DOI: 10.32347/2412-9933.2025.61.151-159
УДК 005.8:005.94

Левицький Володимир Володимирович
Аспірант кафедри інформаційних технологій,
https://orcid.org/0000-0003-1829-488X
Київський національний університет будівництва і архітектури, Київ

ОПТИМІЗАЦІЯ РУХУ ТРАНСПОРТУ В ПРОСТІЙ МЕРЕЖІ ЗА ДОПОМОГОЮ ГЛИБОКОГО НАВЧАННЯ З ПІДКРІПЛЕННЯМ

Анотація. *Оптимізація транспортного потоку в міських умовах залишається одним із ключових викликів сучасних досліджень, навіть попри значний обсяг наукових праць, присвячених цій темі. Незважаючи на досягнення, ця проблема все ще не має універсального рішення, яке б ефективно працювало в реальних сценаріях. Однією з основних складностей є опрацювання великого масиву вхідних даних, зокрема даних про дорожній рух, що постійно надходять із датчиків, встановлених по всій міській дорожній мережі. Традиційно, через масштабність завдання, дослідники зосереджувалися на розробці систем із локалізованими агентами. Такі агенти зазвичай управляють трафіком на окремих перехрестях, при цьому їхня координація здійснюється в рамках багатопотокових агентних систем. Однак в сучасних підходах враховується обсяг і складність вхідних даних завдяки застосуванню методів глибокого навчання. Зокрема, пропонується використання алгоритму глибокого детермінованого градієнта політики (DDPG), на основі якого можна обробляти великі вхідні масиви даних. У рамках експериментального дослідження була випробувана проста модель перехрестя, щоб перевірити ефективність підходу. Алгоритм DDPG засвідчив переваги в простій моделі порівняно з Q-learning. DDPG видавав нагороду 3-4 бали в діапазоні, тоді як нагорода Q-learning була в діапазоні 2-4 бали. Для оцінювання продуктивності підходу DDPG порівняно із Q-learning і випадковими таймінгами основним критерієм є середня винагорода за епізод. DDPG і Q-навчання досягають схожих рівнів винагороди, проте DDPG демонструє стабільну конвергенцію (0.04-0.21 бали), тоді як Q-learning залишається нестабільним (0.04-0.43 бали). Дослідження продуктивності внутрішнього епізоду засвідчує, що DDPG досягає покращень переважно ближче до кінця епізоду. Загалом цей алгоритм показав себе успішно для такого сценарію, а отримані результати можуть слугувати основою для подальших удосконалень і застосувань у складніших дорожніх сценаріях.*

Ключові слова: DDPG; Aimsun; Q-learning; дорожній рух

Вступ

Міста еволюціонували від планування, орієнтованого на пішохідну зону, до завантажених транспортних вузлів, які повинні мати, крім тротуарів і доріг для авто, трамвайні колії і місце для метро. Ця трансформація принесла виклики в управлінні дорожнім рухом, що привело до впровадження таких інструментів, як сигнали світлофора, ліхтарі та систематичне планування транзиту для ефективного регулювання міської мобільності.

Сучасні світлофори працюють у двох основних режимах: постійних програм і активованих систем. У фіксованих програмах або завчасно запланованих елементах керування використовується попередньо визначена тривалість для червоної, жовтої та зеленої фаз, незалежно від умов руху в реальному часі. Навпаки, активовані світлофори адаптують свої фази на основі локальних датчиків руху, розташованих поблизу перехресть. Хоча цей динамічний підхід покращує локалізоване управління дорожнім рухом, йому бракує координації з найближчими перехрестями, що робить його непридатним для густонаселених міських районів [1; 2].

Важливо те, що жодна система не використовує дані про транспортні потоки по всьому місту. Незважаючи на розгортання розгалужених мереж виявлення транспортних засобів, здатних прогнозувати затори, ця інформація часто обмежується звичайними реакціями, такими як залучення поліції для перенаправлення руху. Розширені стратегії для підвищення продуктивності світлофора з використанням таких даних залишаються недостатньо використаними.

Аналіз останніх досліджень і публікацій

Машинне навчання дає можливість вдосконалити процес контролю за дорожнім рухом, враховуючи оптимізацію часу сигналу на основі

умов дорожнього руху. У наведених нижче роботах досліджували можливості машинного навчання в контролі дорожнім рухом. У роботі [3] описано систему навчання з підкріпленням з використанням симулятора Green Light District, яка стала основою для досліджень в цьому напрямі. Інші підходи включають нечітку логіку та багатоагентні системи, де агенти, що контролюють окремі перехрестя, або обмінюються інформацією, або відповідають на спільні дані в налаштуваннях підключених транспортних засобів [4; 5]. Однак ці методи зазвичай стосуються ізольованих перехресть або невеликих мереж, покладаючись на часткові дані про трафік і не в змозі використати весь спектр доступної інформації.

Одним із методів вирішення проблеми може бути глибоке навчання з підкріпленням (ГНП). Такий вид алгоритму належить до алгоритмів навчання з підкріпленням, які використовують глибоку нейронну мережу як апроксиматор функції значення. Зростання їхньої популярності пов'язане з успіхом Deep Q-Networks (DQN) у грі в ігри Atari, використовуючи як вхідні дані необроблені пікселі гри [6].

У DQN є нейронна мережа, яка отримує стан середовища як вхідні дані і генерує як вихідні значення Q-значення для кожної з можливих дій, використовуючи функцію втрат, яка передбачає дотримання напрямку градієнта.

