعداد ابعاد s) نیز هست. در لایه‌ی دوم نیز تعداد ٣٤ عدد به عنوان خروجی محاسبه می‌شوند که هم تعداد با چراغ‌های موجود در شهر است. در این لایه از تابع فعال‌سازی Sigmoid استفاده شده است که تمام اعداد خروجی را به بازه‌ی [0,1] منتقل می‌کند. هر عدد خروجی به تعداد فازهای تعریف شده برای چراغ مربوطه نگاشت می‌شود. به عنوان مثال برای یک چراغ راهنمایی با 4 فاز، عدد خروجی شبکه در ٤ ضرب شده و به عددی صحیح در بازه‌ی [0,4] تبدیل می‌شود که اندیس یکی از فازهای چراغ مربوطه است.



شکل ۹. ساختار شبکه Actor تک لایه

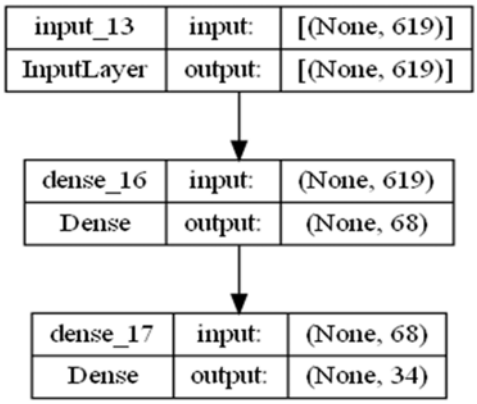
ساختار شبکه‌ی Critic در ساده‌ترین حالت نیز به صورت زیر است. از آنجا که Critic هم وضعیت‌ها[۳۹] و هم اقدامات (خروجی‌های Actor) را به عنوان ورودی می‌گیرد، این شبکه دارای دو شاخه‌ی ورودی می‌باشد. شاخه‌ی سمت راست شاخه‌ی مربوط به اقدام‌ها (اکشن‌ها) است که تعداد ٣٤ عدد به آن وارد می‌شود.

شاخه‌ی سمت چپ نیز مربوط به وضعیت‌ها است که همانطور که پیشتر اشاره شد، ٦١٩ ورودی دارد. ورودی‌های این دو شاخه در یک لایه‌ی Concatenate به هم چسبانده شده و یک لایه با تعداد ٦٥٣ مقدار تشکیل داده‌اند. در نهایت این لایه به یک لایه‌ی خروجی متصل شده که تنها یک عدد (همان مقدار Q برای وضعیت و اکشن فعلی) را محاسبه می‌کند. این ساختار برای Actor و Critic با تغییر در تعداد لایه‌ها و تعداد ورودی و خروجی‌ها یکی از ساختارهای رایج در مثال‌های یادگیری تقویتی می‌باشد.

