

**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»**

**Багдат Мергуль Жумабайкызы**

**Об одной задаче оптимизации движения транспортных потоков**

**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»

**ДОПУЩЕН К ЗАЩИТЕ**

Заведующая кафедрой ВМиМ

канд. физ-мат. наук

ассоциированный профессор

*Г.А. Тулешева*

«06» *июня* 2023 г.



**ДИПЛОМНАЯ РАБОТА**

На тему: Об одной задаче оптимизации движения транспортных потоков

6B06103 – Математическое и компьютерное моделирование

Выполнил

Багдат М.Ж.

Рецензент

Д.ф. - м.н., профессор департамента  
«Математика» университета имени

Сулеймана Демиреля

*Д.С. Даирбеков*

«05» *июня* 2023 г.



Научный руководитель

кандидат физико-математических наук

*Р.Н. Зимин*

«05» *июня* 2023 г.

Алматы 2023

Алматы 2023  
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РЕСПУБЛИКИ КАЗАХСТАН

Казахский 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, доцент

Г.А.Тулешева



2023 г.

**ЗАДАНИЕ**

**на выполнение дипломного проекта**

Обучающемуся: *Багдат Мергуль Жумабайкызы*

Тема: *Об одной задаче оптимизации движения транспортных потоков*

Утверждена приказом проректора по академической работе: № 408 – П/Ө

от "23" ноября 2022г.

Срок сдачи законченного проекта: "22" мая 2023 г.

Исходные данные к дипломному проекту:

*А) описание метода обучения с подкреплением;*

*Б) документация приложения SUMO;*

*В) техническая документация к подключению TraCI;*

Перечень подлежащих разработке в дипломном проекте вопросов: (с точным указанием обязательных чертежей): *представлены 15 слайда презентации.*

Рекомендуемая основная литература: *из 10 наименований.*

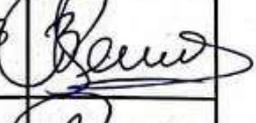
## ГРАФИК

подготовки дипломного проекта

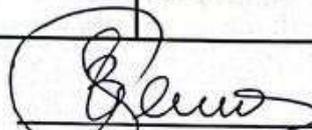
Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
Теория по методам управления дорожного движения	02.02.2023	Выполнено
Создание симуляционной модели	28.02.2023	Выполнено
Выбор метода машинного обучения для работы с задачей. Обучение выбранного метода	16.03.2023	Выполнено
Проведение экспериментов и анализ результатов	20.04.2023	Выполнено

## Подписи

консультантов и нормоконтролера на законченный дипломный проект с указанием относящихся к ним разделов проекта

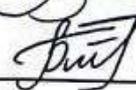
Наименование разделов	Консультанты, Ф. И.О. (уч.степень, звание)	Дата подписания	Подпись
Теория по методам управления дорожного движения	канд. физ.-мат. наук, Зимин Р.Н.	02.02.2023	
Создание модели	канд. физ.-мат. наук, Зимин Р.Н.	28.02.2023	
Обучение выбранного метода	канд. физ.-мат. наук, Зимин Р.Н.	16.03.2023	
Проведение экспериментов и анализ результатов	канд. физ.-мат. наук, Зимин Р.Н.	20.04.2023	
Нормоконтролер	канд. физ.-мат. наук, Шатманов Ж.Ж.	05.06.2023	

Научный руководитель



Р.Н. Зимин

Задание принял к исполнению обучающийся



М.Ж. Багдат

Дата

«15» 01 2022г.

## Протокол

### о проверке на наличие неавторизованных заимствований (плагиата)

Автор: Багдат Мергуль Жумабайкызы

Соавтор (если имеется):

Тип работы: Дипломный проект

Название работы: Об одной задаче оптимизации движения транспортных потоков.doc

Научный руководитель: Решат Зимин

Коэффициент Подобия 1: 4.7

Коэффициент Подобия 2: 2.7

Микропробелы: 0

Знаки из здругих алфавитов: 1

Интервалы: 0

Белые Знаки: 8

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 02.06.2023г.

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: Багдат Мергуль Жумабайқызы**

**Тақырыбы: Об одной задаче оптимизации движения транспортных потоков.doc**

**Жетекшісі: Решат Зимин**

**1-ұқсастық коэффициенті (30): 4.7**

**2-ұқсастық коэффициенті (5): 2.7**

**Дәйексөз (35): 0.7**

**Әріптерді ауыстыру: 1**

**Аралықтар: 0**

**Шағын кеңістіктер: 0**

**Ақ белгілер: 8**

**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 жіберілісін.

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

**Негіздеме:**

Күні 02.06.2023г.

Кафедра меңгерушісі



## Протокол

### о проверке на наличие неавторизованных заимствований (плагиата)

Автор: Багдат Мергуль Жумабайкызы

Соавтор (если имеется):

Тип работы: Дипломный проект

Название работы: Об одной задаче оптимизации движения транспортных потоков.doc

Научный руководитель: Решат Зимин

Коэффициент Подобия 1: 4.7

Коэффициент Подобия 2: 2.7

Микропробелы: 0

Знаки из других алфавитов: 1

Интервалы: 0

Белые Знаки: 8

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 02.06.2023г.

  
проверяющий эксперт

## ОТЗЫВ

### НАУЧНОГО РУКОВОДИТЕЛЯ

На дипломную работу  
Багдат Мергуль Жумабайкызы  
6В06103 — Математическое и компьютерное моделирование

Тема: Об одной задаче оптимизации движения транспортных потоков

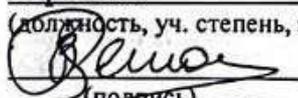
Перед Багдат М. Ж. стояла следующая задача — рассмотреть способы оптимизации движения транспортных средств для уменьшения заторов на дорогах. В качестве инструмента решения было предложено использовать методы машинного обучения.

Была построена дорожно-транспортная сеть для создания симуляций на ней: она состоит из двух регулируемых перекрестков. Для построения модели использовалось программное обеспечение для симуляции транспортных потоков SUMO. В качестве оптимизации потоков транспортных средств на построенных перекрестках была организована «зеленая волна». Были сгенерированы данные светофорных планов и различных скоростей. С помощью метода «Q-learning» была получена оценка для каждого действия в определенном состоянии дорожно-транспортной сети. На основании полученных оценок были получены оптимальные светофорные планы для организации «зеленой волны». Следует отметить, что «зеленая волна» создавалась одновременно в двух встречных направлениях.

Дипломная работа Багдат Мергуль Жумабайкызы состоит трех основных частей: теоретической части, основного результата и заключения.

Считаю, что Багдат М. Ж. справилась с поставленной ей задачей, ее дипломная работа соответствует выдвигаемым к таким работам требованиям. Считаю, что дипломная работа заслуживает оценки «100», а Багдат М. Ж. — 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)  
«05» июня 2023 г.

## РЕЦЕНЗИЯ

на дипломную работу

Багдат Мергуль Жумабайкызы

6B06103 — Математическое и компьютерное моделирование

На тему: Об одной задаче оптимизации движения транспортных потоков

### ЗАМЕЧАНИЯ К РАБОТЕ

В пояснительной записке, предоставленной на рецензирование, рассматривается одна задача оптимизации движения транспортных потоков — задача выбора и оценка двух методов уменьшения заторов и дальнейшая их оптимизация.

В первой главе данной дипломной работы описаны методы управления дорожным движением, выбранные для оптимизации. Были рассмотрены преимущества выбранных методов и описана общая характеристика их параметров координации.

Во второй главе была построена дорожно-транспортная сеть для создания симуляций на ней: она состоит из двух регулируемых перекрестков. Для построения модели использовалось программное обеспечение для симуляции транспортных потоков SUMO.

В третьей главе генерировались данные светофорных планов и различных скоростей. Приведены параметры регулирования в виде пространства состояний, которое состоит из скоростей транспортных средств. Был создан список пространства действий с различными светофорными планами, описанными посредством длин зеленой и красной фазы. Для выгрузки данных из SUMO для последующего анализа использовалось программное обеспечение TraCI. С помощью метода «Q-learning» была получена оценка для каждого действия в определенном состоянии дорожно-транспортной сети.

В четвертой главе рассматривается задача оптимизации светофорных планов, полученных в третьей главе. Параметром координации выбрано смещение светофорного плана для второго перекрестка. Благодаря симуляциям получены выборки оптимизированных светофорных планов для различных скоростей, с учетом смещения. Выведены графики зависимостей времени ожидания машин на перекрестках от светофорных планов для различных значений смещения.

### Оценка работы

Считаю, что дипломный проект «Об одной задаче оптимизации движения транспортных потоков» заслуживает оценки «100», а студент Багдат М. Ж. — присуждения академической степени бакалавра по специальности 6B06103 — «Математическое и компьютерное моделирование».

Рецензент

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

  
«05»  2023 г.  
Кадр  
Қазақстан Республикасының Ғылым және Білім Қазіреті

Н.С. Даирбеков

## **АННОТАЦИЯ**

В данном проекте была рассмотрена задача оптимизации движения транспортных потоков путем создания движения зеленой волны. Транспортная сеть моделируется на программе SUMO – DLR. Дальнейшая ее оптимизация происходит благодаря методу Q-learning.

В результате были получены оптимальные светофорные планы для создания безостановочного движения машиной. Была создана общая модель, по которой можно рассчитать и подобрать план для различных скоростей транспортного потока.

## **АҢДАТПА**

Бұл жобада Жасыл толқын қозғалысын құру арқылы көлік ағындарының қозғалысын оңтайландыру тапсырмасы қарастырылды. Көлік желісі SUMO – DLR бағдарламасында модельденеді. Оны одан әрі оңтайландыру Q-learning әдісінің арқасында жүзеге асырылды.

Нәтижесінде машинаның тоқтаусыз қозғалысын жасау үшін оңтайлы бағдарлам жоспарлары алынды. Көлік ағынының әртүрлі жылдамдықтарына бағдарлам жоспарларын тандай алатын жалпы модель құрастырылды.

## **ABSTRACT**

In this project, was considered the task of optimizing the movement of traffic flows by creating a green wave movement. The transport network is modeled on the program SUMO — DLR. Its further optimization is due to the Q-learning method.

As a result, optimal traffic light signal plans were obtained to create a non-stop movement by car. Was created a general model, according to which it is possible to calculate and select a plan for different speed of traffic flow.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1 Постановка задачи	8
1.1 Методы управления дорожным движением	8
1.2 Способы уменьшения заторов	9
1.2.1 Заторы на сужениях дорог и прямое решение этой проблемы	9
1.2.2 Основная характеристика движения «Зеленая волна»	12
1.3 Q-learning как метод оптимизации	14
2 Симуляционная модель	16
2.1 SUMO DLR	16
2.2 Создание дорожной сети	17
2.3 Создание путей и транспортных средств	18
2.4 TraCI	19
3 Модель обучения	21
3.1 Модель с сужением дороги и трудности возникшие для нее	21
3.2 Обучающаяся модель Q – learning для создания движения «Зеленая волна»	23
3.3 Функция для взаимодействия модели и среды	26
3.4 Заполнение Q – таблицы	29
4 Эксперименты и результаты	29
4.1 Смещения светофорного плана	29
4.2 Получение оптимизированного светофорного плана и смещения для 2 перекрестков	30
4.2.1 Общие характеристики модели	30
4.2.2 Анализ полученных данных для скорости 7 м/с	31
4.2.2 Анализ полученных данных для всех скоростей в общей картине	35
ЗАКЛЮЧЕНИЕ	37
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	38
Приложение А	39
Приложение Б	40
Приложение В	43

## ВВЕДЕНИЕ

Для двадцать первого века использование светофоров для управления дорожным движением является посредственной вещью. Но нельзя сказать что просто установив светофор можно избавиться от всех проблем возникающих на дорожной сети. Для каждого города, особенно для таких больших городов как Алматы и Астана, понятие «час — пик» является чем то обыденным. Это показывает что для уменьшения заторов и конфликтов на дорогах простых светофоров недостаточно. Правильное использование этого инструмента является огромным и единственным ключом улучшения управления дорожного движения.

В данной работе рассматривается задача оптимизации движения транспортных потоков двумя различными сценариями. Первый сценарий заключается в уменьшении времени затора на сужениях дорог, за счет подключения ускорения в горлышке дорожной сети. Второй сценарий рассматривает задачу оптимизации светофорных планов для создания движения «зеленой волны». Термин «зеленая волна» подразумевает собой проезд перекрестков без встречи красного сигнала, то есть безостановочное движение. За счет данного движения появляется возможность уменьшения заторов на дорогах, так как взвод транспортных средств пройдя один перекресток создает бесперебойное движение по всей магистрали.

Идея решения задачи в первом сценарии заключается в получении изменении скоростей, учитывая время замедления и прохода параллельного потока.

Для лучшей оптимизации светофорных планов для второго сценария будет использован метод Q – learning. Главным преимуществом метода является то, что автоматизированная система может принимать мелкие решения и обучаться учитывая все детальные изменения.

Методы будут внедряться и обучаться на модели дорожной сети созданной на программе SUMO – DLR.

