

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ КАЗАХСТАН**

**Некоммерческое акционерное общество «Казахский национальный
исследовательский технический университет имени К.И.Сатпаева»**

Институт автоматизации и информационных технологий

Кафедра «Высшая математика и моделирование»

Нагай Анастасия Сергеевна

**Математическое моделирование транспортных потоков на примере управления
светофорами**

ДИПЛОМНАЯ РАБОТА

6B06103 – Математическое и компьютерное моделирование

Алматы 2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество «Казакский национальный исследовательский
технический университет имени К.И.Сатпаева»

Институт автоматик и информационных технологий

Кафедра «Высшая математика и моделирование»



ДИПЛОМНАЯ РАБОТА

На тему: «Математическое моделирование транспортных потоков на примере управления
светофорами»

6B06103 – Математическое и компьютерное моделирование

Выполнил

Нагай А.С.

Рецензент

доктор физ-мат. наук, профессор
департамента «Математика»
университета имени С. Демиреля
Н.С. Даирбеков
"06" июня 2023 г.



Научный руководитель

канд. физ-мат. наук, асоц. проф.
Р. Н. Зимин
"06" июня 2023 г.

Алматы 2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество «Казакский национальный
исследовательский технический университет имени К.И. Сатпаева»

Институт автоматизации и информационных технологий

Кафедра «Высшая математика и моделирование»

6B06103 – Математическое и компьютерное моделирование

УТВЕРЖДАЮ
Заведующая кафедрой ВМиМ
канд. физ-мат. наук,
ассоциированный профессор
П. А. Гулешева
«06» июля 2023 г.



ЗАДАНИЕ

на выполнение дипломной работы

Обучающемуся: Нагай А.С.

Тема: «Математическое моделирование транспортных потоков на примере
управления светофорами»

Утверждена приказом проректора по академической работе: № 408-П/Ө
от «23» ноября 2022 г.

Срок сдачи законченной работы: «22» мая 2023 г.

Исходные данные к дипломной работе: техническая документация к SUMO,
TraCI и Python

Краткое содержание дипломной работы:

- а) теория по моделированию транспортных потоков;
- б) создание симуляционной модели в программе SUMO;
- в) проведение обучения с помощью алгоритма Q-learning;
- г) проведение экспериментов и анализ результатов.

Рекомендуемая основная литература: техническая литература по Q-learning и
математическому моделированию транспортных потоков, статьи, посвященные
программе SUMO и адаптивному управлению из 8 наименований.



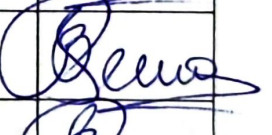


ГРАФИК

подготовки дипломной работы

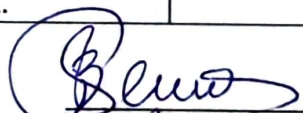
Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
Теория по моделированию транспортных потоков	27.01.23	Выполнено
Создание симуляционной модели в программе SUMO	28.02.23	Выполнено
Обучение с помощью алгоритма Q-learning	21.03.23	Выполнено
Проведение экспериментов и анализ результатов	18.04.23	Выполнено

Подписи

консультантов и нормоконтролера на законченную дипломную работу с указанием относящихся к ним разделов

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Теория по моделированию транспортных потоков	канд. физ.-мат. наук, ассоц. проф. Зимин Р.Н.	2.06.23	
Создание симуляционной модели в программе SUMO	канд. физ.-мат. наук, ассоц. проф. Зимин Р.Н.	2.06.23	
Обучение с помощью алгоритма Q-learning	канд. физ.-мат. наук, ассоц. проф. Зимин Р.Н.	2.06.23	
Проведение экспериментов и анализ результатов	канд. физ.-мат. наук, ассоц. проф. Зимин Р.Н.	2.06.23	
Нормоконтролер	канд. физ.-мат. наук, ассоц. проф. Шатманов Ж.Ж.	05.06.2023	

Научный руководитель

 Р.Н. Зимин

Задание принял к исполнению обучающийся

 А.С. Нагай

Дата

« 09 » января 2023 г.

АНДАТПА

Дипломдық жұмыста бағдарламды басқару мысалында көлік ағындарын математикалық модельдеу қарастырылады. Негізгі мақсат — көлік ағындарын оңтайландыру мақсатында көлік желісінің екі қиылысының бірі үшін адаптивті басқаруды жүргізу болып табылады.

Python бағдарламалау тілінде жүзеге асырылған күшейтілген оқыту алгоритмі Q-learning бағдарлам жоспарын табу құралы ретінде әрекет етеді, ал SUMO бағдарламалық құралы арқылы жасалған симуляциялық моделі оқу ортасына айналады.

Бейімделетін бағдарлам жоспарын алғаннан кейін оның жұмысы келесі критерийлер бойынша бекітілген жоспарлармен салыстырылады:

- 1) автомобильдер саны бойынша қиылыстың шеттері арасындағы тепе-теңдікті сақтау;
- 2) қиылыстың шетіндегі автомобильдер санының азайту.

АННОТАЦИЯ

В данной дипломной работе рассматривается математическое моделирование транспортных потоков на примере управления светофорами. Основной целью является внедрение адаптивного управления для одного из двух перекрестков транспортной сети в целях оптимизации транспортных потоков.

В качестве инструмента для поиска светофорного плана выступит алгоритм обучения с подкреплением – Q-learning, реализованный на языке программирования Python, а средой обучения станет симуляционная модель, созданная при помощи программного обеспечения SUMO.

После получения адаптивного светофорного плана будет произведено сравнение его работы с фиксированными планами по следующим критериям:

- 1) сохранение баланса между ребрами перекрестка по числу автомобилей;
- 2) снижение числа автомобилей на ребрах перекрестка.

ABSTRACT

The thesis considers the mathematical modeling of traffic flows on the example of traffic lights control. The main goal is to conduct adaptive control for one of the two intersections of the transport network in order to optimize traffic flows.

The reinforcement learning algorithm – Q-learning, implemented in the Python programming language, will act as a tool for finding a traffic light plan, and the learning environment will be a simulation model created through SUMO software.

After receiving an adaptive traffic light plan, its operation will be compared with fixed plans according to the criteria:

- 1) maintaining the balance between the edges of the intersection by the number of cars;
- 2) reducing the number of cars on the edges of the intersection.

СОДЕРЖАНИЕ

Введение	7
1 Моделирование транспортных потоков	8
1.1 Основные подходы и методы моделирования	8
1.2 Модель, лежащая в основе SUMO	9
2 Создание симуляционной модели	12
2.1 Особенности программы SUMO	12
2.2 Характеристики транспортной сети	13
2.3 Формирование спроса на трафик с помощью jtrrouter	14
2.4 Запуск симуляции через файл конфигурации sumocfg	15
2.5 Взаимодействие с симуляцией при помощи TraCI	18
3 Обучение с помощью алгоритма Q-learning	20
3.1 Принцип работы обучения с подкреплением	20
3.2 Применение Q-learning в управления светофорами	22
4 Эксперименты и результаты	26
4.1 Определение начального положения системы	26
4.2 Поиск адаптивного светофорного плана для одного перекрестка	27
4.3 Поиск адаптивного светофорного плана для двух перекрестков	30
4.4 Сравнение адаптивного светофорного плана с фиксированными	33
Заключение	37
Список использованной литературы	38
Приложение А	39
Приложение Б	40
Приложение В	43

ВВЕДЕНИЕ

На сегодняшний день основным способом регулирования движения на дорогах является использование светофоров. Однако в современных городах с каждым годом увеличивается число транспортных средств, что приводит к серьезной проблеме в виде дорожных заторов, с которыми традиционные методы управления не всегда могут справиться. Это заставляет задуматься о создании адаптивных светофоров, способных приспосабливаться к меняющейся обстановке на дороге.

В данной работе будет исследована возможность использования математических моделей и алгоритмов машинного обучения для оптимизации работы светофорного плана на примере двух перекрестков.

Актуальность работы обусловлена необходимостью повышения эффективности управления транспортными потоками, уменьшения пробок и повышения общего качества транспортной системы.

Целью дипломной работы является внедрение адаптивного управления транспортными потоками с помощью алгоритмов машинного обучения, позволяющего сократить количество заторов на одном из двух перекрестков и сохранить баланс по числу автомобилей между всеми направлениями.

Задачи исследования:

- 1) создание симуляционной модели дорожной сети с помощью программы SUMO;
- 2) анализ транспортного потока и адаптивное управление с помощью алгоритма Q-learning;
- 3) сравнение эффективности адаптивного сигнального плана с фиксированными.

1 Моделирование транспортных потоков

1.1 Основные подходы и методы моделирования

Математическое моделирование транспортных потоков берет свое начало в середине XX века. Причиной этому послужили увеличение числа автомобилей в городах, как следствие, необходимость в управлении движением с целью повышения пропускной способности дорог и уменьшения количества пробок.

Выделяют два основных подхода в данной сфере: детерминистический и вероятностный (или стохастический) [4]. В первом случае предполагается, что поведение транспортных средств и их взаимодействие друг с другом определяется набором правил и параметров, следовательно, случайности и вариации в системе отсутствуют. Стохастический подход, напротив, рассматривает поток как вероятностный процесс. В нем учитываются вероятности различных событий и случайных факторов.

Методы моделирования транспортных потоков делятся на группы: макро-моделирование, микро-моделирование, мезомоделирование и субмикро-моделирование [5].

Макроскопические модели или модели-аналоги описывают транспортный поток, воспринимая его как физический поток, то есть характеристики и поведение отдельно взятых транспортных средств не рассматриваются. Они могут быть основаны на принципах гидродинамики и уделяют внимание таким переменным, как плотность движения и средняя скорость.

Микроскопические модели в некотором смысле являются противоположностью макроскопических моделей. Здесь учитываются индивидуальные характеристики транспортных средств. Для каждого автомобиля генерируется отдельный маршрут и поведение на дороге. Пример такой модели — модель следования за лидером. Основная идея заключается в том, что каждый автомобиль следует за лидером (транспортным средством, находящимся впереди), определяющим его поведение. Например, изменение скорости или резкое торможение головного транспортного средства вызовет изменение параметров всех автомобилей, стоящих за ним.

Мезоскопические модели — это промежуточный уровень между микроскопическими и макроскопическими моделями. В них транспортные средства группируются на основе схожих характеристик и затем задаются параметры для каждой отдельной группы.

Субмикроскопические модели — это более расширенная версия микроскопических моделей, содержащая дополнительные параметры. Например, вращение двигателя или приоритетные действия водителя.

Большинство программ, которые используют сегодня для создания симуляционных моделей транспортных потоков, основаны именно на

микроскопическом моделировании. Среди них как коммерческие программные пакеты, например, такие как PTV Vissim, имеющий в своих инструментах интерфейс с 3D-визуализацией, так и бесплатные программы с открытым исходным кодом. К последним относятся MatSIM и SUMO («Моделирование городской мобильности»). В данной работе будет использована именно программа SUMO.

1.2 Модель, лежащая в основе SUMO

В документации SUMO указано, что в основе программы используется модель, разработанная изначально Гиппсом, а затем дополненная Стефаном Краусом в 1998 году [3]. Это микроскопическую модель следования за автомобилем, в основе идея предотвращения столкновений. В данной модели водитель держится от впереди идущего транспортного средства на безопасном расстоянии и скорости, которые позволяют адаптироваться к изменению движения лидера и предотвратить аварийную ситуацию [6].

Целью Гиппса было создание модели, которая описывала бы интуитивное поведение водителей на дороге. По этой причине он ввел дополнительную переменную, помимо тех, что используется для описания движения автомобиля. Все параметры для n -ого автомобиля представлены в таблице 1.1 [6].

Таблица 1.1 — Параметры модели, определенные Гиппсом

Параметры	Описание
a_n	Максимальное ускорение
$b_n, (b_n < 0)$	Максимальное замедление (торможение)
s_n	Сумма длины автомобиля и минимальной дистанции до следующего транспортного средства
V_n	Желаемая скорость
$x_n(t)$	Положение автомобиля в момент времени t
$v_n(t)$	Скорость автомобиля в момент времени t
τ	Время реакции водителя

Модель описывает движение n -ого автомобиля. При этом транспортное, следующее перед ним, обозначается как $n - 1$. Общее правило движения вытекает из нескольких уравнений.

Первое уравнение Гиппса для описания модели ограничивает скорость водителя так, чтобы она не превышала желаемой скорости:

$$v_n(t + \tau) \leq v_n(t) + 2.5a_n\tau \left(1 - \frac{v_n(t)}{V_n}\right) \sqrt{0.025 + \frac{v_n(t)}{V_n}} \quad (1.1)$$

Следующие два уравнения определяют положения n -ого и $n - 1$ автомобилей при резком торможении лидирующего транспортного средства:

$$x_{n-1}^* = x_{n-1}(t) - \frac{v_{n-1}(t)^2}{2b_{n-1}} \quad (1.2)$$

$$x_n^* = x_n(t) + 0.5\tau[v_n(t) + v_n(t + \tau)] - \frac{v_n(t + \tau)^2}{2b_n} \quad (1.3)$$

Для предотвращения столкновений должно выполняться условие:

$$x_{n-1}^* - s_n \geq x_n^* \quad (1.4)$$

Поставляя в неравенства (1.4) уравнения (1.2) и (1.3), получаем ограничение на скорость n -ого автомобиля при торможении лидирующего транспортного средства:

$$v_n(t + \tau) \leq b_n\tau + \sqrt{b_n^2\tau^2 - b_n(2[x_{n-1}(t) - s_{n-1} - x_n(t)] - v_n(t)\tau - \frac{v_{n-1}(t)^2}{b'})} \quad (1.5)$$

где b' — это b_{n-1} .

Обозначим правые части неравенства (1.1) и (1.5), как A и B соответственно. Тогда общее правило выбора скорости в модели Гиппса выглядит следующим образом:

$$v_n(t + \tau) = \min\{A, B\} \quad (1.6)$$

Краус дополнил данную модель, добавив значения безопасной и «желаемой» [6]. Безопасная скорость вычисляется по формуле v_{safe} :

$$v_{safe} = v_l(t) + \frac{g(t) - v_l(t)\tau}{\tau_b + \tau} \quad (1.7)$$

где $v_l(t)$ — скорость лидера;

l — длина автомобиля;

$g(t)$ — отставание от лидирующего автомобиля в момент времени t ;

τ_b — шкала времени, равная $\frac{\bar{v}}{b}$, где b — максимальное замедление.

Так как безопасная скорость может быть больше, чем способно развить транспортное средство, результирующая или «желаемая» скорость выбирается по правилу:

$$v_{\text{des}} = \min[v_{\text{max}}, v(t) + a(v)\Delta t, v_{\text{safe}}(t)] \quad (1.8)$$

где v_{max} — максимальная скорость;

v — скорость;

a — ускорение;

Δt — продолжительность шага моделирования.