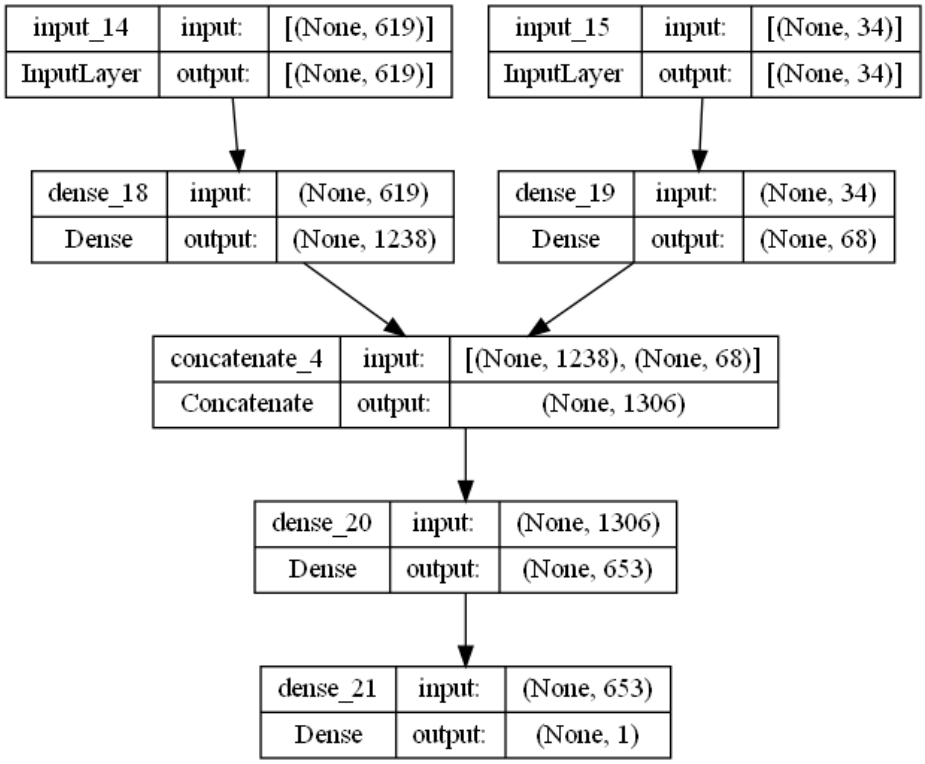


شکل ۱۰. ساختار شبکه Critic

در حالتی دیگر، به هرکدام از شبکه‌ها تعدادی لایه‌ی میانی با تابع فعال سازی Relu افزوده شده است. اینکار عموماً توانایی شبکه‌ها را در یادگیری مسئله‌های غیرخطی و پیچیده افزایش می‌دهد، اما گاها ممکن است دچار مشکل Overfittig شویم. ساختار شبکه‌های Actor و Critic در حالت دوم به ترتیب بصورت زیر است.



شکل ۱۱. ساختار شبکه Actor با افزودن لایه میانی با تابع فعال سازی Relu

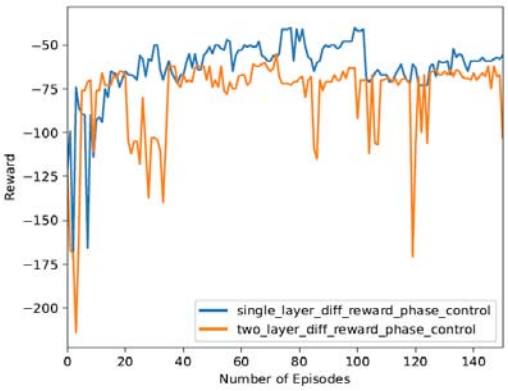


شکل ۱۲. ساختار شبکه Critic با افزودن لایه میانی با تابع فعال سازی Relu

٤-۳- عملکرد و نتایج الگوریتم DDPG

یکی از نتایج انجام یادگیری توسط دو ساختار تک لایه و چندلایه به صورت نمودار زیر مشهود است. در اینجا مقدار پاداش کسب شده توسط عامل در هر اپیزود نمایش داده شده است. در اپیزودهای ابتدایی مقدار پاداش کمتر بوده و پس از