# 1 Управление дорожным движением

## 1.1 Методы управления дорожным движением

Дорожное движение – это движение транспортных средств и людей в определенном месте или с определенной целью. Истинное происхождение контроля за дорожным движением неизвестно, но исторические свидетельства показывают, что проблемы с дорожным движением возникли, когда появились главные дороги. Пробки на дорогах – одна из главных проблем в каждой развивающейся стране. Заторы могут появиться как систематически так и случайно. Такие заторы вызывают задержки на перекрестках и приводят к экономическим потерям. Эти потери могут проявляться в виде неэффективного управления временем, поскольку транспортные средства должны ждать смены светофора; растрата расходных материалов, таких как топливо, и выброс окиси углерода из транспортных средств представляют опасность для здоровья окружающей среды и качества жизни людей. Одним из распространенных подходов к борьбе с дорожными заторами является строительство инфраструктуры, такой как дороги, мосты, объездные полосы, эстакады и множество других. Становится все труднее создавать больше инфраструктуры, чтобы, по крайней мере, уменьшить пробки на дорогах. Необходимо учитывать не только высокую стоимость, но и нехватку места и экологический ущерб от строительства новой дороги. Подразумевается, что строительство новых дорог и расширение существующих приводит только к дополнительному трафику, который продолжает увеличиваться до тех пор, пока пиковые заторы не вернуться на прежний уровень.

Общий транспортный поток во многом зависит от работы перекрестков на пути движения потока. Транспортные перекрестки – это сложные места на любом шоссе. Конфликты на перекрестке различны для разных типов перекрестков. Следовательно, тип системы управления дорожным движением, которая должна быть развернута, зависит от интенсивности движения, геометрии перекрестка и важности дороги, но не ограничивается ими. Управление дорожным движением может быть достигнуто на различных уровнях: пассивное управление, полуконтроль или активный контроль благодаря светофорам. [3]

При пассивном регулировании дорожного движения практически отсутствует явный контроль, выходящий за рамки соблюдения правил дорожного движения. Дорожные знаки и разметка в основном регулируют движение. Такое пассивное управление трафиком эффективно только там, где объем трафика очень мал.

При половинчатом регулировании движения или частичном регулировании дорожного движения водителей аккуратно направляют по поворотам (кольцевым развязкам) и каналам (дорожным островкам), чтобы избежать конфликтов. Но это может привести появлению еще одного затора уже в соседних перекрестках, на объезд и «бег» от заторов времени может уйти

столько же или даже больше чем ожидание в нем. Такой подход эффективен только при небольших объемах трафика.

Активное управление дорожным движением с использованием светофора основано на подходе с разделением времени. В определенное время и с использованием соответствующих сигналов движение определенных транспортных средств ограничено, в то время как некоторым другим транспортным средствам разрешается проезжать через перекресток. Сигналы могут работать в нескольких режимах: сигналы фиксированного времени или сигналы приведенные в действие в зависимости от ситуации. В сигналах фиксированного времени время цикла, фазы и интервал каждого сигнала фиксированы. При динамическом (активируемом) сигнале светофора контроллер использует входные данные от детекторов (датчиков) для настройки синхронизации и фазирования сигнала в пределах, установленных программированием контроллера. [5]

## **1.2 Способы уменьшения заторов**

В данной работе было выбрано две различные идеи уменьшения заторов. Рассматриваются они для различных видов дорог. Первая это на сужение дорог, за счет расчета скорости. Вторая создание движения зеленой волны, за счет управления светофорами.

### **1.2.1 Заторы на сужениях дорог и прямое решение этой проблемы**

Сужение дорог — уменьшение количества однонаправленных полос в участке магистрали. Изменения в количестве полос могут быть по различным причинам. Основная часть этих случаев была заложена изначально в геометрии этой дороги, из-за внешних факторов состояния окружающей среды. Например, начиная с какой-то точки, вдоль дороги растут деревья.

По анализу [10] сужение дороги дает большое количество преимуществ и широко используется для «создания» дополнительной полосы. Изначально дорога является 4 полосой, но в какой-то части магистрали только 2 из них используется на постоянной основе. Остальные 2 полосы остаются для экстренных случаев и в основное спокойное время могут использоваться лишь транспортными средствами, которые собираются повернуть в предстоящем кусочке дороги. Также в статье указано, что одним из преимуществ сужения дорог является уменьшение длины расстояния между сторонами поперечной дороге. Но это является преимуществом только в общей картине, а в частности для пешеходов и другим дорогам, пересекающим в суженной части.

За счет уменьшения полос, транспортными средствами, которые до этого были рассеяны по  $n$  различным полосам, создается задача быстрого перестроения в дорогу с меньшим количеством полос.

Правила поведения транспортного средства во время столкновения с сужением дорог полностью опирается на дорожные знаки. Предупреждением о уменьшении количества полос служит сам знак «сужение дороги». В зависимости от геометрии дороги различаются три вида. [7]

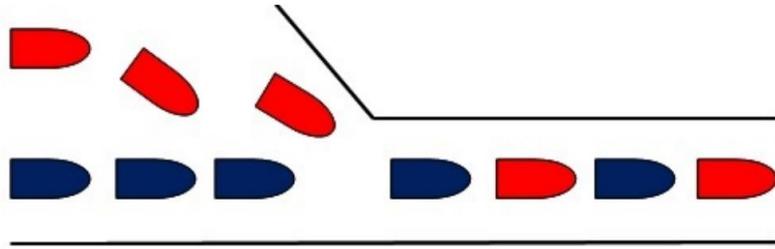


Рисунок 1.1 – Схема вклинивания машин в другую полосу

Основная идея решения задачи является увеличение скорости на втором участке дороги для более быстрого рассеивания потока и уменьшении времени затвора за счет ускорения в начале самого стыка, восстановить потерянное за счет вклинивания расстояние между машинами. Сделать это нужно до того, как начнет ускоряться следующая машина.

Прямое нахождение изменения скорости после сужения дороги, назовем ее 2 частью, происходит благодаря следующему алгоритму.

Изначальные данные которые есть у нас:

$v_0$  — скорость машин до горлышка в каждом ряду

$a$  — ускорение в горлышке

$d_0$  — расстояние между машинами до горлышка

$d$  — расстояние между машинами после горлышка.

$l$  — длина машины

Считаем, что машина вклинивается прямо посередине двух других, которые были изначально на первой полосе. Тогда

$$d = \frac{d_0 - l}{2} \quad (1.1)$$

Следовательно, нужно увеличить расстояние на  $\Delta d$ , которое находится по формуле:

$$\Delta d = d_1 - d = d_1 - \frac{d_0 - l}{2} \quad (1.2)$$

Простейшая формула для нахождения пройденного расстояния при равноускоренном движении:

$$\Delta d = \int_0^T at dt = \frac{aT^2}{2} \quad (1.3)$$

$$T = \sqrt{\frac{2\Delta d}{a}} = \sqrt{\frac{2d_1 - d_0 + l}{a}} \quad (1.4)$$

$$\Delta d = d_1 - \frac{d_0 - l}{2} = \frac{2d_1 - d_0 + l}{2} \quad (1.5)$$

За время  $T$  машина проезжает расстояние  $vT$ . Следующая машина не должна ускоряться, следовательно,:

$$\frac{d_0 - l}{2} \geq v_0 T = v \sqrt{\frac{2d_1 - d_0 + l}{a}} \quad (1.6)$$

Максимальная плотность потока достигается при равенстве. Пусть  $x = d_0 - l$ . Тогда:

$$ax^2 + 4v_0^2 x - 8v_0^2 d_1 = 0 \quad (1.7)$$

Находим дискриминант:

$$\frac{D}{4} = 4v_0^4 x + 8av_0^2 d_1 = 4v_0^2 (v_0^2 + 2ad_1) \quad (1.8)$$

Находим корень:

$$x = \frac{-2v_0^2 \pm 2v_0 \sqrt{1 + 2ad_1/v_0^2}}{a} = \frac{2v_0^2}{a} \left( -1 \pm \sqrt{1 + \frac{2ad_1}{v_0^2}} \right) \quad (1.9)$$

Подставляем в начальную формулу. Окончательное формула нахождения скорости в части 2 после сужения и расстояния между машинами будут следующими:

$$d_0 = l + \frac{2v_0^2}{a} \left( -1 \pm \sqrt{1 + \frac{2ad_1}{v_0^2}} \right) \quad (1.10)$$

$$v_0 = \frac{(d_0 - l)\sqrt{a}}{2\sqrt{2d_1 - (d_0 - l)}} \quad (1.11)$$

## 1.2.2 Основная характеристика движения «Зеленая волна»

Скоординированное управление светофорами всегда было эффективным способом повышения безопасности движения и эффективности работы на перекрестках. Близко расположенные светофоры вдоль магистрали обычно координируются с общей длиной цикла и соответствующими смещениями, так что взвод транспортных средств может максимально встречать зеленый свет при движении по всей магистрали, чтобы обеспечить максимальную ширину полосы пропускания зеленой волны, уменьшить задержку и количество остановок, а также обеспечить бесперебойное движение транспорта по главной дороге. Этот подход называется координированным контролем артерий по зеленой волне. Он управляет светофорами соседних перекрестков, чтобы один за другим включать

зеленый свет в соответствии с определенной временной последовательностью в необходимом направлении. [7]

Встречая красный сигнал светофора, транспортные средства замедляют ход и останавливаются перед стоп — линией, а затем остаются на холостом ходу в ожидании зеленого сигнала, наконец, ускоряются, чтобы выехать с перекрестка после того, как светофор загорится зеленым, весь процесс включает в себя состояние ускорения, холостого хода, замедления транспортного средства. Согласно анализу [4], при таких условиях вождения режим постоянной скорости имеет незначительное значение, что не приводит к увеличению потребления энергии и снижению выбросов.

Также нужно учитывать человеческий фактор. Для любого водителя будет намного приятнее и спокойнее ехать на постоянной скорости без необходимости остановок на так нелюбимый ими красный свет.

Двунаправленная конструкция зеленой волны в режиме симметричного входа обычно использует алгебраический метод для определения наилучшей общей длины цикла и смещения путем поиска идеального интервала пересечения, который лучше всего соответствует фактическому интервалу пересечения, чтобы система управления коридором создавала полосу пропускания зеленой волны как можно больше и получить наилучший эффект координации. [5]

Руководство, принятое при выборе параметров координации, основано на устоявшейся практике и техниках. Эти настройки включают длину цикла, раздельное распределение и оптимизацию смещения.

Продолжительность цикла — это время в секундах, которое требуется сигналу для завершения одного полного цикла индикации. Он указывает временной интервал между включением зеленого цвета для одного подхода и до следующего включения зеленого цвета.

Разбиения - это часть времени, выделяемая каждой фазе на пересечении. Они рассчитываются на основе поэтапности пересечения и ожидаемого спроса. Разбиения могут быть выражены либо в процентах от цикла, либо в секундах. Разделение обычно включает в себя желтый период, полностью красный период и зеленый период. Для реализации в контроллере сигналов сумма фазовых разделений должна быть равна (или меньше) длине цикла.

Смещение – значение, с разницей на которое следующий перекресток будет подключать светофорный план. Сдвиг включения фаз. Прямой расчет оптимального смещения осуществляется только для уже выбранного светофорного плана, учитывая геометрию транспортной сети.

В данной работе параметрами координации светофоров были взяты длины раздельных распределений красной и зеленой фазы. Был задан исходный режим регулирования с одинаковой длительностью красной и зеленой фазы. Затем путем последовательного изменения длин фаз находят минимальный интегральный критерий качества управления.

Для определения производительности системы светофоров необходимо учитывать три основных фактора: пропускная способность перекрестка, длина очереди и время ожидания. [5][6]

Пропускная способность представляет количество транспортных средств, выезжающих с перекрестка за определенный период времени. На пропускную способность влияют различные факторы, как состояние дороги, уровень квалифицированности водителей и весь спектр транспортных средств на этой дороге. Сигнальный план светофора должен быть оптимизирован для обработки как можно большего количества транспортных средств в течение периода времени.

Под длиной очереди понимается количество автомобилей, ожидающих на перекрестке. Чем дольше идет разгрузка перекрестка, тем длиннее становится очередь появившаяся из-за затора. И чем дальше находится транспортное средство по очереди от самого перекрестка, тем больше скапливается время его ожидания.

А время ожидания представляет собой время, необходимое для выезда с перекрестка после прибытия транспортного средства на перекресток. Время ожидания можно рассчитать, как для каждого транспортного средства, так и для группы.

Обычно для определения ситуации используется среднее значение длины очереди или времени ожидания. Для нашего случая основным фактором является время застоя машины на каждом узле дороги. Наблюдение ведется над одним транспортным средством, поэтому очереди, образовавшиеся на перекрестках не учитываются. Основной задачей движения «зеленая волна», это обнулить время ожидания. Когда на одном перекрестке загорается зеленый свет, водители могут беспрепятственно проезжать последующие перекрестки без помех со стороны красного света. Только группа автомобилей может использовать зеленую волну до того, как временная полоса прервется, чтобы уступить дорогу другим транспортным потокам. Длина группы автомобилей определяется благодаря длине зеленой фазы при данном светофорном плане и будет являться пропускной способностью перекрестка.

