

## МОДЕЛИРОВАНИЕ АДАПТИВНОГО СВЕТОФОРНОГО РЕГУЛИРОВАНИЯ ДЛЯ ОПТИМИЗАЦИИ ТРАНСПОРТНЫХ ПОТОКОВ НА ПЕРЕКРЕСТКЕ

Кузьмин Д.В.<sup>1</sup>, Багинова В.В.<sup>1</sup>, Коробовский К.М.<sup>1</sup>, Торицын А.М.<sup>1</sup><sup>1</sup> Российский университет транспорта

**Аннотация:** в статье рассматривается применение среды имитационного моделирования AnyLogic для исследования и оптимизации работы регулируемого перекрестка. Основная цель исследования — разработка и сравнительный анализ двух альтернативных алгоритмов адаптивного светофорного управления. Предлагаемые алгоритмы в режиме реального времени анализируют совокупность показателей транспортного потока (интенсивность, длину очереди, время ожидания) и на этой основе динамически регулируют продолжительность фаз светофорного цикла. Это позволяет провести виртуальные испытания алгоритмов в контролируемых условиях, объективно оценить их эффективность по ключевым метрикам (пропускная способность, среднее время задержки транспорта) и выявить наиболее устойчивый и производительный метод управления для различных сценариев загрузки. Статья представляет интерес для специалистов в области интеллектуальных транспортных систем (ИТС), инженеров организации дорожного движения, а также для исследователей и студентов, занимающихся вопросами моделирования и оптимизации городской мобильности. Результаты исследования могут быть использованы для практического обоснования внедрения адаптивных систем на проблемных перекрестках, дальнейшей разработки и калибровки более сложных алгоритмов управления, создания цифровых двойников транспортных узлов для планирования и принятия решений.

**Ключевые слова:** дискретно-событийное моделирование, автомобильный перекресток, автомобильный поток, транспортная система, оптимизация работы перекрестка.

© Кузьмин Д.В., Багинова В.В., Коробовский К.М., Торицын А.М.

Поступила 12.11.2025, одобрена после рецензирования 12.12.2025, принята к публикации 24.10.2025.

Для цитирования:

Кузьмин Д.В., Багинова В.В., Коробовский К.М., Торицын А.М. Моделирование адаптивного светофорного регулирования для оптимизации транспортных потоков на перекрестке // Логистика и управление цепями поставок. - 2025. - Т. 22, №4 (117). - С. 51–67.

Информация об авторах:

Кузьмин Д.В., — к.т.н., доцент, доцент кафедры «Логистика и управление транспортными системами», ФГБОУ ВО «Российский университет транспорта» РУТ (МИИТ), e-mail: kuzmindv@rut-miit.ru.

Багинова В.В., — д.т.н., профессор, профессор кафедры «Логистика и управление транспортными системами», ФГБОУ ВО «Российский университет транспорта» РУТ (МИИТ).

Коробовский К.М., — магистрант кафедры «Логистика и управление транспортными системами», ФГБОУ ВО «Российский университет транспорта» РУТ (МИИТ).

Торицин А.М., — магистрант кафедры «Логистика и управление транспортными системами», ФГБОУ ВО «Российский университет транспорта» РУТ (МИИТ).

## ВВЕДЕНИЕ

Эффективность работы городской транспортной сети во многом определяется пропускной способностью ее ключевых узлов – регулируемых перекрестков. Традиционные светофоры с жестко фиксированным циклом и длительностью фаз, заданными по усредненным историческим данным, часто оказываются неспособны адаптироваться к стохастической природе транспортного потока. Это приводит к неоптимальному распределению «зеленого» времени, возникновению избыточных задержек, росту очередей и, как следствие, к снижению общего уровня транспортного обслуживания, увеличению выбросов и непроизводительным затратам времени участников движения [1].

Актуальным решением данной проблемы является внедрение адаптивных (интеллектуальных) систем светофорного регулирования. В отличие от статических, такие системы

способны в режиме реального времени анализировать текущую ситуацию на подходах к перекрестку и динамически корректировать логику своей работы, стремясь минимизировать ключевые негативные показатели. Однако разработка, тестирование и сравнение эффективности различных адаптивных алгоритмов в реальных городских условиях сопряжена со значительными организационными и финансовыми рисками. В этой связи имитационное моделирование становится незаменимым инструментом, позволяющим в безопасной виртуальной среде создавать цифровые двойники перекрестков, разрабатывать и всесторонне оценивать различные стратегии управления.

Целью настоящего исследования является разработка и сравнительный анализ двух альтернативных алгоритмов адаптивного светофорного управления с использованием среды имитационного моделирования AnyLogic.

## ОПИСАНИЕ МОДЕЛИ

Модель реализована в виде схематичной пространственной разметки без детализации, воспроизводящей реальный объект, поскольку первостепенной задачей исследования является апробация работоспособности и оценка

эффективности предложенных алгоритмов управления. Соответственно, на рисунке 1 представлена условная схема классического четырёхстороннего перекрестка.

