

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА УПРАВЛЕНИЯ ПЕРЕКРЕСТКОМ АВТОМОБИЛЬНЫХ ДОРОГ

Е.И. Сукач, М.В. Биза

Гомельский государственный университет имени Франциска Скорины

INTELLIGENT CONTROL SYSTEM FOR ROAD INTERSECTION

E.I. Sukach, M.V. Biza

Francisk Skorina Gomel State University

Аннотация. Иллюстрируется подход к созданию интеллектуальных систем управления объектами с использованием машинного обучения с подкреплением на примере системы управления перекрестком. Описывается имитационная модель перекрестка, выбранная в качестве среды обучения. Приводятся результаты сравнительного анализа эффективности работы различных алгоритмов обучения. Приводятся результаты применения градиента политики Монте-Карло для обучения модели перекрестка.

Ключевые слова: транспортная сеть, машинное обучение с подкреплением, нейронные сети, пропускная способность, безопасность, системы управления, градиент политики.

Для цитирования: Сукач, Е.И. Интеллектуальная система управления перекрестком автомобильных дорог / Е.И. Сукач, М.В. Биза // Проблемы физики, математики и техники. – 2023. – № 4 (57). – С. 87–93. – DOI: https://doi.org/10.54341/20778708_2023_4_57_87. – EDN: UQUVAZ

Abstract. An approach to the creation of intelligent object control systems using machine learning with reinforcement is illustrated using the example of an intersection control system. The simulation model of the intersection, chosen as the learning environment, is described. The results of a comparative analysis of the performance of various learning algorithms are presented. The results of applying the Monte Carlo policy gradient to train the intersection model are presented.

Keywords: transport network, reinforcement learning, neural networks, throughput, security, control systems, policy gradient.

For citation: Sukach, E.I. Intelligent control system for road intersection / E.I. Sukach, M.V. Biza // Problems of Physics, Mathematics and Technics. – 2023. – № 4 (57). – P. 87–93. – DOI: https://doi.org/10.54341/20778708_2023_4_57_87 (in Russian). – EDN: UQUVAZ

Введение

Перекресток автомобильных дорог является одним из самых загруженных участков транспортной сети (ТС), на функционирование которого оказывает влияние много случайных факторов. Управление обслуживанием транспортных и пешеходных потоков обычно регулируется светофорами, параметры работы которых зачастую не учитывают динамику изменения интенсивности динамических потоков. В результате образуются транспортные пробки, приводящие к росту временных и денежных затрат на перемещение транспортных единиц, росту аварий и ухудшению экологической обстановки. Поэтому актуальна проблема разработки программных средств, позволяющих обосновать выбор режима управления движением на перекрестке с учетом изменяющейся транспортной и пешеходной нагрузки на исследуемый участок ТС.

Одним из наиболее часто используемых методов оценки пропускной способности ТС является метод имитационного моделирования, позволяющий исследовать динамику функционирования объектов на высоком уровне детализации

с учетом вероятностных параметров перемещения транспортных единиц и изменяющихся условий внешней среды [1]. Разработанные авторами модели позволяют выбрать рациональный вариант структурной организации функционирования ТС с учетом заданного пользователем критерия, определить величину максимального потока и распределение транзитных потоков по сети с учетом вероятностных параметров нагрузки на сеть. При этом динамика обслуживания транспорта на перекрестках не исследовалась. Разработка детальной имитационной модели перекрестка автомобильных дорог позволит пополнить библиотеку параметризованных имитационных моделей программного комплекса «TRANZIT» [2] и расширить класс решаемых задач с использованием новой модели.

В статье рассматриваются вопросы создания интеллектуальной системы управления перекрестком автомобильных дорог с использованием обучения с подкреплением. Основой разработанной системы является имитационная модель «Перекресток», реализованная в современной среде имитационного моделирования

AnyLogic Personal Learning Edition 8.8.2, представляющей огромный выбор компонентов и функционала для создания имитационных моделей [3]. Модель «Перекресток» выступает в качестве среды для реализации обучения с подкреплением (RL, reinforcement learning) – области машинного обучения, в которой обучение происходит через взаимодействия с окружающей средой. При этом обучаемый получает информацию о последствиях своих действий, а не о том, какие действия следует выполнять [4].