### **1.3 Q-learning как метод оптимизации**

Обучение с подкреплением — это метод машинного обучения, при котором агент учится вести себя в окружающей среде, выполняя действия и видя результаты действий. [8] Изначально модель не имеет каких-либо знаний и данных, но благодаря произведению действий обучается и узнает о этой среде. Алгоритм обучения с подкреплением основан на том что, с каждым изменением ситуации модель извлекает урок от полученного результата и выбирает следующее действие основываясь на нем. За каждое действие он получает обратную связь, вознаграждение при улучшении состояния и наказание при его ухудшении. Данный метод будет очень удобен для нашей задачи так как

управление светофорами для улучшения состояния дорог и уменьшение конфликтов сама по себе задача без идеального ответа. Так же преимуществом метода является, то что автоматизированная система может принимать многие решения и обучаться, учитывая все детальные изменения.

- Основными элементами системы обучения с подкреплением являются:
- Обучающий агент с подкреплением – это сущность, которую мы обучаем принимать правильные решения. Например, робот, которого обучают передвигаться по дому без сбоев.
- Окружающая среда – это внешнее пространство, с которым взаимодействует агент. Например, комната или дом по которому передвигается робот.
- Политика – алгоритм, лежащий на основе выбора действий. [9]

Обучение с подкреплением имеет несколько различных алгоритмов. Основными из них являются SARSA (State-Action-Reward-State-Action), Q – Learning и Deep Q – Network.

Q – Learning – это политика обучения с подкреплением, которая позволяет найти следующее наилучшее действие, учитывая текущее состояние. Он выбирает это действие случайным образом и стремится максимизировать вознаграждение. У Q-Learning нет ограничений на выбор действий. Работает в динамичных, неопределенных условиях. Алгоритм сам по себе адаптивный и создан таким образом, что напрямую реагирует на изменения окружающей среды.

Модели Q – learning работают в рамках итеративного процесса, который включает в себя множество компонентов, работающих вместе, чтобы помочь обучить модель. Итеративный процесс включает в себя обучение агента путем изучения окружающей среды и обновления модели по мере продолжения исследования. Составляющими компонентами модели являются следующие:

- Состояние  $S_t$  — определяет текущее положение агента и характеристики окружающей среды, изменившиеся благодаря проведенному действию. Все зависит от самой задачи.

- Действие  $A_t$  — акт, который агент делает на текущем временном шаге. Мы заранее знаем набор действий (решений), которые агент может выполнить. Чем больше этот набор, тем быстрее и лучше обучается модель.

- Вознаграждение  $R_t$  – это награда получаемое агентом за изменение ситуации. Она будет положительной при улучшении ситуации и отрицательной при обратном случае. Алгоритм расчета вознаграждения напрямую зависит от полученной ситуации и его характеристик.

Основная цель агента — максимизировать общее вознаграждение, которое он получает в долгосрочной перспективе. Взаимодействие агента с окружающей средой является непрерывным и происходит в течение времени  $t$  следующим образом:

$$S_t \rightarrow (R_{t+1}, S_{t+1}) \quad (1.12)$$

В схематической версии это выглядит так:



Рисунок 1.2 – Схема работы агента

Для лучшего изучения процесса строится так называемые Q – таблицы. Q — таблица включает столбцы и строки со списками вознаграждений за лучшие действия каждого состояния в конкретной среде. Она помогает агенту оценить ситуацию и понять какое действие именно в этом состоянии даст наибольшее вознаграждение.

Столбцами таблицы являются действия, которые может предпринять агент, а строки состоят из различных ситуаций, с которыми он может столкнуться. По мере того как агент взаимодействует с окружающей средой и получает обратную связь в виде вознаграждений или штрафных, значения в Q — таблице обновляются, чтобы отразить то, чему научилась модель.

$$Q(S_{t+1}, A_{t+1}) \rightarrow Q(S_t, A_t) + \alpha [R_{t+1} + \max_a Q(S_{t+1}, a) - Q(S_t, A_t)] \quad (1.13)$$

$Q(S_{t+1}, A_{t+1})$	Новое значение переменной Q
$Q(S_t, A_t)$	Значение Q в предыдущем шаге
$\alpha$	Скорость обучения модели
$R_{t+1}$	Значение вознаграждения за этот шаг
$\max_a Q(S_{t+1}, a)$	Функция выбора лучшего действия из всего набора, основываясь на максимальном вознаграждении

## 2 Симуляционная модель

### 2.1 SUMO DLR

Виртуальный эксперимент с использованием моделирования дорожного движения обеспечивает эффективный способ решения и оптимизации существующих и предстоящих проблем.

Симулятор: SUMO DLR («Моделирование городской мобильности») [1] – это микроскопическая, интер – и мультимодальная, пространственно непрерывная и дискретная во времени платформа моделирования транспортных потоков.

Симулятор SUMO используется для моделирования и анализа поведения транспортных средств, потоков дорожного движения, работы светофоров, движения пешеходов и общественного транспорта. Он позволяет создавать имитацию трафика, также является набором приложений, которые помогают подготовить и выполнить моделирование сценария движения. Благодаря нему можно отслеживать и изучать эффективности стратегий управления трафиком, тестировать новые различные решения в области мобильности и оптимизации транспортных систем.

SUMO на самом деле представляет собой набор множества различных программ, которые необходимо использовать совместно для настройки и запуска симуляции. Они автоматизируют основные задачи по созданию, выполнению и оценке моделирования дорожного движения, такие как импорт сети, расчет маршрута, визуализация и расчет выбросов.

Преимущества данного приложения:

- бесплатный для использования и общедоступен;
- имеет широкий спектр всех составляющих деталей дорожного движения;
- детализирована до мельчайших аспектов модели транспортной сети;
- яркий графический интерфейс;
- благодаря различным встроенным приложениям есть доступ для связывания с различными методами машинного обучения и автоматизирования;
- удобен для выгрузки и загрузки данных;
- представлена обучающая документация [1];

Для создания подходящей комплексной среды моделирования используется комбинация моделирования транспортного средства, которое включает в себя цифровой прототип кооперативного и автоматизированного транспортного средства, моделирование дорожно-транспортной сети, которое обеспечивает окружающую среду, и моделирование взаимодействия, включающее функции сотрудничества используя дополнительные приложения и интерфейсы прикладного программирования. Поведение других участников дорожного движения учитывается в среде моделирования дорожного движения.

## 2.2 Создание дорожной сети

Создание дорожной сети на симуляторе SUMO можно осуществить различными способами. [1] Первый способ включает в себе более графический метод, в котором создание имитационной модели происходит напрямую благодаря встроенному приложению «netedit». Здесь можно «нарисовать» дорогу как на белом листе. Программа также позволяет дальнейшее ее редактирование, как добавление перекрестков, светофоров, остановки для общественного транспорта с помощью сетевых атрибутов.

Вторым способом является описание элементов геометрии необходимой дорожной сети в текстовом файле по скрипту используя языки программирования как Python или XML.

Третьим способом является создание с помощью приложения под названием «netgenerate». Для этого задаются параметры как количество дорог, типы перекрестков и вероятности их соединений. Используя их, программа генерирует случайную дорожную сеть.

Как четвертый способ SUMO поддерживает импорт дорожных сетей из различных форматов как OpenStreetMap, благодаря которым можно использовать готовые данные о существующих дорожных сетях и загружать их.

Основной составляющей дорожной сети SUMO являются узлы и соединяющие их ребра. В обычном понимании узлы – это перекрестки. Они состоят из положения, формы и правил полосы отвода, которые могут быть заменены светофором. Ребра представляют собой однонаправленные соединения между двумя узлами и содержат фиксированное количество полос. У полосы можно задать геометрию, информацию о разрешенных на ней классах транспортных средств и максимально допустимую скорость. Помимо этого, базового представления о дорожной сети, дорожные сети SUMO включают планы светофоров и соединения между полосами движения на перекрестках, описывающие, какие полосы могут быть использованы для выхода на ту или иную полосу. Дополнительные элементы инфраструктуры, такие как автобусные остановки, детекторы и знаки изменения скорости, хранятся в отдельных файлах и загружаются в начале моделирования.

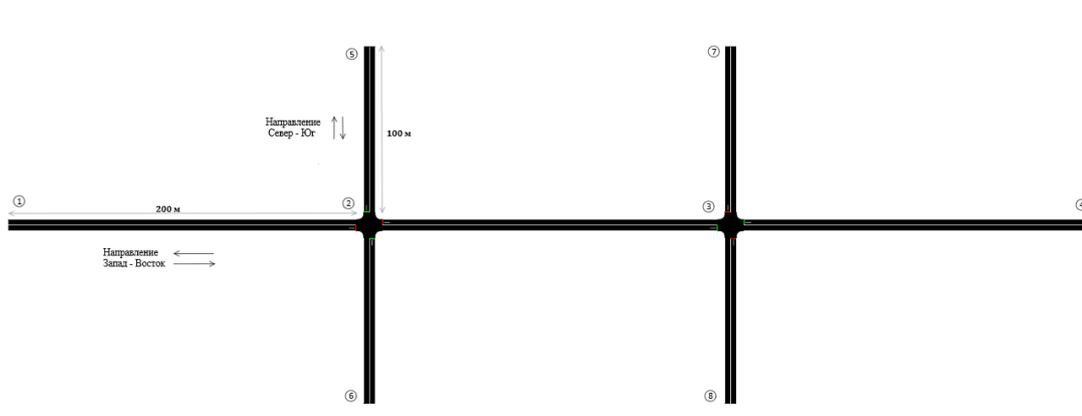


Рисунок 2.1 – Пример симуляционной модели транспортной сети

## 2.3 Создание путей и транспортных средств

Следующим шагом после получения транспортной сети это создание транспортных средств и путей для них. Для начала необходимо создать маршрут, который после уже будет указан машинам.

Для создания маршрута есть два различных способа:

- используя приложение netedit, выбирая ребра, соединенные между собой уже в созданной ранее транспортной сети вручную;
- описать геометрию маршрута через ребра по скрипту в файле формата edge.xml.

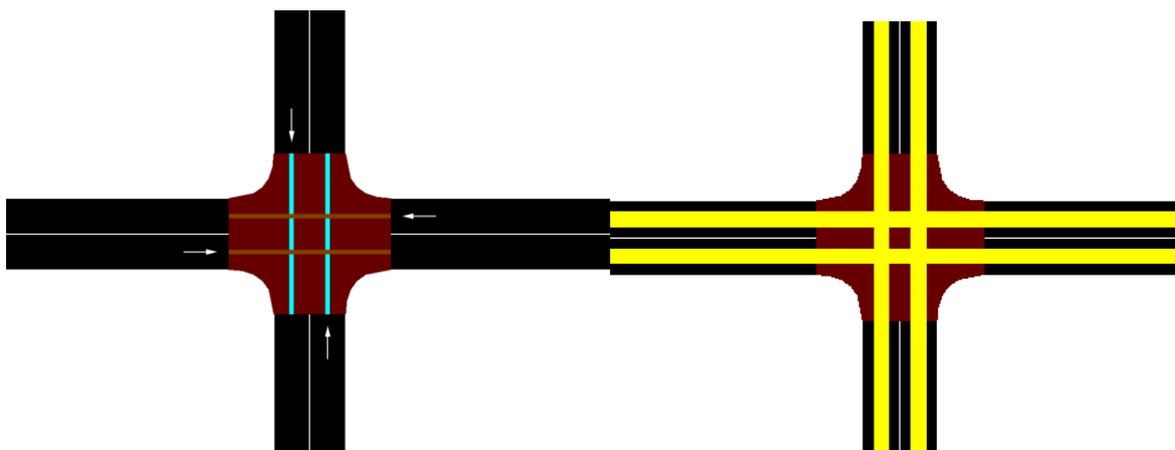


Рисунок 2.2 – Связи между узлами.

Рисунок 2.3 – Маршруты.

Для каждого маршрута присваивается имя для дальнейшего доступа к нему. Если при создании маршрута будут выбраны несоединенные между собой ребра он не будет создан и программа выдаст ошибку. Можно прописать игнорирование ошибки, но это приведет к телепортации транспортных средств.

Так как в симуляционной модели были соединены лишь ребра в направлении вперед, как показано на рисунке 2.2, маршруты были созданы только в этих направлениях, что можно увидеть на рисунке 2.3.

В этом же файле создаются транспортные средства. SUMO – DLR имеет широкий спектр транспортных средств начиная с велосипедов до грузовых машин. Тип транспортного средства – это абстрактный элемент, который оказывает большое влияние на другие элементы. При создании типового вида описываются следующие характеристики:

- название;
- длина;
- ускорение – разгонная способность автомобилей этого типа (в  $\text{м/с}^2$ );
- замедление – тормозная способность автомобилей этого типа (в  $\text{м/с}^2$ );
- максимальная скорость – скорость которую может развить автомобиль данного типа;
- минимальное расстояние – пустое место после лидирующего транспортного средства;

- несовершенства водителя – насколько плохо управляет машиной водитель.

