

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук

Кафедра программирования и информационных технологий

*Адаптивное светофорное регулирование на перекрестке с помощью
программы моделирования транспортных потоков*

ВКР Магистерская диссертация

09.04.02 Информационные системы и технологии

Информатика как вторая компетенция

Допущено к защите в ГЭК

Зав. кафедрой _____ Н.А. Тюкачев, к.ф.-м.н., доцент _____.2016

Обучающийся _____ В.А. Нечипоренко, 2 курс, д/о

Руководитель _____ Н.А. Тюкачев, к.ф.-м.н., доцент

Воронеж 2016

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных наук

Кафедра программирования и информационных технологий

УТВЕРЖДАЮ

заведующий кафедрой

_____ Н.А. Тюкачев

подпись, расшифровка подписи

___.___.2016

ЗАДАНИЕ
НА ВЫПОЛНЕНИЕ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ
ОБУЧАЮЩЕЙСЯ Нечипоренко Виталии Андреевны

1. Тема работы «Адаптивное светофорное регулирование на перекрестке с помощью программы моделирования транспортных потоков», утверждена решением ученого совета факультета компьютерных наук от __.__.2016
2. Направление подготовки 09.04.02 Информационные системы и технологии
3. Срок сдачи студентом законченной работы __.__.2016
4. Календарный план:

№	Структура ВКР	Сроки выполнения	Примечание
	Введение	04.09.15-09.09.15	
1	Постановка задачи	10.09.15-15.09.15	
2.1	Анализ цели и объектов моделирования	16.09.15-31.10.15	
2.2	Классификация моделей по уровню детализации	01.11.15-14.11.15	
2.3	Микроскопические модели	15.11.15-23.11.15	
2.4	Статистические распределения в теории транспортных потоков	24.11.15-30.11.15	
2.5	Подходы к управлению транспортными потоками на регулируемых перекрестках	01.12.15-07.12.15	
2.6	Адаптивное светофорное регулирование	08.12.15-15.12.15	
2.7	Анализ существующих инструментов оптимизации светофорного регулирования	16.12.15-15.01.16	
2.8	Алгоритм нахождения оптимального цикла светофора	16.01.15-14.02.16	
3	Описание разработанного программного обеспечения	15.02.16-01.03.16	
3.1	Структура программного обеспечения	02.03.16-14.03.16	
3.2	Функционал программного обеспечения	15.03.16-15.04.16	
3.3	Интерфейсная часть	16.04.16-29.04.16	
4	Экспериментальное исследование	01.05.16-24.05.16	
	Заключение	25.05.16-26.05.16	
	Список использованных источников	27.05.16-28.05.16	
	Приложения	29.05.16-30.05.16	

Обучающаяся

Руководитель

Реферат

Текстовый документ 58 с., 28 рисунков, 12 формул, 35 источников, 2 приложения.

АДАПТИВНОЕ СВЕТОФОРНОЕ РЕГУЛИРОВАНИЕ, ДОРОЖНЫЙ ЗАТОР, ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ, МИКРОМОДЕЛИРОВАНИЕ, СВЕТОФОРНЫЙ ОБЪЕКТ, ПРОГРАММА МОДЕЛИРОВАНИЯ, ТРАНСПОРТНЫЙ ПОТОК, УПРАВЛЕНИЕ ТРАНСПОРТНЫМИ ПОТОКАМИ.

Объектами исследования являются транспортный поток перед перекрестком и адаптивное светофорное регулирование.

Цель настоящей ВКР – разработка и исследование имитационной модели автотранспортных потоков для создания методики оптимизации светофорного регулирования на перекрестке, а также оценка эффективности полученной методики на заданной модели перекрестка.

В ходе работы представлены методы проектирования и оценки эффективности режимов регулирования на пересечениях. Особое внимание уделено параметрам транспортного потока, которые необходимы при проектировании режимов регулирования. Рассмотрены методики оценки эффективности регулируемых пересечений, а также программные продукты, применяемые при проектировании режимов регулирования.

Методы и средства проведения работы. При решении поставленных задач использовались методы математического и компьютерного моделирования, вычислительного эксперимента, математической статистики, теория графов, теория транспортных потоков, теория массового обслуживания.

Основные разделы – постановка задачи, анализ задачи, обзор существующих решений, теоретическая часть, используемый математический аппарат, реализация, апробация, тестирование, заключение.

Содержание

Введение	5
1 Постановка задачи	7
2 Анализ задачи	9
2.1 Анализ цели и объектов моделирования	9
2.2 Классификация моделей по уровню детализации	13
2.3 Микроскопические модели	15
2.4 Статистические распределения в теории транспортных потоков	20
2.5 Алгоритмы работы светофора. Дорожные контроллеры	21
2.6 Адаптивное светофорное регулирование	22
2.7 Существующие инструменты оптимизации светофорного регулирования	26
2.8 Алгоритм нахождения оптимального цикла светофора	29
3 Описание разработанного программного обеспечения	33
3.1 Функционал программного обеспечения	33
3.2 Структура программного обеспечения	33
3.3 Интерфейсная часть	44
3.4 Тестирование и апробация полученных результатов	45
Заключение	48
Список использованных источников	49
Приложение А. Исходный код алгоритма расчета параметров IDM	54
Приложение Б. Исходный код алгоритма поиска кратчайшего пути до узлового соединения	55

Введение

В настоящее время проблема перегруженности автомобильных дорог для большинства крупных городов является одной из центральных, требующих первоочередное решение. Проблема минимизации автомобильных пробок и заторов в российских мегаполисах чрезвычайно актуальна. Острота транспортной проблемы требует системного подхода к ее решению.

Мероприятия могут иметь как административный характер, так и заключаться в строительстве новых и расширении действующих дорожных развязок. Можно улучшить ситуацию на узловых участках дорожной сети за счет оптимизации светофорного регулирования. Это направление экономически выгодно, на его реализацию затрачивается немного времени. Компьютерное моделирование дорожного движения позволяет быстро и дешево узнать, например, к какому результату приведет изменение режима работы светофора без необходимости проводить данный эксперимент на реальном светофоре.

Имитационное моделирование – это метод исследования, при котором изучаемая система заменяется моделью, с достаточной точностью описывающей реальную систему, и с ней проводятся эксперименты с целью получения информации об изучаемой системе.

К имитационному моделированию прибегают, когда:

- дорого или невозможно экспериментировать на реальном объекте;
- невозможно построить аналитическую модель (в системе есть причинные связи, случайные переменные);
- необходимо симитировать поведение системы во времени.

Традиционно модели дорожного движения классифицируются по уровню детализации на макроскопические, микроскопические и мезоскопические. Отнесение модели к одному из классов определяет возможные области ее применения.

В микроскопических моделях транспортные потоки образуются в

результате взаимодействия отдельно моделируемых участников дорожного движения. Эти модели обеспечивают детальную имитацию передвижений и поведения участников движения.

Внедрение технических средств адаптивного координированного управления транспортными потоками позволяет до 20 % увеличить скорость сообщения по сравнению с жестким светофорным регулированием.

В России ведутся научно-исследовательские работы, направленные на изучение транспортных потоков и на разработку программного обеспечения. Применяются новые подходы в области транспортных потоков, например, используется теория систем массового обслуживания для создания математической модели регулируемого перекрестка [1] [2].

Транспортная лаборатория иркутского государственного университета (ИрГТУ) является разработчиком программы оптимизации светофорного регулирования «Светофор» [3]. Специалистами Института Системного Анализа РАН была разработана программа TransNet – программа моделирования транспортных потоков в сети крупного города [4]. Специалистами компании Solaris Development и AGA Group разработан программный комплекс «Артерия 3.0» [5], пакет имитационного моделирования AIMSUN [6], оптимизационный пакет регулирования дорожного движения Transyt-7FR [7].

В зарубежных странах достигнуты значительные успехи в разработке и применении современных программных комплексов, помогающих эффективно управлять транспортной сетью города. Примерами таких программных продуктов являются PTV VISSIM, PTV VISUM [8] [9].

Стоит отметить, что существуют программы, которые моделируют движение транспортных средств на отдельном участке улично-дорожной сети [10][11][12].

1 Постановка задачи

Необходимо разработать имитационную модель, позволяющую моделировать транспортные потоки. Данная система должна отвечать следующим требованиям:

- моделирование транспортных потоков на регулируемом перекрестке;
- определение наилучшего режима работы светофора;
- сохранение результатов моделирования;
- вывод результатов статистического эксперимента;
- обработка результатов статистического эксперимента.

К разрабатываемой модели предъявляются следующие требования и ограничения:

- микроскопический имитационный подход к моделированию;
- безаварийное движение транспортных средств (далее ТС);
- при проезде перекрестка ТС могут изменить направление движения прямо, налево, направо. Если из текущей полосы совершение маневра невозможно, происходит попытка смены полосы движения;
- наличие светофоров без дополнительных секций на каждом направлении перекрестка;
- задание произвольного режима работы светофора;
- получение разнообразной информации о ТС (количество автомобилей, прибывающих на перекресток на различные фазы светофора, длина очереди перед светофорами);
- возможность конфигурации проезжей части (установка количества полос движения) и регулируемого пересечения;
- учет зависимости интенсивности дорожного движения от времени и пропускной способности;

- учет различных транспортных средств (габариты, предпочтительная скорость передвижения);
- графическое отображение результатов работы модели в режиме реального времени.
- возможность использования модели для интеллектуального управления дорожным движением (ДД) подразумевает наличие адаптивного режима работы светофора.

2 Анализ задачи

Для моделирования улично-дорожной сети необходимо воспользоваться методами имитационного моделирования, которое в отличие от аналитического моделирования позволяет описывать системы произвольной сложности. Имитационная модель реализует логику функционирования системы и взаимодействие элементов системы во времени, учитывая наиболее существенные причинно-следственные связи, присущие рассматриваемой системе. Для реализации имитационной модели необходимо первоначально [13]:

- определить объект и цели моделирования;
- составить концептуальную модель;
- определить общую математическую модель;
- разработать общий моделирующий алгоритм.

2.1 Анализ цели и объектов моделирования

Для анализа информации об объекте управления необходимо заложить в систему некое представление об этом объекте, которое и называется моделью. Объектом управления для интеллектуальной транспортной системы являются транспортные потоки. Источником информации об объекте управления являются датчики и детекторы на дороге, смежные информационные системы и ввод данных оператором.

Целью создания модели является оценка эффективности работы дорожной сети в зависимости от конфигурации сети и способа светофорного регулирования. Оценка эффективности производится на основе информации о передвижении транспортных средств в рамках модели.

Необходимо выделить основные составляющие модели и их параметры, а также определить основные характеристики транспортных средств, на основе которых будет проводиться анализ эффективности работы моделируемой дорожной сети.

Взаимодействие модели с внешней средой заключается в возможности извне динамически управлять режимом работы светофоров. Соответственно, необходимо описать способ такого управления светофорами.

В данной работе объектом моделирования является регулируемый перекресток (рисунок 1). Несмотря на то, разработано большое множество компьютерных программ для анализа и моделирования регулируемых пересечений, предоставляющих информацию о задержках на перекрестке, длинах очередей на подходах к нему, потоке насыщения, нет ни одной программы, позволяющей предоставить сразу все эту информацию как для существующих, так и для проектируемых условий движения. Так, например, если два пересечения расположены слишком близко друг к другу, одна компьютерная программа может быть полезной для определения задержек на пересечении, но не является эффективным средством для определения других параметров транспортного потока таких, как длина очереди.

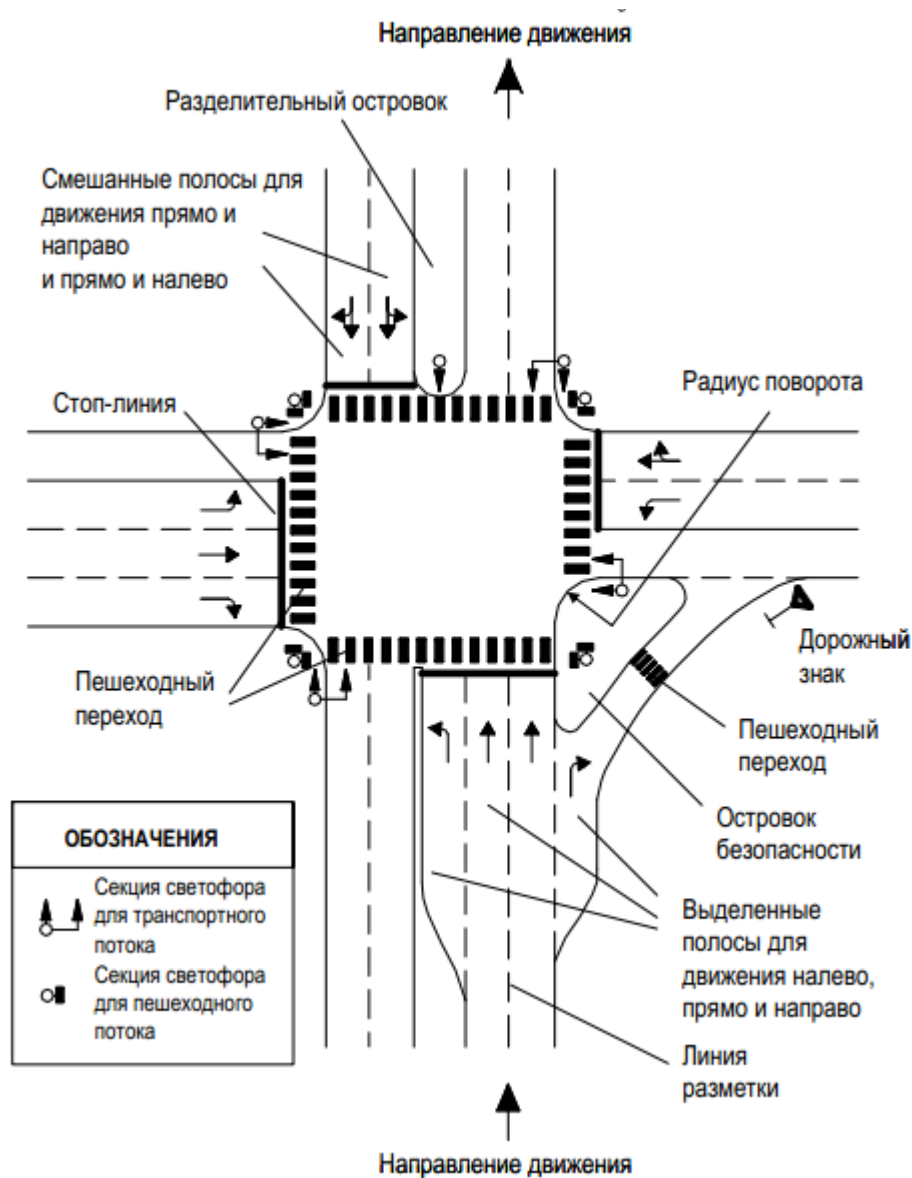


Рисунок 1 – Геометрические элементы регулируемого перекрестка и элементы управления транспортными и пешеходными потоками

Систему перекрестков со светофорами можно представить в виде графа, вершинами которого являются перекрестки, а ребрами – улицы. Например, как это изображено на рисунке 2.

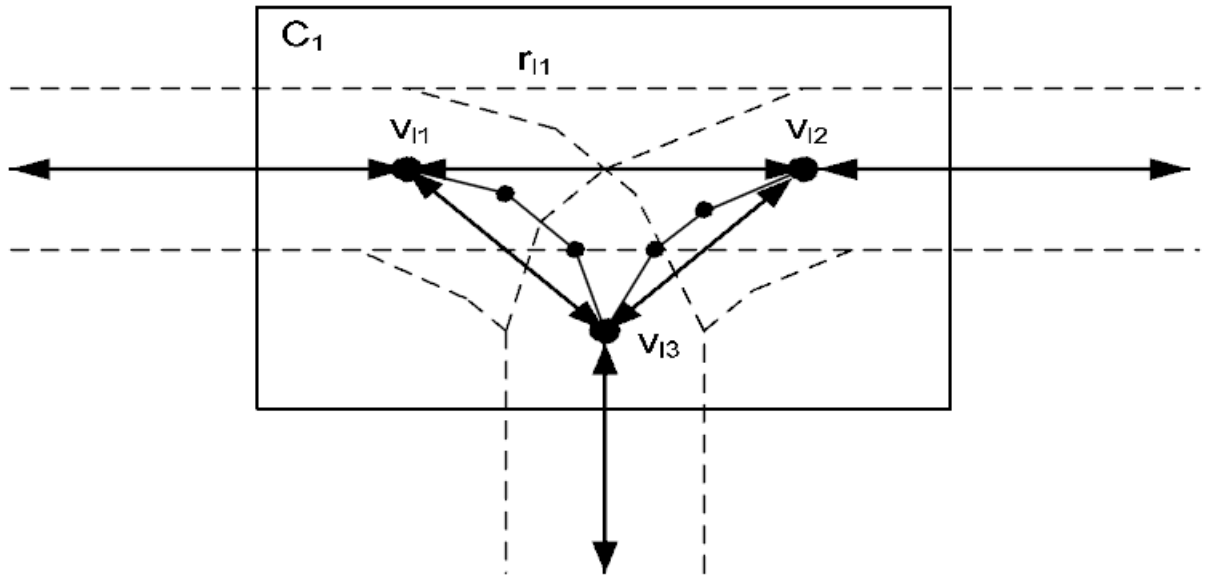


Рисунок 2 – Представление Т-образного перекрестка в виде двухслойного ориентированного графа

Граф состоит из двух слоев: топологического – для представления пространственных характеристик дорожной сети и логического – для представления семантических свойств.