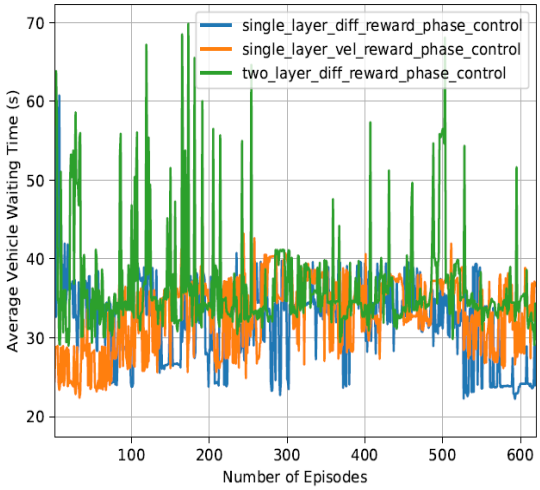
کمی یادگیری مقدار پاداش رو به افزایش است. در نمودار آبی رنگ از ساختار تک لایه استفاده شده است و در نمودار نارنجی از ساختار دو لایه. ساختار تک لایه در اینجا سریع‌تر می‌تواند به پاداش‌های بیشتر دست پیدا کند که یکی از دلایل آن کمتر بودن تعداد پارامترهای شبکه و راحت‌تر بودن یادگیری است. در درازمدت اما عملکرد دو شبکه به یکدیگر نزدیک‌تر شده است. پاداش استفاده شده در هر دو نمودار زیر از نوع differential است که به معنی تفاوت خودروهای به حرکت درآمده و متوقف شده است. منفی بودن پاداش تا انتها به معنی این است که با وجود بهبود عملکرد همچنان تعداد خودروهایی که متوقف می‌شود بیشتر از تعداد خودروهایی است که به حرکت در می‌آیند.



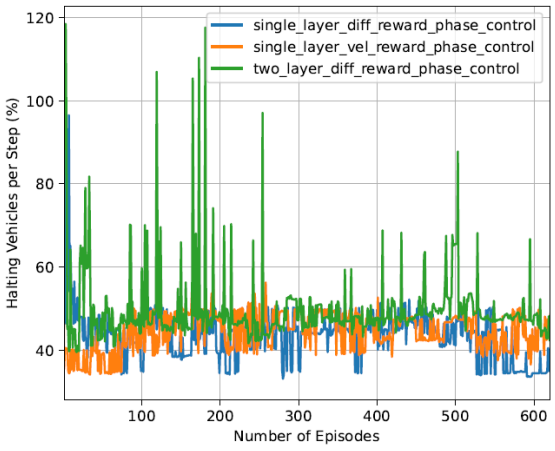
شکل ۱۳. انجام یادگیری در روش DDPG و نمایش مقدار پاداش کسب شده توسط عامل درهراپیزود

معیارهای دیگری برای بررسی عملکرد کنترل ترافیک نیز می‌تواند مورد بررسی قرار گرفته‌اند، که یکی از آن‌ها درصد خودروهای متوقف شده در شهر در هرگام[40] از یادگیری، در نمودار زیر برای دو ساختار تک لایه (که با رنگ آبی نمایش داده شده است) و چند لایه (که با رنگ سبزنمایش داده شده است) مقایسه شده است که در اینجا نیز همانطور که مشاهده می‌کنید ساختار تک لایه راحت تر توانسته است به بهبود عملکرد منجر شود. یکی از دلایل این عملکرد بهتر می‌تواند به بیشتر بودن پارامترهای شبکه‌ی دولایه مرتبط باشد و ممکن است پس از طی زمان بسیار بیشتر شبکه‌ی دولایه حتی بتواند به عملکرد بهتری نیز دست یابد اما با توجه به میزان آموزش[41] انجام شده در این پروژه این امر محقق نشده است.

دیگر معیار سنجش، متوسط زمان توقف خودروها است که در این زمینه نیز مشابه معیار قبلی شبکه‌ی تک لایه عملکرد بهتری از خود نشان داده است. نکته‌ی حائز اهمیت دیگر در این نتایج، بهبود نسبی تابع پاداش differential معرفی شده نسبت به پاداش بر اساس سرعت متوسط خودروها است. همانطور که در نمودارهای بالا مشهود است، پس از حدود ۶۰۰ اپیزود، روش differential معرفی شده موفق به بهبود زمان توقف خودروها و درصد خودروهای متوقف شده است در حالیکه روش سرعت متوسط (که با رنگ نارنجی نمایش داده شده است) با وجود عملکرد بهتر نسبت به شبکه‌ی دو لایه، از روش differential بهتر نبوده است. قابل ذکر است که تعداد لایه های بکار رفته در روش سرعت متوسط نیز برابر با یک است.