Кроме этого при необходимости можно описать много различных дополнительных условий как выброс CO<sub>2</sub>, время задержки, время начала движения и так далее, но основными и обязательными являются вышеперечисленные.

В реальной жизни ускорение транспортного средства зависит от его физических возможностей и других эффектов, таких как сопротивление воздуха и другие. Для предотвращения того, чтобы транспортные средства в моделировании двигались быстрее, чем это возможно в действительности, рассчитывается желаемая скорость.

После получения типовой версии создается необходимое количество транспортных средств. Можно создать их вручную каждую по отдельности. Обязательными атрибутами транспортного средства являются:

- название;
- тип транспортного средства, созданный ранее;
- маршрут;
- время появления в симуляции с отсчетом от его начала;
- цвет.

Также можно создать повторяющиеся появления транспортных средств – потоки. Для всего потока будет один и тот же маршрут и тип транспортного средства. В нем можно регулировать количество машин и их частоту появления в симуляции. Описание транспортного потока состоит из тех же атрибутов что и для транспортного средства, с добавлением их частоты и временных промежутков продолжения потока.

Созданный файл формата net.xml, где описана геометрия дорожной модели и файл формата edge.xml, где описаны транспортные средства и их маршруты благодаря скрипту конвертируется в файл формата sumocfg.xml. Данный файл уже создает полную симуляцию сценария. Так же конвертацию можно сделать, используя приложение netconvert. Запуск симуляции сценария можно произвести благодаря приложению sumo – gui[]. Полная реализация транспортной модели представлена в Приложении А.

## 2.4 TraCI

Для дальнейшего взаимодействия с созданной симуляционной моделью используется встроенный интерфейс прикладного программирования под названием TraCI(сокращение от Traffic Control Interface)[2]. Благодаря предоставляемым различным функциям через него можно управлять поведением объектов симуляции, а также получать данные о их состоянии в реальный момент времени.

Используя TraCI, можно создавать пользовательские приложения и скрипты для взаимодействия с SUMO и получения таких данных, как местоположение транспортного средства, скорость, маршруты и время в пути. Эта информация может быть использована для анализа транспортного потока, оптимизации времени подачи сигналов светофора, изучения влияния изменений инфраструктуры или разработки и тестирования алгоритмов управления дорожным движением.

TraCI имеет два различных вида функций. Первый вид используется для получения данных симуляционной модели и имеют общий вид как `traci.-.get...`, где вместо тире прописывается компонент дорожной сети о которой мы хотим получить информацию, а в конце прописывается какая именно информация нужна.

Второй вид выглядит очень схоже по скрипту, но используется для задания и изменения каких-либо параметров компонента дорожной сети. Скрипт выглядит как `traci.-.set...`

В целом, TraCI расширяет возможности SUMO, предоставляя интерфейс для взаимодействия внешних программ с симуляцией, позволяя исследователям и практикам эффективно анализировать и оптимизировать транспортные системы.

### 3 Модель обучения

#### 3.1 Модель с сужением дороги и трудности возникшие для нее

Для задачи с сужением дороги, чье прямое решение описано в главе 1.2.1, была получена следующая транспортная модель:



Рисунок 3.1 – Транспортная модель

Как показано на рисунке 3.1 идет изменение количества дорог из 2 в 1. Для правильного соединения перехода для узла посередине был задан специальный тип под названием zipper. Он отвечает за перетек двух полос в одну, как нам и необходимо.

Было создано два маршрута и транспортный поток, именно поток машин, для реализации решения. В модели задаются входные интенсивности транспортных потоков с точностью до одного транспортного средства.

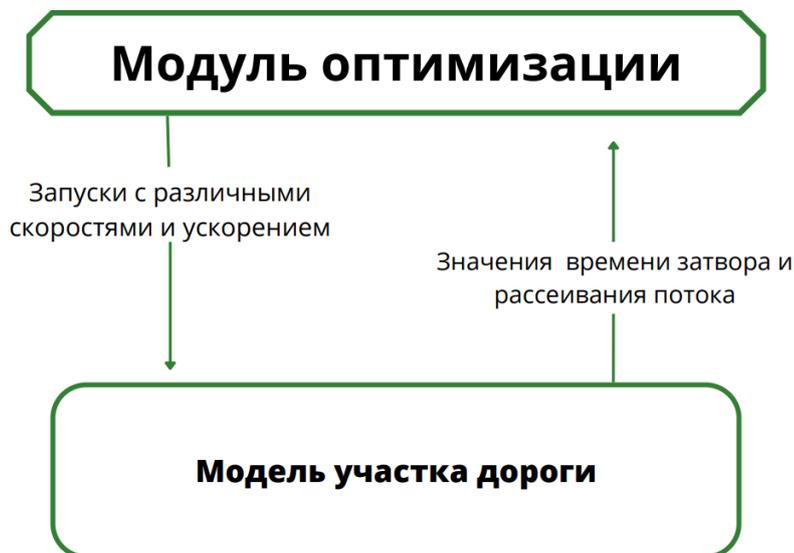
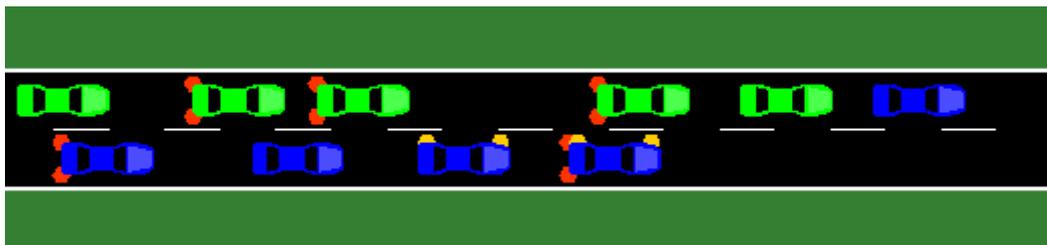


Рисунок 3.2 – Схема алгоритма действий

Модуль оптимизации запускает прогоны модели. Для каждого нового прогона меняем ускорение и начальную скорость. Записываются входные данные для фиксации прогона. А выходными данными у нас являются время затвора и время его рассеивания. Как мы понимаем количество транспортных средств напрямую влияет на время рассеивания, поэтому работаем с одним и тем же потоком, не меняя его интенсивность. В схематической версии выглядит как показано на рисунке 3.2.

На данном этапе возникла проблема с транспортной моделью, полученной на SUMO. Она не давала получить задуманный нами поток, и меняла его характеристики по логике уже заложенной в самом SUMO.

Логика SUMO для решения задачи заключалась также в изменении скорости, для получения слаженного потока во второй части дороги уже с меньшим количеством полос. Но отличие было в том, что если у нас была идея посчитать скорость для второго потока и ускорить машины, SUMO приостанавливала машинки в первой части дороги (до сужения), таким образом, что полосы были не полностью в потоке а в шахматном порядке.



Рисунок—3.3 Поведение машин на симуляционной модели.

Машины в каждой линии по отдельности начинают замедляться и останавливаться (что видно по красным фарам), до момента пока между ними не образуется расстояние, в которое влезает машина с учетом минимального расстояния между соседними машинами.

К сожалению, данное решение хоть и может теоретически являться ответом, в жизни такой волны получить будет невозможно, и это в экологическом ключе приносит больше проблем нежели само образование затора.

Для обхода проблемы с моделью была создана следующая транспортная сеть:

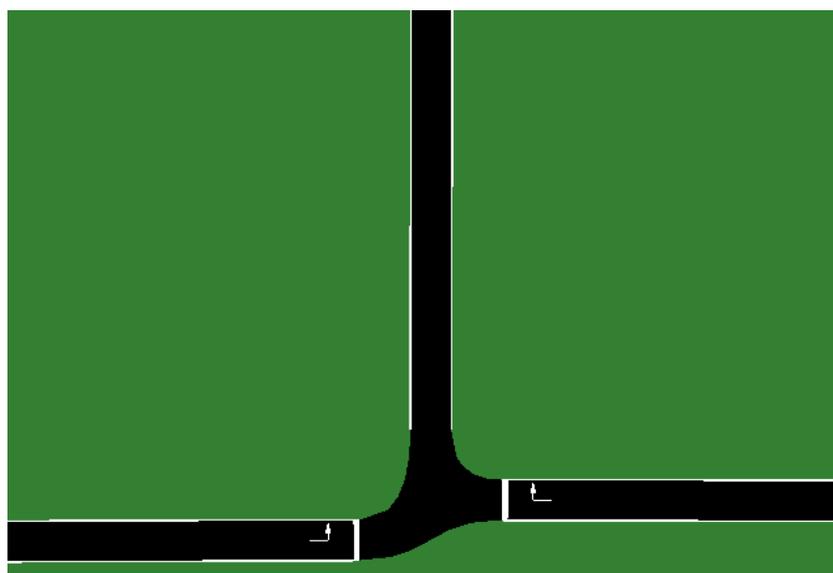


Рисунок 3.4 – Дорожная модель

Но здесь SUMO уже не позволяла задать для соединяющего узла тип zipper. А все остальные типы узлов игнорируют факт того что идет параллельный поток для обеих дорог. Часть из них ставит в приоритет одну дорогу и пропустив весь транспортный поток этой дороги, потом только переходит к пропуску второй дороги. А оставшая часть хоть и позволяет обеим полосам одновременно переходить в суженую полосу, делает это без регулирования. Для транспортных средств из двух разных полос, приехавших в одно время, происходило столкновение. А для такого случая SUMO просто телепортирует одну из машин, что к сожалению, в реальности получить в данный момент несмотря на хороший прогресс технологий нельзя.

Встроенный модуль, который отвечает за логику и алгоритм решения такой задачи в SUMO называется openGap. Является встроенное начиная с 2 версии. Но получить версию до нее, к сожалению, не получилось.

### 3.2 Обучающаяся модель Q – learning для создания движения «Зеленая волна».

Для обучающейся модели данной работы была создана следующая симуляционная модель (Рисунок 2.1):

- состоит из 8 узлов и 7 ребер;
- ребра являются однополосные;
- длины ребер в направлении Запад — Восток 200 метров;
- длины ребер в направлении Север — Юг 100 метров;
- узлы под номерами 2 и 3 снабжены светофорами;
- ребра соединены только в направлении вперед (без поворотов);
- для упрощения не снабжены автобусными остановками и пешеходами.

Все эти данные описываются и сохранены в файле d.net.xml из Приложения А.

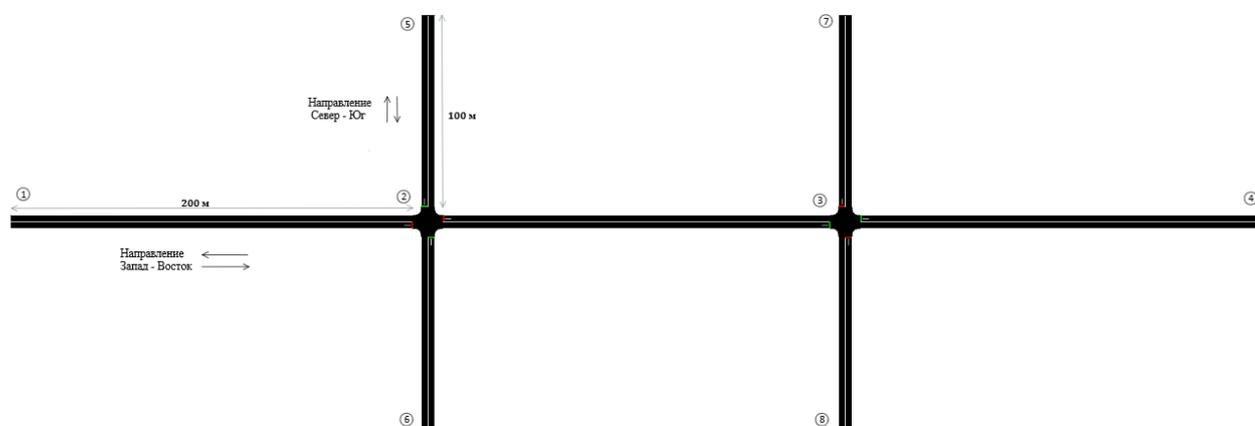


Рисунок 3.5 – Симуляционная модель транспортной сети

Для данной задачи Q – learning применяется для определения необходимого светофорного плана в зависимости от скорости машины на каждом перекрестке. Связь с симуляционной моделью, показанной на рисунке 3.1 происходит благодаря классу IntersectionController.

Как было описано в главе 1.3 компонентами обучающейся модели являются пространство действий, состояний и функция вознаграждения.