Первый слой задается в виде ориентированного графа:

$$G_t = \langle V_t, R_t \rangle, \quad (1)$$

где V_t – множество топологических вершин, соответствующих пространственным точкам виртуального мира; R_t – множество топологических ребер, определяющих наличие проезжей части между топологическими узлами.

Второй слой задается в виде ориентированного графа:

$$G_l = \langle V_l, R_l \rangle, \quad (2)$$

где $V_l \subseteq V_t$ – множество логических вершин, базирующихся на топологических вершинах; R_l – множество логических ребер, определяющих наличие логического участка дороги между логическими узлами.

На основе последнего могут быть введены модели представления перекрестков:

$$C_i = \langle V_{li}, R_{li} \rangle, \quad (3)$$

где $V_{li} \subseteq V_l$ – множество логических вершин, образующих i -ый перекресток;

$R_{li} \subseteq R_l$ – множество логических ребер, соединяющих узлы перекрестка.

2.2 Классификация моделей по уровню детализации

Созданные модели исторически разделились на три группы: макро–, мезо– и микромодели, различающиеся степенью детализации моделируемого объекта.

Как правило, макроскопические модели имеют невысокие требования к компьютеру. Среди программных продуктов, реализующих макроскопический подход, можно отметить следующие: PTV VISSUM (интерфейс программы приведен на рисунке 3), SimTraffic, PARAMICS. Главным преимуществом макромоделей по настоящее время остается скорость расчета показателей управления. Главным недостатком данных моделей является слабая точность в предзаторных и заторных условиях движения.

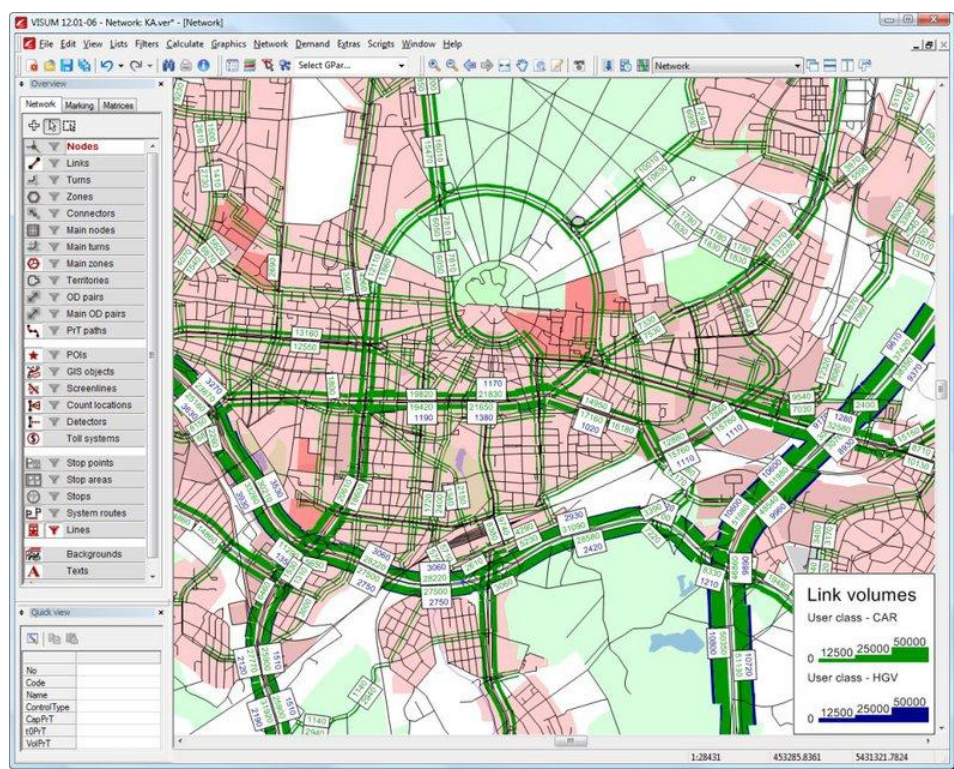


Рисунок 3 – Интерфейс программы для макромоделирования PTV VISSUM

В противовес макромоделям были разработаны микромодели. В данном случае детально имитируется движением транспортных средств с целью установления показателей эффективности функционирования локального участка улично-дорожной сети. Микромодели оперируют

конкретными объектами из реального мира такими, как регулируемый перекресток, транспортная развязка, сеть улиц, автомобиль. Характер поведения автомобиля описывается с помощью правил, которые определяют, когда автомобиль ускоряется, замедляет скорость, перестраивается в другой ряд, а также когда и как автомобиль выбирает и меняет свой маршрут следования.

Если провести обобщение микроскопических моделей, то основными исходными данными для их функционирования являются количество полос для движения, ширина полос для движения, структура транспортного потока, интенсивность транспортных потоков, допустимая скорость движения, приоритетные направления движения, параметры светофорного регулирования. В результате работы микроскопических моделей, как правило, получают следующие выходные данные: длина очереди, задержки транспортных средств, средняя скорость, максимальная или минимальная скорость, время движения, анимационное изображение локального участка сети.

Главным достоинством данных моделей является существенно большая точность и гибкость, позволяющая производить моделирование транспортных потоков с наиболее сложными топологиями транспортной сети. Главным недостатком является высокая вычислительная сложность, не позволяющая использовать данный класс моделей при адаптивном управлении в реальном времени.

К программным пакетам, позволяющим строить микроскопические симуляционные модели, относят: MITSIM (Microscopic Traffic Simulator), PTV VISSIM, AIMSUN (Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks). Интерфейс пользователя последнего программного продукта приведен на рисунке 4.

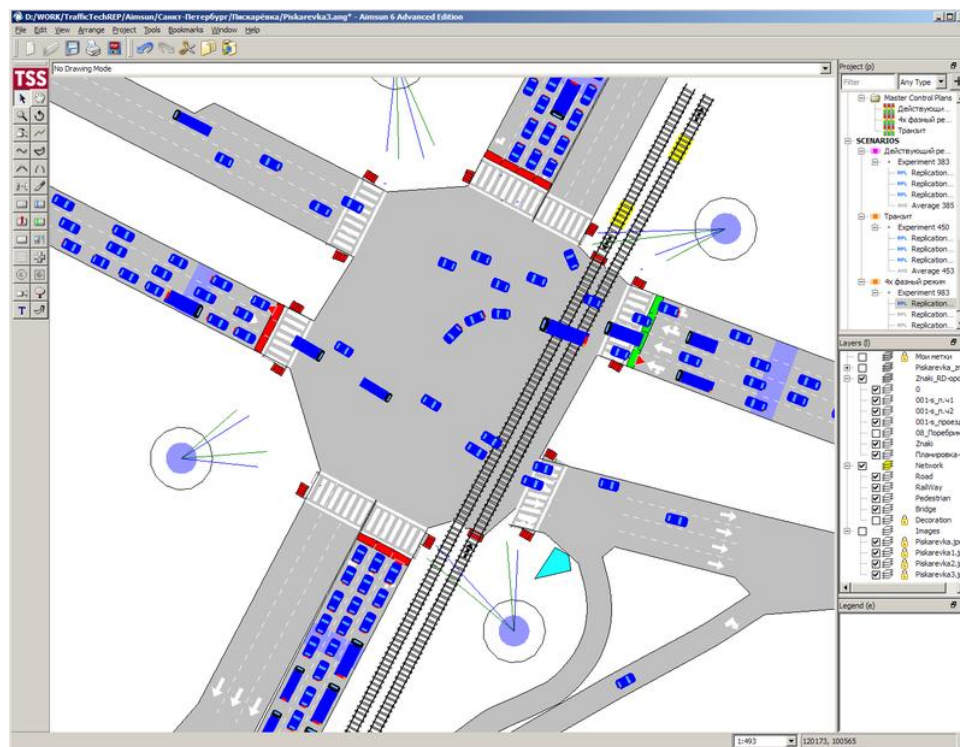


Рисунок 4 – Интерфейс программы для микромоделирования AIMSUN

Наиболее перспективным компромиссом являются мезомодели динамики [14], так как они описывают автомобили на высоком уровне детализации (элемент микромоделирования), а их поведение и взаимодействие – на низком уровне (элемент макромоделирования). В этом случае также как и при макроскопическом моделировании, могут быть получены характеристики транспортного потока (скорость, плотность, задержки на перекрестке). Мезоскопическое моделирование применяется там, где желательно использовать микроскопические модели, но невозможно из-за большого размера транспортной сети или ограниченности ресурсов, которые требуется затратить на создание и отладку сети.

Инструментами, реализующими мезоскопическое имитационное моделирование, являются: CONTRAM (Continuous Traffic Assignment Model), DYNAMIT-P, DYNASMART, FASTLANE, DTASQ, MesoTS [15].

2.3 Микроскопические модели

Исходя из того, что создаваемая модель будет использоваться в первую очередь для оптимизации работы светофоров, является целесообразным использование в ней микроскопического подхода, в рамках которого можно

выделить несколько основных направлений.

Известными микроскопическими моделями являются: модель следования за лидером, модель смены ряда, модель клеточных автоматов Нагеля [16] и модель «умного водителя» Мартина Трайбера, Ансгара Хенека и Дирка Хельбинга [17].

Модель следования за лидером описывает характер ускорения и торможения транспортного средства при взаимодействии его с другими участниками дорожного движения. Модель смены ряда учитывает условия принятия решения для перестроения автомобиля в другой ряд. Одна из моделей смены ряда называется MOBIL (Minimizing Overall Braking decelerations Induced by Lane changes). Модель выбора маршрута описывает механизм определения оптимального маршрута исходя из места пребывания и места назначения, а также правила реагирования на изменяющуюся ситуацию на дороге во время следования пути. Программа микромоделирования AIMSUN имитирует поведение каждого автомобиля в потоке по всем вышеперечисленным моделям поведения.

Модель оптимальной скорости была предложена М. Бандо и характеризуется следующими особенностями: автомобиль будет придерживаться максимальной скорости с достаточным расстоянием до следующего автомобиля, а также двигаться с оптимальной скоростью, опираясь на расстояние до следующего автомобиля. Движение автомобиля задается через уравнение ускорения (4):

$$\dot{v} = \frac{dx}{dt^2} = \alpha [V_{optimal}(\Delta x) - v], \quad (4)$$

где α – скорость реакции водителя (обычно α равно 1); Δx – дистанция до впереди едущего автомобиля, $\Delta x = x_{i+1} - x_i$, v – текущая скорость автомобиля.

Функция $V_{optimal}$ обозначает оптимальную скорость:

$$\Delta x = th(\Delta x - 2) + th(2), \quad (5)$$

где th – гиперболический тангенс.

Модель умного водителя (The Intelligent Driver Model или сокращенно IDM) относится к классу моделей следования за лидером и описывается через обыкновенное дифференциальное, где расстояние и время являются непрерывными переменными. Кроме фактической скорости автомобиля $v(t)$ и дистанции до ведущего автомобиля $s(t)$ в модели также учитывается разность скоростей $\Delta v(t)$, которая играет стабилизирующую роль и является неотъемлемой частью реального потока автомобилей, особенно когда поток становится плотным и возникает вероятность столкновения [17].

Параметры модели «умный водитель» приведены на рисунке 5 (автомобиль $\alpha - 1$ является лидером для автомобиля α).

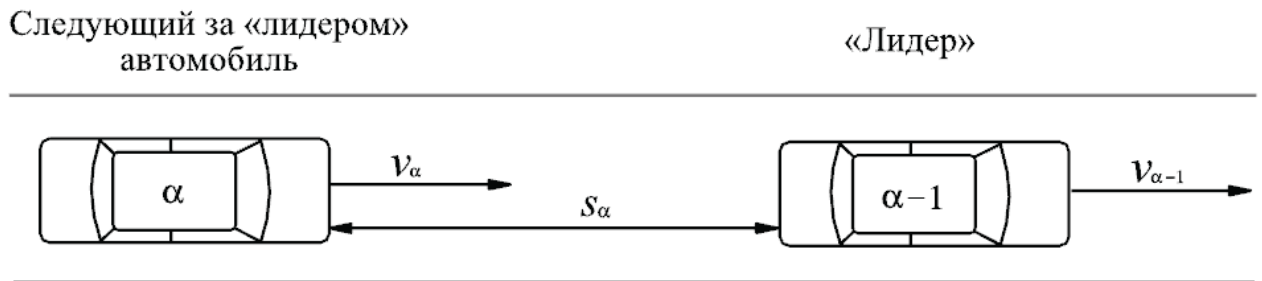


Рисунок 5 – Параметры модели «умный водитель»

В этой модели ускорение задается формулой (6):

$$\dot{v} = \frac{dv_a}{dt} = f(s_a, v_a, \Delta v_a) = a \left[1 - \left(\frac{v_a}{v_0} \right)^\delta - \left(\frac{s'(v_a, \Delta v_a)}{s_a} \right)^2 \right], \quad (6)$$

где a – максимальное ускорение; v_a – текущая скорость; v_0 – желаемая скорость (скорость, с которой автомобиль перемещался бы в свободном потоке); s_a – фактическая дистанция между автомобилями в данный момент времени; s' – эффективная минимальная дистанция; δ – экспонента ускорения.

Уравнение ускорения автомобиля задается в виде (7):

$$\dot{v}_{speed}(v) = a \left[1 - \left(\frac{v_a}{v_0} \right)^\delta \right]. \quad (7)$$

Функция ускорения позволяет плавно увеличивать скорость автомобиля v_a от нуля до желаемого значения v_0 . Значение экспоненты ускорения δ определяет, как уменьшается ускорение автомобиля по мере

достижения желаемой скорости передвижения. При $\delta \rightarrow \infty$ транспортное средство будет набирать желаемую скорость с постоянным ускорением. При $\delta=1$ автомобиль плавно достигает желаемой скорости. Однако в реальной жизни параметр δ оказывается между двух случаев ($\delta=1$ и $\delta \rightarrow \infty$), и на практике часто принимают $\delta=4$.

Функция торможения необходима для того, чтобы предотвратить столкновение транспортных средств.

Уравнение торможения автомобиля задается в виде (8):

$$\dot{v}_{break}(s_{\alpha}, v_{\alpha}) = \begin{cases} -a \left(\frac{s_0 + v_{\alpha} T'}{s_{\alpha}} \right)^2, & s_{\alpha} \in [1, 20]; \\ 0, & s_{\alpha} > 20, \end{cases} \quad (8)$$

где s_{α} – дистанция до впереди едущего автомобиля (измеряется в метрах), s_0 – минимальное расстояние между автомобилями, которое сохраняется даже в пробке, T – желаемый временной интервал – время движения автомобиля со скоростью v_{α} до столкновения с впереди едущим автомобилем, v_{α} – текущая скорость.

Таким образом, автомобиль плавно набирает скорость от 0 до желаемой v_0 с вычисляемым ускорением \dot{v}_{speed} , которое зависит от скорости v_{α} в данный момент времени и от заданного максимального значения ускорения a . Снижение скорости в случае возникновения препятствия перед автомобилем осуществляется с помощью функции \dot{v}_{break} , которая зависит от расстояния до препятствия s_{α} , скорости автомобиля v_{α} , от параметров T и s_0 .