شکل ۱۴. مقایسه درصد خودروهای متوقف شده در شهر در هرگام از یادگیری برای دوساختارتک لایه و چند لایه



شکل ۱۵. مقایسه متوسط زمان توقف خودروها در شهر در هرگام از یادگیری برای دوساختارتک لایه و چند لایه

۵- نتیجه‌گیری

در این مطالعه با استفاده از روش‌های نوین و فراابتکاری از جمله یادگیری تقویتی و اعمال آن در شبکه شهری فرض شده، تلاش شده است که زمان سیکل چراغ راهنمایی در بهترین و بهینه‌ترین حالت خود قرار گرفته و در صورت امکان از انباشت صف و افزایش طول آن در پشت چراغ‌های راهنمایی کاسته شود و نتایج مطالعه نشان می‌دهد با انتخاب و استفاده این روش فرا ابتکاری، می‌توان انتظار بهبود در وضعیت تراکم ترافیک شبکه معابر را داشت. با بررسی‌های اولیه صورت گرفته، مقالات در زمینه‌های مختلفی از جمله بهینه سازی زمانبندی چراغ راهنمایی، هماهنگ‌سازی چراغ‌های راهنمایی با یکدیگر و بطورکلی تسهیل عبور و مرور در شرایط اشباع و بحرانی، در راستای بهینه سازی زمانبندی چراغ راهنمایی و کاهش حجم ترافیک و تلاش در جهت روانسازی جریان ترافیک و همچنین در موارد مورد نیاز، انتخاب تابع هدف مناسب برای انجام این اقدامات، مطالعاتی انجام شده است و باید سعی شود خلا این اقدامات در تحقیقات پیش رو با بکارگیری روش‌های به روز و کارآمد، به صورت مناسب جبران گردد. در این تحقیق منطقه‌ای از شهر لندن در کشور کانادا مورد مطالعه و شبیه‌سازی قرار گرفت که دارای شبکه‌ای منظم و شطرنجی و همچنین چراغ‌های راهنمایی نزدیک به یکدیگر است و با داده‌های فرضی، شرایط اشباع را برای مدت زمان معین که شامل یک ساعت اوج هم می‌شود، با استفاده از الگوریتم‌های یادگیری تقویتی DQN و DDPG، شبیه‌سازی کردیم. همانطور که اشاره شد، یادگیری تقویتی یک روش آموزش یادگیری ماشین است که بر اساس پاداش دادن به رفتارهای مطلوب و یا تنبیه رفتارهای نامطلوب است. به طور کلی، یک عامل یادگیری تقویتی قادر است محیط خود را درک و تفسیر کند، اقداماتی انجام دهد و از طریق آزمون و خطا یاد بگیرد. این استراتژی برای یادگیری نحوه‌ی تصمیم‌گیری، خصوصا در سیستم‌هایی که مدلسازی آنها دشوار یا هزینه بر است می‌تواند بسیار سودمند باشد. در ابتدای امر نقشه منطقه‌ای از شهر مورد نظر انتخاب شد و سپس به فرمتی قابل اجرا در نرم‌افزار سومو تبدیل شد. سپس دو نوع الگوریتم برای بهینه‌سازی از زیر مجموعه الگوریتم‌های یادگیری تقویتی به نام‌های DQN و DDPG مورد استفاده قرار گرفت. در ارتباط با الگوریتم DQN، یک الگوریتم ساده و در عین حال بسیار قدرتمند برای ایجاد یک برگه تقلب برای عامل است که کمک می‌کند تا دقیقاً کدام عمل انجام شود. اما اگر این برگه‌ی تقلب خیلی طولانی باشد، استفاده از آن دشوار می‌شود. این دشواری از دو جهت حائز اهمیت است: اول، مقدار حافظه مورد نیاز برای ذخیره و به‌روزرسانی آن جدول با افزایش تعداد حالت‌ها افزایش می‌یابد. دوم، مقدار زمان مورد نیاز برای بررسی هر حالت برای ایجاد جدول Q مورد نیاز غیر واقعی خواهد بود.