При этом правила изменения скорости и перемещения описываются уравнениями:

$$v(t + \Delta t) = \max[0, v_{\text{des}}(t) - \eta], \quad (1.9)$$

$$x(t + \Delta t) = x(t) + v\Delta t \quad (1.10)$$

где η — случайное возмущение, отражающее случайную динамику модели, происходящую при резком ускорении или торможении.

Случайное возмущение определяется уравнением:

$$\eta = a\varepsilon \quad (1.11)$$

где ε — случайная величина от 0 до 1.

2 Создание симуляционной модели

2.1 Особенности программы SUMO

SUMO — это программа для моделирования дорожного движения и транспортных систем. Ее основные характеристики включают следующие пункты:

1) возможность создания гибкой дорожной сети с различными типами дорог, перекрестками, светофорами, пешеходными зонами и другими элементами;

2) разнообразие типов транспортных средств, такие как легковые автомобили, грузовики, автобусы, велосипеды, каждый из которых может быть настроен с учетом индивидуальных характеристик;

3) наличие гибкого графического интерфейса, который позволяет визуализировать дорожную сеть и движение транспортных средств;

4) интеграция с другими инструментами моделирования и симуляции для генерации потоков транспортных средств, создания расписаний движения и применения алгоритмов управления;

5) предоставление различных интерфейсов и API для расширения функциональности, что дает возможность разрабатывать собственные алгоритмы управления, модели движения и другие компоненты для адаптации SUMO под конкретные задачи;

6) поддержка различных форматов данных для импорта и экспорта дорожных сетей, потоков транспортных средств и других параметров моделирования. Это позволяет интегрировать данные из различных источников и обмениваться информацией.

SUMO состоит из нескольких отдельных программ и инструментов. Основные компоненты SUMO включают [7]:

1) `sumo-gui` — графический интерфейс пользователя, который предоставляет возможность визуализации дорожной сети, движения транспортных средств и анализа результатов симуляции;

2) `sumo` — командная строка, используемая для запуска симуляций и выполнения различных операций `sumo`, таких как импорт и экспорт данных, генерация потоков транспортных средств и анализ результатов.

3) `netedit` — интерактивный инструмент для создания и редактирования дорожных сетей;

4) `netconvert` — утилита командной строки для конвертации дорожных сетей из одного формата в другой;

5) `duarouter` — инструмент для определения маршрутов транспортных средств на основе заданной дорожной сети и потоков транспортных средств;

6) `sumolib` — библиотека Python, предоставляющая программный интерфейс для работы с данными и функциями SUMO. С помощью `sumolib` можно импортировать и экспортировать данные, управлять дорожными сетями,

транспортными средствами и другими компонентами SUMO, а также выполнять анализ результатов симуляции;

7) `traci` — интерфейс связи сумо, позволяющий внешним программам получать информацию о текущем состоянии симуляции, управлять транспортными средствами, светофорами и другими аспектами моделирования в реальном времени;

8) `flowrouter` — инструмент для генерации потоков транспортных средств на основе заданных параметров, таких как исходные и конечные точки, распределение и скорость движения. `Flowrouter` используется для создания случайных или сценарных потоков транспортных средств в симуляции;

9) `jtrrouter` — это инструмент в составе SUMO, предназначенный для генерации маршрутов транспортных средств на основе заданных исходных и конечных точек, а также других параметров, таких как предпочтения маршрутизации и транспортные сети.

Благодаря большому количеству инструментов, SUMO позволяет выполнять моделирование различных сценариев движения и создавать реалистичные модели дорожного движения. Таким образом, данная программа предоставляет возможность для более точного моделирования и имитации транспортных потоков на дорогах и перекрестках.

2.2 Характеристики транспортной сети

Транспортная сеть SUMO — это граф, состоящий из узлов и ребер. Узлы представляют собой перекрестки, развязки или другие точки на дороге, где возможно изменение направления движения транспортных средств. Ребра — это дороги, соединяющие эти узлы. Каждое ребро однонаправленно и имеет свою длину, форму, количество полос, ограничение скорости [1].

SUMO также позволяет определять дополнительные характеристики для каждого узла и ребра, такие как наличие светофоров, пешеходных переходов и других объектов, которые влияют на поток транспорта.

Для создания дорожной сети в SUMO необходимо определить узлы и дуги, их характеристики и связи между ними. Эти данные описываются вручную в трех XML-файла [7]:

1) `test.nod.xml` — файл, содержащий информацию о узлах дорожной сети, а именно идентификатор, координаты и наличие светофора;

2) `test.edg.xml` — файл, содержащий информацию о ребрах дорожной сети, включая идентификатор ребра, начальный узел и конечный, количество полос, допустимую скорость;

3) `test.con.xml` — файл, содержащий информацию о правилах соединения ребер и перемещения из одного ребра в другое.

Эти XML-файлы являются частью конфигурационной структуры SUMO. С их помощью создается единый файл, содержащий полную информацию о транспортной сети. Для этой в три вышеперечисленных файла подаются на вход в инструмент командной строки netconvert. Он преобразует их в формат, который может быть использован SUMO для имитации дорожного движения (рисунок 2.1).

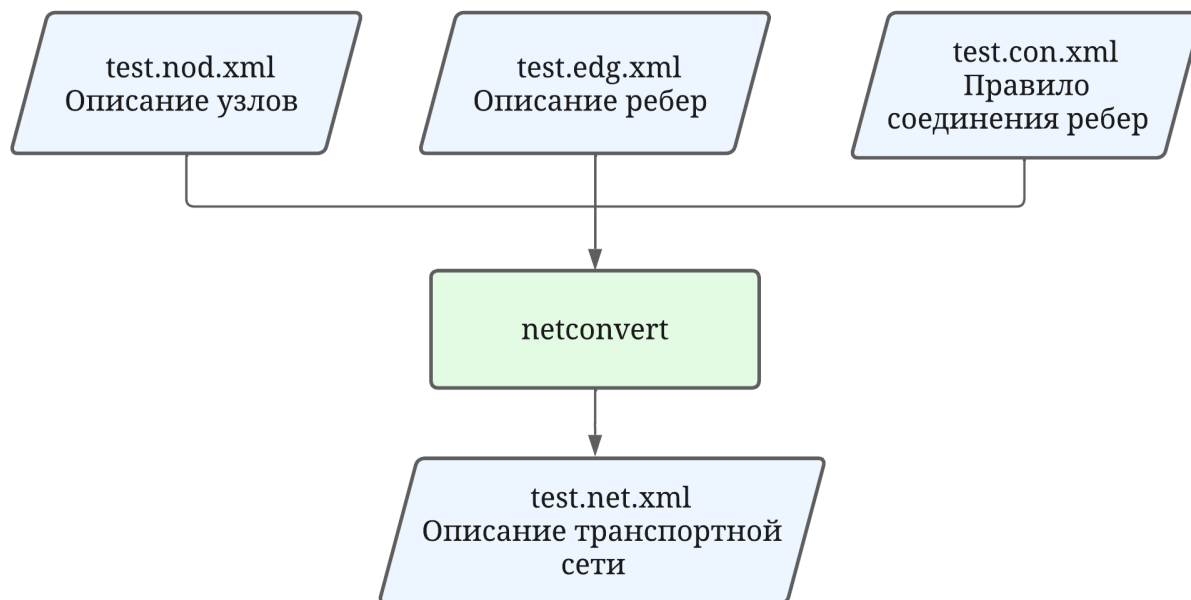


Рисунок 2.1 — Схема, отражающая принцип работы netconvert

В результате генерируются единый test.net.xml, содержащий все характеристики необходимой транспортной сети, который будет использован для создания модели спроса и конфигурационного файла.

2.3 Формирование спроса на трафик с помощью jtrrouter

Как уже упоминалось выше, SUMO проводит только микроскопические симуляции. Это метод математического моделирования, который используется для имитации поведения индивидуальных транспортных средств в реальном времени. Соответственно, каждый автомобиль моделируется как отдельный объект, со своими характеристиками, параметрами и маршрутом. Количество всех таких транспортных средств и описание из маршрутов, называют спросом на трафик. В контексте математического моделирования транспортных потоков, это один из основных параметров, которые используются для имитации дорожного движения в программных средствах, таких как SUMO.

Один из способов генерирования спроса на трафик является использование инструмента `jtrrouter`. Данное приложение предназначено для маршрутизации на основе объема потока и процентов поворота на перекрестках. Как и `netconvert`, оно работает, принимая на вход XML-файлы:

1) `test.flow.xml` — файл, содержащий информацию о транспортных потоках в симуляции SUMO, такую как идентификатор потока, его источник, время генерирования, вероятность появления транспортного средства в каждую симуляционную секунду или же точное количество транспортных средств в час, которое должно быть введено в систему;

2) `test.turn.xml` — файл, определяющий приоритетность поворотов на перекрестках в симуляции SUMO.

Помимо этого, необходимо передать файл, содержащий информацию о транспортной сети `test.net.xml`, описание которого представлено выше. `Jtrrouter` принимает эти файлы и генерирует единый файл `test.rou.xml` (рисунок 2.2), где описаны индивидуальные транспортные средства со своими характеристиками в виде идентификатора, маршрута и времени отправления.

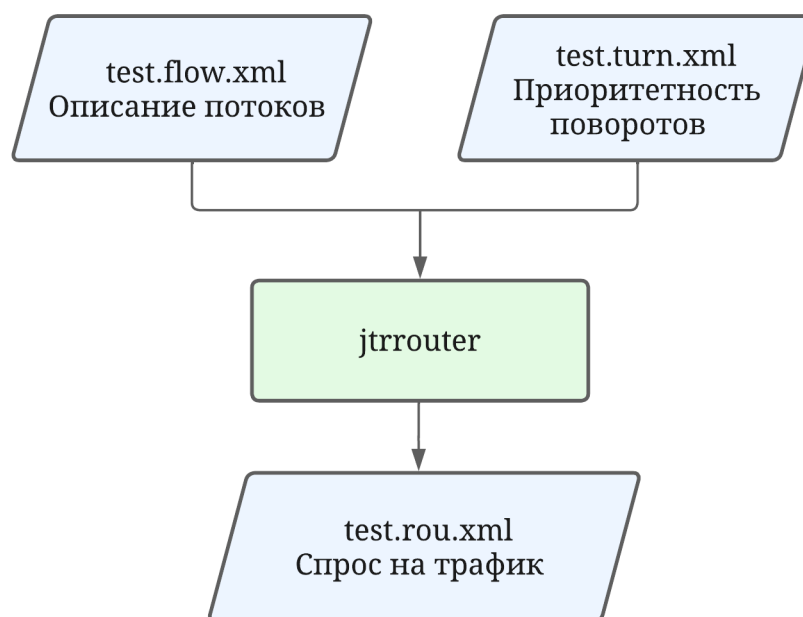


Рисунок 2.2 — Схема, отражающая принцип работы `jtrrouter`

2.4 Запуск симуляции через файл конфигурации `sumocfg`

Запуск симуляции осуществляется через создание файла конфигурации `sumocfg` и команды «`sumo-gui -c sumocfg`» [7]. Файл конфигурации SUMO, также известный как `sumocfg`, представляет собой XML-файл, используемый для

определения параметров симуляции в программе SUMO. Помимо этого, файл он позволяет настраивать различные аспекты симуляции, чтобы создать реалистичную модель дорожного движения, анализировать потоки транспортных средств, оптимизировать управление светофорами и проводить эксперименты.

Файл конфигурации `sumocfg` содержит ряд входных параметров. Например:

- 1) путь к файлу дорожной сети (`test.net.xml`), определяющему узлы и дороги в симуляции;
- 2) путь к файлу маршрутов (`test.rou.xml`) с информацией о планах движения для транспортных средств;
- 3) путь к файлу с дополнительными настройками (`test.add.xml`), в котором зачастую настраиваются фазы светофорного плана;
- 4) путь к файлу настроек пользовательского интерфейса (`test.settings.xml`), определяющий внешний вид и функциональность графического интерфейса SUMO;
- 5) время начала симуляции в симуляционных секундах;
- 6) время окончания симуляции в симуляционных секундах.

Это лишь некоторые из входных параметров, которые можно настроить в файле `sumocfg`. В зависимости от конкретной задачи и настроек симуляции, могут быть добавлены и другие, позволяющие детально настроить модель дорожного движения и управление транспортными потоками. Но для создания симуляционной модели двух перекрестков будет достаточно только этих входных данных.

Схема создания симуляционной модели SUMO проиллюстрирована на рисунке 2.3.

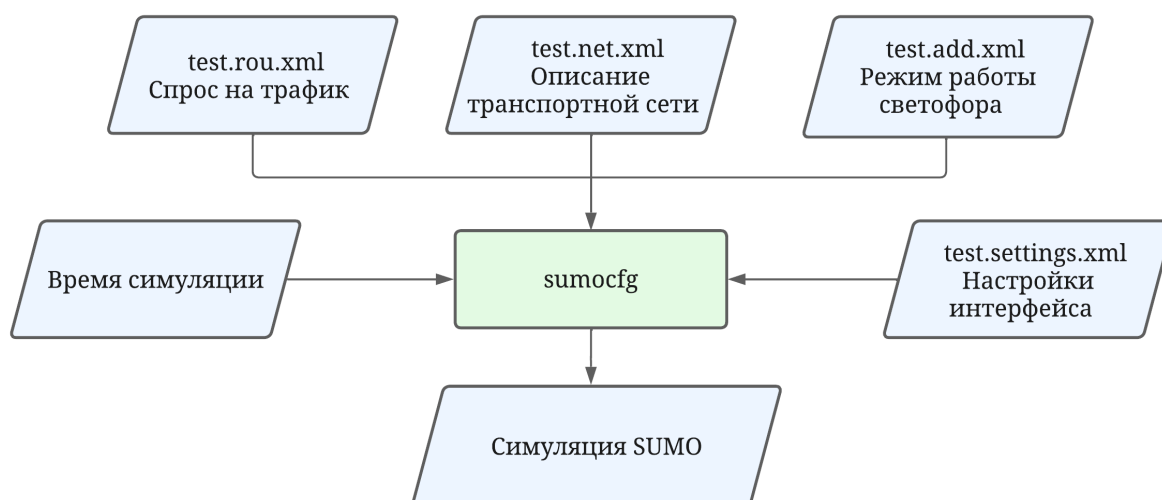


Рисунок 2.3 — Схема создания симуляции в SUMO

Пример транспортной сети SUMO, состоящей из двух перекрестков представлен на рисунке 2.4.