Исходя из (6) и (7) уравнение (5) можно переписать в упрощенном виде, состоящем из двух частей (9):

$$\dot{v} = \dot{v}_{speed}(v) + \dot{v}_{break}(s, v, \Delta v). \quad (9)$$

Стохастичность моделирования достигается за счет того, что параметры, характеризующие индивидуальные особенности стиля вождения водителя, вычисляются отдельно для каждого автомобиля случайным образом в соответствии с равномерным распределением с разбросом, равным

20%. В качестве исходных значений временная задержка T принимает различные значения. Минимально допустимое значение параметра – 0.5 секунды. Рекомендованное значение для T – 2 секунды (1 секунда на реакцию водителя и 1 секунда на торможение), для s_0 – 1.5 метра, для a – 2 м/с².

Каждый автомобиль при необходимости может менять полосу движения, при этом проверяется возможность безопасного перестроения. Как только расстояние до впереди едущего автомобиля становится меньше допустимого. Предпринимается попытка сменить ряд. Сначала проверяется, можно ли перестроиться в левый ряд, если нельзя (маневр приведет к резкому торможению других транспортных средств или аварии), тогда проверяется возможность перестроиться в правый ряд. Параметр s_β задает дистанцию безопасности, которая проверяется перед перестроением автомобиля, также s_β отвечает за целесообразность перестроения в новый ряд, так как проверяется наличие в новом ряду места для движения вперед.

Эффективным и информативным способом моделирования движения совокупности транспортных средств по магистрали являются клеточные автоматы. Модели, основанные на клеточных автоматах, являются дискретными: дорога разбивается на секции определенной длины, а время – на шаги. Каждая секция дороги может быть либо занята одним транспортным средством, либо быть пустой.

Концепция клеточных автоматов была введена Дж. Фон Нейманом в 50-е годы XX века. Применять клеточные автоматы для моделирования транспортных потоков предлагалось в работе [11]. Пример автомата представлен массивом ячеек длины N и ширины M (число полос) на рисунке 6. Длина ячейки l_i характеризует разрешающую способность автомата.

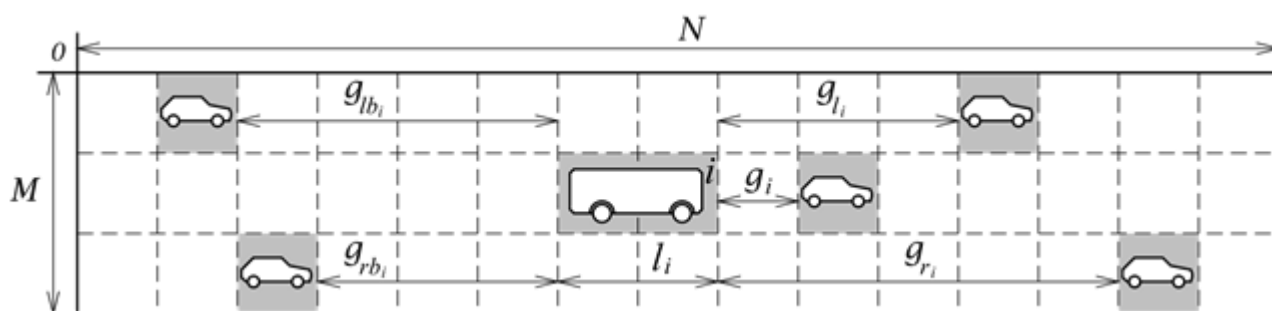


Рисунок 6 – Многополосный транспортный клеточный автомат 3x15

На массиве в произвольном порядке располагаются транспортные средства, позиция i -го автомобиля определяется переменными n_i (ячейка) и m_i (полоса). Каждая машина принадлежит к определённому типу (легковой, грузовой, автобус). Число ячеек, необходимое для представления автомобиля, различается в зависимости от его типа. Максимальная скорость автомобиля также зависит его типа. Текущий момент времени (итерация) обозначается переменной t .

Применение клеточных автоматов в моделировании дорожного движения является молодым и перспективным направлением, характеризуемым высокой вычислительной скоростью и эффективностью. Этот подход может быть классифицирован как имитационный и позволяет представлять динамику системы во времени. Кроме того, разделение машин по типам в этом случае является простой задачей, что отличает его от аналитических моделей.

2.4 Статистические распределения в теории транспортных потоков

Микро- и мезоскопические модели базируются на гипотезе о статистическом распределении прибытий транспортных средств к данному участку УДС. Входной поток автомобилей задается с помощью так называемых генераторов (имитация мест, где появляются транспортные средства). У каждого генератора указываются следующие параметры:

- место создания автомобилей;
- интенсивность (количеством автомобилей, приезжающих к перекрестку за определенный промежуток времени);

- направление движения и максимальная скорость автомобилей.

После чего генератор с заданным интервалом времени создает автомобили в указанном месте.

В результате анализа существующих законов распределения был выбран способ генерации временных интервалов между созданием автомобилей по показательному распределению. Интенсивность входного потока λ .

Функция показательного распределения имеет вид (10):

$$F(x) = \begin{cases} 0, & x < 0; \\ 1 - e^{-\lambda x}, & x \geq 0. \end{cases} \quad (10)$$

Рассмотрим случай $x \geq 0$ и обозначим $y = F(x) = 1 - e^{-\lambda x}$. Из данной формулы найдем x (11):

$$x = -\frac{\ln(1-y)}{\lambda}, \quad (11)$$

где y – случайная величина, равномерно распределенная на отрезке $[0, 1]$; x – случайная величина, имеющая показательное распределение с параметром λ . Значение параметра y генерируется с помощью датчика случайных чисел – функции Random, которая возвращает непрерывную случайную величину, равномерно распределенную на заданном отрезке. В нашем случае $y \in [0; 1]$.

2.5 Алгоритмы работы светофора. Дорожные контроллеры

На сегодняшний день при управлении транспортными потоками применяется множество различных методов. Далее приведены таковые от наиболее старых до наиболее современных:

- Контроллеры жесткого управления с фиксированными длительностями фаз (разрешающих сигналов) по отдельным направлениям перекрестка. Светофорные сигналы переключаются по одной или нескольким заранее заданным временным программам. Такие контроллеры предназначены для управления дорожным движением на перекрестках с мало изменяющейся в течение дня интенсивностью движения.

- Вызывные устройства, которые обеспечивают переключение светофорных сигналов по вызову пешеходами или транспортными средствами, прибывающими с прилегающих к магистрали улиц. Длительности разрешающих сигналов для транспортных средств, как и в предыдущем случае, фиксированы.
- Контроллеры адаптивного управления обеспечивают непостоянную длительность фаз разрешающих сигналов. Они предназначены для управления движением на перекрестках, где интенсивность движения часто меняется в течение суток. Длительность сигналов так же, как и всего цикла регулирования, меняется в заранее заданных пределах от минимального до максимального значения.

Выбор метода управления зависит от степени загруженности транспортной сети. В случае перегруженной транспортной системы эффективными являются только адаптивные методы управления. Таким образом, рассматривая задачу оптимизации управления ТП в мегаполисе актуально рассматривать структурное адаптивное управление.

2.6 Адаптивное светофорное регулирование

Одной из перспективных областей в сфере управления дорожным движением являются адаптивные системы. Адаптивные (интеллектуальные) системы управления дорожным движением – это системы, которые изменяют режим работы светофоров в соответствии с текущей ситуацией на дороге.

Для того чтобы понимать с какими параметрами работают управляющие алгоритмы, необходимо знать базовые определения светофорного регулирования [18]:

Такт регулирования (Interval) – период действия определенной комбинации светофорных сигналов.

Фаза регулирования (Signal Phase) – совокупность основного и следующего за ним промежуточного такта.

Цикл регулирования (Signal Cycle) – периодически повторяющаяся совокупность всех фаз.

Понятия цикла, фазы и интервала, наглядно изображены на рисунке 7.

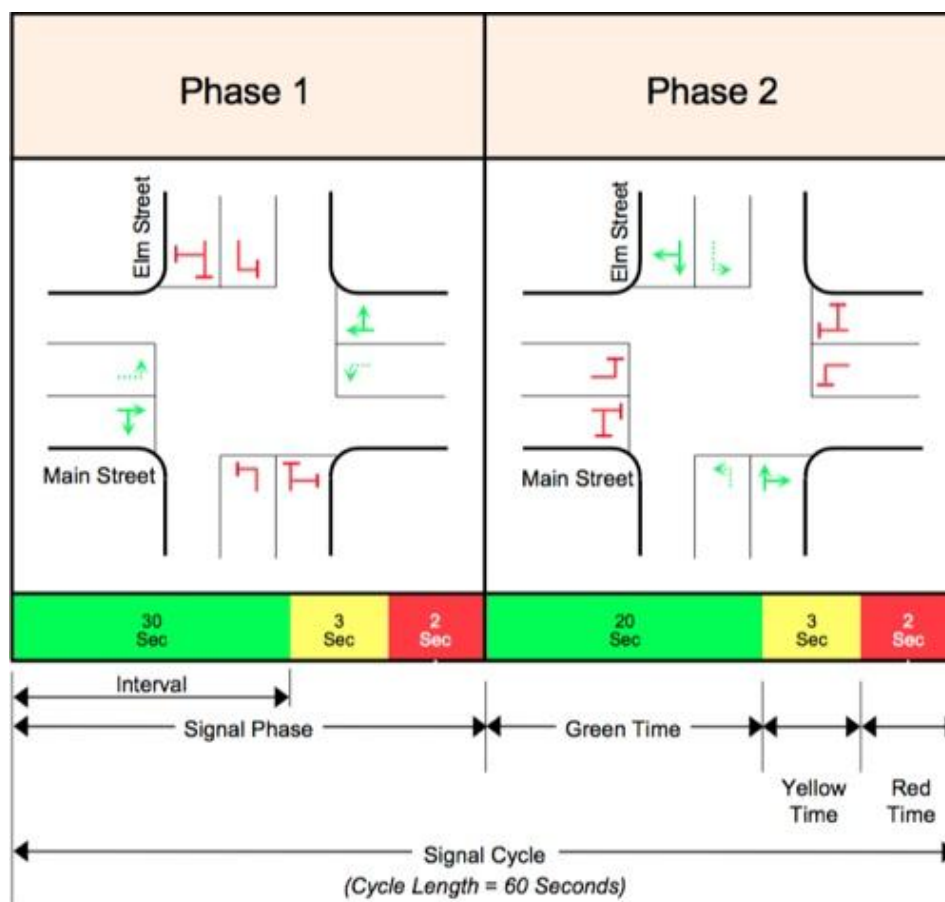


Рисунок 7 – Такты, фазы и циклы работы светофорного объекта

Секция регулирования (Split) – процент цикла регулирования, выделенный каждой из фаз регулирования [19]. Другими словами, изменяя процент времени на фазу, можно управлять длительностью зеленого сигнала на наиболее нагруженном направлении. На отдельно стоящем перекрестке это дает уменьшение задержек.

Смещение (Offset) – разница (в секундах или процентах от цикла регулирования) между часами на конкретном перекрестке и мастер-часами (на сети перекрестков) [19].

Рисунок 8 наглядным образом иллюстрирует понятие смещения.

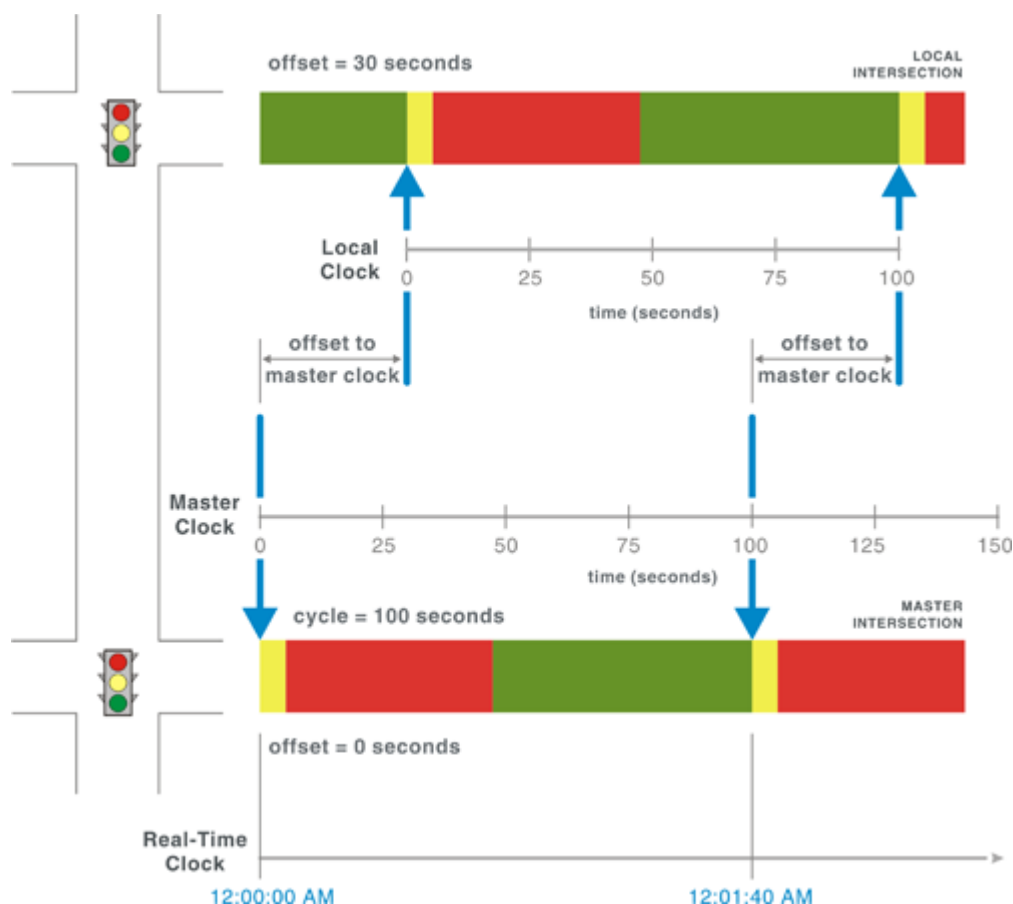


Рисунок 8 – Определение смещения регулирования светофорного объекта

Адаптивное управление как способ автоматизирования движения транспортных потоков на регулируемых перекрестках находит отражение во многих работах. Например, реализация подобного контроллера приведена в работе [20].

Рассмотрим более подробно пример адаптивного алгоритма (далее АС-1), который применяется в единой технологической платформе «Электронный городской» для информационно-транспортной системы на базе RFID-технологий, разработанный научно-техническим холдингом «ВЗГЛЯД».

На перекрестке располагаются два типа детекторов: D1 и D2. Детекторы D1 располагаются на подъезде к перекрестку. Они считывают номера автомобилей в момент проезда через них и вносят номера в память. Детекторы D2 необходимы для учета автомобилей, покидающих перекресток, и располагаются непосредственно на границе перекрестка.

Взаимодействие между детекторами и светофором происходит следующих образом:

- 1 У всех автомобилей системой считывается регистрационный номер;
- 2 Во время работы красной фазы светофора детекторы D1 считывают номера проехавших через них автомобилей и фиксируют их в список;
- 3 В момент, когда загорается зеленый свет, детектор D1 прекращает записывать номера в список, а детектор D2 начинает удалять из списка номера автомобилей, проезжающих через него.
- 4 В момент, когда последний из автомобилей проезжает через детектор D2 и список становится пустым, светофор переключается на красную фазу и переходит в начало цикла. Данное правило повторяется для всех проезжих частей дороги.

Некоторые адаптивные алгоритмы [21-23] находят применение при разработке пакетов имитационного моделирования и не лишены недостатков. Адаптивный алгоритм, работающий по количеству автомобилей, реализован в программном комплексе PTV VISSIM. В этом алгоритме не учитывается, по каким направлениям они прибывают и убывают, в отличие от AC-1, который учитывает номера автомобилей, а значит, стремится пропустить зарегистрированные в списке автомобили.

Наряду с такими подходами, как транспортно-зависимое управление с оптимизацией параметров регулирования в реальном времени, применяется также «мягкое» программирование светофорных объектов с использованием нечеткой логики (НЛ) [24]. Подобные решения нашли применение в работах [25][26]. При использовании аппарата НЛ дискретные уровни интенсивностей движения представляют непрерывной величиной, что в случае дорожного регулирования позволяет гибко приспособить задачу к особенностям трафика конкретного перекрестка путём обучения экспертным способом или на основе обучающей выборки.