Першим успіхом навчання з підкріпленням за допомогою нейронних мереж як апроксимації функції був TD-Gammon [7]. Попри початковий ентузіазм у науковій спільноті, запропонований підхід виявився малоефективним при застосуванні до інших задач, що зумовило його подальше відхилення [8]. Основною причиною його невдачі була недостатня стабільність, яка випливає з нижче-наведених причин:

- Нейронна мережа була навчена зі значеннями, які були згенеровані на ходу, тому такі значення були послідовними за своєю природою, а отже, сильно корелювали зі значеннями в недалекому минулому (тобто не були незалежними та однаково розподіленими) [9].
- Коливання політики з невеликими змінами значень Q, які змінюють розподіл даних.
- Занадто великі кроки оптимізації, коли відбувається отримання великих винагород.

Щоб зменшити такі проблеми зі стабільністю, у роботах [10] автори застосували такі заходи.

- Відтворення досвіду: зберігати пам'ять про попередні дії та нагороди й тренувати нейронну мережу за допомогою випадкових вибірок з цього набору замість використання даних у реальному часі, тим самим усуваючи проблему часової автокореляції. Це можливо завдяки позаполітичній природі Q-навчання, яке дає змогу використовувати оновлення, що не обов'язково походять від політики, яка виконується.
- Зменшення винагороди: масштабувати та обмежувати значення винагород у діапазоні [−1, +1], щоб ваги не збільшувалися під час зворотного поширення помилки.
- Цільова мережа: використовувати окрему DQN, де одна мережа обчислює цільові значення, а інша накопичує оновлення ваг, які періодично завантажуються в першу. Це допомагає уникнути осциляцій у політиці через незначні зміни Q-значень.

Однак DQNs призначені для задач із обмеженою кількістю можливих дій, тому вони не підходять для безперервних просторів дій, як у випадку, де значною перешкодою є масштабованість алгоритмів керування світлофорами. З іншого боку, метод Deep Deterministic Policy Gradient (DDPG) [11] природно адаптується до такого типу задач. Як випливає з його назви, він поєднує класичний підхід актор-критик у навчанні з підкріпленням із детерміністичним градієнтом політики [12].

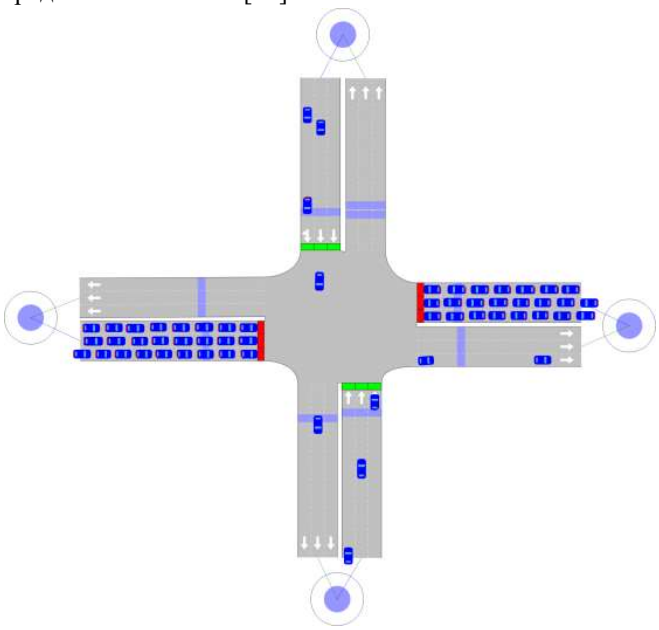

Рисунок 1 – Проста мережа трафіку в Aimsun

Оригінальне формулювання алгоритму градієнта політики було запропоноване в [10], де було доведено теорему градієнта політики для стохастичної політики.

Мета статті

Метою статті є експериментальна перевірка ефективності алгоритму DDPG для оптимізації управління транспортними потоками в міських умовах.

Основну увагу приділено оцінюванню продуктивності запропонованого підходу DDPG порівняно з Q-learning на базі симульованої моделі

перехрестя. Також має бути оцінена і проаналізована еволюція алгоритмів DDPG та Q-learning на базі симульованої моделі перехрестя в певний момент часу.

Виклад основного матеріалу

Концепції симуляції дорожнього руху

Щоб оцінити ефективність пропонованого підходу, використовується симулятор трафіку. Зокрема, вибрано Aimsun [13], стороннє програмне забезпечення мікроскопічного моделювання дорожнього руху, яке моделює сценарії дорожнього руху, включаючи дороги, перехрестя, світлофори тощо.

Основою будь-якої симуляції руху є мережа, що являє собою дороги та перехрестя, де рухаються транспортні засоби. До деяких доріг приєднані центроїди, які діють як джерела та поглиначі транспортних засобів. Кількість транспортних засобів, створених або поглинених центроїдами, регулюється матрицею попиту «початок-пункт», де кожна комірка представляє потік транспортних засобів між певним центроїдом відправлення та призначення. Щоб імітувати реальну динаміку трафіку, різні матриці попиту можна застосовувати в різні періоди часу під час моделювання.

Дороги в мережі часто включають детектори руху, які імітують індукційні петлі, вбудовані в землю. Ці детектори збирають ключові дані про дорожній рух, такі як кількість транспортних засобів, середня швидкість і відсоток заповненості.

