

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

---

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Пензенский государственный университет  
архитектуры и строительства»  
(ПГУАС)

**А.А. Власов, Н.А. Орлов**

**УПРАВЛЕНИЕ  
НАСЫЩЕННЫМИ ТРАНСПОРТНЫМИ  
ПОТОКАМИ В ГОРОДАХ**

Пенза 2014

УДК 656.1  
ББК 39.3  
В58

Рецензенты: доктор технических наук, профессор  
Ю.В. Родионов (ПГУАС);  
доктор технических наук В.В. Коновалов

**Власов А.А.**  
В58 Управление насыщенными транспортными потоками в городах:  
моногр. / А.А. Власов, Н.А. Орлов. – Пенза: ПГУАС, 2014. – 188 с.  
**ISBN 978-5-9282-1133-2**

Рассмотрены основные вопросы построения современных адаптивных систем управления дорожным движением. Обобщен и систематизирован мировой опыт разработки систем, использующих обратную связь «транспорт – управление – транспорт». Изложены способы управления изолированными пересечениями и транспортными сетями в условиях насыщенного движения. Рассмотрены математические модели и алгоритмы оптимального управления транспортными потоками.

Монография подготовлена на кафедре «Организация и безопасность движения» и предназначена для студентов, обучающихся по направлению 23.04.01 «Технология транспортных процессов», аспирантов и инженерно-технических работников, занимающихся проектированием, внедрением и эксплуатацией АСУДД.

**ISBN 978-5-9282-1133-2**

© Пензенский государственный университет  
архитектуры и строительства, 2014  
© Власов А.А., Орлов Н.А., 2014

## О Г Л А В Л Е Н И Е

ВВЕДЕНИЕ .....	5
1. АНАЛИЗ СУЩЕСТВУЮЩИХ МЕТОДОВ УПРАВЛЕНИЯ СВЕТОФОРНЫМИ ОБЪЕКТАМИ.....	6
1.1 Существующие методы светофорного регулирования .....	8
1.2. Алгоритмическое обеспечение локального управления светофорными объектами .....	11
1.3. Алгоритмическое обеспечение магистрального управления светофорными объектами .....	14
1.3.2. Магистральные методы управления на основе прогрессии .....	16
1.3.2. Магистральные методы управления сигнальными планами, основанные на индексе потерь.....	21
1.3.3. Интеграция методов потерь и прогрессии для магистрального управления .....	26
1.3.4. Управление в реальном времени путем коррекции предрасчитанных сигнальных планов.....	27
1.4. Управление в условиях заторов.....	29
2. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ УПРАВЛЕНИЯ НАСЫЩЕННЫМИ ТРАНСПОРТНЫМИ ПОТОКАМИ .....	32
2.1. Математическая модель задержки и длины очереди на изолированном перекрестке .....	32
2.2. Математическая модель сетевого движения транспортных потоков.....	41
2.3. Динамика очереди транспортных средств перед регулируемым перекрестком .....	47
3. ПОСТАНОВКА ЗАДАЧИ УПРАВЛЕНИЯ НАСЫЩЕННЫМИ ТРАНСПОРТНЫМИ ПОТОКАМИ .....	52
3.1. Управление транспортными потоками с использованием прогнозирующей модели .....	52
3.2. Формализация задачи локального управления светофорным объектом .....	55
3.3. Формализация задачи управления магистралью .....	64
4. АПРОБАЦИЯ МЕТОДИКИ УПРАВЛЕНИЯ НАСЫЩЕННЫМИ ТРАНСПОРТНЫМИ ПОТОКАМИ .....	69
4.1. Методика проведения исследований.....	69
4.2. Калибровка микроскопической модели.....	71
4.3. Оценка эффективности локального управления светофорным объектом .....	79

4.4. Оценка эффективности сетевого управления светофорными объектами методом балансировки объемов движения.....	88
ЗАКЛЮЧЕНИЕ.....	93
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	95
ПРИЛОЖЕНИЯ .....	103
Приложение 1 .....	103
Приложение 2 .....	108
Приложение 3 .....	111
Приложение 4 .....	122
Приложение 5 .....	124
Приложение 6 .....	125
Приложение 7 .....	128
Приложение 8 .....	132
Приложение 9 .....	134
Приложение 10 .....	136
Приложение 11 .....	143
Приложение 12 .....	153
Приложение 13 .....	161

## ВВЕДЕНИЕ

Рост автотранспортного парка сопровождается резким увеличением интенсивности движения на автомобильных дорогах и городских улицах. Транспортная инфраструктура, в свою очередь, не может развиваться в таком же высоком темпе. Строительство новых и реконструкция существующих дорог является дорогим и трудоёмким процессом, требующим значительного времени. В результате транспортные заторы в городах стали повседневным явлением, с каждым годом их число и продолжительность неуклонно увеличатся. Заторы приводят к экономическим потерям из-за продолжительных задержек, которые они вызывают, и они также оказывают негативное воздействие на окружающую среду, увеличивая уровень шума и загрязнение токсичными компонентами отработанных газов.

Предотвратить или, по крайней мере, уменьшить перегрузку транспортной сети возможно за счет более эффективного использования имеющихся дорог в сочетании с использованием современных методов управления дорожным движением.

Данное научное исследование, посвященное разработке методов повышения эффективности организации дорожного движения на регулируемых пересечениях в условиях насыщенного движения.

# 1. АНАЛИЗ СУЩЕСТВУЮЩИХ МЕТОДОВ УПРАВЛЕНИЯ СВЕТОФОРНЫМИ ОБЪЕКТАМИ

Транспортным потоком можно управлять тремя способами – остановкой транспортных средств, изменением параметров движения, информацией и изменением направления движения. В условиях городского движения используются все три вышеуказанных метода управления.

**Управление путем остановки транспортных средств.** Принцип управления путем остановки транспортных средств является основой управления транспортными потоками в городах. Типичным для этого способа является управление посредством светофоров. На каждый транспортный поток, движущийся через перекресток, воздействуют три световых сигнала — зеленый, желтый и красный. Однако при внешней простоте управляющие воздействия на светофорную сигнализацию имеют достаточно сложную структуру, обусловленную спецификой объекта управления. Данные воздействия можно представить в виде многомерного вектора, компонентами которого являются:

- длительности промежуточных тактов (переходных интервалов), во время которых происходят закрытие и подготовка к открытию направлений движения;
- длительность цикла (Cycle) – интервал времени, в течение которого происходит смена всех разрешенных тактов;
- разбиение (Split) эффективной длительности цикла между фазами регулирования;
- временные сдвиги (Offset) между включениями фаз на соседних перекрестках, влияющие на возможность безостановочного движения транспортных средств по дорожной сети;
- последовательность фаз в цикле, обуславливающая очередность открытия движений на перекрестке.

В дальнейшем под сигнальным планом (планом координации для сети перекрестков) будем понимать многомерный вектор, компонентами которого являются перечисленные параметры, устанавливаемые для каждого перекрестка.

Кроме светофоров в качестве исполнительных элементов для остановки транспортных средств могут использоваться и управляемые дорожные знаки «Въезд запрещен», светофоры для движения по полосам или механические препятствия – шлагбаумы.

Существенным недостатком управления путем остановки является ухудшение экологических условий, вызванное повышенным расходом

топлива, расходуемого транспортными средствами для восстановления кинетической энергии после остановки.

**Управление путем изменения характеристик движения.** Управление путем изменения характеристик движения транспортных потоков заключается в том, что транспортное средство не останавливается, но изменяет параметры своего движения, как, например, скорость на основании знака «Ограничение максимальной скорости» или транспортное средство движется по указанной полосе. Этот способ управления используется не только для гармонизации транспортного потока при управлении движением по полосам автомагистралей, но он имеет большое значение и в городских системах обеспечения безопасности движения, когда ограничение максимальной скорости вводится перед опасным поворотом или водитель получает информацию о другой опасности.

Исполнительными элементами в этом случае являются управляемые дорожные знаки на базе световодов или светодиодов. Информация на этих знаках оперативно реагирует на изменяющиеся параметры транспортного потока. Эффективность таких систем получила высокую оценку, так как динамические изменения адекватнее воспринимаются водителями и, следовательно, ими соблюдаются.

**Управление информацией и направлением движения.** Третья категория управления транспортным потоком основана на изменении маршрута движения в результате получения водителем информации об условиях движения. Этот способ управления все чаще используется в городах при перегрузке дорог и заключается в направлении транспортного потока в обход участка, на котором образуются заторы. В эту категорию управления входит управление информацией, предназначенной как для индивидуального транспортного средства (Vehicle Information and Communication System), так и в целом для транспортного потока (Traffic Flow Information System).

Информационные и направляющие системы в зависимости от характера объекта управления располагаются:

- непосредственно в автомобиле при управлении индивидуальными транспортными средствами;
- рядом с дорогой при воздействии на весь транспортный поток.

Принципиальная разница между информационными и направляющими системами заключается в том, что при управлении информацией водитель получает информацию о транспортной ситуации на его пути, и принимает решение о дальнейшем маршруте движения самостоятельно. При воздействии направляющих систем транспортные средства направляются принудительно на новый маршрут движения.

В случае применения информационных систем в индивидуальных транспортных средствах водитель может отказаться от предложенного

маршрута движения. Результаты некоторых европейских проектов показали, что существенные изменения в транспортной ситуации наблюдаются в том случае, если 10-15 % водителей следуют рекомендациям информационных систем.

В случае применения направляющих систем, транспортные средства направляются на новые маршруты движения управляемыми запрещающими и предписывающими знаками. Этот способ управления предъявляет жесткие требования к техническому оснащению, так как он нуждается в обозначении маршрута движения по всей ее длине.

В городских условиях при применении информационных и направляющих систем необходима согласованность изменения маршрутов движения с работой светофорных объектов.

## 1.1 Существующие методы светофорного регулирования

Светофорное регулирование первоначально вводилось для обеспечения безопасного пересечения конфликтующих транспортных потоков автомобилей и пешеходов. Рост транспортного спроса вскоре показал, что при установке светофора появляется возможность осуществлять (при обеспечении безопасности движения) более или менее эффективное воздействие на транспортный поток в рамках транспортной сети. Указанный эффект послужил основой при разработке стратегий оптимального управления, выполняющих минимизацию полного времени движения.

Светофорное регулирование – основной способ управления в транспортных сетях городов. Современное состояние управления транспортными потоками в большинстве городов можно вообще характеризовать так, что устройства управления (узлы) управляются по фиксированному графику или по состоянию транспортного потока (адаптивное управление). Существенная разница заключается в том, что для управления по графику времени не нужны детекторы, но система неспособна реагировать на какие-либо изменения транспортного потока. В случае адаптивного управления имеются детекторы, которые фиксируют присутствие транспортных средств, и устройство управления, реагирующее на мгновенные условия в узле или транспортной сети.

**Временно-зависимое (автономное) управление.** Управление ведется по временному графику. Применение метода требует всестороннего изучения характеристик состояния транспортных потоков в районе управления. Информацию получают путем статистического анализа данные о характеристиках движения транспортных потоков (интенсивности и состава движения) за прошлые годы, измеренных в характерных точках транспортной сети, и на их основании определяется режим работы системы управления.



На основании эпизодических измерений параметров транспортных потоков ( $\mathbf{X}$ ) производится, в основном вручную, анализ эффективности действующих планов –  $F_{\text{Э}}(\mathbf{X})$  и сравнение ее с контрольными значениями –  $F_{\text{КЭ}}(\mathbf{X})$ .

При  $F_{\text{КЭ}}(\mathbf{X}) - F_{\text{Э}}(\mathbf{X}) \geq \mathbf{A}$  производится перерасчет библиотеки планов управления (координации)  $\{\mathbf{Y}_i\}$  и контрольных значений времен их действия  $\{t_k\}$ . Ввод в действие плана  $\mathbf{Y}_i$ , производится при совпадении реального времени  $t_p$  с контрольным  $t_k$ .

При расчетах оптимизируется длительность разрешающих сигналов, продолжительность цикла и временной сдвиг. Затем они вводятся в устройства управления и производят смену сигнальных планов в зависимости от времени суток или дня года.

Временно-зависимое управление имеет определенные преимущества:

- возможность простого контроля;
- простота модификации сигнальных планов;
- относительно низкие расходы на оборудование и установку.

Вместе с тем, имеются существенные недостатки метода:

- невозможно повысить эффективность использования времени разрешающих сигналов;
- нельзя покрыть пики интенсивности;
- нельзя устранить возникшие транспортные заторы.

**Транспортно-зависимое (traffic-responsive) управление** характеризуется тем, что для различных состояний транспортных потоков на сети заранее рассчитываются сигнальные планы, которые хранятся в устройствах управления или в центре управления движением. В области управления устанавливаются стратегические детекторы транспорта. Логика управления производит выбор конкретных сигнальных планов по состоянию транспортного потока в зоне расположения указанных детекторов транспорта.

Смена сигнальных планов происходит в результате анализа параметров транспортных потоков, который обычно сводится к их усреднению в течение заранее заданного периода времени  $T_y$  и получению оценки:

$$\vec{X}(T_y) = \frac{\sum^n \vec{X}_i}{n},$$

где  $n$  – количество измерений в течение  $T_y$ ;  $\vec{X}_i$  – многомерный вектор – результат измерения параметров в заранее заданных сечениях дорожной сети.

После получения оценки  $\vec{X}(T_y)$  осуществляется подбор ближайшего многомерного контрольного вектора  $\vec{X}_k(i)$  из заранее заданного множества ( $i=1...M$ ) по минимуму вектора разницы  $\vec{R}_{\min}$

$$\vec{X}_i(T_y)R < \vec{X}_k(i) = \vec{R}_{\min}.$$

Для каждого контрольного вектора имеется заранее рассчитанный сигнальный план. При принятии решения о вводе сигнального плана взамен действующего необходимо учитывать наличие переходного интервала  $T_{\text{пер}}$ , в течение которого эффект от координированного управления резко снижается и практически становится равным нулю. Поэтому решение о смене плана координации  $Y_i$ , на план  $Y_j$ - целесообразно принимать при условии

$$F_3(Y_i)T_y - F_3(Y_j)(T_y - T_{\text{пер}}),$$

где  $F_3(Y_i)$  – эффективность управления за единицу времени при действующем плане и соответственно разнице;  $F_3(Y_j)$  – эффективность нового плана.

Практические исследования, проведенные на действующих системах, показывают, что длительность  $T_{\text{пер}}$  составляет два, три цикла управления. Оценки  $F_3(Y_j)$  и  $F_3(Y_i)$  могут быть получены моделированием. Длительность  $T_y$  может быть выбрана по критерию минимальной ошибки усреднения измеряемых параметров.

Практически невозможно оборудовать детекторами транспорта все разрешенные направления движения в районе управления. Обычно возникает необходимость выбора представительного подмножества точек измерения. В этом случае целесообразным было бы соблюдение равенства:

$$\frac{dX^*}{dt} = \frac{dX(t)}{dt},$$

где левая часть представляет полную производную изменения нормы вектора параметров в точках измерения, правая – полную производную нормы вектора параметров на всех разрешенных направлениях движения.

Соответствующим подбором точек измерения возможно обеспечить:

$$\frac{dX^*}{dt} - \frac{dX(t)}{dt} \leq C,$$

где  $C$ – допустимое отклонение.

Методика выбора точек измерения основана на трудоемких длительных измерениях с выявлением максимальной области представительства в каждой предполагаемой точке.

**Управление в реальном времени.** Под этим термином понимается такое системное управление, при котором запаздывание управляющих воздействий, рассчитываемых в процессе функционирования системы, не превышает интервала, при котором можно пренебречь нестационарностью транспортных потоков. Системы, реализующие управление в реальном времени, получили название адаптивные системы управления дорожным движением (Adaptive Traffic Control Systems – ATCS). Они требуют наличия развитой системы контроля состояния транспортных потоков и коммуникационной инфраструктуры, обеспечивающей связь периферийного оборудования с управляющим центром.

Адаптивные системы управления дорожным движением, как правило, оптимизируют в реальном времени некоторый функционал:

$$Q(X,U) \rightarrow \min ,$$

где  $X$  – вектор состояния управляемых объектов;  $U$  – вектор соответствующих управляющих воздействий.

Хотя соответствующая задача об оптимальном управлении может быть сформулирована для любой транспортной сети, ее решение в реальном времени и реализация в системе управления имеет ряд особенностей:

- переключение режимов красный-зеленый светофора требует введения дискретных переменных, которое делает задачу об оптимизации комбинаторной;
- размер задачи для целой сети является очень большим;
- высокий уровень неопределенности, вызванный непредсказуемыми и трудноизмеримыми возмущениями (аварии, парковка в неустановленных местах, пешеходные переходы, блокирование пересечений, и т.д.) транспортного потока;
- измерения состояний транспортного потока являются локальными (через детекторы транспорта) и очень зашумленными из-за различных эффектов;
- имеются жесткие ограничения реального времени.

В связи с этим, стратегии управления дорожным движением используют различные упрощения или решают только часть связанных задач управления движением.

## 1.2. Алгоритмическое обеспечение локального управления светофорными объектами

Цель алгоритмов местного гибкого регулирования состоит в том, чтобы обеспечить нахождение оптимального управления для указанного периода времени для изолированного пересечения. Есть два подхода к оптимизации управления – бинарный подход и метод катящийся горизон-

та. При использовании бинарного подхода оптимальное решение принимается на основе анализа решения о смене фазы, рассматривающего преимущество продления текущего зеленого сигнала против потерь при его завершении. В методе катящегося горизонта управление оптимизировано в пределах горизонта прогноза, используя для этого поисковые методы оптимизации или динамическое программирование.

Одним из наиболее известных алгоритмов адаптивного управления – метод поиска разрыва в транспортном потоке, который, как правило, используется на пересечениях с двухфазной схемой движения. Для каждой фазы устанавливается минимальная длительность разрешающих сигналов. Если никакое транспортное средство не проходит через детектор транспорта во время действия минимального разрешающего сигнала, происходит переход к следующей фазе. Если же было обнаружено транспортное средство, создается временной интервал (экипажное время) который позволяет транспортному средству покинуть пересечение. Во время указанного интервала любое обнаруженное транспортное средство продляет действие разрешающего сигнала до тех пор, пока не появится достаточно большой интервал между транспортными средствами или длительность фазы не превысит предустановленного максимального значения. В алгоритме поиска разрыва в потоке также может учитываться транспортный спрос на конфликтном направлении при решении о переходе к следующей фазе.

Более сложный вариант был предложен в [79]. Нахождение параметров сигнала основано на выборе между продлением разрешающего сигнала и его завершением. Алгоритм модифицирует рассчитываемые сигнальные планы в малых временных шагах (порядка 2 секунд). В каждом временном шаге используется функция оптимизации, основанная на задержке транспортных средств, для оценки преимуществ продления разрешающего сигнала на 2, 4, 6, 8 и 10 секунд, в сравнении с немедленным завершением фазы. Если преимущества положительны, текущий разрешающий сигнал продляется в течение двух секунд, и функция управления вычисляется вновь. Однако, в данном алгоритме, длительность последующей запрещающей фазы, как предполагается, равна длине запрещающей фазы в предыдущем цикле. Данное предположение не корректно, так как не учитывается количество автомобилей, стоящей в очереди на запрещающем сигнале.

В методе **Traffic Optimization Logic** (TOL), предложенном в [19], оптимальное управление продлевает разрешающий сигнал на  $h$  секунд или немедленно завершает на основе вычисления значений целевой функции системы. В отличие от алгоритма Миллера, TOL назначает продолжительность следующей запрещающей фазы как продолжительность разрешающей фазы, необходимой для обеспечения разгрузки очереди автомобилей. Кроме того, TOL рассматривает только два варианта настройки текущего

разрешающего сигнала – завершение фазы немедленно или продление на  $h$  секунд.

Алгоритм **Microprocessor Optimized Vehicle Actuation** (MOVA), предложенный в [106], принимает решение относительно длительности разрешающего сигнала, основываясь на величине транспортного потока и информации о длине очереди, полученной от детекторов. В MOVA каждому подходу предоставляют длительность разрешающего сигнала, необходимую для разъезда очереди между стоп – линией и датчиком, расположенным вверх по ходу движения в 40 метрах. После завершения минимального времени, разрешающий сигнал продляется до тех пор, пока, по крайней мере, одна полоса движения подхода не разгрузится ниже нормы насыщения. По окончании разъезда очереди решение о продлении или окончании разрешающего сигнала выполняется на основе индекса работы, который является комбинацией задержки и остановок транспортных средств.

Алгоритмы, использующие бинарный подход рассматривает только очень короткий горизонт прогноза и, следовательно, не гарантирует глобального оптимума управления. Следовательно, их работа в свободных условиях движения хуже, чем транспортно-зависимых системы управления.

Метод катящегося горизонта для решения задачи оптимизации сигнальных планов был изначально предложен Gartner N.H. в методе Optimization Policies for Adaptive Control (OPAC) [39]. Алгоритм разработан для оптимизации работы изолированных пересечений в реальном времени. Он использует метод последовательного поиска, для определения оптимального сигнального плана. В OPAC оптимальный сигнальный план, определяется для периода времени, известного как стадия оптимизации или горизонт прогноза, разделенного на множество временных интервалов. Реализуемые последовательности сигналов определены для всех временных интервалов, причем последовательность, обеспечивающая минимальный индекс работы, выбирается как оптимальный план.

Индекс работы в данном случае представляется взвешенной суммой задержек и остановок. Оптимальный план переключения реализуется только для первых нескольких временных интервалов, для которых состояние транспортного потока непосредственно может быть измерено детекторами транспорта. Полевые исследования показали, что система OPAC является наиболее эффективной при высоком объеме движения. Был разработан вариант OPAC для работы в системе координированного управления группой пересечений [43]. Данный вариант OPAC использовал метод ограниченного поиска оптимальных последовательностей, который не гарантирует глобальный оптимум.

**Controlled Optimization of Phases (COP)** [100], оптимизирует управление сигналами светофоров в пересечении, используя динамическое программирование для минимизации индекса работы, который является комбинацией задержек, остановок, и длин очереди, с ограничениями на минимальный разрешающий сигнал и переходный интервал для обеспечения безопасности. Метод также позволяет определить порядок, а при необходимости и пропуск отдельных фаз. Алгоритм COP очень чувствителен к прибытию транспортных средств. Это ожидаемо, так как три составляющих индекса работы являются функциями прибытия транспортного средства, следовательно, если предсказание прибытия транспортного средства будет неверным, то сигнальный план будет не эффективным.

Алгоритм COP использует упрощенную, краткосрочную модель предсказания потока прибытия транспортных средств, используя для этого данные детекторов, сигнальные планы и длины очередей расположенных вверх по ходу движения пересечений. Кроме того, модель предсказания использует постоянные скорости свободного потока для оценки времени прибытия автомобилей в пересечение.

### 1.3. Алгоритмическое обеспечение магистрального управления светофорными объектами

Основой магистрального управления светофорными объектами послужили исследования эффекта распада групп автомобилей [59] при их движении от стоп-линии (рис. 1.1).

Проведенный анализ задержки автомобилей при предрассчитанных режимах работы светофоров, используя наблюдаемые профили транспортного потока, позволил сделать следующие выводы:

- детерминированное время задержки (первое слагаемое в аппроксимированной формуле задержки), сильно зависит от запаздывания включением разрешающего сигнала светофора, расположенного в начале транспортной связи (эффект смещения);

- минимальное время задержки, наблюдаемое при оптимальном смещении, существенно увеличивается при увеличении расстояния между пересечениями;

- смещение, по всей видимости, не влияет на составляющую задержки перенасыщения.

На основе указанных исследований были разработаны методы координированного управления городскими транспортными сетями. Наибольшее распространение получили методы максимизации прогрессии групп (в отечественной литературе известен как метод максимизации ширины ленты времени) и методы, минимизирующие функцию потерь.

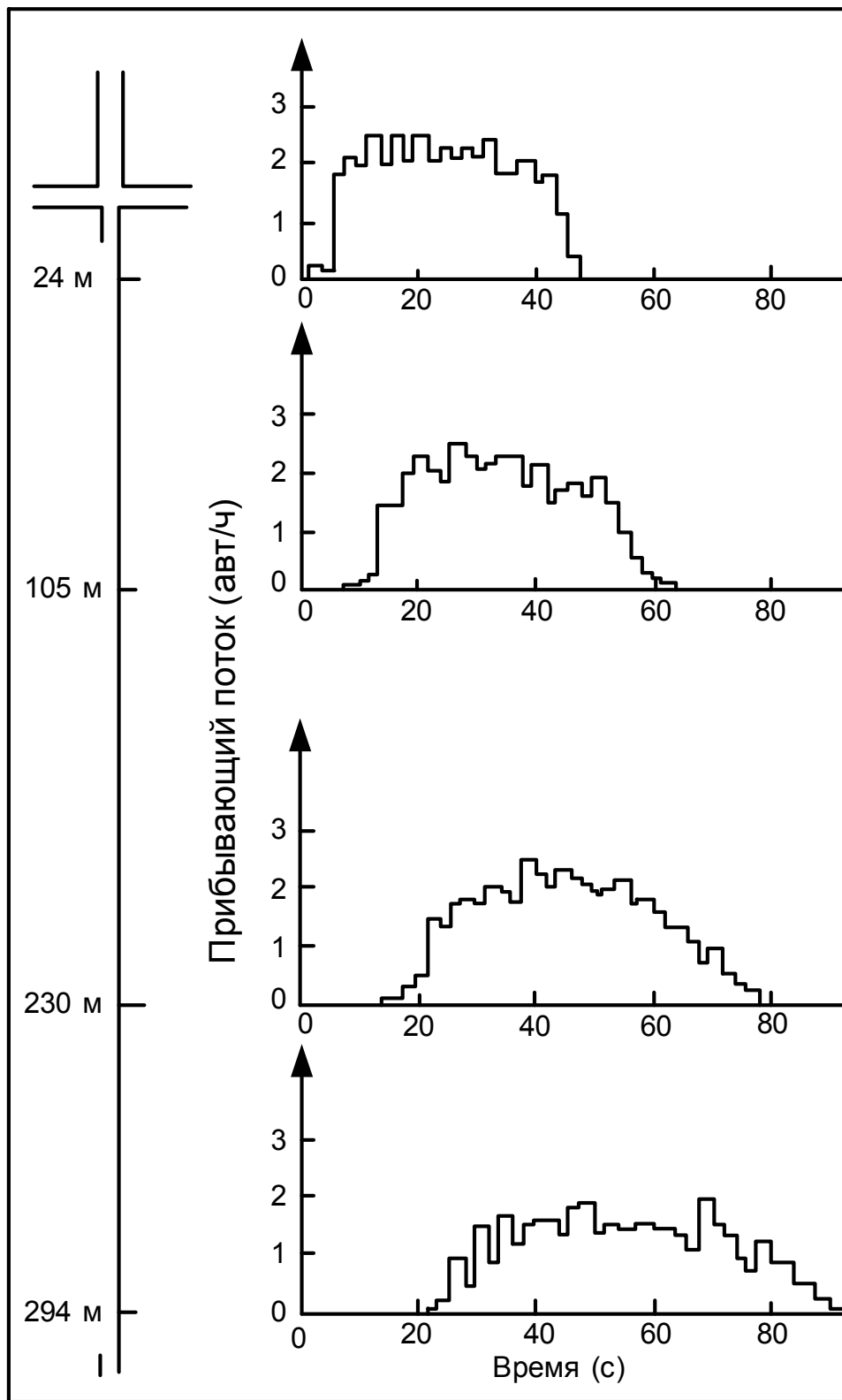


Рис. 1.1. Распад групп автомобилей по данным Hiller and Rothery

### 1.3.2. Магистральные методы управления на основе прогрессии

Большинство методов максимизации прогрессии пытается установить разрешающие сигналы светофоров таким образом, чтобы число автомобилей, которые пересекают перекресток в магистральном направлении без остановки, было максимальным. Наибольшую известность получили методы PASSER и MAXBAND.

PASSER (Progression Analysis and Signal System Evaluation Routine) является макроскопической детерминированной моделью оптимизации, изначально разработанной Messer C. J. [75]. Метод претерпел ряд модификаций и известен как PASSER II-90. Он использует метод итерационного градиентного поиска для определения наилучшей комбинации последовательностей фаз и длин цикла для максимизации прогрессии вдоль магистрали в обоих направлениях.

PASSER первоначально определяет оптимальные отношения спроса к пропускной способности и использует их для вычисления длительности фаз. Длительность цикла, фаз и величина сдвига варьируются для определения оптимального сигнального плана, который минимизирует полное воздействие на группы прогрессии.

MAXBAND – метод максимизации магистральной прогрессии, разработанный Little, J., Kelson M. D. и Gartner N. H. [69]. MAXBAND выполняет вычисления оптимальной длительности цикла, сдвиг, величину рекомендуемой скорости и последовательности фаз, обеспечивающих максимизацию прогрессии. Вычисление длительности фаз в зависимости от объемов движения и пропускной способности полос движения производится с использованием формулы Вебстера. Кроме того MAXBAND использует оценку времени разезда остаточной очереди, накопленную во время запрещающего сигнала, обеспечивая ее разезд перед прибытием группы автомобилей по магистральному направлению.

Задача оптимизации в MAXBAND сформулирована как смешанная целочисленная задача линейного программирования:

$$b = \bar{b} \rightarrow \max ,$$

при ограничениях

$$\left. \begin{array}{l} w_i + b \leq 1 - r_i \\ \bar{w}_i + \bar{b} \leq 1 - \bar{r}_i \end{array} \right\} i = 1, \dots, n$$

$$\begin{aligned} & (w_i + \bar{w}_i) - (w_{i+1} + \bar{w}_{i+1}) + (t_i + \bar{t}_i) + \Delta_i - \Delta_{i+1} = \\ & = -(1/2)(r_i + \bar{r}_i) + (1/2)(r_{i+1} + \bar{r}_{i+1}) + (\bar{\tau}_i + r_{i+1}) + m_i, \end{aligned}$$

$$i = 1, \dots, n - 1$$

$$m_i = \text{integer}$$



$$b, \bar{b}, w_i, \bar{w}_i \geq 0, i = 1, \dots, n.$$

где  $b$  и  $\bar{b}$  – ширина ленты прогрессии в прямом и обратном направлении;  $r_i$  и  $\bar{r}_i$  – запрещающий сигнал в светофорном цикле  $S_i$ ;  $w_i$  ( $\bar{w}_i$ ) – временной интервал между правой (левой) границей запрещающего сигнала и левой (правой) границей ленты прогрессии;  $t_i$  и  $\bar{t}_i$  – время движения между перекрестками в прямом и обратном направлении;  $\tau_i$  и  $\bar{\tau}_i$  – время разъезда остаточной очереди соответственно в прямом и обратном направлении;  $\Delta_i$  – сдвиг.

На рис. 1.2 приведена диаграмма путь – время для сигнального плана, рассчитанного по методу MAXBAND. Как видим, решение представляет собой симметричные ленты прогрессии в прямом и обратном направлениях.

При наличии доминантного направления с большей интенсивностью движения целевая функция MAXBAND может быть модифицирована следующим образом:

$$b + k\bar{b} \rightarrow \max ;$$

где  $k$  – весовой коэффициент, учитывающий соотношение интенсивностей движения в прямом и обратном направлениях.

Следует отметить, что и MAXBAND и PASSER II используют предположение, что имеется незначительный транспортный поток, входящий в магистральную улицу с второстепенного направления или совершающий левый поворот. Оба указанных метода не учитывают фактические объемы движения и используют однородные ленты времени для прямого и обратного направления. В полевых исследованиях установлено, что они не могут гарантировать оптимальную прогрессию для случая, когда интенсивность движения по длине магистрали существенно меняется.

Дальнейшим развитием MAXBAND стал метод MULTIBAND [40]. В отличие от MAXBAND и PASSER II, MULTIBAND он предполагает возможность использование ленты времени переменной ширины.

Благодаря этому учитывается изменение в объемах движения вдоль магистрали. Целевая функция задачи оптимизации MULTIBAND при этом принимает вид:

$$\frac{1}{n-1} \sum_{i=1}^{n-1} (a_i b_i + \bar{a}_i \bar{b}_i) \rightarrow \max,$$

где  $a_i$  и  $\bar{a}_i$  – весовые коэффициенты для связей в прямом и обратном направлении.

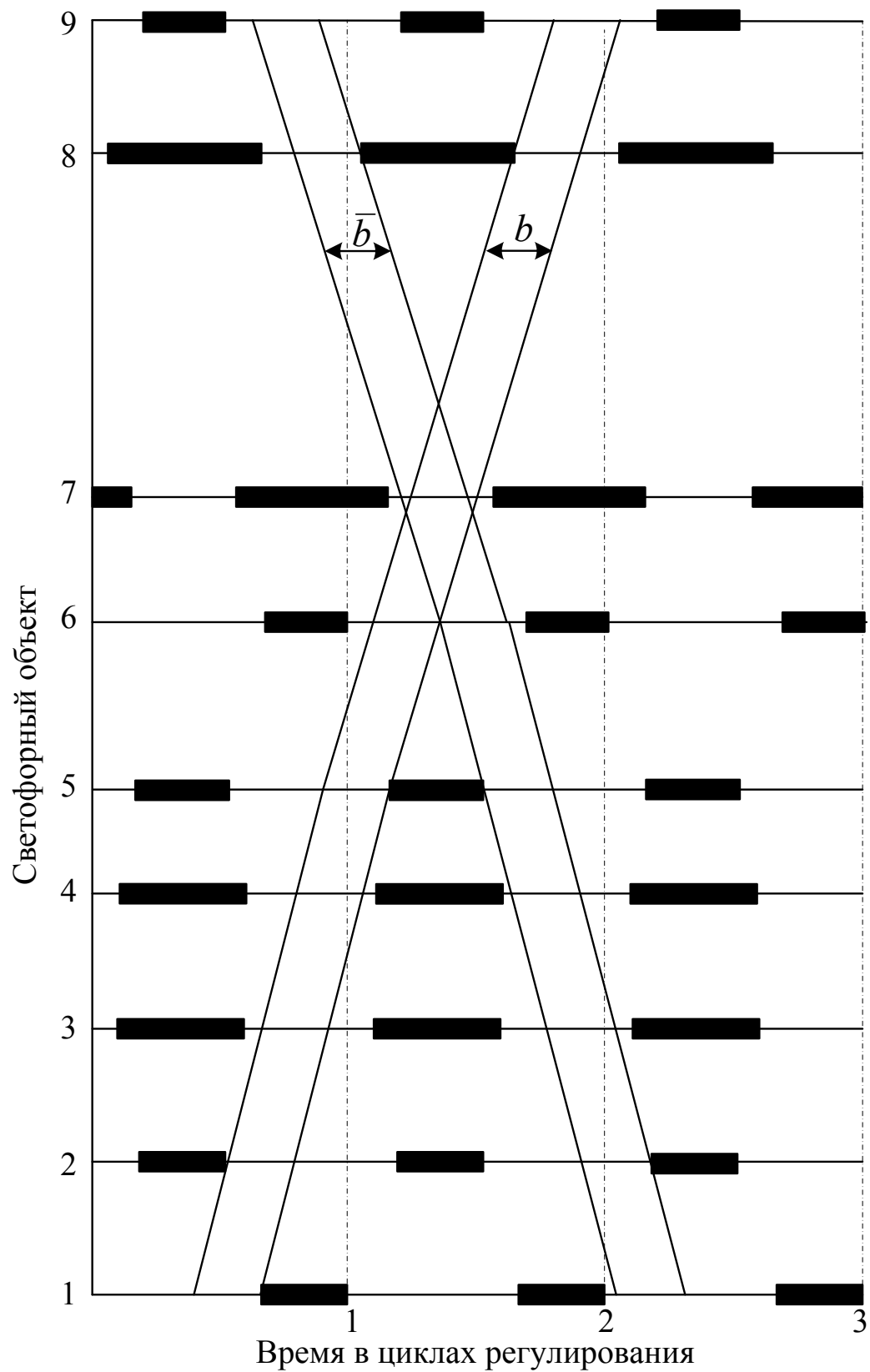


Рис. 1.2 Лента времени прогрессии, полученная методом MAXBAND

Весовые коэффициенты обычно устанавливались следующим образом:

$$a_i = \left( \frac{v_i}{s_i} \right)^p, \quad \bar{a}_i = \left( \frac{\bar{v}_i}{\bar{s}_i} \right)^p;$$

где  $p$  – целое число, как правило используется значения  $p=0, 1, 2, 3, 4$ ;  $v_i$  и  $\bar{v}_i$  – интенсивность движения;  $s_i$  и  $\bar{s}_i$  – поток насыщения.

Весовые коэффициенты могут быть нормализованные следующим образом:

$$\sum_{i=1}^{n-1} a_i = n-1, \quad \sum_{i=1}^{n-1} \bar{a}_i = n-1.$$

Получаемая в результате решения задачи оптимизации MULTIBAND лента прогрессии будет иметь переменную ширину (рис. 1.3).

Рассмотренные методы максимизации прогрессии позволяют учитывать время разъезда остаточной очереди, причем его продолжительность устанавливается пользователем отдельно для каждой связи. Все вышеупомянутые методы максимизации прогрессии используются вне реального времени для подготовки предрассчитанных сигнальных планов на основе статистических данных.

В отличие от ранее упомянутых методов максимизации прогрессии, REALBAND [33] обеспечивает координацию сигналов светофоров на магистральных сетях в реальном времени. Метод позволяет формировать переменные группы прогрессии на основе фактических данных состояния транспортного потока, весах и скорости движения групп автомобилей.

Задача оптимизации стоит в минимизации индекса работы всей сети. REALBAND идентифицирует группы автомобилей, обрабатывая данные от детекторов транспорта, а затем предсказывает их движение по сети.

На уровне управления пересечением оптимальная длительность фаз и их последовательность определена алгоритмом COP, причем местная оптимизация ограничена общесетевыми установками. На уровне управления пересечением может выполняться корректировать время начала и окончания фазы, основываясь на данных о потоке на данной связи.

Оптимизация выполняется на горизонт прогноза путем формирования дерева кандидатов решения. Ветвь, которая приводит к наименьшей стоимости, выбирается как оптимальный сигнальный план. REALBAND требует начального решения относительно длительности фаз, которое используется как начальное условие. Кроме того, метод эффективен только для свободных условий движения и не подходит для насыщенных состояний транспортного потока.

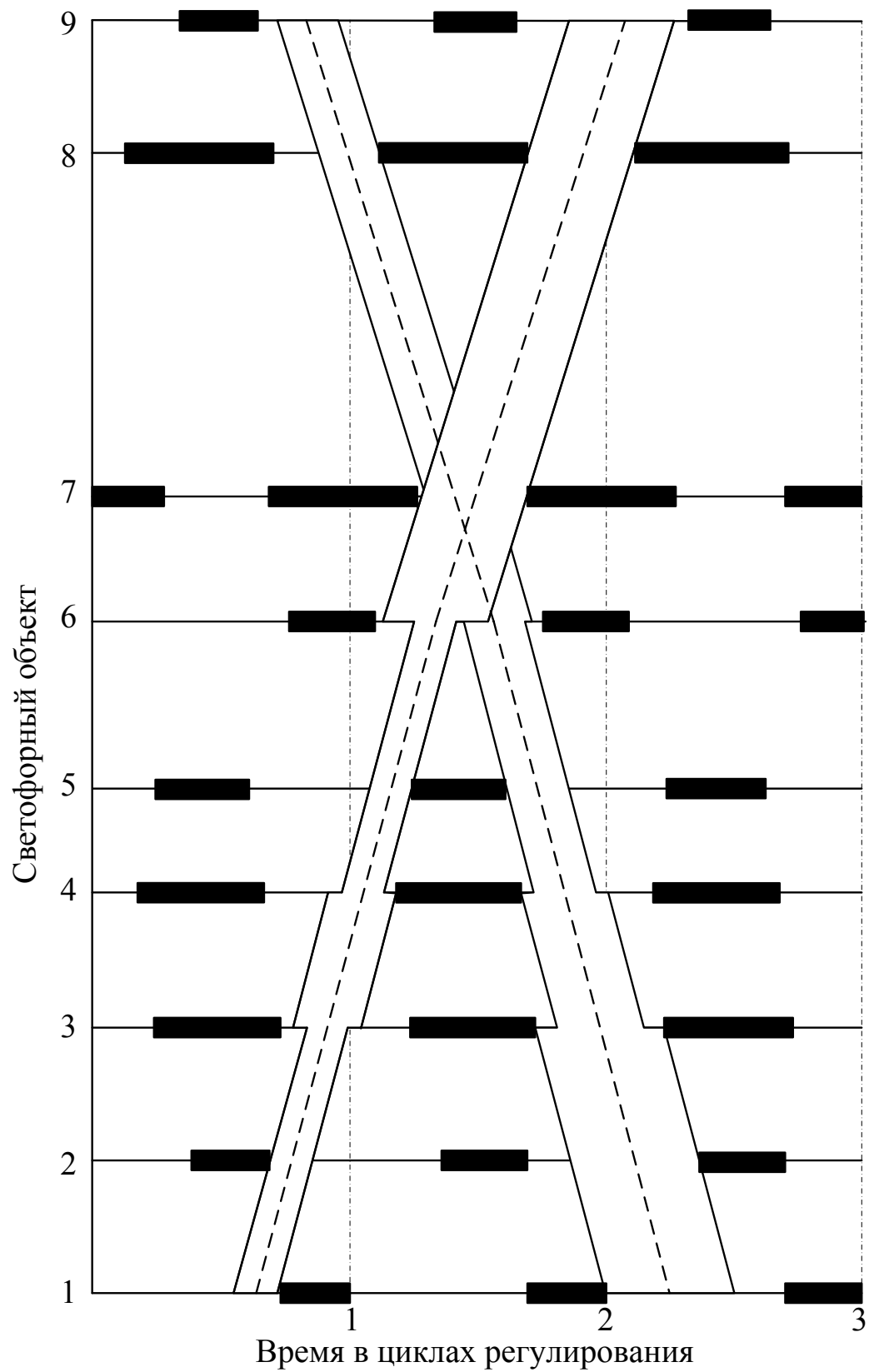


Рис. 1.3 Лента времени прогрессии, полученная методом MULTIBAND

### 1.3.2. Магистральные методы управления сигнальными планами, основанные на индексе потерь

В основе большинства систем управления, которые минимизируют функцию потерь, использован подход, впервые реализованный в программе TRANSYT [49, 108]. В основе метода лежит мезоскопический инструмент моделирования динамики очереди. TRANSYT оптимизирует сигнальный план путем минимизации индекса потерь ( $DI$ ),:

$$DI = \sum_{i=1}^n \left\{ (w_{d_i} d_i + K w_{S_i} S_i) + U_i (w_{d_{i-1}} d_{i-1} + k w_{S_{i-1}} S_{i-1}) + QP \right\} \rightarrow \min ,$$

где  $d_i$  – задержка на связи  $i$  (связей  $n$ ) и на дополнительной, определенном пользователем входной связи  $i-1$ ;  $K$  – установленный пользователем фактор штрафа остановки, выражающий важность остановок относительно задержки;  $S_i$  – остановки на связи  $i$  (и так же для связи  $i-1$ );  $w_{d_i}$ ,  $w_{S_i}$  – определенные для связи весовые факторы для задержки ( $d$ ) и остановки ( $S$ ) на связи  $i$  (и  $i-1$ );  $U_i$  – бинарная переменная, которая принимает значение '1', если для связи была установлена дополнительная надбавка, ноль иначе;  $QP$  – штраф очереди.