2.7 Существующие инструменты оптимизации светофорного регулирования

Для формирования интегральных моделей, позволяющих производить расчёт показателей качества управления, а также оптимизацию параметров светофора, используются специализированные пакеты, такие как: PTC VISSIM, TRANSYT-7F, SUMO (Simulation of Urban Mobility), MITSIMLab (Microscopic Simulation Laboratory), PASSER V, Synchro. Детальный анализ возможностей представленных пакетов можно найти в отчёте проекта SMARTTEST [27]. К сожалению, большинство пакетов труднодоступно для исследования ввиду закрытости исходного кода и платной лицензии. Далее подробно рассматриваются некоторые из них.

Современным инструментом для оценки эффективности внедрения интеллектуальных транспортных систем на дорогах является создание моделей функционирования объектов с использованием современного программного обеспечения, например технологий моделирования PTV.

Программный комплекс PTV VISSIM позволяет имитировать движение транспорта и пешеходов (имитационное моделирование), что позволяет ему быть необходимым инструментом при анализе проектных и организационных решений, если речь идёт о планировании движения.

Программа позволяет [8]:

- Оценить влияния типа пересечения дорог на пропускную способность (нерегулируемый перекрёсток, регулируемый перекрёсток, круговое движение).
- Осуществлять проектирование, тестирование и оценку влияния режима работы светофора на характер транспортного потока (рисунок 9).
- Осуществлять детальную имитацию движения каждого участника движения.

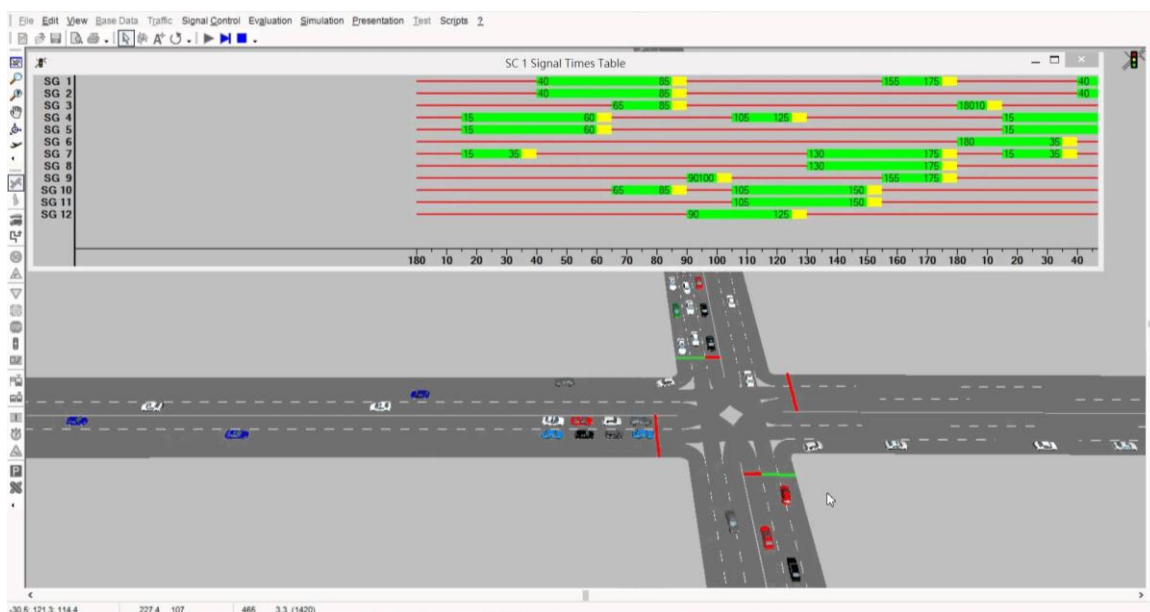


Рисунок 9 – Модель движения транспортных средств в PTV VISSIM

Реализованный в программе PTV Vision VISSIM алгоритм, работает по количеству проезжающих через перекресток автомобилей, где запрещающий сигнал включается, когда последний автомобиль на одной из дорог покидает перекресток. Однако этот алгоритм не учитывает, по каким направлениям прибывают и убывают автомобили, а также пропускную способность каждого из направлений.

Оптимизационный пакет регулирования Transyt-7FR – единственное на сегодняшний день в мире программное средство, которое сочетает в себе возможность проведения оптимизационного расчета режимов работы светофора с современным представлением о макроскопическом моделировании транспортных потоков, учитывающим такие факторы, как прирост очереди, изменение параметров транспортных потоков [7]. Возможности интерфейса и алгоритмов программы приведены на рисунке 10.

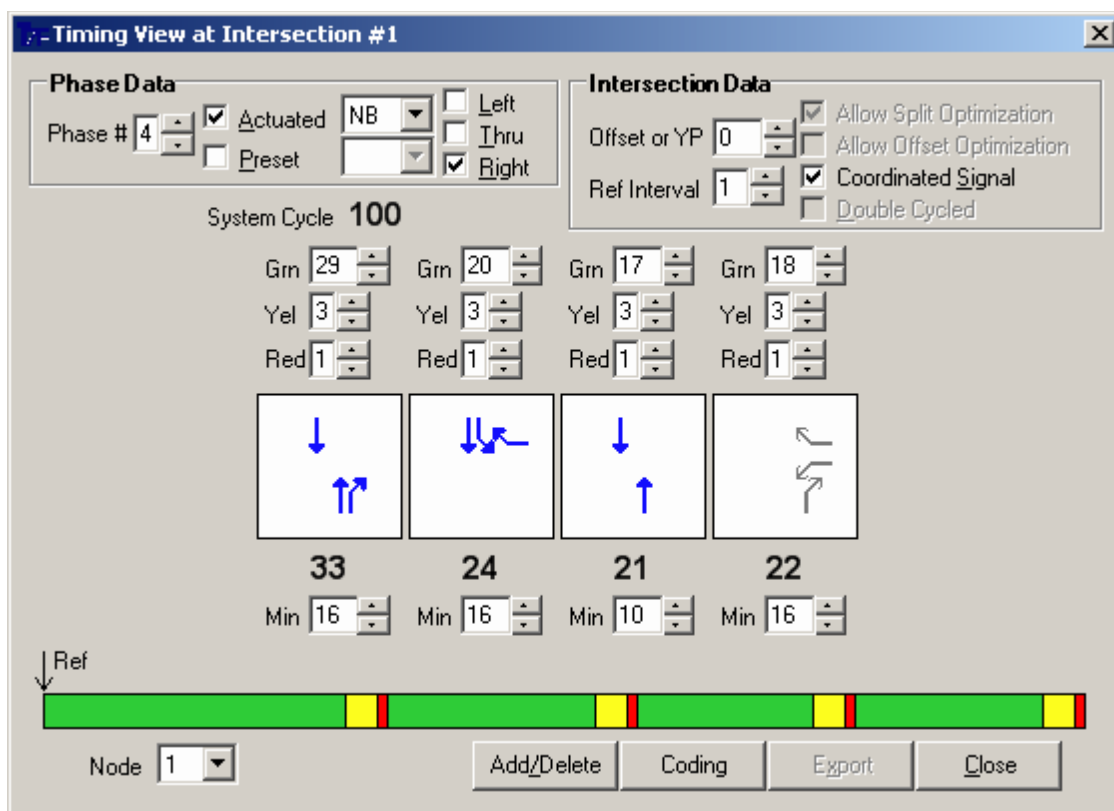


Рисунок 10 – Интерфейса программного пакета для конфигурирования «умных» светофоров

Среди программ, применяемых с целью определения оптимальных параметров светофорного регулирования, можно также отметить PASSER V, PRESYNCHRO, OSCADY [28].

В настоящее время в России число программных продуктов в области организации дорожного движения на регулируемых пересечениях также увеличивается. Рассмотрим возможности отечественного программного продукта на примере программы «СВЕТОФОР» [3].

Программа «СВЕТОФОР» обеспечивает проектирование режимов жесткого регулирования при пофазном управлении движения (рисунок 11), а также оценку эффективности режимов регулирования.

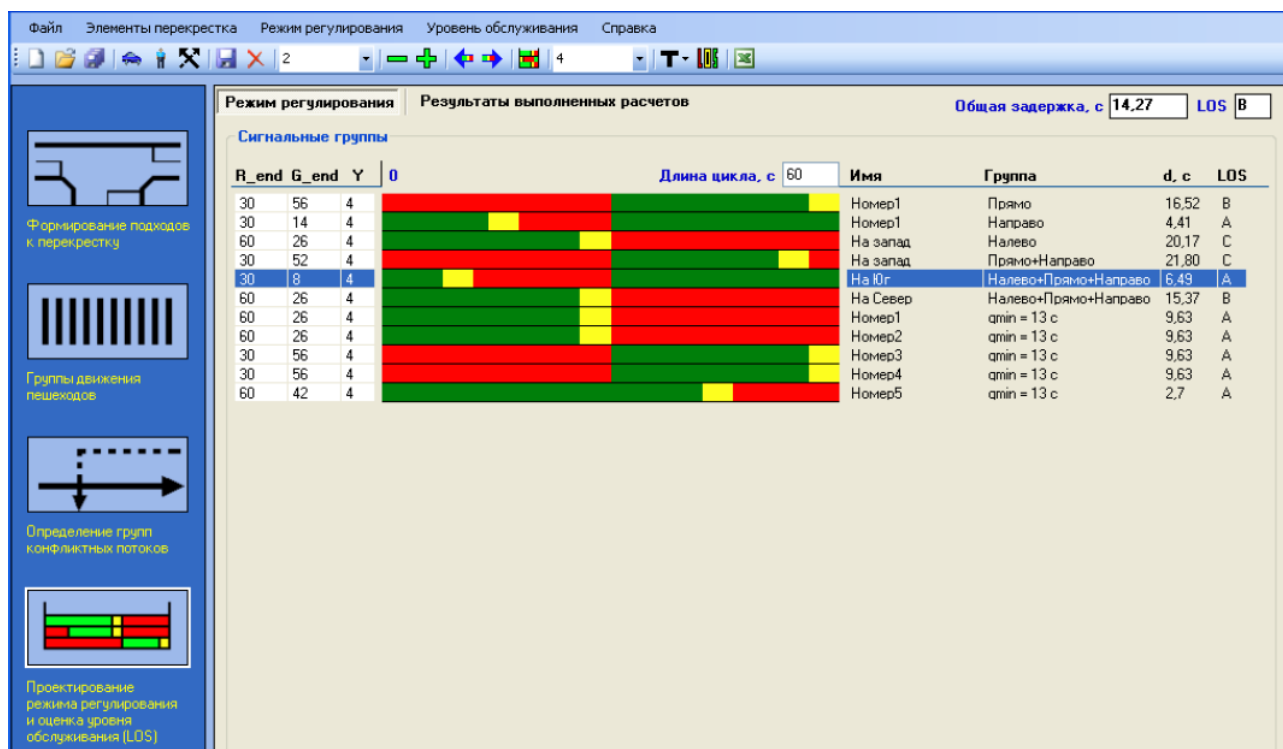


Рисунок 11 – Окно проектирования режима регулирования

Программа предоставляет расчет задержек регулирования на всех элементах регулируемого пересечения (полоса, группа движения, подход, перекресток в целом), позволяя минимизировать общую задержку за счет баланса фазовых коэффициентов.

В программе можно изменять такие параметры, как длительности цикла регулирования и основных тактов, потерянное время в начале и в конце фазы, количество фаз регулирования (до пяти фаз), интенсивности движения по всем направлениям на перекрестке, количество полос на каждом из подходов, их ширину и виды движения на них. Также представляется возможным учитывать влияние на движение транспорта, вызванное таким внешним фактором, как влияние предыдущего светофорного объекта на рассматриваемый перекресток.

2.8 Алгоритм нахождения оптимального цикла светофора

В рамках задачи требуется определить, при каких условиях перед светофором не будет скапливаться очередь.

Рассмотрим математическую модель проезда транспортным потоком многополосного регулируемого перекрестка, базирующуюся на

использовании теории систем массового обслуживания (СМО) [13] с изменяющейся интенсивностью обслуживания. Такой подход предложен и изучен в работах [10][17]. Числом обслуживающих каналов m будет количество полос для проезда перекрестка в одну сторону ($m=1,2,3$).

Задача заключается в исследовании зависимости длины очереди от интенсивности входного потока, а также нахождения оптимального режима работы светофора при заданных интенсивностях входного потока автомобилей.

Пусть всего имеется N полос движения, включая входящие в перекресток и выходящие из нее (рисунок 12).

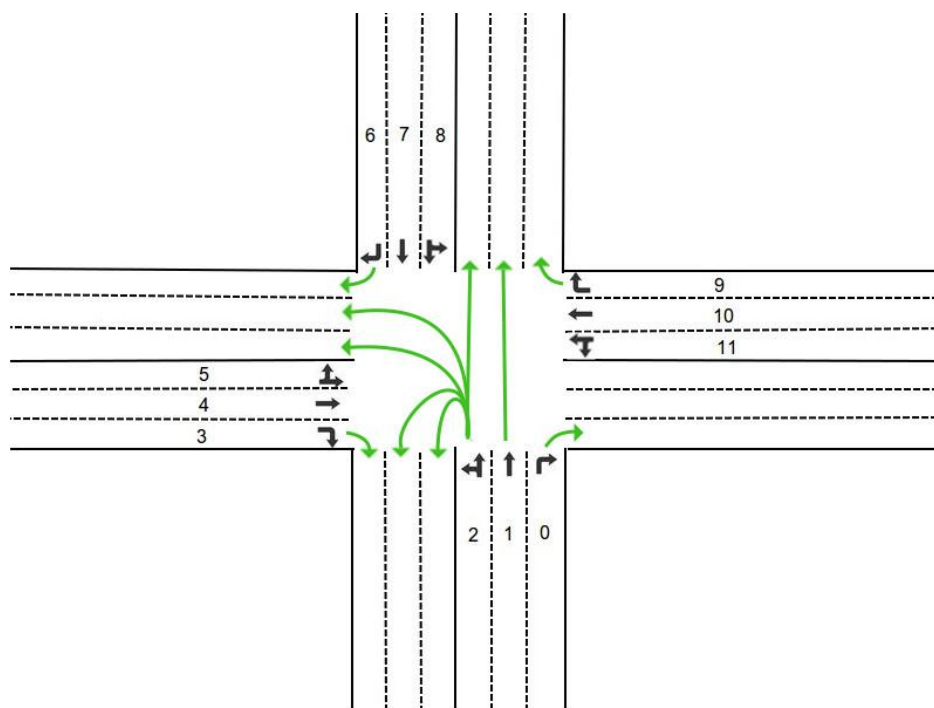


Рисунок 12 – Перекресток со схемой движения на каждой полосе

Возможные направления движения полос указаны черными знаками. Зеленые стрелки показывают одну фазу светофора, в которой разрешено движение для групп совместимых полос.

Каждую полосу можно рассматривать как независимое СМО, типа $M/M/1/m$ FIFO (First in First out). Здесь, используется компонентное символическое обозначение системы массового обслуживания, предложенное Кендаллом (Candall):

- λ_i – интенсивность потока автомобильного транспортного средства (далее АТС) для i -ой полосы;
- μ_i – интенсивность обслуживания i -ой полосы;
- q_i – текущая длина очереди i -ой полосы;
- T – длина интервала одного цикла;
- τ_i – интервал времени разрешающего (зеленого) сигнала i -ой полосы;
- ρ_i – интенсивность нагрузки полосы, то есть в среднем за время выхода одного АТС с перекрестка приезжает в среднем ρ_i на перекресток с i -ой полосой.

$$Q_i(\bar{\lambda}, \mu_i, q_i) = q_i + \lambda_i(T - \tau_i) + \tau_i \rho_i - w_i, \quad w_i = \begin{cases} q_i + \tau_i \rho_i, & \mu_i \tau_i \geq q_i + \tau_i \rho_i \\ \mu_i \tau_i, & \text{else} \end{cases} \quad (12)$$

где, Q_i – прогноз длины (загрузка) очереди в конце текущего или в начале следующего интервала.

Формула (12) предложенная в данной работе является основой для расчета и управления светофорных сигналов. Эту формулу можно интерпретировать следующим образом. По прогнозу на основе входных параметров, λ_i, μ_i , имея q_i АТС на i -ой полосе, за время запрещающего (красный) сигнала $(T - \tau_i)$ приезжают $(T - \tau_i)\lambda_i$ АТС, выезжает w_i из данной полосы, и приезжают $\tau_i \rho_i$ АТС во время разрешающего (зеленый) сигнала. Понятно, что длины очередей не могут быть отрицательным, поэтому функция w_i условная. Если выезжающие АТС больше чем текущей длины очереди и приезжающие АТС во время зеленого сигнала ($\mu_i \tau_i \geq q_i + \tau_i \rho_i$), то из очереди вычитается текущая длина и плюс к этому, количество АТС которое добавляется во время зеленого сигнала $(q_i + \tau_i \rho_i)$, иначе вычитается столько сколько выезжает за это время $(\mu_i \tau_i)$.