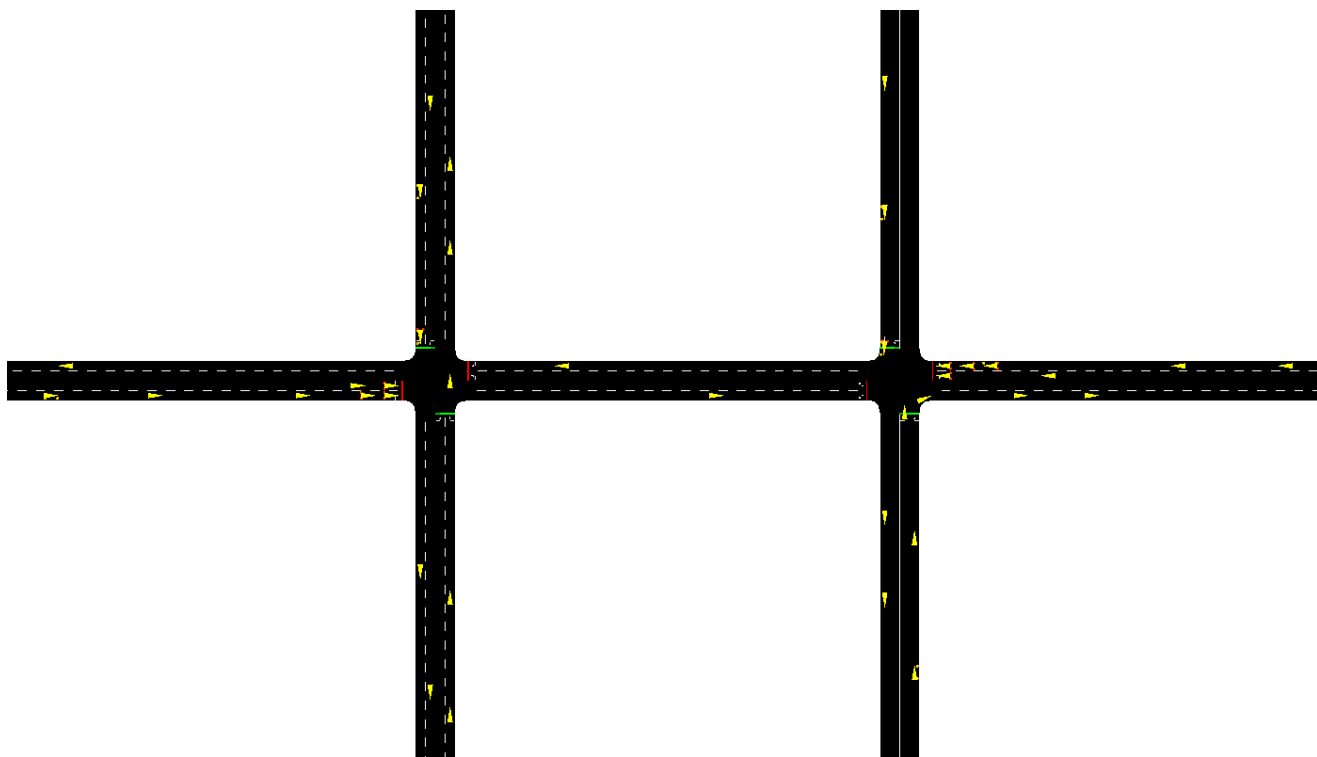


Рисунок 2.4 —Транспортная сеть, сгенерированная SUMO

При подключении файла `test.settings.xml`, где произведены настройки пользовательского интерфейса можно придать сети более реалистичный вид. Пример продемонстрирован на рисунке 2.5.

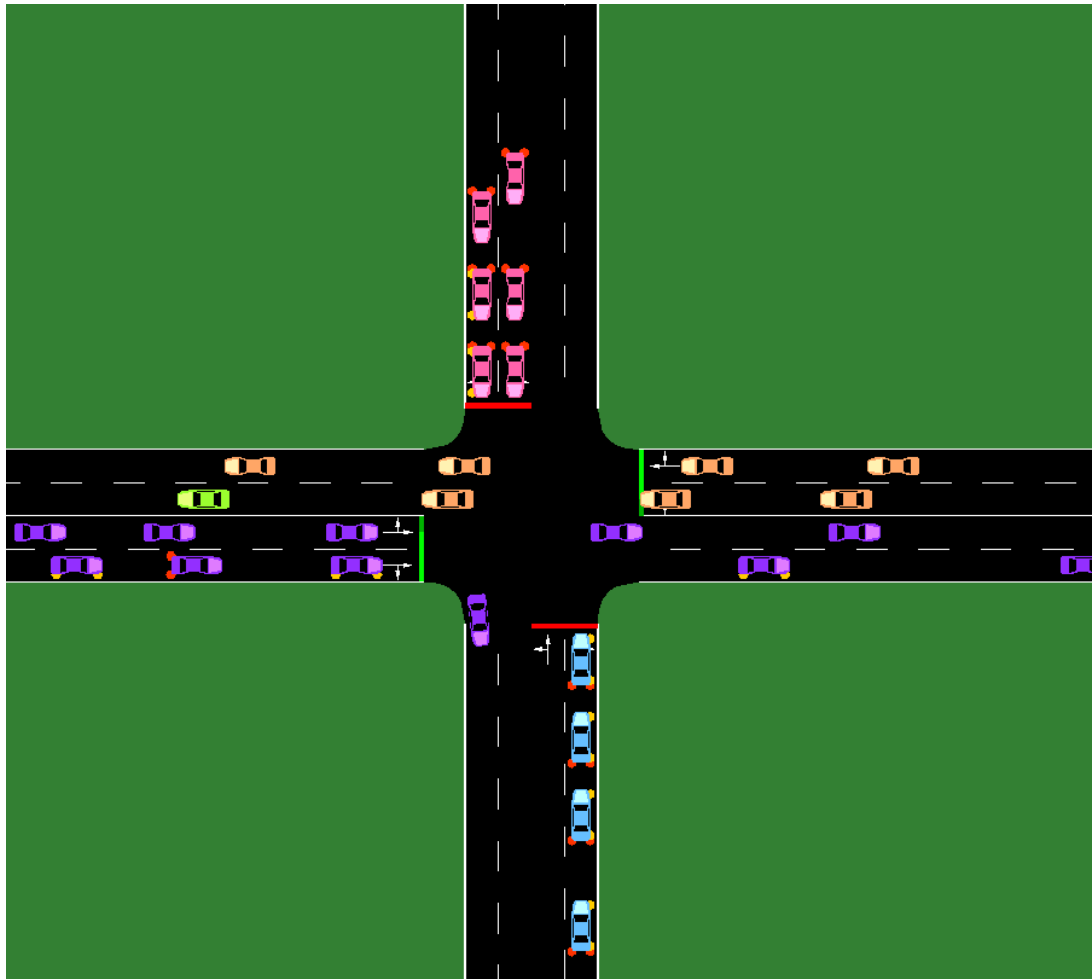


Рисунок 2.5 — Часть транспортной сети при подключении настроек интерфейса

2.5 Взаимодействие с симуляцией при помощи TraCI

Очевидно, что создание одной только симуляционной модели недостаточно для проведения обучения. Необходим инструмент, который позволяет получать внешней программе, где реализован алгоритм машинного обучения, информацию о текущем состоянии симуляции и воздействовать на параметры транспортной сети. Для этих целей существует TraCI.

TraCI — это интерфейс управления трафиком в программе SUMO [8]. Он предоставляет возможность взаимодействия с симуляцией транспортного потока в режиме реального времени.

С помощью TraCI можно получать информацию о различных объектах SUMO, таких как транспортные средства, светофоры, участки дорог и полосы движения, зоны контроля или детекторы. При этом можно воздействовать на их параметры. Например, изменять скорость, маршрут, ускорение транспортных средств, настраивать фазы светофора, управлять полосами движения и прочее.

Использование всех возможностей TraCI возможно после запуска через него симуляционной модели. Для этого необходимо прописать дополнительный входной параметр в конфигурационный файл `sumocfg` под названием `traci_server`, и внести туда значение удаленного порта. Далее требуется скрипт для запуска симуляции на любом из поддерживаемых TraCI языков программирования (например, Python, Java, C++, C) [8]. В данной работе использовался Python.

После установки и импорта модуля `TraCI` в скрипте прописывается путь до конфигурационного файла и к исполняемому файлу `sumo-gui.exe`, задается порт, значение которого такое же, что и в файле `sumocfg`, и команда для начала симуляции `traci.start()`. На этом этапе открывается окно графического интерфейса SUMO. Это значит, что запуск модели через TraCI прошел успешно. Для начала симуляции задается цикл, в котором число итераций равняется времени симуляции, и внутри него выполнять симуляционные шаги с помощью `traci.simulationStep()` [10]. Таким образом, на любой секунде через цикл и дополнительные команды TraCI можно выгружать любые данные по транспортной модели в режиме реального времени.

Пример реализации скрипта TraCI для получения данных транспортной сети представлен в Приложении А.

3 Обучение с помощью алгоритма Q-learning

3.1 Принцип работы обучения с подкреплением

В задаче управления светофорами выбор метода машинного обучения играет важную роль, так как он должен обеспечивать достижение максимальной производительности системы и учитывать ее динамический характер. Для этих целей подходит обучение с подкреплением, преимущество которого является взаимодействие со средой и получении награды за действия. Конечной задачей алгоритма является максимизация суммарного вознаграждения.

Основные термины для описания алгоритма [15]:

- 1) агент — система, которая выполняет действия для получения награды;
- 2) среда — окружение, в котором действует агент;
- 3) награда (r) — некоторое значение, которое агент получает от среды в ответ на определенное действие;
- 4) действие (a) — инструкция, передаваемая агентом в среду;
- 5) состояние (s) — описание текущего положения среды;
- 6) политика (π) — стратегия, используемая агентов при операции выбора действия.

Процесс обучения с подкреплением можно представить в виде цикла. На каждой итерации агент выполняет действие, которое максимизирует его ожидаемую награду. Затем он получает обратную связь в виде награды и обновляет свое состояние. После этого все шаги повторяются: агент выбирает новые действия и получает новые награды, в процессе взаимодействия со средой [15].

Данный цикл представлен на рисунке 3.1:

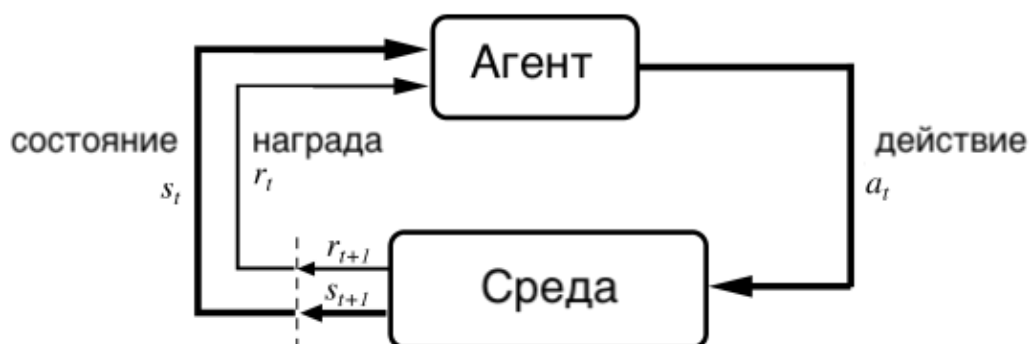


Рисунок 3.1 — Принцип работы обучения с подкреплением

Существует две категории алгоритмов обучения с подкреплением: базирующиеся на моделях и безмодельные. Модель в данном случае служит для предсказания динамики среды, то есть она используется для определения вероятности перехода пары «состояние-действие» в следующее состояние. По этой

причине алгоритмы, базирующиеся на моделях, могут более точно прогнозировать будущие награды. Безмодельные же алгоритмы используют опыт и целевую функцию для оценки действий в каждом состоянии, а не вероятности. В результате им требуется меньше вычислительных ресурсов, и обучение происходит быстрее. Это является большим преимуществом при увеличении пространства действий и состояний. Данный факт был важен для выбора алгоритма обучения, применяемого в работе [15].

Одним из популярных безмодельных алгоритмов обучения с подкреплением является Q-learning. Он не использует политику и считает, что все действия способны привести к оптимальному решению. Это снимает ограничение на операцию выбора и дает возможность системе производить более самостоятельное исследование окружения. Именно по этой причине в задаче управления светофорами был выбран табличный Q-learning. Его работа основана на правиле обновления функции $Q(s, a)$, которая оценивает ожидаемую сумму вознаграждений, получаемых агентом при выборе действия a в состоянии s [11]:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right] \quad (3.1)$$

где α — коэффициент скорости обучения ($0 < \alpha \leq 1$);

r — вознаграждение, получаемое агентом при переходе из состояния s_t в состояние s_{t+1} с помощью действия a_t ;

γ — дисконтирующий фактор ($0 < \gamma \leq 1$).

Выбор скорости обучения, как и дисконтирующего фактора, в алгоритме Q-learning является важным аспектом, который может влиять на эффективность обучения. Скорость обучения определяет, насколько быстро агент будет обновлять значения в своей Q-таблице на основе новой информации. Если скорость обучения слишком высока, агент может быстро сходиться к определенным значениям, но может иметь трудности в адаптации к новым ситуациям. Если скорость обучения слишком низкая, обучение может занимать слишком много времени, прежде чем агент достигнет оптимальной стратегии. Обычно начинают с относительно высокой скорости обучения и постепенно уменьшают ее по мере продвижения в обучении.

Дисконтирующий фактор определяет важность будущих вознаграждений по сравнению с немедленными вознаграждениями. Он влияет на то, как агент будет учитывать будущие выгоды при принятии решений. Значение дисконтирующего фактора должно быть в диапазоне от 0 до 1. При дисконтирующем факторе, близком к 0, агент будет уделять большее внимание немедленным вознаграждениям, игнорируя будущие выгоды. И наоборот, при дисконтирующем факторе, близком к 1, агент будет более долгосрочно ориентирован, принимая во внимание будущие вознаграждения.

В реализации алгоритма Q-learning функция $Q(s, a)$ представлена в виде таблицы. Она предназначена для хранения оценок ценности для всех возможных пар «состояние-действие». Это значит, что Q-значение в ячейке представляет собой оценку того, насколько успешно данное действие в определенном состоянии.

Изначально Q-таблица инициализируется случайными значениями или нулями. В процессе обучения агент взаимодействует со средой, получая награды, на основе которых происходит обновление таблицы, и учится различать более оптимальные действия.

На каждой итерации выбор действия происходит с использованием ϵ -жадной стратегии. При этой стратегии агент с вероятностью ϵ выбирает случайное действие и с остальной вероятностью выбирает действие, имеющее наибольшее Q-значение для текущего состояния. Это позволяет агенту балансировать между исследованием новых действий и использованием уже изученных [2].

Постепенно после определенного числа эпох система находит оптимальные действия для каждого состояния и Q-таблица сходится к оптимальным Q-значениям.

3.2 Применение Q-learning в управления светофорами

В данном исследовании Q-learning будет применен для определения адаптивного светофорного плана на транспортной сети, состоящей из двух перекрестков. Таким образом, сигнальный контроллер рассматривается как агент. При этом только один светофор является адаптивным, второй же остается фиксированным с симметричным сигнальным планом, длина зеленой фазы которого равна 24 секунды, а желтой — 4. Цель состоит в том, чтобы минимизировать общее количество машин на первом перекрестке и при этом сохранить баланс между всеми направлениями по числу транспортных средств.