Штраф очереди используется для минимизации вероятности блокирования пересечений очередью. При использовании моделирования по связям указанная величина вычисляется следующим образом:

$$QP = Q B_i W_q (q_i - qc_i)^2 ,$$

где  $Q$  – бинарная переменная, установленная пользователем '1', если штраф очереди включен в  $DI$  '0' иначе;  $B_i$  – бинарная переменная '1', если максимум фронта очереди ( $q_i$ ) превышает вместимость связи '0' иначе;  $W_q$  – штраф всей сети относился к излишкам очереди;  $q_i$  – вычисленный максимум фронта очереди на связи  $i$ ;  $qc_i$  – вместимость для связи  $i$ .

Для сверхнасыщенных состояний применяется пошаговое моделирование. В этом случае штраф очереди не может быть вычислен, как указано выше, так как автомобилю не позволяют войти в переполненную связь и  $(q_i - qc_i) = 0$ . Штраф очереди при пошаговом моделировании определяется следующим образом:

$$QP = Q \cdot W_F \cdot F_i ,$$

где  $F_i$  – число шагов, когда связь переполнена;  $W_F$  – общесетевой штраф переполнения очереди.

Величина весовых факторов задержки  $w_{d_i}$ ,  $w_{S_i}$  по умолчанию являются дополнительным расходом топлива.

В TRANSYT используются следующие процедуры оптимизации – движение по градиенту и генетический алгоритм. Сдвиг и продолжительность разрешающих сигналов изменяется в итерационной процедуре, вычисляющей индекс работы и сравнивая его с предыдущим значением для проверки на любое улучшение.

Хотя TRANSYT в состоянии оптимизировать установки сигнала для магистральных сетей и является одной из наиболее широко используемых средств для минимизации задержек, у него имеются и недостатки. Например, TRANSYT предполагает, что средний транспортный спрос остается постоянным.

SCOOT (Split, Cycle, and Offset Optimization Technique) основан на модели движения транспортных средств, используемых в TRANSYT [72]. Метод определяет разбиения фаз, сдвиги и длительность цикла минимизируя продолжительность задержек и число остановок. Система при работе использует данные о транспортном спросе, полученном от детекторов транспорта в форме циклических профилей потока.

Основная философия оптимизации SCOOT состоит в минимизации возмущения в сигнале управления. В системе используется три оптимизатора, которые вычисляют разбиение эффективной длительности цикла между фазами, сдвиг и длительностью цикла, внося изменения с шагом 4 секунды.

Оптимизатор цикла вычисляет степень насыщенности для всех пересечений и определяет критическое, по которому устанавливает длительность цикла в сети. При этом целью оптимизации является обеспечение степени насыщенности на критическом направлении не более 0,90. Также имеется возможность установки половины длительности цикла для отдельных пересечений при условии обеспечения прогрессии. Процедура оптимизации длительности цикла выполняется каждые пять минут.

Оптимизатор разбиения, перед каждым изменением фазы, минимизирует степень насыщения на всех подходах к пересечению, увеличивая или уменьшая длительность разрешающих сигналов на 4 секунды или оставляя их неизменными.

Оптимизатор смещения минимизирует индекс работы, как функцию задержек и остановок в каждом пересечении единожды в цикл регулирования с шагом 4 секунды.

Полевые исследования показали, что эффект от внедрения SCOOT наибольший, когда транспортные потоки близки к насыщенности. Учитывая, что SCOOT не требует долгосрочных прогнозов относительно данных транспортного потока, он не чувствителен к пропуску отдельных транспортных средств детекторами транспорта.

Sydney Coordinated Adaptive Traffic System (SCATS) была разработана Australian Roads and Traffic Authority of New South Wales в 1970-х [70, 71]. Основными задачами системы является:

- уменьшение в целом числа остановок, задержек и времени движения в сети;
- максимизация пропускной способности;
- уменьшение длин очередей, полученных на основе информации от детекторов транспортных средств, расположенных на каждой полосе движения.

У SCATS реализована двухуровневая координированная иерархическая архитектура управления, реализующая на соответствующих уровнях стратегическое и тактическое управление.

На стратегическом уровне управления определяется оптимальная длительность цикла, сдвиг и длительность фаз для области управления на основе информации о состоянии транспортного потока. Оптимальное время цикла определено из условия поддержания степени насыщенности на уровне 0.9 для самой загруженной полосы движения. Оптимальные длительности фаз определены таким образом, чтобы поддерживать равные степени насыщенности на конфликтных направлениях, тем самым минимизируя среднюю задержку. Оптимальный сдвиг находится путем минимизации числа остановок и задержки транспортного потока на магистрали.

На тактическом уровне управления выполняются операции формирования сигнального плана применительно отдельных пересечений в пределах ограничений, наложенных стратегическим уровнем управления. Модификация сигнальных планов включают такие операции как пропуск фазы, завершение фазы, или продление фазы до максимальной величины. При этом основная магистральная фаза не может быть пропущена или сокращена. Любое время, сэкономленное во время цикла, используется последующими фазами или будет добавлено к основной фазе для поддержания равной длительности цикла системе.

Следует отметить, что в SCATS общая длина цикла назначается на все пересечения в магистральном направлении, даже если некоторые из них при этом работают не в оптимальном режиме.

Исследование, проводимое Australian Research Board, установило, что SCATS не показал существенного сокращения времени прохождения в сравнении с предрассчитанными сигнальными планами в TRANSYT. Однако наблюдалось значительное сокращение числа остановок [70], что по всей видимости явилось следствием особенностей работы алгоритма оптимизации на локальном уровне.

Urban Traffic OPTimisation by Integrated Automation (UTOPIA) разработана компанией MIZAR в 1980-х и предназначена для управления свето-

формой сигнализацией и предоставления приоритета в проезде транспорту общего пользования [73]. UTOPIA, в отличие от SCOOT и SCATS, имеет в своей основе распределенную архитектуру управления. Задача управления транспортной сетью разделена на более простые и строго взаимосвязанные подзадачи. Декомпозиция задачи управления выполнена в соответствии со следующим топологическим правилом – область управления выделяется с перекрытием зон, где каждая зона логически сосредоточена на центральном пересечении и включает также соседние (рис. 1.4). Задача оптимального управления определена для каждой зоны и учитывает данные движения и информацию о режимах работы светофоров по всем пересечениям в пределах зоны.

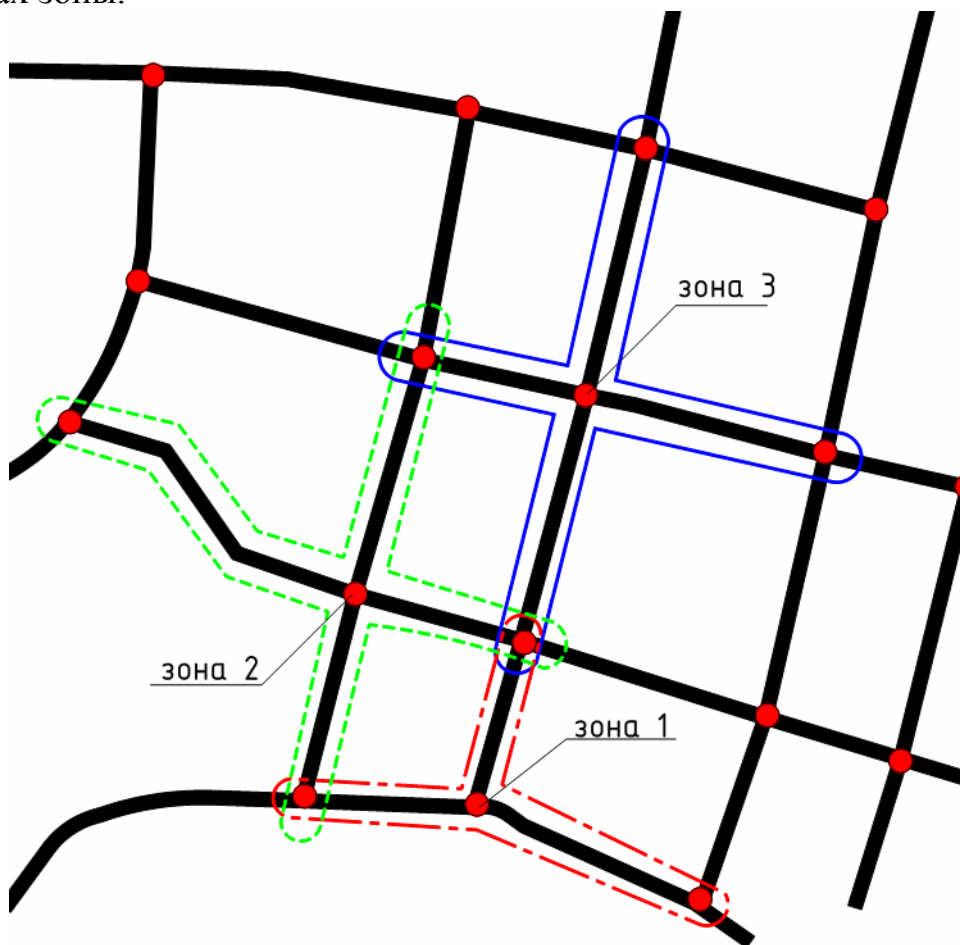


Рис. 1.4 Зоны управления в UTOPIA

Решение зональной задачи управления определяет режим работы светофорного объекта, которое будет исполнено только на центральном пересечении, но вследствие перекрытия между соседними зонами, строго согласовано с управлением во всех соседних пересечениях.

Основная задача UTOPIA состоит в оптимизации в реальном времени сигнальных планов с целью минимизации полной социально-экономической стоимости передвижений в транспортной сети. Затраты на передви-



жения обычно выражаются в продолжительности задержки движения, эмиссии токсичных компонентов отработанных газов автомобилей и времени движения индивидуального и общественного транспорта.

При расчете управляющих воздействий учитываются взвешенные метрики прогнозируемой и измеренной интенсивности движения, а так же прогнозируемое время прибытия общественного транспорта к пересечению. УТОPIA использует метод «катящегося горизонта» в центральных и локальных уровнях для решения следующих задач:

- на центральном уровне на основе макроскопической модели решается задача оптимального управления движением в сети. Для каждого пересечения определяются средние и максимальные продолжительности фаз, сдвигов и веса для всех элементов целевой функции локальной задачи оптимизации. Оптимизация выполняется на горизонт прогноза 30–60 минут (в зависимости от размеров сети) и обновляется каждые 5 мин. Каждое решение актуально не больше 5 минут, после чего заменяется на новое.

- на уровне пересечения, выполняется расчет оптимального сигнального плана путем минимизации целевой функции локального перекрестка на период 120 с и обновляется каждые 3 с.

Целевая функция на уровне пересечения, включает:

- время движения автомобиля на входящих в пересечение транспортных связях;
- число остановок автомобилей на входах в пересечение;
- избыток очереди на входах в пересечение (очереди, которые превышают пороговый уровень в зависимости от емкости входа);
- время сообщения на индивидуальном автотранспорте;
- время сообщения на общественном автотранспорте.

Веса элементов целевой функции определены с учетом времени действия горизонта действующего сигнального плана и фазовых ограничений (например, минимальной и максимальной длительности разрешающих сигналов). УТОPIA позволяет использовать разные веса для различных связей и правил предоставления приоритета.

Зональная итерационная процедура оптимизации позволяет оперативно обнаружить вариации спроса и своевременно реагировать на них. Вследствие тесного информационного взаимодействия зон, возмущение, вызванное последствиями любой вариации спроса в одной зоне, быстро распространяются по всем смежным зонам. Данная схема управления позволяет УТОPIA обеспечивать полностью адаптивное оптимальное управление.

Кроме того, схема управления УТОPIA физически реализует принцип иерархической децентрализованной архитектуры управления. На локальном уровне SPOT выполняет функции управления пересечением, а на центральном уровне динамически определяются весовые коэффициенты

локальных целевых функций и межзональные взаимодействия, используемые при локальной оптимизации.

Сетевая модификация ОПАС так же относится к распределенным системам управления движением на магистралях. Впервые ОПАС использовалась для координированного управления 16 светофорными объектами в Reston, Вирджиния [95]. Данная версия ОПАС известна как Virtual Fixed Cycle (VFC-ОПАС). Алгоритм VFC-ОПАС определяет разбиения фазы, оптимизируя индекс работы, взвешенную комбинацию задержек и остановок при ограничениях на минимальную и максимальную длительность разрешающего сигнала, длительность цикла и сдвига.

VFC-ОПАС определяет актуальный цикл сети по длительности цикла критического пересечения с самой высокой задержкой. VFC-ОПАС увеличивает или уменьшает длину цикла, только если предыдущие два решения совпадают с принимаемым. Система разрешает изменения в действующем цикле только на 1 секунду в каждом шаге управления. В конце каждого цикла выполняется оптимизация смещения на основе данных от смежных пересечений. ОПАС позволяет пропускать фазу при отсутствии спроса.

При сверхнасыщенных состояниях цель управления состоит в максимизации пропускной способности. ОПАС обеспечивает максимальный разрешающий сигнал в насыщенных фазах с ограничениями на длину цикла. Таким образом, приоритет предоставляется той фазе, которая позволяет наибольшему числу автомобиля покинуть пересечение. В то же время, в пересечениях расположенных ранее по ходу движения, длительность разрешающих сигналов устанавливается таким образом, чтобы наименьшее число автомобилей входило в насыщенную связь.

Полевые исследования VFC-ОПАС [43] показали, что его использование привело к увеличению в среднем на 10 % времени прохождения и задержки. Таким образом, технология управления ОПАС оказалась неспособной определить оптимальные длительности цикла, фаз и сдвига.

### 1.3.3. Интеграция методов потерь и прогрессии для магистрального управления

Для преодоления некоторых из недостатков программы TRANSYT, [25] объединил возможности обеспечения прогрессии, предлагаемые моделью MAXBAND и минимизации задержки программой TRANSYT-7F. Метод MAXBAND использовался для поиска начального решения, которое впоследствии использовалось в TRANSYT-7F. Результаты исследования показали, что, в целом указанный подход позволяет улучшить качество рассчитанных сигнальных планы, однако при этом прогрессия не гарантируется.

В последствие была разработана модифицированная версия TRANSYT-7F [26] включающая в систему ограничений обеспечение двухсторонней прогрессии. Аналогично оригинальному варианту TRANSYT-7F, указанный вариант производит поиск оптимальных сдвигов и длительностей фаз градиентным методом. Отличие состоит в том, что после каждого изменения сигнального плана выполняется проверка на обеспечение прогрессии, сгенерированной MAXBAND. Таким образом, длительность фаз и смещение, рассчитанные в TRANSYT ограничены лентами прогрессии MAXBAND. Эксперименты показывают, что метод приводит к существенным сокращениям задержек на главной дороге и задержки сети в целом, обеспечивая прогрессию. Однако при этом ухудшаются условия движения на второстепенных дорогах.

Chang, E. C. [24] модифицировал PASSER II с целью минимизировать магистральную задержку, сохраняя решение для ленты прогрессии. Сдвиг на пересечении определялся в пределах имеющегося резерва времени путем минимизации задержки, тем самым метод обеспечивает двухстороннюю прогрессию при уменьшении задержки. В отличие от а ранее указанного метода, данный подход не дискриминирует второстепенные направления. Необходимое дополнительное время разрешающего сигнала для обеспечения прогрессии на магистральном направлении принимается не за счет второстепенных направлений, а используется, если это возможно, избыточная длительность фаз. Следует отметить, что длительность разрешающих сигналов была фиксированной и не могла изменяться для уменьшения задержки.

#### 1.3.4. Управление в реальном времени путем коррекции предрассчитанных сигнальных планов

В 1970-х – начале 1980-х обозначились проблемы в использовании транспортно-зависимых систем управления, проводящих выбор предварительно рассчитанного сигнального плана. Флуктуации в интенсивности движения компенсировались за счет разработки нескольких сигнальных планов, покрывающих различные сценарии спроса и подсистемы выбора смены этих планов.

Однако исследования показали серьезные недостатки данного метода. К моменту включения сигнального плана, интенсивность движения изменяется и недавно выбранный план больше не отвечает текущим условиям движения. Увеличение частоты смены сигнальных планов может улучшить соответствие условиям движения, но система проводит большую часть времени в переходах между сигнальными планами.

Одним из способов компенсации отклонения контрольных параметров выбранного сигнального плана от текущего состояния транспортного по-

тока является его коррекция. Именно этот способ предполагалось использовать при разработке отечественной АСУДД «Сигнал» [11]. Исходными предпосылками применения общей коррекции сигнального плана служат два основных допущения.

Первое сводится к аппроксимации зависимости скорости движения транспортных потоков от их интенсивности линейной функцией:

$$v_{mn} = v_{св} - K_1 J,$$

где  $v_{св}$  – усредненное значение нормы вектора скорости свободного движения автомобилей при условии их независимого движения;  $K_1$  – коэффициент пропорциональности и приведения размерности;  $J$  – норма вектора интенсивности движения.

Данное допущение приемлемо в левой части основной транспортной диаграммы для небольших диапазонов изменения  $v$  и  $J$ .

Второе допущение предполагает, что интенсивность движения и оптимальный цикл управления также связаны линейной зависимостью:

$$T_{opt} = K_2 J',$$

где  $J'$  – норма вектора интенсивности движения, по которой принимается решение о выборе и оптимизации цикла координированного управления.

В пределах 5–10 %-го изменения предположение о линейности также практически приемлемо. С учетом принятых допущений в контуре общей коррекции производились следующие операции:

1) вычисляется разница между вектором–результатом измерения параметров транспортных потоков, и контрольным вектором, соответствующим введенному в действие плану координации:

$$\vec{R}_i = \vec{X}_i(T_y) - \vec{X}_k(i),$$

2) определяется коэффициент коррекции плана координации как функции от  $\vec{R}_i$ :

$$K_{кор} = F(\vec{R}_i \chi),$$

где  $\chi$  – коэффициент пересчета;

3) все параметры плана координации умножаются на полученный коэффициент, в том числе – цикл длительности тактов переключения светофорной сигнализации, временные сдвиги между включениями фаз управления, т. е.

$$Y_{икор} = Y_i K_{кор}.$$

Развитием данного метода является подход, использованный в стратегии управления светофорными объектами ТУС [34].

Задача управления светофорными объектами сводится к задаче линейно-квадратичного управления, которая допускает решение с обратной связью (feedback):

$$\mathbf{g}(k) = \mathbf{g}(k-1) - \mathbf{L} \cdot \mathbf{x}(k),$$

где  $\mathbf{g}(k)$ ,  $\mathbf{g}(k-1)$  – соответственно рассчитанный и текущий сигнал светофора;  $\mathbf{x}(k)$  – измеренное состояние транспортного потока;  $\mathbf{L}$  – матрица усиления.

TUC разработана в рамках проекта TABASCO и в настоящее время реализован на части городских транспортных сетей Глазго (Великобритания) и Chania (Греция). Полевые исследования показали приемлемые результаты, особенно в условиях насыщенных транспортных потоков.

#### 1.4. Управление в условиях заторов

Принципы реализации управления в условиях насыщенного движения рассмотрим на примере системы SCOOT, реализующей централизованное адаптивное управление транспортными потоками. Централизация системы позволяет использовать общесетевые стратегии управления: пропуск пиковой загрузки; противозаторовое управление и нормирование объемов движения на подходах к зонам заторов; централизованный приоритет транспорта общего пользования [60, 72]. Управление в условиях сверхнасыщенности движения в SCOOT основывается на следующих методах:

- смена значимости фактора перегрузки транспортных связей  $QP$  (при перегрузке транспортной связи производится смена ее коэффициентов значимости);
- нормирование и смена весовых коэффициентов  $w_{di}$  и  $w_{si}$  целевой функции.

Фактор значимости перегрузки определяется для каждой связи и используется для воздействия на оптимизацию разбиения в пользу связи, на которой обнаружен затор. Смещение при перегрузке связи является фиксированной величиной, определяемой инженером экспертно, и используется только в условиях затора. Нормирование въезда позволяет уменьшить время разрешающего сигнала на подходах к переполненному участку и тем самым ликвидировать затор.

Несколько иной подход к задаче предотвращения и ликвидации транспортных заторов использован в системе MOTION [63]. Главное отличие MOTION от традиционных АСУДД (таких как SCOOT, UTOPIA, SCATS) заключается в том, что оперативное посекундное управление сигнальными группами отдельного перекрестка отделено от уровня адаптивного управления сетью. Дорожные контроллеры используют

локальное адаптивное управление («actuated traffic control»), а на сетевом уровне решаются задачи общесетевого характера и каждые 5–15 минут производится обновление настроек дорожных контроллеров.

Управление в случае возникновения дорожно-транспортных происшествий или заторов обеспечивается специальным модулем системы – СИМ. Он использует предварительно подготовленные сценарии, качество которых зависит от опыта транспортных инженеров. Модуль СИМ обеспечивает в общей сложности двенадцать стратегий управления. Рассмотрим подробнее некоторые из них.

Если транспортный спрос превышает пропускную способность управляемой области транспортной сети, то используется стратегия нормирования транспортного предложения. Для этого на основании транспортно-технического анализа выбираются транспортные узлы, на которых производится сдерживание входящего в область транспортного потока. Выбор узлов производится по следующим критериям:

- узел в направлении большого значения транспортного спроса на входах в область;
- узел, задержка в котором не будет иметь негативных экологических последствий для данной области;
- узел, увеличение очереди транспортных средств в котором не скажется отрицательно на транспортном потоке в соседней области.

В случае внедрения данной стратегии управления, в выбранных узлах устанавливаются укороченные разрешающие сигналы:

$$tGi = MGG + (1 - SI / 100) \cdot (tGn - MGG),$$

где  $tGi$  – длительность сокращенного зеленого сигнала;  $MGG$  – минимальная длительность сокращенного зеленого сигнала;  $SI$  – весовой коэффициент;  $tGn$  – нормальная длительность зеленого сигнала.

Если был установлен факт возникновения транспортного затора, то увеличивается весовой коэффициент  $SI$  и соответственно сокращается длительность зеленого сигнала. При улучшении транспортных условий настолько, что логическая функция, идентифицирующая факт образования затора, не выполняется, то коэффициент  $SI$  автоматически уменьшается вдвое и новое значение длительности зеленого сигнала вводится в расчеты MOTION. Ситуация повторно оценивается через 15 минут. Минимальное значение индекса  $SI$  может быть равным 20, при меньших значениях устанавливается нормальная длительность разрешающего сигнала.

Стратегия предотвращения блокировки поперечных направлений используется для того, чтобы воспрепятствовать возникновению длинных очередей на перегонах, способных заблокировать соседние узлы. Управление заключается в согласовании длительности разрешающих сигналов и

временных сдвигов двух соседских узлов так, чтобы не превышалась пропускная способность перегона между ними.

Как видим, основной задачей управления насыщенными транспортными потоками является балансировка транспортного спроса и предложения на заранее определенных транспортным инженером связях (перегонах городских улиц). Балансировка достигается путем модификации весовых коэффициентов целевой функции управления или определяющих соотношений, причем значение корректирующих коэффициентов устанавливается транспортным инженером на основе собственного опыта.

## 2. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ УПРАВЛЕНИЯ НАСЫЩЕННЫМИ ТРАНСПОРТНЫМИ ПОТОКАМИ

### 2.1. Математическая модель задержки и длины очереди на изолированном перекрестке

Продолжительность задержки транспортных средств получила широкое применение в качестве критерия оптимизации управления, как на отдельном пересечении, так и сети. Существует два основных подхода к определению задержки транспортных средств. Первый подход основан на макроскопической модели описания транспортного потока и определяет задержку как потери времени на вынужденные остановки транспортных средств перед перекрестками, железнодорожными переездами, при заторах на перегонах, а также из-за снижения скорости транспортного потока по сравнению со средней скоростью свободного движения на данном участке дороги:

$$d = \sum_i l_i \left( \frac{1}{v_i^{free}} - \frac{1}{v_i} \right); \quad (2.1)$$

где  $v_i^{free}$  – скорость свободного движения на  $i$ -м участке;  $v_i$  – наблюдаемая скорость движения на  $i$ -го участке;  $l_i$  – длина  $i$ -го участка.

Второй подход определяет задержку в терминах системы массового обслуживания как среднее время ожидания автомобиля в очереди. Под длиной очереди при этом понимается количество транспортных средств, ожидающих обслуживания перед перекрестком.

На практике первый подход, как правило, используется для оценки качества проектных решений по организации движения, в то время как второй подход к определению задержки получил распространение при решении задач оптимального управления светофорными объектами.

По характеру описания свойств транспортного потока принято различать детерминированные и стохастические модели задержки. В практике управления дорожным движением получили распространение модели, сочетающие в себе как детерминированную, так и стохастическую составляющие задержки.

Детерминированная составляющая задержки определяется в условиях стационарности движения:

- в начале разрешающего сигнала очередь равна нулю;
- транспортные средства прибывают одинаковыми группами с интенсивностью движения  $q$  за цикл;



– разъезд осуществляется одинаковыми группами с интенсивностью равной потоку насыщения  $S$  при наличии очереди, и с интенсивностью равной интенсивности прибытия при ее разъезде;

– спрос транспортных средств не превышает пропускной способности элемента перекрестка, которую определяют как отношение потока насыщения  $S$  к отношению разрешающего сигнала ко времени цикла ( $g/c$ ).

Упрощенно процесс формирования очереди транспортных средств можно представить в форме диаграммы (рис.2.1).

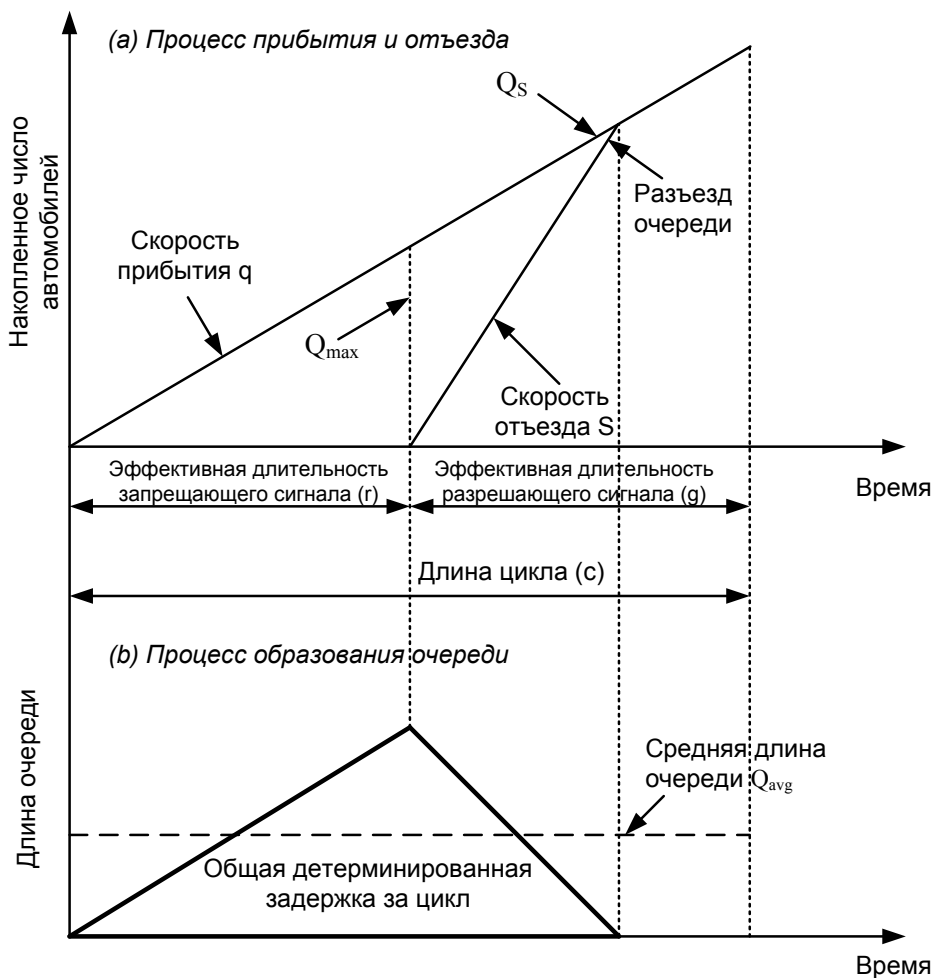


Рис. 2.1. Детерминированная составляющая модели образования задержек

На запрещающий сигнал  $r$  формируется очередь транспортных средств с интенсивностью  $q(t)$ . При включении разрешающего сигнала  $g$  начинается разъезд очереди с интенсивностью  $q-S$ , при этом количество транспортных средств в очереди принимает максимальное значение  $Q_{\max}$ . Область под линией профиля очереди (рис. 2.2) представляет собой детерминированную задержку за цикл.

Впервые модель оценки задержки при жестком светофорном регулировании была предложена Beckmann M. [20] с допущением о биноми-

нальном процессе прибытия транспортных средств и детерминированном характере процесса обслуживания:

$$d = \frac{c - g}{c \cdot (1 - q / S)} \cdot \left( \frac{Q_0}{q} + \frac{c - g + 1}{2} \right), \quad (2.2)$$

где  $c$  – длительность цикла;  $g$  – эффективная длительность зеленого сигнала;  $q$  – интенсивность прибытия транспортных средств;  $S$  – интенсивность разъезда из очереди;  $Q_0$  – величина остаточной очереди.

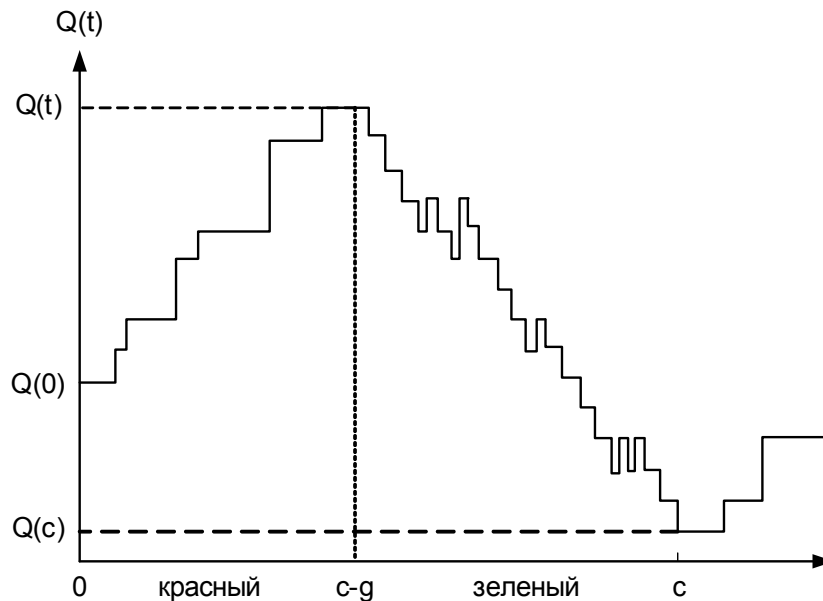


Рис. 2.2. Процесс образования очереди в течение одного цикла (по исследованиям D.McNeil 1986)

Позднее Little в [68] исследовал величину задержки на регулируемом пересечении при пуассоновском распределении транспортного потока, Darroch J.N. [30] процесс прибытия представлял обобщенным пуассоновским процессом:

$$d = \frac{(c - g)}{2 \cdot c \cdot (1 - q / S)} \cdot \left[ (c - g) + \frac{2 \cdot Q_0}{q} + \frac{1}{S} \cdot \left( 1 + \frac{I}{1 - q / S} \right) \right], \quad (2.3)$$

где  $var(.)$  – среднеквадратическое отклонение ( $.$ ),  $I = \frac{var(A)}{q \cdot h}$ ;  $q$  – интенсивность прибытия;  $h$  – длина интервала во времени;  $A$  – число прибытий в течение интервала  $A = q \cdot h$ .

McNeil D.R. [74] исследовал величину задержки в предположении об общности процесса прибытия транспортных средств и постоянстве вре-

мени разъезда. Общая задержка за цикл представлялась как сумма двух составляющих:

$$W = W_1 + W_2, \quad (2.4)$$

где  $W_1$  – общая задержка в фазе красного сигнала;  $W_2$  – общая задержка в фазе зеленого сигнала.

$$W_1 = \int_0^{c-g} (Q(0) + A(t))dt, \quad (2.5)$$

$$W_2 = \int_{c-g}^c (Q(t))dt, \quad (2.6)$$

где  $Q(t)$  – длина очереди в момент времени  $t$ ;  $A(t)$  – кумулятивное число прибытий в момент времени  $t$ .

В работе [56] Herman R. и Weiss G. рассматривая случай сложного пуассоновского процесса прибытия и общего процесса обслуживания получили следующую модель:

$$d = \frac{(c-g)}{2 \cdot c \cdot (1-q/S)} \left[ (c-g) + \frac{2}{q} \cdot \left[ 1 + \frac{(1-q/S) \cdot (1-B^2)}{2 \cdot S} \right] \cdot Q_0 + \frac{1}{S} \cdot \left( 1 + \frac{I+B^2 \cdot q/S}{1-q/S} \right) \right]. \quad (2.7)$$

Исследование уравнения показало, что в случае отсутствия начальной задержки ( $Q_0 = 0$ ) и случайности транспортного процесса ( $I=0$ ), полученная задержка является постоянной составляющей задержки. Этот компонент может быть получен из простой модели с постоянным прибытием за цикл и процессом обслуживания, описанным выше. В более общем случае уравнение (2.7) требует дополнительных данных о размере очереди и величине остаточной очереди – основных ограничениях использования полученной формулы на практике.

Необходимость учета остаточной очереди послужила стимулом в проведении исследований по получению закрытой формы уравнения остаточной очереди [48, 86, 87, 30], однако оно так и не было получено.

Необходимость оценки остаточной очереди и наличие строгих ограничений, накладываемых предположением о законе распределения прибытия транспортных средств, ограничивает применимость «точных» моделей на практике. Это положило начало исследованиям по получению приближенных формул (моделей с элементами аппроксимации) для оценки задержек с более простыми и ограничениями, которые способны учитывать разнообразие реальных условий.

Первая, широко известная приближенная формула, была получена Webster F.V. [109, 110] при сочетании теоретического подхода и численного моделирования:

$$d = \frac{c \cdot (1 - g/c)^2}{2 \cdot [1 - (g/c) \cdot x]} + \frac{x^2}{2 \cdot q \cdot (1 - x)} - 0,65 \cdot \left( \frac{c}{q^2} \right) x, \quad (2.8)$$

где  $d$  – средняя задержка за цикл, с;  $c$  – длина цикла регулирования, с;  $g$  – эффективная длительность зеленого сигнала, с;  $x$  – степень насыщения (отношение интенсивности движения к пропускной способности);  $q$  – интенсивность прибытия, авт/ч.

Первая составляющая уравнения (2.8) представляет задержку, при условии стационарности прибытия транспортного потока, а вторая составляющая определяет случайную составляющую процесса. Последняя известна как «случайная задержка», в предположении о пуассоновском процессе прибытия и постоянной интенсивности разезда транспортных средств, которая соответствует пропускной способности. Третья составляющая представляет собой корректирующее слагаемое.

Дальнейшая аппроксимация уравнения (2.7) была направлена на упрощение за счет сокращения третьего и четвертого члена уравнения, которые, как правило, имеют намного меньший порядок, чем первые два. Этот подход использован в работе Miller A.J. [79] при выводе аппроксимирующей формулы:

$$d = \frac{(1 - g/c)}{2(1 - q/s)} \left[ c(1 - g/c) + \frac{2Q_0}{q} \right]. \quad (2.9)$$

Формула получена путем исключения второго и третьего члена уравнения (2.7). Miller также получил выражение для насыщенной очереди при пуассоновском прибытии и времени обслуживания равному продолжительности разрешающего сигнала:

$$Q_0 = \frac{\exp[-1.33 \cdot \sqrt{S \cdot g} \cdot (1 - x) / x]}{2 \cdot (1 - x)}, \quad (2.10)$$

В последствие Newell модифицировал формулу (2.9) путем добавления дополнительного корректирующего члена, позволившего повысить точность определения задержки при средней загрузке:

$$d = \frac{c \cdot (1 - g/c)^2}{2 \cdot (1 - q/S)} + \frac{Q_0}{q} + \frac{(1 - g/c) \cdot I}{2 \cdot S \cdot (1 - q/S)^2}. \quad (2.11)$$

Дальнейшие попытки повышения точности моделей не привели к существенным результатам.

Приведенные выше модели задержки требуют достижения стохастического равновесия. Для его достижения требуется бесконечное время при стабильных условиях движения (прибытие, обслуживание, регулирование). При низком отношении интенсивности потока к пропускной способности это достигается в разумных пределах времени, поэтому такие модели являются приемлемой аппроксимацией реальных процессов.

При достижении интенсивности движения пропускной способности, время, необходимое для достижения такого равновесия, обычно превышает период, в течение которого спрос является стабильным. Кроме того, во многих случаях интенсивность движения превышает пропускную способность, при этом нарушаются ограничения моделей.

Попытки преодолеть указанные ограничения были предприняты с использованием двух различных подходов. Наиболее простой подход предполагает функции интенсивности прибытия и разъезда рассматривать как детерминированный процесс. В общем случае модель определяет кумулятивное число прибытий  $A(t)$ :

$$A(t) = \int_0^t q(\tau) d\tau, \quad (2.12)$$

и число обслуживаний  $D(t)$  при условии длины очереди в течение периода  $[0, t]$ :

$$D(t) = \int_0^t S(\tau) d\tau. \quad (2.13)$$

Текущее количество транспортных средств в системе равно:

$$Q(t) = Q_o(t) - A(t) - D(t) \quad (2.14)$$

и средняя задержка транспортных средств, в течение периода  $[0, T]$ :

$$d = \frac{1}{A(t)} \cdot \int_0^T Q(t) dt. \quad (2.15)$$

Можно использовать данную модель для регулируемого пересечения с известным режимом управления при замене  $C(t)$  на  $S(t)$  в уравнении (2.14):

$C(t) = 0$ , если горит красный сигнал;

$C(t) = S(t)$ , если горит зеленый сигнал и  $Q_o(t) > 0$ ;

$C(t) = q(t)$ , если горит зеленый сигнал и  $Q(t) = 0$ .

Детерминированные модели, подобные уравнению (2.12) применимы только при  $x \ll 1$  или  $x \gg 1$ . В иных условиях имеется тенденция недооценки величины очередей и задержек, так как не учитывают дополнительные очереди, которые возникают из-за стохастических колебаний  $q$  и  $C$ .

Второй подход предусматривает моделирование транспортного потока на перекрестке, предполагая стабильность процессов прибытия и обслуживания но, не предполагая обязательного достижения стохастического равновесия. При этом проводится оценка средней задержки, и длины очереди для рассматриваемого периода времени. Данный подход предусматривает аппроксимацию интенсивности прибытия некоторой функцией времени (ступенчатая, параболическая или треугольная функция) и определение соответствующей значения задержек.

В последнее время популярность приобрел метод трансформации координат. Хотя метод и использует стабилизированную кривую очереди, полученную по классической теории, в результате получается временно-зависимая формула задержки.

Основой развития данного метода послужили следующие наблюдения:

– при низкой степени насыщения ( $x \ll 1$ ) задержка всегда равна такому значению, которое может быть при постоянной плотности в течение всего времени;

– при высокой степени насыщения ( $x \gg 1$ ) задержка может быть описана с разумной степенью точности следующим детерминированным выражением:

$$d = d_1 + \frac{T}{2} \cdot (x - 1), \quad (2.16)$$

где  $d_1$  – задержка, наблюдаемая при низкой плотности (постоянная задержка);  $T$  – анализируемый период, в течение которого интенсивность движения стабильна;

– модели задержки при стабильных состояниях асимптотически приближаются к оси  $y (y = q/S)$ , т.е. дает бесконечную задержку при насыщенном состоянии ( $x = q/C = 1$ ). Метод трансформации координат передвигает первоначальную кривую метода стабильных состояний до асимптотического приближения к детерминированной линии перенасыщения (рис. 2.3) – второй составляющей уравнения (2.16).

Горизонтальное расстояние между предложенной кривой задержки и ее асимптотой равно расстоянию между кривой «стабильной» модели и вертикальной линией  $x=1$ .

Имеется два ограничения относительно применения формулы:

- 1) первоначальная длина очереди равна нулю (в начале периода  $[0, T]$ );
- 2) интенсивность движения постоянна в течение периода  $[0, T]$ .

В результате моделирования установлена адекватность полученного метода. Таким образом, такая техника является полезной в практике. Принципиальный недостаток заключается в добавление ограничений, которые недостаточно теоретически обоснованы. Указанные выше трудности были

преодолены другими авторами путем аппроксимации действительной интенсивности движения ступенчатой функцией.