Применение современных средств разработки программного обеспечения [5] и опыт разработчиков позволили реализовать различные алгоритмы обучения с подкреплением, которые были использованы для обучения среды «Перекресток», превращающие её в интеллектуальную систему, адаптирующуюся под изменяющуюся транспортную нагрузку.

1 Имитационная модель «Перекресток»

Для создания модели перекрёстка был использован участок дороги, представленный на рисунке 1.1. Предполагалось, что пересекающиеся дороги имеют двустороннее движение, по две полосы движения в каждом направлении. С помощью библиотеки «Презентация» среды моделирования было добавлено изображение, которое использовалось в качестве основы будущей модели «Перекрёсток».

Для организации моделирования перекрёстка была выполнена формализация объекта: размечены дороги, настроены светофоры, задана интенсивность поступления потоков машин и параметры появления потоков пешеходов.

Блок «Дорога» из Библиотеки дорожного движения [6] позволил сделать необходимую разметку дороги. В свойствах созданной дороги было выбрано число полос основного и встречного движения, определён цвет и покрытие дороги, задана ширина дороги и другие её параметры.

Случайный процесс перемещения транспортных средств и пешеходов на перекрёстке также был определён с использованием Библиотеки

дорожного движения среды AnyLogic, позволяющей планировать и прогнозировать перемещение динамических потоков в ТС. Описание объекта было выполнено с использованием диаграмм процессов, которые формировались с использованием функции ‘drag-and-drop’. Динамические единицы (транспортные средства и пешеходы) представлены в виде агентов, имеющих индивидуальные физические параметры.

Для описания логики движения пешеходов были созданы агенты «Pedestrian», которые затем были использованы в элементах, участвующих в моделировании пешеходных потоков и задающих интенсивность появления пешеходов, направление их перемещения, скорость перемещения и другие параметры. Элемент «Ped Source» отвечал за создание пешеходов, «Ped Select Out» определял направления пешеходов по определённым заранее вероятностям или условиям на один из пяти выходных портов, «Ped Sink» служил для удаления пешеходов, «Ped Go To» переводил пешеходов в заданное заранее с помощью целевой линии, области или точки место моделируемого пространства. Движение пешехода заканчивалось при пересечении заданного конечного пункта. С помощью элемента «Многоугольный узел» секции «Разметка пространства» было задано допустимое поведение пешеходов на дороге и на светофорах, указано, при каких фазах пешеходам можно переходить дорогу, а при каких не следует (рисунок 1.2).

Формирование транспортных потоков было задано с использованием элементов «Car Source», «Car Move To», «Car Dispose» Библиотеки дорожного движения. Элемент «Car Source» создаёт автомобили и помещает их в указанное место дорожной сети, «Car Move To» управляет движением созданных машин, а «Car Dispose» удаляет данные машины из модели. Автомобиль едет только тогда, когда он находится в блоке «Car Move To». В свойствах этих элементов была задана интенсивность прибытия транспортных единиц, указывалось, на какой дороге они появляются, по какой полосе едут и

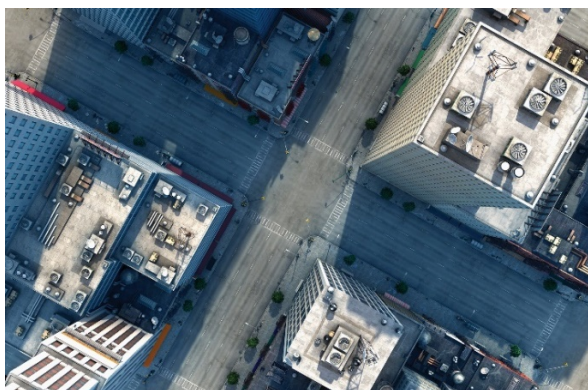


Рисунок 1.1 – Фотография перекрёстка автомобильных дорог

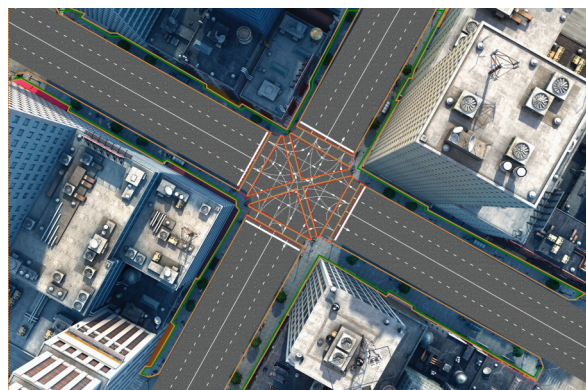


Рисунок 1.2 – Разметка модели «Перекрёсток»

куда направляются, определены параметры поведения динамических единиц (агентов) при поступлении в сеть, не нахождении пути и при выходе из сети.