Перекресток имеет зеленую, красную и желтую фазы. Сигнальный план каждого цикла включает в себя следующие компоненты:

- Зеленая фаза (Север — Юг) — красная фаза (Восток — Запад)
- Желтая фаза (Север — Юг) — красная фаза (Восток — Запад)
- Красная фаза (Север — Юг) — зеленая фаза (Восток — Запад)
- Красная фаза (Север — Юг) — желтая фаза (Восток — Запад)

Элемент действия — это точное значение продолжительности зеленой и красной фазы в не конфликтующих направлениях, следовательно, длина одного полного цикла светофорного плана не является фиксированной. Минимальная продолжительность — 12 секунд. Максимальная продолжительность зависит от размерности пространства действий и выбирается аналитическим образом, в зависимости от скорости машин. Для увеличения точности и пространства действий шагом изменений светофорного плана является 1 секунда. Длина желтой фазы остается неизменной и всегда равна трем секундам.

После каждого цикла агент выбирает действие из пространства действий:

$$a_0=(12,12)$$

$$a_1=(12,13)$$

...

$$a_n=(12+\sqrt{n}, 12+\sqrt{n})$$

Эти действия изменяют сигнальный план следующего цикла.

В этой работе значение размера пространства действий  $n$  равно 100. Общий вид пространства действий выглядит следующим образом:

$$A=\{a_0, a_1, \dots, a_{100}\} \tag{3.1}$$

Элемент пространства состояний представляет время ожидания транспортных средств в каждом ребре и скорость транспортных средств в направлении Восток — Запад. Следовательно, на каждом временном шаге, сигнал состояния состоит из вектора длиной в 7 элементов:

- speed – скорость транспортных средств (м/с);
- EW (node2) – время ожидания машины для ребра d5;
- WE (node2) – время ожидания машины для ребра d1;
- EW' (node3) – время ожидания машины для ребра d4;
- WE' (node3) – время ожидания машины для ребра d2;
- grDurNS – длина зеленой фазы для направления Север – Юг;

- $grDurWE$  – длина зеленой фазы для направления Восток — Запад;

$$S_t = \{speed, EW, WE, EW', WE', grDurNS, grDurWE\} \quad (3.2)$$

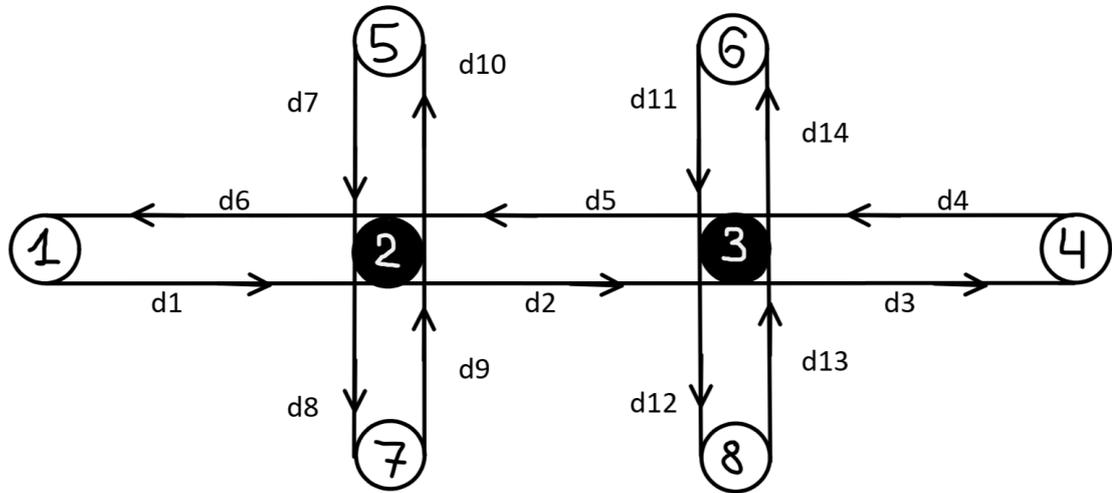


Рисунок 3.6 – Схема дорожной сети.

Помимо времени ожиданий транспортных средств в вектор состояния можно добавить другие дополнительные элементы влияющие на состояние симуляции. Например, загруженность дорог, количество машин, выбросы  $CO_2$  и так далее. Выбор элементов полностью зависит от задачи и так как для нас главная цель это создание зеленой волны, нас интересует лишь время ожидания машин на ребрах транспортной сети.

Пространство состояний будет иметь размерность 11. Для обучающейся модели было выбрано 11 различных скоростей начиная с 7 м/с до 17 м/с с размерностью шага в 1 м/с.

Формула вознаграждения должна четко отражать цели задачи. Суть движения зеленая волна, это обнуление времени ожиданий транспортных средств. Поэтому она выглядит следующим образом:

$$R_t(S_t, S_{t+1}) = -\beta(WE + EW + WE' + EW') \quad (3.3)$$

где  $\beta$  это коэффициент нормализации, принадлежащий промежутку  $(0,1)$ .

Для случаев того, когда время ожидания машин хотя бы в одном ребре отлична от нуля, агент получает отрицательное значение функции вознаграждения. Но в случае беспрепятственного проезда вознаграждение будет максимальной, а именно равна нулю.

Коэффициент нормализации  $\beta$  необходима для того чтобы полученное отрицательное вознаграждения не сильно уменьшала общее значения вознаграждения. Транспортное средство, попавшееся на красный свет, будет ждать на этом перекрестке как максимум длину красной фазы, и как минимум

его половину продолжительности. За счет чего сумма ожидания всех 4 транспортных средств может даже превышать 50.

### 3.3 Функция для взаимодействия модели и среды

Был создан class IntersectionController, в котором описываются функции для взаимодействия с симуляцией.

Функция setTrafficLightLogic — создает светофорный план и задает его на необходимый для нас перекресток. Входными данными являются grDurNS, grDurWE, где grDurNS — длина зеленой фазы для направления восток-запад и соответственно красная фаза для направления юг-север, grDurWE — длина зеленой фазы для направления юг-север и соответственно красная фаза для направления восток-запад.

Создание светофорного плана происходит благодаря функции traci.trafficlight.Logic с заданной длиной зеленой и красной фазы. Строка представляет собой набор определений света из rRgGyYoO, для красного, зеленого, желтого, выключенного, где строчные буквы означают, что поток должен замедлиться. После уже созданный и сохраненный план задается на перекресток благодаря функции traci.trafficlight.setProgram.

Функция delayOfCarsInDir — считывает время ожидания машин на дороге. Для этого используется функция traci.edge.getWaitingTime. Входным данным является название ребра, с которого нужно посчитать время.

Функция updateMeasures — записывает полученные данные и сохраняет для дальнейшей обработки.

Функция setspeed – задает скорость для всех ребер этого перекрестка. Использует функцию traci.edge.setMaxSpeed.

Полная реализация класса и функций представлена в Приложении Б,

### 3.4 Заполнение Q – таблицы

Для получения данных запускается симуляция на 12000 секунд для каждого состояния среды. Так как запуск по отдельности требует ручного закрытия программы каждый раз, симуляция запускается одна общая. Каждые 120 секунд симуляции является одним циклом. В начале каждого цикла агент сначала получает сигнал состояния, затем выполняет действие и, наконец, получает вознаграждение за это действие.

Исходя из размерностей пространства состояний и пространства действий Q – таблица будет иметь размер 11 на 100. Обновление значений таблицы происходит на каждом цикле симуляции.

Обучающаяся модель не рассматривает смещение как координационный параметр оптимизации и можно считать, что обучается только на одном

перекрестке, так как для второго перекрестка в связи с симметрией значения времени ожидания будут идентичными.

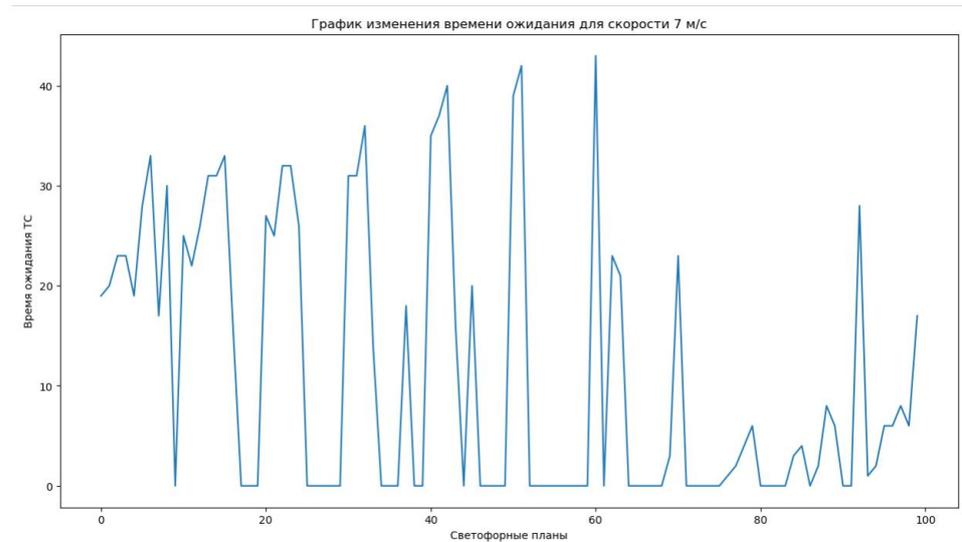


Рисунок 3.7 – График изменения времени ожидания для скорости 7 м/с

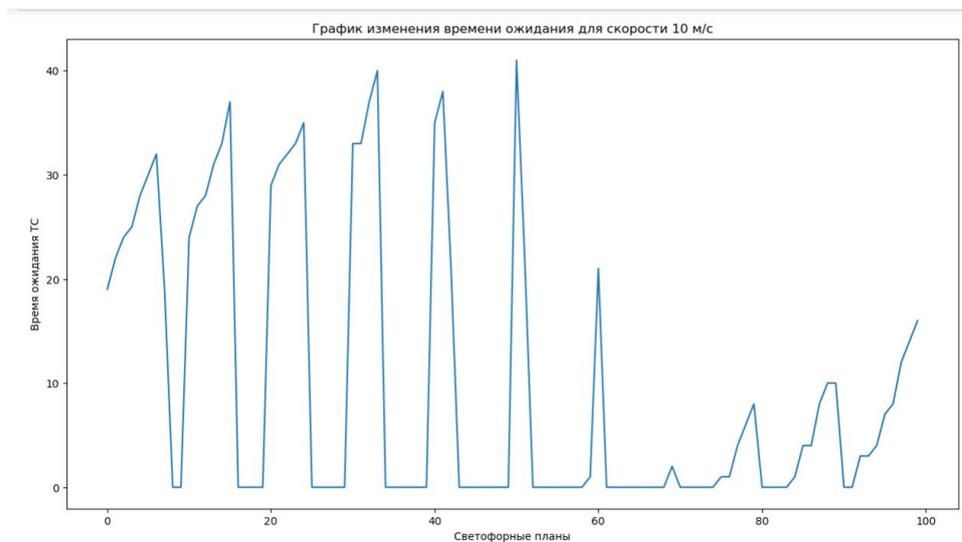


Рисунок 3.8 – График изменения времени ожидания для скорости 10 м/с

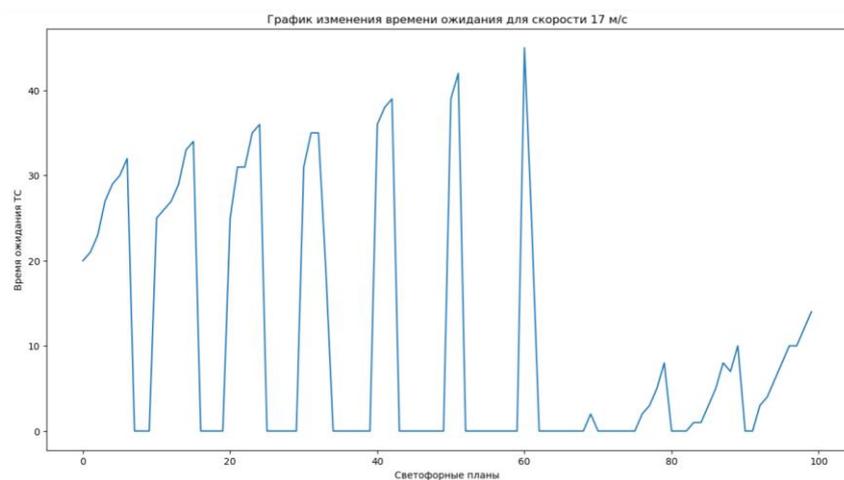


Рисунок 3.9 – График изменения времени ожидания для скорости 17 м/с

Рассматривать всю полученную таблицу и значения вознаграждений не будем. Но для показательного сравнения используем графики. В рисунках 3.7, 3.8 и 3.9 можно увидеть график изменения значения времени ожидания машин со скоростями 7 м/с, 10 м/с и 17 м/с на перекрестках по направлению Восток — Запад в зависимости от заданного светофорного плана для всего пространства действий.

Полная таблица записывается в файл для дальнейшей обработки и использования для самой модели как опору. Как мы помним, вознаграждение рассчитывается благодаря формуле (3.3), и оно отрицательного значения для действий с временем ожидания не равно нулю. То есть на графиках высота столбцов — это не рост вознаграждения, а как раз-таки наоборот, его отклонение от нужного нам значения. Количество действий, дающих наивысшее вознаграждение для каждого состояния разное и напрямую зависит от скорости транспортных средств. Отличительной чертой данной модели является то что он для довольно большого количества действий получает нужный ему результат, время ожидания равно нулю.