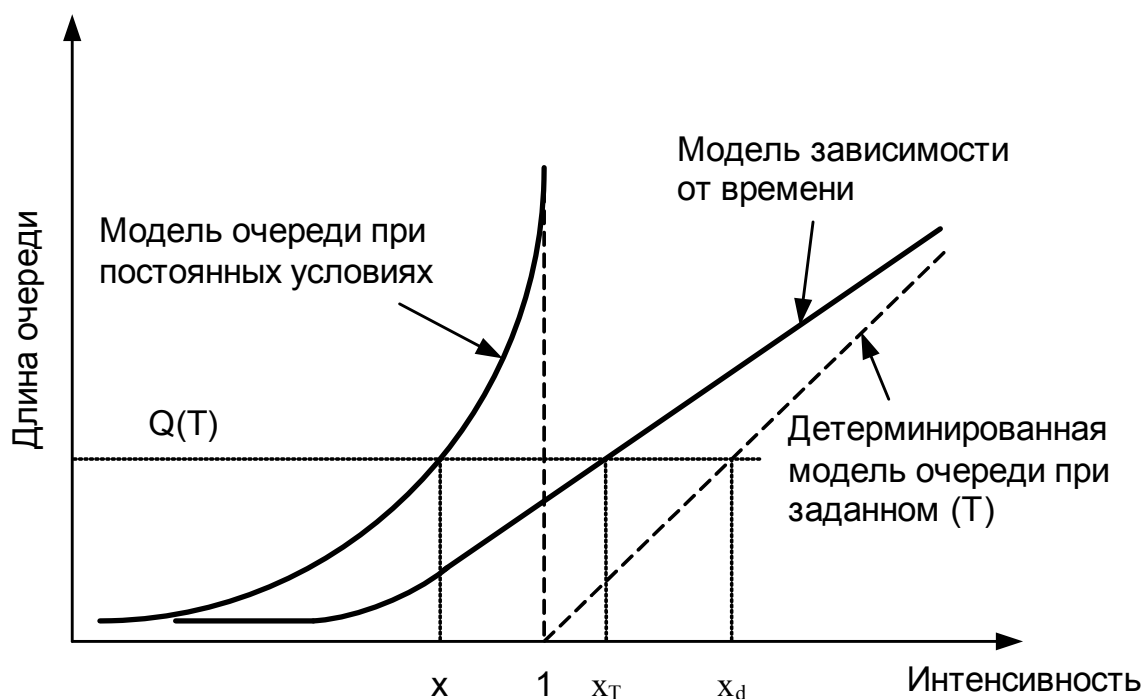


Рис. 2.3. Метод трансформации координат

Для выполнения практических расчётов задержки транспортных средств как в условиях не насыщенного движения ( $v/c < 1,0$ ), так и в условиях перенасыщения ( $v/c > 1,0$ ) хорошо себя зарекомендовала модель Highway Capacity Manual (HCM) [57, 58]. Модель задержки была получена с использованием, как эмпирических данных, так и теоретических соотношений. Уравнение включает три составляющих задержки:

$$d = d_1 PF + d_2 + d_3, \quad (2.17)$$

где  $d_1$  — постоянная составляющая задержки (uniform delay), с/авт;  $PF$  — фактор прогрессии;  $d_2$  — случайная задержка (incremental delay), с/авт.;  $d_3$  — задержка перенасыщения (initial queue delay).

Постоянная составляющая задержки вычисляется как:

$$d_1 = 0,5 \cdot C \frac{(1 - g/C)^2}{1 - (\min(1, X) \cdot g/C)}, \quad (2.18)$$

где  $C$  — длительность цикла, с;  $g/C$  — отношение длительности разрешающего сигнала к циклу регулирования для группы полос движения;  $X$  — степень насыщенности группы полос движения ( $v/c$ );  $c$  — пропускная

способность группы полос движения  $c = s \cdot g / C$ , авт/ч;  $s$  – поток насыщения для группы полос движения, авт/ч;  $v$  – транспортный спрос.

Случайная задержка определяется с учетом не стационарности процесса формирования очереди:

$$d_2 = 900T \left\{ (X - 1) + \sqrt{(X - 1)^2 + \frac{8kIX}{cT}} \right\}, \quad (2.19)$$

где  $T$  – продолжительность периода анализа в часах;  $k$  – коэффициент корректировки, учитывающий способ управления (адаптивное или предрасчитанное);  $I$  – коэффициент корректировки, учитывающий влияние соседних перекрестков.

Для предрасчитанного управления величина  $k = 0,50$ . Для транспортно-зависимых контроллеров величина  $k$  вычисляется как функция степени насыщенности и  $k_{\min}$ , используя следующую формулу:

$$k = (1 - 2 \cdot k_{\min})(X - 0,5) + k_{\min}, \quad (2.20)$$

где  $X$  – величина, меньшая 0,5 устанавливается равной 0,5;  $k_{\min}$  – временной интервал, характеризующий наличие разрыва в транспортном потоке.

Результирующее влияние величины  $k$  должно уменьшить задержку при транспортно-зависимом управлении, в условиях низких степеней насыщенности.

Величина  $I$ -фактора равна 1,0 для изолированных регулируемых пересечений. Для координированных пересечений, величина  $I$ -фактора является функцией от степени насыщенности расположенных по ходу движения направлений ( $X_u$ ):

$$I = 1,0 - 0,91X_u^{2,68}. \quad (2.21)$$

Когда степень насыщенности восходящего потока больше чем 1,0, устанавливается равным 1,0, что в свою очередь ограничивает  $I$ -фактор минимальной величиной 0,09. Результирующее влияние  $I$ -фактора должно уменьшить задержку потоков, которые насыщены по ходу движения. Другими словами, движение в ненасыщенных условиях, имеет более широкое разнообразие профилей транспортного потока, таким образом, увеличивая вероятности неэффективного использования фазы на расположенном вниз по ходу движения пересечении.

Задержка перенасыщения определяется величиной остаточной очереди на анализируемом периоде по формуле

$$d_3 = \frac{1800 \cdot Q_b(1+u) \cdot t}{cT}; \quad (2.22)$$



$$t = \begin{cases} 0 \text{ при } Q_b = 0 \\ \min\left(T, \frac{Q_b}{c(1 - \min(1, X))}\right); \end{cases}$$

$$u = \begin{cases} 0 \text{ при } t < T \\ 1 - \frac{cT(1 - \min(1, X))}{Q_b}; \end{cases}$$

где  $Q_b$  – величина остаточной очереди, авт;  $t$  – продолжительность неудовлетворения спроса, ч.

## 2.2. Математическая модель сетевого движения транспортных потоков

Периодичность работы светофорного объекта формирует на выходе группы автомобилей, движущиеся по исходящим транспортным связям. Было замечено, что вследствие дисперсии скоростей транспортных средств, происходит изменение параметров группы вплоть до ее полного распада. Это явление получило название распада или дисперсия группы. Расеу G.M. [88] получил распределение времени прохождения автомобилей  $f(\cdot)$  по транспортной связи исходя из нормального распределения скоростей и неограниченной возможности обгона:

$$f(\tau) = \frac{D}{\tau^2 \sigma \sqrt{2\pi}} \exp\left[-\frac{\left(\frac{D}{\tau} - \frac{D}{\bar{\tau}}\right)^2}{2\sigma^2}\right], \quad (2.23)$$

где  $D$  – расстояние от стоп-линии до точки наблюдения;  $\tau$  – индивидуальное время прохождения автомобиля расстояния  $D$ ;  $\bar{\tau}$  – среднее время прохождения расстояния  $D$ ;  $\sigma$  – дисперсия скорости.

Распределение времени прохождения используется для преобразования профиля транспортного потока при движении по транспортной связи на дистанцию  $D$ :

$$q_2(t_2)dt_2 = \int_{t_1} q_1(t_1) f(t_2 - t_1) dt_1 dt_2, \quad (2.24)$$

где  $q_2(t_2)dt_2$  – общее количество автомобилей, проходящих через некоторую точку, расположенную на некотором расстоянии от стоп-линии в интервале  $(t, t+dt)$ ;  $d_1(t_1)dt_1$  – общее количество автомобилей, проходящих

через стоп-линию;  $f(t_2 - t_1)$  – плотность вероятности времени движения  $(t_2 - t_1)$  согласно формуле (2.24).

Дискретный вариант формулы (2.24):

$$q_2(j) = \sum_i q_1(i) g(j - i), \quad (2.25)$$

где  $i$  и  $j$  – дискретные интервалы гистограммы прибытия.

Для прикладных расчетов в работе [6] предложена аппроксимация зависимости  $q(t)$  кусочно-постоянной функцией для расчета сдвига включения разрешающих сигналов в сети со светофорным регулированием.

Прибывающий к перекрестку поток характеризуют изменения во времени (при постоянных средних характеристиках) интенсивности  $q(t)$ . На выходе перекрестка поток определяется временной длиной  $p$  группы автомобилей и средней интенсивностью  $q'$  интенсивностью потока в течение эффективного зеленого сигнала (рис. 2.4).

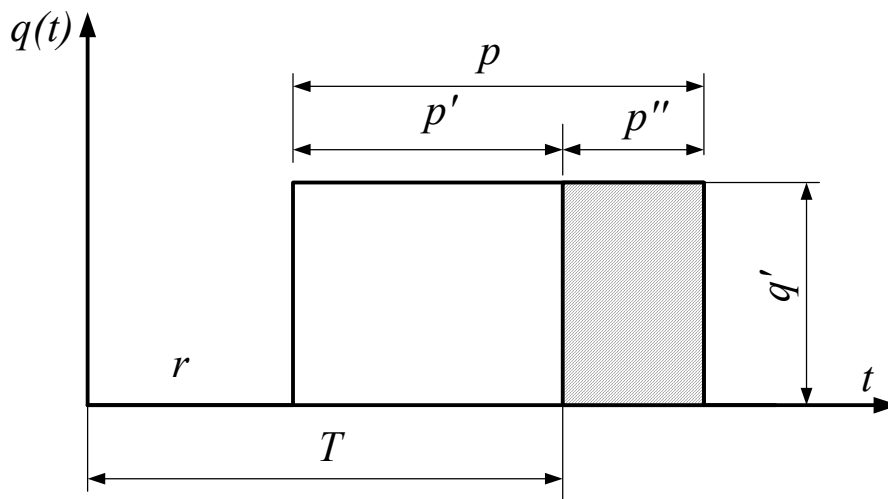


Рис. 2.4. Аппроксимация группового движения кусочно-постоянной функцией

Прибывающий к перекрестку поток характеризуют изменением во времени (при постоянных средних характеристиках) интенсивности  $q(t)$ . На выходе с перекрестка поток определяется временной длиной  $p$  группы автомобилей и средней интенсивностью потока  $q'$  в течение эффективного зеленого сигнала.

Временная длина определяется следующим образом:

$$p = p' + pp'', \quad (2.26)$$

где  $p'$  – длительность эффективного зеленого сигнала.

Величина  $p''$  обусловлена поворотными потоками на перегоне и рассчитывается по формуле

$$p'' = \frac{q_{\text{пов}} T}{q'}, \quad (2.27)$$

где  $q_{\text{пов}}$  – суммарная интенсивность поворотных потоков.

Причем величина  $q'$  связана со средней интенсивностью  $q$ , соотношением

$$q' = \frac{qT}{t_{\text{эф}}}; \quad (2.28)$$

где  $t_{\text{эф}}$  – эффективная длительность зеленого сигнала.

По мере удаления от перекрестка характеристики (модельные) потока изменяются. Если  $t$  – среднее время движения (от перекрестка) до рассматриваемого сечения перегона, то в этом сечении  $p(t)$  и  $q(t)$  могут быть определены по формулам:

$$p(t) = p(0)e^{0,008t}, \quad (2.29)$$

$$q(t) = q(0) \frac{p(0)}{p(t)}, \quad (2.30)$$

$$p(0) = p' + p'', \quad (2.31)$$

$$q(0) = q'. \quad (2.32)$$

Причем если оказывается, что  $p(t) > T$  то в силу физических соображений принимается  $p(t) = T$ .

Длительность задержки  $z$  и число остановок  $N$  транспортных средств (за цикл) могут быть определены следующим образом:

$$z = q \cdot r \cdot \alpha + \frac{q \cdot \alpha^2}{2} \left( \frac{q}{C} - 1 \right) + q \cdot \gamma \cdot \beta + q \cdot \left( \frac{q}{C} - 1 \right) \cdot \frac{\beta^2}{2}; \quad (2.33)$$

где

$$\alpha = \min \left( x, \frac{C \cdot r}{C - q} \right); \beta = \min \left( p, \frac{C \cdot \gamma}{C - q} \right);$$

$$\gamma = \max \left( 0, r + \frac{q \cdot x}{C} - \tau \right); x = \max (\tau + p - T, 0);$$

$$N = q(\alpha + \beta).$$

Величина момента  $\tau$  появления лидера в цикле рассчитывается по формуле

$$\tau = \delta_1 - \delta_2 + 0,8\bar{t} + r - (n-1) \cdot T, \quad (2.34)$$

где  $\bar{t}$  – время движения по (перегону).

Причем величина  $n$  выбирается таким образом, чтобы имело место неравенство:

$$0 \leq \tau < T.$$

Более точно (с учетом пуассоновского характера потока) величина задержки (за цикл) может быть оценена по формуле:

$$z = q \cdot r \cdot \alpha + \frac{q \cdot \alpha^2}{2} \left( \frac{q}{C} - 1 \right) + q \cdot \gamma \cdot \beta + q \left( \frac{q}{C} - 1 \right) \frac{\beta^2}{2} + \frac{0,4 \cdot x_1^2 \cdot T}{1 - x_1}; \quad (2.35)$$

где

$$x_1 = \frac{q \cdot T}{C \cdot (T - r)}.$$

Средняя длительность задержки  $z_1$  и среднее число остановок  $N_1$  транспортного средства определяются, как

$$z_1 = \frac{z}{q \cdot T}, \quad (2.36)$$

$$N_1 = \frac{N}{q \cdot T}. \quad (2.37)$$

Полученные расчетные соотношения позволяют найти задержку при остановке транспортных средств. Величина полной задержки  $z'$  (с учетом снижения скорости в зоне перекресток) находится приближенно по формуле

$$z' = 1,04 \cdot z + 6,43. \quad (2.38)$$

К недостаткам приведенной модели сетевого движения следует отнести фиксированную степень дисперсии группы автомобилей и предположение о том, что все прибывающие к перекрестку автомобили останавливаются на стоп-линии.

Несколько иной подход использован в модели, использованной при разработке программы TRANSYT. Моделирование движения производится на основе преобразования профилей транспортных потоков при их движении по транспортным связям и через перекрестки. Различают три типа профилей – IN-профиль, GO-профиль и OUT-профиль.

ГО-профиль (профиль потока насыщения) – это норма потока, который проезжал бы через стоп-линию при насыщенном разрешающем сигнале. Фактически ГО-профиль представляет собой профиль потока насыщения на регулируемом пересечении. Для конфликтующих потоков ГО-профиль определяется согласно соответствующим моделям просачивания.

OУТ-профиль (профиль исходящего потока)– это профиль потока, фактически проезжающего стоп-линию. При наличии очереди OУТ-профиль равен ГО-профилю, после разъезда очереди на оставшееся время эффективного разрешающего сигнала OУТ-профиль равен IN-профилю:

$$OУT_t = \begin{cases} 0, & \text{если } u(t) = 0; \\ GO_t, & \text{если } u(t) = 1 \text{ и } m_t > 0; \\ IN_t, & \text{если } u(t) = 1 \text{ и } m_t = 0, \end{cases} \quad (2.39)$$

где  $u(t)$  – бинарная функция управления, принимающая значение «1» при включенном разрешающем сигнале и «0» при запрещающем;  $m_t$  – количество автомобилей в очереди.

Количество автомобилей в очереди определяется следующей итерационной процедурой:

$$m_t = \max\{(m_{t-1} + q_t - s_t), 0\}, \quad (2.40)$$

где  $m_t$  – число автомобилей в очереди в интервале времени  $t$  на данной связи;  $q_t$  – число автомобилей, прибывающих за интервал  $t$  согласно IN-профилю;  $s_t$  – число автомобилей отъезжающих за интервал  $t$ , согласно ГО-профилю.

На каждой внутренней связи сети происходит преобразование OУТ-профиля в соответствующий IN-профиль следующим рекуррентным уравнением:

$$v_{(t+\beta T)} = F \cdot v_t + \left[ (1 - F) \cdot v_{(t+\beta T-1)} \right], \quad (2.41)$$

где  $v_{(t+\beta T)}$  – предсказанная интенсивность в интервале времени  $t + \beta T$  предсказанной группы;  $v_t$  – интенсивность начальной группы во время шага  $t$ ;  $\beta$  («бета») – эмпирический фактор, равный в общем случае 0,8;  $T$  – время прохождения маршрута на связи, в шагах;  $F$  – фактор сглаживания, где

$$F = (1 + \alpha \cdot \beta \cdot T)^{-1}, \quad (2.42)$$

где  $\alpha$  («альфа»-фактор) – полученная опытным путем константа, именуемая фактором дисперсии группы (*PDF*).

Отметим, что использованное  $T$  в приведенном выше уравнении является временем прохождения связи до стоп-линии или до конца очереди.

Эффект фактора дисперсии групп ( $PDF$ ) основан на натуральных исследованиях [59]. TRANSYT предполагает значение альфа-фактора по умолчанию 0,35, которое также основано на эмпирических исследованиях, выполненных Transport and Road Research Laboratory.

В основе популярности TRANSYT лежит возможность калибровки  $PDF$ , что обеспечивает высокую реалистичность результатов. В идеале  $PDF$  должен быть откалиброван таким образом, чтобы профили потока модели как можно ближе соответствовали наблюдаемым профилям потока. При отсутствии данных полевых наблюдений предполагается использование табличных значений  $PDF$  (табл. 2.1).

Т а б л и ц а 2 . 1

Значения фактора дисперсии групп в зависимости от условий движения

$PDF$	Характеристика дороги	Описание состояний
0,5	Тяжелые условия	Комбинация стоянки, умеренные к тяжелые повороты, умеренный или интенсивный пешеходный поток, узкая ширина полосы движения. Поток транспорта типичный для городского движения.
0,35	Умеренные условия	Светофорные секции для поворотных потоков, светофорное регулирование пешеходных потоков, нормальная ширина полосы движения. Магистральное или хорошо организованное городское движение.
0,25	Свободные условия	Отсутствие парковок, выделенные поворотные полосы, широкая проезжая часть. Магистральные дороги регулируемого движения.

В TRANSYT используется модель задержки HCM, причем постоянная задержка определяется путем суммирования моделируемого профиля очереди, что не требует для оценки фактора прогрессии:

$$d_1 = \frac{\sum_t^N m_t}{N}; \quad (2.43)$$

где  $m_t$  – длина очереди в числе автомобиля во время шага  $t$ ;  $N$  – число шагов в цикле.

Кроме того, при использовании пошагового моделирования TRANSYT явно определяет остаточную очередь (если она существует) и соответственно задержку перенасыщения  $d_3$ .

### 2.3. Динамика очереди транспортных средств перед регулируемым перекрестком

Приведенные выше модели задержки и очереди транспортных средств предполагают, что все прибывающие к пересечению транспортные средства размещаются на стоп-линии. В свободных условиях движения данное допущение является приемлемым. В условиях насыщенного движения становится важным не только количество автомобилей в очереди, но и ее пространственно-временное распределение по связи. Необходимость учета пространственно-временного распределения очереди по транспортной связи рассмотрим на примере, приведенном на рис. 2.5.

Введем следующие обозначения для описания динамики очереди:

$f_{in}$  – положение фронта очереди, м;

$f_{out}$  – положение фронта отъезда транспортных средств, определяющего точку отъезда очередного транспортного средства.

Положение фронта очереди  $f_{in}$  определяет точку остановки прибывающих транспортных средств. При отсутствии остаточной очереди, в момент включения запрещающего сигнала формируется очередь, фронт которой в дискретном виде определяется выражением (рис. 2.5а):

$$f_{in}(k+1) = f_{in}(k) + q(k) \cdot L_{veh} \cdot \Delta; \quad (2.44)$$

где  $f_{in}(k)$  – положение фронта очереди в текущий момент, м;  $f_{in}(k+1)$  – положение фронта очереди в последующий момент, м;  $\Delta$  – шаг моделирования, с;  $L_{veh}$  – средний динамический габарит автомобиля в очереди, м;  $q(k)$  – интенсивность движения, авт/с.

В указанных условиях положение фронта очереди также может быть определено как:

$$f_{in}(k+1) = Q(k+1) \cdot L_{veh}. \quad (2.45)$$

При включении разрешающего сигнала отъезжающие транспортные средства формируют фронт отъезда  $f_{out}$  (рис. 2.5б). При этом очередь распадается на две части. Первую часть составляют неподвижные транспортные средства, вторую – движущиеся от фронта разъезда к стоп-линии. Можем записать следующее выражение, определяющее положение фронта отъезда:

$$f_{out}(k+1) = f_{out}(k) + V_{out} \cdot \Delta; \quad (2.46)$$

где  $V_{out}$  – скорость распространения фронта отъезда.

Динамика  $f_{out}$  определяется особенностями движения транспортных средств по полосе без возможности совершения обгона – движение  $n+1$  автомобиля возможно после того, как начнет движение  $n$  автомобиль.

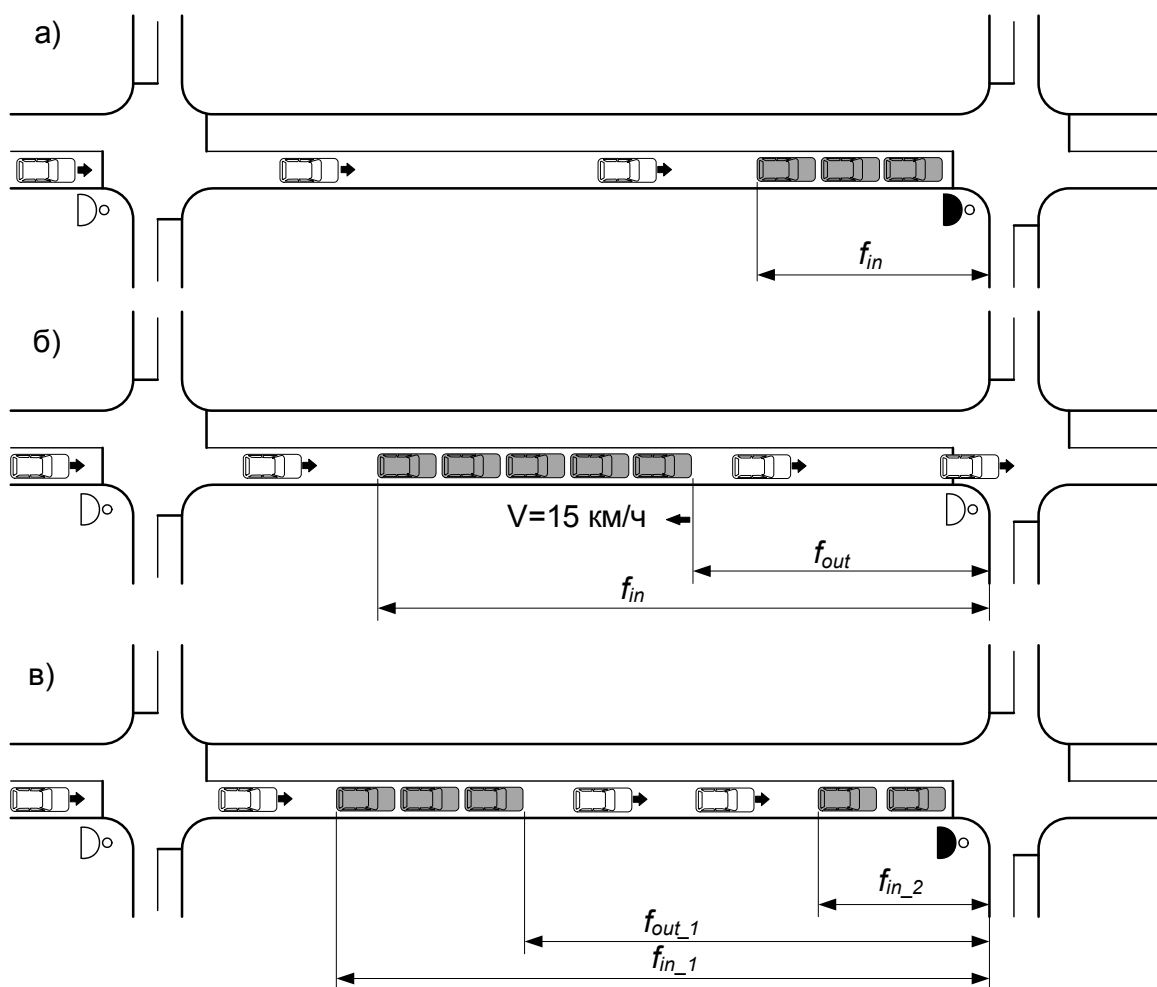


Рис. 2.5 Динамика образования очереди при насыщенном движении

Соответственно скорость распространения фронта отъезда может быть определена как:

$$V_{out} = \frac{L_{vech}}{\tau_{out}}, \quad (2.47)$$

где  $\tau_{out}$  – время реакции водителя на отъезд впереди идущего автомобиля, с.

При среднем динамическом габарите автомобиля 5,3 м и времени реакции водителя 1,2 с скорость распространения фронта отъезда составит 4,42 м/с (15,9 км/ч). Данные натурных исследований, приведенные в [4] показали, что скорость распространения фронта отъезда постоянна и в среднем составляет  $V_{out}=15 \text{ км/ч}$ . С учетом стабильности  $V_{out}$  формулу (2.46) запишем следующим образом:

$$f_{out}(k+1) = f_{out}(k) + 4,167 \cdot \Delta. \quad (2.48)$$

Следует отметить, что при включении разрешающего сигнала светофора (точка  $b$  на рис. 2.6) количество автомобилей в очереди начинает



сокращаться, в то время как положение фронта очереди продолжает увеличиваться. Соответственно выражение (2.45), содержащее количество автомобилей в очереди, уже не может быть использовано для вычисления фронта очереди. Исчезновение фронта очереди, и соответственно начало движения всех транспортных средств (точка  $a$  на рис. 2.6) происходит при выполнении условия:

$$f_{in}(k) = f_{out}(k). \quad (2.49)$$

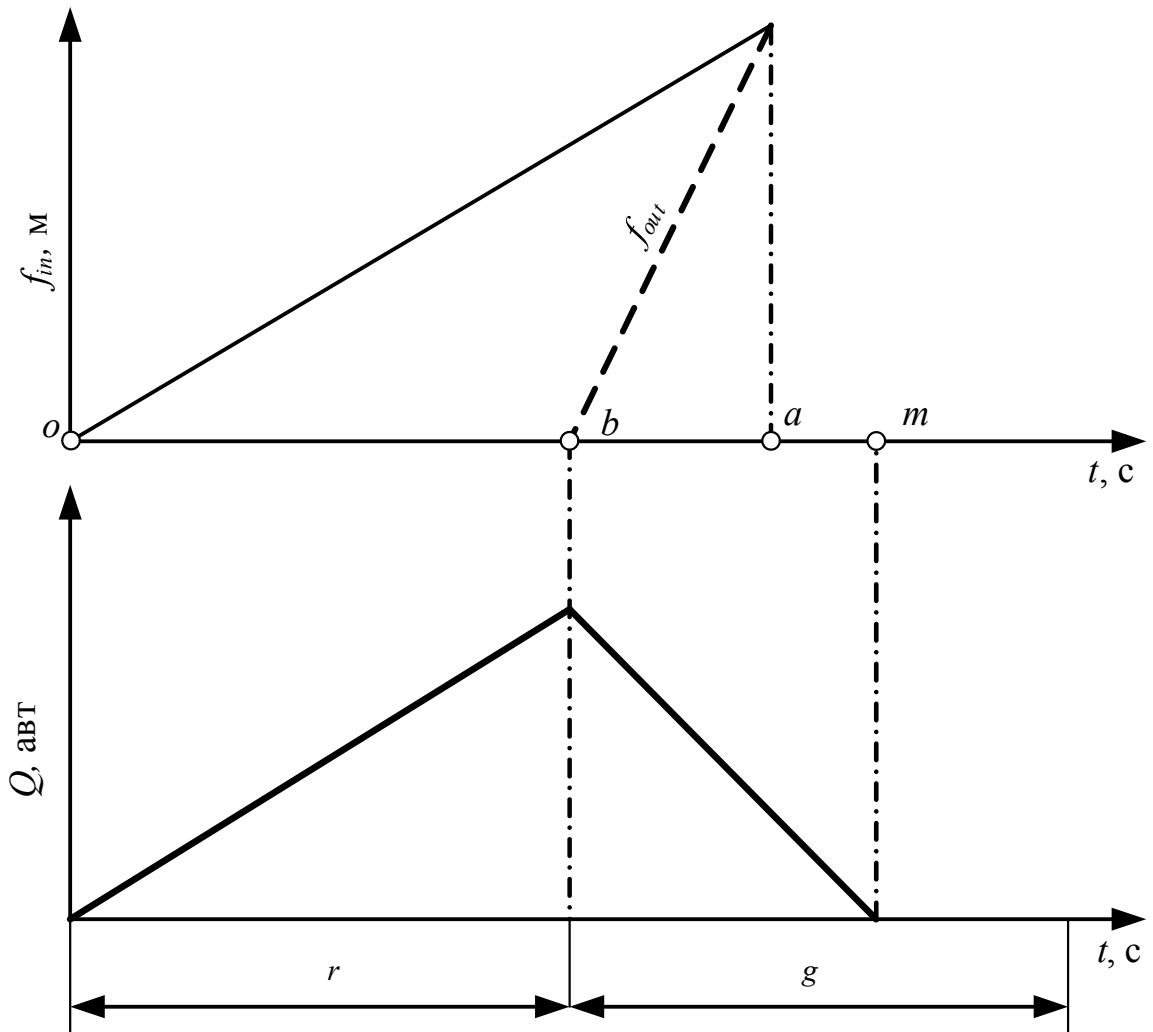


Рис. 2.6 Диаграмма соотношения очереди транспортных средств и положения фронта очереди

С учетом формулы (2.49) положение фронта очереди будет определяться следующим образом:

$$f_{in}(k+1) = \begin{cases} 0 & \text{при } f_{out}(k) = f_{in}(k) \\ f_{in}(k) + q(k) \cdot L_{vech} \cdot \Delta \end{cases} \quad (2.50)$$

Отметим, что очередь транспортных средств после исчезновения фронта очереди продолжает существовать (на участке от точки  $a$  до  $mQ(k) > 0$ ).

Выключение разрешающего сигнала светофора вновь инициирует процесс образования фронта очереди. При достаточно большой длине перегона и интенсивности движения возможно одновременное наличие двух и более фронтов очереди и разъезда на перегоне (рис. 2.5в). Таким, каждый светофорный цикл инициирует на транспортной связи формирование собственного фронта образования и разъезда очереди.

Сформулируем условие блокирования входящих в перегон транспортных связей с учетом приведенных выше зависимостей:

$$f_{in}(k) > L_{edg}; \quad (2.51)$$

где  $L_{edg}$  – длина перегона, м.

Выражение (2.51) определяет условие блокирования входящих транспортных связей в произвольный момент времени, в том числе и при разрешающем сигнале светофора.

Определим момент исчезновения фронта очереди и его положение при этом на транспортной связи. Выделим в цикле регулирования два характерных момента времени. В первый момент времени включен запрещающий сигнал светофора и формируется фронт очереди:

$$f_{in}^I = Q_o \cdot L_{vech} + \sum_{i=0}^r q_i \cdot L_{vech}, \quad (2.52)$$

где  $Q_o$  – остаточная очередь на связи, авт;  $q_i$  – количество прибывших автомобилей в интервале  $i$ , авт;  $r$  – длительность запрещающего сигнала.

Во второй период включается разрешающий сигнал и положение фронта очереди с учетом (2.52) будет определено следующим образом:

$$f_{in}^{II} = Q_o \cdot L_{vech} + \sum_{i=0}^b q_i \cdot L_{vech} + \sum_{j=b}^a q_j \cdot L_{vech}, \quad (2.53)$$

где  $a$  – длительность разрешающего сигнала, во время которого существует фронт очереди.

Момент исчезновения фронта очереди определяется положением точки  $a$ , найдя которую можем решить поставленную задачу. Согласно рис. 2.6 точка  $a$  может быть найдена из выражения:

$$f_{out} = V_{out} \cdot (a - b). \quad (2.54)$$

С учетом (2.49) получим уравнение:

$$V_{out} \cdot (a - b) = Q_o \cdot L_{vech} + \sum_{i=0}^b q_i \cdot L_{vech} + \sum_{j=b}^a q_j \cdot L_{vech}; \quad (2.55)$$

Решение уравнения (2.55) относительно  $a$  при задании интенсивности движения в виде циклических профилей возможно численными методами. Приняв допущение о равномерном характере распределения интенсивности движения в цикле регулирования, преобразуем уравнение (2.55) к следующему виду:

$$V_{out} \cdot (a - b) = Q_o \cdot L_{vech} + \bar{q} \cdot b \cdot L_{vech} + \bar{q} \cdot (a - b) \cdot L_{vech}; \quad (2.56)$$

где  $\bar{q}$  – средняя интенсивность движения в цикле регулирования, авт/с.

Решая уравнения (2.56) относительно  $a$  получим:

$$a = \frac{(Q_o + \bar{q} \cdot b) \cdot L_{vech}}{V_{out} - \bar{q} \cdot L_{vech}} + b. \quad (2.57)$$

Максимальное удаление фронта очереди от стоп-линии  $F^{in}$  в светофорном цикле будем вычисляться как:

$$F^{in} = a \cdot \bar{q} \cdot L_{vech}, \quad (2.58)$$

или с учетом формулы (2.57):

$$F^{in} = \left[ \frac{(Q_o + \bar{q} \cdot b) \cdot L_{vech}}{V_{out} - \bar{q} \cdot L_{vech}} + b \right] \cdot \bar{q} \cdot L_{vech}. \quad (2.59)$$

При использовании предрассчитанных сигнальных планов гарантировать отсутствие блокирования транспортной связи можно при включении разрешающего сигнала с запаздыванием, равным времени распространения фронта разъезда очереди по всей транспортной связи:

$$\delta_{sat} = \frac{L_{edg}}{4,167}. \quad (2.60)$$

где  $\delta_{sat}$  – сдвиг включения разрешающего сигнала при заторе.

Использование сдвига включения разрешающего сигнала светофоров, вычисленного согласно выражению (2.60), позволит исключить блокирование магистрального направления.

### 3. ПОСТАНОВКА ЗАДАЧИ УПРАВЛЕНИЯ НАСЫЩЕННЫМИ ТРАНСПОРТНЫМИ ПОТОКАМИ

Проблема управления насыщенными транспортными потоками принципиально отличается от проблем управления при свободном или групповом движении. Основное отличие состоит в том, что в условиях насыщенного движения традиционными методами управления светофорными объектами невозможно обеспечить приемлемый для водителей уровень обслуживания. Состояние светофорного объекта характеризуется превышением интенсивности движения пропускной способности пересечения и, как следствие, неограниченным ростом очереди перед ним.

В транспортных сетях проблема осложняется из-за невозможности локализовать режим перенасыщения в пределах данного регулируемого пересечения, первоначально явившегося источником образования транспортного затора. Неограниченный рост очереди блокирует транспортные связи предыдущего по ходу движения перекрестка, что в свою очередь вследствие связности сети способствует распространению затора на всю сеть. Таким образом, важнейшей задачей управления движением в условиях насыщенного движения является предотвращение возникновения условий возникновения общесетевых заторов.

#### 3.1. Управление транспортными потоками с использованием прогнозирующей модели

Одним из перспективных методов управления, который может быть успешно применен для управления насыщенными транспортными потоками, является управление с использованием прогнозирующих моделей (Model Predictive Control – MPC). Основы данного подхода к управлению были заложены в начале 60-х годов применительно к управлению процессами и оборудованием в нефтехимическом и энергетическом производстве, для которых применение классических методов синтеза систем управления было затруднено в связи со сложностью их математических моделей. В настоящее время сфера практического приложения MPC-методов существенно расширилась, включая задачи управления транспортными потоками [23, 54, 67].

Основным достоинством управления с прогнозирующей моделью является простота формирования обратной связи в сочетании с высокой адаптивностью метода. MPC-контроллеры позволяют управлять многомерными объектами со сложной структурой, включающей нелинейности, способны оптимизировать процессы в режиме реального времени с учетом

ограничений на управляющие и управляемые переменные, учитывать неопределенности в описании объектов и возмущений.

Основой MPC – управления является схема управления динамическими объектами по принципу обратной связи. На каждом шаге управления решается задача оптимального управления на прогнозируемый период  $k_s$  (именуемый горизонтом прогноза) и реализуется на горизонте управления  $k_c$ , следующим образом:

1) рассматривается математическая модель объекта управления, начальными условиями для которой служит его текущее состояние. При заданном программном управлении выполняется моделирование состояния объекта на горизонте прогноза  $k_s$ ;

2) выполняется оптимизация программного управления, целью которого служит приближение регулируемых переменных прогнозирующей модели к оптимальным значениям на горизонте прогноза. Оптимизация осуществляется с учётом ограничений, накладываемых на управляющие и регулируемые переменные;

3) на временном шаге  $k_c$ , именуемом горизонтом управления и составляющем фиксированную малую часть горизонта прогноза  $k_s$ , реализуется найденное оптимальное управление и осуществляется измерение (или оценка по измеренным переменным) фактического состояния объекта на конец шага управления;

4) горизонт прогнозирования смещается на горизонт управления  $k_s$  и задача оптимального управления решается вновь (повторяются п. 1 – 3).

Горизонт прогноза  $k_s$  и горизонт управления  $k_c$  связаны соотношением:

$$k_s = n \cdot k_c, \quad (3.1)$$

где  $n$  – достаточно большое целое число, как правило  $n=5 \dots 15$ .

Используя в своей основе оптимальное управление, MPC – управление реализует полноценное управление с обратной связью в реальном времени. Благодаря этому обеспечивается возможность эффективного управления в условиях неопределенности, вызванной действием различных возмущающих факторов и несовершенством математической модели объекта управления.

В общем случае на процесс движения транспортных средств оказывает влияние различные возмущающие факторы  $\mathbf{d}(k_s)$  (изменение транспортного спроса, возникновение инцидентов на транспортной сети, изменение погодных условий и т.д.), приводящие к изменению его состояния  $\mathbf{x}(k_s)$  (рис. 3.1). Состояние транспортного процесса, как правило, характеризуется такими макроскопическими характеристиками как скорость, интенсивность, длина очереди и т.д. MPC–контроллер формирует управление  $\mathbf{u}(k)$ , которое может включать как директивное воздействие в форме сигналов светофоров и ограничений скорости движения, так и элементы косвенного управления (информирование водителей о рекомендуемых маршрутах движения).

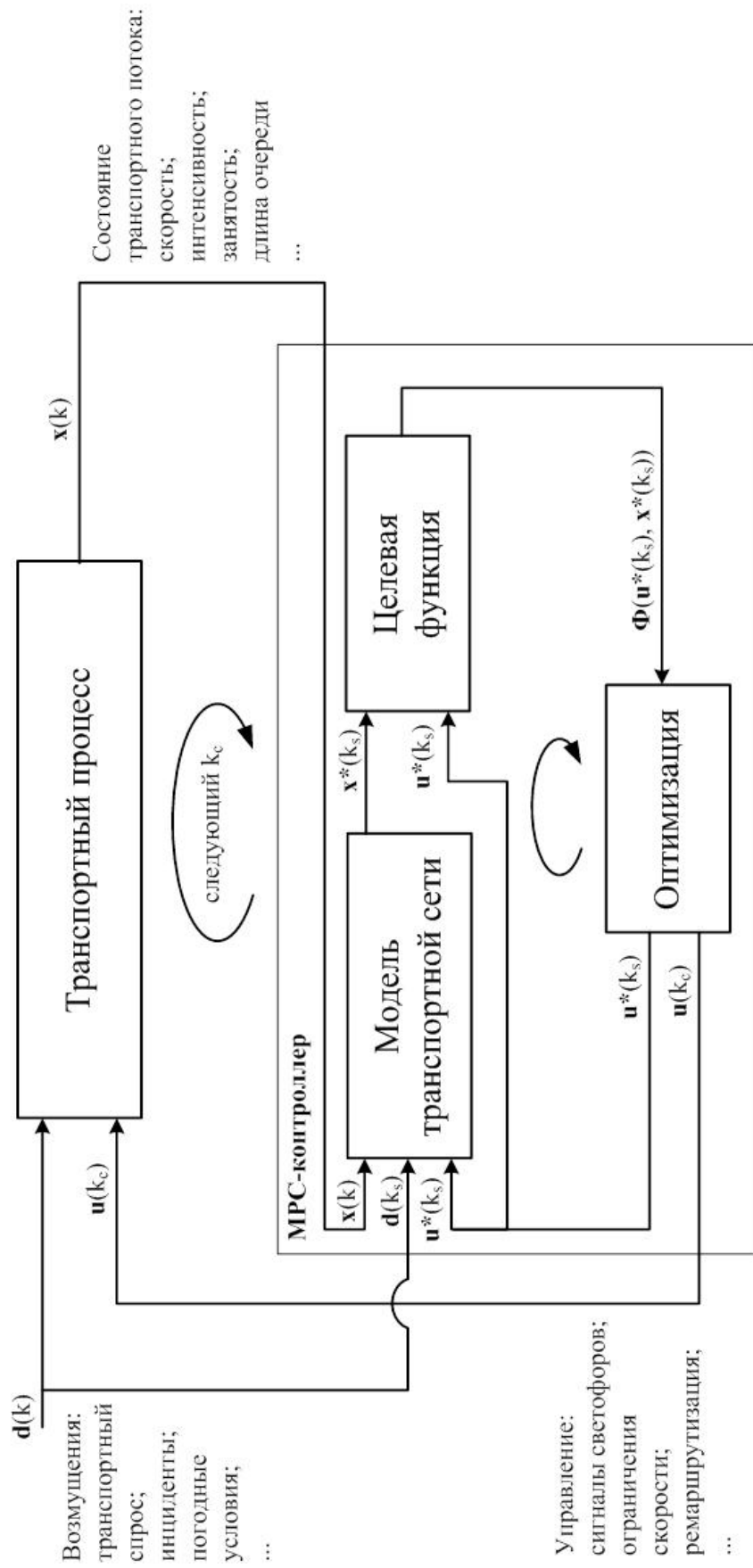


Рис. 3.1. Структурная схема MPC-управления

Формирование управления  $\mathbf{u}(k_s)$  производится путем решения задачи оптимизации функционала управления:

$$\Phi(\mathbf{u}^*(k_s), \mathbf{x}^*(k_s)) \rightarrow \min(\max); \quad (3.2)$$

где  $\Phi(\cdot)$  – функционал управления;  $\mathbf{u}^*(k_s)$  и  $\mathbf{x}^*(k_s)$  – соответственно управление и состояние системы на горизонте прогноза  $k_s$ .

Следует отметить следующие особенности управления с прогнозирующей моделью:

- в качестве прогнозирующей модели может использоваться любая модель транспортной сети, позволяющая предсказать ее состояние на горизонте прогноза  $k_s$ ;

- в зависимости от используемой технологии оптимизации на управление  $\mathbf{u}(k)$  и состояние объекта управления  $\mathbf{x}(k)$  могут быть ограничения как линейного, так и не линейного характера;

- подход предусматривает оптимизацию функционала, характеризующего качество процесса управления, в режиме реального времени;

- необходима непосредственное измерение или оценка состояния объекта управления на каждом шаге управления;

- для работы в реальном масштабе времени необходимо, чтобы решение оптимизационной задачи осуществлялось достаточно быстро, в пределах допустимого запаздывания.