Для применения Q-learning в управления светофорами необходимо определить среду, пространство действий и состояний, функцию награды.

В качестве среды выступит симуляционная модель SUMO. С помощью библиотеки TraCI («Интерфейс управления дорожным движением») будет получен доступ к моделируемому объекту в режиме реального времени. Таким образом агент сможет получать необходимые данные о транспортной сети и воздействовать на нее.

Для определения награды и пространства состояний из среды каждый раз будет выгружаться информация об общем количестве транспортных средств в четырех направлениях на перекрестке с адаптивным светофорным планом: NS , SN , WE , EW :

- 1) NS_t — количество транспортных средств в направлении Север-Юг;
- 2) SN_t — количество транспортных средств в направлении Юг-Север;

- 3) WE_t — количество транспортных средств в направлении Запад-Восток;
 4) EW_t — количество транспортных средств в направлении Восток-Запад.

Каждый из четырех компонентов классифицирован по трем категориям, в зависимости от количества транспортных средств, учитывая, что максимально ребро длиной 150 метров вмещает 40 легковых автомобилей (таблица 3.1).

Таблица 3.1 — Классификация полученных параметров по числу транспортных средств

Категория	Количество транспортных средств, n
L	$0 \leq n \leq 13$
M	$13 < n \leq 26$
H	$n > 26$

Таким образом, элемент пространства состояний можно представить в виде [12]:

$$s_t = (NS'_t, SN'_t, WE'_t, EW'_t) \quad (3.2)$$

где $NS'_t, SN'_t, WE'_t, EW'_t$ — классифицированное количество легковых автомобилей по четырем направлениям.

Размерность пространства состояний составляет 3^4 или 81. Пример элементов из этого пространства: LLLL, LMHN, HLML.

Следующим этапом является определение пространства действий, которое связано непосредственно со светофорным планом.

Сигнальный план в SUMO включает в себя следующие компоненты:

- 1) зеленый сигнал в направлении Север-Юг и Юг-Север – Красный сигнал в направлении Запад-Восток и Восток-Запад;
- 2) желтый сигнал;
- 3) зеленый сигнал в направлении Запад-Восток и Восток-Запад – Красный сигнал в направлении Север-Юг и Юг-Север;
- 4) желтый.

Для упрощения задачи желтый сигнал при адаптивном светофоре будет фиксированным с продолжительностью 4 секунды. Все повороты во время этой фазы запрещены.

Агент воздействует непосредственно на зеленую фазу. Минимальная ее продолжительность — 12 секунд, максимальная — 60 секунд. Таким образом, элемент пространства действий состоит из двух компонентов: зеленая фаза в направлении Север-Юг ($grPhaseNS$) и зеленая фаза в направлении Запад-Восток

(*grPhaseWE*). В противоположных направлениях сигнальный план будет отражаться симметрично.

Можно представить элемент из пространства действий в виде [12]:

$$a = (grPhaseNS, grPhaseWE) \quad (3.3)$$

С шагом $dt = 6$ секунд генерируем все возможные продолжительности зеленых фаз светофоров для двух компонент от 12 до 60 секунд. Всего таких вариантов на одну компоненту — девять. Это значит, что размерность пространства действий 9^2 или 81. Пример элемента из этого пространства: (12,12), (12,24), (60,42).

Как уже упоминалось выше, целью обучения является создание адаптивного плана с помощью, которого будет минимизировано общее количество машин и при этом сохранен баланс по числу автомобилей между всеми направлениями перекрестка. По этой причине функция наград должна предусматривать наказания в случае, если данные условия после выполнения действия агентом нарушаются.

При переходе из одного состояние в другое:

$$s_t = (NS'_t, SN'_t, WE'_t, EW'_t) \rightarrow s_{t+1} = (NS'_{t+1}, SN'_{t+1}, WE'_{t+1}, EW'_{t+1}) \quad (3.4)$$

вознаграждение вычисляется по формуле [13]:

$$R(s_t, s_{t+1}) = \beta(\min(\mu - NS_{t+1}, \mu - SN_{t+1}, \mu - WE_{t+1}, \mu - EW_{t+1}) + (1 - \beta)(-max(NS_{t+1}, SN_{t+1}, WE_{t+1}, EW_{t+1}))) \quad (3.5)$$

$$\text{где } \mu = \frac{(NS_t + SN_t + WE_t + EW_t)}{4}; \quad (3.6)$$

β – коэффициент, контролирующий приоритетность условий ($0 < \beta < 1$).

В первой части уравнения дается информация о том, насколько система достигла равновесия. Вторая часть уравнения отвечает за сокращение общего количества транспортных средств по всем направлениям и накладывает штраф по наибольшему значению. Очевидно, что награда всегда будет равна отрицательному значению и система будет наказываться в любом случае, независимо от состояния и действия. Параметр β – это компромисс между сокращением числа автомобилей и условием равновесия. Так как оба условия в равной степени важны, значение этого коэффициента равно 0.5.

Исходя из размерности пространства состояний и пространства действий Q-таблица имеет размерность 81 столбец на 81 строку. Изначально она будет заполнена нулями.

Для проведения обучения необходимо задать параметры обучения: дисконтирующий фактор, скорость обучения, количество эпох обучения и значения ε . Эти данные представлены в таблице 3.1.

Таблица 3.1 — Параметры обучения

Название параметра	Обозначение	Значение
Скорость обучения	α	0.6
Дисконтирующий фактор	γ	0.8
Параметр жадной стратегии	ε	0.1
Периодичность выполнения действия в секундах	Δt	100
Общее время симуляции в секундах	t	35000
Количество эпох	n	300

Реализация алгоритма Q-learning на языке Python представлена в Приложении Б.

4 Эксперименты и результаты

4.1 Определение начального положения системы

Начальный светофорным план, с которым начинает работать симуляция, 18 секунд — зеленая фаза в направлении Север-Юг, 30 секунд — зеленая фаза в направлении Запад-Восток, и симметрично в противоположных направлениях. Данное начальное положение светофора было выбрано для наглядности работы алгоритма обучения. Однако, можно начать с любого светофорного плана и на сходимость метода Q-learning это не повлияет.

SUMO начинает генерировать машины, начиная с первой симуляционной секунды. Это значит, что машины копятя постепенно с течением времени и вводить агента с первых секунд будет бесполезно. По этой причине начальное состояние фиксируется на 100 секунде симуляции, когда на всех направлениях уже скопилось достаточное количество машин. С учетом описанного выше сигнального плана к этому времени уже образуется дорожный затор в направлении Север-Юг. С этого момента начинает работать агент и выполняет действие, приводящее к следующему состоянию, фиксирующемуся уже на 200 секунде. На основе этих данных считается награда по формуле (3.5) и обновляется Q-таблица по правилу, описанному в формуле (3.1).

Начальное состояние и начальный светофорный план генерируются значениями, описанными выше, в начале каждой новой эпохи. При этом Q-таблица сохраняет все свои уже записанные значения и продолжает заполняться новыми данными. Было проведено обучение на 300 эпохах и это заняло в общей сложности более 4 часов.

В конце обучения Q-таблица сохраняется в отдельный файл для использования его при тестировании.

Отличие тестовой программы от программы с обучением заключается в стратегии выбора действий: если во время обучения использовалась ϵ -жадная стратегия с показателем ϵ равным 0.1, то есть агент выполнял случайное действие с вероятностью 10% для изучения большего числа значений в таблице, то в тестовой программе случайные действия уже не выполняются, а выбираются только те, что приводят максимальному общему вознаграждению, так как необходимость в исследовании среды пропадает.

Задача тестовой программы заключается в сборе показаний транспортной сети при каждом вмешательстве агента, которые будут использованы для анализа производительности работы алгоритма и построения графиков. Эти данные включают: количество машин по четырем направлениям, общий список состояний, действий, выбранных агентом, и вознаграждений. Отдельно был написан скрипт для получение таких же данных, за исключением вознаграждений, для фиксированных светофорных планов в целях сравнения эффективности.

4.2 Поиск адаптивного светофорного плана для одного перекрестка

Изначально для отладки работы алгоритма, было произведено обучение и поиск адаптивного светофорного плана транспортной сети, состоящей из одного перекрестка.

Согласно шагам, описанным в главе 1, была создана симуляционная модель одного перекрестка. Ее можно описать следующими характеристиками:

- 1) пять узлов;
- 2) на узлах с идентификатором «0» установлен светофор;
- 3) восемь однонаправленных ребер, длина каждого составляет 150 метров;
- 4) для упрощения модели исключены пешеходные переходы, автобусные полосы и велосипедные дорожки;
- 5) каждое ребро имеет по две полосы;
- 6) движение перекрестков двустороннее;
- 7) скорость транспортных средств не превышает 60 км/ч;
- 8) для упрощения модели генерируются только легковые автомобили;
- 9) время, в течение которого генерируются автомобили, составляет 35000 симуляционных секунд;
- 10) на перекрестке вероятность того, что автомобиль поедет прямо, составляет 0.7;
- 11) вероятность поворота направо — 0.3, повороты налево запрещены для упрощения обучения.

Схема одного перекрестка представлена на рисунке 4.1.

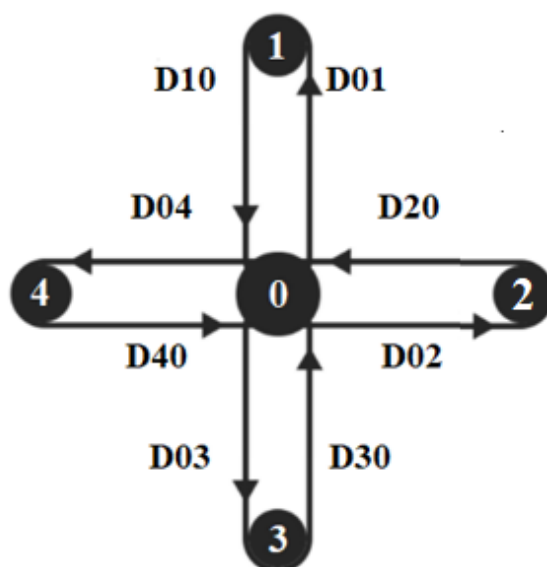


Рисунок 4.1 — Схема одного перекрестка

Сначала была попытка провести обучение на симуляционной модели с вероятностным потоком. В `test.flow.xml` для каждого ребра была задана вероятность появления транспортного средства в каждую симуляционную секунду. Это значение для всех ребер равнялось 0.3. Но в таком случае Q-learning плохо сходится, так как система находится в нестабильном состоянии и после выполнения в одном состоянии определенного действия, симуляционная модель может выдать разные следующие состояния, вследствие появления новых машин. Таким образом, получаются совершенно разные значения для одной и той же ячейки в Q-таблицы и затрудняется поиск оптимального светофорного плана [12]. По этой причине все дальнейшие эксперименты были проведены в фиксированном потоке: транспортное средство появляется на каждом из входных ребер раз в 3 секунды.

В результате для одного перекрестка был получен оптимальный светофорный план, минимизирующий общее количество автомобилей на каждом ребре и сохраняющий баланс между всеми направлениями по числу транспортных средств. Данный план составляет 24 секунды зеленой фазы во всех направлениях.

На рисунке 4.2 и 4.3 представлены графики изменения числа автомобилей в разных направлениях в зависимости от времени при адаптивном светофорном плане.

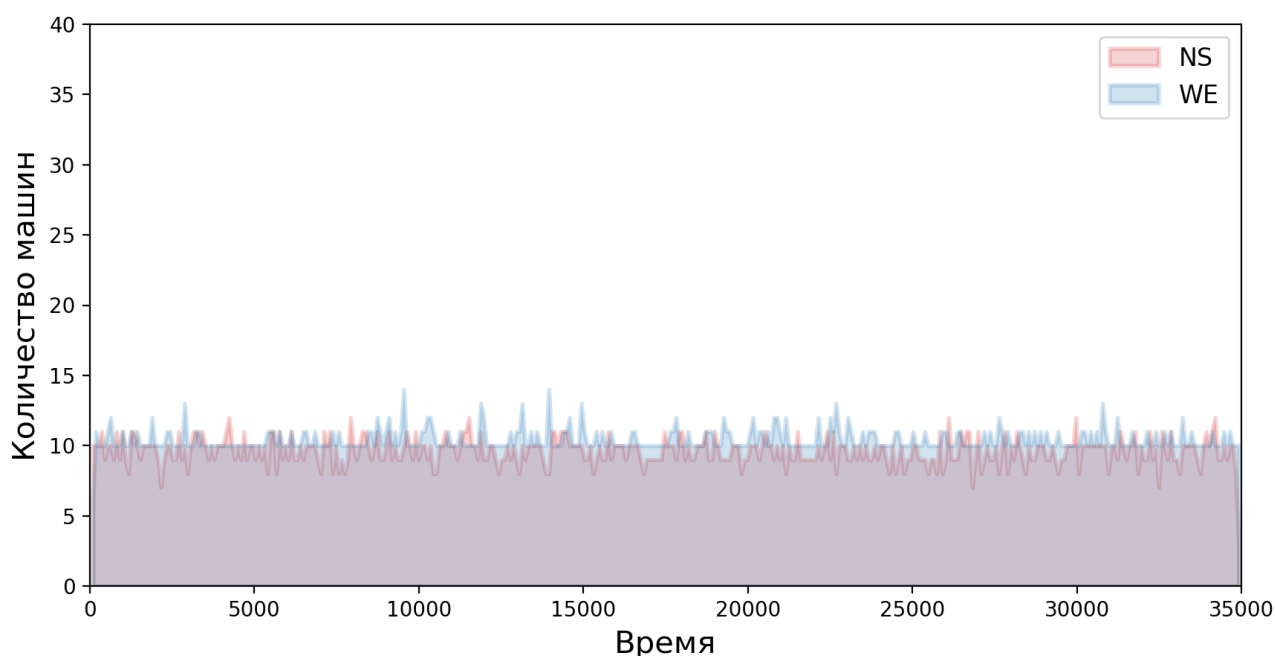


Рисунок 4.2 — Количество транспортных средств при адаптивном светофорном плане в направлении Север-Юг и Запад-Восток

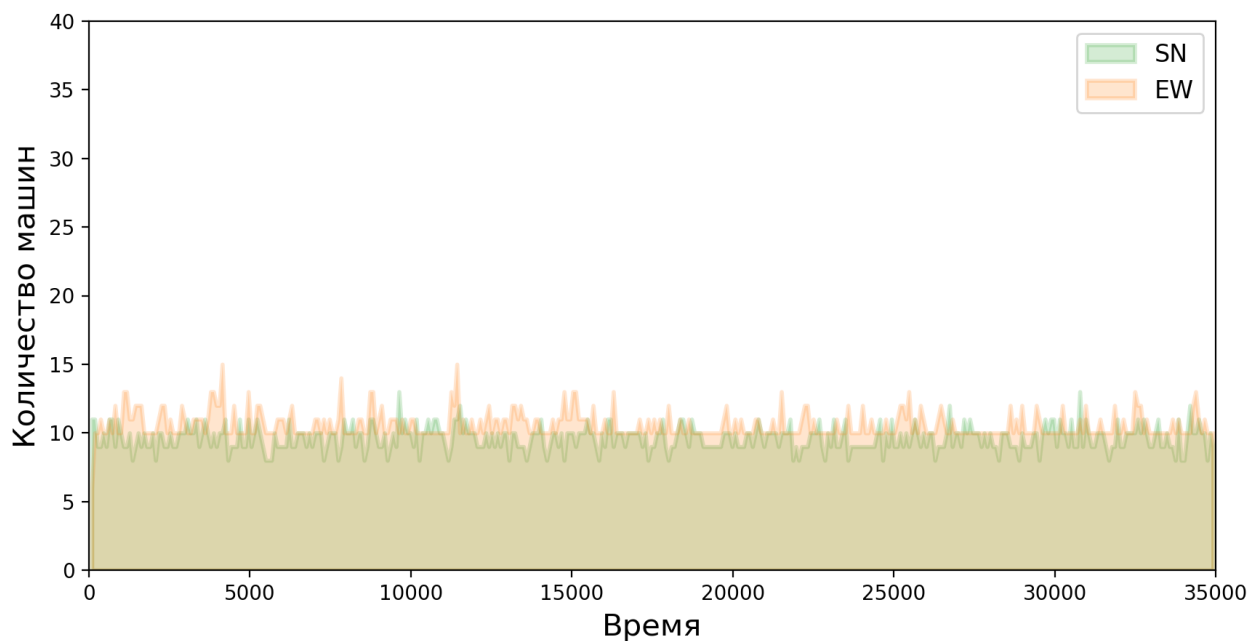


Рисунок 4.3 — Количество транспортных средств при адаптивном светофорном плане в направлении Юг-Север и Восток-Запад

Исходя из этих двух графиков несмотря на то, что есть колебания в количестве машин, по всем четырем направлениям транспортные средства распределены равномерно и количество машин на каждом ребре большую часть симуляции составляет меньше 13, что относится к низкому количеству, согласно категориям из таблицы 3.1.