Диаграммы, отражающие пример логики движения машин в модели «Перекрёсток», приведены на рисунках 1.3 и 1.4 (данные диаграмм задают одно и то же движение транспортных потоков).

Для регулирования движения транспорта была определена логика работы светофоров. Для этого в секцию «Блоки» Библиотеки дорожного движения был добавлен блок «Traffic Light». В свойствах «Traffic Light» были рассмотрены различные режимы работы для стоп-линий и для соединителей полос перекрёстка. Варьировались настройки длительности каждой из фаз светофоров. При запуске моделирования указывались начальные значения времени фаз светофоров, во время имитации эти значения изменялись, что позволило оптимизировать работу светофоров. В блоке «Road Network Description» были включены отображения пробок для более наглядной

симуляции дорожного движения. Кроме того, данный блок использовался для получения доступа к управлению транспортными единицами, находящимися в моделируемой дорожной сети. Были исследованы возможности выбора различных действий, которые выполнялись при добавлении автомобиля в дорожную сеть, въезде на дорогу и др.

Для сбора статистики по времени перемещения всех машин в ТС была создана популяция агентов с параметром «времяПоявления», для которых вычислялось «времяПроезда» при завершении их перемещения в сети. Для отображения результатов моделирования использовалась «Гистограмма» из секции «Диаграммы» палитры «Статистика». Для более чёткого понимания о загруженности ТС в целом вычислялось среднее время проезда, которое отображалось на графике. Кроме этого, вычислялось время, которое транспортные единицы провели на определённых участках перекрёстка и которое в процессе имитации отображалось на графиках.



Рисунок 1.3 – Диаграмма логики движения транспорта на перекрёстке

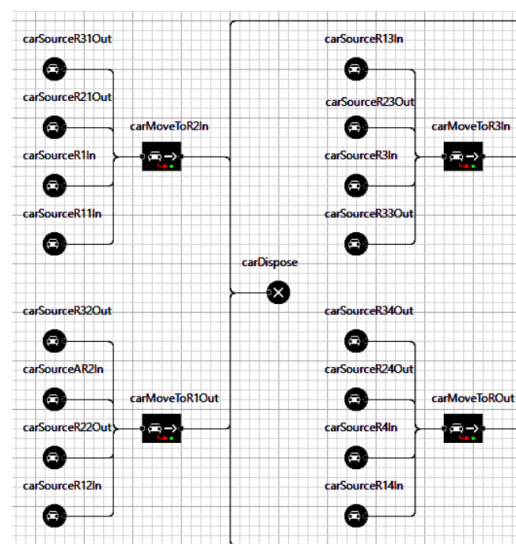


Рисунок 1.4 – Оптимизированная диаграмма логики движения транспорта на перекрёстке



Рисунок 1.5 – Модель перекрёстка в процессе имитации (2D)

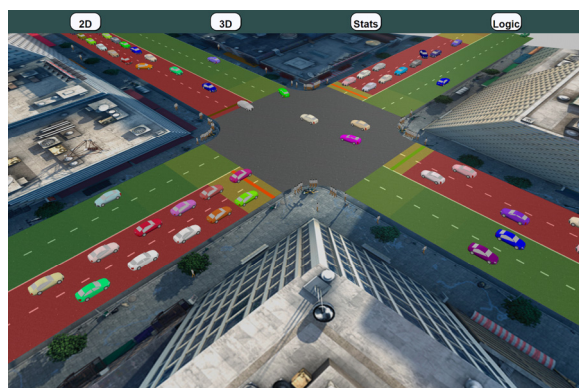


Рисунок 1.6 – Модель перекрёстка в процессе имитации (3D)