С учетом выше изложенного, синтез MPC – управления будет включать следующие операции:

- постановку задачи оптимального управления и ограничений на управляемые переменные  $\mathbf{u}(k)$  и состояние объекта  $\mathbf{x}(k)$ ;

- выбор модели транспортной сети и величины шагов горизонта управления и прогноза;

- выбор технологии оптимизации, обеспечивающей решение задачи оптимального управления в пределах допустимого запаздывания.

### 3.2. Формализация задачи локального управления светофорным объектом

Сформулируем задачу управления с прогнозирующей моделью применительно к оптимальному управлению изолированным перекрестком. Используемая в отечественной [5, 16] и зарубежной [57, 58] практике методика расчета режимов работы светофорных объектов предусматривает определение длительность светофорного цикла  $C$  по следующей формуле:

$$C = \frac{1,5 \cdot T_n + 5}{1 - Y}, \quad (3.3)$$

где  $T_n$  – суммарное потерянное время за цикл. с;  $Y$  – суммарный фазовый коэффициент  $Y = \sum_{i \in F} \max(q_{i,j}/s_j)$ ;  $F$  – множество фаз регулирования на пересечении;  $j$  – группа полос движения,  $j \in J$ ;  $J$  – множество смежных полос, движение по которым регулируется отдельными сигналами светофоров;  $q_{i,j}$  – интенсивность движения в  $i$  фазе по полосе движения, авт/ч;  $s_j$  – поток насыщения по полосе движения, авт/ч.

Полученная длительность светофорного цикла распределяется между фазами пропорционально критическим фазовым коэффициентам:

$$g_i = \frac{(C - T_p) \cdot \max(q_{i,j}/s_j)}{Y}, \quad (3.4)$$

где  $g_i$  – длительности  $i$  фазы регулирования.

Расчёт длительности фаз по формулам (3.3)–(3.4) обеспечивает коэффициент загрузки для критических групп полос движения на уровне 0,85–0,9. При достаточно высокой интенсивности движения  $v_{i,j}$  использование указанных формул становится затруднительным по следующим причинам:

- при  $Y \rightarrow 1,0$  длительность светофорного цикла  $C \rightarrow \infty$ ;
- при  $Y > 1,0$  длительность светофорного цикла  $C < 0$ .

Кроме того, не контролируемое увеличение длины очереди на входящих транспортных связях может привести к блокированию предыдущих по ходу движения перекрестков и привести к образованию общесетевого затора. Как правило, указанный процесс возникает на одной из входящих в регулируемый перекресток транспортных связей в результате суточных колебаний интенсивности движения или одновременного уменьшения пропускной способности (вследствие изменения погодных условий или закрытия полос движения в результате ДТП, при проведении ремонтных работ и т.д.).

В условиях кратковременного перенасыщения возможно решение задачи недопущения перехода локального транспортного затора в общесетевую за счет контролируемого перераспределения очереди по всем входящим транспортным связям. Соответствующую задачу оптимального управления светофорным объектом можно записать следующим образом:

$$\sum_{n=1}^N d_j [g_i(t + n \cdot k_s), q_{i,j}(t + n \cdot k_s)] \cdot q_{i,j}(t + n \cdot k_s) \rightarrow \min, \quad (3.5)$$

при ограничениях

$$g_{\min} \leq g_i(t + n \cdot k_s) \leq g_{\max}; \quad (3.6)$$



$$F_j^{in}(t + n \cdot k_c) \leq L_j, \quad (3.7)$$

где  $d_j(\cdot)$  – функция средней задержки транспортных средств;  $g_i(t + n \cdot k_c)$  – длительность разрешающего сигнала в  $i$  фазе на  $n$  шаге управления, с;  $g_{\min}$  и  $g_{\max}$  – соответственно, минимальная и максимальная длительность разрешающего сигнала, с;  $F_j^{in}(t + n \cdot k_c)$  – максимальное удаление фронта очереди от стоп-линии, вычисленное по формуле (2.59);  $L_j$  – длина группы полос движения  $j$ , м.

Функция  $d_i(\cdot)$  должна обеспечивать вычисление задержки как в условиях свободного, так и насыщенного движения. Воспользуемся моделью задержки, приведенной в Highway Capacity Manual [5]:

$$d = d_1 PF + d_2 + d_3, \quad (3.8)$$

где  $d_1$  – постоянная составляющая задержки, с/авт;  $PF$  – фактор прогрессии;  $d_2$  – случайная задержка, с/авт;  $d_3$  – задержка перенасыщения, с/авт.

Постоянная составляющая задержки вычисляется как:

$$d_1 = 0,5 \cdot C \frac{(1 - g/C)^2}{1 - (\min(1, X) \cdot g/C)}, \quad (3.9)$$

где  $X$  – степень насыщенности группы полос движения ( $q/c$ );  $c$  – пропускная способность группы полос движения  $c = s \cdot g/C$ , авт/ч.

Случайная задержка определялась с учетом не стационарности процесса формирования очереди:

$$d_2 = 900T \left\{ (X - 1) + \sqrt{(X - 1)^2 + \frac{8kIX}{cT}} \right\}, \quad (3.10)$$

где  $T$  – продолжительность периода анализа в часах;  $k$  – коэффициент корректировки, учитывающий способ управления (для преедрассчитанного управления принимается равным 0,50);  $I$  – коэффициент корректировки, учитывающий влияние соседних перекрестков (для изолированных регулируемых пересечений равен 1,0).

Задержка перенасыщения определяется величиной остаточной очереди на анализируемом периоде  $T$  по формуле:

$$d_3 = \frac{1800 \cdot Q_b (1 + u) \cdot t}{cT}; \quad (3.11)$$

$$t = \begin{cases} 0 & \text{при } Q_b = 0 \\ \min \left( T, \frac{Q_b}{c(1 - \min(1, X))} \right) & \end{cases}$$

$$u = \begin{cases} 0 & \text{при } t < T \\ 1 - \frac{cT(1 - \min(1, X))}{Q_b} & \end{cases}$$

где  $Q_b$  – величина остаточной очереди, авт;  $t$  – продолжительность неудовлетворения спроса, ч.

Постоянная составляющая задержки  $d_1$ , в период наличия остаточной очереди будет вычисляться при  $X=1,0$ , для периода  $(T-t)$  отсутствия остаточной очереди будет использоваться актуальное значение  $X$ :

$$d_1 = 0,5 \cdot C \frac{(1-g/C)^2}{1-g/C} \cdot \frac{t}{T} + 0,5 \cdot C \frac{(1-g/C)^2}{1-(\min(1, X) \cdot g/C)} \cdot PF \cdot \frac{T-t}{T}. \quad (3.12)$$

Положение очереди  $Q_j$  ограничения (5) в Highway Capacity Manual определяется формулой:

$$Q = Q_1 + Q_2; \quad (3.13)$$

$$Q_1 = PF_2 \frac{\frac{v_L C}{3600} \left(1 - \frac{g}{C}\right)}{1 - \min(1, X_L) \frac{g}{C}}; \quad (3.14)$$

$$Q_2 = 0,25 \cdot c_L T \left[ (X_L - 1) + \sqrt{(X_L - 1)^2 + \frac{8k_B X_L}{c_L T} + \frac{16k_B Q_{bL}}{(c_L T)^2}} \right], \quad (3.15)$$

где  $v_L$  – интенсивность движения по полосе движения, авт/ч;  $s_L$  – поток насыщения по полосе движения, авт/ч;  $c_L$  – поток по полосе движения, авт/ч;  $Q_{bL}$  – остаточная очередь по полосе движения, авт;  $X_L$  – загруженность полосы движения ( $q_L/c_L$ );  $PF_2$  – коэффициент прогрессии;  $k_B$  – коэффициент, связанный с сокращением времени прибытия к фронту очереди.

Как видим поставленная задача оптимизации работы светофорного объекта (3.5)–(3.7) является задачей нелинейного целочисленного программирования с ограничениями как линейного (3.6), так и нелинейного (3.7) вида.

Введем функцию штрафа:

$$F_{pen}(x) = \begin{cases} \inf & \text{если } x > 0 \\ 0 & \text{иначе} \end{cases}, \quad (3.16)$$

где  $\inf$  – достаточно большое число, эквивалентное бесконечности.

Модифицируем задачу (3.5)–(3.7) путем введения функции штрафа (3.16) в целевую функцию:

$$\sum_{n=1}^N d_j [g_i(t+n \cdot k_s), v_{i,j}(t+n \cdot k_s)] \cdot q_{i,j}(t+n \cdot k_s) + \sum_n F_{pen} (F_j^{in}(t+n \cdot k_c) - L_j) \rightarrow \min. \quad (3.17)$$

Использование MPC – управления требует измерения возмущений и переменных состояния объекта управления. Функционирование регулируемого пересечения можно представить структурной схемой, изображенной на рис. 3.2.

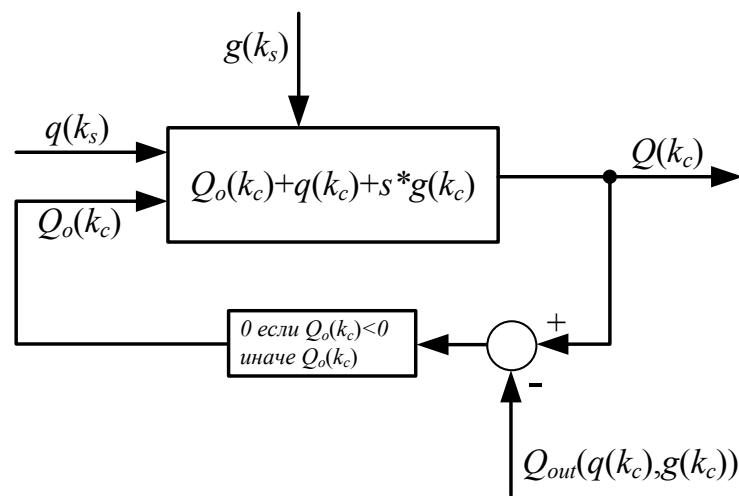


Рис. 3.2. Структурная схема функционирования регулируемого пересечения

Состояние регулируемого пересечения будем характеризовать длиной очереди транспортных средств  $Q(k_c)$ , которая функционально связана с задержкой. Длина очереди  $Q(k_c)$  определяется интенсивностью движения  $q(k_s)$ , действием разрешающего сигнала  $g(k_s)$  и наличием остаточной очереди  $Q_o(k_c)$ . Возмущающей переменной в данном случае является интенсивность движения, управляющей – длительность разрешающих сигналов  $g(k_s)$ .

Оценка состояния светофорного объекта и вычисление значений целевой функции управления по формулам (3.8)–(3.15), (3.17) на горизонте прогноза  $k_s$ .

Для решения поставленной задачи могут быть использованы численные методы оптимизации, такие как эволюционные алгоритмы, метод роя частиц, имитация отжига и др., при этом наиболее широкое применение при расчете режимов работы светофорных объектов получили генетические алгоритмы [1, 3].

При расчете режимов работы изолированного светофорного объекта размер области возможных решений  $P$  определен количеством фаз регулирования  $N$  и размахом возможных значений длительности разрешающих сигналов:

$$P = \left( \frac{g_{\max} - g_{\min}}{\Delta g} \right)^N. \quad (3.18)$$

где  $\Delta g$  – шаг приращения длительности разрешающих сигналов, с.

На практике схемы движения с числом фаз регулирования  $N$  более 4 не используется, а длительность разрешающего сигнала по условиям безопасности определена на диапазоне значений  $7 \leq g_i \leq 90$  с. С учетом указанных особенностей области возможных решений, целесообразно решение задачи (3.17) методом перебора по сетке возможных решений с шагом  $\Delta g$ .

Существенно сократить количество вычислений возможно путем использования итерационной процедуры поиска оптимального решения с переменным шагом  $\Delta g$ :

- 1) устанавливаем начальные значения  $\Delta g_1, g_{\min}, g_{\max}$ ;
- 2) сгенерировать сетку решений *SolveGrid* по каждой фазе в диапазоне значений  $g_{\min} \leq g_i \leq g_{\max}$  с шагом  $\Delta g$ ;
- 3) для каждого узла решения из *SolveGrid* по формуле (3.17) вычислить значение целевой функции;
- 4) решение  $green \in SolveGrid$ , имеющее минимальное значение целевой функции является оптимальным;
- 5) модифицируем начальные значения  $\Delta g_2, g_{\min} = green - \Delta g_1, g_{\max} = green + \Delta g_1$ ;
- 6) выполнить п. 2 – 4.

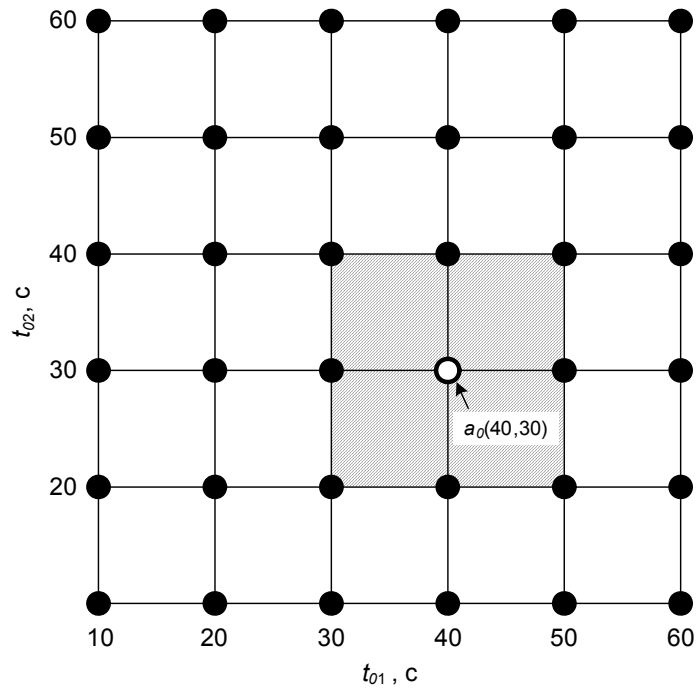
Поясним работу алгоритма на примере поиска длительности фаз, при двухфазной схеме разезда приведенной на рис. 3.3. На первой итерации генерируется сетка решений с шагом  $\Delta g_1 = 10$  с и  $g_{\min} = 7$  и  $g_{\max} = 60$  с (рис. 3.3а), после чего находим значения целевой функции для каждого узла. Пусть минимальное значение целевой функции найдено в точке  $a_0$  с координатами (40, 30), соответствующее длительности фаз  $t_{o1} = 40$  и  $t_{o2} = 30$  с.

Таким образом, положение оптимального решения локализовано в районе точки  $a_0$  с радиусом 10 с.

На второй итерации производим генерацию новой сетки решений с шагом  $\Delta g_2 = 5$ ,  $g_{\min} = 30$  и  $g_{\max} = 50$  для первой фазы движения, и соответственно  $g_{\min} = 20$  и  $g_{\max} = 40$  для второй. Количество вычислений значений целевой функции может быть сокращено за счет использования результатов вычислений, выполненных на первой итерации (на рис. 3.3б

для точек, обозначенных квадратом). Найденная на второй итерации точка оптимума  $a_1$  с координатами (35, 25) может быть принята в качестве окончательного решения или использована в качестве точки локализации оптимума на последующей итерации.

а)



б)

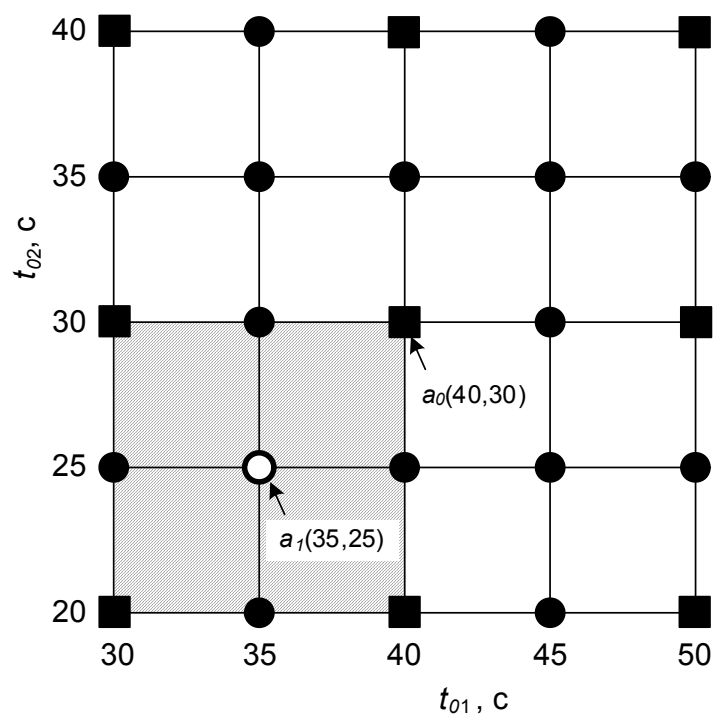


Рис. 3.3. Процедура поиска оптимальных длительностей фаз с переменным шагом  $\Delta g$

Измерения состояния транспортного потока выполняются детекторами транспорта, принцип работы и технические возможности которых можно найти в [9, 11, 105]. В не зависимости от используемой технологии обнаружения транспортных средств, детекторы транспорта позволяют выполнять точечную оценку следующих параметров транспортного потока:

– интенсивности движения в виде посекундного профиля  $q(t)$  или усредненных значений  $\bar{q} = \sum q(t)$ ;

– средней скорости транспортного потока  $\bar{v}$ ;

– занятости контролируемого сечения  $\phi = \sum t_{vech} / t$  ( $t_{vech}$  – время нахождения автомобилей в зоне контроля,  $t$  – продолжительность наблюдения).

Требуемая для выполнения вычислений значений целевой функции (3.17) величина остаточной очереди  $Q_o$  не может быть непосредственно измерена детекторами транспорта. Она может быть вычислена, учитывая закономерности формирования и разъезда очереди.

Рассмотрим функционирование детектора транспорта, расположенного на расстоянии  $L_{BQ}$  от стоп-линии (рис. 3.4). Вычисление остаточной очереди может быть выполнено дважды за цикл регулирования в характерные моменты времени.

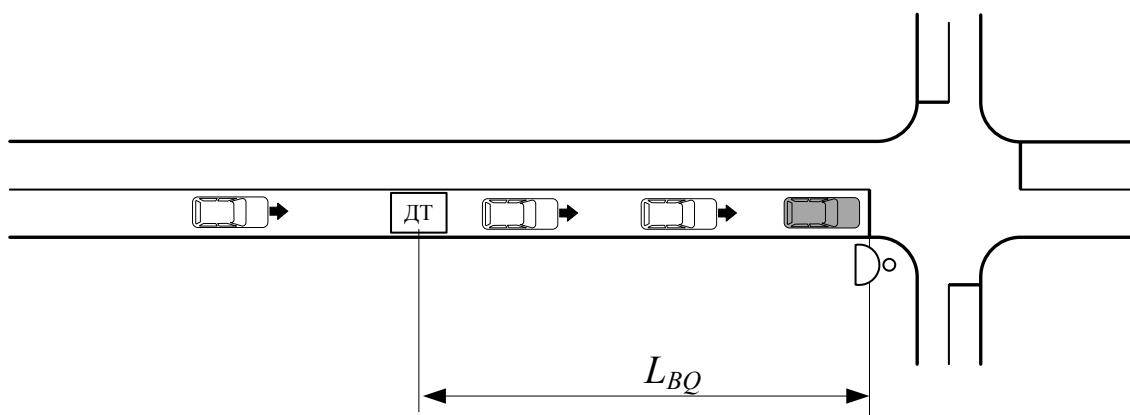


Рис. 3.4. Схема к определению остаточной очереди при помощи детектора транспорта

Первый раз – в момент достижения фронтом очереди детектора транспорта при включении запрещающего сигнала, второй – в момент разъезда очереди на разрешающий сигнал. Рассмотрим подробнее каждый из указанных вариантов оценки величины остаточной очереди  $Q_o$ .

При включении запрещающего сигнала светофора, как отмечено ранее, формируется фронт очереди, распространяющийся от стоп-линии в направлении детектора транспорта. Положение фронта очереди определяется

выражением (2.53). Подставим в левую часть указанного выражения положение детектора транспорта  $L_{BQ}$ :

$$L_{BQ} = (Q_o + \bar{q} \cdot t) \cdot L_{vech}. \quad (3.19)$$

Из полученного уравнения (3.19) выразим искомую величину остаточной очереди  $Q_o$ :

$$Q_o^{obs} = \max \left( 0, \frac{L_{BQ}}{L_{vech}} - \bar{q} \cdot t \right). \quad (3.20)$$

При задании интенсивности движения в форме циклического профиля величина остаточной очереди  $Q_o$  вычисляется по формуле:

$$Q_o^{obs} = \max \left( 0, \frac{L_{BQ}}{L_{vech}} - \sum_{t_c=g}^T q(t) \right). \quad (3.21)$$

При включении разрешающего сигнала разъезд очереди характеризуется наличием потока насыщения, что позволяет идентифицировать его окончание по временным интервалам между транспортными средствами:

$$f^{clear}(\Delta_{vech}) = \begin{cases} 1 & \text{если } \Delta_{vech} > \Delta_{crit}; \\ 0 & \text{иначе,} \end{cases} \quad (3.22)$$

где  $\Delta_{vech}$  – интервал между автомобилями, с;  $\Delta_{crit}$  – критический интервал, определяющий наличие потока насыщения, с.

Количество автомобилей, остановленных за цикл регулирования, будет вычислено следующим образом:

$$Q_o^{obs} = \frac{L_{BQ}}{L_{vech}} + \sum_{t=g}^{t_{clear}} q(t), \quad (3.23)$$

где  $t_{clear}$  – время разъезда очереди в зоне детектирования, с;  $q(t)$  – количество автомобилей, проехавших через детектор транспорта за цикл измерений, авт/с.

Величина  $t_{clear}$  определяется бинарной функцией  $f^{clear}(\Delta_{vech})$  следующим итерационным алгоритмом:

1.  $t_{clear} = 0$ ;
2. **fort** =  $g, g+\Delta, \dots, g+T$ ;
3. **if**  $f^{clear}(\Delta_{vech}) < 1$ :
4.  $t_{clear} = t_{clear} + \Delta$
5. **else:break**
6.  $t = t + \Delta$
7. **end**

Величину остаточной очереди найдем как разницу между наблюдаемой очередью  $Q^{obs}$  и ожидаемой  $Q^p$  при отсутствии остаточной очереди:

$$Q_o^{obs} = Q^{obs} - Q^p. \quad (3.24)$$

Величину ожидаемой очереди  $Q^p$  вычислим по формулам (3.13)–(3.15) при  $Q_0 = 0$ :

$$Q^p = PF_2 \frac{\frac{v_L C}{3600} \left(1 - \frac{g}{C}\right)}{1 - \min(1, X_L) \frac{g}{C}} + 0.25 \cdot c_L T \left[ (X_L - 1) + \sqrt{(X_L - 1)^2 + \frac{8k_B X_L}{c_L T}} \right]. \quad (3.25)$$

Полученные выражения (3.20), (3.21) и (3.22)–(3.25) позволяют определить как величину остаточной очереди, так и количество остановленных автомобилей. При выборе места установки детекторов транспорта следует учитывать следующие факторы:

- оценка остаточной очереди возможна только при достижении фронтом очереди детектора транспорта;
- протяженность оцениваемой остаточной очереди не может превышать расстояния от стоп-линии до точки установки детектора транспорта.

Отметим, что использование, как предложенной, так и иной формулировки задачи локального управления светофорным объектом позволяет исключить блокирование смежных пересечений только в ограниченный период. В том случае, если не удастся локализовать транспортный затор на уровне отдельного пересечения, требуется решение указанной задачи на уровне сети.

### 3.3. Формализация задачи управления магистралью

Проблемы управления насыщенными транспортными потоками принципиально отличается от проблем управления при свободном движении, а транспортных сетях проблема осложняется из-за невозможности локализовать режим перенасыщения в пределах отдельного перекрестка, первоначально явившегося источником возникновения затора. Отрицательную роль играет фактор связности между отдельными перекрестками – неограниченно возрастающая очередь на одном перекрестке блокирует транспортные потоки предыдущего по ходу движения перекрестка, который в свою очередь, может заблокировать работу смежных перекрестков и т.д. Указанный процесс может полностью заблокировать на длительное время движение в отдельной области транспортной сети.



В связи с этим, основной задачей систем управления в условиях насыщенного движения должно являться предотвращение случаев возникновения транспортных заторов, а при возникновении – максимально быстрая ликвидация их последствий.

Естественный путь решения задачи предотвращения заторов состоит в устранении причин, вызвавших перегрузку и узком месте сети (если только такими причинами не являлись дорожно-транспортное происшествие или какое-либо специальное мероприятие массового характера). Поскольку управляющая система не в состоянии увеличить пропускную способность соответствующего перекрестка, единственным путем снижения риска возникновения заторов является своевременное ограничение количества транспорта, прибывающего в опасное сечение сети.

Рассмотрим формирование транспортных потоков внутри транспортной сети при постоянном транспортном спросе. Введем следующие обозначения:

$J$  – множество транспортных связей области управления;

$j^{pr}$  – множество «защищенных» транспортных связей, на которых необходимо обеспечить движение без образования заторов,  $j^{pr} \in J$ ;

$N$  – множество регулируемых пересечений.

Рассмотрим процесс формирования транспортных потоков в пределах отдельного пересечения (рис. 3.5) происходит преобразование входящих  $q_a^{arriv}$  транспортных потоков в исходящие  $q_b^{leaving}$ .

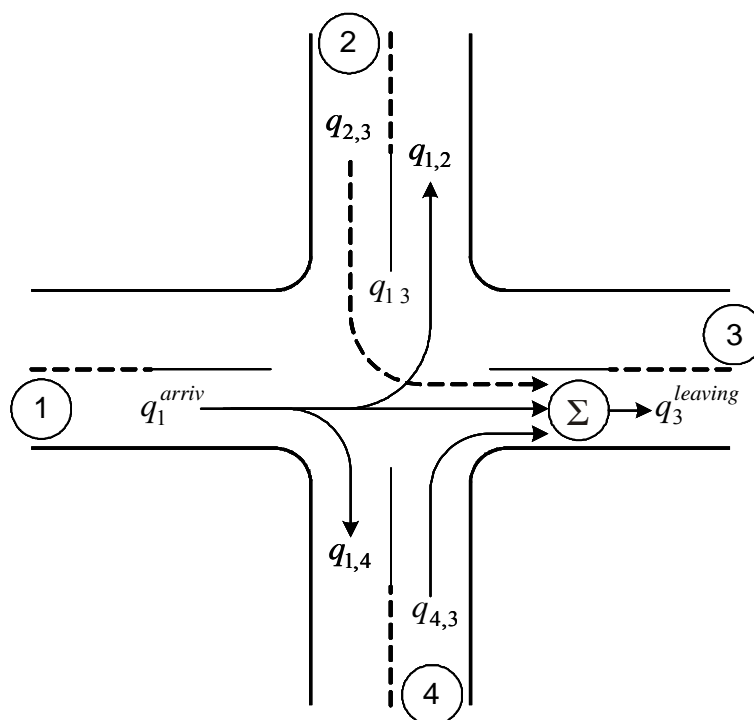


Рис. 3.5. Формирование транспортных потоков на пересечении

Каждый из входящих потоков  $q_a^{arriv}$  в пределах перекрестка делится на  $m$  потоков следующим образом:

$$q_a^{arriv} = \sum \beta_{a,b} q_a^{arriv}; \quad (3.26)$$

где  $\beta_{a,b}$  – доля входящего потока  $q_a^{arriv}$ , движущегося со связи  $a$  на связь  $b$ ,  $\sum \beta_{a,b} = 1.0$ ;  $a$  – множество связей, входящих в перекресток,  $a \in J$ ;  $b$  – множество связей, исходящих из перекрестка,  $b \in J$ ;  $J$  – множество транспортных связей области управления.

Интенсивность движения на исходящей с перекрестка связи  $b$  будем определять как сумму поворотных потоков на указанную связь:

$$q_b^{leaving} = \sum_a \beta_{a,b} q_a^{arriv}. \quad (3.27)$$

Активное воздействие сигналов светофоров накладывает ограничения на количество проезжающих через перекресток автомобилей. Интенсивность исходящих потоков в зависимости от длительности разрешающих сигналов будет определяться формулой:

$$q_b^{leaving} = \sum_a \min \left( \beta_{a,b} q_a^{arriv}, \frac{c_a \cdot g_{n,a}}{T_n} \right), \quad (3.28)$$

где  $c_a$  – поток насыщения связи  $a$ ;  $g_{n,a}$  – длительность разрешающего сигнала на связи  $a$  пересечения  $n \in N$ ;  $T_n$  – длительность светофорного цикла на пересечении  $n$ .

Рассмотрим формирование транспортных потоков в сети на примере участка, включающего два смежных перекрестка (рис. 3.6).

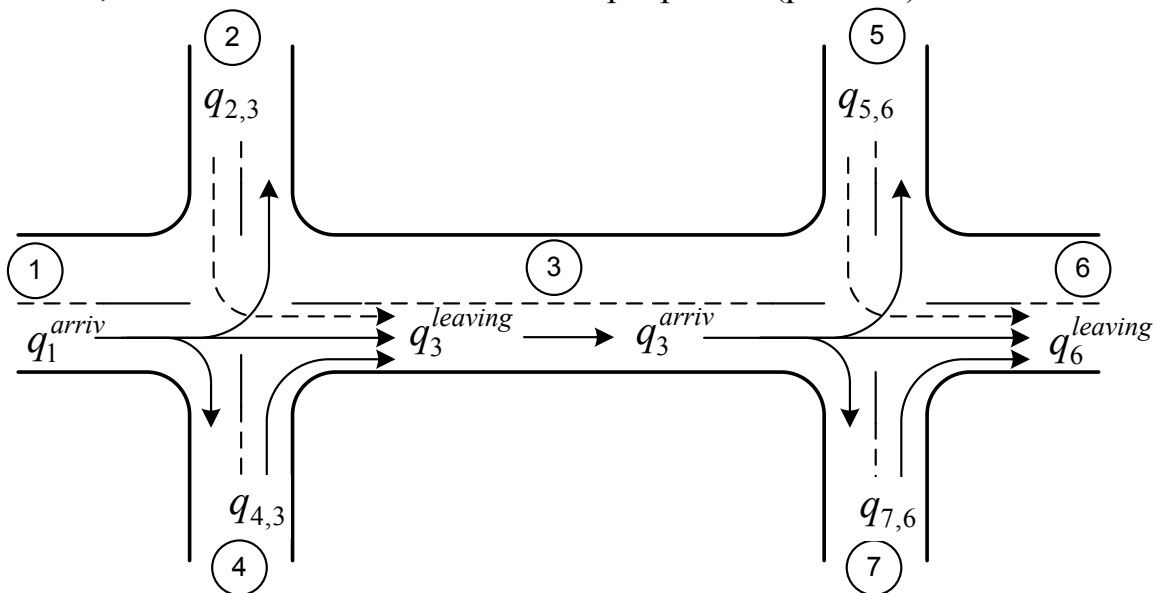


Рис. 3.6. Формирование потоков в транспортной сети

Однозначно интенсивность движения определена на внешних для рассматриваемой сети связях  $\{1, 2, 4, 5, 7\}$ . На внутренней и исходящей связи  $\{3, 6\}$  интенсивности движения зависят от режимов работы светофорных объектов. Причем, если для определения интенсивности движения на связи 3 известны все входящие потоки, то для связи 6 вначале требуется определить не известную на момент начала расчета интенсивность  $q_3^{arriv}$ .

Таким образом, в условиях воздействия сигналов светофоров оценка интенсивности движения в сети не может быть выполнена одномоментно на всех связях  $J$ . Для получения всех значений интенсивностей движения на связях сети необходимо выполнение следующей итерационной процедуры:

- 1) зададим кортеж  $M$ , определяющий интенсивность входящих потоков на связях  $j$ ;
- 2) задаем кортеж  $K$ , который будет включать связи транспортной сети, для которых определены все входящие потоки;
- 3) выполняем расчет входящих потоков для связей кортежа  $M$  по формуле (3.26);
- 4) связи, для которых определены все входящие потоки, перенесем в кортеж  $Q$ ;
- 5) выполним расчет исходящих потоков по формуле (3.28);
- 6) если кортеж  $M$  не пуст, выполняем п. 3–5, иначе процедура завершается.

В результате выполнения процедуры получаем кортеж  $Q$ , в котором определены все входящие потоки связей  $J$  с учетом принятого управления  $g$ .

В общем случае, причиной образования транспортного затора является возникновение остаточных очередей на подходе  $u$  вследствие превышения количества вошедших в связь автомобилей  $q_u^{leaving}$  числа ее покинувших  $q_u^{arriv}$ . Выход автомобилей из зоны управления, как правило, происходит по магистральным улицам общегородского или районного значения, соответственно множество  $j^{pr}$  включает связи указанных улиц.

Рассмотрим решение задачи управления светофорными объектами, обеспечивающей балансировку транспортного спроса и предложения в пределах отдельной области транспортной сети. Сформулируем задачу оптимального управления светофорными объектами как задачу максимизации пропускной способности сети:

$$\sum_j q_j^{arriv}(t + n \cdot k_s) \rightarrow \max; \quad (3.29)$$

при ограничениях:

$$q_{j^{pr}}^{leaving}(t + n \cdot k_s) - q_{j^{pr}}^{arriv}(t + n \cdot k_s) \leq 0; \quad (3.30)$$

$$g_{\min} \leq g(t + n \cdot k_s) \leq g_{\max}, \quad (3.31)$$

$$F_j^{in}(t + n \cdot k_c) \leq L_j, \quad (3.32)$$

где  $\sum_j q_j^{arriv}(t + n \cdot k_s)$  – суммарная интенсивность отъезда в районе управления,  $j \in J$ ;  $(t + n \cdot k_s)$  – управляющие воздействия (множество длительностей сигналов управления на светофорных объектах);  $q_{jpr}^{leaving}(t + n \cdot k_s)$  – интенсивность входящих в транспортную связь транспортных потоков;  $q_{jpr}^{arriv}(t + n \cdot k_s)$  – интенсивность исходящих с транспортной связи транспортных потоков;  $g_{\max}$ ,  $g_{\min}$  – соответственно верхние и нижние ограничения на длительность сигналов управления.

## 4. АПРОБАЦИЯ МЕТОДИКИ УПРАВЛЕНИЯ НАСЫЩЕННЫМИ ТРАНСПОРТНЫМИ ПОТОКАМИ

### 4.1. Методика проведения исследований

Транспортный поток, как объект исследования обладает рядом особенностей, требующих особого внимания при планировании и проведении экспериментов. Можно выделить следующие особенности:

- пространственная распределённость объекта исследований;
- динамический характер поведения;
- невозможность получить полную информацию о состоянии транспортной сети путем проведения натуральных измерений. Имеющиеся в распоряжении исследователей технические средства позволяют проводить только выборочную оценку состояния транспортного потока на стационарных постах (выборочная оценка состояния отдельных сечений транспортной сети) или передвижными автомобилями – лабораториями (выборочная оценка отдельных маршрутов движения);
- ограниченность в проведении экспериментов по соображениям экономического характера и обеспечения безопасности движения.

В связи с указанными особенностями объекта исследований основным методом в изучения становится математическое моделирование транспортных потоков и в частности микроскопическое моделирование. В настоящее время имеются как проприетарные программные средства микроскопического моделирования дорожного движения (AIMSUN, VISSIM, CORSIM и др.), так и свободное программное обеспечение (MITSIM, SUMO).

В качестве среды разработки имитационной модели транспортной сети выбрана программа с открытым исходным кодом SUMO (Simulation of Urban MObility) разработанная в Институте транспортных систем немецкого аэрокосмического центра (Institute of Transportation Systems at the German Aerospace Center) в 2000 году. Основной задачей разработки программы была поддержка сообщества исследователей инструментом для реализации и проведения оценки собственных алгоритмов управления светофорными объектами или новые подходы к управлению транспортными потоками. С целью упрощения реализации исследователями собственных алгоритмов управления был разработан модуль TraCI, обеспечивающий управление моделью из внешних программ, написанных на языке программирования Python.

Процедура проведения имитационного моделирования (рис. 4.1) включает в себя три этапа – подготовку транспортной сети и сценария

моделирования, собственно имитацию, составление отчетов и обработку результатов.

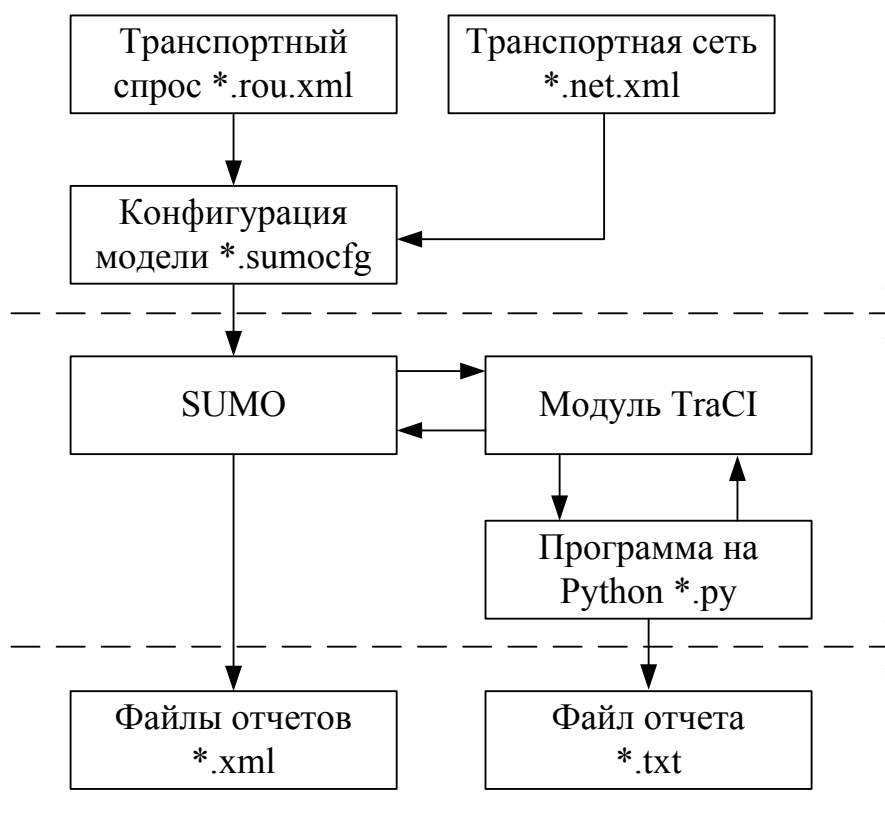


Рис. 4.1. Схема имитационного моделирования

Программа SUMO предоставляет пользователям несколько способов создания транспортных сетей – генерацию абстрактных сетей, конвертирование описания реальных сетей из файлов картографических систем (\*.shp, \*.xml и др.) и файлов описания транспортного графа. Последний способ предусматривает описание расположения транспортных узлов с указанием их типа (в том числе регулируемых) и задание транспортных связей между ними. При помощи программы netconvert производится конвертация файлов описания транспортного графа в файл транспортной сети \*.net.xml.

Транспортный спрос задается в виде отдельного файла с расширением \*.rou.xml, в котором приводится описание всех транспортных средств, входящих в сеть. Описание включает уникальный идентификатор транспортного средства, время и идентификатор транспортной связи, на которой он появится, параметры транспортного средства (желаемая скорость, максимальное ускорение и др.) и маршрут движения.

Для создания транспортного спроса имеется несколько программ, использующих файл транспортной сети и описание транспортного спроса в виде матриц корреспонденций, данных с детекторов транспорта или

интенсивности движения на входах и доли поворотных потоков на перекрестках. С учетом особенностей исследуемых методов управления, целесообразно задание транспортного спроса в виде интенсивности движения на входах в транспортную сеть и доли поворотных потоков на перекрестках.

Программа SUMO в зависимости от поставленной задачи позволяет создавать различные сценарии моделирования, описываемые в файле конфигурации \*.sumocfg. Сценарий содержит следующую информацию:

- файл транспортной сети;
- файл транспортного спроса;
- файлы дополнений (файлы, содержащие информацию о размещении детекторов транспорта, содержании файлов отчетов и др.);
- продолжительность моделирования;
- информацию о передаче управления моделированием модулю TraCI.

Указанный подход позволяет создавать различные сценарии моделирования для одной и той же комбинации транспортной сети и транспортного спроса, при этом функций по управлению светофорными объектами будут реализовываться внешними приложениями через модуль TraCI (спецификация функций TraCI приведена в приложении 13). Тем самым обеспечивается создание равных условий моделирования при сравнении различных методов управления транспортными потоками.

Непосредственно имитации движения транспортных средств выполняется программой sumogui. В случае использования модуля TraCI программа на языке Python выполняет запуск программы sumogui с указанием в качестве параметра запуска файл конфигурации и впоследствии пошаговое управление процессом моделирования. Модуль TraCI содержит необходимые функции для управления светофорными объектами и опроса состояния виртуальных детекторов транспорта.

Результаты имитационного моделирования в SUMO доступны в реальном времени через модуль TraCI, а по окончании моделирования могут быть сохранены в файлах отчетов в формате \*.xml. Для последующего анализа доступны переменные, описывающих состояния элементов транспортной сети, отдельных автомобилей, светофорных объектов и т.д. как в исходном, так и в агрегированном виде.

## 4.2. Калибровка микроскопической модели

В процессе разработки имитационной модели неизбежно возникают ошибки следующего рода [15]:

- ошибки спецификации (вызваны несовершенством математической модели движения транспортных средств);

– ошибки детализации (вызваны качеством представления транспортной сети);

– ошибки входных данных (связаны со сбором, вводом или редактированием натуральных данных).

В связи с неизбежностью возникновения ошибок при построении имитационных моделей, встает вопрос оценки их адекватности и необходимости повышения качества. Последовательность шагов повышения качества транспортной модели включает в себя верификацию, валидацию и калибровку модели.

Процесс верификации относится к исходным данным, участвующим в модели, в то время как термин «валидация» относится к проверке работоспособности непосредственно алгоритма расчета, т.е. к качеству построенных определяющих соотношений. Калибровка модели в свою очередь заключается в том, чтобы обеспечить воспроизведение типичного поведения реального транспортного потока при моделировании. В процессе калибровки параметры модели корректируются до достижения приемлемого (качественного и количественного) соответствия между модельными и натурными наблюдаемыми данными. Процедура калибровки модели носит итерационный характер и включает:

– сбор натуральных данных о параметрах транспортной сети, проведение выборочных наблюдений за состоянием транспортного потока;

– установку начальных оценок параметров модели, используя доступную информацию относительно геометрических и функциональных особенностей транспортных средств и транспортной сети;

– моделирование транспортного потока и извлечение информации о состоянии транспортной сети в контрольных сечениях;

– сравнение моделируемого и полученного в результате наблюдений состояний транспортного потока;

– если различия являются значимыми, то необходима модификация параметров модели с повторением процедуры калибровки;

– при не значимых различиях процесс калибровки заканчивается.