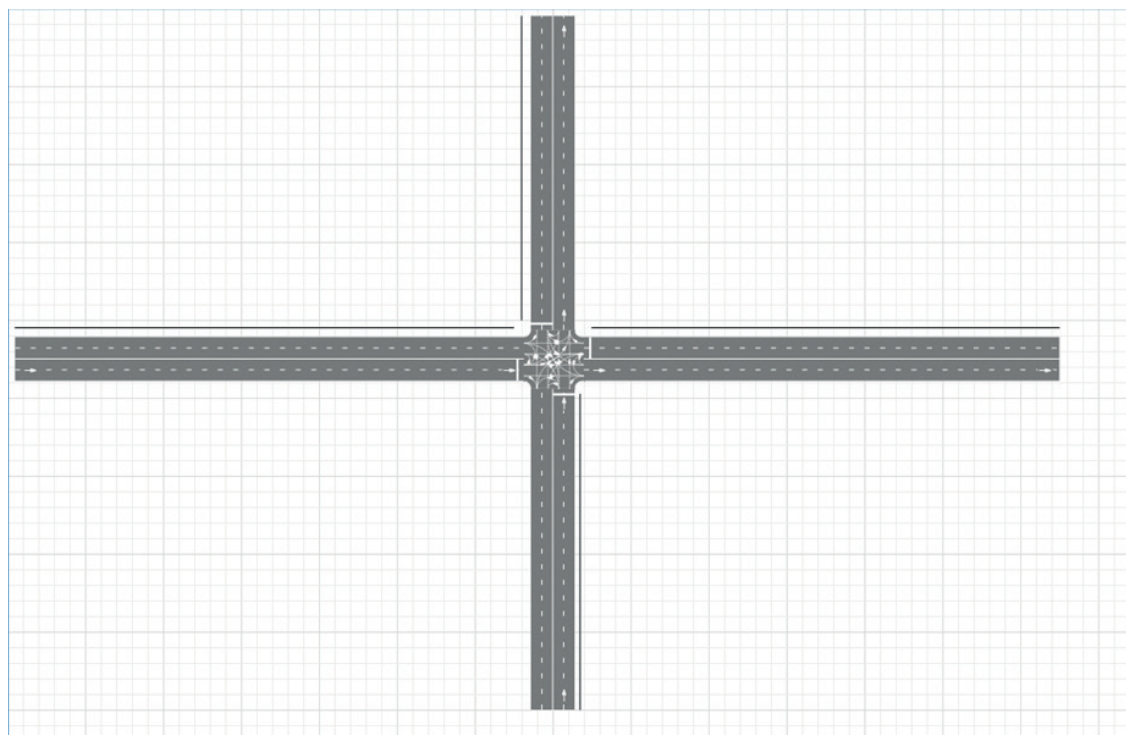


Рисунок 1 — Схема классического четырёхстороннего перекрестка.

Регулирование движения на перекрестке организовано с помощью четырёх светофорных объектов, расположение которых показано на рисунке 2. Работа системы реализуется в рамках двухфазного цикла:

- Фаза 1: Разрешающее движение (зелёный сигнал) предоставляется транспортным

средствам, следующим в вертикальном направлении.

- Фаза 2: Разрешающее движение (зелёный сигнал) предоставляется транспортным средствам, следующим в горизонтальном направлении.

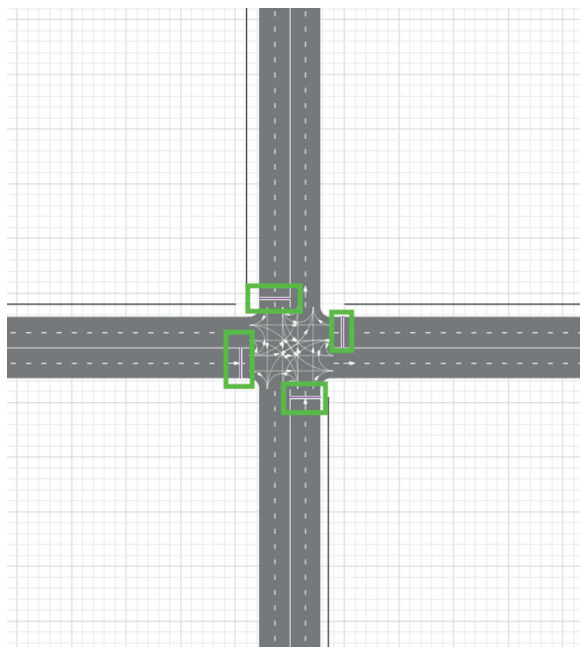


Рисунок 2 — Расположение светофорных объектов на перекрестке.

Модель включает в себя панель управления, предназначенную для настройки интенсивности транспортных потоков, поступающих с четырех направлений (север, юг, запад, восток). Данный инструмент позволяет проводить вычислительные эксперименты и оцени-

вать работоспособность адаптивных алгоритмов светофорного регулирования в широком диапазоне нагрузочных сценариев (рисунок 3), интенсивность потока со всех направлений может принимать значение от 0 до 1000 автомобилей в час.

ИНТЕНСИВНОСТЬ ПОТОКА АВТ/Ч.	
ЮГ	<input type="text" value=""/>
Север	<input type="text" value=""/>
Запад	<input type="text" value=""/>
Восток	<input type="text" value=""/>

Рисунок 3 — Вид панели управления интенсивностью потока.

В модель интегрированы элементы информационной визуализации (инфографики), обеспечивающие мониторинг ключевых показателей эффективности. К ним относятся: среднее время проезда перекрестка транспортными средствами и гистограмма распределения этого времени. Данные показатели

позволяют провести сравнительный анализ производительности адаптивных алгоритмов с эталонным случаем — классической системой со статическим светофорным регулированием, где длительность фаз жестко фиксирована и не зависит от текущей интенсивности потока (рисунок 4).