Помимо этого, можно оценить изменение функции наград на протяжении всего времени симуляции на рисунке 4.4

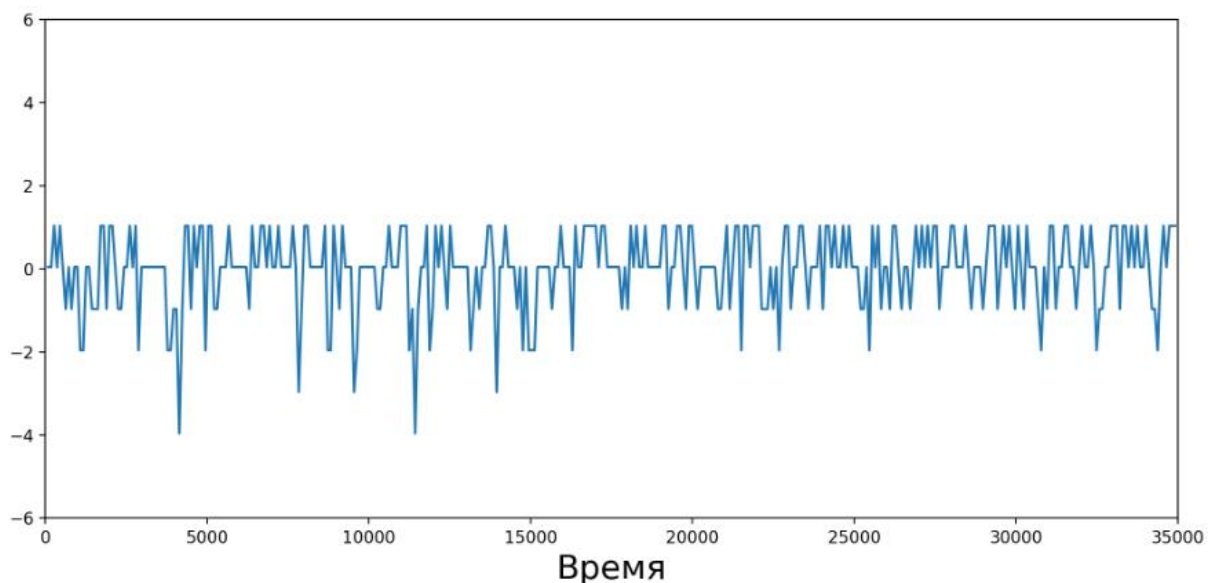


Рисунок 4.4 — Отклонение награды от среднего значения

На данном графике видно, что отклонение награды от среднего значения несущественно, за исключением некоторых точек. Так что можно говорить о том, что алгоритм сходится.

4.3 Поиск адаптивного светофорного плана для двух перекрестков

Для двух перекрестков характеристиками транспортной сети SUMO следующие:

- 1) восемь узлов;
- 2) на узлах с идентификаторами «0» и «2» установлены светофоры;
- 3) четырнадцать однонаправленных ребер, длина каждого составляет 150 метров;
- 4) для упрощения модели исключены пешеходные переходы, автобусные полосы и велосипедные дорожки;
- 5) каждое ребро имеет по две полосы;
- 6) движение перекрестков двустороннее;
- 7) скорость не превышает 60 км/ч.

Схема транспортной сети с нумерацией узлов и названиями ребер представлена ниже (рисунок 4.5).

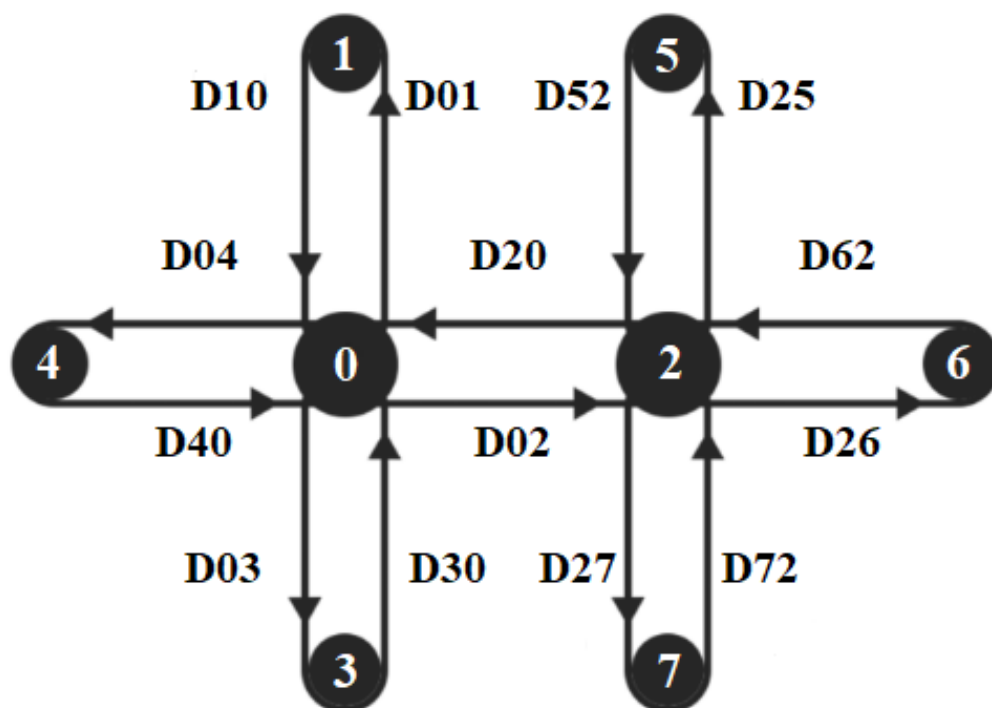


Рисунок 4.5 – Схема двух перекрестков

Характеристики спроса на трафик:

- 1) для упрощения модели генерируются только легковые автомобили;
- 2) время, в течение которого генерируются автомобили, составляет 35000 симуляционных секунд;
- 3) на перекрестке вероятность того, что автомобиль поедет прямо, составляет 0.7;
- 4) вероятность поворота направо — 0.3, повороты налево запрещены для упрощения обучения.

Код всех используемых XML-файлов для создания двух перекрестков представлен в Приложении В.

В случае двух перекрестков, как уже говорилось выше, только один светофорный план будет адаптивным, второй — фиксированным. Как и с одним перекрестком при вероятностном потоке алгоритм не сходится из-за нестабильного состояния системы. По этой причине поток, используемый в обучении, будет фиксированным: генерируются 3 транспортных средства в секунду по трем на всех входных узлах. Проблема возникает только на общем ребре (D20). Так как повороты на первом и втором перекрестке транспортной сети задаются через вероятности, на данное ребро подается случайный поток. Это вызывает ту же проблему, что была описана выше в случае с одним перекрестком: алгоритм постоянно переобучается. Цель данного этапа работы: провести обучение на такой симуляционной модели и оценить работу агента.

После обучения в процессе тестирования был получен адаптивный светофорный план. Независимо от начального состояния системы, большую часть времени использовался тот же сигнальный план, что и в случае с одним перекрестком, а именно составляет 24 секунды зеленой фазы во всех направлениях. Процентное соотношение разных типов сигнальных планов представлено в таблице 4.1.

Таблица 4.1 — Соотношение используемых агентом сигнальных планов

Зеленая фаза в направлении Север-Юг и Юг-Север, в секундах	Зеленая фаза в направлении Восток-Запад и Запад-Восток, в секундах	Частота использования, %
24	24	90.5
42	36	4.9
18	36	4.6

На рисунке 4.6 и 4.7 представлены графики изменения числа автомобилей в разных направлениях в зависимости от времени при адаптивном светофорном плане после проведения обучения.

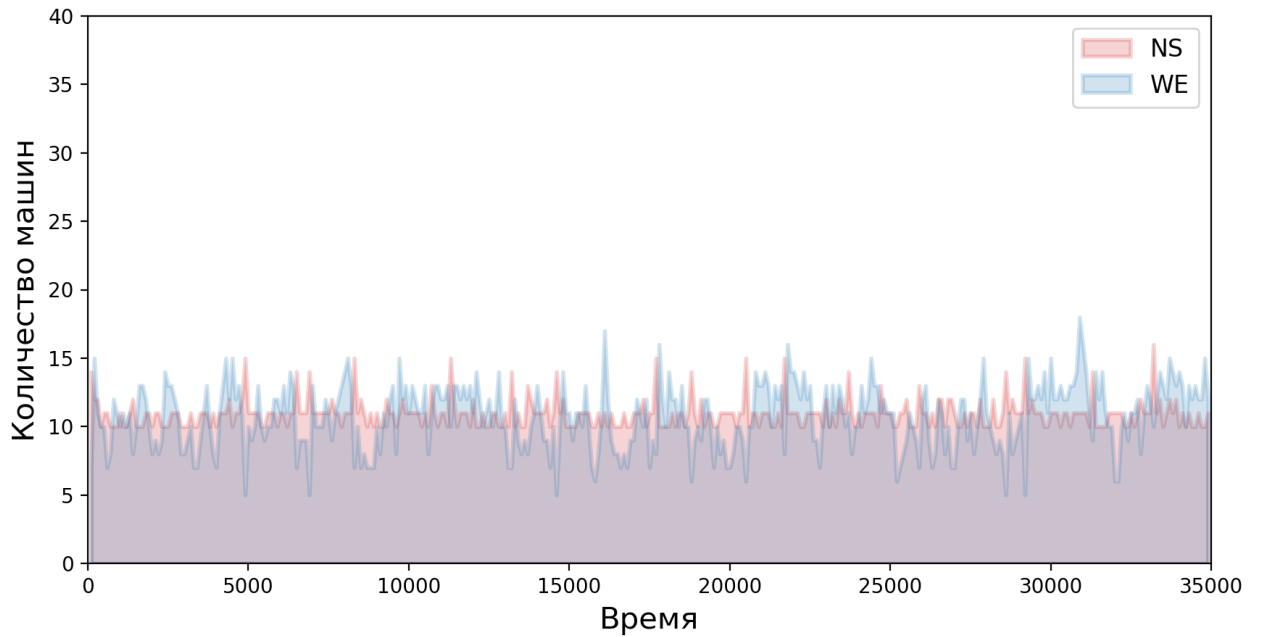


Рисунок 4.6 — Количество транспортных средств при адаптивном светофорном плане в направлении Север-Юг и Запад-Восток

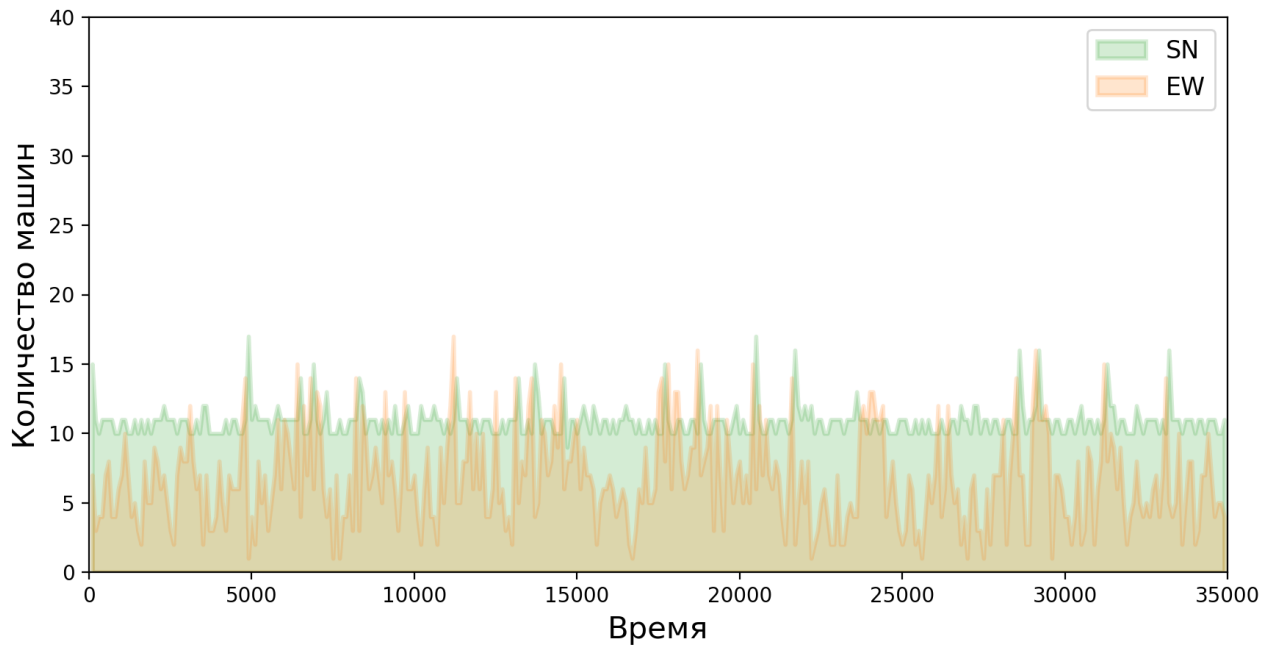


Рисунок 4.7 — Количество транспортных средств при адаптивном светофорном плане в направлении Юг-Север и Восток-Запад

По графикам видно, что в двух независимых направлениях количество машин стабилизировалось и колеблется около 10. Проблема возникает в направлениях Запад-Восток и Восток-Запад. При этом, разброс машин по общему ребру намного больше, чем по свободному. Это как раз и вызвано тем, что на одной стороне перекрестка задан вероятностный поток и агент не может к нему адаптироваться.