В общем случае выбор переменных, используемых для калибровки имитационной модели, во многом зависит от поставленной задачи моделирования и возможности проведения измерений в реальном транспортном потоке. Чаще всего при калибровке используются такие переменные как интенсивность движения, скорость движения, занятость контрольного сечения, время движения. Применительно к микроскопическим имитационным моделям калибровке подлежит распределение скоростей в транспортном потоке при фиксированной интенсивности движения (наблюдаемой в реальных условиях движения).

Процедуру калибровка модели (рис. 4.1) можно условно разделить на два этапа. На первом этапе производится выполнение моделирования и



просмотр анимации. Это облегчает оценку поведения транспортного средства и оценку его полного правдоподобия.

На этом этапе выявляются грубые ошибки в построении модели. При просмотре анимации следует в первую очередь обратить внимание на корректность соблюдения скоростного режима, смены полосы движения, выполнение перестроений на перегоне и на пересечениях.

На втором этапе проводится статистическая оценка адекватности модели по наблюдаемым параметрам по общеизвестным статистическим критериям. Мерой адекватности построенной модели могут служить средние абсолютные и относительные ошибки, нормализованные ошибки, критерии согласия [7, 15].



Рис. 4.1 Алгоритм калибровки микроскопических моделей транспортного потока

Как правило, количественная оценка адекватности транспортных моделей производится по следующим критериям [2]:

– средняя абсолютная ошибка (MAE) – количественная мера, оценки близости моделируемых  $x_i^{sim}$  и наблюдаемых  $x_i^{obs}$  значений:

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i^{sim} - x_i^{obs}| = \frac{1}{n} \sum_{i=1}^n |e_i|.$$

– нормализованная ошибка среднего квадратичного (RMSNE):

$$RMSNE = \sqrt{\frac{1}{n} \sum_{i=1}^n \left( \frac{x_i^{sim} - x_i^{obs}}{x_i^{obs}} \right)^2};$$

– абсолютный процент ошибки (MAPE):

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{x_i^{sim} - x_i^{obs}}{x_i^{obs}} \right|.$$

Как видим, оценка адекватности транспортных моделей производится по средним значениям наблюдаемых параметров. Подобный подход оправдан применительно к макроскопическим моделям (описывающим состояние транспортного потока средними значениями скорости, интенсивности и плотности движения) и недопустим применительно к микроскопическим моделям (определяющим состояние транспортного потока через скорости отдельных автомобилей).

На работу городской транспортной сети в значительной степени оказывает влияние не только средние значения скоростей и интенсивностей движения, но и характер их вариаций. Например, модели задержки транспортных средств на регулируемых пересечениях включают не только среднее значение интенсивности прибытия транспортных средств, но и ее дисперсию. В связи с этим, для микроскопических моделей актуальным является оценка соответствия наблюдаемого в модели распределения калибруемого параметра его распределению в реальном транспортном потоке по критерию согласия. Указанные критерии выполняют проверку гипотезы об однородности выборок, произведенные из генеральных совокупностей с неизвестными теоретическими функциями распределения  $F_1(x)$  и  $F_2(x)$ . Проверяемая нулевая гипотеза вида  $H_0 : F_1(x) = F_2(x)$  против конкурирующей  $H_1 : F_1(x) \neq F_2(x)$ .

Адекватность модели в этом случае можно оценивать при помощи критериев согласия  $\chi^2$  Пирсона ( $\chi$ -квадрат) или Колмогорова-Смирнова. Последний критерий более предпочтителен, так как может быть применен

для выборок, объем которых меньше, чем минимально допустимый для корректного использования критерия  $\chi^2$ .

Критерий Колмогорова-Смирнова использует ту же самую идею, что и критерий Колмогорова. Однако различие заключается в том, что в критерии Колмогорова сравнивается эмпирическая функция распределения с теоретической, а в критерии Колмогорова-Смирнова сравниваются две эмпирические функции распределения.

Статистика критерия Колмогорова-Смирнова имеет вид:

$$\lambda' = \sqrt{\frac{n_1 n_2}{n_1 + n_2}} \cdot \max |F_{n_1}(x) - F_{n_2}(x)|,$$

где  $F_{n_1}(x)$  и  $F_{n_2}(x)$  – эмпирические функции распределения, построенные по двум выборкам с объемами  $n_1$  и  $n_2$ .

Гипотеза  $H_0$  отвергается, если фактически наблюдаемое значение статистики  $\lambda'$  больше критического  $\lambda'_{кр}$ , т.е.  $\lambda' > \lambda'_{кр}$ , и принимается в противном случае.

При малых объемах выборок ( $n_1, n_2 \leq 20$ ) критические значения  $\lambda'_{кр}$  для заданных уровней значимости критерия можно найти в специальных таблицах. При  $n_1, n_2 \rightarrow \infty$  (а практически при  $n_1, n_2 \geq 50$ ) распределение статистики  $\lambda'$  сводится к распределению Колмогорова для статистики  $\lambda$ . В этом случае гипотеза  $H_0$  отвергается на уровне значимости  $\alpha$ , если фактически наблюдаемое значение  $\lambda'$  больше критического  $\lambda_\alpha$ , т.е.  $\lambda' > \lambda_\alpha$ , и принимается в противном случае.[7]

Выбор управляемых переменных, настройка которых обеспечит требуемое качество модели, зависит от специфики конкретной математической модели движения, используемой в программной среде. В SUMO используется модифицированная дискретная по времени и непрерывная в пространстве модель движения Краусса [98]. Модель основана на определении интервала между ведомым транспортным средством и лидером, необходимым для безопасной остановки:

$$v_{safe}(t) = -\tau \cdot b + \sqrt{(\tau \cdot b)^2 + v_{leader}(t-1)^2 + 2 \cdot b \cdot g_{leader}(t-1)},$$

где  $v_{safe}(t)$  – безопасная скорость в момент времени  $t$ , м/с;  $\tau$  – время реакции водителя ведомого автомобиля, с;  $b$  – максимальное замедление, м/с<sup>2</sup>;  $v_{leader}(t)$  – скорость лидера во время  $t$ , м/с;  $g_{leader}(t)$  – дистанция между передней частью ведомого автомобиля и задним бампером лидера во время  $t$ , м.

При ограничениях на значения желаемой скорости:

$$v_{des}(t) = \min \{v_{safe}(t), v(t-1) + a, v_{max}\},$$

где  $v_{des}(t)$  – желаемая скорость, м/с;  $v(t)$  – текущая скорость, м/с;  $a$  – максимальное ускорение, м/с<sup>2</sup>;  $v_{max}$  – максимальная скорость, м/с.

Одной из основных особенностей модели Краусса является предположение, что водитель не обязательно должен развивать желаемую скорость, что добавляет важные особенности в поведение водителей. Например, случайное отличие скоростей автомобилей от желаемых приводит к спонтанному образованию заторов и медленному старту автомобилей на светофорах, характерному для реальных условий. Данная особенность поведения водителей учтена путем включения в модель стохастического замедления:

$$v(t) = \max\{0, v_{des}(t) - r \cdot a \cdot \varepsilon\},$$

где  $r$  – случайное число между 0 и 1;  $\varepsilon$  – степень индивидуальности (несовершенства) водителя автомобиля, принимающее значение между 0 и 1;  $v(t)$  – конечная скорость автомобиля в момент времени  $t$ , м/с.

При разработке программы в исходную модель Краусса было внесено два изменения. Первое заключалось в использовании линейной функции ускорения, обеспечивающей его уменьшение с ростом скорости:

$$a(v) = a \left( 1 - \frac{v}{v_{max}} \right).$$

Второе – в уменьшении индивидуальности (несовершенства) водителей при разгоне на малых скоростях:

$$v(t) = \max\{0, v_{dawdle, new}(t)\},$$

где

$$v_{dawdle, new}(t) = \begin{cases} v_{des}(t) \cdot \varepsilon \cdot r & \text{если } v_{des}(t) < a(v_{des}(t)); \\ v_{veh}(t) - \varepsilon \cdot r \cdot a v_{veh}(t) & \text{в остальных случаях.} \end{cases}$$

В программе имеется возможность настройки модели движения двумя способами. Первый заключается в задании различных максимальных желаемых скоростей для отдельных потоков на этапе задания транспортного спроса, второй – в модификации максимально желаемой скорости входящих в транспортную сеть автомобилей через интерфейс модуля *Traci*.

Последний способ является более универсальным и приемлемым для решения разнообразных задач моделирования. Модуль *Traci* позволяет устанавливать желаемую скорость движения  $v_{max}$  автомобилей, вновь появляющихся в транспортной сети.

Сбор исходных данных для калибровки микроскопической модели производился путем измерения скоростей движения отдельных автомобилей в условиях городского движения на ул. Окружная г. Пенза. Измерение

скоростей и определение объема выборки производились в соответствии с известной методикой [5]. Распределение скоростей движения статистически достоверно (с уровнем значимости 0,967) может быть описано лог-нормальным законом распределения со следующими параметрами  $m=4,56$  и  $\sigma=0,031$  (рис. 4.2).

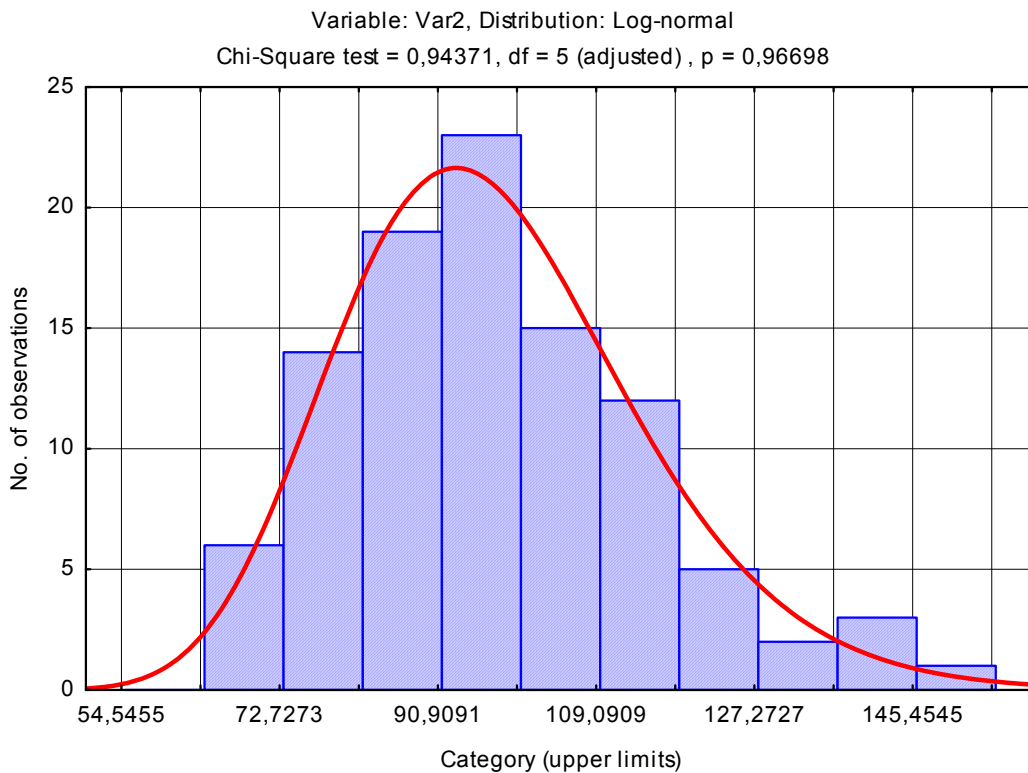


Рис. 4.2. Распределение скоростей движения на ул. Окружная г. Пенза

Калибровка модели движения транспортных средств производилось на фрагменте транспортной сети, воспроизводящей условия движения исследуемой улице. Для регистрации скоростного режима в модели транспортной сети были установлены виртуальные детекторы транспорта.

Модификация максимально движения производилась функцией *SpeedModification()*, путем запроса списка всех вошедших в сеть автомобилей и установке максимальную желаемую скорость функцией *traci.vehicle.setMaxSpeed()*. Значения максимальной желательной скорости генерировалась при помощи стандартного модуля *random* языка *Python* по логнормальному закону распределения функцией *random.lognormvariate(mu, sigma)*. Результаты измерений скоростей движения при моделировании представлены в табл. 4.1.

Таблица 4.1

Сводная таблица распределения частот скоростей транспортного потока

Номер интервала	Интервалы скоростей, км/ч	Частоты	
		В реальных условиях	При моделировании
1.	40-50	0	1
2.	50-60	0	1
3.	60-70	4	2
4.	70-80	11	11
5.	80-90	19	17
6.	90-100	27	34
7.	100-110	17	21
8.	110-120	12	5
9.	120-130	5	8
10.	130-140	3	0
11.	140-150	2	0

Обозначим  $n_{i_1}^{нак}$  и  $n_{i_2}^{нак}$  – накопленные частоты выборок скоростей в реальных условиях и при моделировании, а через  $F_{n_1}(x_i) = n_{i_1}^{нак} / n_1$  и  $F_{n_2}(x_i) = n_{i_2}^{нак} / n_2$  – значения их эмпирических функций распределения соответственно. Обработанные результаты расчетов сведены в табл. 4.2.

Из последнего столбца табл. 4.2 определим  $\max |F_{n_1}(x_i) - F_{n_2}(x_i)| = 0,09$ .  
Значение критерия Колмогорова-Смирнова равно:

$$\lambda' = \sqrt{\frac{100 \cdot 100}{100 + 100}} \cdot 0,09 = 0,636$$

Таблица 4.2

$x_i$	$n_{i_1}^{нак}$	$n_{i_2}^{нак}$	$F_{n_1}(x_i)$	$F_{n_2}(x_i)$	$ F_{n_1}(x_i) - F_{n_2}(x_i) $
40	0	1	0	0,01	0,01
50	0	2	0	0,02	0,02
60	4	4	0,04	0,04	0
70	15	15	0,15	0,15	0
80	34	32	0,34	0,32	0,02
90	61	66	0,61	0,66	0,05
100	78	87	0,78	0,87	<b>0,09</b>
110	90	92	0,9	0,92	0,02
120	95	100	0,95	1	0,05
130	98	100	0,98	1	0,02
140	100	100	1	1	0

Критическое значение критерия для уровня значимости 0,05 равно 1,36 [6]. С учетом, что неравенство  $\lambda' > \lambda_{0,05}$  ложно, принимается гипотеза о согласии распределения скоростей движения в реальных условиях и при моделировании, на основании чего можно сделать вывод об адекватности микроскопической модели движения.

### 4.3. Оценка эффективности локального управления светофорным объектом

Для объективной оценки эффективности предложенного метода локального управления работой светофорного объекта в программе SUMO создана микроскопическая имитационная модель пересечения двух городских улиц – горизонтальной и вертикальной (рис. 4.3).

Горизонтальная улица (связи 1 и 3) имеет на подходе к пересечению по 2 полосы движения в каждом направлении, вертикальная (связи 2 и 4) – по 1 полосе. Для выполнения левых поворотов выделены дополнительные полосы протяженностью 300 м по горизонтальной улице (связи 1.300 и 3.300) и 90 м по вертикальной (связи 2.300 и 4.300). Протяженность подходов к перекрестку горизонтальной улицы составила 600 м, вертикальной – 400 м. Содержание файлов описания транспортной сети в формате xml программы SUMO, использованных для создания модели локального пересечения приведены в прил. 1.

На моделируемом пересечении организован бесконфликтный пропуск пешеходных и транспортных средств (прил. 1):

- в первой фазе производится пропуск потоков прямого направления и правоповоротных потоков по горизонтальной улице;
- во второй фазе пропускаются левоповоротные потоки с горизонтальной улицы;
- в третьей фазе производится пропуск потоков прямого направления и правоповоротных потоков по вертикальной улице;
- в четвертой фазе пропускаются левоповоротные потоки с вертикальной улицы;
- в пятой фазе производится пропуск пешеходов.

Транспортный спрос в модели задан в файле routes.rou.xml, сгенерированном программой jtrrouter.exe на основе интенсивности движения на входах в транспортную сеть (файл flows.flows.xml) и доли поворотных потоков на перекрестке (файл turns.turns.xml).

Интенсивность на входах в транспортную сеть (связи 1, 2, 3 и 4) в процессе моделирования варьировалась по трем периодам (рис. 4.4).

В первом периоде (с 0 по 30 мин) интенсивность движения составляла около 300 авт/ч на полосу движения по всем подходам к пересечению.

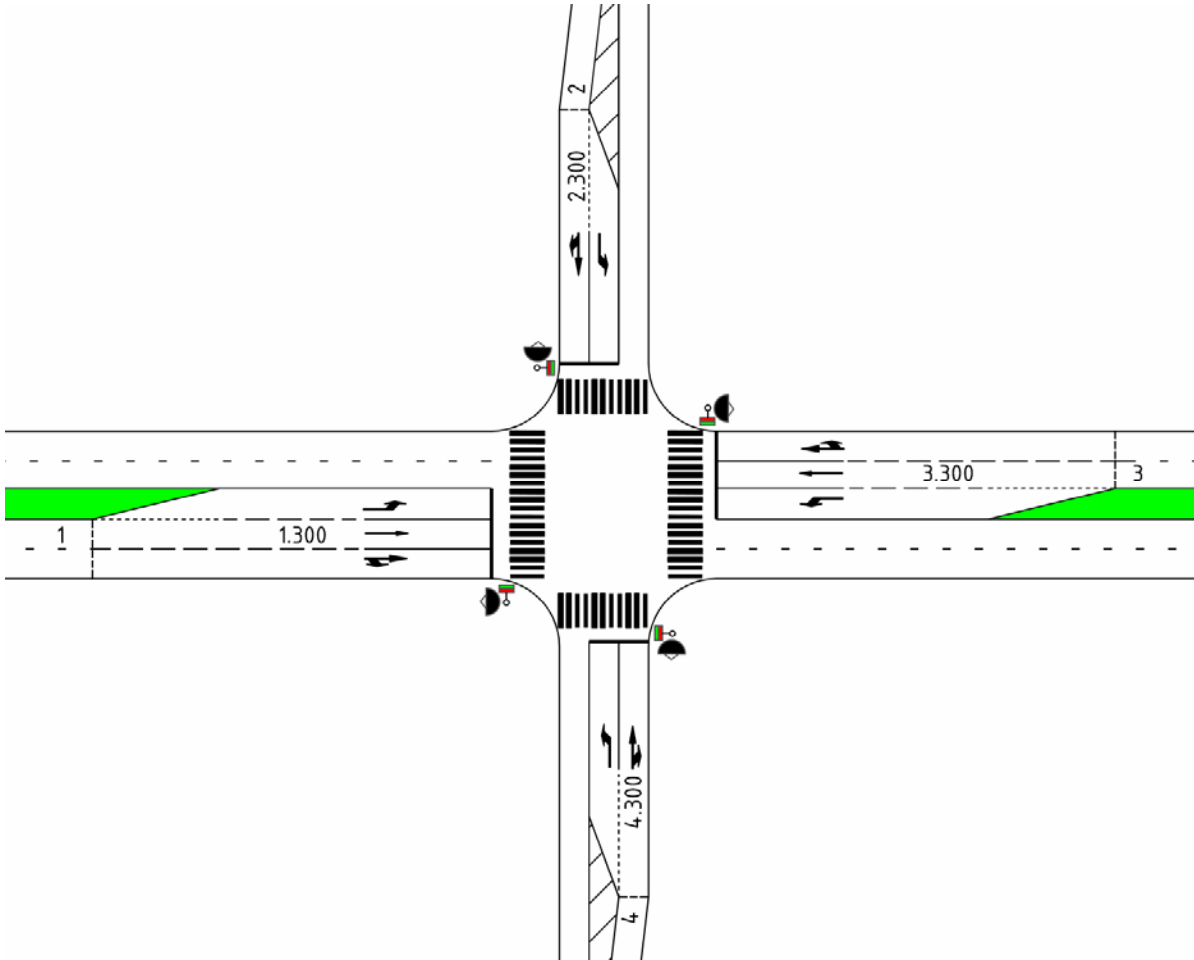


Рис. 4.3. Схема перекрестка

Во втором периоде (с 30 по 90 мин.) интенсивность по горизонтальной улице (связи 1 и 3) увеличена до 500–600 авт/ч на полосу движения. В третьем периоде моделирования (с 90 по 120 мин) по горизонтальной улице интенсивность движения уменьшена до 400 авт/ч на полосу движения, а по вертикальной (связи 2 и 4) – увеличена до 500–600 авт/ч на полосу движения.

Доля поворотных потоков по вертикальной улице за весь период моделирования постоянна и составляла:

- правоповоротные потоки 20 %;
- левоповоротные потоки 20 %;
- прямое направление 60 %.

По горизонтальной улице доля поворотных потоков варьировалась в следующих пределах:

- правоповоротные потоки 22,7 ... 10 %;
- левоповоротные потоки 21,4 ... 10 %;
- прямое направление 77,1 ... 56,9 %.



Содержание исходных файлов, использованных для генерации транспортного спроса, приведено в прил. 2.

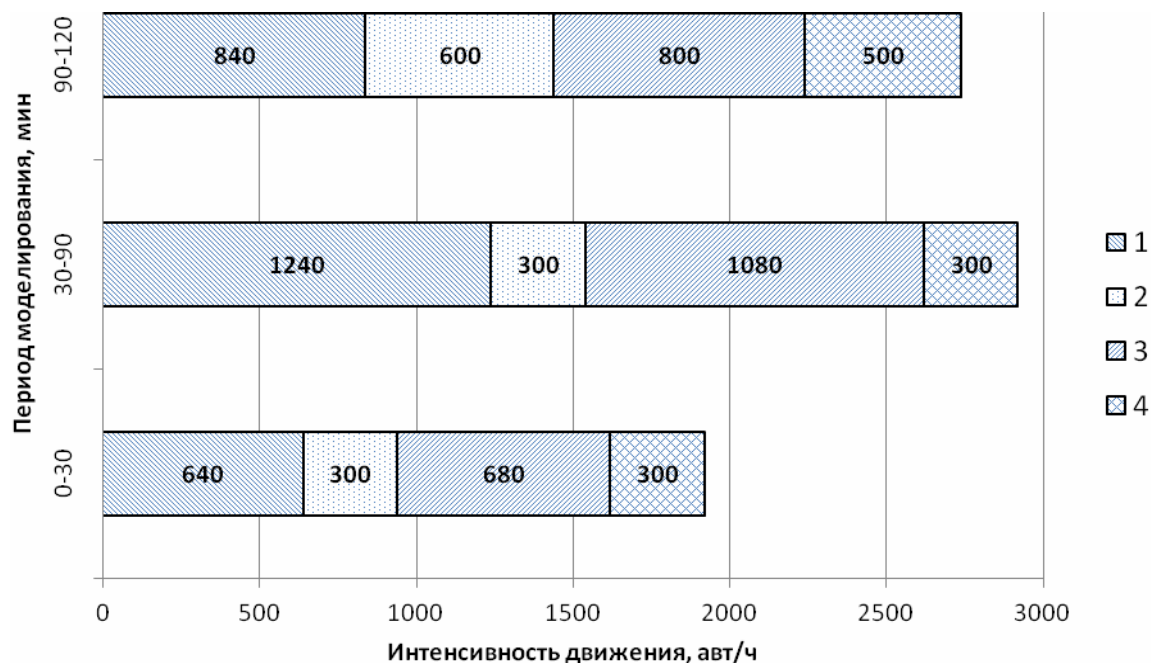


Рис. 4.4. Динамика транспортного спроса

Управление процессом моделирования и работой виртуального дорожного контроллера производилось программой LocalControll.py, написанной на языке программирования Python. Листинг указанной программы приведен в прил. 3. Для взаимодействия с программой SUMO использовался модуль TraCI, обеспечивающий:

- модификацию желаемой скорости движения всех входящих в сеть автомобилей по логнормальному закону распределения с параметрами, полученными в п. 4.2;
- измерение интенсивности движения при помощи виртуальных детекторов транспорта, имитирующих работу индуктивных петлевых детекторов;
- установку рассчитанных режимов работы светофорного объекта в имитационной модели.

В качестве базового метода управления принята традиционная методика расчета режима работы светофорного объекта по формулам (3.3)–(3.4). В конце каждого цикла регулирования производился перерасчет длительности фаз на основании данных об интенсивности движения, измеренных виртуальными детекторами транспорта.

Измерение интенсивности движения и последующее сглаживание полученных результатов производилось в соответствии с рекомендациями, приведенными в работе [10].

Предлагаемое управление предусматривало расчет длительности фаз в конце каждого цикла регулирования. Длительности фаз определялись путем решения ранее предложенной в п. 3.2 задачи оптимизации. Оценка продолжительности задержки транспортных средств на горизонте прогнозирования, входящей в целевую функцию управления (3.17), проводилась по формулам (3.8)–(3.15). Оценка остаточной очереди выполнялась на основе измерений состояния транспортного потока виртуальными детекторами транспорта по формуле (3.20). Штраф за превышение фронтом очереди максимального предустановленного значения назначался путем умножения величины задержки на весовой коэффициент  $w = 6$ . По результатам моделирования формировался файл отчетов aggregated\_900.xml, содержащий агрегированную по 15 минутным интервалам информацию о состоянии транспортных связей (прил. 5 и 6).

Суммарная задержка транспортных средств по всем связям при базовом и предлагаемом управлении приведена на рис. 4.5.

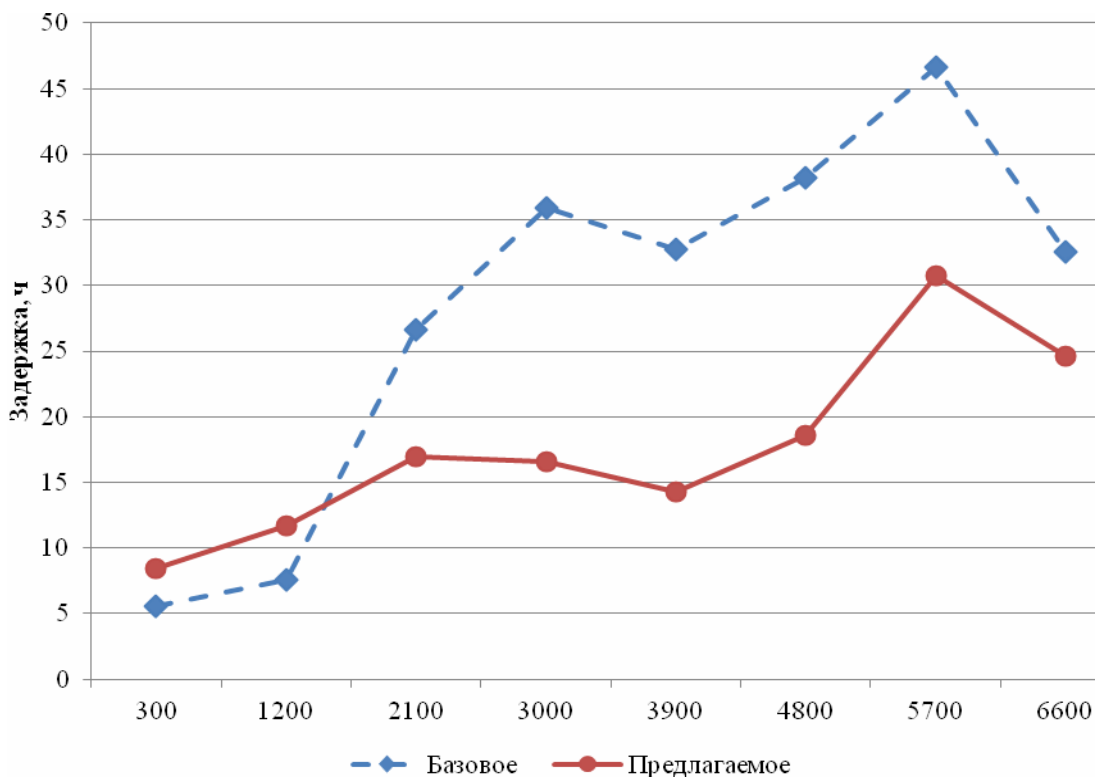


Рис. 4.5. Задержка транспортных средств

При интенсивности движения 300 авт/ч на полосу (начальный период моделирования с 0 по 30 мин) использование базового управления обеспе-

чивает более эффективное функционирование пересечения, выраженное в меньшей на 35 % величиной задержки сравнении с предлагаемым методом управления. В дальнейшем при увеличении интенсивности движения наблюдается преимущество предложенного метода управления над базовым методом, выражаемое в снижении задержки транспортных средств в 1,3–2,3 раза и увеличении количества вошедших в моделируемую транспортную сеть автомобилей (рис. 4.6) в 1,21–1,53 раза в сравнении с базовым методом.

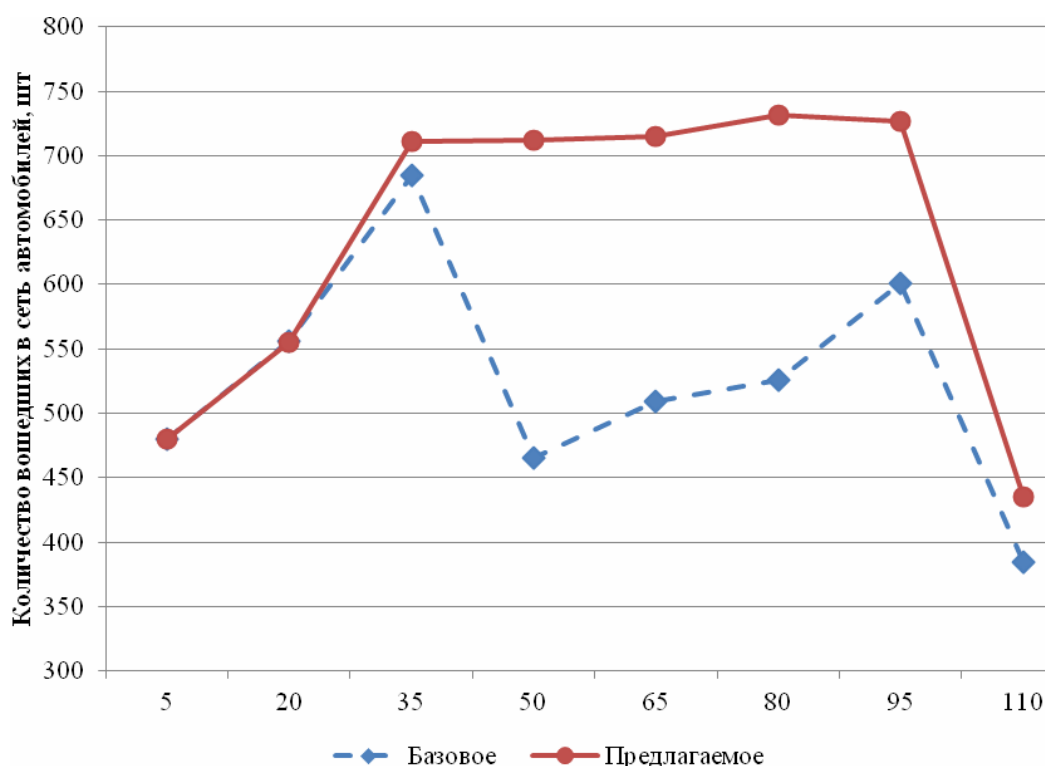


Рис. 4.6. Количество вошедших в транспортную сеть транспортных средств

В начальный период моделирования очередь транспортных средств на регулируемом пересечении локализовалась в пределах связей 1.300, 2.300, 3.300 и 4.300. При последующем увеличении интенсивности движения длина очередей возросла и распространилась на связи 1, 2, 3 и 4. Рассмотрим динамику плотности транспортного потока и времени движения на указанных транспортных связях (рис. 4.7–4.14).

При использовании базового управления на связях 1 и 3 горизонтальной улицы с 30 по 50 минуту моделирования происходило увеличение плотности транспортного потока и в результате блокирование связей. На вертикальной улице в указанный период увеличение длины очередей, и соответственно повышения плотности транспортного потока, не наблюдалось. Предпосылки к образованию транспортного затора на вертикальной улице в виде повышения плотности транспортного потока выше

критического значения  $\rho_{crit} = 39 - 46$  авт/км наблюдаются в третий период моделирования (с 90 мин) при увеличении интенсивности движения до 500–600 авт/ч.