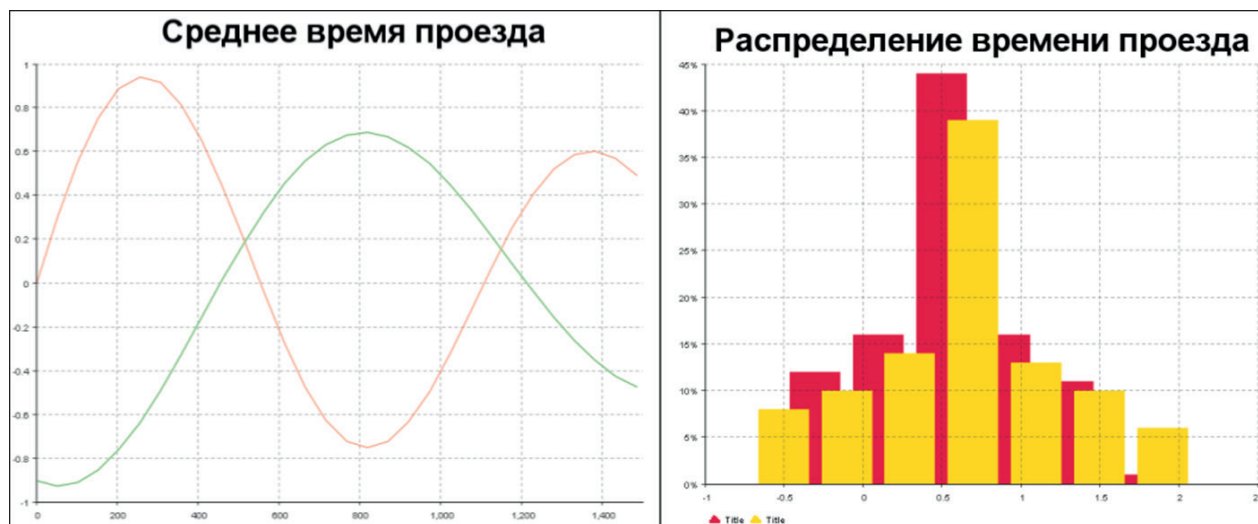


Рисунок 4 — Вид инфографики в модели.

Основу функционирования модели составляет диаграмма процессов, созданная с использованием встроенной автомобильной

библиотеки (Road Traffic Library) среды имитационного моделирования AnyLogic (рисунок 5).

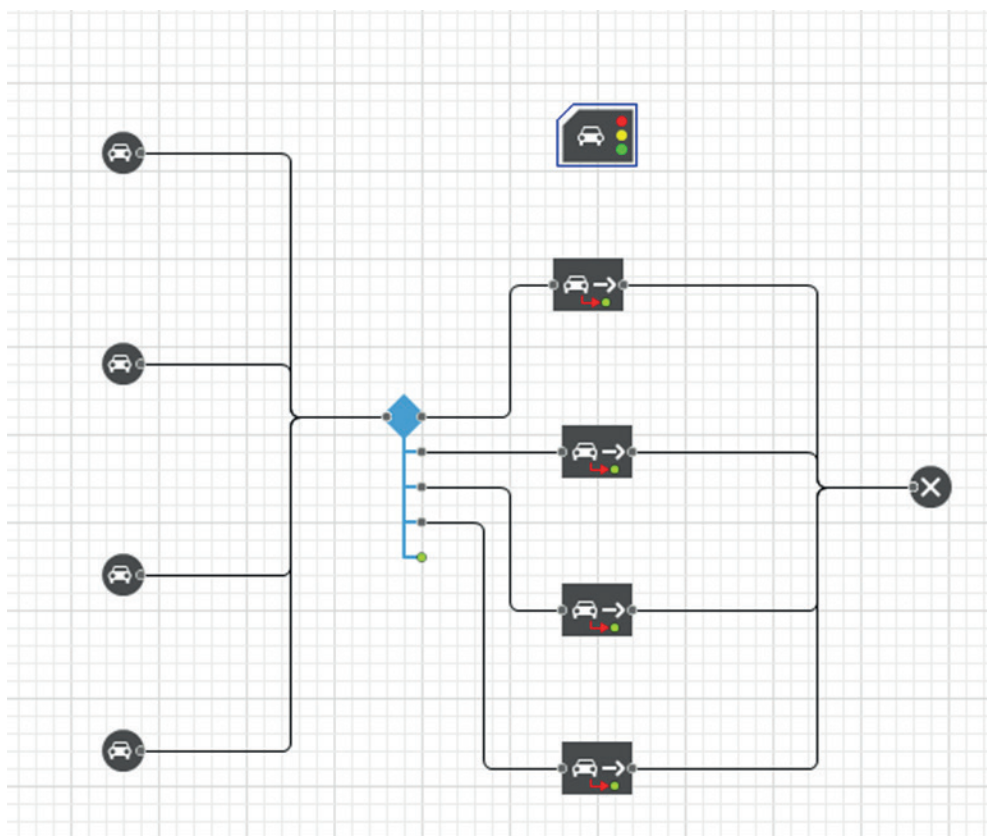

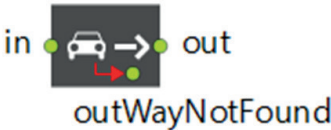





Рисунок 5 — Вид диаграммы процесса модели.

Основные функции каждого блока описаны в таблице 1.

Таблица 1

Функции блоков Anylogic, использованные для построения диаграммы процесса модели

Название блока	Графическое обозначение блока на диаграмме процесса (рис. 5)	Основная функция блока
Car Source		Создает автомобили и пытается поместить их в указанное место дорожной сети.
CarMoveTo		Блок, который управляет движением автомобиля.
Traffic Light		Моделирует светофор
SelectOutput5		Блок направляет входящих агентов в один из пяти выходных портов в зависимости от выполнения заданных (детерминистических или заданных с помощью вероятностей) условий.
CarDispose		Удаляет машины из модели.

В основе логики адаптивного светофорного алгоритма лежит комплексное использование параметров, функций и планируемых событий модели. Параметры обеспечивают конфигурацию и настройку системы, функции реализуют расчетные и управляющие проце-

дуры, а события, запланированные на будущий момент времени, используются для инициации ключевых операций, таких как анализ текущего состояния и переключение сигнальных фаз<sup>1</sup>.