Світлофор регулює рух на перехрестях. Ліхтарі на певному перехресті координуються, щоб запобігти блокуванню: коли один напрямок зелений, інший червоний. Така координація забезпечує впорядковане використання перехрестя. Стан усіх світлофорів на перехресті протягом певного періоду називається фазою, яка визначається станом і тривалістю вогнів. Послідовність фаз називається планом управління, який циклічно повторюється протягом часу. Сусідні перехрестя часто синхронізують свої плани керування, щоб максимізувати потік транспортних засобів і мінімізувати зупинки.

Після початку симуляції центроїди випускають транспортні засоби на основі матриці попиту, спрямовуючи їх до центроїдів призначення. Для досягнення реалістичних умов часто встановлюється період розігріву, протягом якого центроїди генерують транспортні засоби для заповнення мережі до офіційного початку симуляції.

Рис. 1 ілюструє просту транспортну мережу, змодельовану в Aimsun, що складається з одного перехрестя. Транспортні засоби зупиняються на червоне світло, рухаються через зелене світло, а центроїди випромінюють або поглинають транспортні засоби. Детектори дорожнього руху, розміщені на в'їздах і виїздах з доріг, збирають дані протягом моделювання.

DDPG та Q-learning

Більшість алгоритмів НП (навчання з підкріпленням) покладаються на рівняння Беллмана:

$$V^{\pi}(s) = R\big(s,\pi(s)\big) + \gamma \sum_{s'} P(s'|s,\pi(s))V^{\pi}(s'), \quad (1)$$

яке описує очікувану довгострокову винагороду (тобто значення V) за виконання дії, передбаченої деякою політикою π, коли в стані s відома його миттєва винагорода R(s, π(s)), γ – коефіцієнт дисконтування ($0 \le \gamma < 1$), що визначає, наскільки важливіші поточні винагороди порівняно з майбутніми [14]. $\sum_{s'} P(s'|s,\pi(s))V^{\pi}(s')$ – сума ймовірностей переходу в стан s' з поточного стану s після виконання дії $\pi(s)$. $V^{\pi}(s')$ – очікувана довгострокова винагорода у майбутньому стані s'.

Deep Deterministic Policy Gradients – це алгоритм НП, який поєднує в собі елементи Q-learning (оцінки функції корисності) та методу актор-критик (actor-critic).

Теорема 1 (градієнт політики). Для будь-якого Марковського процесу прийняття рішень, якщо параметри θ політики оновлюються пропорційно до градієнта її продуктивності ρ, тоді θ гарантовано збіжиться до локально оптимальної політики для ρ. Градієнт при цьому обчислюється так:

$$\Delta\theta \approx \alpha \frac{\partial \rho}{\partial \theta} = \alpha \sum_s d^{\pi}(s) \sum_a \frac{\partial \pi(s,a)}{\partial \theta} Q^{\pi}(s,a), \quad (2)$$

де α є позитивним розміром кроку, а $\frac{\partial \rho}{\partial \theta}$ – градієнт продуктивності політики. $\sum_s d^{\pi}(s)$ – зважений вплив станів, де d^{π} визначається як зважена знижена ймовірність станів, які зустрічаються, починаючи з s_0 і дотримуючись політики π:

$$d^{\pi}(s) = \sum_{t=0}^{\infty} \gamma^t P(s^t = s|s_0,\pi),$$

де γ – коефіцієнт дисконтування, що враховує вплив майбутніх нагород. $\sum_a \frac{\partial \pi(s,a)}{\partial \theta}$ – градієнт політики $\pi(s,\ a)$, який показує, як зміна параметрів θ впливає на ймовірність вибору дії a в стані s. $Q^{\pi}(s,a)$ – функція Q, яка відображає очікувану сукупну винагороду, якщо в стані s вибрати дію a і далі слідувати політиці π [10].

Цю теорему було додатково розширено в тій самій статті для випадку, коли замість політики π використовується апроксимаційна функція f, як показано в теоремі 2.

Теорема 2 (градієнт політики з апроксимацією функції). Теорема градієнта політики залишається справедливою для апроксимаційної функції f(s, a; w), яка представляє політику π, якщо оновлення ваг w прагнуть до нуля при збіжності до π:

$$\sum_s d^{\pi}(s) \sum_a \pi(s,a)[Q^{\pi}(s,a) - f(\mathrm{s},\mathrm{a};\ w)] \frac{\partial f(\mathrm{s},\mathrm{a};w)}{\partial w} = 0, \quad (3)$$

$[Q^{\pi}(s,a) - f(\mathrm{s},\mathrm{a};\ w)]$ – різниця між істинним Q значенням і апроксимацією f, $\frac{\partial f(\mathrm{s},\mathrm{a};w)}{\partial w}$ – градієнт апроксимаційної функції $f(\mathrm{s},\mathrm{a};\ w)$ щодо її параметрів w.

Якщо f сумісна з параметризацією політики в тому сенсі, що: $\frac{\partial f(\mathrm{s},\mathrm{a};w)}{\partial w} = \frac{\partial \pi(s,a)}{\partial \theta}\frac{1}{\pi(s,a)}$, де $\frac{1}{\pi(s,a)}$ – множник, що враховує зворотну ймовірність дії a у стані s. Тоді, $\frac{\partial \rho}{\partial \theta} = \sum_s d^{\pi}(s)\sum_a \frac{\partial \pi(s,a)}{\partial \theta} f(\mathrm{s},\mathrm{a};w)$ [10].