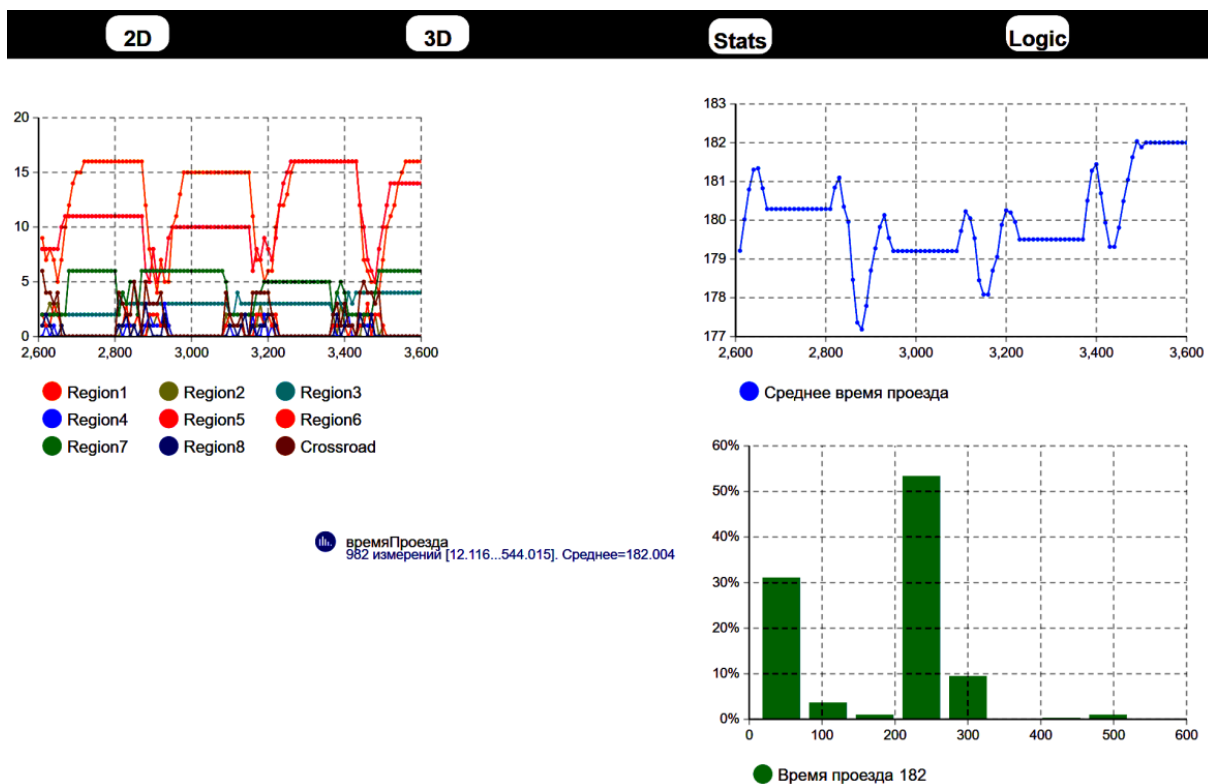


Рисунок 1.7 – Результаты моделирования дорожной ситуации с использованием модели «Перекрёсток» и логики дорожного движения, изображённой на рисунке 1.3