График отклонения награды от среднего значения тоже показывает нестабильное состояние (рисунок 4.8)

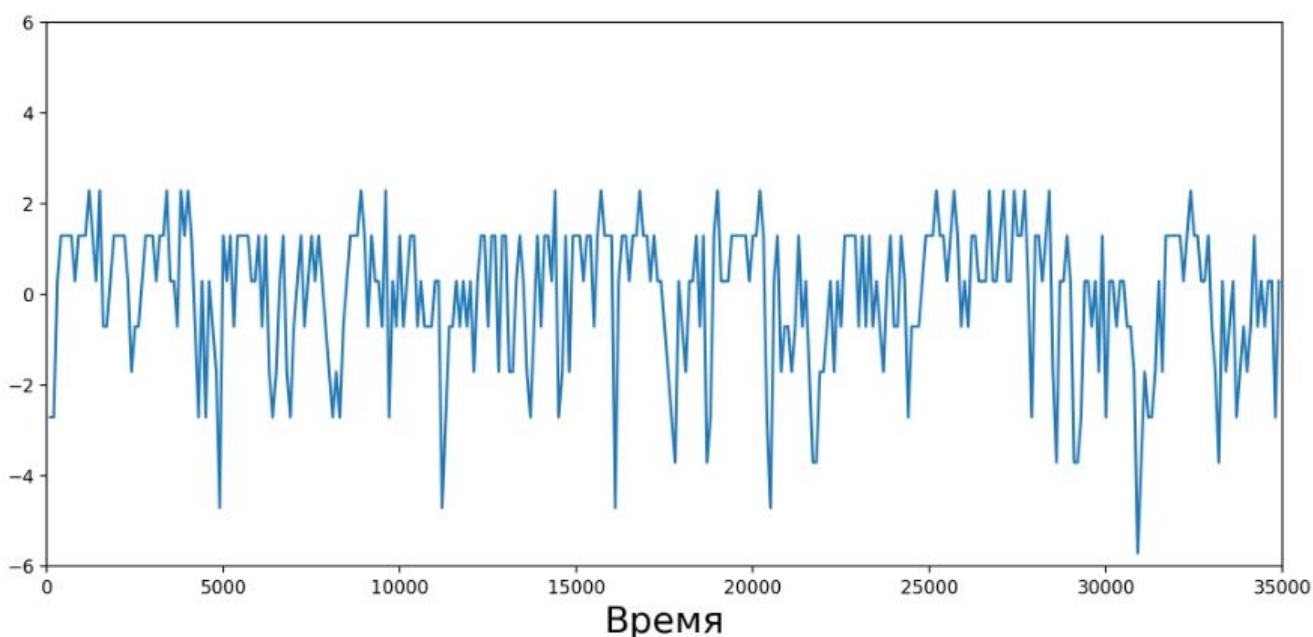


Рисунок 4.8 — Отклонение награды от среднего значения

Награда колеблется около среднего значения, но есть случаи, когда отклонения существенны. Как уже было сказано, на общем ребре поток задается вероятностями поворота транспортных средств, но ситуацию усложняет и то, что агент вмешивается в разные фазы второго светофора. Всё это приводит к ситуации, когда на ребрах D20 и D40 появляется дополнительное количество автомобилей и агент начинает перебирать другие светофорные планы прежде, чем опять вернуться к оптимальному значению.

4.4 Сравнение адаптивного светофорного плана с фиксированными

Для оценки работы адаптивного светофорного плана сравним его с фиксированными, длины зеленых фаз которых составляют 18 и 30 секунд во всех

направлениях. Была запущена симуляция через TraCI, выгружены все необходимые данные в виде csv-таблиц и построены графики.

Ниже представлены графики изменения числа автомобилей с течением времени при фиксированных сигнальных планах (рисунки 4.9, 4.10, 4.11, 4.12).

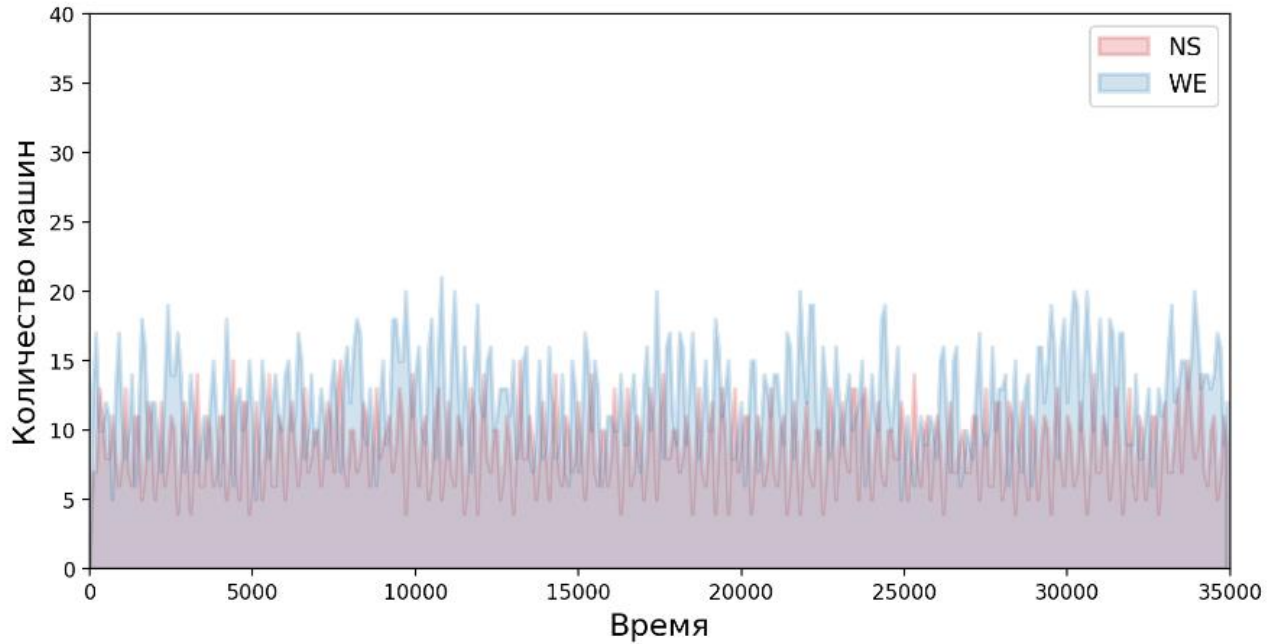


Рисунок 4.9 — Количество транспортных средств при адаптивном светофорном плане в направлении Север-Юг и Запад-Восток с планом 18 секунд

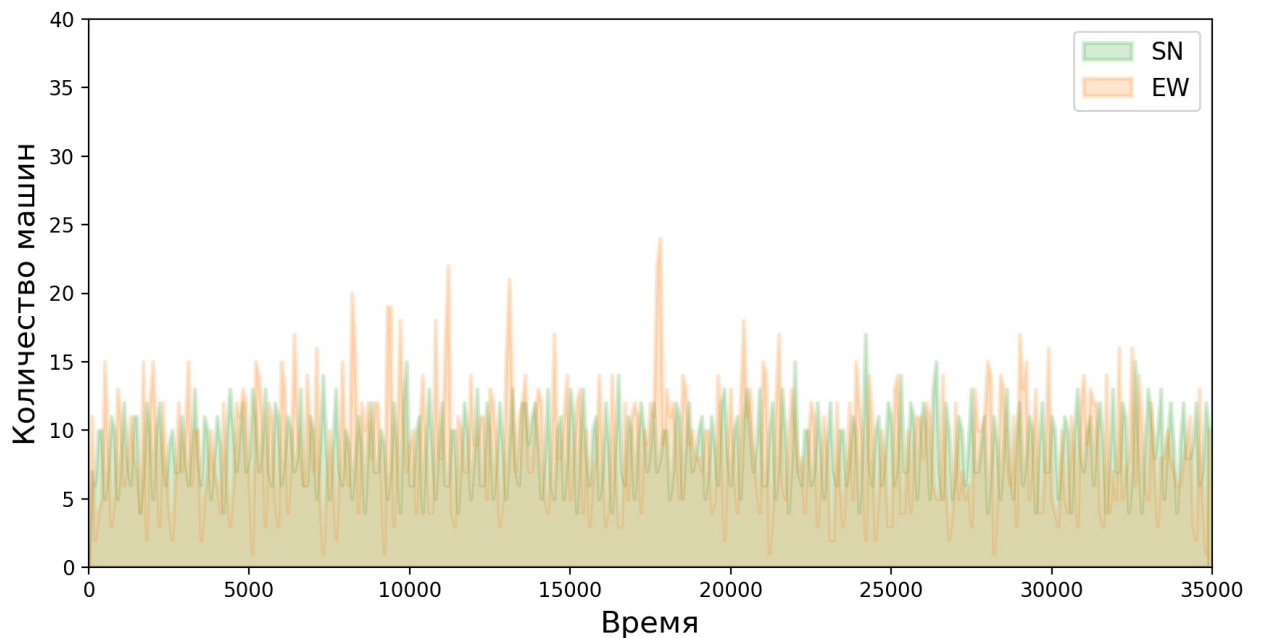


Рисунок — 4.10 Количество транспортных средств при адаптивном светофорном плане в направлении Юг-Север и Восток-Запад с планом 18 секунд

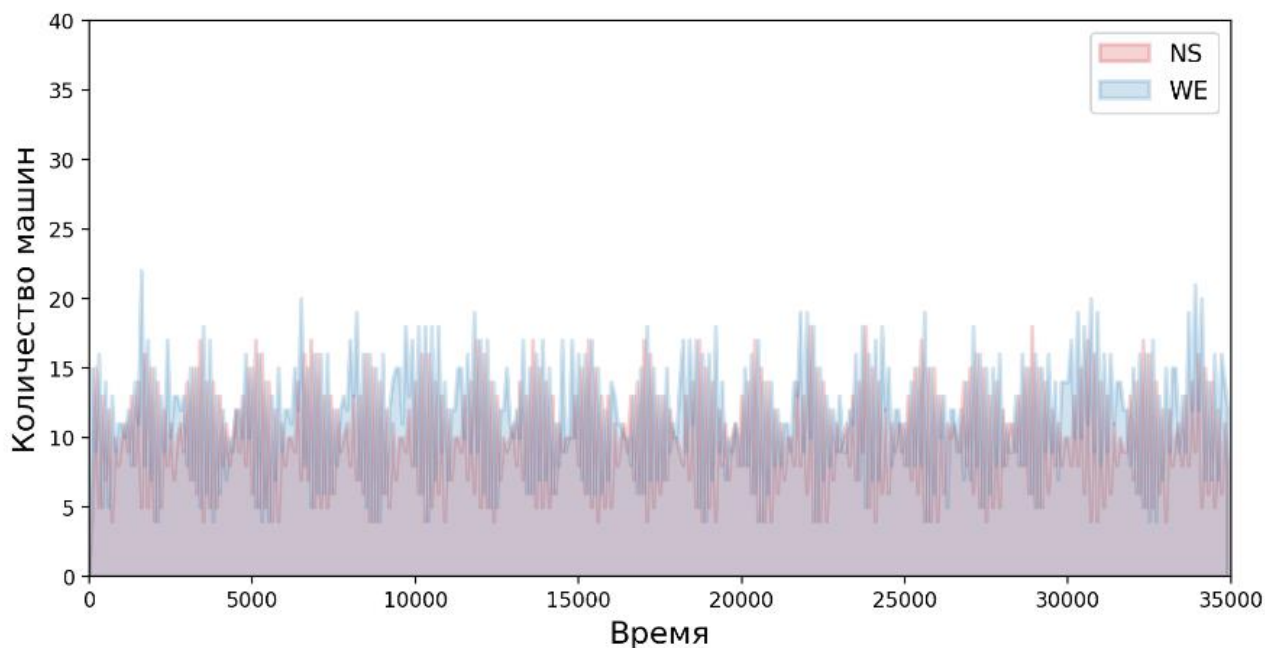


Рисунок 4.11 — Количество транспортных средств при адаптивном светофорном плане в направлении Север-Юг и Запад-Восток с планом 30 секунд

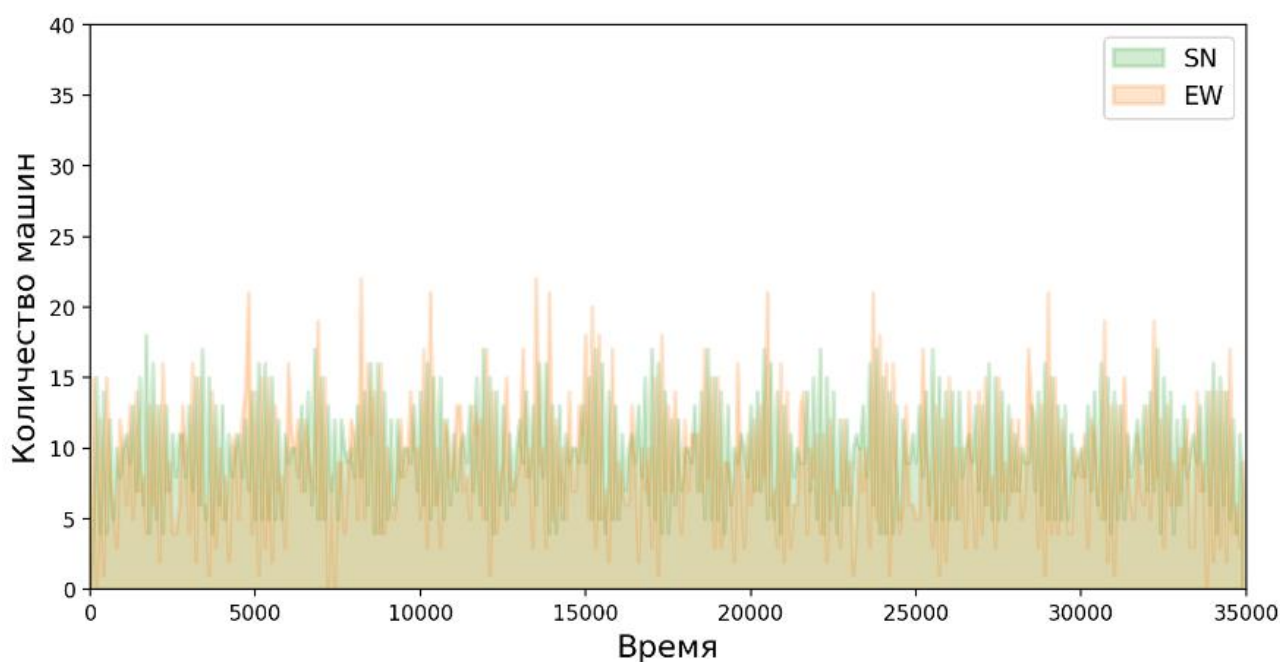


Рисунок 4.12 — Количество транспортных средств при адаптивном светофорном плане в направлении Юг-Север и Восток-Запад с планом 18 секунд

На графиках, представленных выше, видны сильные колебания по числу автомобилей во всех направлениях. Очевидно, что фиксированные светофорные планы не справляются с задачей сохранения баланса по числу транспортных средств во всех направлениях.