Для підвищення стабільності до DDPG можуть бути застосовані ті самі заходи, що й до DQNs, а саме: зменшення винагород, відтворення досвіду і використання окремої цільової мережі.

Для реалізації останнього заходу в DDPG додаються дві додаткові цільові мережі актора та критика, які використовуються для обчислення цільових значень Q, відокремлених від основних актора та критика. Основні мережі оновлюються на кожному кроці, а їхні ваги використовуються для поступового оновлення цільових мереж.

Незважаючи на те, що формули і підходи DDPG та Q-learning є відомими й детально описаними в літературі, їх застосування до задачі оптимізації транспортного потоку в міських умовах на основі реальних сценаріїв не отримало належного висвітлення. У цьому дослідженні новизна полягає в інтеграції цих алгоритмів із реалістичними симуляціями дорожнього руху, що базуються на мікроскопічному підході Aimsun. Також запропоновано нові метрики оцінювання ефективності управління, такі як показник швидкості (speed_score), який дає змогу точніше враховувати вплив на транспортні потоки.

Додатково дослідження демонструє переваги DDPG порівняно з Q-learning у сценаріях із безперервним простором дій, що раніше не було детально досліджено в контексті моделювання світлофорів. Це забезпечує як теоретичний внесок у стабільність і адаптивність алгоритмів навчання з підкріпленням, так і практичну значущість для подальшого використання в реальних міських транспортних мережах.

Отже, результати пропонованого експерименту розширюють межі застосування алгоритмів навчання з підкріпленням, забезпечуючи новий погляд на їх використання для складних, динамічних систем транспортного управління.

Вхідні дані

Використання системи моделювання для оцінювання програми глибокого навчання для керування дорожнім рухом дає змогу повністю спостерігати за дорожньою ситуацією. Однак, щоб зробити систему застосовною в реальних сценаріях, дуже важливо, щоб її вхідні дані могли бути отримані з інформації, яка зазвичай доступна в міських транспортних мережах.

Найдоступнішим і найпоширенішим джерелом даних про дорожній рух є детектори руху, датчики, розподілені по мережі, які вимірюють активність транспортних засобів. Серед різних типів найбільш поширеними є індукційні петлеві сповіщувачі, вбудовані під дорожнє покриття. Ці детектори надають дані в реальному часі, коли над ними проїжджають транспортні засоби. Основна інформація, яку вони зазвичай надають, містить нижченаведене:

- Кількість транспортних засобів: кількість транспортних засобів, що проїхали через детектор протягом періоду відбору проб.
- Середня швидкість: середня швидкість транспортних засобів протягом періоду вибірки.
- Заповненість: відсоток часу, протягом якого автомобіль займає зону детектора, корисний для виявлення заторів.

Щоб забезпечити сумісність із реальними системами, вхідні дані обмежуються для конкретної моделі такими стандартними виходами детектора руху: кількість транспортних засобів, середня швидкість і кількість людей. Крім того, ця модель містить повний опис транспортної мережі, включаючи схему доріг, перехресть та їх сполучення.

Це обмеження гарантує, що пропонований підхід може плавно переходити від моделювання до практичного застосування, використовуючи дані, які зазвичай доступні в інфраструктурі міського руху.

Дотримуючись самообмеження використовувати лише ті дані, які дійсно доступні в реальному сценарії, здійснено розроблення зведення стану трафіку на основі кількості транспортних засобів, середньої швидкості та заповненості.

Отже, визначається метрика з назвою "показник швидкості" (*speed_score*), яка для детектора i обчислюється за формулою:

$$speed\_score_i = min\left(\frac{avg\_speed_i}{\max\_speed_i}, 1.0\right), \quad (4)$$

де avg_speed_i позначає середнє значення швидкостей, виміряних детектором руху i, а $\max\_speed_i$ позначає максимальну швидкість на дорозі, де розташований детектор i. Необхідно звернути увагу, що оцінка швидкості, таким чином, коливається в межах [0, 1]. Цей показник стане основою для розробки представлення як стану середовища, так і винагород для алгоритму навчання з підкріпленням.

У цьому експерименті використовується мікроскопічний симулятор трафіку, який поділяє симуляцію на окремі кроки. На кожному кроці

моделюється невеликий фіксований проміжок часу, і стан транспортних засобів (наприклад положення, швидкість, прискорення) оновлюється відповідно до динаміки системи. За замовчуванням цей проміжок часу становить 0.75 секунди, і цей параметр залишився незмінним.

Однак цей проміжок часу є занадто коротким, щоб відобразити зміни в кількості транспортних засобів, які фіксують детектори. Тому потрібен більший період, протягом якого дані агрегуються. Назвемо цей період кроком епізоду (episode step) або просто "кроком", якщо це не викликає плутанини.

Таким чином:

1. Дані збираються на кожному кроці симуляції, але агрегуються лише кожен крок епізоду.

2. Агреговані дані передаються алгоритму DDPG як вхідні дані.

3. Для об'єднання показників швидкості (speed_scores) за кілька кроків симуляції використовується зважене середнє, де ваги визначаються пропорціями кількості транспортних засобів.

4. Таймінги світлофорів, згенеровані алгоритмом DDPG, застосовуються протягом наступного кроку епізоду.