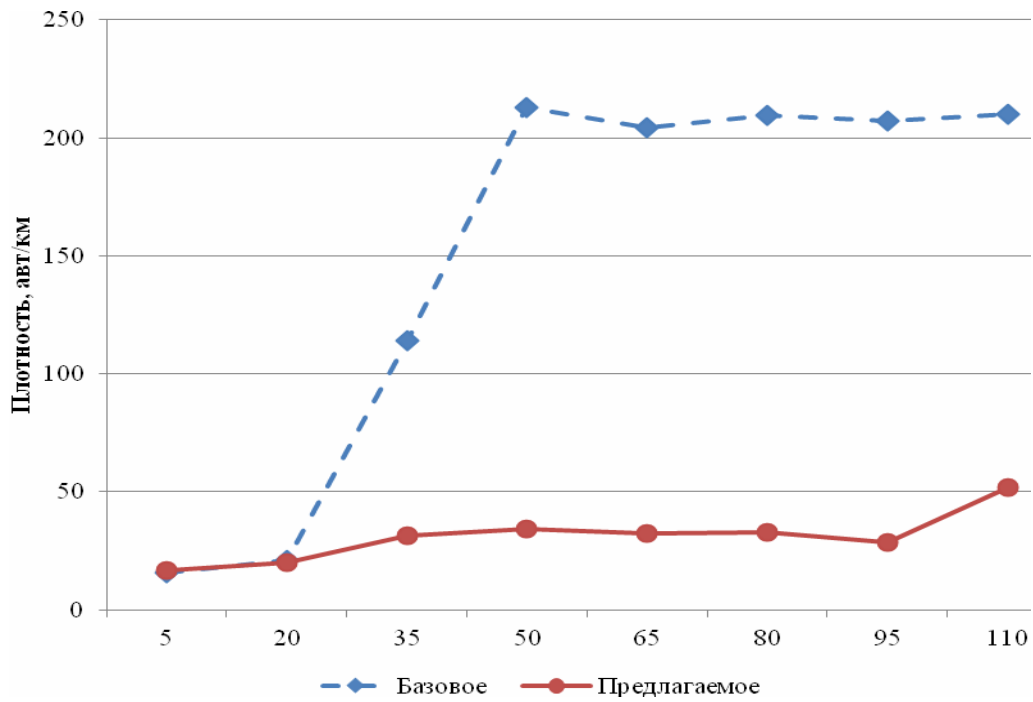


Рис. 4.7. Плотность транспортного потока на связи 1

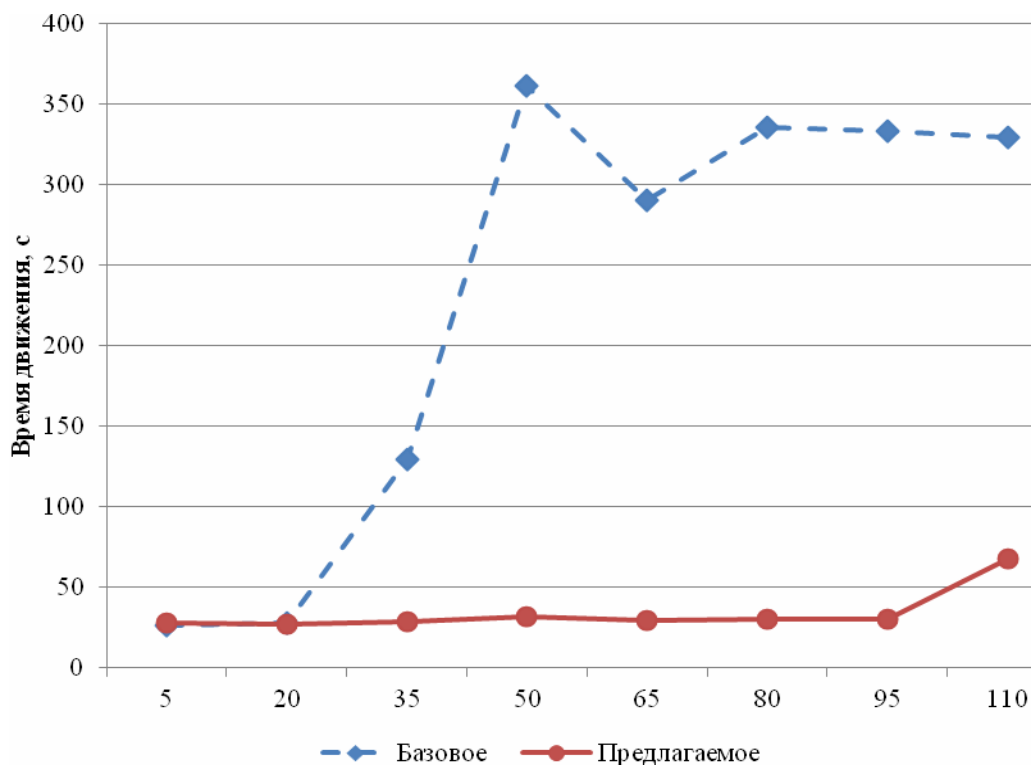


Рис. 4.8. Время движения по связи 1

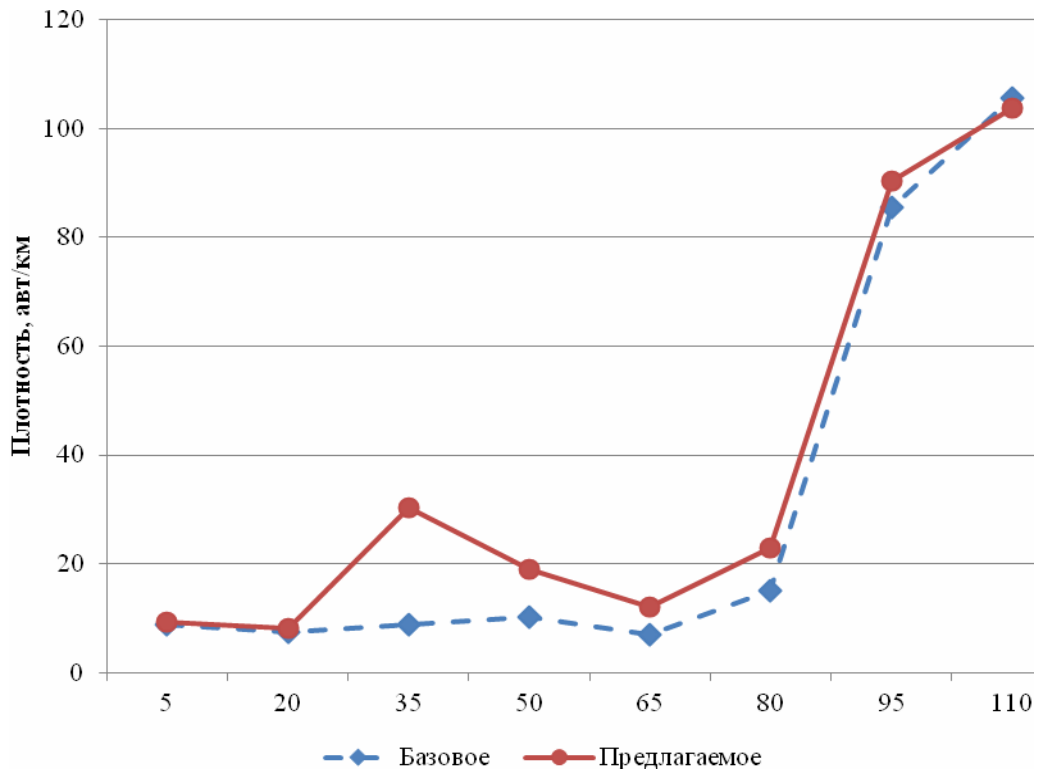


Рис. 4.9. Плотность транспортного потока на связи 2

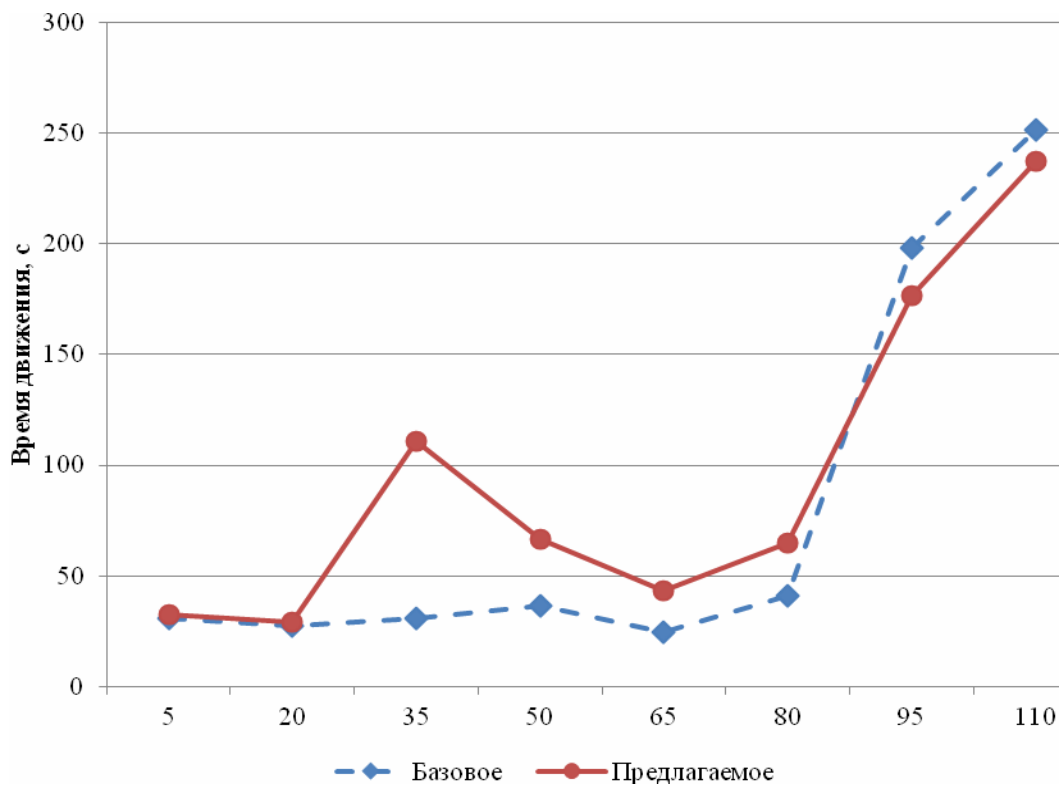


Рис. 4.10. Время движения по связи 2

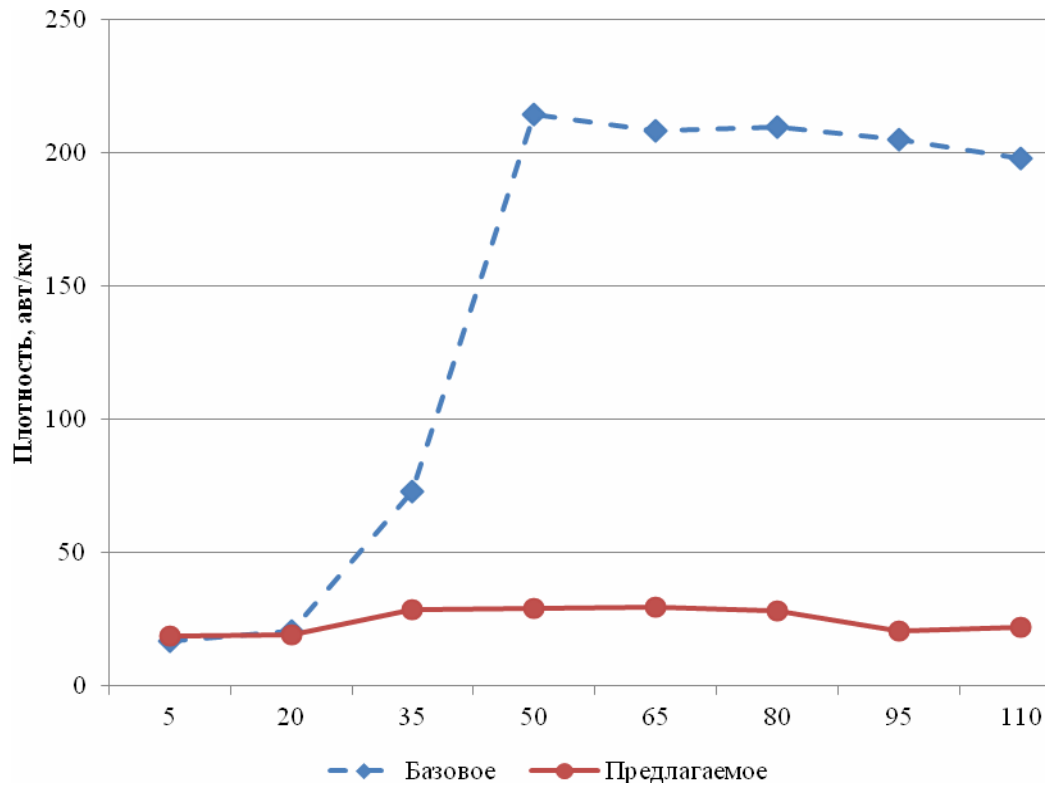


Рис. 4.11. Плотность транспортного потока на связи 3

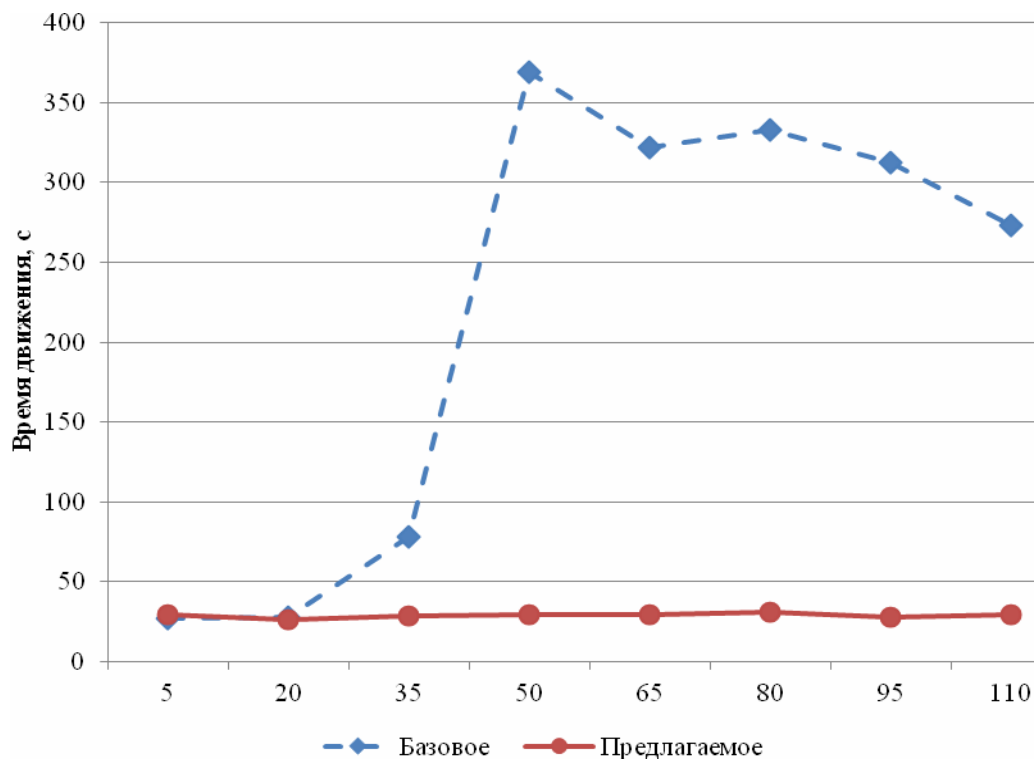


Рис. 4.12. Время движения по связи 3

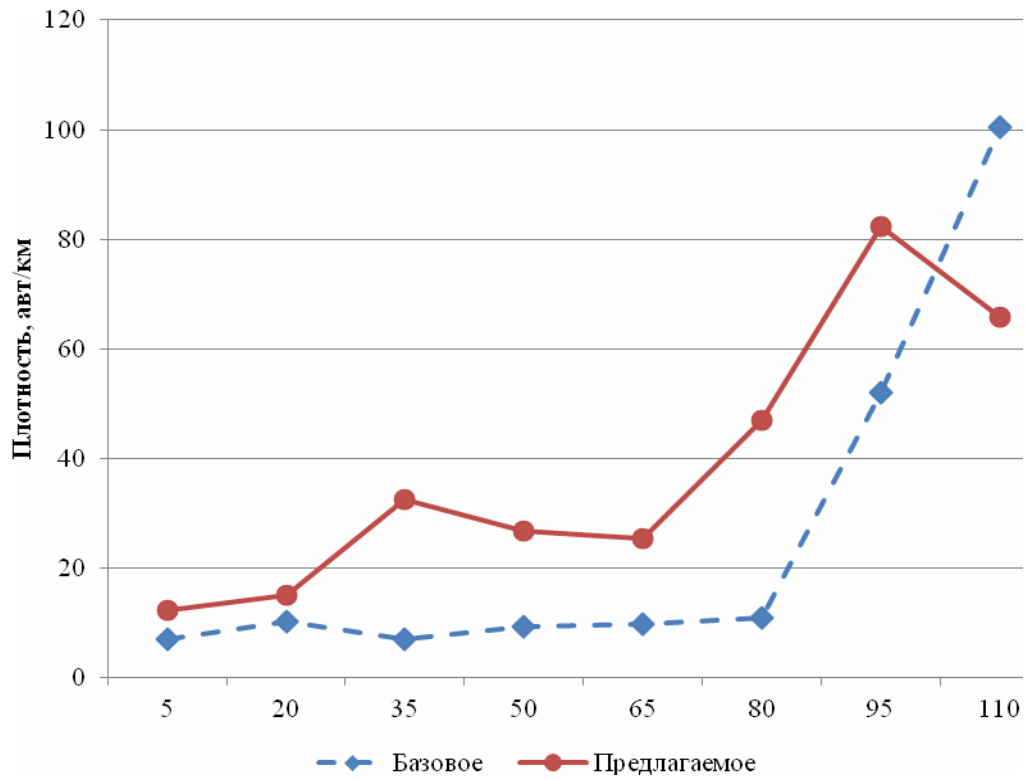


Рис. 4.13. Плотность транспортного потока на связи 4

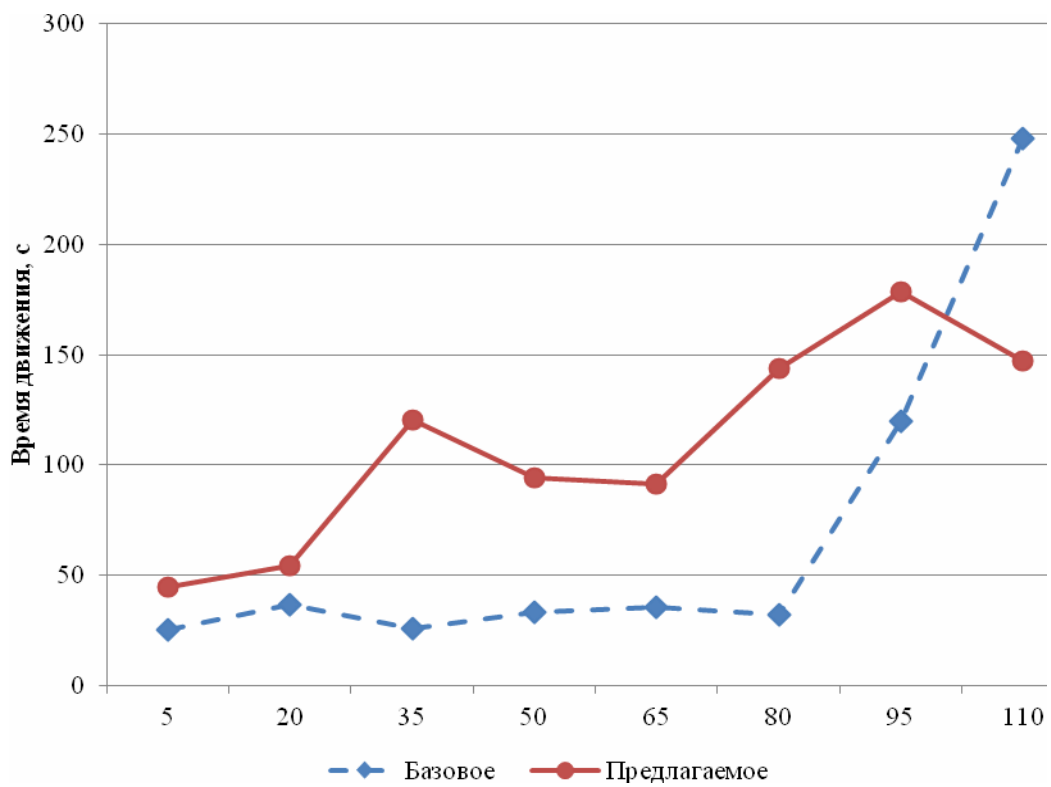


Рис. 4.14. Время движения по связи 4

Предлагаемый метод управления минимизирует взвешенную задержку транспортных средств, что ожидаемо, обеспечивает более благоприятные условия движения на направлениях с большим транспортным спросом. Так за период моделирования плотность транспортного потока на связях 1 и 4 горизонтальной улицы практически не превышала критического значения. На связях 2 и 4 вертикальной улицы плотность транспортного потока сопоставима со значениями, полученными при базовом управлении, при этом превышение плотностью значения  $\rho_{crit}$  наблюдалось только в третий период моделирования.

Результаты моделирования использования предложенного метода локального управления показали его работоспособность и эффективность в сравнении с традиционным методом расчета режимов работы светофорных объектов.

#### 4.4. Оценка эффективности сетевого управления светофорными объектами методом балансировки объемов движения

Для объективной оценки эффективности предложенных методов управления работой светофорных объектов, в программе SUMO создана микроскопическая имитационная модель транспортной сети, включающая 5 регулируемых пересечений (рис. 4.15). Исходные файлы описания узлов и связей транспортной сети приведены в прил. 7. Схема пофазного разъезда транспортных средств приведена на рис. 4.16. На пересечениях *A*, *B* и *C* пропуск автомобилей производится в четыре фазы (I-IV фаза), а на пересечении *D* предполагается наличие пятой, выделенной пешеходной фазы продолжительностью 20 с.

Интенсивность движения на входах в транспортную сеть представлена на рис. 4.15 и в прил. 8. Распределение потоков по направлениям движения в пределах перекрестков в направлении север – юг: прямо – 85 %, направо – 10 %, налево – 5 %; в направлении запад – восток: прямо – 55 %, направо – 35 %, налево – 10 %.

В качестве базового алгоритма управления принят общеизвестный метод адаптивного управления – алгоритм поиска разрыва в потоке (в зарубежной литературе именуемый «actuated traffic control»). Фрагмент файла описания транспортной сети с указанным методом управления приведен в прил. 9.

Для реализации, предложенной в п. 3.3 методики расчета режимов работы светофорных объектов методом балансировки объемов движения, разработана программа на языке программирования Python (прил.10). Рассчитанные указанной программой режимы работы светофорных



объектов устанавливались в режиме предрасчитанного управления в модели сети.

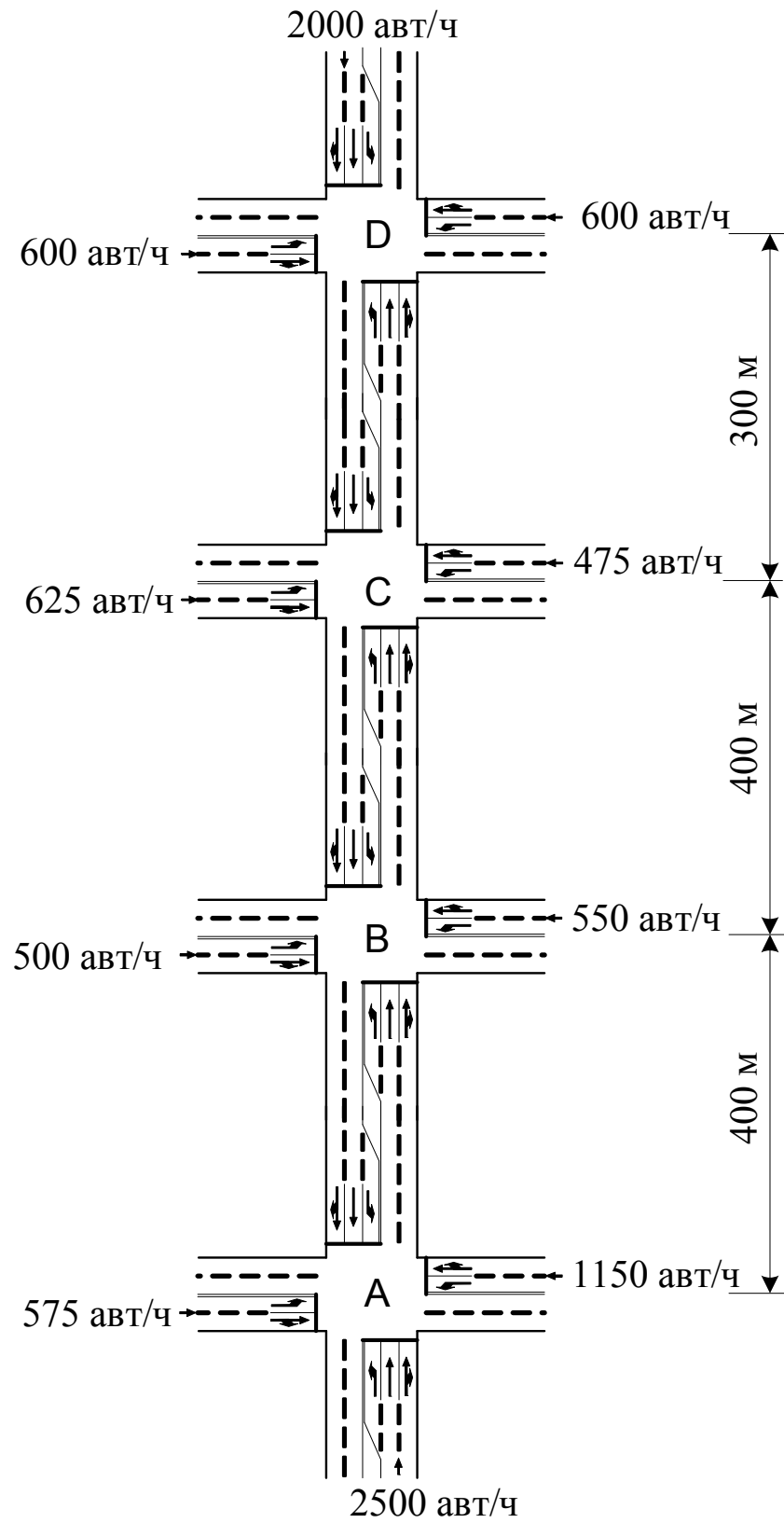


Рис. 4.15. Схема транспортной сети со светофорным регулированием

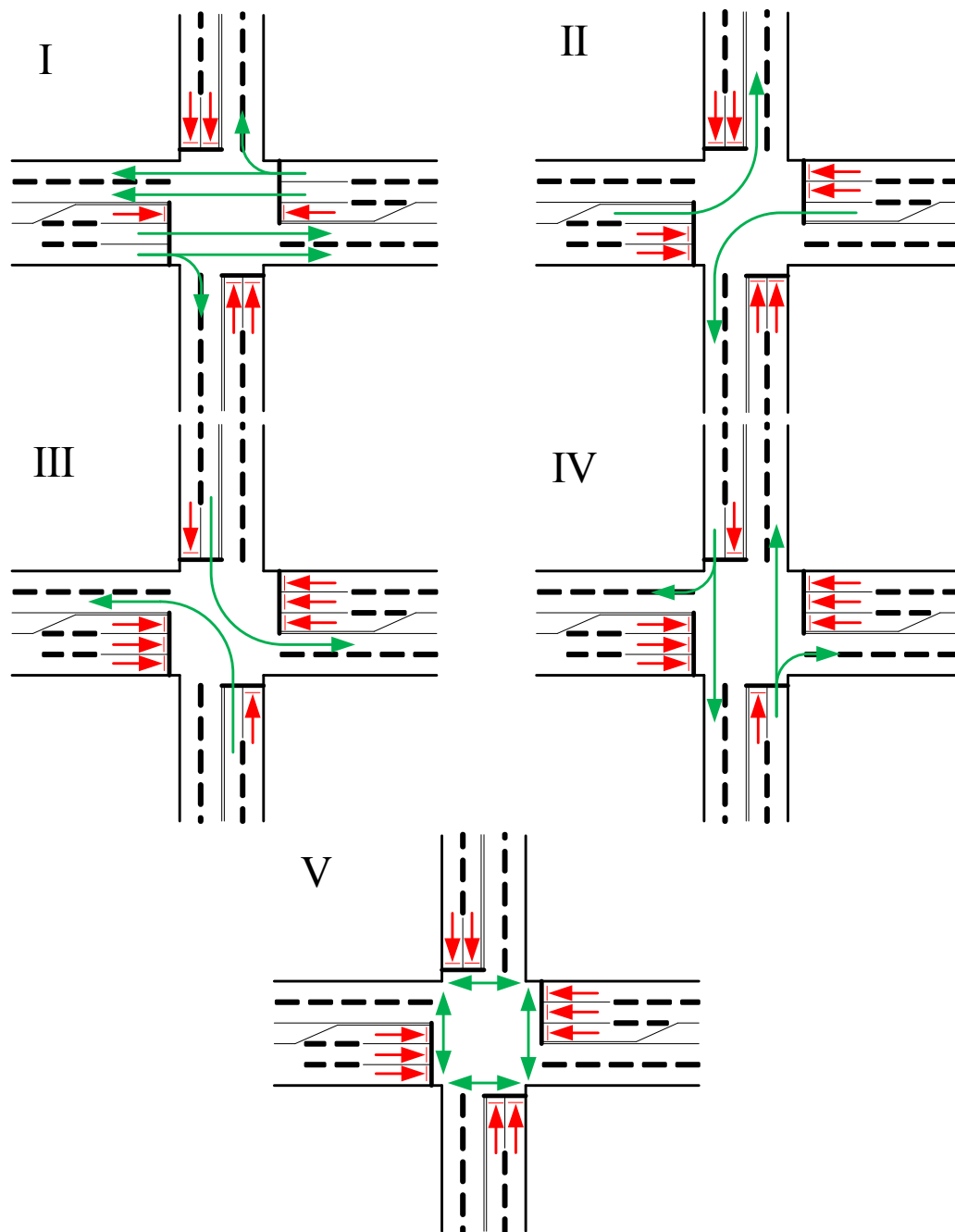


Рис. 4.16. Фазы движения транспортных средств

Результаты моделирования при базовом методе управления и предложенном приведены в прил. 11 и 12.

При базовом методе управления наличие выделенной пешеходной фазы на пересечении *D* инициирует образование транспортного затора на перегоне *C–D*. В прошествии 20-25 минут затор принимает общесетевой характер, блокируя связь *B–C*, а впоследствии и весь маршрут движения *A–B–C–D*. Плотность транспортного потока в зоне затора колебалась в пределах 205–100 авт/км в расчете на одну полосу движения (прерывистая линия на рис. 4.17).

При установке режима работы светофоров, рассчитанного по формулам (3.29)–(3.32) образование затора не происходило, при этом наблюдались незначительные колебания плотности транспортного потока в районе 20 авт/км в расчете на одну полосу движения (сплошная линия на рис. 4.17). Благодаря этому суммарная задержка транспортных средств на перегонах маршрута движения  $A-B-C-D$  сократилась в 2,5–4 раза (рис. 4.18).

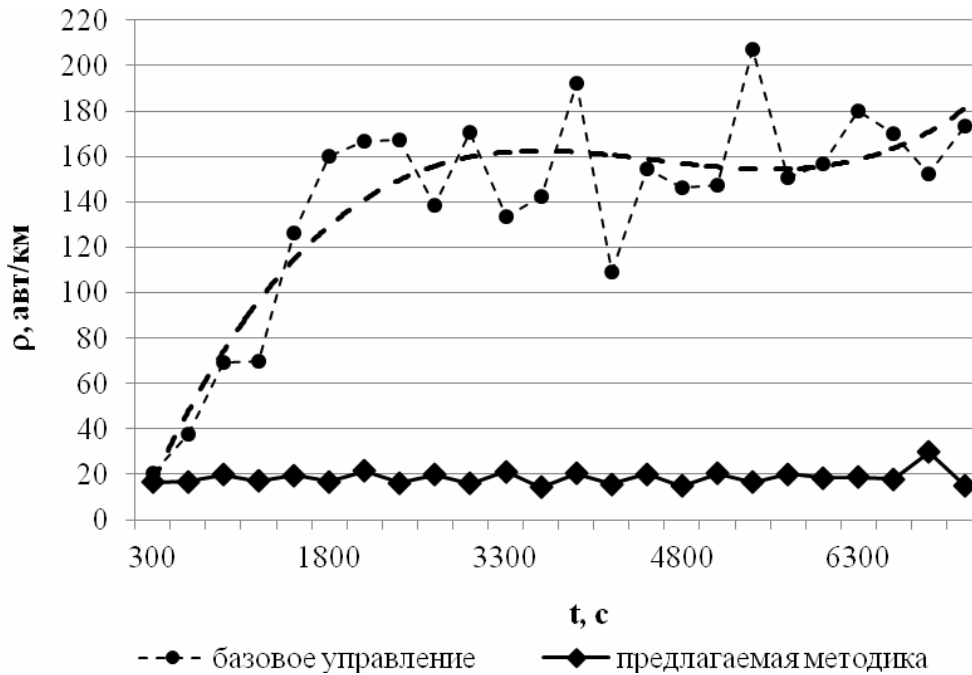


Рис. 4.17. Динамика плотности транспортного потока на связи  $B-C$

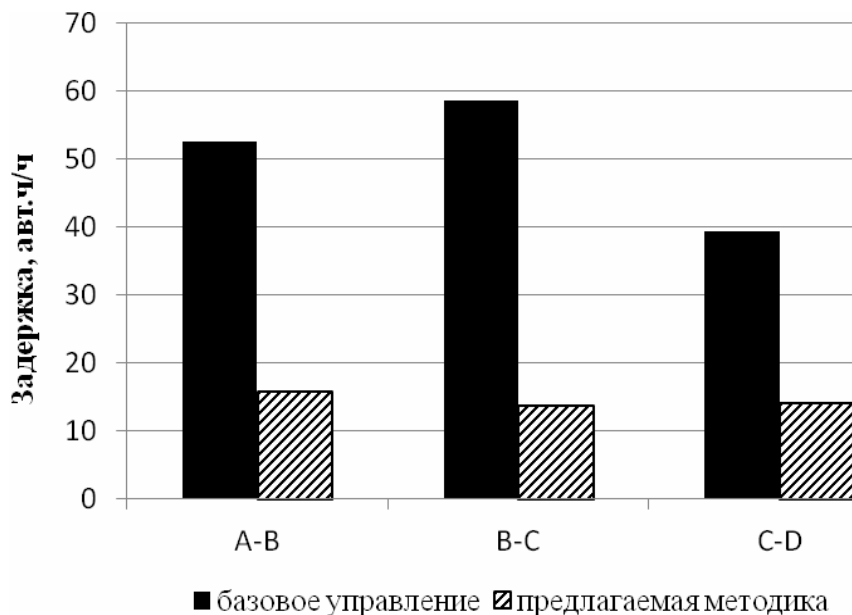


Рис. 4.18. Суммарная задержка на маршруте движения  $A-B-C-D$

Среднее время движения транспортных при базовом способе управления составила 234,63 с, а при установке режимов работы светофоров, рассчитанных по предлагаемой методике – 211,67 с. Среднее время ожидания разрешающего сигнала соответственно уменьшилось со 118,49 с до 101,96 с.

Как видим, использование предложенной методики позволяет не допустить образование общесетевого затора, уменьшить время движения в сети на 9,79 % и сократить среднюю задержку на 13,95 % в сравнении с алгоритмом адаптивного управления по поиску разрыва в потоке.

## ЗАКЛЮЧЕНИЕ

Анализ публикаций и существующего методического обеспечения в области управления светофорными объектами позволяет сделать следующие выводы:

- основным методом управления транспортными потоками в условиях городского движения является управление путем остановки транспортных средств. Управление реализуется при помощи светофорных объектов путем установки параметров многомерного вектора управления, компонентами которого являются длительность цикла регулирования, разбиение его эффективной длительности между фазами и временные сдвиги включения фаз;

- в условиях насыщенного движения системы с временно-зависимым управлением не в состоянии обеспечить должный уровень транспортного обслуживания участников дорожного движения;

- системы с адаптивным управлением выполняют как оптимизацию сигнальных планов, минимизируя функцию потерь, так и эвристическое управление на основе базы правил. В последнем случае используются как методы теории автоматического управления, так и эвристические алгоритмы, основанные на использовании функциональных зависимостей между параметрами, описывающими условия движения и сигнальными планами. Следует отметить, что эвристическое управление требует тщательной настройки в лабораторных условиях на этапе проектирования и, как правило, не предусматривает внесения изменений в контур управления в полевых условиях;

- системы оптимального управления транспортными потоками, как правило, используют метод «катящегося горизонта». Использование указанного подхода к управлению позволяет спрогнозировать будущее состояние транспортной системы при выбранном управлении, что при соответствующей постановке задачи управления может обеспечить предотвращение образования общесетевых транспортных заторов;

- область возможных решений систем оптимального управления светофорными объектами ограничивается: временными рамками базовых сигнальных планов; ограничениями предметной области оптимизации (например, отдельная оптимизация сдвига, цикла и разбиения в SCOOT); ограничениями приращения длительности сигналов, когда с целью исключения значительных флуктуаций сигнальных планов ограничивается область поиска оптимума (ограничения на приращение длительности сигналов до 2-4 с);

– задача предотвращения и ликвидации транспортных заторов в современных системах управления дорожным движением решается следующими методами: пропуском пиковой загрузки по магистральному направлению; нормированием транспортного предложения в сети; предотвращением блокирования поперечных направлений. Необходимое решение достигается путем модификации весовых коэффициентов целевой функции;

– зона воздействия противозаторового управления, а так же значения весовых коэффициентов целевой функции, обеспечивающих предотвращение или ликвидацию транспортного затора, устанавливаются транспортным инженером экспертным методом на основе собственного опыта.

Рассмотрено использование управления с прогнозирующей моделью для управления насыщенными транспортными потоками. Основным достоинством метода является простота формирования обратной связи в сочетании с высокой адаптивностью метода. Метод позволяет осуществлять оптимальное управление светофорными объектами с учетом различных ограничений на управляющие и управляемые переменные.

Синтез управления с прогнозирующей моделью требует математическое описание объекта управления – очереди и задержки транспортных средств. Используемые в практике управления дорожным движением модели очереди и задержки транспортных средств на регулируемых пересечениях, включают детерминированные и стохастические составляющие. Указанные модели были получены как с использованием теоретического подхода, так и элементов аппроксимации. В условиях насыщенного движения транспортных потоков оценка длины очереди и задержек транспортных средств возможна только с использованием моделей, полученных для нестационарных условий движения, в частности Highway Capacity Manual 2000.

Для использования указанных моделей требуется выполнения измерений или оценки остаточных очередей на регулируемых направлениях движения. Существующие методики оценки длин очередей позволяют производить оценку количества автомобилей в очереди, чего не достаточно для определения условий и момента блокирования транспортных связей. Введены понятия фронта очереди и фронта разъезда очереди, предложена математическая модель, описывающая их динамику на транспортной связи. Условие блокирования транспортных связей сформулировано как превышение фронтом очереди ее длины. На основании анализа полученных аналитических зависимостей предложена методика расчета сдвига включения разрешающего сигнала, исключающая блокирования предварительно установленного магистрального направления движения.

Оценка эффективности предложенных методов управления, проведенная с использованием программы микроскопического моделирования SUMO, показала их эффективность в условиях насыщенного движения.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Адаптивные системы управления дорожным движением в городах [Текст]: моногр. / А.А. Власов. – Пенза: ПГУАС, 2012. – 162 с.
2. Брайловский, Н.О. Управление движением транспортных средств [Текст] / Н.О. Брайловский, Б.И. Грановский. – М.: Транспорт, 1975. – 112 с.
3. Власов, А.А. Эволюционный метод расчета режимов работы светофорной сигнализации [Текст] / А.А. Власов, В.Ю. Акимова, М.В. Широков // Вестник Харьковского национального автомобильно-дорожного университета. – 2009. – № 47. – С. 97-99.
4. Врубель, Ю.А. Исследование в дорожном движении [Текст]: учеб.-метод. пособие к лабораторным работам для студентов специальности 1-44 01 02 «Организация дорожного движения» / Ю.А. Врубель. – Минск: БНТУ, 2007. – 178 с.
5. Капитанов, В.Т. Методика расчета светофорных циклов [Текст] / В.Т. Капитанов, С.В. Шауро. – М.: Всесоюзный НИИ безопасности движения МВД СССР, 1979. – 51 с.
6. Капитанов, В.Т. Расчет параметров светофорного регулирования [Текст] / В.Т. Капитанов. – М.: ВНИИ БД МВД СССР, 1981. – 96 с.
7. Кремер, Н.Ш. Теория вероятностей и математическая статистика [Текст]: учебник для вузов / Н.Ш. Кремер. – 2-е изд., перераб. и доп. – М.: ЮНИТИ-ДАНА, 2004, –573с.
8. Методика предупреждения сетевых транспортных заторов [Текст] /Р.Ю. Лагерева, А.Ю. Михайлов, С.В. Лагерева // Вестник НЦБЖД. – 2010. – №5. – С. 82-88.
9. Печерский, М.П. Автоматизированные системы управления дорожным движением [Текст] / М.П. Печерский, В.Г. Хорович. – М.: Транспорт, 1979. – 175 с.
10. Проблемы применения алгоритмов оптимизации параметров светофорного цикла в реальном времени [Текст] / А.А. Власов, М.В. Широков // Вестник МАДИ (ГТУ) – 2009. – №17. – С. 103–108.
11. Системы и средства автоматизированного управления дорожным движением в городах [Текст] / Е.Б. Хилажев, В.С. Соколовский, В.М. Гурьев, Я.И. Зайденберг. – М.: Транспорт, 1984. – 183 с.
12. Теоретические принципы возникновения и упреждения заторовых состояний на автодорогах [Текст] / В.Г. Живоглядов, О.Н. Бахтина // Известия высших учебных заведений. Северо-Кавказский регион. Технические науки. – 2004. – №3. – С. 103-105.

13. Управление движением транспортных средств [Текст] / Н.О. Брайловский, Б.И. Грановский. – М.: Транспорт. 1978 г. 112 с.
14. Управление насыщенными регулируруемыми пересечениями [Текст] / Р.Ю. Лагерев, А.В. Зедгенизов, А.Г. Левашев // Организация и безопасность дорожного движения в крупных городах: сб. докладов десятой международной конференции «Организация и безопасность дорожного движения в крупных городах. Инновации: ресурс и возможности». – СПб.: СПбГАСУ, 2012. – С 148 – 154.
15. Якимов, М.Р. Транспортное планирование: создание транспортных моделей городов [Текст]: моногр. / М.Р. Якимов. – М.: Логос, 2013. – 188 с.
16. ОДМ «Руководство по прогнозированию интенсивности движения на автомобильных дорогах» [Текст].
17. Allgower F., Badgwell T. A., Qin J. S., Rawlings J.B., and Wright S.J. Nonlinear predictive control and moving horizon estimation – an introductory overview. In Springer-Verlag, pages 391–449, Chapter 12, 1999.
18. Allsop R. B., “SIGSET: A computer program for calculating traffic capacity of signal-controlled road junctions,” Traffic Eng. Control, vol. 12, pp. 58–60, 1971.
19. Bang, K. L. Optimal Control of Isolated Traffic Signals. In Traffic Engineering and Control, 17(7), 1976, pp. 288-292.
20. Beckmann, M. J., C. B. Mc Guire, and C. B. Winsten (1956). Studies in the Economics in Transportation. New Haven, Yale University Press
21. Berg, M. van den, Integrated Control of Mixed Traffic Networks using Model Predictive Control, T2010/4, April 2010, TRAIL Thesis Series, the Netherlands
22. Busch F. and Kruse G. MOTION for SITRAFFIC. In International IEEE Conference on Intelligent Transportation Systems, pages 61–64, Oakland, California, 2001. 62
23. Camponogara E., Jia D., Krogh B. H., and Talukdar S.. Distributed model predictive control. IEEE Control Systems Magazine, 22(1):44–52, February 2002.
24. Chang, E. C., and Messer C. J.. Minimum Delay Optimization of a Maximum Bandwidth Solution to Arterial Signal Timing. In Transportation Research Record 1005, TRB, National Research Council, Washington, DC, 1985, pp. 89-95.
25. Cohen, S. L. Concurrent Use of MAXBAND and TRANSYT Signal Timing Programs for Arterial Signal Optimization. In Transportation Research Record 906, TRB, National Research Council, Washington, DC, 1983, pp. 81-84.
26. Cohen, S. L., and C. C. Liu. The Bandwidth-Constrained TRANSYT Signal Optimization Program. In Transportation Research Record 1057, TRB, National Research Council, Washington, DC, 1986, pp. 1-9.



27. Crenshaw, P., “ACS Lite,” presented at the TRB A3A18 Meeting, 2000 Midyear Meeting Minutes, Seattle, Wash., July 9–10, 2000.
28. Curtis, E., “ACS Lite: The Next Generation of Traffic Signal Control,” presented at the Winter Workshop: Adding to Your Traffic Engineering Toolkit, Columbus, Ohio, Feb. 28, 2007.
29. D’Ans G. C. and Gazis D. C., “Optimal control of oversaturated store-and-forward transportation networks,” *Transp. Sci.*, vol. 10, pp. 1–19, 1976.
30. Darroch, J. N. On the Traffic-Light Queue. *Ann. Math. Statist.*, 35, pp. 380-388.
31. Davis L. C.. Effect of adaptive cruise control systems on traffic flow. *Physical Review E*, 69:1–8, 2004.
32. Davison E. J. and Ozguner U., “Decentralized control of traffic networks,” *IEEE Trans. Automat. Contr.*, vol. 28, pp. 677–688, 1983.
33. Dell’Olmo, P., and P. Mirchandani. REALBAND: An Approach for Real-Time Coordination of Traffic Flows on Networks. In *Transportation Research Record 1494*, TRB, National Research Council, Washington, DC, 1995, pp. 106-116.
34. Diakaki C., Dinopoulou V., Aboudolas K., Papageorgiou M., Ben-Shabat E., Seider E. and Leibov A., “Extensions and new applications of the signal control strategy TUC,” presented at the 82nd Annu. Meeting Transportation Research Board, 2003. Paper no. 03-3040.
35. Donati, F., V. Mauro, G., Roncolini, and M. Vallauri, “A Hierarchical Decentralised Traffic Light Control System,” *Proceedings from IFAC 9th World Congress*, Vol. II, 11G/A-1, 1984.
36. Fehon, K., Chong R., and Black J., “Adaptive Traffic Signal System for Cupertino California,” *Proceedings of the 4th Asia– Pacific Transportation Development Conference*, 16th Annual Meeting of ICTPA, Apr. 18–20, 2003, 14 pp.
37. Friedrich, B., Sachse T., Hoops M., Jendryschik W., and Reichert G., “Balance and Varia Methods for Traffic Adaptive Control,” *Proceedings of the Second World Congress on the ITS*, Yokohama, Japan, 1995, pp. 2356–2361.
38. Gartner N. H., Pooran F. J., and Andrews C. M., “Implementation of the OPAC adaptive control strategy in a traffic signal network,” in *Proc. 4th IEEE Conf. Intelligent Transportation Systems*, 2001, pp. 197–202.
39. Gartner, N. H. OPAC: A Demand-Responsive Strategy for Traffic Signal Control. In *Transportation Research Record 906*, TRB, National Research Council, Washington, DC, 1983, pp. 75-81.
40. Gartner, N. H., Assmann S. F., Lasaga F., and Hou D. L.. MULTIBAND – A Variable-Bandwidth Arterial Progression Scheme. In *Transportation Research Record 1287*, TRB, National Research Council, Washington, DC, 1990, pp. 212-222

41. Gartner, N. H., Tarnoff P. J., and Andrews C. M.. Evaluation of Optimized Policies for Adaptive Control Strategy. In Transportation Research Record 1324, TRB, National Research Council, Washington, DC, 1991, pp. 105-114.
42. Gartner, N.H., Demand-Responsive Decentralized Urban Traffic Control, Report DOT/RSPA/DPB-50/81/24, U.S. Department of Transportation, Washington, D.C., 1982.
43. Gartner, N.H., Pooran F.J., and Andrews C.M., "Implementation and Field Testing of the OPAC Adaptive Control Strategy in RT-TRACS," Journal of the Transportation Research Board, Washington, D.C., 2002, pp. 148–156.
44. Gazis D. C., Potts R. B., "The oversaturated intersection," in Proc. 2nd Int. Symp. Traffic Theory, 1963, pp. 221–237.
45. Gazis D. C., Traffic Theory. Boston, MA: Kluwer, 2002.
46. Gazis, D. C. (1974). Traffic Science. A Wiley-Intersection Publication, pp. 148-151
47. Grubba, J., Allyn D., Barbaresso J., and Beaubien R.. Selecting Real-Time Adaptive Traffic Control for Oakland County, Michigan. Presented at the 72nd Annual Meeting, Transportation Research Board, Washington, DC, 1993.
48. Haight, F. A. (1959). Overflow At A Traffic Flow. Biometrika. Vol. 46, Nos. 3 and 4, pp. 420-424.
49. Hale D.K., Traffic Network Study Tool. TRANSYT-7F, User's Guide, McTrans Center, University of Florida, 2008
50. Hall S., Chaib-draa B., and Laumonier J. Car platoons simulated as a multiagent system. In Proceedings of the 4th Workshop on Agent-Based Simulation, pages 57–63, Montpellier, France, 2003.
51. Head, K. L., Mirchandani P. B., and D. Sheppard. Hierarchical Framework for Real Time Traffic Control. In Transportation Research Record 1360, TRB, National Research Council, Washington, DC, 1992, pp. 82-88.
52. Head, K. L., Mirchandani P. B., and Shelby S. The RHODES Prototype: A Description and Some Results. Presented at the 77th Annual Meeting of the Transportation Research Board, Washington, DC, 1998.
53. Head, K.L., Mirchandani P.B., and Shepherd D., "A Hierarchical Framework for Real-Time Traffic Control," Transportation Research Record 1360, Transportation Research Board, National Research Council, Washington, D.C., 1992.
54. Hegyi. A. Model Predictive Control for Integrating Traffic Control Measures. PhDthesis, Delft University of Technology, Delft, The Netherlands, 2004.
55. Henry, J.J., Farges J.L., and Tufal J., "The PRODYN Real Time Traffic Algorithm," Proceedings of the IFAC Symposium, Baden–Baden, Germany, 1983.

56. Herman R. and Weiss G. H.. Comments on the highway crossing problem. *Operations Research*, 9(6):828–840, 1961.
57. Highway Capacity Manual. Transportation Research Board, Washington, D.C. 2010. ISBN 978-0-309-16077-3
58. Highway Capacity Manual. Transportation Research Board, Washington, D.C. 2000. ISBN 0-309-06681-6
59. Hillier, J. A. and Rothery R. The Synchronization of Traffic Signals for Minimum Delays. *Transportation Science*, 1(2), pp. 81-94.
60. Hunt, P.B., Robertson D.I., Bretherton R.D., and Winton R.I., “SCOOT—A Traffic Responsive Method of Coordinating Signals,” Report TRRL 1014, Transport and Road Research Laboratory, Crowthorne, Berkshire, U.K., 1981, 44 pp.
61. Improta G. and Cantarella G. E., “Control systems design for an individual signalised junction,” *ransp. Res. B*, vol. 18, pp. 147–167, 1984.
62. Kim K. J. and Bell M. G. H., “Development of an integrated traffic control strategy for both urban signalised and motorway networks,” presented at the Proc. 1st Meeting EURO Working Group on Urban Traffic and Transportation, Landshut, Germany, 1992.
63. Kruse, G., “MOTION – Signal Control for Urban Road Networks,” presented at the TRB Midyear Traffic Signal Committee Meeting, Adaptive Traffic Signal Control Workshop, Pacific Grove, Calif., July 12–14, 1998. Friedrich et al. 1995
64. Lee J. B., K. Ozbay. Calibration of a Macroscopic Traffic Simulation Model Using Enhanced Simultaneous Perturbation Stochastic Approximation Methodology TRANSPORTATION RESEARCH BOARD’S 87TH ANNUAL MEETING, WASHINGTON, D.C., 2008
65. Leutzbach W. An Introduction to the Theory of Traffic Flow. Springer-Verlag, Berlin, Germany, 1988.
66. Lim J. H., Hwang S. H., Suh I. H., and Bien Z., “Hierarchical optimal control of oversaturated urban networks,” *Int. J. Control*, vol. 33, pp. 727–737, 1981.
67. Lin, S., Efficient Model Predictive Control for Large-Scale Urban Traffic Networks, T2011/3, April 2011, TRAIL Thesis Series, the Netherlands
68. Little, J. D. C. Approximate Expected Delays for Several Maneuvers by Driver in a Poisson Traffic. *Operations Research*, 9, pp. 39-52.
69. Little, J., Kelson M. D., and Gartner N. H. MAXBAND: A Program for Setting Signals on Arteries and Triangular Networks. In *Transportation Research Record 795*, TRB, National Research Council, Washington, DC, 1981, pp. 40-46.
70. Lowrie, P. R. The Sydney Coordinated Adaptive Traffic System: Principles, Methodology, Algorithms. *Proceedings of the International*

Conference on Road Traffic Signaling, London, England, IEEE Conference Publication No. 207, 1982, pp. 67-70.