## ОПИСАНИЕ АЛГОРИТМОВ

Для каждого из рассматриваемых алгоритмов управления было задано условное обозначение, используемое в дальнейшем изложении: Алгоритм 1 и Алгоритм 2.

Алгоритм 1 реализует концепцию динамического распределения времени «зеленого» сигнала на регулируемом перекрестке. В отличие от статического управления с фиксированными фазами, данный метод адаптирует дли-

<sup>1</sup> AnyLogic 9: Руководство пользователя [Электронный ресурс] / The AnyLogic Company. — Версия 2025 г. — Режим доступа: <https://anylogic.help/>. (Дата обращения: 06.11.2025).



тельность цикла светофора и пропорции фаз в реальном времени, основываясь на текущей загрузке подъездов.

Ключевая идея заключается в том, чтобы измерять плотность скопления транспортных средств (количество автомобилей) на каждой из конфликтующих направлений (например, горизонтальном и вертикальном). На основе этих данных система:

1. Корректирует общую длительность цикла светофорного регулирования, увеличивая её при высокой общей загрузке перекрестка для обслуживания большего числа автомобилей.

2. Пропорционально распределяет доступное «зеленое» время между фазами. Направление с более высокой плотностью движения получает более продолжительный разрешающий сигнал, что позволяет эффективнее разгружать образующиеся очереди.

3. Обеспечивает оперативную реакцию на изменение потока, гарантируя минимальное гарантированное время для каждой фазы и предотвращая неоправданно долгое ожидание на малозагруженных подходах.

Далее стоит рассмотреть работу первого алгоритма по этапам.

Этап 1. Измерение входных данных: оценка загрузки направлений

Алгоритм начинается с получения первичных данных — оценки текущей нагрузки на каждое из конфликтующих направлений. В данном случае направления условно обозначены как дорога А (например, горизонтальное) и дорога В (вертикальное).

```
double densityA = roadA.nCars(true);
```

```
double densityB = roadB.nCars(false);
```

Переменные densityA и densityB представляют собой количество автомобилей на заданном сегменте подхода к стоп-линии и являются основными входными переменными для всех последующих расчетов.

Общая плотность (*totalDensity*) вычисляется как сумма загрузок обоих направлений и отражает совокупную нагрузку на перекресток:

```
double totalDensity = densityA + densityB;
```

Этап 2. Адаптация коэффициента реакции и расчет общего времени цикла

На втором этапе определяется, насколько агрессивно система будет реагировать на увеличение загрузки. Для этого вводится коэффициент чувствительности *k*.

```
double k;
if (totalDensity <= maxCars) {
    k = k_min + (k_max - k_min) * totalDensity / maxCars; // Линейный рост
} else {
    k = k_max; // Достигнут верхний предел реакции
}
```

где:

- *maxCars* — пороговое значение общей плотности (например, 12 автомобилей).

- *k\_min*, *k\_max* — минимальное и максимальное значения коэффициента (например, 0.1 и 5.0).

Логика работы: При низкой загрузке (*totalDensity* ≤ *maxCars*) коэффициент *k* линейно возрастает от *k\_min* до *k\_max*. Это позволяет плавно увеличивать длительность цикла по мере роста числа автомобилей. При высокой загрузке (*totalDensity* > *maxCars*) коэффициент фиксируется на максимальном значении *k\_max*, предотвращая чрезмерное увеличение времени цикла.

Расчет общего времени цикла светофора (*totalCycleTime*) выполняется по формуле:

```
double totalCycleTime = baseTime + totalDensity * k;
```

где:

- *baseTime* — базовое время цикла (константа, определяющая длительность при нулевой загрузке, включая гарантированные минимумы и время желтого сигнала/паузы).

Данная формула показывает, что общее время цикла складывается из фиксированной базовой части и переменной части, пропорциональной текущей общей загрузке (*totalDensity*), взвешенной на адаптивный коэффициент *k*.

Этап 3. Распределение «зеленого» времени между фазами

На заключительном этапе рассчитанное общее время цикла *totalCycleTime* распределяется между фазами пропорционально их текущей загрузке.

А. Расчет веса (доли) для направления А:

Вес *weightA* представляет собой долю направления А в общей загрузке перекрестка.

```
double weightA = densityA / totalDensity;
```

Б. Расчет предварительной длительности «зеленого» для направления А:

Длительность фазы прямо пропорциональна загрузке соответствующего направления.

```
double newGreenA = totalCycleTime * weightA;
```

В. Наложение ограничений (защита от экстремальных значений):

```
newGreenA = Math.max(minGreen, Math.min(maxGreen, newGreenA));
```

где:

- *minGreen* — минимально гарантированная длительность «зеленого» сигнала (например, 10 сек.), обеспечивающая безопасный пропуск пешеходов и транспортных средств, подъехавших к самому началу фазы.

- *maxGreen* — максимально допустимая длительность «зеленого» сигнала (например, 90 сек.), предотвращающая недопустимо долгое ожидание на противоположном направлении.

Функция *Math.min (max Green, new GreenA)* ограничивает время сверху, *Math.max(minGreen, ...)* — снизу.

Г. Присвоение результата переменным светофора:

В зависимости от того, какое физическое направление представляет *roadA* (горизонтальное "road1"/"road3" или вертикальное), рассчитанное и ограниченное значение *newGreenA* присваивается соответствующей переменной длительности фазы (*tIH* для горизонтальной или *tIV* для вертикальной фазы). Если общая загрузка равна нулю (*totalDensity == 0*), каждой фазе устанавливается минимальное гарантированное время *minGreen*<sup>2</sup>.