Тривалість кроку епізоду була вибрана методом перебирання сітки (grid search). Оптимальним значенням визначено 120 секунд.

Щоб зберегти вектор стану навколишнього середовища, використовується оцінка швидкості (4), оскільки вона не лише належним чином підсумовує завантаженість мережі, але також включає поняття максимальної швидкості кожної дороги. Отже, вектор стану має один компонент на детектор, кожен з яких визначено формулою:

$$state_i = speed\_score_i\,, \qquad (5)$$

де $state_i$ – вектор стану; $speed\_score_i$ – показник швидкості.

Обґрунтування вибору показника швидкості полягає в тому, що чим вищий показник швидкості, тим вище швидкість транспортних засобів відносно максимальної швидкості на дорозі, а отже, тим вищий потік транспорту.

У реальному управлінні дорожнім рухом такі інструменти, як світлофори та тимчасові знаки, регулюють потік. Для спрощення моделювання цей метод фокусується на світлофорах, уникаючи хаотичних результатів, не керуючи безпосередньо окремими кольорами світла. Натомість світлофори синхронізуються на перехрестях, чергуючи фази (наприклад, зелений для одного напрямку, червоний для перпендикуляра).

Керування зосереджено на регулюванні тривалості фаз, а не загального часу циклу, зберігаючи синхронізацію між перехрестями для більш плавного руху. Функція softmax (також відома як нормалізована експоненціальна функція) нормалізує коригування фази, забезпечуючи постійність загального циклу, тоді як масштабування застосовується до 80% тривалості фази для підтримки мінімального часу.

Винагороди забезпечують зворотний зв’язок з алгоритмом навчання з підкріпленням щодо ефективності попередніх дій. Хоча в деяких підходах використовуються показники часу в дорозі (наприклад, довжина черги, затримки), цей метод покладається на реальні дані детекторів, як-от кількість транспортних засобів і швидкість. Обраним показником є оцінка швидкості, але для вимірювання впливу агента вводиться базова лінія: оцінка швидкості з моделювання без втручання агента.

Винагорода розраховується як різниця між оцінкою швидкості та базовою лінією, зважена за кількістю транспортних засобів, щоб визначити пріоритетність зон із високим трафіком, і масштабується за коефіцієнтом α, щоб утримувати винагороду в межах керованого діапазону за формулою:

$$reward_i = \alpha \cdot count_i \cdot (speed\_score_i - baseline_i) = \\ = \alpha \cdot count_i \cdot \left[min\left(\frac{avg\_speed_i}{\max\_speed_i}, 1.0\right) - baseline_i\right], \quad (6)$$

де $reward_i$ – винагорода; $baseline_i$ – базова лінія; $count_i$ – кількість транспортних засобів.

Щоб нормалізувати винагороди і тримати їх у межах [-1, +1], було розглянуто розділення на загальну кількість транспортних засобів на всіх детекторах, але відхилено, оскільки це робить винагороди на різних етапах моделювання непорівнянними (менша кількість транспортних засобів за час t принесе вищі винагороди). Замість цього коефіцієнт масштабування α = 1/50 був емпірично обраний, щоб зберегти масштабовані значення близько 1.0, забезпечуючи керовані шкали градієнта. На відміну від зменшення винагороди, це зберігає інформацію про розміри винагороди.

Кожен детектор обчислює свою винагороду незалежно на кожному кроці часу, без комбінації значень, і все це надходить безпосередньо в процес оптимізації DDPG. Через стохастичну природу мікросимулятора результати залежать від випадкового початкового числа, тому для забезпечення узгодженості базові лінії беруться із симуляцій з відповідними початковими значеннями.

Експерементальні установки

Щоб оцінити алгоритм глибокого НП, було протестовано мережу трафіку зростаючої складності. Алгоритм DDPG контролює всі фази світлофора, використовуючи дані детектора всієї мережі. Було проведено порівняння з Q-навчанням, яке керує

окремими фазами перетину за допомогою вхідних даних локального детектора та категоріальних просторів стану/дій, а також випадкового агента, який призначає випадкові моменти часу.

Агент Q-навчання використовує мозаїчне кодування для дискретизації значень простору станів на чотири діапазони, контролюючи тривалість фаз із попередньо визначеними співвідношеннями (наприклад, 0,2, 0,5, 1,0). Ця мультиагентна настройка гарантує, що агенти обмінюються деякими вхідними даними, але діють незалежно, використовуючи коригування фаз для підтримки постійної тривалості циклу.

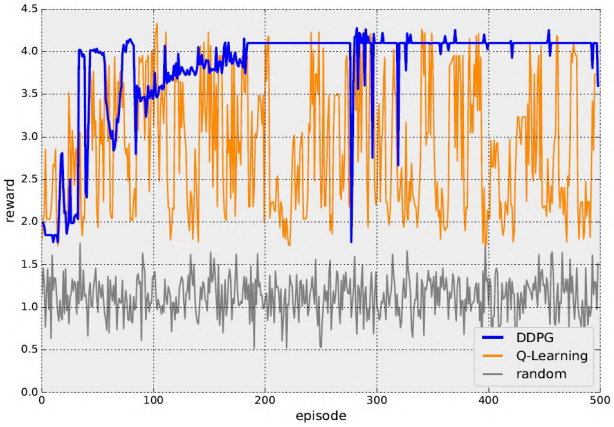

Рисунок 2 – Порівняння продуктивності алгоритму в мережі

Випадковий агент застосовує випадкові хронометражі в межах [0, 1] з подальшим коригуванням фази для послідовних циклів. Через стохастичну природу симулятора дорожнього руху було проведено експерименти з рандомізованим початковим числом, щоб запобігти переобладнанню. Результати були агреговані для кількох прогонів із відображенням середніх, максимумів або мінімумів відповідно до надійного аналізу.