Проведем количественную оценку возникновения состояний (рисунок 4.13). Отметим, что при минимизации количества автомобилей по всем направлениям, состояние должно принимать значение LLLL.

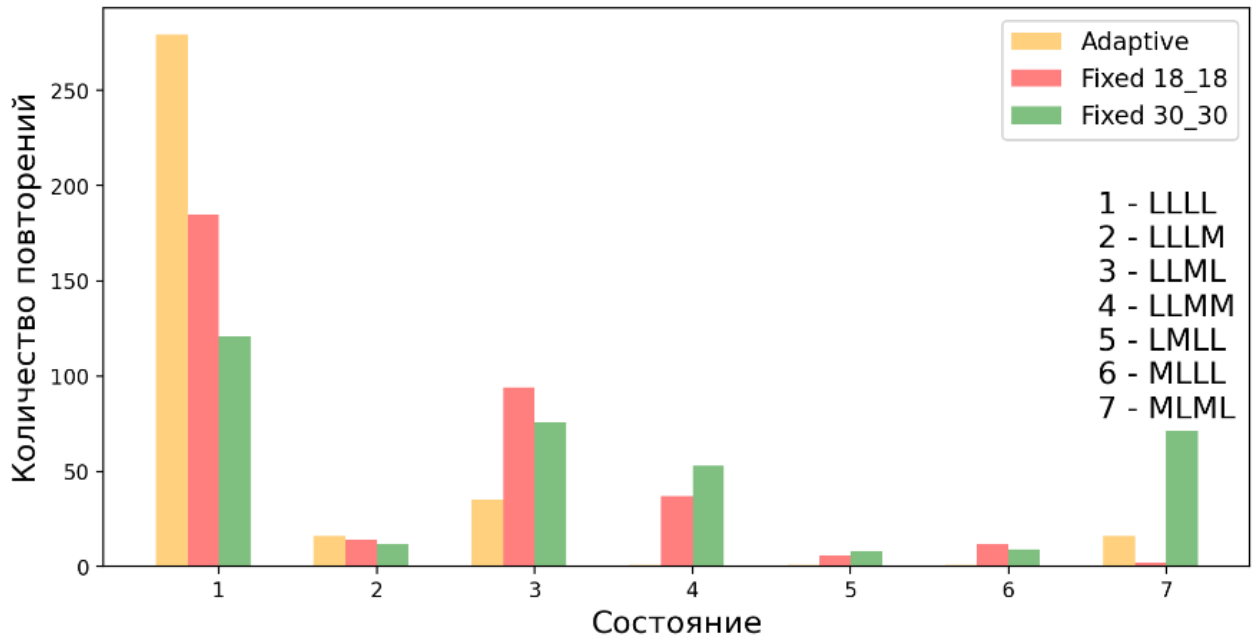


Рисунок 4.13 — Количественная оценка возникновения состояний

В состоянии LLLL чаще всего пребывает система с адаптивным светофорным планом. Фиксированные планы показывают результат ниже. Если говорить о процентном соотношении, то адаптивный план приводит к оптимальному значению (состоянию LLLL) в 80 процентах случаев, фиксированный с зеленой фазой 18 секунд – в 53 процентах, а с сигнальным планом 30 секунд – в 34 процентах.

Исходя из всех выше представленных графиков, можно судить о том, что адаптивный светофорный план работает эффективнее фиксированных по критерию минимизации машин на ребрах перекрестка и приводит систему в более равновесное состояние по количеству машин, хотя и имеет некоторые колебания на рисунке 4.6 и 4.7.

ЗАКЛЮЧЕНИЕ

В данной дипломной работе было рассмотрено математическое моделирование транспортных потоков на примере управления светофорами. Основная цель заключалась в проведении адаптивного управления для одного из двух перекрестков транспортной сети в целях оптимизации транспортных потоков и оценке его эффективности.

В качестве инструмента для поиска светофорного плана выступал алгоритм обучения с подкреплением – Q-learning, реализованный на языке программирования Python, а средой обучения стала симуляционная модель, созданная через программное обеспечение SUMO.

В первой главе были описаны подходы и методы математического моделирования транспортных потоков, разобрана математическая модель, используемая в программе SUMO.

Во второй главе представлены основные правила создания модели транспортной сети и спроса на трафик в SUMO.

В третьей главе рассматривались теория по обучению с подкреплением, особенности алгоритма Q-learning и необходимые для задачи пространства действий, состояний и функция награды, подобраны параметры обучения.

В четвертой главе содержалась информация о проведении экспериментов сначала на транспортной сети, состоящей из одного перекрестка, а затем из двух и было произведено сравнение работы адаптивного светофорного плана с фиксированными на двух перекрестках по двух критериям:

- 1) сохранение баланса между ребрами перекрестка по числу автомобилей;
- 2) снижение числа автомобилей на ребрах перекрестка.

В результате проведенных экспериментов в программе SUMO выяснилось, что адаптивный светофорный план на двух перекрестках оказался более производительным, по сравнению с фиксированными. Он эффективно справляется с задачей уменьшения числа автомобилей на ребрах перекрестка, но у этого метода есть ограничение, связанное с равномерным распределением транспортных средств. Причиной этому является наличие случайного потока на общем ребре в транспортной сети двух перекрестков, связанное с поворотами на втором перекрестке, которые задаются через вероятности. Q-learning плохо адаптируется в таких случаях и происходит постоянное переобучение агента, которое мешает сохранению баланса по числу автомобилей между всеми ребрами. Но несмотря на это, алгоритм довольно близко приближается к оптимальной производительности.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 P. A. Lopez, Microscopic Traffic Simulation using SUMO // 21st International Conference on Intelligent Transportation Systems (ITSC). — 2018. — С. 2575-2582.
- 2 Материалы из интернета — Понимание Q-learning, проблема «Прогулка по скале». URL: <https://habr.com/ru/articles/443240/>.
- 3 D. Krajzewicz, G. Hertkorn and P. Wagner, SUMO (Simulation of Urban MObility) // 2002 4th Middle East Symposium on Simulation and Modelling.
- 4 Семенов В.В. Математическое моделирование динамики транспортных потоков мегаполиса, 2004, С. 10-20.
- 5 Материалы из интернета — Методы моделирования транспортного потока. URL: <https://fundamental-research.ru/ru/article/view?id=40856>.
- 6 Craig B. Rafter and Simon Box. Investigating the effects of mixed driver reaction times in the transport network // 2016 5th Symposium of the European Association for Research in Transportation.
- 7 Документация SUMO. URL: <https://sumo.dlr.de/docs/index.html>.
- 8 Документация TraCI. URL: <https://sumo.dlr.de/docs/TraCI>.
- 9 Документация Python. URL: <https://docs.python.org/3/>.
- 10 Материалы из интернета — TraCI — Traffic Control Interface — PyPI. URL: <https://pypi.org/project/traci/>.
- 11 Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction, 2017, С. 107.
- 12 Kurmankhojayev D., Suleymenov N., Tolebi G. Online model-free adaptive traffic signal controller or an isolated intersection // 2017 International MultiConference on Engineering, Computer and Information Sciences (SIBIRCON). — 2017. — С. 109-112.
- 13 Tolebi G., Dairbekov N.S., Kurmankhojayev D., Mussabayev R. Reinforcement Learning Intersection Controller // 2018 14th International Conference on Electronics Computer and Computation (ICECCO). — 2018. — С. 206-212.
- 14 Teknomo, Kardi, Q-Learning by Examples, 2005. URL: <http://people.revoledu.com/kardi/tutorial/ReinforcementLearning/index.html>
- 15 Материалы из интернета — Введение в различные алгоритмы обучения с подкреплением (Q-Learning, SARSA, DQN, DDPG). URL: <https://habr.com/ru/articles/561746/>.

Приложение А

```
import os, sys

import time

if 'SUMO_HOME' in os.environ:
    tools = os.path.join(os.environ['SUMO_HOME'], 'tools')
    sys.path.append(tools)
else:
    sys.exit("please declare environment variable 'SUMO_HOME'")

import traci
import traci.constants

#sumoBinary = "C:"+os.sep+"Sumo"+os.sep+"bin"+os.sep+"sumo-gui.exe"
sumoBinary = r'C:\Program Files (x86)\Eclipse\Sumo\bin\sumo-gui.exe'
sumoCfg = r'C:\Users\siane\Desktop\test2\test.sumocfg'
sumoCmd = [sumoBinary, "-c", sumoCfg, "--start"]

traci.start(sumoCmd)

(20, 'SUMO 1.14.1')

traci.gui.setSchema("View #0", "real world")

step = 0
while step < 1000:
    #this runs one simulation step
    traci.simulationStep();

    vehicles=traci.vehicle.getIDList();
    for i in range(0,1):
        #print(len(vehicles))
        #print(vehicles[i])
        #get actual speed
        print("Speed ", vehicles[i], ":",traci.vehicle.getSpeed(vehicles[i]), " m/s")
    step += 1
traci.close()
```

Рисунок А.1 – Скрипт для запуска симуляции SUMO через TraCI на языке Python

Приложение Б

Ниже представлена реализация основных классов для запуска алгоритма Q-learning.

```
from sumo_rl.exploration.epsilon_greedy import EpsilonGreedy

class QAgent:

    def __init__(self, action_space, alpha=0.5, gamma=0.95,
                 exploration_strategy=EpsilonGreedy()):
        self.state = None
        self.action_space = action_space
        self.action = None
        self.alpha = alpha
        self.gamma = gamma
        self.q_table = {('L', 'L', 'L', 'L'): [0 for _ in range(self.action_space.n)]}
        self.exploration = exploration_strategy
        self.acc_reward = 0

    def set_starting_state(self, starting_state):
        self.state = starting_state
        if starting_state not in self.q_table:
            self.q_table[starting_state] = [0 for _ in range(self.action_space.n)]

    def get_qtable(self):
        return self.q_table

    def set_qtable(self, q_table):
        self.q_table = q_table

    def act(self):
        """Choose action based on Q-table."""
        self.action = self.exploration.choose(self.q_table, self.state,
                                             self.action_space)
        return self.action

    def learn(self, next_state, reward, done=False):
        if next_state not in self.q_table:
            self.q_table[next_state] = [0 for _ in range(self.action_space.n)]

        s = self.state
        s1 = next_state
        a = self.action
        self.q_table[s][a] = self.q_table[s][a] + self.alpha * (
            reward + self.gamma * max(self.q_table[s1]) - self.q_table[s][a]
        )
        self.state = s1
        self.acc_reward += reward
```

Рисунок Б.1 – Реализация агента Q-learning

Продолжение Б

```
class IntersectionController(object):
    def __init__(self, id, edges, signal_plan, acq_time_interval, yellow_time):
        self.id = id
        self.edges = edges
        self.signal_plan = signal_plan
        self.cycle_count = 0
        self.measure_time_step = acq_time_interval
        self.measurements = []
        self.yellow_time = yellow_time
    def get_intersection_id(self):
        return self.id
    def getEdgeIds(self):
        return self.edges
    def set_signal_plan(self, signal_plan):
        self.signal_plan = signal_plan
    def set_traffic_light_logic(self, grDurNS, grDurWE):
        gyrDef = traci.trafficlight.Logic(
            'SP'+self.id,
            0,
            0,
            [
                traci.trafficlight.Phase(grDurNS, 'GGGrrrrrGGGrrrrr', grDurNS, grDurNS),
                traci.trafficlight.Phase(self.yellow_time, 'yyyyrrrrrryyyyrrrrr',
                    self.yellow_time, self.yellow_time),
                traci.trafficlight.Phase(grDurWE, 'rrrrGGGgrrrrrGGGg', grDurWE, grDurWE),
                traci.trafficlight.Phase(self.yellow_time, 'rrrryyyyrrrrrryyyy',
                    self.yellow_time, self.yellow_time)])
        traci.trafficlight.setProgramLogic(self.id, gyrDef)
        traci.trafficlight.setPhase(self.id, 0)
        traci.trafficlight.setProgram(self.id, 'SP'+self.id)
    def num_of_cars_in_dir(self, direction):
        vehicle_num = 0
        for edge in self.getEdgeIds()[direction]:
            vehicle_num += traci.edge.getLastStepVehicleNumber(edge)
        return vehicle_num
    def encode(self, state):
        return tuple([self._help(num) for num in state])
    def _help(self, num):
        if num <= 13:
            return 'L'
        elif num <= 26:
            return 'M'
        else:
            return 'H'
    def get_state(self):
        return [
            self.num_of_cars_in_dir('NS'), self.num_of_cars_in_dir('SN'),
            self.num_of_cars_in_dir('WE'), self.num_of_cars_in_dir('EW')]
    def mean_num_of_car(self):
        return sum(self.get_state())/4.0
    def reward(self, beta, mean):
        first = beta*(min(mean - np.array(self.get_state())))
        second = (1 - beta) * (- max(np.array(self.get_state())))
        return first+second
```

Рисунок Б.2 – Реализация класса «перекресток»

Продолжение Б

```
import numpy as np

class EpsilonGreedy:

    def __init__(self, initial_epsilon=1.0):
        self.epsilon = initial_epsilon

    def choose(self, q_table, state, action_space):
        if np.random.rand() < self.epsilon:
            action = int(action_space.sample())
        else:
            action = np.argmax(q_table[state])
        return action
```

Рисунок Б.3 - Реализация ϵ -жадной стратегии

Приложение В

Для создания симуляционной модели двух перекрестков необходимо наличие файлов: test.con.xml, test.edg.xml, test.nod.xml, test.nod.xml, test.flow.xml, test.turn.xml, которые описывают основные характеристики транспортной сети и спроса на трафик. Все параметры внутри данных файлов задаются вручную.

```
1 <nodes>
2   <node id="0" x="0.0" y="0.0" type="traffic_light" tlType="static"/>
3   <node id="2" x="+150.0" y="0.0" type="traffic_light" tlType="static"/>
4
5   <node id="1" x="0.0" y="+150.0" type="unregulated"/>
6   <node id="3" x="0.0" y="-150.0" type="unregulated"/>
7   <node id="4" x="-150.0" y="0.0" type="unregulated"/>
8
9   <node id="5" x="+150.0" y="+150.0" type="unregulated"/>
10  <node id="6" x="+300.0" y="0.0" type="unregulated"/>
11  <node id="7" x="+150.0" y="-150.0" type="unregulated"/>
12 </nodes>
```