در این شرایط راه حل، استفاده از روش‌های تقریب تابع و روش‌های یادگیری ماشین است. شبکه‌های عصبی رایج‌ترین تکنیک برای تقریب زدن مقدار Q هستند و جایگزین جداول ذخیره‌ی Q می‌شوند. در صورتیکه در روش Q-Learning از یک شبکه‌ی عصبی که وضعیت سیستم را به عنوان ورودی بگیرد و برای تمام اکشن‌های ممکن مقدار ارزش (همان Q) اقدام را خروجی دهد، نام الگوریتم حاصل Deep Q-Learning است که نتایج این الگوریتم برای شبکه، نتایج مطلوبی نبود که این امر موجب شد ما از الگوریتم DDPG برای شبکه استفاده کنیم.

DDPG برخلاف DQN برای محیط‌های با اقدام پیوسته مناسب است. در این روش به دنبال یافتن یک تقریب کننده (مثلاً شبکه‌ی عصبی) برای سیاست بهینه هستیم که از آن با عنوان Actor نیز یاد می‌شود. اما در پروسه‌ی یادگیری، نیاز به داشتن تعدادی ورودی و جواب‌های متناظر صحیح برای هرکدام از ورودی‌ها وجود دارد. برای این منظور از مقدار Q در هر حالت و هر اقدامی که شبکه‌ی Actor اتخاذ می‌کند استفاده می‌کنیم. از آنجا که وضعیت‌ها و اقدام‌ها (اکشن‌ها) پیوسته هستند و می‌توانند هر مقداری را دارا باشند، نمی‌توان مانند الگوریتم‌های رایج، از جداول برای ذخیره‌ی مقادیر Q استفاده نمود. از اینرو باید برای تخمین Q نیز از یک تقریب‌گر (شبکه‌ی عصبی) استفاده نمود که به آن شبکه‌ی Critic گفته می‌شود. در دو حالت این الگوریتم را بررسی کردیم، تک لایه و دو لایه. همانطور که در بخش چهارم اشاره شد، در حالت اول شبکه دارای دو شاخه ورودی می‌باشد که شاخه سمت راست مربوط به اقدام‌ها (اکشن‌ها) و شاخه سمت چپ مربوط به وضعیت‌ها است که ورودی‌های این دو شاخه در یک لایه با نام Concatenate به هم چسبانده شده و یک لایه واحد را تشکیل می‌دهند که در نهایت این لایه به یک لایه خروجی متصل شده که تنها یک عدد که همان Q برای وضعیت و اقدام‌ها (اکشن‌ها) می‌باشد را محاسبه می‌کند. در حالتی دیگر، به هرکدام از شبکه‌ها، تعدادی لایه‌ی میانی با تابع فعال سازی Relu افزوده شده است، که اینکار عموما توانایی شبکه‌ها را در یادگیری مسئله‌های غیرخطی و پیچیده افزایش می‌دهد.

در نمودار مربوط به میزان پاداش دریافتی، ساختار تک لایه سریع‌تر می‌تواند به پاداش‌های بیشتر دست پیدا کند که یکی از دلایل آن همانطور که اشاره شد کمتر بودن تعداد پارامترهای شبکه و راحت تر بودن یادگیری است. در دراز مدت اما عملکرد دو شبکه به یکدیگر نزدیکتر شده است. معیارهای دیگری برای بررسی عملکرد کنترل ترافیک نیز مورد بررسی قرار گرفتند، که یکی از آن ها درصد خودروهای متوقف شده در شهر در هرگام از یادگیری و دیگری متوسط زمان توقف خودروها است، در این بخش نیز همانطور که مشاهده شد، ساختار تک لایه راحت‌تر توانسته است به بهبود عملکرد منجر شود. یکی از