Проста мережа, показана на рис. 1, складається лише з перехрестя двох 2-смугових доріг. На перехресті транспортні засоби можуть їхати прямо або повертати праворуч. Поворот ліворуч заборонено, що спрощує динаміку руху та світлофорні фази. Є 8 детекторів (на кожній дорозі один детектор перед перехрестям і ще один після нього).

У групі світлофорів є дві фази: фаза 1 дозволяє горизонтальний рух, а фаза 2 дозволяє вертикальний рух. Фаза 1 триває 15 секунд, а фаза 2 триває 70 секунд, з 5-секундною міжфазою. Фази 1 і 2 мають незбалансовану тривалість навмисно, щоб горизонтальна дорога накопичувала транспортні засоби протягом тривалого часу. Це дає можливість цьому алгоритму легко покращувати потік трафіку за допомогою зміни тривалості фази. Моделювання триває 1 годину, а попит на транспортні засоби є постійним: на кожну пару центроїдів припадає 150 транспортних засобів.

Результати

Щоб оцінити продуктивність підходу DDPG порівняно як із звичайним Q-навчанням, так і з випадковими таймінгами в даній тестовій мережі, основним еталонним показником має бути середня винагорода (reward) за епізод (зауважте, що, як описано в (5), насправді існує вектор винагород, з одним елементом на детектор у мережі, тому обчислюється середня винагорода) найкращого експериментального випробування, розуміючи «найкращий» експеримент як той, де максимально отримано середню винагороду за епізод (episode).

На рис. 2 можна знайти порівняння продуктивності для мереж. І підхід DDPG, і класичне Q-навчання досягають однакових рівнів винагороди. З іншого боку, помітні відмінності в конвергенції обох підходів: хоча Q-навчання є нестабільним, DDPG залишається надзвичайно стабільним після досягнення максимальної продуктивності.

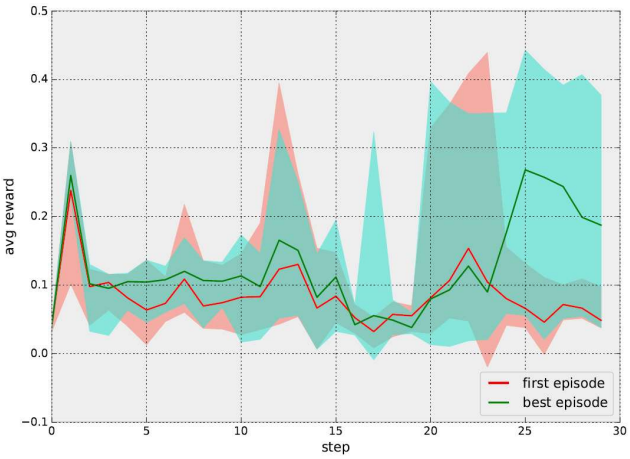

Рисунок 3 – Еволюція алгоритму DDPG в мережі в епізоді

Можна далі досліджувати поведінку алгоритму, вивчаючи продуктивність внутрішнього епізоду: рис. 3 показує продуктивність першого епізоду алгоритму DDPG і його продуктивність у найкращому епізоді; продуктивність показана як середня винагорода (avg reward) за кроком (step) із мінімальними та максимальними діапазонами (середнє, мінімальне та максимальне значення обчислюються для всіх випробувань, виконаних для одного експерименту). Очевидно, що покращення порівняно з базовим рівнем не є постійними протягом епізоду, але помітні до кінця.

Таку саму модель поведінки можна побачити у продуктивності алгоритму Q-навчання в епізоді, показаному на рис. 4.

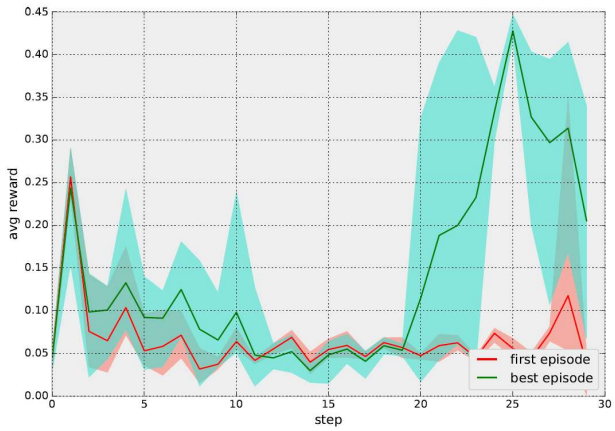

Рисунок 4 – Еволюція алгоритму Q-навчання в мережі в епізоді

Також важливо звернути увагу на недоліки цього дослідження.

Спрощеність моделі. У цьому дослідженні використовується проста симуляційна мережа з одним перехрестям, яка не повністю відображає складність реальних міських транспортних систем.

У реальних умовах значно більше перехресть, потоки є багатонаправленими, а динаміка трафіку більш варіативна.