## 4 Эксперименты и результаты

### 4.1 Смещения светофорного плана

Как было сказано в главе 1.2 смещение светофорного плана на соседнем перекрестке также является важным координационным параметром движения зеленая волна. Прямой расчет смещения подразумевает собой расчет временного промежутка за которое транспорт проезжает расстояние между двумя перекрестками, за которыми ведется наблюдение. Также на выбор смещения напрямую влияет светофорный план первого перекрестка. Таблица значений, полученная благодаря обучающейся модели, показала, что для любой скорости транспортных средств, оптимизированного светофорного плана не один и не два. Отсюда получаем вывод, что для каждого такого светофорного плана есть возможность расчета смещения для получения зеленой волны в обе стороны. Но необходимо понимать, что заново обучать модель нам нет нужды. А всего лишь нужно чтобы уже обученная модель рассматривая лишь действия, которые приносят максимальное вознаграждение могла выбрать уже пару, состоящую из светофорного плана и смещения для каждой скорости.

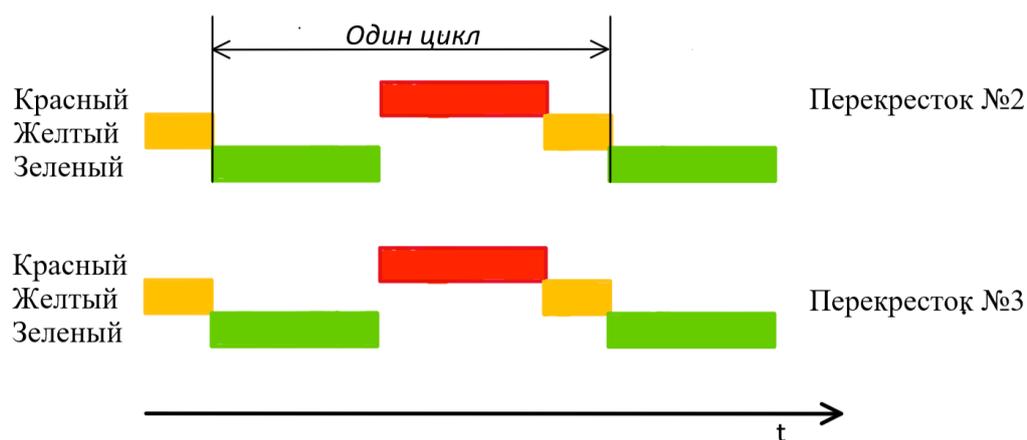


Рисунок 4.1 – Схема работы светофорного плана по сценарию 1

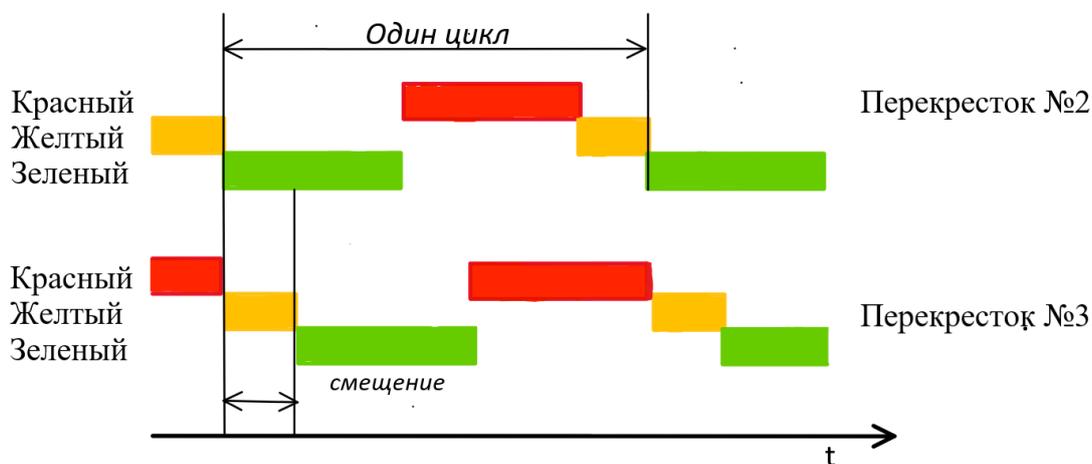


Рисунок 4.2 – Схема работы светофорного плана по сценарию 2

Выбор и расчет смещения поднимает создания движения зеленой волны на уровень выше. Она позволит эффективно управлять зеленой волной и увеличить ее общую длину. На рисунке 4.1 показана схема развития светофорного плана без смещения параллельно для двух перекрестков. Рисунок 4.2 показывает, как происходит смещение в схематической версии.

## **4.2 Получение оптимизированного светофорного плана и смещения для 2 перекрестков**

### **4.2.1 Общие характеристики модели**

Для эксперимента мы будем работать с той же транспортной моделью что была создана для обучающейся модели метода Q – learning, состоящей из двух перекрестков. Смещение равно нулю рассматривается как нулевая модель для оценки улучшения общей ситуации на сравнении с ней.

Пространство действий остается такой же как было для обучающейся модели, отличие лишь в том, что тестовая модель будет выбирать какое действие нужно предпринять, основываясь на Q – таблице. Для заданной скорости она будет брать светофорный план которое дает нулевое время ожидания для перекрестка номер 2. Принцип основывается на том, что эти действия дают вознаграждение равное нулю. То есть мы уже знаем, что именно на данных светофорных планах, машина не останавливалась на одном перекрестке и проходила на зеленую фазу. Далее настроив светофорный план для перекрестка номер 2, подключаем смещение благодаря изменению времени настройки светофорного плана для перекрестка под номером 3.

Пространство событий для данной модели будет состоять из значений скоростей и иметь размерность 11, начиная с 7 м/с до 17 м/с. В целом модель будет работать со всем пространством событий, но изменения параметров мы будем наблюдать и сравнивать только для трех значений (7 м/с, 10м/с и 17 м/с).

Результаты обучающейся модели показали, что оптимизационных светофорных планов для создания зеленой волны не один, а несколько для каждой скорости. Поэтому и в этот раз запускаем не до получения одной пары светофорного плана и смещения, а до рассмотрения, всех действий, которые выбирает модель самостоятельно и считает лучшей.

Будем рассматривать 4 разных сценария с различными значениями смещения начиная с 3 секунд до 6 секунд. Минимальное значение смещения было выбрано за счет длины самой короткой фазы светофорного плана, а максимальное значение за счет расчета половины длины минимального значения зеленой фазы.

Наблюдение ведется за двумя транспортными средствами. Один в направлении Запад — Восток, второй в обратном, Восток — Запад. Если для первой машины время ожидания не меняется для ребра  $d_1$  (он ранее в обучающейся модели для этого светофорного плана подъезжал на зеленую фазу),

но меняется для ребра d2, так как светофорный план сместился и включение зеленой фазы уже будет в другую секунду симуляции. Для второй машины, маршрут которого идет в направлении Восток — Запад через ребра d4, d5 и d6, время ожидания машин изменится для всех ребер, и основные изменения мы получим именно благодаря нему. Поэтому каких-либо изменений для функции вознаграждения мы также не получим.

Q – таблица данной модели уже уменьшится в размерах, так как пространство действий уменьшится, в связи с тем, что модель самостоятельно выбирает лишь хорошие. Размерности таблицы будут для каждого состояния разное.

Скрипт модели прикреплен в Приложении В.

#### 4.2.2 Анализ полученных данных для скорости 7 м/с

Для каждого заданного смещения получены времена ожидания машин на всех четырех наблюдаемых ребрах в направлении Восток — Запад. Рассмотрим полученные значения для скорости 7 м/с в виде графиков для более лучшего сравнения и анализа (рисунки 4.3, 4.4, 4.5 и 4.6).

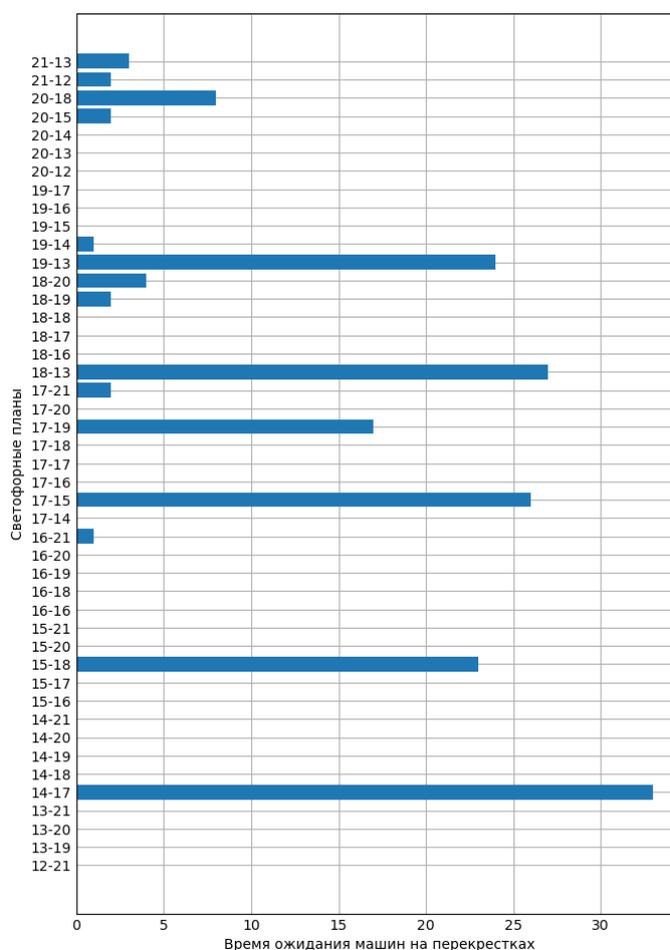


Рисунок 4.3 – График времени ожидания машин на перекрестках на скорости 7 м/с для смещения равное 3 секундам

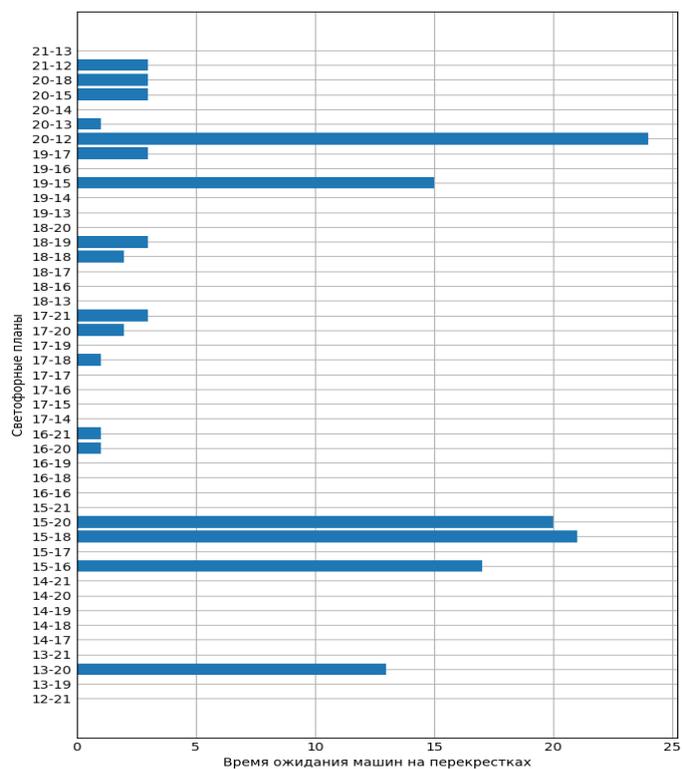


Рисунок 4.4 – График времени ожидания машин на перекрестках на скорости 7 м/с для смещения равное 4 секундам

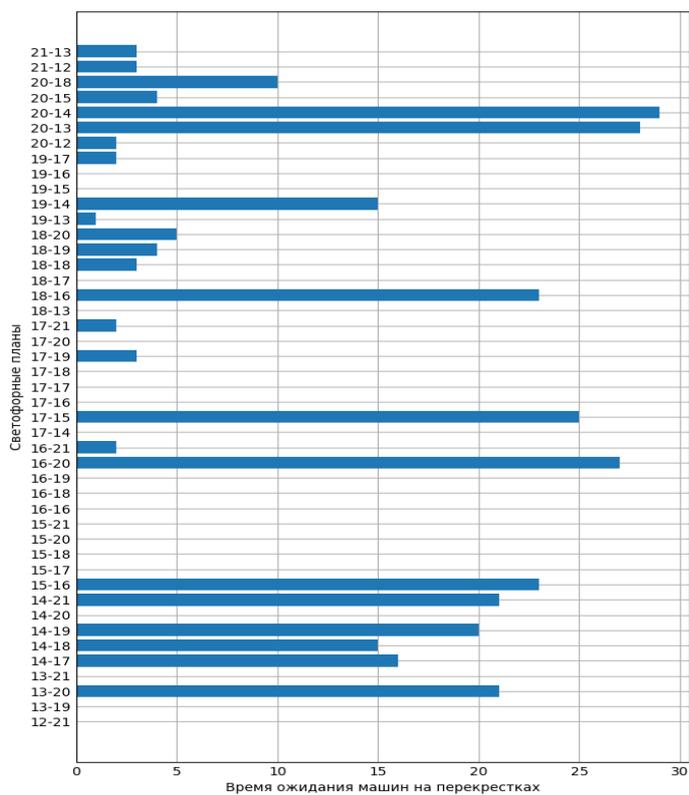


Рисунок 4.5 – График времени ожидания машин на перекрестках на скорости 7 м/с для смещения равное 5 секундам