В данной работе количество отказов является одним из показателей эффективности системы управления светофорами. Основная цель оптимизации является оптимальное распределение времени цикла светофора между N светофорами. Выбор такого подхода для оптимизации обосновывается тем, что входящие потоки и длины полос не подлежат параметрам управления или оптимизации. Другими словами, длина полоса и параметры как интенсивность входящего/выходящего потока являются входными параметрами системы, а не подлежат изменению. Таким образом, в качестве задачи оптимизации, ищется оптимальное распределение времени между светофорами, так чтобы время распределилось соответственно загрузке каналов. Для вычисления загрузки каналов будем использовать формулу (12). Эта формула дает прогноз о загрузках полос в конце каждого цикла светофора.

Каждое направление дороги состоит только из одной полосы или из нескольких полос. Также, что в реальной ситуации один светофор управляет движением одной или нескольких полос. В предлагаемой модели, каждая отдельная полоса является отдельной независимой СМО. Такой подход позволяет данной модели иметь гибкость и универсальность при применении модели для любых конфигурации дорог и перекресток.

В данном случае оптимальность означает максимальное использование пропускной способности (зеленого сигнала) перекрестка. Другими словами, необходимо избегать ситуацию, когда у светофора горит зеленый сигнал, а очередь АТС за ним пустая.

3 Описание разработанного программного обеспечения

С учетом всех факторов, в качестве средств реализации требуемых программных модулей для решения поставленных задач были выбраны следующие программные продукты и технологии:

- язык программирования C# и платформа .NET 4.0;
- интегрированная среда разработки (IDE) Microsoft Visual Studio 2012;
- библиотека микроскопического имитационного моделирования CityTrafficSimulator.

3.1 Функционал программного обеспечения

Построение дорожной сети и имитация движения автомобилей выполняется в двухмерной системе координат. Дорожная сеть – это взаимосвязанные проезжие части с установленными на них светофорами и местами генерации автомобилей. На карте моделируемого города выполняется построение дорог с определенным количеством полос, учитывается длина и ширина проезжей части, а на перекрестках устанавливаются светофоры. В определенных местах на карте устанавливаются генераторы – источники транспортного потока. После чего модель запускается: генераторы создают автомобили, автомобили движутся, светофоры управляют движением.

3.2 Структура программного обеспечения

Транспортная модель представлена классом ModelManager – наследником интерфейсов ITickable и ISavable. С точки зрения пользователя данный класс решает следующие задачи:

- создание модели, добавление транспортных потоков и настройка их параметров;
- доступ к спискам всех потоков, транспортных средств, перекрестков и светофоров;
- добавление обработчиков событий, запуск и остановка модели.

С точки зрения других элементов модели ModelManager:

- хранит текущее состояние модели и координирует работу всех ее элементов;
- запускает обработчики происходящих событий.

Список методов и полей класса ModelManager приведен на рисунке 13.

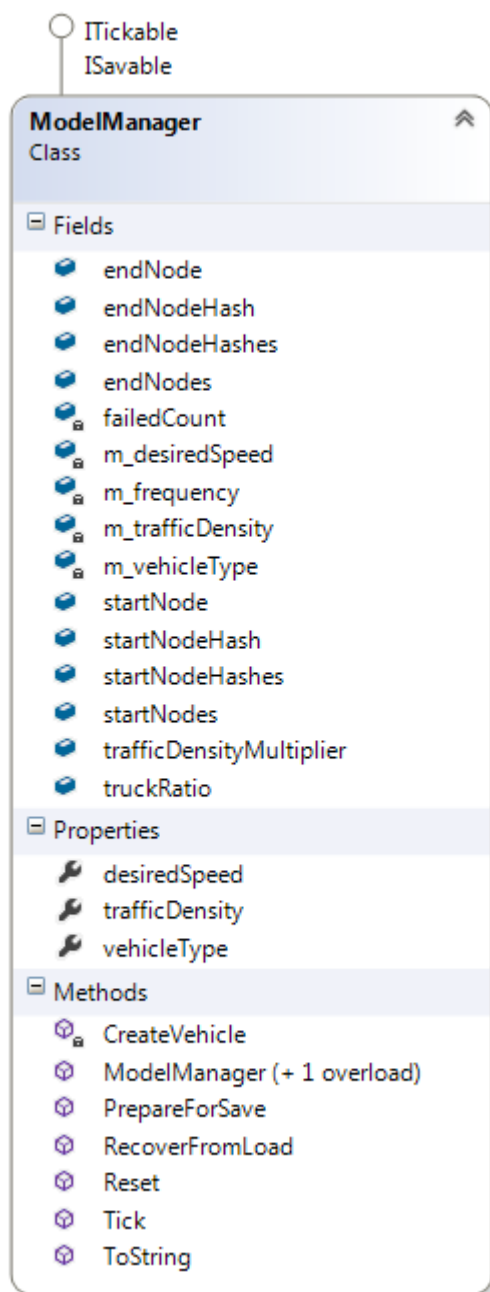


Рисунок 13 – Диаграмма класса ModelManager

Класс TrafficControl контролирует объем транспортного потока, который представлен классом TrafficVolume. Полный список полей и методов указанных классов представлен на диаграмме классов,

изображенной на рисунке 14.

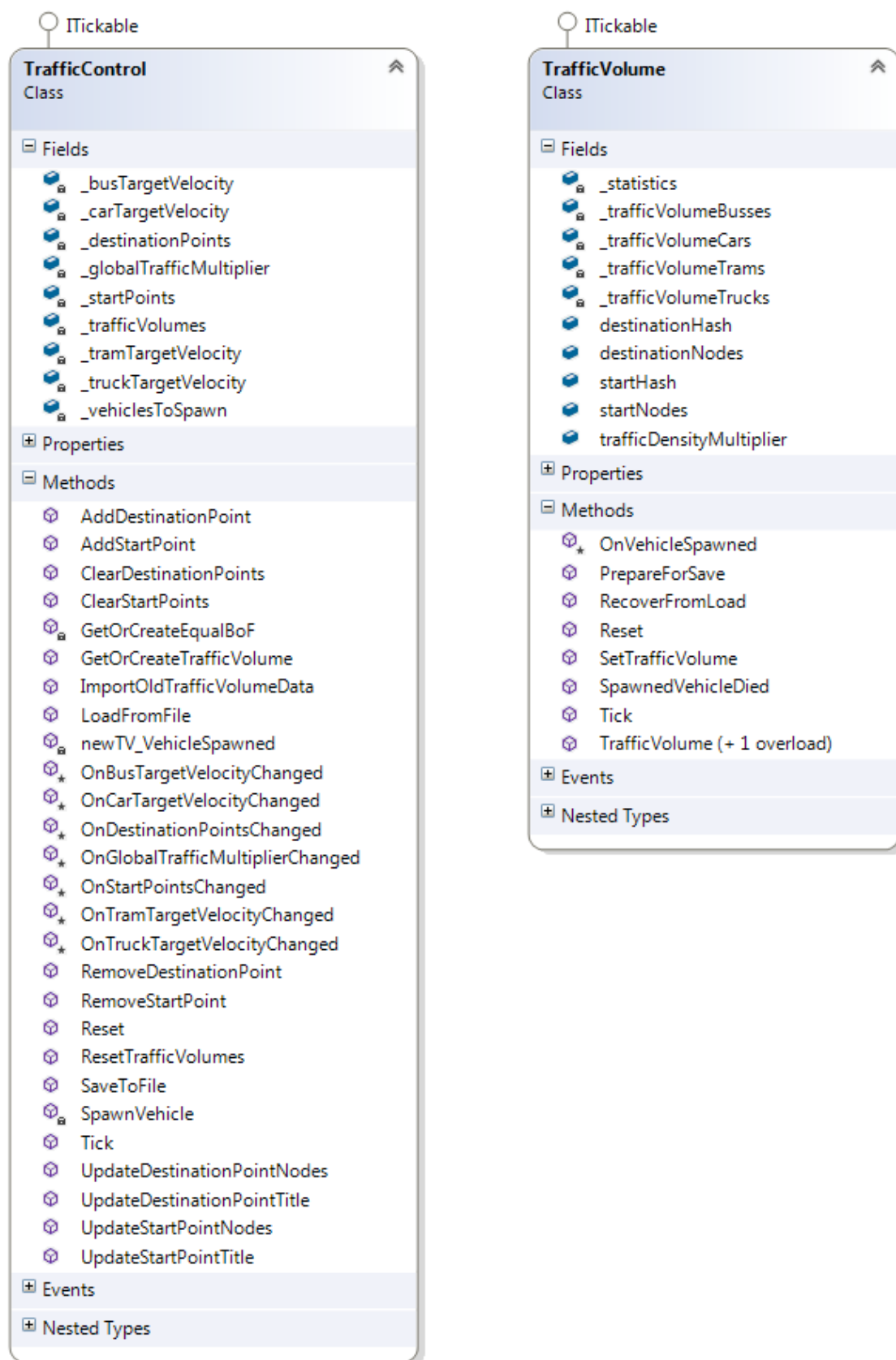


Рисунок 14 – Диаграмма классов TrafficControl и TrafficVolume

С точки зрения пользователя основное назначения класса TrafficControl

– отдавать список транспортных потоков и мест их генерации. С точки зрения других элементов данный класс предназначен для передачи сигналов о наступлении событий, связанных с изменением желаемых скоростей. Также данный класс сбрасывает статистику о накопившихся очередях автомобилей.

Перекресток представлен классом Intersection (рисунок 15).

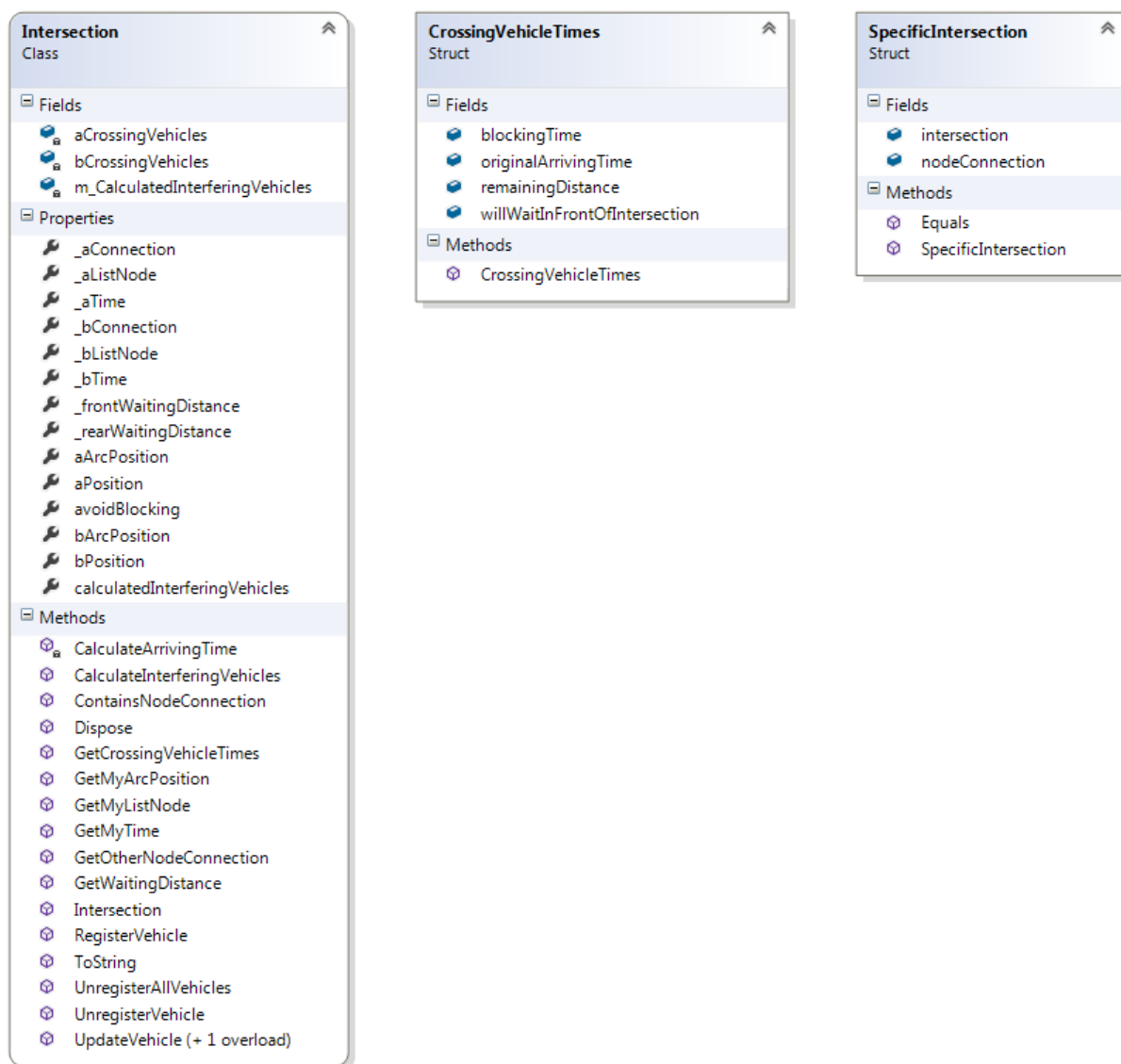


Рисунок 15 – Диаграмма класса Intersection и вспомогательных структур

Конструктор класса Intersection принимает 4 параметра: начальное и конечное узловые соединения, а также моменты времени их пересечения. Диаграмма последовательности вызова методов в конструкторе класса представлена на рисунке 16.

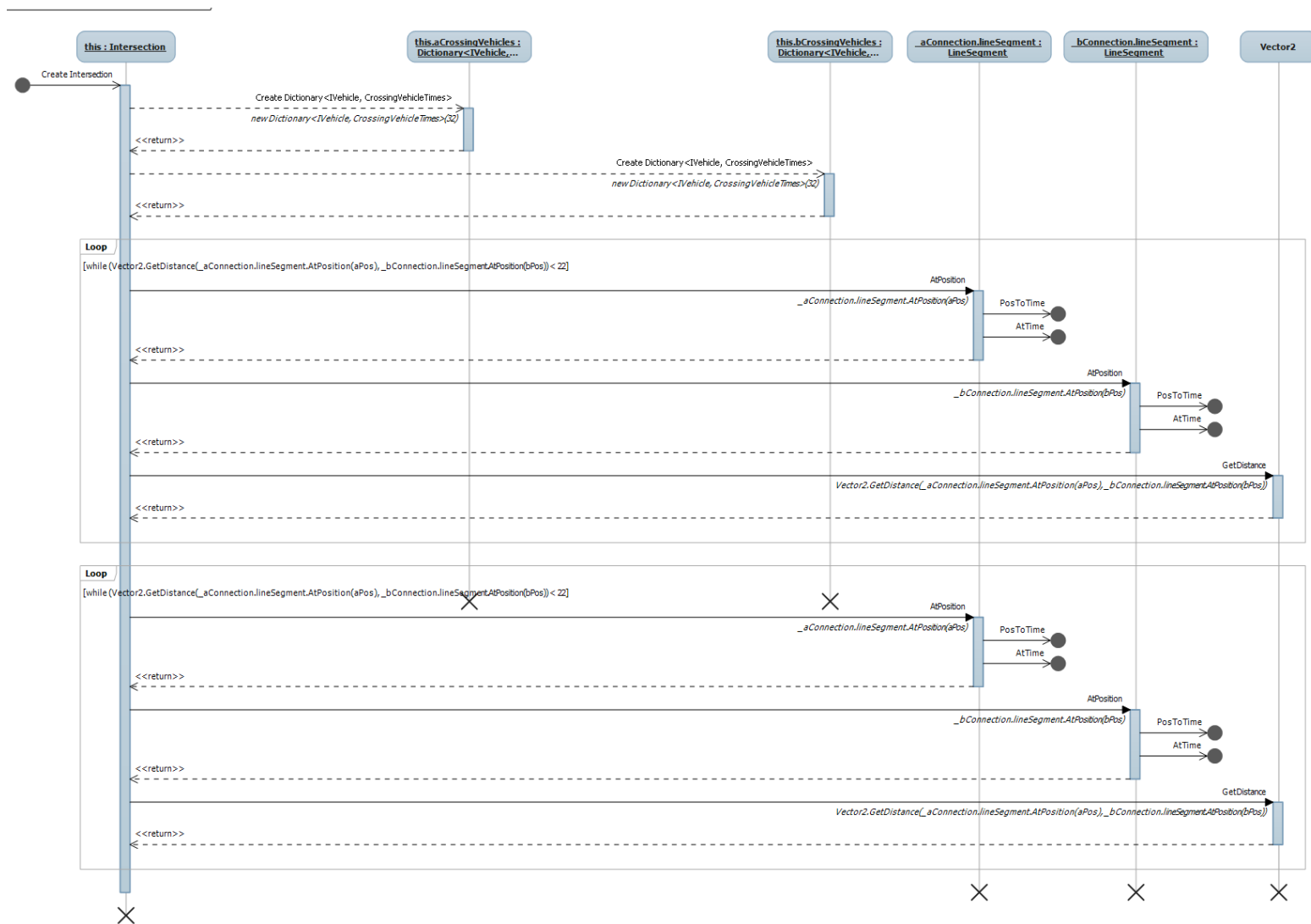


Рисунок 16 – Диаграмма последовательностей вызова конструктора класса Intersection

Класс NodeConnection (рисунок 17) обеспечивает связь между двумя узлами LineNode (полосами движения).