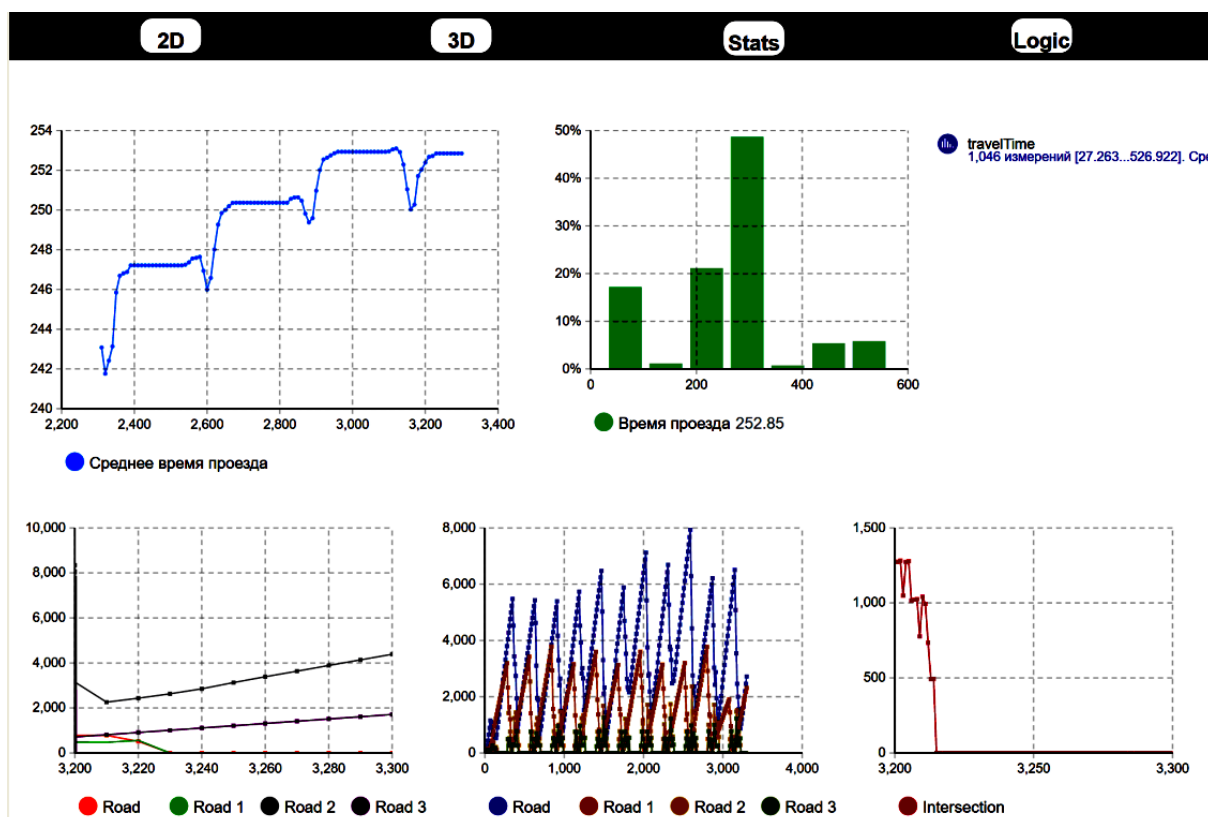


Рисунок 1.8 – Результаты моделирования дорожной ситуации с использованием модели «Перекрёсток» и логики дорожного движения, изображённой на рисунке 1.4

Благодаря управляющим элементам из секции «3D» библиотеки «Презентация» была установлена камера и создана трёхмерная анимация, отражающая процесс перемещения разных транспортных единиц (автомобилей, грузовиков, фур, автобусов и др.). Также для удобства просмотра имитации на заранее обозначенных областях перекрёстка была реализована панель навигации, позволяющая переключаться между режимами отображения имитации.

После запуска имитации (рисунки 1.3, 1.4) и проведения её в течение одного часа были сформированы результаты моделирования (рисунки 1.7 и 1.8).

На верхнем левом графике рисунка 1.7 отображено число машин (ось Y) на различных участках дороги в текущий момент времени моделирования в секундах (ось X). На графике «Среднее время проезда» отображено среднее время проезда машин на всей дороге (ось Y) в определённые моменты времени моделирования в секундах (ось X). Ниже представлена гистограмма, представляющая функцию плотности распределения времени перемещения транспортных единиц (ось X).

На левом верхнем графике рисунка 1.8 представлено среднее время проезда всех машин на дороге (ось Y), находящихся в моделируемом перекрестке в определённый момент времени (в секундах) симуляции (ось X). Справа представлена гистограмма «Время проезда» представляющая функцию плотности распределения времени перемещения транспортных единиц (ось X). На нижних графиках (слева направо) показано время в секундах (ось Y), которое провели в сумме все транспортные единицы, движущиеся в системе (от перекрёстка, к перекрёстку и на перекрёстке, соответственно) в определённый момент времени моделирования в секундах (ось X).