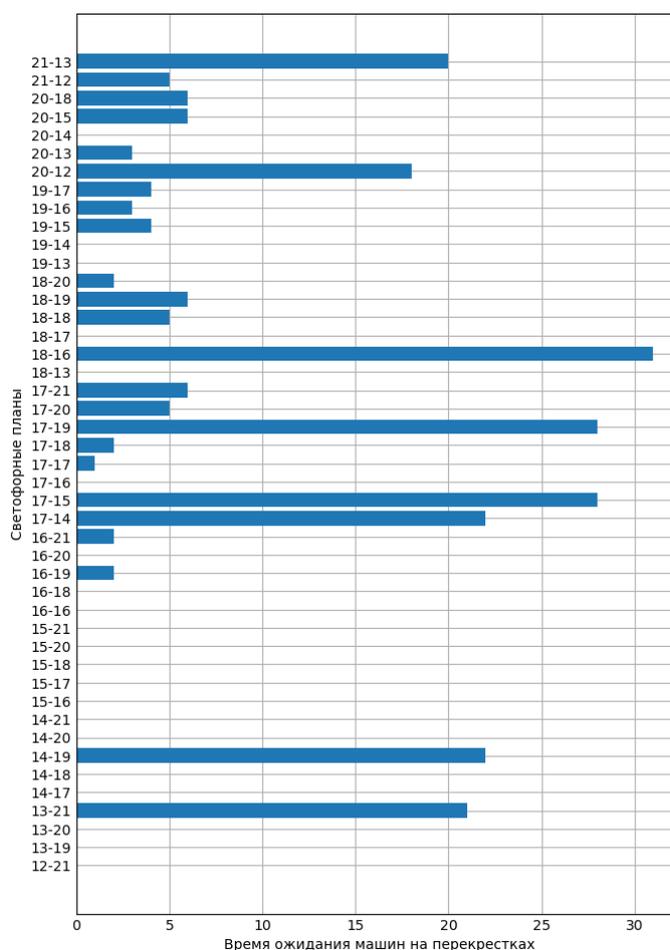


Рисунок 4.6 – График времени ожидания машин на перекрестках на скорости 7 м/с для смещения равное 6 секундам

Прямое влияние смещения на время ожидания машин можно сразу заметить благодаря резкому росту времени ожидания. Как сказано в главе 4.2.1 модель выбирала действия основываясь на Q – таблице, полученной в главе 3.3. Для смещений равное 3 и 4 секундам основная часть изменений получилась меньше чем на 10 секунд. Эта часть была получена для тех сценариев в которых машина, которая двигалась в направлениях Восток — Запад через ребра d4, d5 и d6 встретила красный свет и ждала его изменения. А ждать ей пришлось именно количество времени равное длине оставшейся части красной фазы, что в свою очередь равна длине смещения плюс время разницы длин красной и зеленой фазы.

Для смещения равное 5 секундам мы получили намного меньшую выборку хороших светофорных планов, по сравнению с остальными. Здесь, как мы видим по графику на рисунке 4.5, основная часть изменений произошла больше чем 10 секунд, и даже дошла до 30 секунд. Получается, что обе машины сталкивались с красным светом на каждом светофоре на своем пути. Но мы не можем сказать, что смещение в 5 секунд является плохим выбором. Оптимальные светофорные планы для него, хоть и меньше чем для остальных, но были получены. Также не забываем, что мы делаем анализ лишь для скорости 7 м/с.

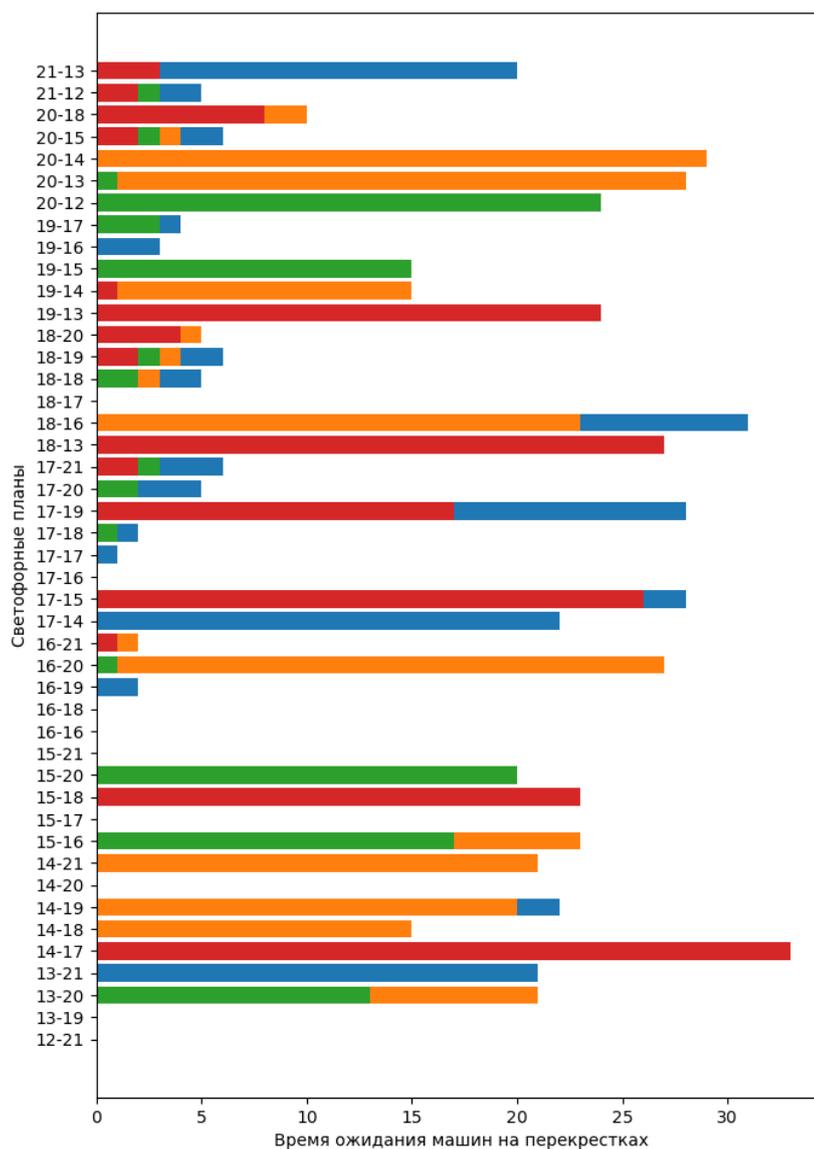


Рисунок 4.7 – Объединенный график времени ожидания машин на перекрестках на скорости 7 м/с для всех смещений

Для общего анализа светофорных планов и смещений объединяем все 4 графика (рисунок 4.7). Разными цветами обозначены значения времен ожидания для различных смещений. Благодаря нему можно наглядно увидеть в каких светофорных планах мы получим нулевое время ожидания для всех четырех различных смещений. Количество оптимизированных светофорных планов, работающих для всех 4 смещений равно 7. Это означает что из изначально рассмотренных 100 нам подходящих оказалось 7. В двух из которых длины зеленой и красной фазы являются одинаковыми, а в 3 различны лишь на 1 секунду.

Так же по этому графику можно увидеть разницу времен ожидания и динамику его изменения для разных смещений. Есть светофорные планы, которые дают нулевое время ожиданий для одного смещения, но при разнице лишь на секунду мы получим уже время ожидания равное 30 секундам.

#### 4.2.2 Анализ полученных данных для всех скоростей в общей картине

Количества оптимизационных светофорных планов для каждого смещения светофорного плана и скорости транспортного средства представлены ниже в таблице 1.

Таблица 1 — Количества оптимизированных светофорных планов

	3 с.	4 с.	5 с.	6 с	Без смещения	Для любого смещения
7 м/с	30	27	20	21	45	7
8 м/с	34	32	24	21	48	18
9 м/с	38	32	29	23	53	21
10 м/с	37	33	29	24	50	21
11 м/с	38	35	30	27	52	25
12 м/с	41	31	27	24	51	18
13 м/с	37	29	25	25	50	19
14 м/с	34	34	27	20	50	20
15 м/с	37	33	24	22	52	17
16 м/с	36	32	29	23	54	20
17 м/с	36	32	29	23	51	18

По таблице 1 можно получить следующие выводы:

1) Общее количество оптимизационных светофорных планов без смещений в среднем равно 50, что является половиной общего количества изначальной выборки. Довольно хороший результат, но не забываем, что тут не идет речь о 50 светофорных планах, которые работают для всех скоростей. Для каждой скорости эта выборка разная и сузилась напрямую в зависимости от скорости транспортных средств. Конечно есть такие, которые работают для всех 11 различным скоростям, но их всего 3. Это:

- 12 секунд – 18 секунд
- 12 секунд – 19 секунд
- 13 секунд – 17 секунд

2) Для каждого смещения было получено больше 20 различных светофорных планов. С ростом смещения количество планов уменьшается, но не в критических размерах. Среднее количество 36, 32, 27 и 23 соответственно для смещений длиной в 3, 4, 5 и 6 секунд.

3) Количество светофорных планов, которые являются оптимизированными для любого смещения в среднем равно 19, что почти составляют одну пятую часть всей изначальной выборки. Отклонение от среднего значение в среднем равно 2. График можно увидеть на рисунке 4.8. Исключение лишь для скорости

7 м/с. К сожалению, найти светофорный план, который работает для всех скоростей и всех смещений не получилось, выборка для каждой скорости разная и как мы уже получили в пункте 1 для всех без смещения работает лишь 3 плана, значит только 3 плана были в каждой выборке.

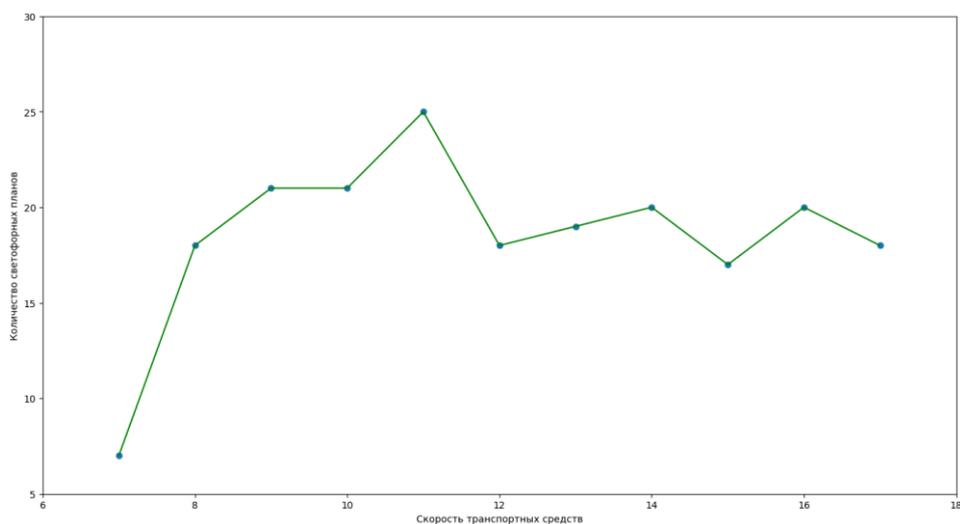


Рисунок 4.8 – График изменения количества оптимизированных светофорных планов для каждой скорости

## ЗАКЛЮЧЕНИЕ

В данной работе рассматривалась задача оптимизации движения транспортных потоков двумя различными сценариями. Первый сценарий заключался в уменьшении времени затора на сужениях дорог, за счет подключения ускорения в горлышке дорожной сети. Второй сценарий рассматривал задачу оптимизации светофорных планов для получения безостановочного движения вдоль магистрали. Основная цель заключалась в уменьшении времени ожидания транспортных средств на ребрах перекрестка.

Создание транспортно-дорожной модели для анализа данных было произведено через программное обеспечение SUMO. Дальнейшая обработка информации и поиск оптимизированных значений производился благодаря модели с политикой Q – learning, реализованной на языке программирования python.

Модель изначально обучалась благодаря подсчету времени ожидания транспортных средств в симуляционной модели состоящей из 2 перекрестков с одинаковыми светофорными планами.

Дальнейшая оптимизация полученной выборки светофорных планов, происходило за счет добавления и изменения смещения светофорного плана для второго перекрестка.