71. Luk, J. Y. K., Sims A. G., and Lowrie P. R. SCATS: Application and Field Comparison with a TRANSYT Optimized Fixed Time System. Proceedings of the International Conference on Road Traffic Signaling, London, England, IEE Conference Publication No. 207, 1982, pp. 71-74.

72. Martin, P. T. and Hockaday S.. SCOOT – An Update. In ITE Journal 65 (1), January 1995, pp. 44-48.

73. Mauro V. and C. Di Taranto, “UTOPIA,” Control, Computers, Communications in Transportation: Selected Papers from the IFAC Symposium, 1990, pp. 245–252.

74. McNeil, D. R. A Solution to the Fixed-Cycle Traffic Light Problem for Compound Poisson Arrivals. J. Appl. Prob. 5, pp. 624-635.

75. Messer, C. J., Whitson R. H., Dudek C. L., and Romano E. J.. A Variable Sequence Multiphase Progression Optimization Program. In Highway Research Record 445, Highway Research Board, National Research Council, Washington, DC, 1973, pp. 24-33.

76. Metastable States in a Microscopic Model of Traffic Flow/ Stefan Krauß, Peter Wagner, Christian Gawron // Physical Review E, – 1997. – volume 55, number 304. – pp. 55-97

77. Michaels R. M.. Perceptual factors in car following. In Proceedings of the 2nd International Symposium for Theory Road Traffic Flow.

78. Michalopoulos P. G. and Stephanopoulos G., “Oversaturated signal systems with queue length constraints—I: Single intersection,” Transp. Res., vol. 11, pp. 413–421, 1977a.

79. Miller, A. J. A Computer Control System for Traffic Network. Proceedings of the Second International Symposium on the Theory of Road Traffic Flow, Paris, France, 1963, pp. 200-220.

80. Miller, A. J. Australian Road Capacity Guide – Provisional Introduction and Signalized Intersections. (by ARRB report ARR No. 123, 1981). ARRB Bulletin No.3.

81. Mirchandani, P. and Head L., “RHODES Traffic-Adaptive Control Systems,” presented at the TRB Annual Meeting, Adaptive Signal Control Systems Workshop, Washington, D.C., Jan. 7–11, 2001.

82. Mueck, J., “Recent Developments in Adaptive Control Systems in Germany,” Proceedings of the 12th World Congress on ITS, San Francisco, Calif., Nov. 6–10, 2005, 11 pp.

83. Mueck, J., “The ‘German’ Approach to Adaptive Network Control,” Traffic Technology International, Erlangen, Germany, Aug. 2008, 13 pp

84. Mueck, J., “The German Approach to Adaptive Network Control,” Proceedings of the 7th European Congress and Exhibition on

Intelligent Transport Systems and Services, Geneva, Switzerland, June 4–6, 2008, 6 pp.

85. Nagel K. Particle hopping models and traffic flow theory. *Physical Review E*, 53:4655–4672, 1996.

86. Newell, G. F. Approximation Methods for Queues with Application to the Fixed-Cycle Traffic Light. *SIAM Review*, Vol.7.

87. Newell, G. F. (1960). Queues for a Fixed-Cycle Traffic Light. *The Annals of Mathematical Statistics*, Vol.31, No.3, pp. 589-597.

88. Pacey, G. M. (1956). The Progress of a Bunch of Vehicles Released from a Traffic Signal. Research Note No. Rn/2665/GMP. Road Research Laboratory, London

89. Papageorgiou M., “An integrated control approach for traffic corridors,” *Transp. Res. C*, vol. 3, pp. 19–30, 1994.

90. Papageorgiou M., Posch B. and Schmidt G. Comparison of macroscopic models for control of freeway traffic. *Transportation Research Part B*, 17(2):107–116, April 1983.

91. Park E. S., Lim J. H., Suh I. H., and Bien Z., “Hierarchical optimal control of urban traffic networks,” *Int. J. Control*, vol. 40, pp. 813–829, 1984.

92. Partha Chakroborty. Models of vehicular traffic: An engineering perspective. *Physica*, 372(10):151–161, June 2006.

93. Pesti, G., Byrd P.S., Kruse M., and McCoy P.T., Evaluation of the SPOT Adaptive Traffic Control System, FHWA Technology Project No. 97068, Federal Highway Administration, Nebraska Department of Roads, Omaha, May 1999, 35 pp.

94. Pipes L. A.. An operational analysis of traffic dynamics. *Journal of Applied Physics*, 24(1):274–287, 1953. 65

95. Pooran, F. J., Andrews C., and Gartner N. H.. Implementation of the OPAC Adaptive Control System in RT-TRACS. Presented at the ITS World Congress Sixth Annual Meeting, Toronto, Canada, November 1999.

96. Pooran, F., “Optimized Policies for Adaptive Control (OPAC),” presented at the TRB Annual Meeting, Adaptive Traffic Signal Control Systems Workshop, Washington, D.C., Jan. 9, 2000.

97. Pooran, F., “RT-TRACS Adaptive Control Algorithms, VFC OPAC,” presented at the TRB A3A18 Midyear Meeting, Adaptive Traffic Signal Control Systems Workshop, Pacific Grove, Calif., July 12–14, 1998.

98. Recent Development and Applications of SUMO – Simulation of Urban MObility/ Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. // *International Journal On Advances in Systems and Measurements*, – 2012. – 5 (3&4). – pp. 128-138

99. Robertson, D. I. TRANSYT: A Traffic Network Study Tool. Road Research Laboratory Report LR 253, Crowthorne.

100. Sen, S., and Head K. L.. Controlled Optimization of Phases (COP) at an Intersection. In *Transportation Science*, Vol. 31 (1), 1997, pp. 5-17.
101. Shelby, S.G., Bullock D.M., Gettman D., Ghaman R.S., Sabra Z.A. and Soyke N., “Overview and Performance Evaluation of ACS Lite—Low Cost Adaptive Signal Control System, presented at the 87th Annual Meeting of the Transportation Research Board, Washington, D.C., Jan. 13–17, 2008, 17 pp.
102. SIGCAP: A computer program for assessing the traffic capacity of signal-controlled road junctions, *Traffic Eng. Control*, vol. 17, pp. 338–341, 1976.
103. Singh M. G. and Tamura H., “Modeling and hierarchical optimization of oversaturated urban traffic networks,” *Int. J. Control*, vol. 20, pp. 269–280, 1974.
104. *Synchro Studio 8 User Guide*, Trafficware, 2011
105. *Traffic Detector Handbook: Third Edition – Volume I / FHWA-HRT-06-108*, 2006 – 291 p.
106. Vincent, R. A., and Peirce J. R. *MOVA: Traffic Responsive, Self-Optimizing Signal Control for Isolated Intersections*. Transport and Road Research Laboratory Report 170, Crowthorne, U.K., 1988.
107. W. Brilon, R. Koenig, and R. J. Troutbeck. Useful estimation procedures for critical gaps. *Transportation Research Part A*, 33:161–186, 1999.
108. Wallace, C. E. *TRANSYT-7F User's Manual*, Report UF-TRC-U32 FP-06/07. U.S. Department of Transportation, Federal Highway Administration, 1984.
109. Webster F. V., “Traffic signal settings,” Road Research Laboratory, London, U.K., Road Res. Tech. Paper no. 39, 1958.
110. Webster, F. V. *Traffic Signal Settings*. Road Research Laboratory Technical Paper No. 39, HMSO, London.
111. Weiss G. H. and Maradudin A. A. Some problems in traffic delay. *Operations Research*, 10:74–104, 1962.
112. Yang Q. and Koutsopoulos H. N. A microscopic traffic simulator for evaluation of dynamic traffic management systems. *Transportation Research Part C*, 4(3):113–129, June 1996.

# ПРИЛОЖЕНИЯ

## Приложение 1

### Исходные файлы транспортной сети модели локального пересечения

Файл конфигурации net.cfg программы netconvert:

```
<?xml version="1.0" encoding="iso-8859-1"?>

<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/netconve
rtConfiguration.xsd">

<input>
<node-files value="nodes.nod.xml"/>
<edge-files value="edges.edg.xml"/>
    <connection-files value="connect.con.xml" />
</input>

<output>
<output-file value="net.net.xml"/>
</output>

<processing>
<no-turnarounds value="true"/>
<speed-in-kmh value="true"/>
</processing>

</configuration>
```

Файл описания узлов nodes.nod.xml:

```
<nodes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/nodes_fi
le.xsd">

    <node id="TL" x="400" y="400" type="traffic_light"/>
    <node id="1" x="-200" y="400"/>
    <node id="2" x="400" y="800"/>
    <node id="3" x="1000" y="400"/>
    <node id="4" x="400" y="0"/>

</nodes>
```

## Файл описания связей edges.edg.xml:

```

<edges xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/edges_file.xsd">
  <edge id="1" from="1" to="TL" priority="2" numLanes="2"
speed="80.">
  <split pos="0" lanes="0 1"/>
  <split pos="300" lanes="0 1 2"/>
</edge>
  <edge id="2" from="2" to="TL" priority="2" numLanes="2"
speed="80.">
  <split pos="0" lanes="0"/>
  <split pos="300" lanes="0 1"/>
</edge>
  <edge id="3" from="3" to="TL" priority="2" numLanes="2"
speed="80.">
  <split pos="0" lanes="0 1"/>
  <split pos="300" lanes="0 1 2"/>
</edge>
  <edge id="4" from="4" to="TL" priority="2" numLanes="2"
speed="80.">
  <split pos="0" lanes="0"/>
  <split pos="300" lanes="0 1"/>
</edge>

  <edge id="-1" from="TL" to="1" priority="2" numLanes="2"
speed="80."/>
  <edge id="-2" from="TL" to="2" priority="2" numLanes="2"
speed="80."/>
  <edge id="-3" from="TL" to="3" priority="2" numLanes="2"
speed="80."/>
  <edge id="-4" from="TL" to="4" priority="2" numLanes="2"
speed="80."/>

</edges>

```

## Файл описания связей полос движения на пересечении connect.con.xml:

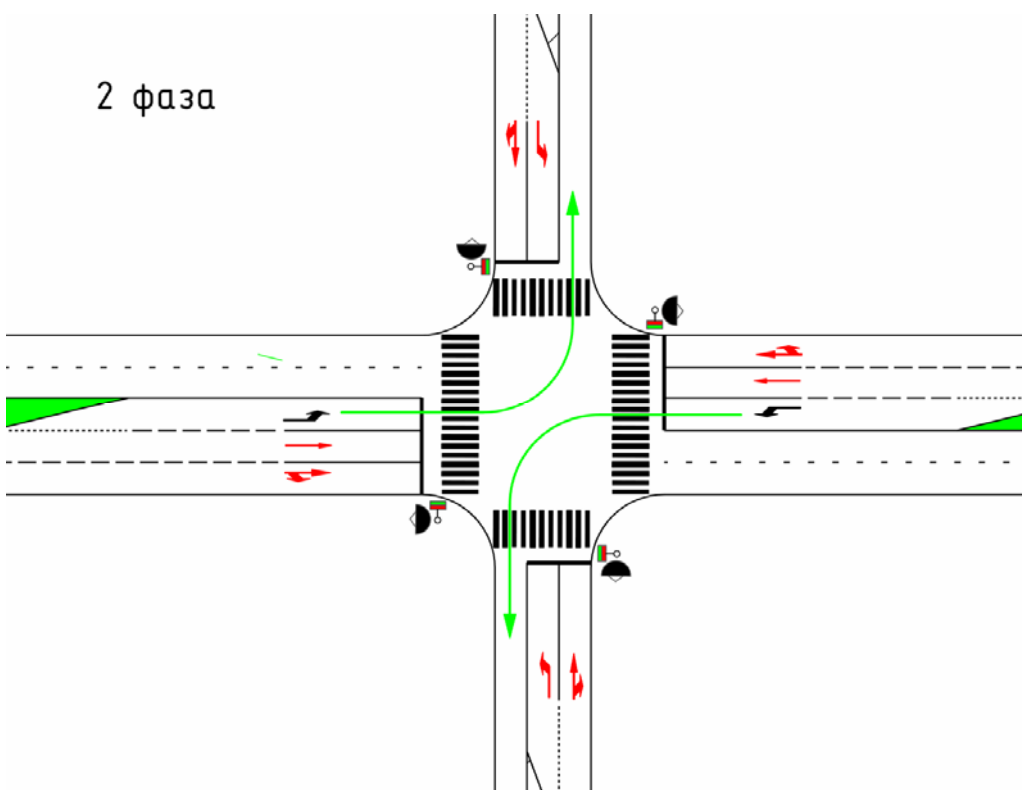
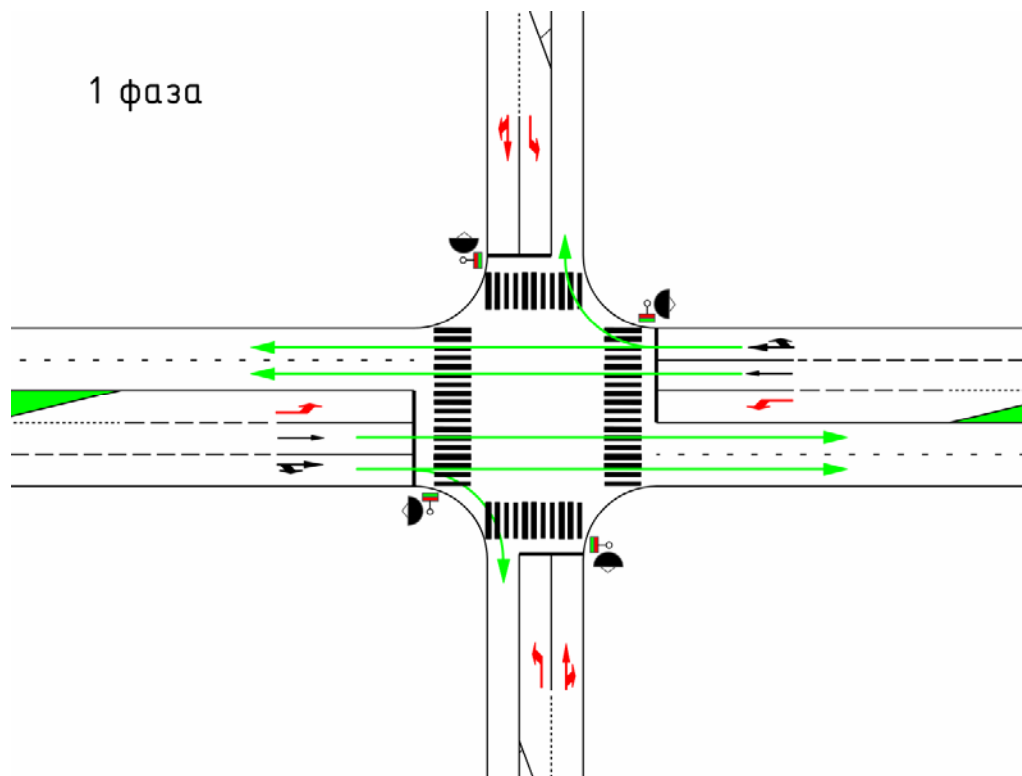
```

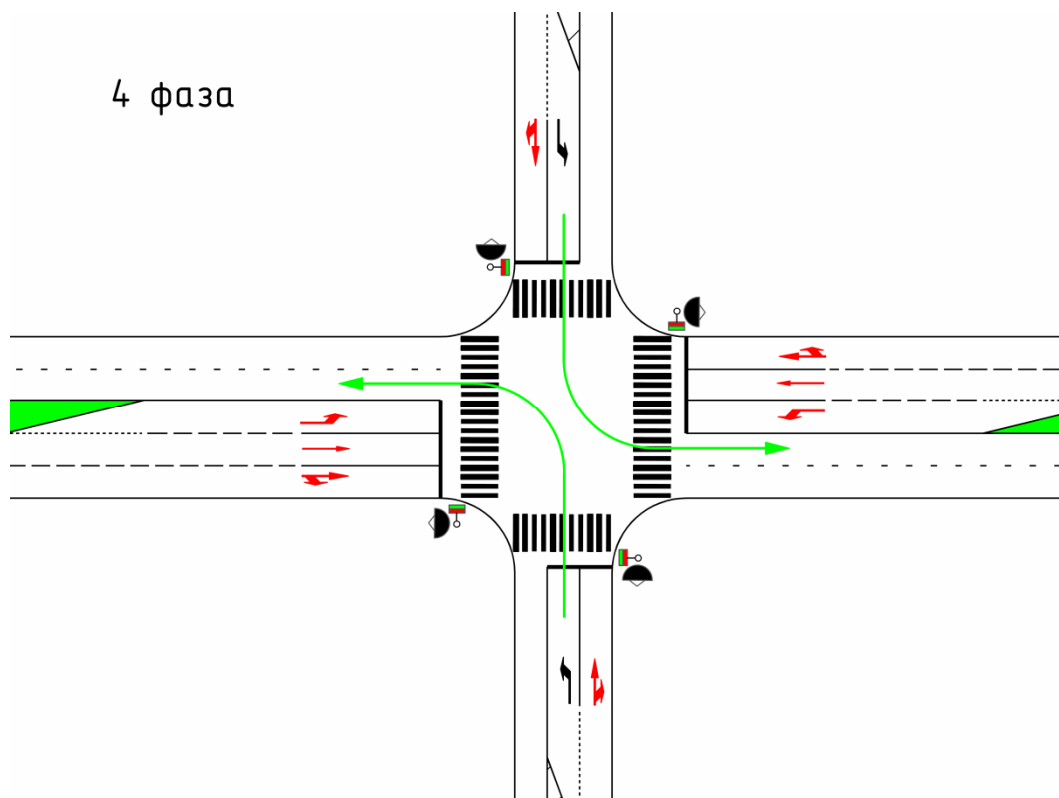
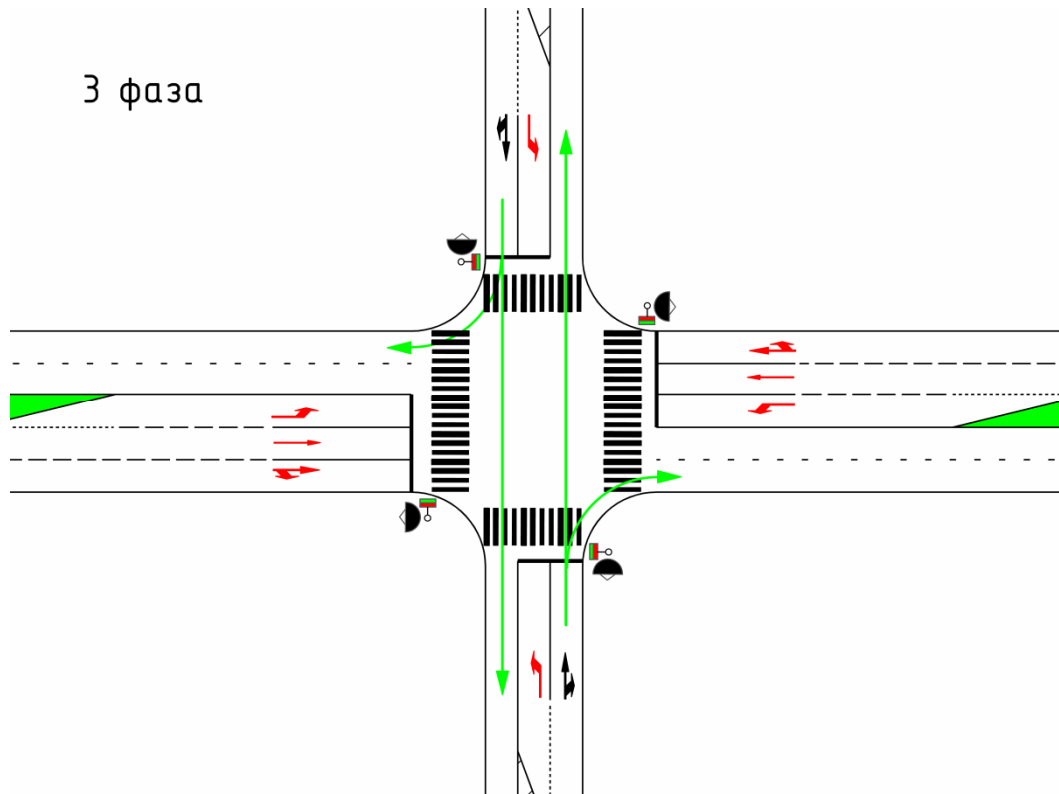
<connections xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/connecti
ons_file.xsd">
  <connection from="4.300" to="-1" fromLane="1" toLane="1"/>
  <connection from="4.300" to="-2" fromLane="0" toLane="0"/>
  <connection from="4.300" to="-3" fromLane="0" toLane="0"/>
  <connection from="2.300" to="-1" fromLane="0" toLane="0"/>
  <connection from="2.300" to="-4" fromLane="0" toLane="0"/>
  <connection from="2.300" to="-3" fromLane="1" toLane="1"/>
</connections>

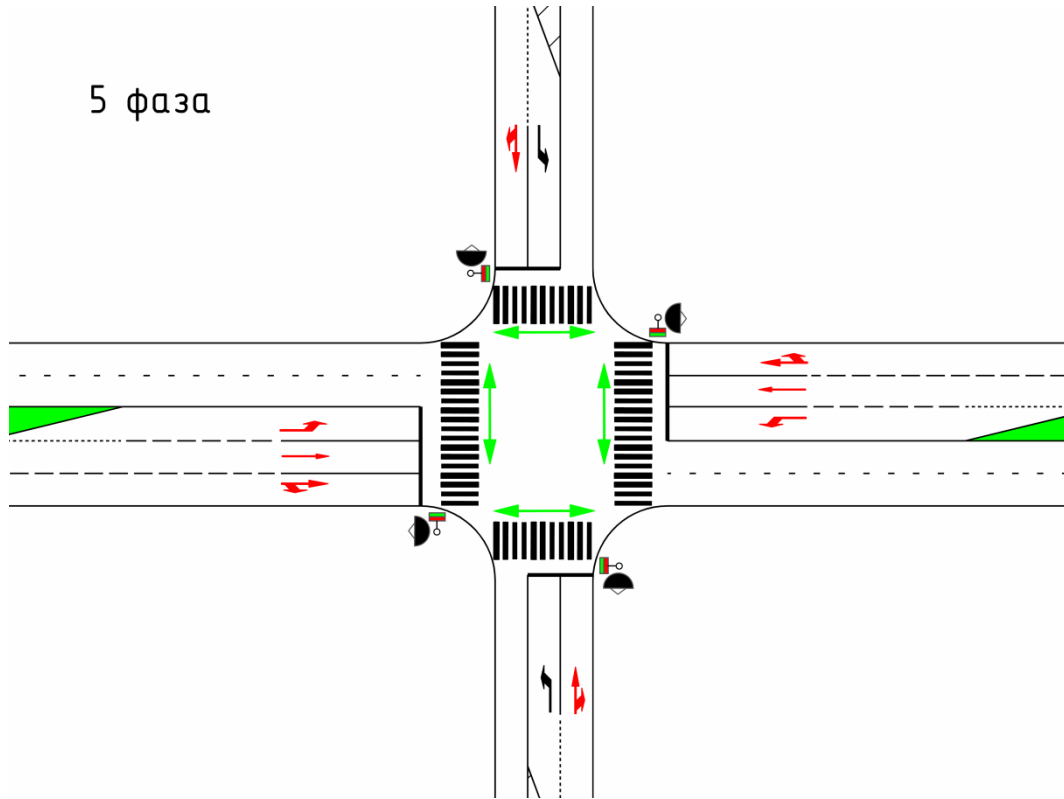
```



Схема разъезда и описание режима работы светофорного объекта в файле сети net\_static.net.xml







```

<tlLogic id="TL" type="static" programID="0" offset="0">
<phase duration="40" state="rrrGGGrrrrGGGr"/>
<phase duration="3" state="rrryyyrrrryyr"/>
<phase duration="16" state="rrrrrrGrrrrrrG"/>
<phase duration="3" state="rrrrrryrrrrrry"/>
<phase duration="34" state="GGrrrrrrGGrrrrr"/>
<phase duration="3" state="yyrrrrrryyrrrrr"/>
<phase duration="11" state="rrGrrrrrrrGrrrr"/>
<phase duration="3" state="rryrrrrrryrrrrr"/>
<phase duration="22" state="rrrrrrrrrrrrrrr"/>
</tlLogic>
    
```

**Исходные файлы транспортного спросамодели локального  
пересечения**

Файл конфигурации config.jtrrcfg программы jtrrouter:

```
<?xml version="1.0" encoding="UTF-8"?>

<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/jtrroute
rConfiguration.xsd">

<input>
<net-file value="net.net.xml"/>
<flow-files value="flows.flows.xml"/>
<turn-ratio-files value="turns.turns.xml"/>
</input>

<output>
<output-file value="routes.rou.xml"/>
</output>

<processing>
<ignore-errors value="true"/>
</processing>

<report>
<no-step-log value="true"/>
</report>

</configuration>
```

## Файл транспортного спроса flows.flows.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<flowdefs>
<flow id="1_0" from="1" begin="0" end="1800" number="320"/>
<flow id="2_0" from="2" begin="0" end="1800" number="150"/>
<flow id="3_0" from="3" begin="0" end="1800" number="340"/>
<flow id="4_0" from="4" begin="0" end="1800" number="150"/>

<flow id="1_1" from="1" begin="1801" end="5400"
number="1240"/>
<flow id="2_1" from="2" begin="1801" end="5400" number="300"/>
<flow id="3_1" from="3" begin="1801" end="5400"
number="1080"/>
<flow id="4_1" from="4" begin="1801" end="5400" number="300"/>

<flow id="1_2" from="1" begin="5401" end="7200" number="420"/>
<flow id="2_2" from="2" begin="5401" end="7200" number="300"/>
<flow id="3_2" from="3" begin="5401" end="7200" number="400"/>
<flow id="4_2" from="4" begin="5401" end="7200" number="250"/>
</flowdefs>

```

## Файл описания поворотных потоков turns.turns.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<turns xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/turns_file.xsd">
<interval begin="0" end="1800">
  <fromEdge id="1.300">
    <toEdge id="-2" probability=".214"/>
    <toEdge id="-3" probability=".595"/>
    <toEdge id="-4" probability=".191"/>
  </fromEdge>

  <fromEdge id="3.300">
    <toEdge id="-2" probability=".227"/>
    <toEdge id="-1" probability=".569"/>
    <toEdge id="-4" probability=".204"/>
  </fromEdge>

  <fromEdge id="2.300">
    <toEdge id="-1" probability=".2"/>
    <toEdge id="-3" probability=".2"/>
    <toEdge id="-4" probability=".6"/>
  </fromEdge>

  <fromEdge id="4.300">
    <toEdge id="-1" probability=".2"/>
    <toEdge id="-2" probability=".6"/>
    <toEdge id="-3" probability=".2"/>
  </fromEdge>
</interval>

```

## Окончание прил. 2

```
<interval begin="1801" end="5400">
  <fromEdge id="1.300">
    <toEdge id="-2" probability=".1"/>
    <toEdge id="-3" probability=".771"/>
    <toEdge id="-4" probability=".129"/>
  </fromEdge>

  <fromEdge id="3.300">
    <toEdge id="-2" probability=".169"/>
    <toEdge id="-1" probability=".731"/>
    <toEdge id="-4" probability=".1"/>
  </fromEdge>

  <fromEdge id="2.300">
    <toEdge id="-1" probability=".2"/>
    <toEdge id="-3" probability=".2"/>
    <toEdge id="-4" probability=".6"/>
  </fromEdge>

  <fromEdge id="4.300">
    <toEdge id="-1" probability=".2"/>
    <toEdge id="-2" probability=".6"/>
    <toEdge id="-3" probability=".2"/>
  </fromEdge>
</interval>

<interval begin="5401" end="7200">
  <fromEdge id="1.300">
    <toEdge id="-2" probability=".214"/>
    <toEdge id="-3" probability=".595"/>
    <toEdge id="-4" probability=".191"/>
  </fromEdge>

  <fromEdge id="3.300">
    <toEdge id="-2" probability=".227"/>
    <toEdge id="-1" probability=".569"/>
    <toEdge id="-4" probability=".204"/>
  </fromEdge>

  <fromEdge id="2.300">
    <toEdge id="-1" probability=".2"/>
    <toEdge id="-3" probability=".2"/>
    <toEdge id="-4" probability=".6"/>
  </fromEdge>

  <fromEdge id="4.300">
    <toEdge id="-1" probability=".2"/>
    <toEdge id="-2" probability=".6"/>
    <toEdge id="-3" probability=".2"/>
  </fromEdge>
</interval>

</turns>
```

## Программа управления локальным светофорным объектом LocalControll.py

```
# -*- coding: utf-8 -*-
from xml.dom import minidom
import os, subprocess, sys, random
import math
import numpy as np
import deleyTooll
sys.path.append("C:\\sumo-0.18.0\\bin")
sys.path.append("C:\\sumo-0.18.0\\tools")
import traci

""""----- Объявление констант -----
""""
mu=60.72
sigma=22.5
PORT = 8813
AgregateTime=60
RollingGorizont=5
Lvech=7.5
delta=np.array([-3.0,0.0,3.0])
sumoBinary = 'c:\\sumo-0.18.0\\bin\\sumo-gui.exe'
sumoConfig = "c:\\sumo-0.18.0\\bin\\TraCI_controll.sumo.cfg"
queue="--queue-output queue.xml"

""""
----- Настройки модели -----
""""
edg={}
edg['1.300']={'Capacity':1500.0,'Length':150,'flow':0,'deley':0.0,'Q':0,'Qb':0,'Nlg':1}
edg['2.300']={'Capacity':1500.0,'Length':150,'flow':0,'deley':0.0,'Q':0,'Qb':0,'Nlg':1}
edg['3.300']={'Capacity':1500.0,'Length':100,'flow':0,'deley':0.0,'Q':0,'Qb':0,'Nlg':1}
edg['4.300']={'Capacity':1500.0,'Length':100,'flow':0,'deley':0.0,'Q':0,'Qb':0,'Nlg':1}
edg['1']={'Capacity':3600.0,'Length':300,'flow':0,'deley':0.0,'Q':0,'Qb':0,'Nlg':2}
edg['3']={'Capacity':3600.0,'Length':300,'flow':0,'deley':0.0,'Q':0,'Qb':0,'Nlg':2}
edg['2']={'Capacity':1800.0,'Length':200,'flow':0,'deley':0.0,'Q':0,'Qb':0,'Nlg':1}
edg['4']={'Capacity':1800.0,'Length':200,'flow':0,'deley':0.0,'Q':0,'Qb':0,'Nlg':1}

det_logic={}
det_logic['1']={'composed':['1_0', '1_1'],'subtracted':['1.300_2']}
det_logic['3']={'composed':['3_0', '3_1'],'subtracted':['3.300_2']}
det_logic['2']={'composed':['2_0'],'subtracted':['2.300_1']}
det_logic['4']={'composed':['4_0'],'subtracted':['4.300_1']}
det_logic['1.300']={'composed':['1.300_2'],'subtracted':[]}
det_logic['3.300']={'composed':['3.300_2'],'subtracted':[]}
det_logic['2.300']={'composed':['2.300_1'],'subtracted':[]}
det_logic['4.300']={'composed':['4.300_1'],'subtracted':[]}
```

```

phase={}
phase['0']={'edg':['1','3'],'duration':30,'minDur':7,'maxDur':60,'state':"rrrGGGrrrrGGGr"}
phase['1']={'edg':['fixed'],'duration':3,'minDur':3,'maxDur':3,'state':"rrryyyrrrryyyr"}
phase['2']={'edg':['1.300','3.300'],'duration':16,'minDur':7,'maxDur':40,'state':"rrrrrGrrrrrG"}
phase['3']={'edg':['fixed'],'duration':3,'minDur':3,'maxDur':3,'state':"rrrrryrrrrry"}
phase['4']={'edg':['2','4'],'duration':30,'minDur':7,'maxDur':60,'state':"GGrrrrrGGrrrrr"}
phase['5']={'edg':['fixed'],'duration':3,'minDur':3,'maxDur':3,'state':"yyrrrrryyrrrrr"}
phase['6']={'edg':['2.300','4.300'],'duration':16,'minDur':7,'maxDur':40,'state':"rrGrrrrrrGrrrrr"}
phase['7']={'edg':['fixed'],'duration':3,'minDur':3,'maxDur':3,'state':"rryrrrrryrrrr"}
phase['8']={'edg':['fixed'],'duration':22,'minDur':22,'maxDur':22,'state':"rrrrrrrrrrrrrr"}

tls={}
tls['TL']=phase

queueDet={}
queueDet['1']={'queueDet':'1.300_0','Lbq':150.0,'greenPhase':0,'begRedPhase':2}
queueDet['3']={'queueDet':'3_Fin','Lbq':350.0,'greenPhase':0,'begRedPhase':2}
queueDet['2']={'queueDet':'2_Fin','Lbq':200.0,'greenPhase':4,'begRedPhase':6}
queueDet['4']={'queueDet':'4_Fin','Lbq':200.0,'greenPhase':4,'begRedPhase':6}
queueDet['1.300']={'queueDet':'1.300_2','Lbq':150.0,'greenPhase':2,'begRedPhase':4}
queueDet['3.300']={'queueDet':'3.300_2','Lbq':150.0,'greenPhase':2,'begRedPhase':4}
queueDet['2.300']={'queueDet':'2.300_1','Lbq':60.0,'greenPhase':6,'begRedPhase':8}
queueDet['4.300']={'queueDet':'2.300_1','Lbq':60.0,'greenPhase':6,'begRedPhase':8}

vQueueDet={}
vQueueDet['1']={'laneControll':'1.300_0','Lbq':150.0,'greenPhase':0,'begRedPhase':2,'lo':0.0}
vQueueDet['3']={'laneControll':'3.300_0','Lbq':150.0,'greenPhase':0,'begRedPhase':2,'lo':0.0}
vQueueDet['2']={'laneControll':'2_0','Lbq':100.0,'greenPhase':4,'begRedPhase':6,'lo':90.0}
vQueueDet['4']={'laneControll':'4_0','Lbq':100.0,'greenPhase':4,'begRedPhase':6,'lo':90.0}
vQueueDet['1.300']={'laneControll':'1.300_2','Lbq':150.0,'greenPhase':2,'begRedPhase':4,'lo':0.0}
vQueueDet['3.300']={'laneControll':'3.300_2','Lbq':150.0,'greenPhase':2,'begRedPhase':4,'lo':0.0}
vQueueDet['2.300']={'laneControll':'2.300_1','Lbq':80.0,'greenPhase':6,'begRedPhase':8,'lo':0.0}
vQueueDet['4.300']={'laneControll':'4.300_1','Lbq':80.0,'greenPhase':6,'begRedPhase':8,'lo':0.0}

""""
----- Объявление функций -----
""""

def SpeedModification():
    Depardet=traci.simulation.getDepartedIDList()
    if len(Depardet)>0:
        for n in Depardet:
            speed=random.gauss(mu,sigma)
            if speed<16.0:
                speed=16.0
            traci.vehicle.setMaxSpeed(n,(speed/3.6))

def getCycleTime(phase):
    out=0
    for n in phase.keys():
        out+=phase[n]['duration']
    return float(out)

```



```

def getLostCycleTime(phase):
    out=0
    for n in phase.keys():
        if phase[n]['edg'][0]=='fixed':
            out+=phase[n]['duration']
    return out

def getActivePhase(phase):
    out={}
    for n in phase.keys():
        if not(phase[n]['edg'][0]=='fixed'):
            out[n]=phase[n]['duration']
    return out

def combineArray(a,b):
    out=[]
    for n in range(np.size(a)):
        b_size=np.shape(b)
        for m in range(b_size[0]):
            if np.size(b_size)==2:
                temp=[]
                temp.append(a[n])
                for k in range(np.size(b[m])):
                    temp.append(b[m][k])
                out.append(temp)
            else:
                temp=[a[n]]
                temp.append(b[m])
                out.append(temp)
    return out

def genGridControll(phase,delta):
    ActivePhase=getActivePhase(phase)
    keyPhase=ActivePhase.keys()
    out=delta
    for n in range(np.size(keyPhase)-1):
        out=combineArray(delta,out)
    return out

def setGridControll_(phase,delta):
    ActivePhase=getActivePhase(phase)
    keyPhase=ActivePhase.keys()
    c=genGridControll(phase,delta)
    phaseVariable={}
    (NumPhaseVariarion,NumPhase)=np.shape(c)
    for n in range(NumPhaseVariarion):
        temp={}
        for m in range(NumPhase):
            temp[keyPhase[m]]=ActivePhase[keyPhase[m]]+c[n][m]
        phaseVariable[str(n)]=temp
    return phaseVariable

```

```

def setGridControll(phase,delta):
    ActivePhase=getActivePhase(phase)
    keyPhase=ActivePhase.keys()
    c=genGridControll(phase,delta)
    phaseVariable={}
    (NumPhaseVariation,NumPhase)=np.shape(c)
    for n in range(NumPhaseVariation):
        temp={}
        for m in range(NumPhase):
            temp[keyPhase[m]]=c[n][m]
        phaseVariable[str(n)]=temp
    return phaseVariable

def getHaltPoint(lane):
    Vech=traci.lane.getLastStepVehicleIDs(lane)
    HaltVech=[]
    HaltPoint=[]
    for n in Vech:
        if traci.vehicle.getSpeed(n)<1.0:
            HaltVech.append(n)
    for n in HaltVech:
        HaltPoint.append(traci.lane.getLength(lane)-traci.vehicle.getLanePosition(n))
    return HaltPoint

def getHaltToPoint(HaltPoint,Point,delta=5.0):
    for n in HaltPoint:
        if (n>(Point-delta)and(n<(Point+delta))):
            return 1
    return -1

def getFrontOfStop(Flow,green,Cycle,Nlg,Qo,Lvech):
    b=Cycle-green
    flow=Flow/Nlg
    return ((Qo+b*flow/3600.0)*Lvech/(4.17-Lvech*flow/3600.0)+b)*Lvech*flow/3600.0

"""
----- Объявление классов -----
"""

class event():
    def __init__(self,tlsID):
        self.ID=tlsID
        self.currentPhase=traci.trafficlights.getPhase(self.ID)
        self.EventPhase='None'
        self.EventArgPhase=""
        self.EventCycle='None'
        self.EventHistory={}
    def getTime(self):
        return traci.simulation.getCurrentTime()/1000.0
    def _controll(self):

```

```

newPhase=traci.trafficlights.getPhase(self.ID)
if newPhase==self.currentPhase:
    self.EventPhase=0
    self.EventCycle=0
else:
    self.EventPhase=1
    self.EventArgPhase=traci.trafficlights.getPhase(self.ID)
    if self.EventArgPhase==0:
        self.EventCycle=1
    self.eventTime=self.getTime()
    self.currentPhase=traci.trafficlights.getPhase(self.ID)
    self.EventHistory[self.EventArgPhase]=self.eventTime

class queueObserver2():
    def __init__(self,vQueueDet=vQueueDet,tl='TL'):
        self.vQueueDet=vQueueDet
        self.ID=tl
        self.Lvech=7.5
        self.vQueueDetStatus={}
        self._resetVQueueDetStatus()
    def _resetVQueueDetStatus(self):
        for n in self.vQueueDet.keys():
            self.vQueueDetStatus[n]={'vDetAct':-1,'timeAct':-1,'timeRedOn':-1,'timeGreenOn':-
1,'Qo':0.0,'vDetHistAct':-1}
    def getQo(self,Det,MPC):
        q=MPC.SignalGruppLine[Det]['flow']/(3600.0*MPC.SignalGruppLine[Det]['Nlg'])
        Qo=max(0.0,((self.vQueueDet[Det]['Lbq']+self.vQueueDet[Det]['lo'])/self.Lvech-
q*(self.vQueueDetStatus[Det]['timeRedOn']-self.vQueueDetStatus[Det]['timeAct'])))
        #print 'Det=%s, Qo=%s' % (Det,Qo)
        self.vQueueDetStatus[Det]['Qo']=Qo
    def getVQueueDetStatus(self,MPC):
        for n in self.vQueueDet.keys():
            vDetAct=getHaltToPoint(getHaltPoint(self.vQueueDet[n]['laneControl'],self.vQueueDet[n]['Lbq'])
            if not(self.vQueueDetStatus[n]['vDetAct']==vDetAct):
                self.vQueueDetStatus[n]['vDetAct']=vDetAct
            if self.vQueueDetStatus[n]['vDetAct']==1:
                self.vQueueDetStatus[n]['timeAct']=traci.simulation.getCurrentTime()/1000
                self.getQo(n,MPC)
                self.vQueueDetStatus[n]['vDetHistAct']=1
                MPC.SignalGruppLine[n]['Qb']= self.vQueueDetStatus[n]['Qo']*
MPC.SignalGruppLine[n]['Nlg']
    def _observed(self,Event,MPC):
        #if Event.EventCycle==1:
        # self._resetVQueueDetStatus()
        self.getVQueueDetStatus(MPC)
        if Event.EventPhase==1:
            for n in self.vQueueDetStatus.keys():
                if self.vQueueDet[n]['greenPhase']==Event.EventArgPhase:
                    self.vQueueDetStatus[n]['timeGreenOn']=Event.eventTime
                if self.vQueueDet[n]['begRedPhase']==Event.EventArgPhase:
                    self.vQueueDetStatus[n]['timeRedOn']=Event.eventTime
                if Event.EventCycle==1:
                    if not(self.vQueueDetStatus[n]['vDetHistAct']==1):