دلایل این عملکرد بهتر می‌تواند به بیشتر بودن پارامترهای شبکه‌ی دولایه مرتبط باشد و ممکن است پس از طی زمان بسیار بیشتر شبکه‌ی دولایه حتی بتواند به عملکرد بهتری نیز دست یابد اما با توجه به میزان آموزش انجام شده در این پروژه این امر محقق نشده است. در تحقیق حاضر دو روش یادگیری تقویتی عمیق برای بهبود کنترل ترافیک بر اساس طول صف خودروها در خیابان‌های منتهی به چراغ‌های راهنمایی و رانندگی پیشنهاد و پیاده‌سازی شده‌اند. نخست روش DQN برای کنترل یک چراغ راهنمایی که بیشترین ترافیک در آن مشاهده شده است، که براساس بازخورد گرفتن از طول صفوف کل شهر بررسی شد. با توجه به تعداد اپیزودهای آموزش داده شده در اینجا، همگرایی به عملکرد بهتر ترافیکی حاصل نشد و این نشانگر عدم قابلیت ساده‌سازی مسئله و تقلیل تعداد چراغ‌های مورد نیاز برای کنترل به یک عدد (شلوغ‌ترین چراغ) می‌باشد. بر همین اساس روش دیگری به نام روش DDPG که قابلیت کنترل همزمان چندین چراغ را دارد به کار گرفته شده است. نتایج پیاده‌سازی این روش نشان می‌دهد که با استفاده از DDPG معیارهای ترافیکی مانند میانگین زمان ایستادن خودروها و درصد خودروها ساکن در کل شهر کاهش ملموسی پیدا می‌کنند. در این پروژه، شبکه‌های عصبی با تعداد لایه‌های محدود (نهایتاً دو لایه برای شبکه‌ی actor) مورد استفاده قرار گرفت که دلیل عمده‌ی آن محدودیت‌های سخت افزاری بوده است. افزایش تعداد لایه‌های شبکه‌ها به درک پیچیدگی سیستم کمک می‌کند اما از طرف دیگر باعث کند شدن پروسه‌ی آموزش خواهد شد که در کاربردهای زمان بر مانند یادگیری تقویتی غیرقابل پیاده سازی است. استفاده از واحدهای پردازش گرافیکی جهت آموزش شبکه‌های با تعداد لایه‌های بیشتر برای کاربرد حاضر پیشنهاد می‌شود و به احتمال زیاد باعث بهبود عملکرد خواهد شد. دیگر پیشنهاد موجود جهت استفاده حداکثری از ظرفیت چراغ‌های راهنمایی، افزایش تعداد فازهای چراغ در هر چهار راه است. در تحقیق حاضر تعداد فازهای تعریف شده حداکثر ٤ فاز بوده است در حالیکه تعداد بیشتر فازها، می‌تواند امکان تنوع تصمیم‌گیری توسط شبکه را فراهم آورد. لازم به ذکر است که این کار به دلیل افزایش تعداد خروجی‌های شبکه از نظر زمان آموزش یا نیاز به افزایش لایه‌های شبکه‌ی عصبی چالش برانگیز خواهد بود. در حالت ایده آل بهتر است که تک تک لامپ‌های مربوط به هر چراغ (هر لامپ یک کنترل کننده‌ی یک لینک[42] است و هر لینک واسط یک لین[43] ورودی و خروجی) کنترل شوند تا دست الگوریتم یادگیری تقویتی برای استفاده از انواع ابتکارات جهت باز کردن گره‌های ترافیکی شهر، باز باشد. البته که استفاده از قیود مشخصی برای هدایت شبکه به جواب درست می‌تواند پروسه‌ی یادگیری در چنین شرایطی را آسان‌تر کند. همچنین توصیه می‌شود با استفاده از نرم افزارهای شبیه ساز دیگر نیز این کد اجرا و نتیجه مقایسه شود. همچنین با دیگر الگوریتم‌های زیر مجموعه یادگیری تقویتی نیز مشابه این تحقیق، کد مورد نظر نوشته و با نتایج این تحقیق مقایسه شود تا در روند پردازش یا تسریع عمل مورد نظر نتایج بهینه‌تری به نسبت مشاهده شود و مشابه همین کار با سایر الگوریتم‌ها در گام‌ها و اپیزودهای طولانی‌تر اجرا و نتایج در دراز مدت نیز مقایسه شوند.