Рисунок В.1 – test.nod.xml

```
1 <edges>
2   <edge id="D10" from="1" to="0" numLanes="2" speed="16.67"/>
3   <edge id="D01" from="0" to="1" numLanes="2" speed="16.67"/>
4   <edge id="D20" from="2" to="0" numLanes="2" speed="16.67"/>
5   <edge id="D02" from="0" to="2" numLanes="2" speed="16.67"/>
6   <edge id="D30" from="3" to="0" numLanes="2" speed="16.67"/>
7   <edge id="D03" from="0" to="3" numLanes="2" speed="16.67"/>
8   <edge id="D40" from="4" to="0" numLanes="2" speed="16.67"/>
9   <edge id="D04" from="0" to="4" numLanes="2" speed="16.67"/>
10
11  <edge id="D52" from="5" to="2" numLanes="2" speed="16.67"/>
12  <edge id="D25" from="2" to="5" numLanes="2" speed="16.67"/>
13  <edge id="D26" from="2" to="6" numLanes="2" speed="16.67"/>
14  <edge id="D62" from="6" to="2" numLanes="2" speed="16.67"/>
15  <edge id="D72" from="7" to="2" numLanes="2" speed="16.67"/>
16  <edge id="D27" from="2" to="7" numLanes="2" speed="16.67"/>
17 </edges>
```

Рисунок В.2 – test.edg.xml

Продолжение В

```
1 <connections>
2
3   <connection from="D10" to="D03" fromLane="0" toLane="0" />
4   <connection from="D10" to="D03" fromLane="1" toLane="1" />
5   <connection from="D10" to="D02" fromLane="1" toLane="1" />
6   <connection from="D10" to="D04" fromLane="0" toLane="0" />
7
8   <connection from="D30" to="D01" fromLane="0" toLane="0" />
9   <connection from="D30" to="D01" fromLane="1" toLane="1" />
10  <connection from="D30" to="D02" fromLane="0" toLane="0" />
11  <connection from="D30" to="D04" fromLane="1" toLane="1" />
12
13  <connection from="D40" to="D01" fromLane="1" toLane="1" />
14  <connection from="D40" to="D02" fromLane="0" toLane="0" />
15  <connection from="D40" to="D02" fromLane="1" toLane="1" />
16  <connection from="D40" to="D03" fromLane="0" toLane="0" />
17
18  <connection from="D02" to="D26" fromLane="1" toLane="1" />
19  <connection from="D02" to="D26" fromLane="0" toLane="0" />
20  <connection from="D02" to="D25" fromLane="1" toLane="1" />
21  <connection from="D02" to="D27" fromLane="0" toLane="0" />
22
23  <connection from="D52" to="D27" fromLane="0" toLane="0" />
24  <connection from="D52" to="D27" fromLane="1" toLane="1" />
25  <connection from="D52" to="D26" fromLane="1" toLane="1" />
26  <connection from="D52" to="D20" fromLane="0" toLane="0" />
27
28  <connection from="D72" to="D25" fromLane="0" toLane="0" />
29  <connection from="D72" to="D25" fromLane="1" toLane="1" />
30  <connection from="D72" to="D26" fromLane="0" toLane="0" />
31  <connection from="D72" to="D20" fromLane="1" toLane="1" />
32
33  <connection from="D62" to="D27" fromLane="1" toLane="1" />
34  <connection from="D62" to="D20" fromLane="0" toLane="0" />
35  <connection from="D62" to="D20" fromLane="1" toLane="1" />
36  <connection from="D62" to="D25" fromLane="0" toLane="0" />
37
38 </connections>
```

Рисунок В.3 – test.con.xml

Продолжение В

```
1 <turns>
2   <interval begin="0" end="35000">
3     <fromEdge id="D10">
4       <toEdge id="D02" probability="0.0"/>
5       <toEdge id="D03" probability="0.7"/>
6       <toEdge id="D04" probability="0.3"/>
7     </fromEdge>
8     <fromEdge id="D20">
9       <toEdge id="D01" probability="0.3"/>
10      <toEdge id="D03" probability="0.0"/>
11      <toEdge id="D04" probability="0.7"/>
12    </fromEdge>
13    <fromEdge id="D30">
14      <toEdge id="D01" probability="0.7"/>
15      <toEdge id="D02" probability="0.3"/>
16      <toEdge id="D04" probability="0.0"/>
17    </fromEdge>
18    <fromEdge id="D40">
19      <toEdge id="D01" probability="0.0"/>
20      <toEdge id="D02" probability="0.7"/>
21      <toEdge id="D03" probability="0.3"/>
22    </fromEdge>
23
24    <fromEdge id="D52">
25      <toEdge id="D26" probability="0.0"/>
26      <toEdge id="D27" probability="0.7"/>
27      <toEdge id="D20" probability="0.3"/>
28    </fromEdge>
29    <fromEdge id="D62">
30      <toEdge id="D27" probability="0.0"/>
31      <toEdge id="D20" probability="0.7"/>
32      <toEdge id="D25" probability="0.3"/>
33    </fromEdge>
34    <fromEdge id="D72">
35      <toEdge id="D20" probability="0.0"/>
36      <toEdge id="D25" probability="0.7"/>
37      <toEdge id="D26" probability="0.3"/>
38    </fromEdge>
39    <fromEdge id="D02">
40      <toEdge id="D25" probability="0.0"/>
41      <toEdge id="D26" probability="0.7"/>
42      <toEdge id="D27" probability="0.3"/>
43    </fromEdge>
44  </interval>
45 </turns>
```

Рисунок В.4 – test.edg.xml

Продолжение В

```
1 <flows>
2   <flow id="0" from="D10" begin="0" end="34900" vehsPerHour="1200"/>
3   <flow id="1" from="D30" begin="0" end="34999" vehsPerHour="1200"/>
4   <flow id="2" from="D40" begin="0" end="34999" vehsPerHour="1200"/>
5
6   <flow id="3" from="D52" begin="0" end="34900" vehsPerHour="1200"/>
7   <flow id="4" from="D62" begin="0" end="34999" vehsPerHour="1200"/>
8   <flow id="5" from="D72" begin="0" end="34999" vehsPerHour="1200"/>
9
10 </flows>
```

Рисунок В.5 – test.flow.xml

```
1 <configuration>
2   <input>
3     <net-file value="netgen/test2.net.xml"/>
4     <route-files value="demand/jtrrouter/test2_nonstat_35000.rou.xml"/>
5     <additional-files value="test2.add.xml"/>
6   </input>
7   <time>
8     <begin value="0"/>
9     <end value="35000"/>
10  </time>
11
12  <gui_only>
13    <gui-settings-file value="test2.settings.xml"/>
14  </gui_only>
15
16  <traci_server>
17    <remote-port value="8873"/>
18  </traci_server>
19 </configuration>
```

Рисунок В.6 – test.sumocfg

РЕЦЕНЗИЯ

на дипломную работу студента

Нагай Анастасии Сергеевны

6B06103 — Математическое и компьютерное моделирование

На тему: «Математическое моделирование транспортных потоков на примере управления светофорами»

ЗАМЕЧАНИЯ К РАБОТЕ

В работе, представленной на рецензирование, показано математическое моделирование транспортных потоков на примере управления светофорами. Перед дипломантом стояла задача проведения адаптивного управления для одного из двух перекрестков транспортной сети с целью оптимизации транспортных потоков и оценке его эффективности.

В первой главе рассмотрены подходы и методы математического моделирования транспортных потоков, описана математическая модель, используемая в программе SUMO.

Во второй главе представлены основные правила создания симуляционной модели транспортной сети и спроса на трафик в SUMO.

В третьей главе изложена теория по алгоритму «Q-learning» и описаны необходимые для поставленной задачи пространство действий и пространство состояний, а также функция награды; были подобраны параметры обучения.

Четвертая глава посвящена экспериментам на различных видах транспортной сети: состоящей из одного перекрестка и двух. Для дальнейшей оценки эффективности адаптивного управления собраны необходимые для построения графиков данные.

В результате произведено сравнение работы адаптивного и фиксированных светофорных планов на двух перекрестках по следующим критериям:

- 1) сохранение баланса между ребрами перекрестка по числу автомобилей;
- 2) снижение числа автомобилей на ребрах перекрестка.

Графический и текстовый материал оформлен в соответствии с требованиями ГОСТ, предъявляемыми к оформлению учебных работ.

Оценка работы

Учитывая актуальность дипломной работы, считаю, что дипломная работа заслуживает оценки «100», а студент Нагай А. С. — присвоения степени бакалавра по специальности 6B06103 — «Математическое и компьютерное моделирование».

Рецензент

Профессор департамента
«Математика» университета
имени Сулеймана Демиреля,
д.ф.-м.н., доцент
(должность, уч. степень, звание)


«06» 

Н. С. Даирбеков

2023 г.

ОТЗЫВ

НАУЧНОГО РУКОВОДИТЕЛЯ

На дипломную работу
Нагай Анастасии Сергеевны
6В06103 – Математическое и компьютерное моделирование

На тему: «Математическое моделирование транспортных потоков на примере управления светофорами»

Перед Нагай А. С. стояла следующая задача — рассмотреть способы адаптивного управления светофорами для уменьшения пробок на перекрестках. В качестве инструмента решения было предложено использовать методы машинного обучения.

Были построены две дорожно-транспортные сети для создания симуляций на них: одна состоит из одного регулируемого перекрестка с адаптивным светофорным планом, а другая — из двух, причем один из них имеет фиксированный светофорный план, а на втором подбирается адаптивный. Для построения модели использовалось программное обеспечение для симуляции транспортных потоков SUMO. Следует отметить, что для простоты транспортный поток на перекрестках рассматривался фиксированным. С помощью метода «Q-learning» была получена оценка для каждого действия в определенном состоянии дорожно-транспортной сети. Были собраны необходимые данные для построения графиков, используемых для оценки эффективности адаптивного управления, а также произведено сравнение работы адаптивного светофорного плана с фиксированными.

Дипломная работа Нагай Анастасии Сергеевны состоит трех основных частей: теоретической части, основного результата и заключения.

Считаю, что Нагай А. С. отлично справилась с поставленной ей задачей, ее дипломная работа соответствует всем выдвигаемым к таким работам требованиям. Считаю, что дипломная работа заслуживает оценки «100», а Нагай А. С. — присуждения ей академической степени бакалавра по специальности 6В06103 — «Математическое и компьютерное моделирование».

Научный руководитель
Ассоц. профессор кафедры ВМиМ,

К.Ф.-М.Н.

(должность, уч. степень, звание)

 Зимин Р.Н.

(подпись)

«06» июня 2023 г.

Протокол

о проверке на наличие неавторизованных заимствований (плагиата)

Автор: Нагай Анастасия Сергеевна

Соавтор (если имеется):

Тип работы: Дипломная работа

Название работы: Математическое моделирование транспортных потоков на примере управления светофорами.doc

Научный руководитель: Решат Зимин

Коэффициент Подобия 1: 1.9

Коэффициент Подобия 2: 0

Микропробелы: 4

Знаки из других алфавитов: 0

Интервалы: 0

Белые Знаки: 0

После проверки Отчета Подобия было сделано следующее заключение:

Заимствования, выявленные в работе, является законным и не является плагиатом. Уровень подобия не превышает допустимого предела. Таким образом работа независима и принимается.

Заимствование не является плагиатом, но превышено пороговое значение уровня подобия. Таким образом работа возвращается на доработку.

Выявлены заимствования и плагиат или преднамеренные текстовые искажения (манипуляции), как предполагаемые попытки укрытия плагиата, которые делают работу противоречащей требованиям приложения 5 приказа 595 МОН РК, закону об авторских и смежных правах РК, а также кодексу этики и процедурам. Таким образом работа не принимается.

Обоснование:

Дата 2.06.23

Заведующий кафедрой



**Университеттің жүйе администраторы мен Академиялық мәселелер департаменті
директорының ұқсастық есебіне талдау хаттамасы**

Жүйе администраторы мен Академиялық мәселелер департаментінің директоры көрсетілген еңбекке қатысты дайындалған Плагиаттың алдын алу және анықтау жүйесінің толық ұқсастық есебімен танысқанын мәлімдейді:

Автор: Нагай Анастасия Сергеевна

Тақырыбы: Математическое моделирование транспортных потоков на примере управления светофорами.doc

Жетекшісі: Решат Зимин

1-ұқсастық коэффициенті (30): 1.9

2-ұқсастық коэффициенті (5): 0

Дәйексөз (35): 0.9

Әріптерді ауыстыру: 0

Аралықтар: 0

Шағын кеңістіктер: 4

Ақ белгілер: 0

Ұқсастық есебін талдай отырып, Жүйе администраторы мен Академиялық мәселелер департаментінің директоры келесі шешімдерді мәлімдейді :

Ғылыми еңбекте табылған ұқсастықтар плагиат болып есептелмейді. Осыған байланысты жұмыс өз бетінше жазылған болып санала отырып, қорғауға жіберіледі.

Осы жұмыстағы ұқсастықтар плагиат болып есептелмейді, бірақ олардың шамадан тыс көптігі еңбектің құндылығына және автордың ғылыми жұмысты өзі жазғанына қатысты күмән тудырады. Осыған байланысты ұқсастықтарды шектеу мақсатында жұмыс қайта өңдеуге жіберілсін.

Еңбекте анықталған ұқсастықтар жосықсыз және плагиаттың белгілері болып саналады немесе мәтіндері қасақана бұрмаланып плагиат белгілері жасырылған. Осыған байланысты жұмыс қорғауға жіберілмейді.

Негіздеме:

Күні 2-06-23

Кафедра меңгерушісі



Протокол

о проверке на наличие неавторизованных заимствований (плагиата)

Автор: Нагай Анастасия Сергеевна

Соавтор (если имеется):

Тип работы: Дипломная работа

Название работы: Математическое моделирование транспортных потоков на примере управления светофорами.doc

Научный руководитель: Решат Зимин

Коэффициент Подобия 1: 1.9

Коэффициент Подобия 2: 0

Микропробелы: 4

Знаки из других алфавитов: 0

Интервалы: 0

Белые Знаки: 0

После проверки Отчета Подобия было сделано следующее заключение:

Заимствования, выявленные в работе, является законным и не является плагиатом. Уровень подобия не превышает допустимого предела. Таким образом работа независима и принимается.

Заимствование не является плагиатом, но превышено пороговое значение уровня подобия. Таким образом работа возвращается на доработку.

Выявлены заимствования и плагиат или преднамеренные текстовые искажения (манипуляции), как предполагаемые попытки укрытия плагиата, которые делают работу противоречащей требованиям приложения 5 приказа 595 МОН РК, закону об авторских и смежных правах РК, а также кодексу этики и процедурам. Таким образом работа не принимается.

Обоснование:

Дата 2.06.23


проверяющий эксперт