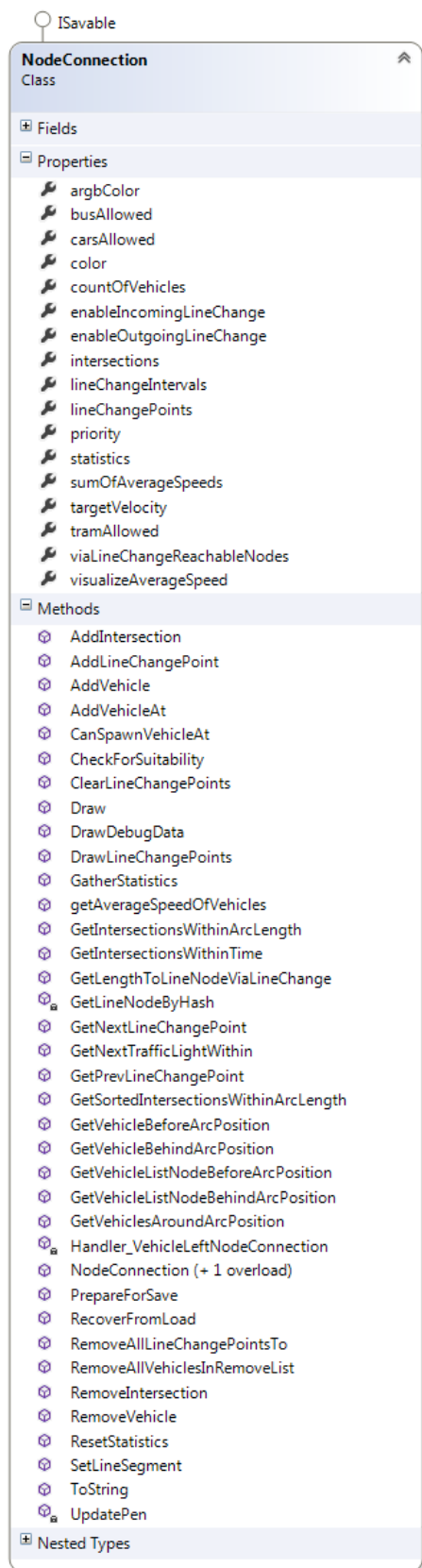


Рисунок 17 – Диаграмма класса NodeConnection

Светофорный объект представлен классом `TrafficLight` (рисунок 18). Внутри реализации светофора хранятся его первоначальное состояние (красный сигнал), список `LineNodes`, к которым данный `TrafficLight` относится.

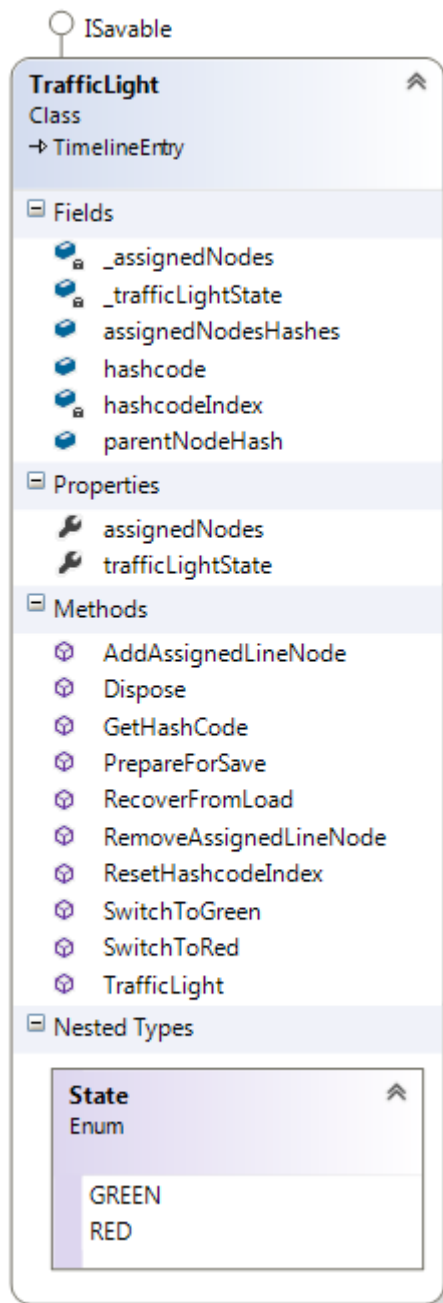


Рисунок 18 – Диаграмма класса `TrafficLight`

Транспортные средства представлены абстрактным классом `IVehicle` (наследник `IDM`), который имеет несколько реализаций – `Car`, `Bus`, `Tram`, `Truck` (рисунок 19).

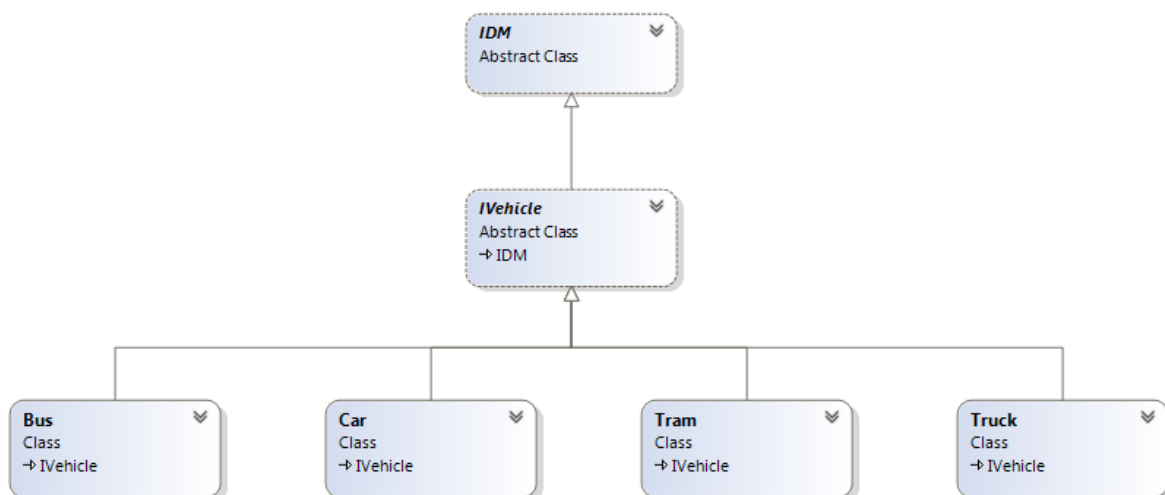


Рисунок 19 – Диаграмма классов транспортных средств

Каждый из этих классов представляет особенности движения тех или иных транспортных средств.

Полное описание полей и методов класса IDM представлено на рисунок 20.

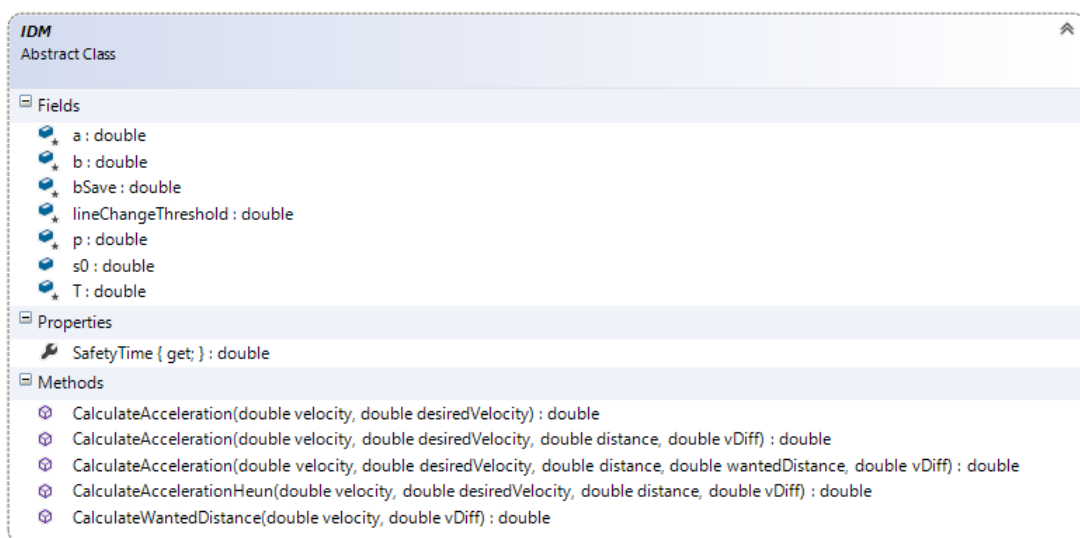


Рисунок 20 – Диаграмма класса IDM

Класс IDM описывает транспортное средство в рамках модели «умного водителя». Реализацией класса IDM является класс IVehicle, в котором реализовано движение автомобиля. Данный абстрактный класс содержит свойства (рисунок 21), которые являются параметрами модели «умный водитель» и впоследствии переопределяются классами-наследниками, реализующими конкретные виды автомобилей.

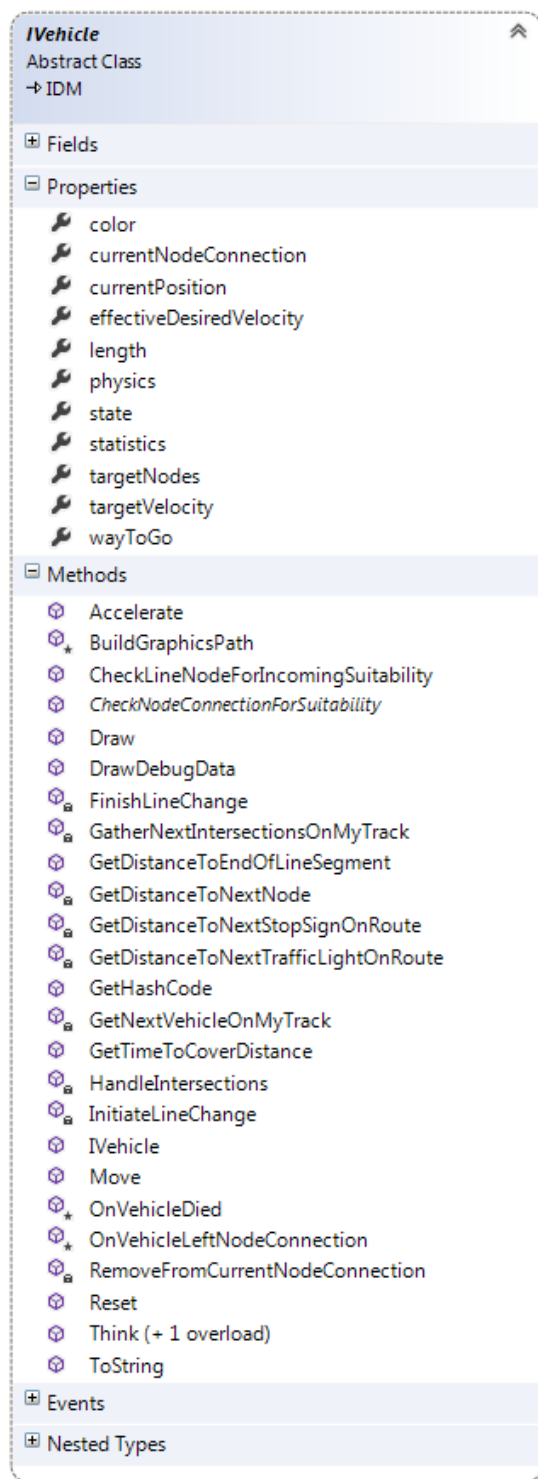


Рисунок 21 – Диаграмма класса IVehicle

Также в классе IVehicle реализован метод `InitiateLineChange`, который инициирует смену полосы движения в рамках модели MOBIL. Метод вычисления желаемого ускорения в соответствии с моделью IDM–`CalculateAccelerationHeun` – приведено на диаграмме последовательностей (рисунок 22).

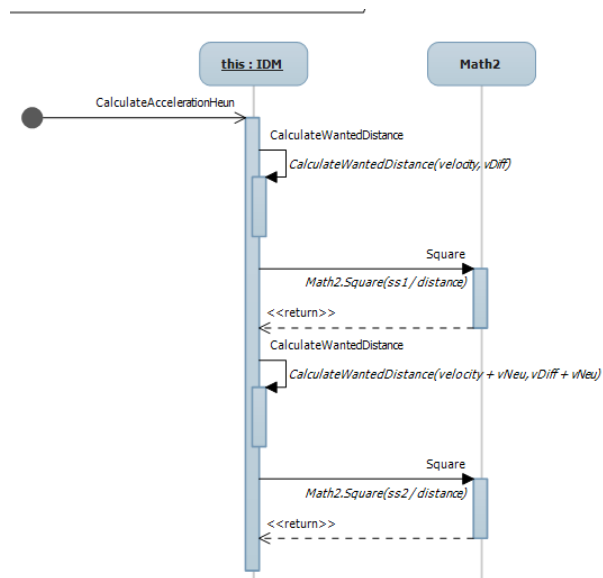


Рисунок 22 – Диаграмма последовательности расчета свободного ускорения

Метод `CalculateAccelerationHeun` принимает четыре аргумента: текущую скорость, желаемую скорость, расстояние до впереди едущего автомобиля, разницу в скорости по сравнению с впереди едущим автомобилем. Листинг методов класса `IDM` приведен в приложении А.

Для выбора оптимального маршрута предназначен класс `Routing` (рисунок 23), использующий классы `PriorityQueue` и `PriorityQueueItem` (рисунок 24).

В методе `CalculateShortestConenction` (поиска кратчайшего пути до конечного узлового соединения) класса `Routing` реализуется алгоритм A^* . Полный листинг метода приведен в приложении Б.