Полный код Алгоритма 1

Логика, описанная выше, интегрирована в следующий программный блок:

```
double densityA = roadA.nCars(true);
double densityB = roadB.nCars(false);
int minGreen = 10;
int maxGreen = 90;
```

```
double totalDensity = densityA + densityB;
```

```
double k_min = 0.1;
```

```
double k_max = 5.0;
```

```
int maxCars = 12;
```

```
double k;
```

```
if (totalDensity <= maxCars) {
    k = k_min + (k_max - k_min) * totalDensity / maxCars;
} else {
    k = k_max;
}
```

```
double totalCycleTime = baseTime + totalDensity * k;
```

```
if (totalDensity > 0) {
    double weightA = densityA / totalDensity;
    double newGreenA = totalCycleTime * weightA;
    newGreenA = Math.max(minGreen, Math.min(maxGreen, newGreenA));
}
```

```
if (roadA.getName() == "road1" || roadA.getName() == "road3") {
    tIH = newGreenA;
} else {
    tIV = newGreenA;
}
} else {
    if (roadA.getName() == "road1" || roadA.getName() == "road3") {
        tIH = minGreen;
    } else {
        tIV = minGreen;
    }
}
```

Далее стоит дать описание алгоритма 1. Алгоритм реализует стратегию дискретной адаптации светофорного цикла, основанную на сравнении суммарной загрузки конфликтующих транспортных осей перекрестка. В отличие от первого алгоритма, который использует плавные математические зависимости, данный метод работает по принципу по-

<sup>2</sup> Руководства по языку Java: Управляющие конструкции, классы, многопоточность [Электронный ресурс] / Oracle Corporation. — Обновлено в 2024 г. — Режим доступа: <https://docs.oracle.com/javase/tutorial/>. (Дата обращения: 06.11.2025).



блочного перераспределения времени между фазами.

Основная логика строится на следующем принципе: время «зеленого» сигнала перераспределяется целыми блоками (например, по 5 секунд) от менее загруженного направления к более загруженному. Алгоритм постоянно сравнивает общее количество автомобилей на подходах, объединенных в две фазы — Север-Юг и Восток-Запад.

Ключевые особенности:

1. Осевой подход: Алгоритм не рассматривает каждое из четырех направлений по отдельности, а агрегирует их в две конфликтующие группы (оси), что соответствует классической двухфазной схеме организации движения.

2. Дискретное управление: Корректировка длительности фаз происходит не по непрерывной формуле, а фиксированными шагами. Это создает более стабильный и предсказуемый режим работы, удобный для реализации в простых контроллерах.

3. Принцип «подарка времени»: Если одна ось значительно загружена, алгоритм «одалживает» у противоположной фазы блок времени (в рамках установленных лимитов), стремясь быстро отреагировать на возникший дисбаланс.

4. Гарантированные минимумы и максимумы: Как и в первом алгоритме, существуют жесткие границы (*minGreen*, *maxGreen*), которые предотвращают как чрезмерно короткие, так и недопустимо долгие фазы, обеспечивая базовый уровень обслуживания и безопасности.

Аналогичным образом стоит рассмотреть Алгоритм 2 по этапам.

Этап 1. Сбор данных: подсчёт автомобилей по направлениям

Алгоритм начинает работу с измерения входных данных — количества транспортных средств, ожидающих на каждом из четырёх подходов к перекрёстку. Для этого используются счётчики автомобилей в соответствующих очередях (секциях накопления) модели.

```
int countNorth = (int) carMoveTo1.size(); //
int countEast = (int) carMoveTo2.size(); //
int countSouth = (int) carMoveTo3.size(); //
```

```
int countWest = (int) carMoveTo4.size(); //
• countNorth, countEast, countSouth,
countWest — это целочисленные переменные,
хранящие текущее количество автомобилей на
каждом подъезде. Эти значения являются ос-
новными сенсорными данными алгоритма.
```

Этап 2. Агрегация данных: объединение направлений в конфликтующие оси

Поскольку модель реализует классическую двухфазную схему движения, направления объединяются в две конфликтующие группы (оси). Это необходимо для корректного сравнения нагрузок, которые конкурируют за одно и то же время в цикле светофора.

```
int loadNorthSouth = countNorth +
countSouth;
```

```
int loadEastWest = countEast + countWest;
```

• *loadNorthSouth* — суммарная нагрузка на ось «Север–Юг» (фаза 1).

• *loadEastWest* — суммарная нагрузка на ось «Восток–Запад» (фаза 2).

• Этот этап преобразует четыре входных параметра в два ключевых, что упрощает логику принятия решения.

Этап 3. Логика принятия решения: сравнение и перераспределение

На этом этапе алгоритм определяет, какая из осей в данный момент более загружена, и выполняет перераспределение времени зелёного сигнала.

А. Определение базовых констант управления:

```
int minGreen = 10; // Минимальная га-
рантированная длительность зелёной фазы
```

```
int maxGreen = 80; // Максимально допу-
стимая длительность зелёной фазы
```

```
int stepChange = 5; // Шаг изменения
(квант времени для перераспределения)
```

```
int defaultGreen = 30; // Значение по умол-
чанию при равной нагрузке
```

Б. Логическое ветвление:

Алгоритм сравнивает *loadNorthSouth* и *loadEastWest* и действует по одному из трёх сценариев:

```
if (loadNorthSouth > loadEastWest) {
```

```
// Сценарий 1: Приоритет оси Север-Юг
greenNorthSouth = Math.min(maxGreen,
greenNorthSouth + stepChange);
```

```

    greenEastWest = Math.max(minGreen,
greenEastWest - stepChange);
}
else if (loadEastWest > loadNorthSouth) {
    // Сценарий 2: Приоритет оси Вос-
ток-Запад
    greenEastWest = Math.min(maxGreen,
greenEastWest + stepChange);
    greenNorthSouth = Math.max(minGreen,
greenNorthSouth - stepChange);
}
else {
    // Сценарий 3: Равная загрузка
    greenNorthSouth = defaultGreen;
    greenEastWest = defaultGreen;
}

```

• Сценарий 1 и 2: Если одна ось загружена больше другой, её зелёное время увеличивается на шаг *stepChange* (но не превысит *maxGreen*), а время противоположной оси уменьшается на тот же шаг (но не упадёт ниже *minGreen*). Это реализует принцип дискретного перераспределения (квантования) времени.