Чутливість до параметрів. Ефективність алгоритму залежить від налаштувань гіперпараметрів, як-от коефіцієнт дисконтування та параметри нейронної мережі. Неправильно підібрані значення можуть знижувати продуктивність і стабільність алгоритму.

Обмеження даних. Симуляція базується на ідеальних умовах, зокрема точності датчиків і відсутності непередбачуваних факторів, як-от погодні умови, аварії чи поведінка водіїв. У реальному житті такі фактори можуть значно впливати на ефективність запропонованих алгоритмів.

Висновок

Мета цього дослідження – експериментальна перевірка ефективності алгоритму DDPG для оптимізації управління транспортними потоками в міських умовах. На основі виконаних симуляцій було досягнуто таких результатів.

Алгоритм DDPG продемонстрував високу ефективність і стабільність у простій моделі транспортної мережі, забезпечуючи стабільну конвергенцію порівняно з Q-learning, який виявився менш стабільним.

Експериментальні результати засвідчили, що DDPG здатен покращувати середню винагороду за епізод ближче до завершення епізоду, що свідчить про його здатність адаптуватися до динаміки трафіку.

Запропонована метрика оцінки – показник швидкості (speed_score) – уможливила точніше оцінити вплив алгоритму на транспортні потоки та продемонструвала свою корисність для навчання алгоритмів.

Попри успішну реалізацію поставленої мети, дослідження має певні обмеження, зокрема спрощеність моделі та відсутність тестування на реальних даних. Ці аспекти потребують додаткового вивчення.

У подальших дослідженнях доцільно розширити модель на складніші транспортні мережі та протестувати підхід із використанням даних реальних міських умов. Це допоможе краще оцінити потенціал алгоритму DDPG для впровадження в реальні системи управління дорожнім рухом.

Список літератури

1. Van der Pol E. Deep reinforcement learning for coordination in traffic light control. 2016. URL: https://www.researchgate.net/publication/315810688_Deep_Reinforcement_Learning_for_Coordination_in_Traffic_Light_Control_MSc_thesis.

2. Lecun Y., Bottou L., Bengio Y., Haffner P. Gradient-based learning applied to document recognition. *IEEE*. 1998. Vol. 86, № 11. С. 2278–2324. DOI: https://doi.org/10.1109/5.726791.

3. LA P., Bhatnagar S. Reinforcement Learning With Function Approximation for Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems*. 2011. Vol. 12, № 2. P. 412–421. DOI: https://doi.org/10.1109/TITS.2010.2091408.

4. Acharya S., Dash K. K., Chaini R. Fuzzy Logic: An Advanced Approach to Traffic Control. *Learning and Analytics in Intelligent Systems*. 2020. DOI: https://dx.doi.org/10.4018/ijide.2014010103.

5. Daneshfar F., Akhlaghian F., Mansoori F. Adaptive and cooperative multi-agent fuzzy system architecture. *14th International CSI Computer Conference*. 2009. P. 30–34. DOI: https://doi.org/10.1109/CSICC.2009.5349439.

6. Mnih V., Kavukcuoglu K., Silver D., Graves A., Antonoglou I., Wierstra D., Riedmiller M. Playing Atari with Deep Reinforcement Learning. 2013. URL: https://doi.org/10.48550/arXiv.1312.5602

7. Tesauro G. Temporal difference learning and td-gammon. *Communications of the ACM*. 1995. Vol. 38, № 3. P. 58–68. DOI: https://doi.org/10.1145/203330.203343.

8. Pollack J. B., Blair A. D. Why did td-gammon work? Advances in Neural Information Processing Systems. 1997. С. 10–16. DOI: https://doi.org/10.1145/203330.203343.

9. Kingma D., Ba J. Adam: A method for stochastic optimization. 2014. URL: https://arxiv.org/abs/1412.6980.

10. Sutton R., Mcallester D. A., Singh S., Mansour Y. Policy Gradient Methods for Reinforcement Learning with Function Approximation. Adv. Neural Inf. Process. Syst. 12. URL: https://dl.acm.org/doi/10.5555/3009657.3009806.

11. Lillicrap T. P., Hunt J. J., Pritzel A., Heess N., Erez T., Tassa Y., Silver D., Wierstra D. Continuous control with deep reinforcement learning. URL: https://doi.org/10.48550/arXiv.1509.02971.

12. Silver D., Lever G., Heess N., Degris T., Wierstra D., Riedmiller M. Deterministic policy gradient algorithms. *31st International Conference on Machine Learning (ICML-14)*. 2014. P. 387–395. URL: http://dx.doi.org/10.13140/RG.2.2.16324.71048.

13. Aimsun Next user manual, version 24.0.1. URL: https://docs.aimsun.com/next/24.0.1/.

14. Levytskyi V., Kruk, Lopuha O., Sereda D., Sapaiev V., Matsiievskyi O. Use of Deep Learning Methodologies in Combination with Reinforcement Techniques within Autonomous Mobile Cyber-physical Systems. 2024 *IEEE*. DOI: https://doi.org/10.1109/SIST61555.2024.10629589.