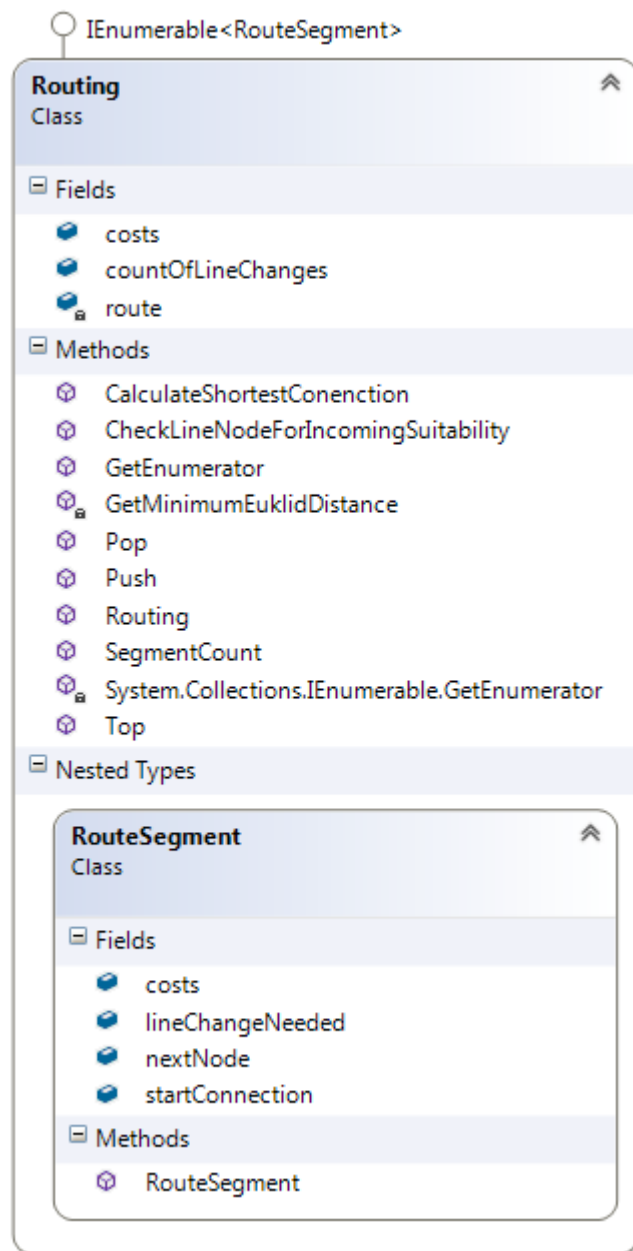


Рисунок 23 – Диаграмма класса Routing

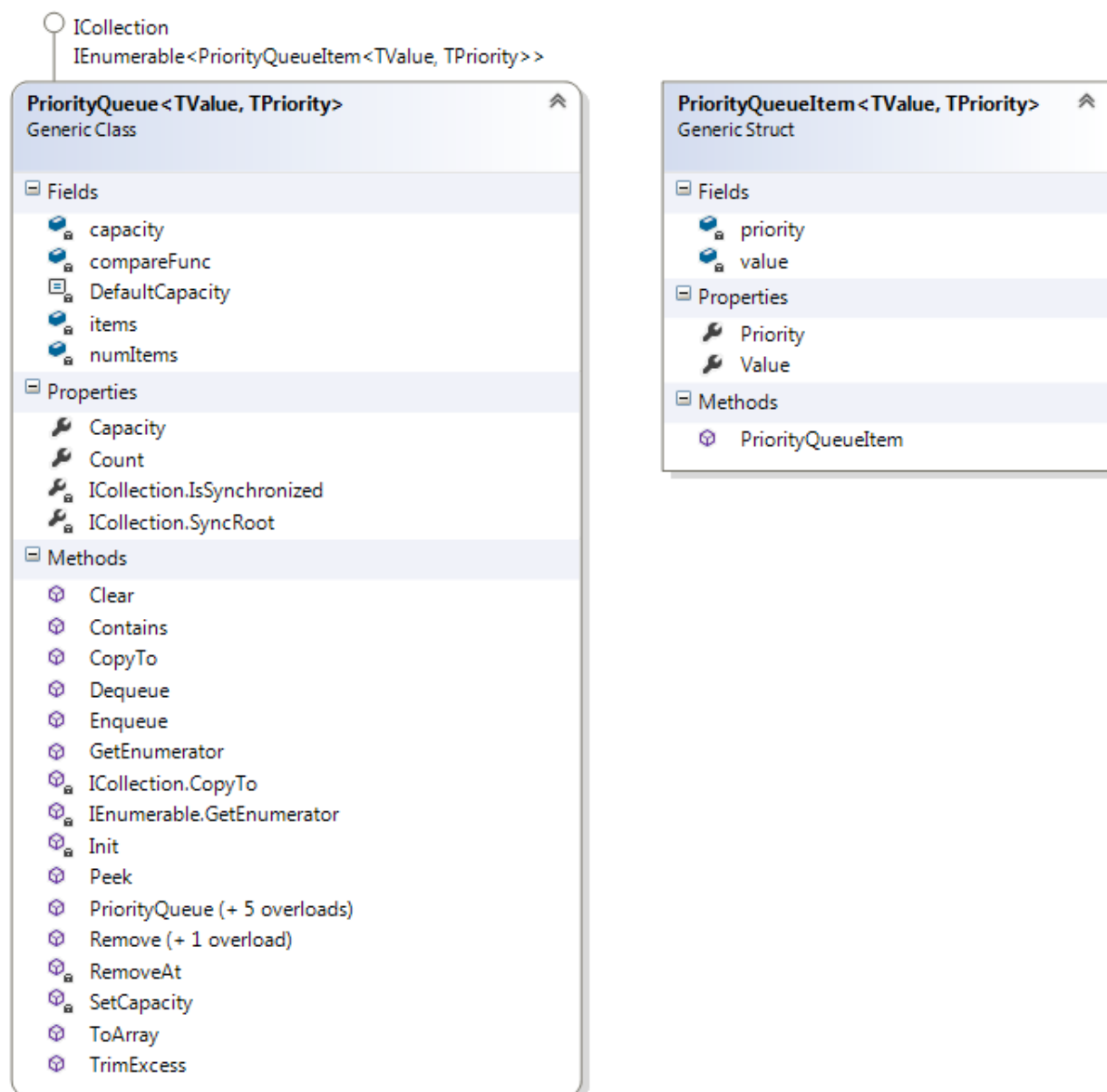


Рисунок 24 – Диаграмма классов `PriorityQueue` и `PriorityQueueItem`

3.3 Интерфейсная часть

На рисунке 25 изображен пользовательский интерфейс в режиме загрузки карты города из XML-файла и конфигурации уличной сети.

На рисунке 26 представлено окно приложения в режиме настройки интервалов и циклов работы светофорных объектов.

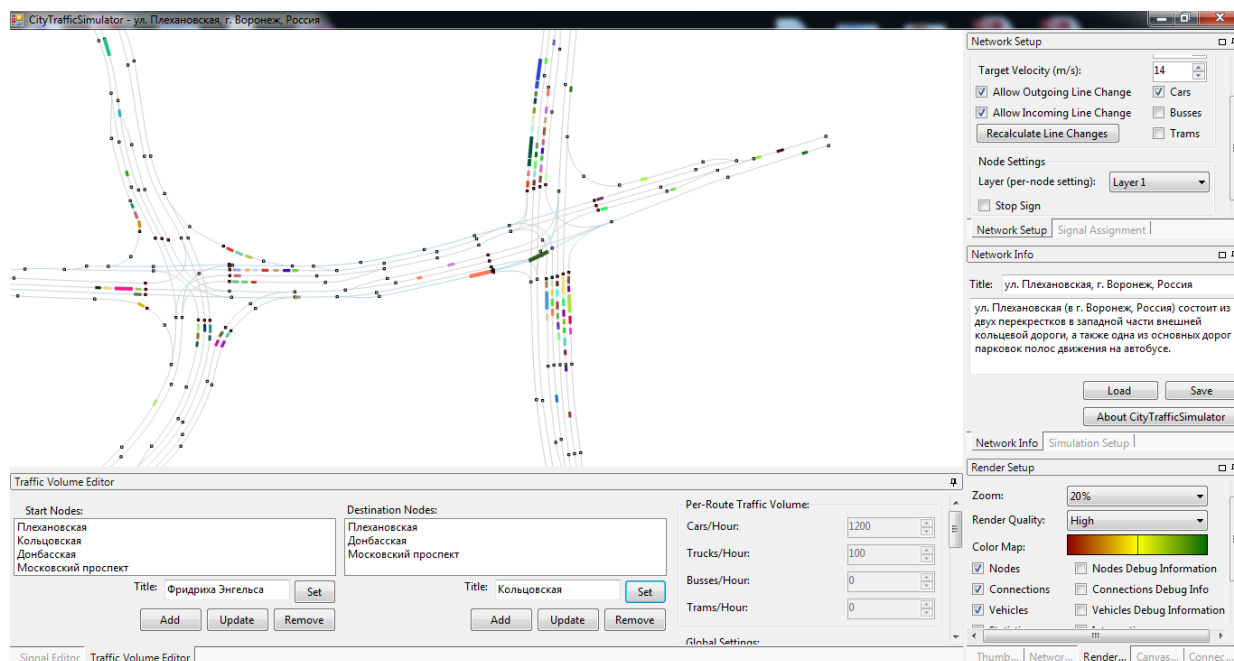


Рисунок 25 – Рабочая область приложения в режиме загрузки карты
настройки уличной сети

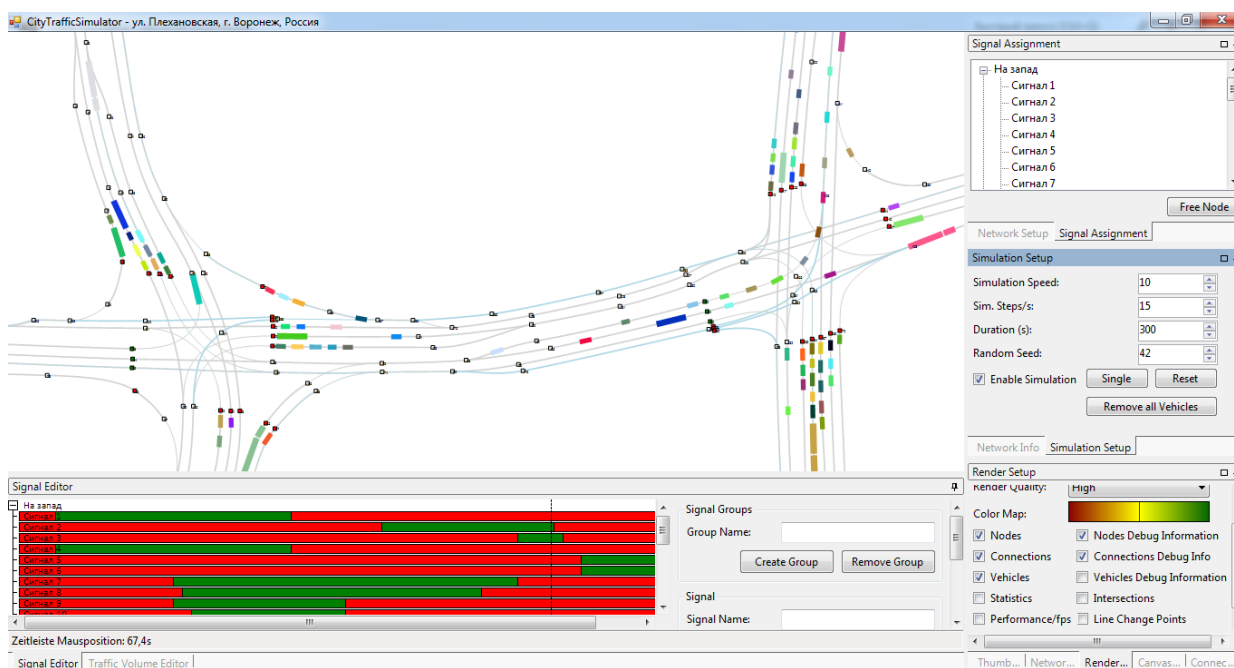


Рисунок 26 – Рабочая область приложения в режиме настройки симуляции и
редактора параметров светофорных объектов

3.4 Тестирование и апробация полученных результатов

Далее приведем результаты сравнения двух моделей: светофора с фиксированной длиной фазы и адаптивной модели. В данном тесте выделено

12 полос с фиксированными интенсивностью входящего и выходящего потока λ_i, μ_i соответственно. В ходе работы программы есть возможность изменить значения λ_i, μ_i . Генерация потоков происходит по показательному закону распределения. Для обоих методов длина интервала времени цикла светофора фиксирована и равна 240 секунд. Интервал моделирования потоков и работы светофор равен примерно 3 часа. Для фиксированной модели задана статическая карта совместимости, а для адаптивной модели карта совместимости определяется динамически в начале каждого цикла светофора. На рисунке 27 синяя и красная линия соответствуют адаптивной и фиксированной модели соответственно. Были выбраны следующие критерии для сравнения:

Mean out per cycle – среднее число выходящих АТС в одном цикле светофора, то есть $\frac{1}{T} \sum_{i=1}^N out_i$, где out_i - число выходящих АТС i -ой полосы из перекрестки и T – длина интервал одного цикла.

Total refusal – сумма отказав всех полос, то есть число сгенерированных АТС, которые не могут поехать на соответствующую полосу из-за отсутствия места для движения вперед.

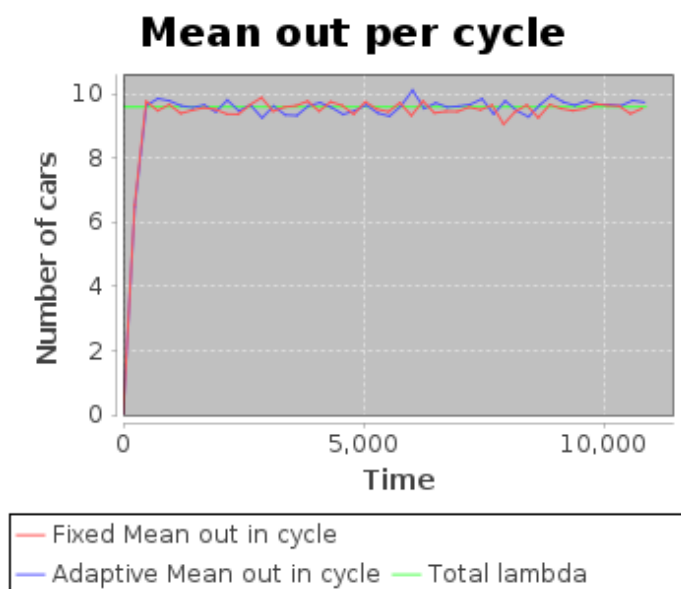


Рисунок 27 – График среднего числа выходящих АТС в одном цикле светофора

Зеленая линия соответствует числу сгенерированных АТС в каждую секунду, то есть $\sum_{i=1}^N \lambda_i$, где λ_i – интенсивность входящего потока. Видно, что число выходящих АТС адаптивной модели не меньше фиксированного в среднем.

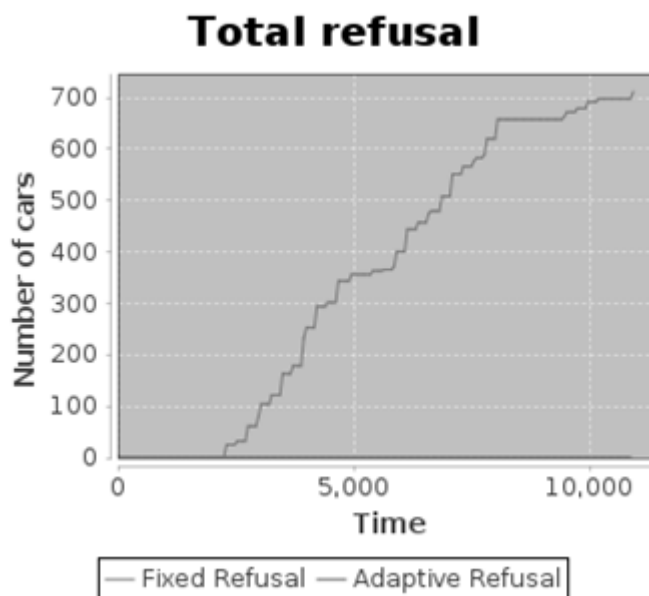


Рисунок 28 – График, изображающий общее количество отказов

Количество отказов является очень важным критерием при сравнении двух моделей, поскольку оно прямо отражает эффективность распределения времени между светофорами. Как видно на рисунке 28 адаптивная модель показывает свою эффективность по сравнению с фиксированной моделью особенно при динамическом изменении входящих параметров таких, как интенсивность входящего или выходящего потоков.

Заключение

В ходе работы была разработана модель улично-дорожной сети со светофорами на перекрестках с учетом всех вышеперечисленных требований:

- разработан микроскопический имитационный подход к моделированию;
- достигнуто безаварийное движение транспортных средств;
- произведен учет многополосных проезжих частей и перестроений транспортных средств с одной полосы на другую при наличии преимущества на другой полосе;
- установлены светофоры без дополнительных секций на каждом перекрестке;
- существует возможность задавать произвольный режим работы светофора;
- в системе предусмотрено получение разнообразной информации о транспортных средствах (количество автомобилей, прибывающих на перекресток на различные фазы светофора, длина очереди перед светофором и т.д.);
- имеется возможность произвольной конфигурации улиц и их пересечений через XML-файл;
- присутствует учет различных транспортных средств (учтены их габариты, предпочтительная скорость движения и т.д.);
- предусмотрено графическое отображение результатов работы модели в режиме реального времени;
- возможно динамически управлять режимом работы светофора.