• Сценарий 3: При равной загрузке устанавливается равное время по умолчанию (*defaultGreen*), что обеспечивает справедливое обслуживание в условиях баланса<sup>3</sup>.

Этап 4. Исполнение решения: назначение новых длительностей фаз

Рассчитанные значения присваиваются управляющим переменным светофора, которые непосредственно определяют длительность следующих зелёных фаз в имитационной модели.

```

parameter = greenNorthSouth; // Длитель-
ность фазы "Север-Юг"
parameter1 = greenEastWest; // Длитель-
ность фазы "Восток-Запад"

```

## ПРОВЕДЕНИЕ ЭКСПЕРИМЕНТА С МОДЕЛЮ

Для проведения экспериментального исследования были выбраны три сценария, моделирующие различную нагрузку на перекресток:

• Сценарий низкой интенсивности: 200 автомобилей в час;

Алгоритм 2 имеет следующий общий вид.

```

int countNorth = (int) carMoveTo.size();
int countEast = (int) carMoveTo1.size();
int countSouth = (int) carMoveTo2.size();
int countWest = (int) carMoveTo3.size();

```

```

int greenNorthSouth = 0;
int greenEastWest = 0;

```

```

int minGreen = 10;
int maxGreen = 80;
int stepChange = 5;

```

```

int loadNorthSouth = countNorth +
countSouth;
int loadEastWest = countEast + countWest;

```

```

if (loadNorthSouth > loadEastWest) {
    greenNorthSouth = Math.min(maxGreen,
greenNorthSouth + stepChange);
    greenEastWest = Math.max(minGreen,
greenEastWest - stepChange);
}
else if (loadEastWest > loadNorthSouth) {
    greenEastWest = Math.min(maxGreen,
greenEastWest + stepChange);
    greenNorthSouth = Math.max(minGreen,
greenNorthSouth - stepChange);
}
else {
    int defaultGreen = 30;
    greenNorthSouth = defaultGreen;
    greenEastWest = defaultGreen;
}

```

```

parameter = greenNorthSouth;
parameter1 = greenEastWest;

```

• Сценарий средней интенсивности: 500 автомобилей в час;

• Сценарий высокой интенсивности: 1000 автомобилей в час.

Длительность каждого вычислительного эксперимента составляет 1500 единиц модельного времени, что обеспечивает выход систе-

<sup>3</sup> Руководства по языку Java: Управляющие конструкции, классы, многопоточность [Электронный ресурс] / Oracle Corporation. — Обновлено в 2024 г. — Режим доступа: <https://docs.oracle.com/javase/tutorial/>. (Дата обращения: 06.11.2025).

мы на установившийся режим работы и получение статистически значимых результатов.

Эффективность разработанных адаптивных алгоритмов оценивается путём сравнения с базовым (эталонным) вариантом управления — статическим светофором с фиксированной длительностью фаз. В качестве такого варианта принята конфигурация, соответствующая среднестатистическим настройкам на реальных перекрестках: длительность первой фазы составляет 45 секунд, второй — 50 секунд.

Сравнительный анализ будет проводиться по двум ключевым показателям:

1. Среднее время проезда перекрестка транспортным средством.

2. Распределение (вариабельность) времени проезда, отражающее стабильность работы системы.

Полученные результаты представлены в таблице 2.

Таблица 2

Результаты эксперимента с моделью

Интенсивность (авто/час)	Алгоритм	Среднее время проезда, сек.	80% автомобилей проезжают не более, чем за (сек.)
200	Алгоритм 1	22.1	25
	Алгоритм 2	22.8	26.5
	Статический светофор	24.5	29
500	Алгоритм 1	40.5	48
	Алгоритм 2	42.7	52
	Статический светофор	51.3	65
1000	Алгоритм 1	78.2	95
	Алгоритм 2	82.4	102
	Статический светофор	112.7	145

Проведённый эксперимент подтвердил гипотезу о существенном превосходстве адаптивных алгоритмов управления светофорным объектом над классическим статическим режимом. Сравнительный анализ по ключевым метрикам — среднему времени проезда и гарантированному времени проезда для 80% транспортных средств — позволил объективно оценить эффективность предложенных решений в различных сценариях нагрузки.

Адаптивные алгоритмы обеспечивают значительное снижение задержек. Во всех исследуемых сценариях (низкой, средней и

высокой интенсивности) оба адаптивных алгоритма показали лучшие результаты, чем статическое управление. Наибольший абсолютный выигрыш наблюдается в условиях насыщенного потока (1000 авто/час), где Алгоритм 1 сократил среднее время проезда на 34.5 секунды (31%), а Алгоритм 2 — на 30.3 секунды (27%) по сравнению со статическим режимом. Для наглядной иллюстрации данной зависимости на Рисунке 6 представлен график «Среднее время проезда в зависимости от интенсивности движения для трёх режимов управления».