2 Создание интеллектуальной системы «Перекресток»

Для организации интеллектуального управления моделированием в AnyLogic был создан эксперимент «Обучение ИИ», в котором были заданы свойства и параметры, используемые для обучения ИИ-агента (агента верхнего уровня) в рамках обучения с подкреплением (RL). Данный эксперимент использовался для выгрузки готовых моделей на платформы, предназначенные для работы с ИИ, с целью обучения с подкреплением.

Для подготовки модели к обучению были заполнены такие поля эксперимента, как «Наблюдение», «Действие» и «Конфигурация». В разделе «Наблюдение» были указаны значения, которые передавались ИИ-агенту в ходе обучения для дальнейшего анализа. В разделе «Действие» были заданы значения, которые ИИ-агент рассчитывает в ходе «эпизода» модели, а затем

назначает переменным или функциям внутри модели прежде, чем перейти к следующему «эпизоду». В разделе «Конфигурация» был сформирован набор значений, определяющих начальное состояние модели перед «эпизодом». Наконец, в разделе «Модельное время» было указано, сколько времени будет проходить имитация.

Для наблюдения были выбраны данные, состоящие из времени, которое агенты провели на определенных участках дороги («getCars(true)» – автомобили, движущиеся по направлению к перекрёстку, «getCars(false)» – автомобили, движущиеся от перекрёстка), фаза светофора, время фаз светофора и текущее время. Действие, которое ИИ-агент может произвести, определялось вектором из 7 целых чисел, которые определяли, что именно делать со временем определённой фазы светофора: уменьшать, увеличивать или ничего не менять. Начальное состояние модели определялось четырьмя параметрами: промежутком времени, через который нужно производить действие («time_between_actions»), и значением времени для каждой фазы светофора.

В качестве награды использовалось значение выражения, полученное в результате вычитания суммы времени перемещения транспортных потоков, полученной до выполненного действия ИИ-агента, от суммы времени, полученной после выполнения действия. При этом суммировалось время, которое транспортные единицы провели, двигаясь по направлению к светофору (на четырех участках дороги) и на перекрёстке.

Реализованная модель «Перекрёсток» и заданный эксперимент «Обучение ИИ» был экспортирован в Microsoft Bonsai для проведения обучения. Для того, чтобы подключить экспортированную модель в PyCharm, использовалась библиотека Alpyne. Она позволила получить доступ к данным имитационной модели и провести обучение.

3 Сравнительный анализ реализованных алгоритмов машинного обучения

В области обучения с подкреплением появляется множество алгоритмов, позволяющих эффективно обучать агентов. Некоторые из них были реализованы и протестированы в игровых средах OpenAI Gym. Кратко остановимся на выводах, которые были сделаны при использовании реализованных алгоритмов.

Такие известные методы обучения, как Q-learning и Deep Q-learning, изучают функцию значения, которая показывает ожидаемую сумму вознаграждений, заданных состоянием и действием обучаемого агента.

Q-Learning – это методика обучения с подкреплением без использования моделей. Она обычно считается «самым простым» алгоритмом обучения с подкреплением. Q-Learning использует

ранее изученные «состояния», которые были исследованы для рассмотрения будущих ходов, и сохраняет эту информацию в «Q-таблице». Для каждого действия, предпринятого из состояния, Q-table должна включать положительное или отрицательное вознаграждение. Эта форма обучения отлично подходит, когда количество ходов ограничено или среда не сложная, поскольку агент запоминает прошлые ходы и с легкостью повторяет их. Однако для более сложных сред со значительно большим количеством состояний Q-table быстро заполняется, что приводит к увеличению времени обучения.

Использование Deep Q-learning предполагает наличие некоторого хранилища с определенным размером, где хранятся последние N опытов агента. При обучении используется случайная выборка определенного размера из памяти воспроизведения и применяется обновление Q-learning. После воспроизведения опыта агент выбирает и выполняет действие в соответствии с ϵ -жадной политикой.

Преимуществами данного алгоритма является то, что каждый шаг опыта потенциально используется во многих обновлениях весов нейронной сети, что позволяет повысить эффективность данных. Использование случайных выборок нарушает корреляции между выборками и, следовательно, уменьшает дисперсию в обновлениях.