Список использованных источников

- 1 Завалицин Д.С. Исследование математической модели регулируемого перекрестка / Д.С. Завалицин, Г.А. Тимофеева // Труды Института математики и механики УрО РАН. Екатеринбург: 2009. – Т. 15, № 4. – С. 108-119.
- 2 Семенов В. В. Смена парадигмы в теории транспортных потоков / В. В. Семенов // Институт прикладной математики им. М. В. Келдыша РАН. – Москва: 2006. – 32 с.
- 3 Краткое руководство по применению программы Светофор (версия 2.0.2) // Транспортная лаборатория ИрГТУ : [сайт]. – URL: http://transport.istu.edu/downloads/manual_ampel2.pdf (дата обращения 29.05.2016).
- 4 Швецов В.И. Основы моделирования транспортных потоков. / В.И. Швецов : [сайт]. – URL: <http://www.isa.ru/transnet/intro/Intro.pdf> (дата обращения 29.05.2016).
- 5 Автоматизированный пакет управления CO Artery // AGA Group Inc.: [сайт]. – URL: <http://www.againc.net/ru/production/its/programms/artery> (дата обращения 29.05.2016).
- 6 Пакет имитационного моделирования Aimsun // AGA Group Inc.: [сайт]. – URL: <http://www.againc.net/ru/production/its/programms/aimsun> (дата обращения 29.05.2016).
- 7 Оптимизационный пакет регулирования ДД Transyt-7FR // AGA Group Inc.: [сайт]. – URL: <http://www.againc.net/ru/production/its/programms/transyt> (дата обращения 29.05.2016).
- 8 Пакет имитационного моделирования дорожного движения PTV VISSIM // PTV Partner : [сайт]. – URL <http://ptv-vision.ru/produkty/vissim> (дата обращения 29.05.2016).
- 9 Программный продукт PTV VISUM // PTV Partner: [сайт]. – URL <http://ptv-vision.ru/produkty/visum> (дата обращения 29.05.2016).

- 10 Ахмадинуров М.М. Оптимизация светофорного регулирования с помощью программы моделирования транспортных потоков / М.М. Ахмадинуров // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». – 2010. – №22 (198). – С. 26-30.
- 11 Алюшин, С.А. Модели, методы и программные средства построения сложных адаптивных систем дорожного движения : автореферат дис. канд. техн. наук : 05.13.11 / Алюшин Сергей Александрович; Москва, Национальный исследовательский ядерный университет «МИФИ»; науч. рук. Н. И. Ильинский. – Москва, 2011. – 22 с.
- 12 Ахмадеева, Р.З. Моделирование движения транспортных потоков [Электронный ресурс] / Р.З. Ахмадеева, В.Л. Дмитриев. Стерлитамакская государственная педагогическая академия им Зайнаб Биитовой. – URL: <http://simulation.su/uploads/files/default/2012-conf-prikl-math-and-mod-41-43.pdf> (дата обращения: 29.05.2016)
- 13 Сирота А.А. Компьютерное моделирование и оценка эффективности сложных систем / А.А. Сирота. – Москва : Техносфера, 2006. – 256 с.
- 14 Osorio, C. A surrogate model for traffic optimization of congested networks: an analytic queueing network approach / C. Osorio, M. Bierlaire // Report TRANSP-OR. – 2009. – P. 1–23.
- 15 Бекмагамбетов М.М. Анализ современных программных средств транспортного моделирования / М.М. Бекмагамбетов, А.В. Кочетков // Исследования, конструкции, технологии. – 2012. – №6 (77). – URL: <http://www.aae-press.ru/f/77/25.pdf> (дата обращения 29.05.2016)
- 16 Rickert, M. Two lane traffic simulations using cellular automata / M. Rickert, K. Nagel, M. Schreckenberg, A. Latour // Physica A: Statistical Mechanics and its Applications. – 1996. – Vol. 231, № 4. – P. 534-550.
- 17 Treiber M. Congested traffic states in empirical observations and microscopic simulations / M. Treiber, A. Hennecke, D. Helbing // Physical Review E. – 2000. – Vol. 62, № 2. – P. 1805-1824.

- 18 Кременец Ю.А., Печерский М.П., Афанасьев М.Б. Технические средства организации дорожного движения: учеб. для вузов / Ю.А. Кременец, М.П. Печерский, М.Б. Афанасьев. – Москва: ИКЦ «Академкнига» - М., 2005.- 279 с.
- 19 Traffic Control Systems Handbook // FHWA Office of Operations : [сайт]. – URL: <http://ops.fhwa.dot.gov/publications/fhwahop06006/index.htm> (дата обращения 29.05.2016)
- 20 Адаптивное управление как способ автоматизирования движения транспортных потоков на регулируемых перекрестках // Агентство инноваций и развития экономических и социальных проектов: [сайт]. – URL: <http://www.innoros.ru/innovation-idea28/ideas/adaptivnoe-upravlenie-kak-sposob-avtomatizirovaniya-dvizheniya-transportnykh> (дата обращения: 29.05.2016)
- 21 Кочерга В.Г. Технические средства современных автоматизированных систем управления дорожным движением / В.Г. Кочерга, Е.Е. Шаталова. – Ростов-на-Дону : РГСУ, 2011. – 108 с.
- 22 Агуреев И.Е. Исследование алгоритмов светофорного регулирования перекрестка при различных параметрах транспортного потока / И.Е. Агуреев , А.Ю. Кретов , И.Ю. Мацур // Известия ТулГУ. Технические науки. – 2013. №7, Ч. 2. – С. 54-61.
- 23 Кретов А.Ю. Обзор некоторых адаптивных алгоритмов светофорного регулирования перекрестков / А.Ю. Кретов // Известия ТулГУ. Технические науки. – 2013. №7-2. – С. 61-67.
- 24 Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений / Л. Заде. – Москва : Мир, 1976. – 166 с.
- 25 Кущенко, Л.Е. Повышение эффективности организации движения в городе на основе минимизации заторов.: дис. канд. техн. наук: 05.22.10 / Кущенко Лилия Евгеньевна. – Белгород, 2015. - 124 с.
- 26 Андронов, С.А. Разработка и исследование имитационной модели светофорного регулирования на основе нечеткой логики в среде AnyLogic / С.А. Андронов // Труды международной конференции ИММОД'2015. – 2015.

– С. 443-449.

27 Review of Micro-Simulation Models // SMARTTEST Project : [сайт].

– URL: <http://www.its.leeds.ac.uk/projects/smartest/append3d.html> (дата обращения 29.05.2016).

28 Левашев А.Г. Проектирование регулируемых пересечений : учеб. пособие / А.Г. Левашев, А.Ю. Михайлов, И.М. Головных. – Иркутск: Изд-во ИрГТУ, 2007. – 210 с.

29 Петров В.В. Автоматизированные системы управления дорожным движением в городах : учеб. для вузов / В.В. Петров. – Омск : СибАДИ, 2007. – 104 с.

30 Adaptive Traffic Control Systems: Domestic and Foreign State of Practice // Transportation research board. Ser/ NCHRP Synthesis 403. – Washington, D.C., US: 2010 – P. 1-105.

31 Посмитный Е.В. Методика адаптивного управления транспортными потоками высокой интенсивности в условиях города на основе мезомодели динамики с применением генетических алгоритмов / Е.В. Посмитный, М.И. Медовщиков // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. – 2012. – №84(10). – URL: <http://ej.kubagro.ru/2012/10/pdf/75.pdf> (дата обращения 29.05.2016).

32 Михеева Т.И. Модели транспортных потоков в интеллектуальных транспортных системах / Т.И. Михеева, С.В. Михеев, И.Г. Богданова // Современные проблемы науки и образования. – 2013. – № 6. – URL: <http://www.science-education.ru/ru/article/view?id=11808> (дата обращения 29.05.2016)

33 Введение в математическое моделирование транспортных потоков / А.В. Гасников [и др.]. – Москва : Изд. МФТИ, 2010. – 362 с.

34 Наумова, Н.А. Теоретические основы и методы автоматизированного управления транспортными потоками средствами мезоскопического моделирования : дис. канд. техн. наук : 05.22.10 / Наумова

Наталья Александровна; Краснодар, Кубанский государственный технологический университет; науч. рук. В.В. Зырянов. – Краснодар, 2015. – 331 с.

35 Семахин А.М. Программный комплекс моделирования транспортных и пешеходных потоков на регулируемом перекрестке / А.М. Семахин, М.А. Шульгин // Молодой ученый. Ежемесячный научный журнал. – 2013. – Т. 1, №7(54). – С. 42-44.

Приложение А. Исходный код алгоритма расчета параметров IDM

```
public double CalculateAccelerationHeun(double velocity, double desiredVelocity,
    double distance, double vDiff)
{
    double ss1 = CalculateWantedDistance(velocity, vDiff);
    double vNeu = a * (1 - Math.Pow((velocity / desiredVelocity), 2) -
Math2.Square(ss1 / distance));
    double ss2 = CalculateWantedDistance(velocity + vNeu, vDiff + vNeu);
    vNeu += a * (1 - Math.Pow(((velocity + vNeu) / desiredVelocity), 2) -
Math2.Square(ss2 / distance));

    return vNeu/2;
}

/// <summary>
/// Рассчитывает желаемое расстояние в IDM
/// </summary>
/// <param name="velocity">Собственная скорость</param>
/// <param name="vDiff">Дифференциальная скорость до впереди едущего автомобиля</param>
/// <returns></returns>
public double CalculateWantedDistance(double velocity, double vDiff)
{
    double ss = s0 + T * velocity + (velocity * vDiff) / (2 * Math.Sqrt(a * b));
    if (ss < s0)
    {
        ss = s0;
    }
    return ss;
}

/// <summary>
/// Вычислить свободное ускорение после IDM
/// </summary>
/// <param name="velocity">Текущая скорость</param>
/// <param name="desiredVelocity">Желаемая скорость</param>
/// <param name="distance">Расстояние до впереди едущего автомобиля</param>
/// <param name="vDiff">Разница в скорости впереди едущего автомобиля</param>
/// <returns></returns>
public double CalculateAcceleration(double velocity, double desiredVelocity, double
    distance, double vDiff)
{
    double ss = CalculateWantedDistance(velocity, vDiff);
    double vNeu = a * (1 - Math.Pow((velocity / desiredVelocity), 2) -
Math2.Square(ss / distance));
    return vNeu;
}

/// <summary>
/// Вычислить свободное ускорение, когда ни одно транспортное средство не выезжает
вперед
/// </summary>
/// <param name="velocity">Собственная скорость</param>
/// <param name="desiredVelocity">Желаемая скорость</param>
/// <returns></returns>
public double CalculateAcceleration(double velocity, double desiredVelocity)
{
    double vNeu = a * (1 - Math.Pow((velocity / desiredVelocity), 2) );
    return vNeu;
}
```

Приложение Б. Исходный код алгоритма поиска кратчайшего пути до узлового соединения

```
/// <summary>
/// Рассчитывается кратчайший путь к targetNode и сохраняет это как Stack в WayToGo
/// Реализация A*-алгоритма
/// </summary>
/// <param name="startNode">Начальный узел должен рассчитываться из кратчайшего
пути</param>
/// <param name="targetNodes">Список конечных узлов, к одному из которых кратчайший
уть должен быть рассчитан</param>
/// <param name="vehicleType">Vehicle type</param>
public static Routing CalculateShortestConnection(LineNode startNode, List<LineNode>
    targetNodes, Vehicle.IVehicle.VehicleTypes vehicleType)
{
    PriorityQueue<LineNode.LinkedLineNode, double> openlist = new
    PriorityQueue<LineNode.LinkedLineNode, double>();
    Stack<LineNode.LinkedLineNode> closedlist = new
    Stack<LineNode.LinkedLineNode>();
    Routing toReturn = new Routing();

    openlist.Enqueue(new LineNode.LinkedLineNode(startNode, null, false), 0);

    do
    {
        PriorityQueueItem<LineNode.LinkedLineNode, double> currentNode =
            openlist.Dequeue();
        if (targetNodes.Contains(currentNode.Value.node))
        {
            closedlist.Push(currentNode.Value);
            LineNode.LinkedLineNode endnode = closedlist.Pop();
            LineNode.LinkedLineNode startnode = endnode.parent;
            while (startnode != null)
            {
                if (!endnode.lineChangeNeeded)
                {
                    toReturn.Push(new RouteSegment(
                        startnode.node.GetNodeConnectionTo(
                            endnode.node), endnode.node, false,
                            startnode.node.GetNodeConnectionTo(
                                endnode.node).lineSegment.length));
                }
                else
                {
                    NodeConnection formerConnection =
                        startnode.parent.node.
                            GetNodeConnectionTo(startnode.node);
                    double length = formerConnection.
                        GetLengthToLineNodeViaLineChange(endnode.node)
                        + Constants.lineChangePenalty;
                    if ((endnode.node.tLight != null) ||
                        (startnode.node.tLight != null))
                        length +=
                            Constants.lineChangeBeforeTrafficLightPenalty;

                    toReturn.Push(new
                        RouteSegment(formerConnection,
                            endnode.node, true, length));
                }
                endnode = startnode;
            }
        }
    }
}
```

```

        startnode = startnode.parent;
    }

    endnode = startnode;
    startnode = startnode.parent;
}
return toReturn;
}

foreach (NodeConnection nc in
    currentNode.Value.node.nextConnections)
{
    if (!nc.CheckForSuitability(vehicleType))
        continue;

    LineNode.LinkedLineNode successor = new
        LineNode.LinkedLineNode(nc.endNode, null, false);
    bool nodeInClosedList = false;
    foreach (LineNode.LinkedLineNode llNode in closedlist)
        if (llNode.node == successor.node)
        {
            nodeInClosedList = true;
            continue;
        }

    if (!nodeInClosedList)
    {
        NodeConnection theConnection =
            currentNode.Value.node.GetNodeConnectionTo(
                successor.node);
        double f = currentNode.Value.length
            + theConnection.lineSegment.length;
        if (currentNode.Value.countOfParents < 3)
        {
            f += theConnection.vehicles.Count *
                Constants.vehicleOnRoutePenalty;
        }
        f += GetMinimumEuklidDistance(successor.node,
            targetNodes);
        f *= 14 / theConnection.targetVelocity;
        f *= -1;

        bool nodeInOpenlist = false;
        foreach (
            PriorityQueueItem<LineNode.LinkedLineNode,
            double> pqi in openlist)
        {
            if (pqi.Value.node == successor.node)
            {
                if (f <= pqi.Priority)
                    nodeInOpenlist = true;
                else
                    openlist.Remove(pqi.Value);
                break;
            }
        }

        if (!nodeInOpenlist)
        {
            successor.parent = currentNode.Value;
            openlist.Enqueue(successor, f);
        }
    }
}

```



```

    }
}

if (currentNode.Value.parent != null)
{
    NodeConnection currentConnection =
currentNode.Value.parent.node.GetNodeConnectionTo(currentNode.Value.node);
    if (currentConnection != null)
    {
        foreach (LineNode ln in
currentConnection.viaLineChangeReachableNodes)
        {

            if (!CheckLineNodeForIncomingSuitability(ln,
vehicleType))
                continue;

            LineNode.LinkedLineNode successor = new
LineNode.LinkedLineNode(ln, null, true);
            bool nodeInClosedList = false;
            foreach (LineNode.LinkedLineNode llNode in
closedlist)
            {
                if (llNode.node == successor.node)
                {
                    nodeInClosedList = true;
                    break;
                }
            }

            if (!nodeInClosedList)
            {
                NodeConnection.LineChangeInterval
lci;

currentConnection.lineChangeIntervals.TryGetValue(ln.hashcode, out lci);

                if (lci.length <
Constants.minimumLineChangeLength)

                    break;

                double f = currentNode.Value.parent.length;
                f +=
currentConnection.GetLengthToLineNodeViaLineChange(successor.node);
                f += (lci.length < 2 *
Constants.minimumLineChangeLength) ? 2 * Constants.lineChangePenalty :
Constants.lineChangePenalty;

                if ((lci.targetNode.tLight != null) ||
(currentConnection.startNode.tLight != null))
                {
                    f += Constants.lineChangeBeforeTrafficLightPenalty;
                    f += GetMinimumEuklidDistance(successor.node,
targetNodes);
                    f *= -1;
                    bool nodeInOpenlist = false;
                    foreach (PriorityQueueItem<LineNode.LinkedLineNode,
double> pqi in openlist)
                    {
                        if (pqi.Value.node == successor.node)
                        {
                            if (f <= pqi.Priority)
                                nodeInOpenlist = true;

```

```

                                else
                                    openlist.Remove(pqi.Value);
                                break;
                            }
                        }
                    if (!nodeInOpenlist)
                    {
                        successor.parent = currentNode.Value;
                        openlist.Enqueue(successor, f);
                    }
                }
            }
        }
    }

    closedlist.Push(currentNode.Value);
}
while (openlist.Count != 0);
return toReturn;
}

```