Для каждого значения смещения было получено различное количество оптимизационных светофорных планов. Аналитическим образом посчитать количество выборки нельзя. Все данные как геометрия дорожно-транспортной сети, скорость машины, светофорный план, длина смещения, характеристики транспортного средства тесно связаны друг с другом и напрямую влияют на все остальные характеристики.

Полученная модель является универсальной. В дальнейшем ее можно использовать не только для симметричных, ну и для любых транспортных сетей со сложной геометрией.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Документация SUMO: <https://sumo.dlr.de/docs/index.html>
- 2 Документация TraCI: <https://sumo.dlr.de/docs/TraCI.html>
- 3 J.F. Cordeau , A survey of optimization models for train routing and scheduling // Transportation Science, 1988 г.
- 4 Ma, D., Zhao, Y. and Liang, B. ,Study of Lightweight Vehicle Emission Characteristics under Different Driving Conditions.// Beijing Automotive, 2010 г.
- 5 N.H. Cartner, Constraining relations among offsets in synchronized signal networks //Transportation Science, 1972 г.
- 6 Статья - Francesco Corman , Andrea D’Ariano , Dario Pacciarelli , Marco Pranzo, Evaluation of green wave policy in real-time railway traffic management, 2017 г.
- 7 Кременец Ю. А., Печерский М. П., Афанасьев М. Б., Технические средства организации дорожного движения. – М.//Академкнига , 2005 г.
- 8 Материалы из интернета - Что такое обучение с подкреплением? URL: <https://www.synopsys.com/ai/what-is-reinforcement-learning.html>
- 9 Ричард С. Саттон, Эндрю Г. Барто., «Обучение с подкреплением», 2010 г.
- 10 Материалы из интернета - Lane Reduction (Road Diet) URL: [http://www.pedbikesafe.org/pedsafe/countermeasures\\_detail.cfm?CM\\_NUM=19](http://www.pedbikesafe.org/pedsafe/countermeasures_detail.cfm?CM_NUM=19)

## Приложение А

```
<edge id="d1" from="1" to="2" priority="-1">
  <lane id="d1_0" index="0" speed="13.89" length="192.80" shape="-200.00,98.40 -7.20,98.40"/>
</edge>
<edge id="d10" from="2" to="4" priority="-1">
  <lane id="d10_0" index="0" speed="13.89" length="92.80" shape="1.60,107.20 1.60,200.00"/>
</edge>
<edge id="d11" from="6" to="3" priority="-1">
  <lane id="d11_0" index="0" speed="13.89" length="92.80" shape="198.40,200.00 198.40,107.20"/>
</edge>
<edge id="d12" from="3" to="7" priority="-1">
  <lane id="d12_0" index="0" speed="13.89" length="92.80" shape="198.40,92.80 198.40,0.00"/>
</edge>
<edge id="d13" from="7" to="3" priority="-1">
  <lane id="d13_0" index="0" speed="13.89" length="92.80" shape="201.60,0.00 201.60,92.80"/>
</edge>
<edge id="d14" from="3" to="6" priority="-1">
  <lane id="d14_0" index="0" speed="13.89" length="92.80" shape="201.60,107.20 201.60,200.00"/>
</edge>
<edge id="d2" from="2" to="3" priority="-1">
  <lane id="d2_0" index="0" speed="13.89" length="185.60" shape="7.20,98.40 192.80,98.40"/>
</edge>
<edge id="d3" from="3" to="8" priority="-1">
  <lane id="d3_0" index="0" speed="13.89" length="192.80" shape="207.20,98.40 400.00,98.40"/>
</edge>
<edge id="d4" from="8" to="3" priority="-1">
  <lane id="d4_0" index="0" speed="13.89" length="192.80" shape="400.00,101.60 207.20,101.60"/>
</edge>
<edge id="d5" from="3" to="2" priority="-1">
  <lane id="d5_0" index="0" speed="13.89" length="185.60" shape="192.80,101.60 7.20,101.60"/>
</edge>
<edge id="d6" from="2" to="1" priority="-1">
  <lane id="d6_0" index="0" speed="13.89" length="192.80" shape="-7.20,101.60 -200.00,101.60"/>
</edge>
<edge id="d7" from="4" to="2" priority="-1">
  <lane id="d7_0" index="0" speed="13.89" length="92.80" shape="-1.60,200.00 -1.60,107.20"/>
</edge>
<edge id="d8" from="2" to="5" priority="-1">
  <lane id="d8_0" index="0" speed="13.89" length="92.80" shape="-1.60,92.80 -1.60,0.00"/>
</edge>
<edge id="d9" from="5" to="2" priority="-1">
  <lane id="d9_0" index="0" speed="13.89" length="92.80" shape="1.60,0.00 1.60,92.80"/>
</edge>
```

Рисунок А.1 - Скрипт создания транспортной модели

```
<!-- VTypes -->
<VType id="type1" length="5.00" maxSpeed="7.00" accel="0.8" decel="4.5" sigma="0.5"/>
<!-- Routes -->
<route id="r_0" edges="d1 d2 d3"/>
<route id="r_1" edges="d4 d5 d6"/>
<route id="r_2" edges="d7 d8"/>
<route id="r_3" edges="d9 d10"/>
<route id="r_4" edges="d11 d12"/>
<route id="r_5" edges="d13 d14"/>
<!-- Vehicles, persons and containers (sorted by depart) -->
<vehicle id="v_1" depart="0.00" type="type1" route="r_0"/>
<vehicle id="v_2" depart="0.00" type="type1" route="r_1"/>
<vehicle id="v_3" depart="0.00" type="type1" route="r_3"/>
<vehicle id="v_4" depart="0.00" type="type1" route="r_5"/>
<vehicle id="v_5" depart="0.00" type="type1" route="r_4"/>
<vehicle id="v_6" depart="0.00" type="type1" route="r_2"/>
```

Рисунок А.2 – Скрипт создания типовой модели машины, маршрутов и потоков транспортных средств

## Приложение Б

```
import os, sys
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'E:\SUMO\bin\sumo-gui.exe'
sumoCfg = r'E:\SUMO\dd.sumocfg'
sumoCmd = [sumoBinary, "-c", sumoCfg, "--start"]
```

Рисунок Б.1 – Скрипт подключения TraCI

```
class IntersectionController(object):
    def __init__(self, id, edges, interphaseDuration, signalPlan, speed):
        self.id = id
        self.edges = edges
        self.interphaseDuration = interphaseDuration
        self.signalPlan = signalPlan
        self.speed = speed
        self.cycleCount = 0
        self.measurements = []
    def getIntersectionId(self):
        return self.id
    def getEdgeIds(self):
        return self.edges
    def getInterphaseDuration(self):
        return self.interphaseDuration
    def getSignalPlan(self):
        return self.signalPlan
    def getCycleLength(self):
        return self.interphaseDuration+self.signalPlan['NS']+self.signalPlan['WE']
    def setSignalPlan(self, signalPlan):
        self.signalPlan = signalPlan
    def getCycleCount(self):
        return self.cycleCount
    def incrementCycleCount(self):
```

Рисунок Б.2 – Класс IntersectionController созданный для обучения модели Q-learning и их взаимодействия с симуляционной моделью

## Продолжение Б

```
def incrementCycleCount(self):
    self.cycleCount += 1
def setTrafficLightLogic(self, grDurNS, grDurWE):
    '''
    It is assumed that the number of phases does not change and always equal to 4 for all controlled intersections.
    The indexation starts from 0.
    '''
    gyrDef = traci.trafficlight.Logic('SP'+self.id,0,0,
    [
        traci.trafficlight.Phase(grDurNS, 'GrGr', grDurNS, grDurNS),
        traci.trafficlight.Phase(3, 'ryr', 3, 3),
        traci.trafficlight.Phase(grDurWE, 'rGrG', grDurWE, grDurWE),
        traci.trafficlight.Phase(3, 'ryr', 3, 3),
    ])
    traci.trafficlight.setProgramLogic(self.id, gyrDef)
    traci.trafficlight.setPhase(self.id,0)
    traci.trafficlight.setProgram(self.id, 'SP'+self.id)
def setspeed(self, speed):
    for edge in self.getEdgeIds()['NS']:
        traci.edge.setMaxSpeed(edge, speed)
    for edge in self.getEdgeIds()['SN']:
        traci.edge.setMaxSpeed(edge, speed)
    for edge in self.getEdgeIds()['WE']:
        traci.edge.setMaxSpeed(edge, speed)
    for edge in self.getEdgeIds()['EW']:
        traci.edge.setMaxSpeed(edge, speed)
def delayOfCarsInDir(self, direction):
    vehicleDelay = 0
    for edge in self.getEdgeIds()[direction]:
        vehicleDelay += traci.edge.getWaitingTime(edge)
    return vehicleDelay
def updateMeasures(self, step):
    self.measurements=[]
    self.measurements.append([
        step,
        self.delayOfCarsInDir('NS'),
        self.delayOfCarsInDir('SN'),
        self.delayOfCarsInDir('WE'),
        self.delayOfCarsInDir('EW'),
        self.signalPlan['NS'],
        self.signalPlan['WE'],
        traci.trafficlight.getPhase(self.id),
    ])
    ])
```

Рисунок Б.3 – Класс IntersectionController созданный для обучения модели Q-learning и их взаимодействию с симуляционной моделью

## Продолжение Б

```
import os,sys
import subprocess
PORT = 8873
interPhase = 3
signalPlans = []
for i in range(10):
    for j in range(10):
        sp = [12+i, 12+j]
        signalPlans.append(sp)
qtable={}
betta=-0.1
for speed in range(7,18):
    state=[]
    spIndx = 0
    times=[]
    rewards=[]
    while spIndx<100:
        step = 0
        node0 = IntersectionController('2', {'NS':['d7'], 'SN':['d9'], 'WE':['d5'],'EW':['d1']}, interPhase,{'NS':signalPlans[spIndx]})
        node1 = IntersectionController('3', {'NS':['d11'], 'SN':['d13'], 'WE':['d4'],'EW':['d2']}, interPhase,{'NS':signalPlans[spIndx]})
        node0.setTrafficLightLogic(signalPlans[spIndx][0],signalPlans[spIndx][1])
        node1.setTrafficLightLogic(signalPlans[spIndx][0],signalPlans[spIndx][1])
        node0.setspeed(speed)
        node1.setspeed(speed)
        dt=[0,0,0,0]
        while step < 120:
            traci.simulationStep()
            node0.updateMeasures(step)
            node1.updateMeasures(step)
            if (node0.measurements[0][3]!=0):
                dt[0]+=1
            if (node0.measurements[0][4]!=0):
                dt[1]+=1
            if (node1.measurements[0][3]!=0):
                dt[2]+=1
            if (node1.measurements[0][4]!=0):
                dt[3]+=1
            step+=1
        state.append(dt+signalPlans[spIndx])
        reward=dt[0]+dt[1]+dt[2]+dt[3]
        rewards.append(reward)
        spIndx+=1
    qtable[speed] = rewards
traci.close()
print(qtable)
```

Рисунок Б.4 – Скрипт обучающейся модели Q-learning

## Приложение В

```
for sm in range(3,7):
    for speed in range(7,18):
        solsp=[]
        state=[]
        spIndx = 0
        times=[]
        rewards=[]
        while spIndx<100:
            if table[speed][spIndx]==0:
                step = 0
                solsp.append(signalPlans[spIndx])
                node0 = IntersectionController('2', {'NS':['d7'], 'SN':['d9'], 'WE':['d5'],'EW':['d1']}, interPhase,{'NS':signalPlans[spIndx][0], 'SN':signalPlans[spIndx][1]})
                node1 = IntersectionController('3', {'NS':['d11'], 'SN':['d13'], 'WE':['d4'],'EW':['d2']}, interPhase,{'NS':signalPlans[spIndx][0], 'SN':signalPlans[spIndx][1]})
                node0.setTrafficLightLogic(signalPlans[spIndx][0],signalPlans[spIndx][1])
                node1.setTrafficLightLogic(signalPlans[spIndx][0],signalPlans[spIndx][1])
                node0.setspeed(speed)
                node1.setspeed(speed)
                dt=[0,0,0,0]
                while step < 120:
                    if step==0:node0.setTrafficLightLogic(signalPlans[spIndx][0],signalPlans[spIndx][1])
                    if step==sm:node1.setTrafficLightLogic(signalPlans[spIndx][0],signalPlans[spIndx][1])
                    traci.simulationStep()
                    node0.updateMeasures(step)
                    node1.updateMeasures(step)
                    if (node0.measurements[0][3]!=0): dt[0]+=1
                    if (node0.measurements[0][4]!=0): dt[1]+=1
                    if (node1.measurements[0][3]!=0): dt[2]+=1
                    if (node1.measurements[0][4]!=0): dt[3]+=1
                    step+=1
                state.append(dt+signalPlans[spIndx])
                reward=dt[0]+dt[1]+dt[2]+dt[3]
                rewards.append(reward)
            spIndx+=1
        print(solsp)
        qtable[sm, speed] = rewards
traci.close()
print(qtable)
```

Рисунок В.1 – Скрипт модели Q-learning