Данные методы имеют много преимуществ и хорошо обучают агентов. Однако можно выделить несколько значительных проблем: они могут иметь большие колебания во время обучения; возникают трудности при большом наборе возможных действий; нужно реализовывать компромисс между разведкой и эксплуатацией.

Все эти проблемы можно решить при помощи градиентов политики. Градиент политики (PG) – это подход к RL, оптимизирующий параметризованную модель политики для ожидаемой отдачи с использованием градиентного подъема. К преимуществам градиента политики можно отнести следующее: при обучении агента просто следуют градиенту в отношении самой политики, что означает, что постоянно совершенствуется сама политика; параметры настраиваются напрямую; имеется возможность выучить стохастические политики [7]. Недостатком градиентов политики является то, что большую часть времени они сходятся на локальном максимуме, а не на глобальном оптимуме, и их обучение может занять много времени. Но и эти проблемы можно решить правильно подобрав политику.

Для обучения модели «Перекресток» использовался градиент политики Монте-Карло. При обучении выполнялись следующие действия: вычислялась логарифмическая вероятность, полученная с помощью функции политики; полученное значение умножалось на функцию оценки; выполнялось обновление веса. Было

обнаружено, что проблема данной политики заключается в усреднении всех действий. Даже если некоторые из них были очень плохими, а балл в итоге получался высоким, данные действия оценивались как хорошие. Из этого следует, что для получения правильной политики нужно произвести много экспериментов. А это приведёт к медленному обучению.

4 Результаты обучения среды «Перекресток»

С использованием AnyLogic Cloud был проведен сравнительный анализ полученных результатов моделирования среды до и после обучения. Для этого были заданы входные и выходные данные в «Конфигурации запуска» модели. Затем модель была экспортирована в облако и создано два эксперимента. В первом эксперименте в качестве входных данных были введены начальные значения, а во втором – результаты, полученные после обучения модели (рисунок 4.1).

После проведения экспериментов с помощью режима сравнения были сопоставлены полученные результаты, которые представлены на рисунках 4.2, 4.3. Из них хорошо видно, что на исследуемом участке сети после обучения среднее время проезда транспортных единиц, как и время нахождения на перекрестке, уменьшилось, то есть пропускная способность перекрестка увеличилась.

Входные данные	
Phase 1	60
Phase 2	60
Phase 3	120
Intensity of arrival of ...	10000
Intensity of arrival of ...	4363454
Intensity of arrival of ...	11111
Intensity of arrival of ...	904876

Рисунок 4.1 – Входные данные для двух экспериментов

Среднее время проезда транспорта уменьшилось с 219,744 секунд до 137,718 секунд. Максимальное время перемещения транспортных единиц было сокращено с 526,922 секунд до 339,384 секунд.

Полученные результаты свидетельствуют об эффективности обучения с использованием метода градиента политики Монте-Карло. Алгоритм стабилен и обладает хорошими свойствами сходимости, однако наблюдается чувствительность к локальным минимумам.