٦- پی‌نوشت‌ها

1. ITS
2. Connected-automated vehicles
3. Connected human-driven vehicles
4. Hamilton–Jacobi–Bellman
5. Model Predictive Controller
6. Back-pressure policy
7. Control loop
8. Adaptive Traffic Signal Control
9. Multi Agent Reinforcement Learning
10. Q learning agents
11. Deep RL
12. Advantage Actor Critic
13. گرادیان سیاست قطعی عمیق اشتراک‌گذاری دانش
14. SUMO: Simulation Of Urban Mobility
15. Train
16. RL
17. Halting Vehicles Per Step
18. Average Vehicle Waiting Time
19. Episode
20. Object-Oriented
21. Semantic
22. Modules
23. Packages
24. Modular
25. Standard Interpreter
26. Binary
27. Ontario
28. Windsor – Quebec
29. Agent
30. Action
31. Supervised Learning
32. Reinforcement Learning
33. Mean squared Bellman Error
34. Deterministic
35. Gradient Ascent
36. Replay Buffer
37. Target Networks
38. Reward
39. States
40. Halting Vehicles Per Step
41. Train
42. Link
43. Lane

۷- مراجع

-Abdi, A., Mosadeq, Z., & Bigdeli Rad, H. (2020). Prioritizing Factors Affecting Road Safety Using Fuzzy Hierarchical Analysis. *Journal of Transportation Research*, 17(3), 33-44.

-Afandizadeh Zargari, S., Bigdeli Rad, H., & Shaker, H. (2019). Using optimization and metaheuristic method to reduce the bus headway (Case study: Qazvin Bus Routes). *Quarterly Journal of Transportation Engineering*, 10(4), 833-849.

-Afandizadeh, S., & Bigdeli Rad, H. (2021). Developing a model to determine the number of vehicles lane changing on freeways by Brownian motion method. *Nonlinear Engineering*, 10(1), 450-460.

-Afandizadeh, S., Aziz Jalali, D., & Bigdeli Rad, H. (2023). Optimal routing for shared autonomous vehicles feeder services in urban networks. *Journal of Transportation Research.*

-Ameri, A., Bigdeli Rad, H., Shaker, H., & Ameri, M. (2021). Cellular Transmission and Optimization Model Development to Determine the Distances between Variable Message Signs. *Journal of Transportation Infrastructure Engineering*, 7(1), 1-16.

-Baldi, S, Michailidis, I, Ntampasi, V, Kosmatopoulos, E, Papamichail, I, & Papageorgiou, M. (2019). A simulation-based traffic signal control for congested urban traffic networks. *Transportation Science*, 53(1), 6-20 .

-Chu, T, Wang, J, Codecà, L, & Li, Z. (2019). Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 21(3), 1086-1095 .

-Eom, M., & Kim, B. I. (2020). The traffic signal control problem for intersections: a review. *European Transport Research Review*, 12, 1-20.

-Farazi, N. P., Zou, B., Ahamed, T., & Barua, L. (2021). Deep reinforcement learning in transportation research: A review. *Transportation research interdisciplinary perspectives*, 11, 100425.

-Guo, Q., Li, L., & Ban, X. J. (2019). Urban traffic signal control with connected and automated vehicles: A survey. *Transportation research part C: emerging technologies*, 101, 313-334.

-Hajisoleimani, M. M., Abdi, A., & Bigdeli Rad, H. (2021). Intermodal Non-Motorized Transportation Mode Choice; Case Study: Qazvin City. *Space Ontology International Journal*, 10(3), 31-46.