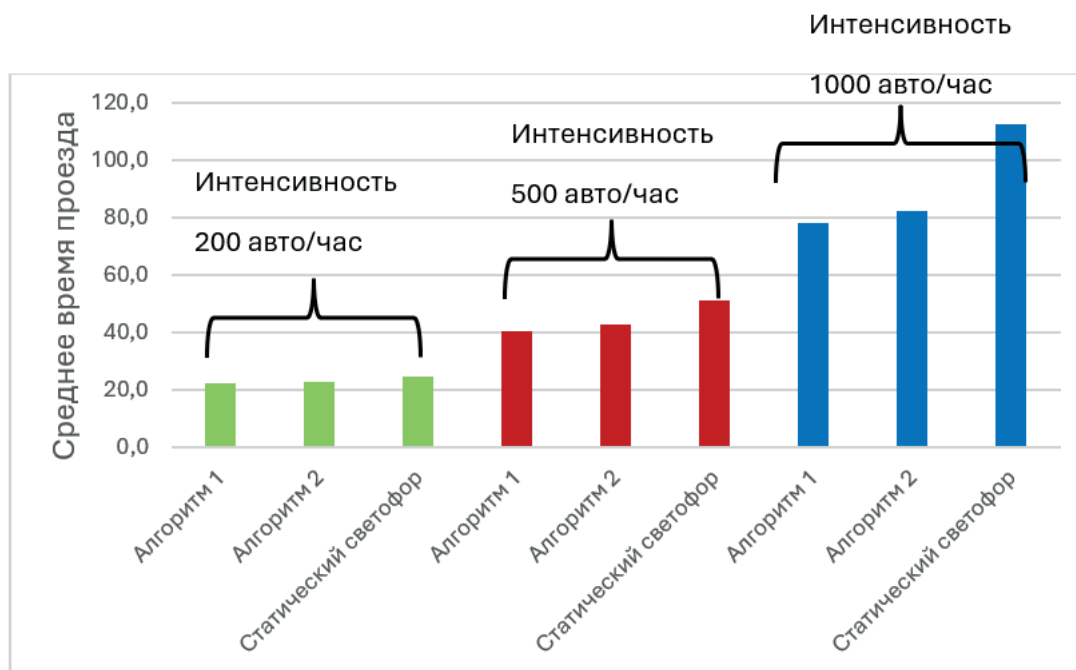


Рисунок 6 — Среднее время проезда в зависимости от интенсивности движения для трёх режимов управления.

Алгоритм 1 демонстрирует стабильное преимущество по всем показателям. Хотя оба адаптивных алгоритма эффективны, Алгоритм 1 (на основе непрерывной зависимости от плотности) показал более высокую эффективность, чем Алгоритм 2 (дискретного перераспределения). Его превосходство составляет в среднем 2–5 секунд по среднему времени и 4–7 секунд по 80-му перцентилю. Это указывает на то, что плавная, аналитически обоснованная корректировка длительности фаз обеспечивает более точное и быстрое реагирование на изменение потока, минимизируя как средние задержки, так и их разброс.

Адаптивное управление повышает предсказуемость и качество обслуживания. Ключевым результатом является улучшение показателя гарантированного времени проезда. Например, при высокой интенсивности Алгоритм 1 гарантирует, что 80% автомобилей проедут перекресток быстрее чем за 95 секунд, в то время как при статическом управлении

этот показатель составляет 145 секунд. Это означает, что адаптивные системы не только уменьшают средние задержки, но и снижают неопределённость для водителей, делая движение более стабильным и прогнозируемым, что является важным фактором комфорта и безопасности.

Эффективность адаптации возрастает с увеличением нагрузки. Сравнение результатов для трёх уровней интенсивности показывает, что относительное преимущество адаптивных алгоритмов возрастает по мере увеличения нагрузки на перекресток. При низкой интенсивности (200 авто/час) разница со статическим режимом невелика (около 10%), что объясняется отсутствием выраженных заторов. Однако при переходе к средним и высоким нагрузкам способность адаптивных систем динамически перераспределять зелёное время позволяет предотвращать лавинообразный рост очередей и эффективнее справляться с пиковыми нагрузками.

## ЗАКЛЮЧЕНИЕ

В данной работе проведено исследование, посвященное разработке и сравнительному анализу двух алгоритмов адаптивного свето-

форного регулирования для изолированного перекрестка с использованием среды имитационного моделирования AnyLogic. Целью

работы было создание и верификация алгоритмов, способных динамически корректировать длительность светофорных фаз на основе текущей загрузки подходов, и оценка их эффективности относительно классического статического управления.

Основные научные и практические результаты работы заключаются в следующем:

- Теоретически обоснована и практически подтверждена значительная эффективность адаптивного управления по сравнению со статическим режимом с фиксированным временем фаз. Наиболее выраженный выигрыш (до 31% по среднему времени проезда) наблюдается в условиях высокой и неравномерной нагрузки, что подтверждает актуальность внедрения подобных систем на загруженных городских перекрестках.

- Установлено, что Алгоритм 1 демонстрирует статистически значимое превосходство над Алгоритмом 2 по всем исследуемым метрикам. Это позволяет сделать вывод о большей эффективности стратегии плавного, аналитически рассчитываемого изменения параметров цикла по сравнению с дискретным пошаговым перераспределением, особенно для сглаживания пиковых нагрузок и уменьшения вариабельности времени задержки.

- Практическая ценность работы заключается в предоставлении готовых, апробированных в виртуальной среде алгоритмических решений и методологии их оценки. Разработанная модель и методика сравнения могут быть использованы городскими службами и проектными организациями для предварительного обоснования внедрения адаптивных светофорных объектов, а также для параметрической настройки таких систем под конкретные условия движения.

- Полученные результаты указывают на важность выбора не только факта адаптивности, но и конкретной математической стратегии управления. Эффективность системы определяется не просто наличием обратной связи, а адекватностью и тонкостью ее алгоритмической реализации.

Таким образом, работа вносит вклад в решение актуальной задачи интеллектуализации городской транспортной инфраструктуры. Предложенные и валидированные алгоритмы представляют собой готовый инструментарий для повышения эффективности работы ключевых элементов улично-дорожной сети — регулируемых перекрестков.