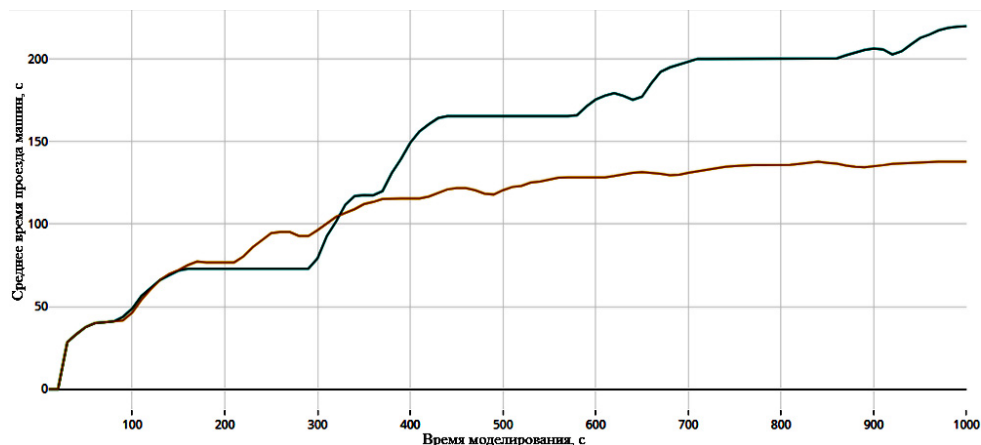


Рисунок 4.2 – Среднее время перемещения транспортных единиц

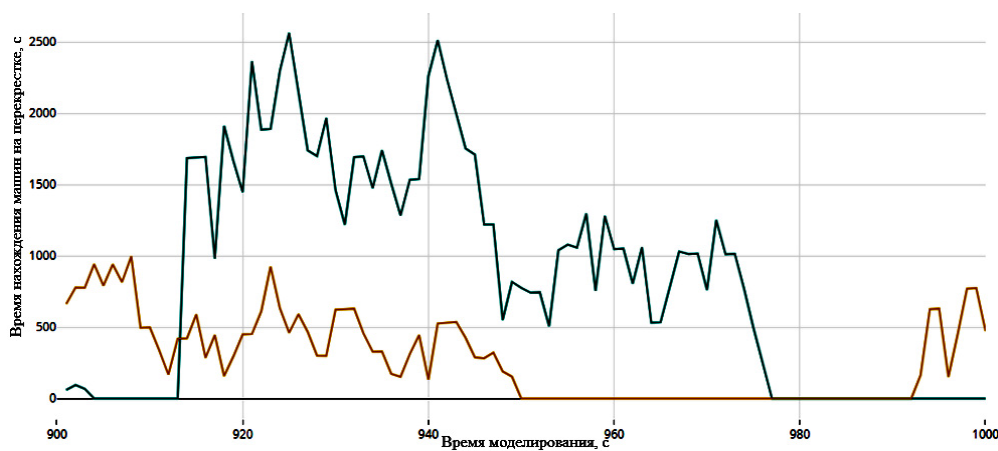


Рисунок 4.3 – Время, которое провели машины на перекрестке

Заключение

Реализованная интеллектуальная система управления перекрестком автомобильных дорог параметризована и при ее небольшой адаптации к реальному объекту моделирования может быть использована для организации эффективного управления на перекрестках, на которых на одном уровне пересекаются несколько потоков транспорта и пешеходов. Это обеспечит оперативность в организации обслуживания транспортных потоков, повысит безопасность перемещения пешеходов и позволит оптимизировать работу транспортной системы в целом.

ЛИТЕРАТУРА

1. Сукач, Е.И. Применение имитационных моделей для принятия решений в системах управления на транспорте / Е.И. Сукач, В.Н. Галушко, П.В. Гируц // Математические машины и системы. – 2010. – № 1. – С. 185–192.
- 2 Сукач, Е.И. Метод перераспределения автомобильных транспортных потоков региона на основе имитационного моделирования / Е.И. Сукач // Реєстрація, зберігання і обробка даних (Data Recording, Storage & Processing). – 2008. – Т. 10, № 3. – С. 37–45.
- 3 Карпов, Ю.Г. Имитационное моделирование систем. Введение в моделирование с

AnyLogic 5 / Ю.Г. Карпов. – СПб.: БХВ-Петербург, 2009. – 390 с.

4. Саттон, Ричард С. Обучение с подкреплением: введение. 2-е изд. / Ричард С. Саттон, Эндрю Дж. Барто; пер. с англ. А.А. Слинкина. – Москва: ДМК Пресс, 2020. – 552 с.

5. Судхарсан, Равичандиран. Глубокое обучение с подкреплением на Python. OpenAI Gym и TensorFlow для профи / Равичандиран Судхарсан. – СПб.: Питер, 2019. – 251 с.

6. Библиотека дорожного движения [Электронный ресурс]. – Режим доступа: <https://www.anylogic.ru/>. – Дата доступа: 15.04.2023.

7. Биза, М.В. Применение градиентов политики в ходе обучения с подкреплением / М.В. Биза, Е.И. Сукач // Материалы XI республиканской научно-практической конференции «Вычислительные методы, модели и образовательные технологии», Брестский государственный университет имени А.С. Пушкина, 22 октября 2022 г., Брест, 2022. – С. 124–125.

Поступила в редакцию 30.06.2023.

Информация об авторах

Сукач Елена Ивановна – к.т.н., доцент
Биза Мария Викторовна – студентка