-Jafari, S, Shahbazi, Z, & Byun, Y.C. (2021). Improving the performance of single-intersection urban traffic networks based on a model predictive controller. *Sustainability*, 13(10), 5630.

-Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., & Michalewski, H. (2019). Model-based reinforcement learning for atari. arXiv preprint arXiv:1903.00374.

-Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., & Pérez, P. (2021). Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6), 4909-4926.

-Kumar, N., Mittal, S., Garg, V., & Kumar, N. (2021). Deep reinforcement learning-based traffic light scheduling framework for sdn-enabled smart transportation system. *IEEE Transactions on Intelligent Transportation Systems*, 23(3), 2411-2421.

-Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., & Srinivas, A. (2020). Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33, 19884-19895.

-Li, Z, Yu, H, Zhang, G, Dong, S, & Xu, C.Z. (2021). Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 125, 103059.

-Ma, D, Xiao, J, & Ma, X. (2021). A decentralized model predictive traffic signal control method with fixed phase sequence for urban networks. *Journal of Intelligent Transportation Systems*, 25(5), 455-468 .

-Michalopoulos, PG, & Stephanopoulos, G. (1977). Oversaturated signal systems with queue length constraints—I: Single intersection. *Transportation Research*, 11(6), 413-421 .

-Moerland, T. M., Broekens, J., Plaat, A., & Jonker, C. M. (2023). Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1), 1-118.

-Qadri, SSSM, Gökçe, MA, & Öner, E. (2020). State-of-art review of traffic signal control methods: challenges and opportunities. *European Transport Research Review*, 12, 1-23.

-Tajalli, M., Mehrabipour, M., & Hajbabaie, A. (2020). Network-level coordinated speed optimization and traffic light control for connected and automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(11), 6748-6759.

-Touhbi, S, Babram, MA, Nguyen-Huu, T, Marilleau, N, Hbid, ML, Cambier, C, & Stinckwich, S. (2017). Adaptive traffic signal control: Exploring reward definition for reinforcement learning. *Procedia Computer Science*, 109, 513-520 .

-Van Katwijk, RT. (2008). Multi-agent look-ahead traffic-adaptive control.

-https://adabvisa.com

-https://sumo.dlr.de/docs/index.html

Traffic Signal Timing in Saturated Mode Using Reinforcement Learning

Shahriar Afandizadeh, Professor, School of Civil Engineering, Iran University of Science and Technology, Tehran, Iran.
Mahmood Ahmadynejad, Professor, School of Civil Engineering, Iran University of Science and Technology, Tehran, Iran.
Alireza Movahedi, M.Sc., Student, School of Civil Engineering, Iran University of Science and Technology, Tehran, Iran.
Hamid Bigdeli Rad, Ph.D., Candidate, School of Civil Engineering, Iran University of Science and Technology, Tehran, Iran.

E-mail: zargari@iust.ac.ir

Received: January 2025- Accepted: April 2025

ABSTRACT

Today, With the expansion of urbanization, the need for a dynamic transportation system is felt more than ever. For this reason, to achieve a stable and orderly system, the control of transportation networks is considered essential. Although the modeling of networks today has become a complex and difficult issue and it faces problems in modeling to be closer to the environmental conditions, in the meantime, the framework of reinforcement learning as a model-independent method can play a better role in controlling and provide us with traffic simulation. In this study, we tried to use different reinforcement learning algorithms, such as DQN and DDPG algorithms, to simulate the considered traffic network in a faster and more regular way, and to be able to determine the influencing factors such as queue length. formed in the streets and traffic lights by using algorithms and proper planning, in a new way to reduce the amount of traffic and to optimize it, and according to the results obtained from the two mentioned algorithms, an algorithm that We propose that it had a better performance as the superior algorithm from the subset of reinforcement learning algorithm and finally our network by reducing the queue length and also reducing the amount of time spent behind traffic lights in urban networks in saturated state, which as a result improves passing and Review and smooth the flow of traffic.

Keywords: Urban Planning, Traffic Flow, Transportation Networks, Reinforcement Learning