Список источников

1. Чжан, Л. Адаптивное управление светофорами на основе оценки длины очереди в реальном времени и методов обучения с подкреплением // Транзакции IEEE по интеллектуальным транспортным системам. — 2024. — Т. 25, № 8. — С. 8765–8777. — DOI: 10.1109/TITS.2024.3361245.
2. Петров, И. В., Смирнов, А. Н. Цифровые двойники городских перекрёстков на основе гибридного моделирования: кейс Москвы // Transportation Research Procedia. — 2024. — Т. 78. — С. 211–218. — DOI: 10.1016/j.trpro.2024.03.027.
3. Лю, С., Ван, Ц., Чжао, Ю. Дискретно-событийное моделирование для адаптивного регулирования светофоров в условиях высокой плотности городского движения // Журнал передовых транспортных технологий. — 2024. — Т. 2024. — Ст. № 9876543. — 12 с. — DOI: 10.1155/2024/9876543.
4. Гаврилов, Д. А., Соколов, М. В. Применение гибридного моделирования для оценки эффективности адаптивных систем регулирования на перекрёстках // Известия Южного федерального университета. Технические науки. — 2025. — Т. 26, № 2. — С. 112–124. — DOI: 10.33902/26301825-2025-2-112.
5. ГОСТ Р 57910–2024. Интеллектуальные транспортные системы. Термины и определения. — Введ. 2024–07–01. — М.: Стандартиформ, 2024. — 24 с.
6. Чэнь, Ю., Ван, Х. Гибридное моделирование на основе дискретно-событийного и агентного подходов для оптимизации городского движения // Практика и теория имитационного моделирования. — 2025. — Т. 142. — Ст. 103287. — DOI: 10.1016/j.simpat.2025.103287.

## SIMULATION OF ADAPTIVE TRAFFIC LIGHT REGULATION TO OPTIMIZE TRAFFIC FLOWS AT AN INTERSECTION

Kuzmin D.V.<sup>1</sup>, Baginova V.V.<sup>1</sup>, Korobovsky K.M.<sup>1</sup>, Toritsyn A.M.<sup>1</sup>

<sup>1</sup> Russian University of Transport

**Abstract:** the article discusses the use of the AnyLogic simulation environment to study and optimize the operation of an adjustable intersection. The main purpose of the research is to develop and compare two alternative algorithms for adaptive traffic light control. The proposed algorithms analyze in real time the totality of traffic flow indicators (intensity, queue length, waiting time) and on this basis dynamically adjust the duration of the phases of the traffic light cycle. This makes it possible to conduct virtual tests of algorithms under controlled conditions, objectively evaluate their effectiveness based on key metrics (throughput, average transport delay time) and identify the most stable and productive management method for various loading scenarios. The article is of interest to specialists in the field of intelligent transport systems (ITS), traffic management engineers, as well as to researchers and students involved in modeling and optimizing urban mobility. The research results can be used for practical justification of the implementation of adaptive systems at problematic intersections, further development and calibration of more complex control algorithms, creation of digital counterparts of transport hubs for planning and decision-making.

**Keywords:** discrete-event modeling, automobile intersection, automobile traffic, transport system, optimization of the intersection.

© Kuzmin D.V., Baginova V.V., Korobovsky K.M., Toritsyn A.M.

Received 12.11.2025, approved 12.12.2025, accepted for publication 12.12.2025.

For citation:

Kuzmin D.V., Baginova V.V., Korobovsky K.M., Toritsyn A.M. Simulation of adaptive traffic light regulation to optimize traffic flows at an intersection. *Logistics and Supply Chain Management*. 2025. Vol 22, Iss 4 (117). pp. 51-67.

Information about the authors:

Kuzmin D.V., — Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Logistics and Management of Transport Systems, Russian University of Transport (MIIT), e-mail: kuzmindv@rut-miit.ru.

Baginova V.V., — Doctor of Technical Sciences, Professor, Professor of the Department of Logistics and Management of Transport Systems, Russian University of Transport (MIIT).

Korobovsky, K.M., — Master's student at the Department of Logistics and Management of Transport Systems, Russian University of Transport (MIIT).

Toritsin A.M., — Master's student at the Department of Logistics and Management of Transport Systems, Russian University of Transport (MIIT).

---

## References

1. Zhang, L. Adaptive traffic light management based on real-time queue length estimation and reinforcement learning methods // *IEEE Transactions on Intelligent Transport Systems*. — 2024. — Vol. 25, No. 8. — pp. 8765-8777. — DOI: 10.1109/TITS.2024.3361245.
2. Petrov, I. V., Smirnov, A. N. Digital twins of urban intersections based on hybrid modeling: the case of Moscow // *Transportation Research Procedia*. — 2024. — Vol. 78. — pp. 211-218. — DOI: 10.1016/j.trpro.2024.03.027.
3. Liu, S., Wang, C., Zhao, Y. Discrete event modeling for adaptive control of traffic lights in conditions of high urban traffic density // *Journal of Advanced Transport Technologies*. — 2024. — T. 2024. — Article No. 9876543. — 12 p. — DOI: 10.1155/2024/9876543
4. Gavrilov, D. A., Sokolov, M. V. Application of hybrid modeling to evaluate the effectiveness of adaptive control systems at intersections // *Izvestiya Yuzhnogo federalnogo universiteta. Technical sciences*. — 2025. — Vol. 26, No. 2. — pp. 112-124. — DOI: 10.33902/26301825-2025-2-112.
5. GOST R 57910-2024. Intelligent transportation systems. Terms and definitions. — Introduction. 2024-07-01. Moscow: Standartinform, 2024. 24 p.
6. Chen, Yu., Wang, H. Hybrid modeling based on discrete-event and agent-based approaches for optimizing urban traffic // *The practice and theory of simulation modeling*. — 2025. — Vol. 142. — Art. 103287. — DOI: 10.1016/j.simpat.2025.103287.