Стаття надійшла до редколегії 04.03.2025

Levytskyi Volodymyr
Postgraduate of the department of information technologies,
https://orcid.org/0000-0003-1829-488X
Kyiv National University of Construction and Architecture, Kyiv

OPTIMIZATION OF TRANSPORT TRAFFIC IN A SIMPLE NETWORK USING DEEP LEARNING WITH REINFORCEMENT

Abstract. *Traffic flow optimization in urban environments remains one of the key challenges of modern research, even despite the significant volume of scientific works devoted to this topic. Despite the achievements, this problem still does not have a universal solution that would work effectively in real-world scenarios. One of the main difficulties is the processing of a large array of input data, in particular, traffic data, which constantly comes from sensors installed throughout the urban road network. Traditionally, due to the scale of the task, researchers have focused on the development of systems with localized agents. Such agents usually manage traffic at individual intersections, while their coordination is carried out within the framework of multi-stream agent systems. However, modern approaches take into account the volume and complexity of input data through the use of deep learning methods. In particular, the use of the deep deterministic policy gradient (DDPG) algorithm is proposed, on the basis of which large input data can be processed. As part of the experimental study, a simple intersection model was tested to verify the effectiveness of the approach. The DDPG algorithm performed better in the simple model compared to Q-learning. DDPG provided a reward in the range of 4-4.3 points, while the reward of Q-learning was in the range of 2-4 points. To evaluate the performance of the DDPG approach compared to Q-learning and random timings, the main criterion is the average reward per episode. DDPG and Q-learning achieve similar reward levels, but DDPG shows stable convergence (0.04-0.21 points), while Q-learning remains unstable (0.04-0.43 points). The study of intra-episode performance shows that DDPG achieves improvements mainly closer to the end of the episode. Overall, this algorithm has proven successful for this scenario, and the results obtained can serve as a basis for further improvements and applications in more complex traffic scenarios.*

Keywords: DDPG; Aimsun; Q-learning; traffic

References

1. Van der Pol E. Deep reinforcement learning for coordination in traffic light control. 2016. URL: https://www.researchgate.net/publication/315810688_Deep_Reinforcement_Learning_for_Coordination_in_Traffic_Light_Control_MSc_thesis.

2. Lecun Y., Bottou L., Bengio Y., Haffner P. Gradient-based learning applied to document recognition. *IEEE*. 1998. Vol. 86, № 11. С. 2278–2324. DOI: https://doi.org/10.1109/5.726791.

3. LA P., Bhatnagar S. Reinforcement Learning With Function Approximation for Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems*. 2011. Vol. 12, № 2. P. 412–421. DOI: https://doi.org/10.1109/TITS.2010.2091408.

4. Acharya S., Dash K. K., Chaini R. Fuzzy Logic: An Advanced Approach to Traffic Control. *Learning and Analytics in Intelligent Systems*. 2020. DOI: https://dx.doi.org/10.4018/ijide.2014010103.

5. Daneshfar F., Akhlaghian F., Mansoori F. Adaptive and cooperative multi-agent fuzzy system architecture. *14th International CSI Computer Conference*. 2009. P. 30–34. DOI: https://doi.org/10.1109/CSICC.2009.5349439.

6. Mnih V., Kavukcuoglu K., Silver D., Graves A., Antonoglou I., Wierstra D., Riedmiller M. Playing Atari with Deep Reinforcement Learning. 2013. URL: https://doi.org/10.48550/arXiv.1312.5602

7. Tesauro G. Temporal difference learning and td-gammon. *Communications of the ACM*. 1995. Vol. 38, № 3. P. 58–68. DOI: https://doi.org/10.1145/203330.203343.

8. Pollack J. B., Blair A. D. Why did td-gammon work? Advances in Neural Information Processing Systems. 1997. C. 10–16. DOI: https://doi.org/10.1145/203330.203343.

9. Kingma D., Ba J. Adam: A method for stochastic optimization. 2014. URL: https://arxiv.org/abs/1412.6980.

10. Sutton R., Mcallester D. A., Singh S., Mansour Y. Policy Gradient Methods for Reinforcement Learning with Function Approximation. Adv. Neural Inf. Process. Syst. 12. URL: https://dl.acm.org/doi/10.5555/3009657.3009806.

11. Lillicrap T. P., Hunt J. J., Pritzel A., Heess N., Erez T., Tassa Y., Silver D., Wierstra D. Continuous control with deep reinforcement learning. URL: https://doi.org/10.48550/arXiv.1509.02971.

12. Silver D., Lever G., Heess N., Degris T., Wierstra D., Riedmiller M. Deterministic policy gradient algorithms. *31st International Conference on Machine Learning (ICML-14)*. 2014. P. 387–395. URL: http://dx.doi.org/10.13140/RG.2.2.16324.71048.

13. Aimsun Next user manual, version 24.0.1. URL: https://docs.aimsun.com/next/24.0.1/.

14. Levytskyi V., Kruk, Lopuha O., Sereda D., Sapaiev V., Matsiievskyi O. Use of Deep Learning Methodologies in Combination with Reinforcement Techniques within Autonomous Mobile Cyber-physical Systems. 2024 *IEEE*. DOI: https://doi.org/10.1109/SIST61555.2024.10629589.

Посилання на публікацію

APA Levytskyi, V. (2025). Optimization of transport traffic in a simple network using deep learning with reinforcement. *Management of Development of Complex Systems*, 61, 151–159, dx.doi.org\10.32347/2412-9933.2025.61.151-159.

ДСТУ Левицький В. В. Оптимізація руху транспорту в простій мережі за допомогою глибокого навчання з підкріпленням. *Управління розвитком складних систем.* Київ, 2025. № 61. С. 151 – 159, dx.doi.org\10.32347/2412-9933.2025.61.151-159.