```

```

        self.vQueueDetStatus[n]['Qo']=0
        MPC.SignalGruppLine[n]['Qb']=0
    else:
        self.vQueueDetStatus[n]['vDetHistAct']=0

class detector():
    def __init__(self,key):
        self.ID=key
        self.AgregateTime=120.0
        self.Windows=5
        self.FlowHistory=[]
        self.SpeedHistory=[]
        self._flow=[]
        self._speed=[]
        self._nextControllPoint=self.AgregateTime
        self.oldVechicle='None'
    def _updateState(self):
        flow=0.0
        if traci.inductionloop.getLastStepVehicleNumber(self.ID)==1:
            vechicle=traci.inductionloop.getLastStepVehicleIDs(self.ID)
            if not(vechicle==self.oldVechicle):
                flow+=1
                self._speed.append(traci.inductionloop.getLastStepMeanSpeed(self.ID))
                self.oldVechicle=vechicle
            self._flow.append(flow)
    def _detControll(self):
        if (traci.simulation.getCurrentTime()/1000)>self._nextControllPoint:
            self.FlowHistory.append(sum(self._flow)/self.AgregateTime*3600.0)
            try:
                self.SpeedHistory.append(sum(self._speed)/len(self._speed))
            except:
                self.SpeedHistory.append(0)
            self._flow=[]
            self._speed=[]
            self._nextControllPoint+=self.AgregateTime
            if self.Windows<len(self.FlowHistory):
                self.FlowHistory.pop(0)
                self.SpeedHistory.pop(0)
        self._updateState()
    def setControll(self,AgregateTime):
        self.FlowHistory.append(sum(self._flow)/AgregateTime*3600.0)
        try:
            self.SpeedHistory.append(sum(self._speed)/len(self._speed))
        except:
            self.SpeedHistory.append(0)
        self._flow=[]
        self._speed=[]
        if self.Windows<len(self.FlowHistory):
            self.FlowHistory.pop(0)
            self.SpeedHistory.pop(0)

```

```

self._updateState()
def getCurrentDensity(self):
    try:
        pos=len(self.FlowHistory)-1
        if pos>=0:
            out=self.FlowHistory[pos]/(self.SpeedHistory[pos]*3.6)
        else:
            out=0.0
    except:
        out=220.0
    return out
def _currentFlow(self):
    try:
        pos=len(self.FlowHistory)-1
        if pos>=0:
            out=self.FlowHistory[pos]
        return out
    except:
        out=-1
    return out
def _currentSpeed(self):
    try:
        pos=len(self.SpeedHistory)-1
        if pos>=0:
            out=self.SpeedHistory[pos]
        return out
    except:
        out=-1
    return out
def setAgregateTime(self,time):
    self.AgregateTime=time
def predictFlow(self):
    if len(self.FlowHistory)<self.Windows:
        return self._currentFlow()
    else:
        windows=np.hamming(self.Windows)
        return sum(np.array(self.FlowHistory)*windows)/windows.sum()
def _clear(self):
    self._flow=[]
    self._speed=[]
    self._nextControllPoint=traci.simulation.getCurrentTime()+self.AgregateTime

class SignalGruppObserver():
    def __init__(self,det_logic):
        self.detKey=traci.inductionloop.getIDList()
        self.Detector={}
        for n in self.detKey:
            self.Detector[n]=detector(n)
        self.SignalGrupp={}
        for n in det_logic.keys():

```

```

self.SignalGrupp[n]={'flow':0.0,'composed':det_logic[n]['composed'],'subtracted':det_logic[n]['subtracted']}
def detectorControll(self):
    for n in self.detKey:
        self.Detector[n]._detControll()
def setAgregateTime(self,time):
    for n in self.detKey:
        self.Detector[n].setAgregateTime(time)
def _getSignalGruppFlow(self):
    flow=0.0
    for n in self.SignalGrupp.keys():
        for m in self.SignalGrupp[n]['composed']:
            flow+=self.Detector[m].predictFlow()
        for m in self.SignalGrupp[n]['subtracted']:
            flow-=self.Detector[m].predictFlow()
        self.SignalGrupp[n]['flow']=flow
    flow=0.0
def getSignalGruppFlow(self):
    self._getSignalGruppFlow()
    out={}
    for n in self.SignalGrupp.keys():
        out[n]=self.SignalGrupp[n]['flow']
    return out

```

```

class controller():
def __init__(self,id,phase,edg):
    self.ID=id
    self.SignalGruppLine={}
    for n in edg.keys():
        self.SignalGruppLine[n]=edg[n]
    self.minTime=13
    self.maxTime=60
    self.phaseDef=phase
    self.baseGrid=setGridControll(self.phaseDef,range(self.minTime,self.maxTime,10))
    self.controllPoint=traci.simulation.getCurrentTime()/1000+getCycleTime(phase)
def _initBaseGrid(self):
    delta=range(self.minTime,self.maxTime,10)
    self.baseGrid=setGridControll(self.phaseDef,delta)
def setPhasesDuration(self):
    TLS=traci.trafficlights.getCompleteRedYellowGreenDefinition(self.ID)
    for n in self.phaseDef.keys():
        TLS[0]._phases[int(n)]._duration=int(self.phaseDef[n]['duration'])*1000
    traci.trafficlights.setCompleteRedYellowGreenDefinition(self.ID,TLS[0])
def getFlow(self,SignalGruppObserver):
    key=self.SignalGruppLine.keys()
    flow=SignalGruppObserver.getSignalGruppFlow()
    for n in key:
        self.SignalGruppLine[n]['flow']=flow[n]
def delayHCM2000(self,PhaseVariation):
    key=PhaseVariation.keys()

```

```

PhaseDef=self.phaseDef
for numVariation in key:
    keyActPhase=PhaseVariation[numVariation].keys()
    for m in keyActPhase:
        PhaseDef[m]['duration']=PhaseVariation[numVariation][m]
    Cycle=getCycleTime(PhaseDef)
    delay={}
    BackOfQueue={}
    InitialQueue={}
    Fin={}
    for n in PhaseDef.keys():
        green=PhaseDef[n]['duration']
        for m in PhaseDef[n]['edg']:
            if not(m=='fixed'):
                flow=self.SignalGruppLine[m]['flow']
                Saturation=self.SignalGruppLine[m]['Capacity']
                Qb=self.SignalGruppLine[m]['Qb']
                Nlg=self.SignalGruppLine[m]['Nlg']

delay[m]=deleyTooll.ControllDeley_HCM2000(Cycle,green,Saturation,flow,Qb,1.0,5*Cycle/3600.0)
*flow*Cycle/3600.0

InitialQueue[m]=deleyTooll.InitialQueue(Cycle,green,flow,Saturation,Qb,5*Cycle/3600.0)
    Fin[m]=getFrontOfStop(flow,green,Cycle,Nlg,InitialQueue[m],7.5)
    PhaseVariation[numVariation]['Delay']=delay
    #PhaseVariation[numVariation]['BackOfQueue']=BackOfQueue
    PhaseVariation[numVariation]['InitialQueue']=InitialQueue
    PhaseVariation[numVariation]['Fin']=Fin
return PhaseVariation
def getBackOfQueue(self,Qo=0.0):
    key=self.SignalGruppLine.keys()
    phase=self.phaseDef.keys()
    out={}
    out2={}
    Cycle=getCycleTime(self.phaseDef)
    for n in phase:
        green=self.phaseDef[n]['duration']
        for m in self.phaseDef[n]['edg']:
            if not(m=='fixed'):
                flow=self.SignalGruppLine[m]['flow']
                Saturation=self.SignalGruppLine[m]['Capacity']
                Nlg=self.SignalGruppLine[m]['Nlg']

out[m]=deleyTooll.BackOfQueue(Cycle,green,flow,Saturation,Nlg,Qo,green/Cycle,Cycle/3600.0)
    out2[m]=deleyTooll.InitialQueue(Cycle,green,flow,Saturation,Qo,Cycle/3600.0)
BQ=%s'%(m,Cycle,green,flow,Saturation,out[m])
    return out,out2
def _findOptControll(self,PhaseVariation):
    PhaseVariation=MPC_controller.delayHCM2000(PhaseVariation)
    key=PhaseVariation.keys()
    for n in key:

```

```

key2=PhaseVariation[n]['Delay'].keys()
sumDelay=0.0
for m in key2:
    sumDelay+=PhaseVariation[n]['Delay'][m]
    if PhaseVariation[n]['Fin'][m]>self.SignalGruppLine[m]['Length']:
        sumDelay+=PhaseVariation[n]['Delay'][m]*6.0 #np.inf
    PhaseVariation[n]['sumDelay']=sumDelay
optControll=PhaseVariation[key[0]]
for n in key:
    if PhaseVariation[n]['sumDelay']<optControll['sumDelay']:
        optControll=PhaseVariation[n]
return optControll
def _setPhaseDef(self,optControll):
    for n in self.phaseDef.keys():
        try:
            self.phaseDef[n]['duration']=optControll[n]
        except:
            a=0
def setOptDelayControll(self,delta=[-5,-3,0,3,5]):
    PhaseVariation=self.baseGrid
    optControll=self._findOptControll(PhaseVariation)
    self._setPhaseDef(optControll)
    PhaseVariation=setGridControll_(self.phaseDef,delta)
    optControll=self._findOptControll(PhaseVariation)
    self._setPhaseDef(optControll)
    self.setPhasesDuration()
    self._initBaseGrid()
    #return optControll
def _controll(self,SignalGruppObserver):
    if self.controllPoint==traci.simulation.getCurrentTime()/1000:
        self.getFlow(SignalGruppObserver)
        self.setOptDelayControll()
        self.controllPoint=traci.simulation.getCurrentTime()/1000+getCycleTime(self.phaseDef)
        #print getCycleTime(self.phaseDef)
def _controll2(self,SignalGruppObserver):
    if self.controllPoint==traci.simulation.getCurrentTime()/1000:
        self.getFlow(SignalGruppObserver)
        Tp=getLostCycleTime(self.phaseDef)
        y={}
        Y={}
        for n in self.SignalGruppLine.keys():
            y[n]=self.SignalGruppLine[n]['flow']/self.SignalGruppLine[n]['Capacity']
        for n in self.phaseDef.keys():
            if self.phaseDef[n]['edg'][0]=='fixed':
                Y[n]=0.0
            else:
                Y[n]=y[self.phaseDef[n]['edg'][0]]
                for m in self.phaseDef[n]['edg']:
                    if Y[n]<y[m]:
                        Y[n]=y[m]
        Ysumm=0.0

```



```

for n in Y.keys():
    Ysumm+=Y[n]
if Y<1.0:
    T=min(160.0,(1.5*Tp+5)/(1.0-Ysumm))
else:
    T=160.0
to={}
for n in Y.keys():
    to[n]=round((T-Tp)*Y[n]/Ysumm)
for n in to.keys():
    if self.phaseDef[n]['edg'][0]=='fixed':
        to[n]='fixed'
    elif to[n]<7.0:
        to[n]=7.0
for n in to.keys():
    if not(to[n]=='fixed'):
        self.phaseDef[n]['duration']==to[n]
self.setPhasesDuration()
print to
self.controllPoint=traci.simulation.getCurrentTime()/1000+getCycleTime(self.phaseDef)

sumoProcess = subprocess.Popen("%s -c %s %s" % (sumoBinary, sumoConfig, queue), shell=True)
traci.init(PORT)
SGO=SignalGruppObserver(det_logic)
MPC_controller=controller('TL',phase,edg)
Event=event('TL')
QueueObserver=queueObserver2()

def simulation(time):
    sim=range(time)
    for n in sim:
        traci.simulationStep()
        SGO.detectorControll()
        MPC_controller._controll(SGO)
        SpeedModification()
        Event._controll()
        QueueObserver._observed(Event,MPC_controller)

simulation(7200)
traci.close()

```

**Файлы конфигурации модели локального пересечения**

Файл конфигурации TraCI\_controll.sumo.cfg программы sumo-gui:

```
<?xml version="1.0" encoding="iso-8859-1"?>

<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/sumoConf
iguration.xsd">

<input>
<net-file value="net_static.net.xml"/>
<route-files value="routes.rou.xml"/>
<additional-files value="det.add.xml
out_Edg.add.xml"/>
</input>

<time>
<begin value="0"/>
<end value="7200"/>
</time>

<report>
<no-duration-log value="true"/>
<no-step-log value="true"/>
</report>

<traci_server>
<remote-port value="8813"/>
</traci_server>

</configuration>
```

Файл расширений out\_Edg.add.xml:

```
<add>
  <edgeData id="dump_900" freq="900"
file="aggregated_900.xml" begin="300" end="7200"/>
</add>
```

## Файл расширений det.add.xml:

```

<sumo-detectors>
<e1Detector id="1_0" lane="1_0" pos="10" freq="900"
file="detector.xml"/>
<e1Detector id="1_1" lane="1_1" pos="10" freq="900"
file="detector.xml"/>
<e1Detector id="3_0" lane="3_0" pos="10" freq="900"
file="detector.xml"/>
<e1Detector id="3_1" lane="3_1" pos="10" freq="900"
file="detector.xml"/>

<e1Detector id="3.300_2" lane="3.300_2" pos="150" freq="900"
file="detector.xml"/>
<e1Detector id="1.300_2" lane="1.300_2" pos="150" freq="900"
file="detector.xml"/>

<e1Detector id="2_0" lane="2_0" pos="10" freq="900"
file="detector.xml"/>
<e1Detector id="2.300_1" lane="2.300_1" pos="30" freq="900"
file="detector.xml"/><e1Detector id="4_0" lane="4_0" pos="10"
freq="900" file="detector.xml"/>
<e1Detector id="4.300_1" lane="4.300_1" pos="30" freq="900"
file="detector.xml"/>

<e1Detector id="2_Fin" lane="2_0" pos="201" freq="900"
file="detector.xml"/>
<e1Detector id="4_Fin" lane="4_0" pos="201" freq="900"
file="detector.xml"/>
<e1Detector id="1_Fin" lane="1_0" pos="251" freq="900"
file="detector.xml"/>
<e1Detector id="3_Fin" lane="3_0" pos="251" freq="900"
file="detector.xml"/>
<e1Detector id="1.300_0" lane="1.300_0" pos="150" freq="900"
file="detector.xml"/>

</sumo-detectors>

```

## Приложение 5

### Результаты моделирования работы базового управления на локальном пересечении

begin	end	id2	sampled Seconds	traveltime	density	occupancy	waitinTime	speed	departed	arrived	entered
300	1200	1	4271,51	26,79	15,9	3,97	6	11,14	160	0	0
300	1200	1,3	11485,46	70,71	43,94	7,32	6333	4,11	0	0	158
300	1200	2	2368,84	30,99	8,82	4,41	0	9,63	75	0	0
300	1200	2,3	4750,78	58,86	60,57	15,14	3440	1,48	0	0	78
300	1200	3	4550,69	26,99	16,94	4,23	2	11,06	170	0	0
300	1200	3,3	12558,92	73,4	48,04	8,01	6950	3,96	0	0	169
300	1200	4	1873,08	24,99	6,97	3,49	0	11,94	75	0	0
300	1200	4,3	4320,99	53,01	55,09	13,77	3126	1,64	0	0	76
1200	2100	1	5672,33	28,21	21,11	5,28	14	10,58	203	0	0
1200	2100	1,3	18375,65	93,75	70,3	11,72	10792	3,1	0	0	203
1200	2100	2	2001,49	27,04	7,45	3,73	0	11,04	75	0	0
1200	2100	2,3	3599,53	46,81	45,89	11,47	2550	1,86	0	0	70
1200	2100	3	5533,35	27,4	20,6	5,15	17	10,89	203	0	0
1200	2100	3,3	17285,13	88,1	66,12	11,02	10420	3,3	0	0	199
1200	2100	4	2728,86	36,35	10,16	5,08	0	8,21	75	0	0
1200	2100	4,3	4993,35	62,34	63,66	15,92	3526	1,4	0	0	75
2100	3000	1	30596,69	129,88	113,89	28,47	10709	2,3	265	0	0
2100	3000	1,3	56370,47	298,4	215,64	35,94	37053	0,97	0	0	205
2100	3000	2	2352,84	30,99	8,76	4,38	0	9,63	75	0	0
2100	3000	2,3	4182,93	51,09	53,33	13,33	2870	1,71	0	0	80
2100	3000	3	19658,08	78,3	73,17	18,29	6481	3,81	270	0	0
2100	3000	3,3	53723,19	259,3	205,52	34,25	36578	1,12	0	0	230
2100	3000	4	1916,68	25,55	7,13	3,57	0	11,68	75	0	0
2100	3000	4,3	3199,68	39,46	40,79	10,2	2136	2,21	0	0	75
3000	3900	1	57146,82	361,88	212,72	53,18	24075	0,82	150	0	0
3000	3900	1,3	61549,18	376,08	235,46	39,24	42001	0,77	0	0	168
3000	3900	2	2736,24	36,59	10,19	5,09	0	8,16	75	0	0
3000	3900	2,3	4311,71	53,31	54,97	13,74	2852	1,63	0	0	74
3000	3900	3	57594,97	369,55	214,39	53,6	18806	0,81	165	0	0
3000	3900	3,3	58935,64	380,03	225,46	37,58	38285,38	0,76	0	0	154

begin	end	id2	sampled Seconds	traveltime	density	occupancy	waitinTime	speed	departed	arrived	entered
3000	3900	4	2490,23	33,31	9,27	4,63	20	8,96	75	0	0
3000	3900	4,3	4422,95	56,28	56,39	14,1	3097	1,55	0	0	75
3900	4800	1	54886,19	290,23	204,3	51,08	15006,61	1,03	191	0	0
3900	4800	1,3	58729,29	327,64	224,67	37,44	38209	0,89	0	0	178
3900	4800	2	1845,31	24,66	6,87	3,43	0	12,11	75	0	0
3900	4800	2,3	3589,35	45,86	45,76	11,44	2579	1,9	0	0	75
3900	4800	3	55948,25	321,92	208,26	52,06	19022	0,93	168	0	0
3900	4800	3,3	60062,2	345,51	229,77	38,29	40593	0,84	0	0	174
3900	4800	4	2659,71	35,49	9,9	4,95	0	8,41	75	0	0
3900	4800	4,3	3872,21	48,91	49,37	12,34	2528	1,78	0	0	75
4800	5700	1	56278,34	336,14	209,49	52,37	26247	0,89	165	0	0
4800	5700	1,3	60133,26	348,15	230,04	38,34	41064	0,83	0	0	167
4800	5700	2	4012,85	40,7	14,94	7,47	605	7,33	100	0	0
4800	5700	2,3	7077,12	76,53	90,23	22,56	5428	1,14	0	0	93
4800	5700	3	56422,49	333,41	210,02	52,51	19225	0,9	169	0	0
4800	5700	3,3	60368,09	357,57	230,94	38,49	41400	0,81	0	0	163
4800	5700	4	2916,32	32,06	10,86	5,43	8	9,31	92	0	0
4800	5700	4,3	5002,98	56,7	63,79	15,95	3591	1,54	0	0	88
5700	6600	1	55729,42	333,7	207,44	51,86	25567	0,89	170	0	0
5700	6600	1,3	60943,24	353,79	233,14	38,86	42213	0,82	0	0	163
5700	6600	2	22959,09	198,08	85,46	42,73	12699,34	1,51	130	0	0
5700	6600	2,3	9954,81	90,53	126,92	31,73	7916	0,96	0	0	104
5700	6600	3	55073,54	312,83	205	51,25	24184,44	0,95	176	0	0
5700	6600	3,3	60058,51	327,36	229,75	38,29	40182	0,89	0	0	174
5700	6600	4	13960,55	119,97	51,97	25,98	6673,38	2,49	125	0	0
5700	6600	4,3	10612,27	99,26	135,3	33,83	8457	0,88	0	0	102
6600	7201	1	37706,48	329,37	210,18	52,55	15930	0,91	112	0	0
6600	7201	1,3	39876,92	351,21	228,44	38,07	26502	0,83	0	0	120
6600	7201	2	18954,97	251,73	105,66	52,83	11433,12	1,19	70	0	0
6600	7201	2,3	6724,37	79,07	128,38	32,1	5182	1,1	0	0	81
6600	7201	3	35559,62	272,86	198,22	49,55	15702	1,09	127	0	0
6600	7201	3,3	40538,35	307,68	232,23	38,71	28020	0,94	0	0	139
6600	7201	4	18037,25	248,05	100,54	50,27	9778,36	1,2	75	0	0
6600	7201	4,3	6420,11	81,22	122,57	30,64	4819	1,07	0	0	75

## Приложение 6

### Результаты моделирования работы предлагаемого метода управления на локальном пересечении

begin	end	id2	sampledSeconds	traveltime	density	occupancy	waitingTime	speed	departed	arrived	entered	left
300	1200	1	4437,38	27,77	16,52	4,13	9	10,75	160	0	0	161
300	1200	1,3	15686,66	95,98	60,01	10	9511	3,03	0	0	161	154
300	1200	2	2468,49	32,5	9,19	4,59	0	9,19	75	0	0	76
300	1200	2,3	4755,85	59,69	60,63	15,16	3245	1,46	0	0	76	76
300	1200	3	5007,77	29,42	18,64	4,66	9	10,15	170	0	0	171
300	1200	3,3	16928,04	100,48	64,76	10,79	10189	2,89	0	0	170	160
300	1200	4	3327,33	44,68	12,39	6,19	282,13	6,68	75	0	0	74
300	1200	4,3	9348,77	124,37	119,19	29,8	7199	0,7	0	0	74	69
1200	2100	1	5407,37	27,13	20,13	5,03	12	11	202	0	0	196
1200	2100	1,3	24281,76	128,23	92,89	15,48	15677	2,27	0	0	196	178
1200	2100	2	2189,65	29,03	8,15	4,08	0	10,28	75	0	0	75
1200	2100	2,3	5158,65	69,34	65,77	16,44	3715	1,26	0	0	75	69
1200	2100	3	5235,27	26,04	19,49	4,87	12	11,46	203	0	0	201
1200	2100	3,3	20152,78	100,02	77,09	12,85	12324	2,9	0	0	202	194
1200	2100	4	4075,82	54,23	15,17	7,59	1216,59	5,5	75	0	0	74
1200	2100	4,3	11476,05	147,15	146,31	36,58	9238	0,59	0	0	74	73
2100	3000	1	8487,1	29,05	31,59	7,9	34	10,28	292	0	0	293
2100	3000	1,3	35899,12	118,12	137,33	22,89	20654	2,46	0	0	293	310
2100	3000	2	8129,61	110,81	30,26	15,13	4091,84	2,69	75	0	0	66
2100	3000	2,3	10442,74	152,46	133,14	33,28	8737	0,57	0	0	66	61
2100	3000	3	7660,92	28,55	28,52	7,13	35	10,45	269	0	0	267
2100	3000	3,3	24277,55	88,16	92,87	15,48	13479	3,29	0	0	267	275
2100	3000	4	8754,73	120,54	32,59	16,29	4794,26	2,48	75	0	0	63
2100	3000	4,3	11064,92	164,83	141,07	35,27	9346	0,53	0	0	63	66
3000	3900	1	9169,96	31,56	34,13	8,53	35	9,46	291	0	0	286
3000	3900	1,3	37580,01	131,19	143,76	23,96	22638	2,21	0	0	285	266
3000	3900	2	5096,51	66,76	18,97	9,49	1604,47	4,47	75	0	0	83
3000	3900	2,3	11335,84	126,54	144,53	36,13	9060	0,69	0	0	83	87
3000	3900	3	7854,07	28,93	29,24	7,31	18	10,32	271	0	0	271
3000	3900	3,3	26775,22	96,41	102,43	17,07	15001	3,01	0	0	271	272

begin	end	id2	sampledSeconds	traveltime	density	occupancy	waitingTime	speed	departed	arrived	entered	left
3000	3900	4	7184,5	94,35	26,74	13,37	3374,94	3,16	75	0	0	82
3000	3900	4,3	9901,84	113,1	126,24	31,56	7892	0,77	0	0	81	82
3900	4800	1	8714,07	29,54	32,44	8,11	19	10,11	295	0	0	299
3900	4800	1,3	36582,9	118,97	139,95	23,32	20512	2,44	0	0	300	307
3900	4800	2	3254,63	43,22	12,11	6,06	869	6,91	75	0	0	77
3900	4800	2,3	10139,5	124,62	129,27	32,32	8519	0,7	0	0	77	75
3900	4800	3	7989,8	29,56	29,74	7,44	29	10,1	270	0	0	269
3900	4800	3,3	19175,18	70,45	73,35	12,23	8929	4,12	0	0	269	262
3900	4800	4	6850,93	91,09	25,5	12,75	3284,44	3,28	75	0	0	72
3900	4800	4,3	10835,11	144,75	138,14	34,54	9126	0,6	0	0	73	69
4800	5700	1	8834,55	30,15	32,88	8,22	21	9,9	293	0	0	294
4800	5700	1,3	39362,85	133,7	150,58	25,1	24338	2,17	0	0	294	282
4800	5700	2	6172,97	64,78	22,98	11,49	1580,58	4,61	100	0	0	83
4800	5700	2,3	9934,92	114,16	126,66	31,67	7892	0,76	0	0	83	84
4800	5700	3	7573,78	30,63	28,19	7,05	12	9,75	247	0	0	249
4800	5700	3,3	29062,32	116,75	111,18	18,53	17433	2,49	0	0	249	240
4800	5700	4	12602,06	143,85	46,91	23,45	6914	2,08	92	0	0	86
4800	5700	4,3	10865,41	121,35	138,53	34,63	8859	0,72	0	0	86	83
5700	6600	1	7753,93	30,5	28,86	7,22	256	9,79	253	0	0	249
5700	6600	1,3	49145,81	202,91	188,01	31,33	34402	1,43	0	0	248	229
5700	6600	2	24312,01	176,51	90,5	45,25	13123,5	1,69	149	0	0	140
5700	6600	2,3	13798,05	96,16	175,92	43,98	11182	0,91	0	0	139	132
5700	6600	3	5530,96	27,8	20,59	5,15	7	10,74	200	0	0	198
5700	6600	3,3	39147,13	197,04	149,76	24,96	28369	1,47	0	0	198	191
5700	6600	4	22093,09	178,53	82,24	41,12	12447	1,67	125	0	0	126
5700	6600	4,3	13605,6	103,19	173,46	43,37	10921	0,84	0	0	125	124
6600	7201	1	9271,5	67,79	51,68	12,92	2825	4,4	140	0	0	131
6600	7201	1,3	46099,52	357,72	264,09	44,01	32710	0,81	0	0	132	126
6600	7201	2	18611,02	237,75	103,74	51,87	8223,43	1,26	79	0	0	70
6600	7201	2,3	11122,89	148,06	212,36	53,09	9116	0,59	0	0	71	74
6600	7201	3	3959,28	29,32	22,07	5,52	19	10,18	133	0	0	137
6600	7201	3,3	31270,46	232,49	179,14	29,86	22190	1,25	0	0	137	128
6600	7201	4	11796,35	147,27	65,76	32,88	4982,36	2,03	83	0	0	71
6600	7201	4,3	10532,34	135,03	201,09	50,27	8519	0,65	0	0	72	77

**Файлы описания транспортной сети**

```
<nodesxmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/nodes_file.xsd"  
>  
  
  <node id="1" x="500" y="500" type="traffic_light"/>  
  <node id="2" x="900" y="500" type="traffic_light"/>  
  <node id="3" x="1300" y="500" type="traffic_light"/>  
  <node id="4" x="1600" y="500" type="traffic_light"/>  
  <node id="5" x="2000" y="500"/>  
  <node id="6" x="500" y="100"/>  
  <node id="7" x="900" y="100"/>  
  <node id="8" x="1300" y="100"/>  
  <node id="9" x="1600" y="100"/>  
  <node id="10" x="500" y="0"/>  
  <node id="11" x="900" y="0"/>  
  <node id="12" x="1300" y="0"/>  
  <node id="13" x="1600" y="0"/>  
  <node id="14" x="500" y="900"/>  
  <node id="15" x="900" y="900"/>  
  <node id="16" x="1300" y="900"/>  
  <node id="17" x="1600" y="900"/>  
  <node id="18" x="500" y="1000"/>  
  <node id="19" x="900" y="1000"/>  
  <node id="20" x="1300" y="1000"/>  
  <node id="21" x="1600" y="1000"/>  
  <node id="22" x="2100" y="500"/>  
  <node id="23" x="100" y="500"/>  
  <node id="24" x="0" y="500"/>  
  
</nodes>
```



```

<edgesxmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/edges_file.xsd
">
    <edge id="1" from="1" to="2" priority="2" numLanes="2"
speed="80.">
<split pos="0" lanes="0 1"/>
<split pos="300" lanes="0 1 2"/>
</edge>
    <edge id="-1" from="2" to="1" priority="2" numLanes="2"
speed="80.">
<split pos="0" lanes="0 1"/>
<split pos="300" lanes="0 1 2"/>
</edge>
    <edge id="2" from="2" to="3" priority="2" numLanes="2"
speed="80.">
<split pos="0" lanes="0 1"/>
<split pos="300" lanes="0 1 2"/>
</edge>
    <edge id="-2" from="3" to="2" priority="2" numLanes="2"
speed="80.">
<split pos="0" lanes="0 1"/>
<split pos="300" lanes="0 1 2"/>
</edge>
    <edge id="3" from="3" to="4" priority="2" numLanes="2"
speed="80.">
<split pos="0" lanes="0 1"/>
<split pos="200" lanes="0 1 2"/>
</edge>
    <edge id="-3" from="4" to="3" priority="2" numLanes="2"
speed="80.">
<split pos="0" lanes="0 1"/>
<split pos="200" lanes="0 1 2"/>
</edge>
    <edge id="4" from="4" to="5" priority="2" numLanes="2"
speed="80."/>
    <edge id="-4" from="5" to="4" priority="2" numLanes="2"
speed="80.">
<split pos="0" lanes="0 1"/>
<split pos="300" lanes="0 1 2"/>
</edge>
    <edge id="5" from="5" to="22" priority="2" numLanes="2"
speed="80."/>
    <edge id="6" from="6" to="1" priority="2" numLanes="2"
speed="80."/>
    <edge id="-6" from="1" to="6" priority="2" numLanes="2"
speed="80."/>
    <edge id="7" from="7" to="2" priority="2" numLanes="2"
speed="80."/>

```

## Продолжение прил. 7

```

    <edge id="-7" from="2" to="7" priority="2" numLanes="2"
speed="80." />
    <edge id="8" from="8" to="3" priority="2" numLanes="2"
speed="80." />
    <edge id="-8" from="3" to="8" priority="2" numLanes="2"
speed="80." />
    <edge id="9" from="9" to="4" priority="2" numLanes="2"
speed="80." />
    <edge id="-9" from="4" to="9" priority="2" numLanes="2"
speed="80." />
    <edge id="-10" from="6" to="10" priority="2" numLanes="2"
speed="80." />
    <edge id="-11" from="7" to="11" priority="2" numLanes="2"
speed="80." />
    <edge id="-12" from="8" to="12" priority="2" numLanes="2"
speed="80." />
    <edge id="-13" from="9" to="13" priority="2" numLanes="2"
speed="80." />
    <edge id="-14" from="1" to="14" priority="2" numLanes="2"
speed="80." />
    <edge id="14" from="14" to="1" priority="2" numLanes="2"
speed="80." />
    <edge id="15" from="15" to="2" priority="2" numLanes="2"
speed="80." />
    <edge id="-15" from="2" to="15" priority="2" numLanes="2"
speed="80." />
    <edge id="16" from="16" to="3" priority="2" numLanes="2"
speed="80." />
    <edge id="-16" from="3" to="16" priority="2" numLanes="2"
speed="80." />
    <edge id="17" from="17" to="4" priority="2" numLanes="2"
speed="80." />
    <edge id="-17" from="4" to="17" priority="2" numLanes="2"
speed="80." />
    <edge id="-18" from="14" to="18" priority="2" numLanes="2"
speed="80." />
    <edge id="-19" from="15" to="19" priority="2" numLanes="2"
speed="80." />
    <edge id="-20" from="16" to="20" priority="2" numLanes="2"
speed="80." />
    <edge id="-21" from="17" to="21" priority="2" numLanes="2"
speed="80." />
    <edge id="22" from="23" to="1" priority="2" numLanes="2"
speed="80." />
<split pos="0" lanes="0 1" />
<split pos="300" lanes="0 1 2" />
</edge>
    <edge id="-22" from="1" to="23" priority="2" numLanes="2"
speed="80." />
    <edge id="-23" from="23" to="24" priority="2" numLanes="2"
speed="80." />

</edges>

```

## Окончание прил. 7

```
<connections xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/connections_file.xsd">
```

```
  <connection from="15" to="-1" fromLane="0" toLane="0"/>
  <connection from="15" to="-7" fromLane="0" toLane="0"/>
  <connection from="15" to="2" fromLane="1" toLane="1"/>

  <connection from="6" to="1" fromLane="0" toLane="0"/>
  <connection from="6" to="-14" fromLane="0" toLane="0"/>
  <connection from="6" to="-22" fromLane="1" toLane="1"/>

  <connection from="14" to="-22" fromLane="0" toLane="0"/>
  <connection from="14" to="-6" fromLane="0" toLane="0"/>
  <connection from="14" to="1" fromLane="1" toLane="1"/>

  <connection from="7" to="2" fromLane="0" toLane="0"/>
  <connection from="7" to="-15" fromLane="0" toLane="0"/>
  <connection from="7" to="-1" fromLane="1" toLane="1"/>

  <connection from="8" to="3" fromLane="0" toLane="0"/>
  <connection from="8" to="-16" fromLane="0" toLane="0"/>
  <connection from="8" to="-2" fromLane="1" toLane="1"/>

  <connection from="16" to="-2" fromLane="0" toLane="0"/>
  <connection from="16" to="-8" fromLane="0" toLane="0"/>
  <connection from="16" to="3" fromLane="1" toLane="1"/>

  <connection from="9" to="4" fromLane="0" toLane="0"/>
  <connection from="9" to="-17" fromLane="0" toLane="0"/>
  <connection from="9" to="-3" fromLane="1" toLane="1"/>

  <connection from="17" to="-3" fromLane="0" toLane="0"/>
  <connection from="17" to="-9" fromLane="0" toLane="0"/>
  <connection from="17" to="4" fromLane="1" toLane="1"/>
```

```
</connections>
```

**Файлы описания транспортного спроса в сети**

```
<?xml version="1.0" encoding="utf-8"?>
<flowdefs>
    <flow id="1" from="22" begin="0" end="7200"
number="5000"/>
    <flow id="2" from="-4" begin="0" end="7200"
number="2000"/>
    <flow id="3" from="6" begin="0" end="7200"
number="1150"/>
    <flow id="4" from="7" begin="0" end="7200"
number="1100"/>
    <flow id="5" from="8" begin="0" end="7200" number="950"/>
    <flow id="6" from="9" begin="0" end="7200"
number="1200"/>
    <flow id="7" from="14" begin="0" end="7200"
number="1150"/>
    <flow id="8" from="15" begin="0" end="7200"
number="1000"/>
    <flow id="9" from="16" begin="0" end="7200"
number="1250"/>
    <flow id="10" from="17" begin="0" end="7200"
number="1200"/>
</flowdefs>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<turns xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/turns_file.xsd"
">
<interval begin="0" end="7200">
    <fromEdge id="6">
        <toEdge id="-14" probability=".55"/>
        <toEdge id="1" probability=".35"/>
        <toEdge id="-22" probability=".10"/>
    </fromEdge>
    <fromEdge id="7">
        <toEdge id="-15" probability=".55"/>
        <toEdge id="2" probability=".35"/>
        <toEdge id="-1" probability=".10"/>
    </fromEdge>
    <fromEdge id="8">
        <toEdge id="-16" probability=".55"/>
        <toEdge id="3" probability=".35"/>
        <toEdge id="-2" probability=".10"/>
    </fromEdge>
    <fromEdge id="9">
        <toEdge id="-17" probability=".55"/>
        <toEdge id="4" probability=".35"/>
        <toEdge id="-3" probability=".10"/>
    </fromEdge>
    <fromEdge id="14">
        <toEdge id="-6" probability=".55"/>
        <toEdge id="-22" probability=".35"/>
        <toEdge id="1" probability=".10"/>
    </fromEdge>
```

```

<fromEdge id="15">
  <toEdge id="-7" probability=".55"/>
  <toEdge id="-1" probability=".35"/>
  <toEdge id="2" probability=".10"/>
</fromEdge>
<fromEdge id="16">
  <toEdge id="-8" probability=".55"/>
  <toEdge id="-2" probability=".35"/>
  <toEdge id="3" probability=".10"/>
</fromEdge>
<fromEdge id="17">
  <toEdge id="-9" probability=".55"/>
  <toEdge id="-3" probability=".35"/>
  <toEdge id="4" probability=".10"/>
</fromEdge>
<fromEdge id="22.300">
  <toEdge id="1" probability=".85"/>
  <toEdge id="-6" probability=".10"/>
  <toEdge id="-14" probability=".05"/>
</fromEdge>
<fromEdge id="1.300">
  <toEdge id="2" probability=".85"/>
  <toEdge id="-7" probability=".10"/>
  <toEdge id="-15" probability=".05"/>
</fromEdge>
<fromEdge id="2.300">
  <toEdge id="3" probability=".85"/>
  <toEdge id="-8" probability=".10"/>
  <toEdge id="-16" probability=".05"/>
</fromEdge>
<fromEdge id="3.200">
  <toEdge id="4" probability=".85"/>
  <toEdge id="-9" probability=".10"/>
  <toEdge id="-17" probability=".05"/>
</fromEdge>
<fromEdge id="-4.300">
  <toEdge id="-3" probability=".85"/>
  <toEdge id="-17" probability=".10"/>
  <toEdge id="-9" probability=".05"/>
</fromEdge>
<fromEdge id="-3.200">
  <toEdge id="-2" probability=".85"/>
  <toEdge id="-16" probability=".10"/>
  <toEdge id="-8" probability=".05"/>
</fromEdge>
<fromEdge id="-2.300">
  <toEdge id="-1" probability=".85"/>
  <toEdge id="-15" probability=".10"/>
  <toEdge id="-7" probability=".05"/>
</fromEdge>
<fromEdge id="-1.300">
  <toEdge id="-22" probability=".85"/>
  <toEdge id="-14" probability=".10"/>
  <toEdge id="-6" probability=".05"/>
</fromEdge>
</interval>
</turns>

```

## Режимы работы светофорных объектов модели SUMO при использовании алгоритма поиска разрыва в потоке

```

<tlLogic id="1" type="actuated" programID="0" offset="0">
  <param key="max-gap" value="3.1"/>
  <param key="passing-time" value="1.9"/>
  <param key="detector-gap" value="2.0"/>
  <phase duration="7" minDur="7" maxDur="51"
state="rrrGGGrrrrGGGr"/>
  <phase duration="3" state="rryyyyrrrryyr"/>
  <phase duration="7" minDur="7" maxDur="20"
state="rrrrrrGrrrrrrG"/>
  <phase duration="3" state="rrrrrryrrrrrry"/>
  <phase duration="7" minDur="7" maxDur="51"
state="GGrrrrrrGGrrrrr"/>
  <phase duration="3" state="yyrrrrrryyrrrrr"/>
  <phase duration="7" minDur="7" maxDur="20"
state="rrGrrrrrrGrrrrr"/>
  <phase duration="3" state="rryrrrrrryrrrrr"/>
</tlLogic>
<tlLogic id="2" type="actuated" programID="0" offset="0">
  <param key="max-gap" value="3.1"/>
  <param key="passing-time" value="1.9"/>
  <param key="detector-gap" value="2.0"/>
  <phase duration="7" minDur="7" maxDur="51"
state="rrrGGGrrrrGGGr"/>
  <phase duration="3" state="rryyyyrrrryyr"/>
  <phase duration="7" minDur="7" maxDur="20"
state="rrrrrrGrrrrrrG"/>
  <phase duration="3" state="rrrrrryrrrrrry"/>
  <phase duration="7" minDur="7" maxDur="51"
state="GGrrrrrrGGrrrrr"/>
  <phase duration="3" state="yyrrrrrryyrrrrr"/>
  <phase duration="7" minDur="7" maxDur="20"
state="rrGrrrrrrGrrrrr"/>
  <phase duration="3" state="rryrrrrrryrrrrr"/>
</tlLogic>
<tlLogic id="3" type="actuated" programID="0" offset="0">
  <param key="max-gap" value="3.1"/>
  <param key="passing-time" value="1.9"/>
  <param key="detector-gap" value="2.0"/>
  <phase duration="7" minDur="7" maxDur="51"
state="rrrGGGrrrrGGGr"/>
  <phase duration="3" state="rryyyyrrrryyr"/>
  <phase duration="7" minDur="7" maxDur="20"
state="rrrrrrGrrrrrrG"/>
  <phase duration="3" state="rrrrrryrrrrrry"/>
  <phase duration="7" minDur="7" maxDur="51"
state="GGrrrrrrGGrrrrr"/>
  <phase duration="3" state="yyrrrrrryyrrrrr"/>
  <phase duration="7" minDur="7" maxDur="20"
state="rrGrrrrrrGrrrrr"/>

```

## Окончание прил. 9

```
<phase duration="3" state="rryrrrrrryrrrr"/>
</tlLogic>
<tlLogic id="4" type="actuated" programID="0" offset="0">
  <param key="max-gap" value="3.1"/>
  <param key="passing-time" value="1.9"/>
  <param key="detector-gap" value="2.0"/>
  <phase duration="7" minDur="7" maxDur="51"
state="rrrGGGrrrrGGGr"/>
  <phase duration="3" state="rrryyyrrrryyr"/>
  <phase duration="7" minDur="7" maxDur="20"
state="rrrrrrGrrrrrrG"/>
  <phase duration="3" state="rrrrrryrrrrrry"/>
  <phase duration="7" minDur="7" maxDur="51"
state="GGrrrrrGGrrrrr"/>
  <phase duration="3" state="yyrrrrryyrrrrr"/>
  <phase duration="7" minDur="7" maxDur="20"
state="rrGrrrrrrrGrrrrr"/>
  <phase duration="3" state="rryrrrrrryrrrr"/>
  <phase duration="25" state="rrrrrrrrrrrrrrrrrrrr"/>
```

</tlLogic>

**Программа расчета режимов работы светофорных объектов в сети  
методом балансировки объемов движения**

```

# -*- coding: utf-8 -*-
from xml.dom import minidom
import numpy as np
import os, sys
from openopt import GLP
from openopt import NLP
sys.path.append("C:\\sumo-0.18.0\\bin\\GM")

"""
----- Привязки и настройки -----
"""
tableOfConformity={}
tableOfConformity['-1.300']='u'-1'
tableOfConformity['-2.300']='-2'
tableOfConformity['-3.200']='-3'
tableOfConformity['-4.300']='-4'
tableOfConformity['1.300']='1'
tableOfConformity['2.300']='2'
tableOfConformity['22.300']='22'
tableOfConformity['3.200']='3'

tls={}
tls['1']={0':35,'1':7,'2':3,'3':25,'4':7,'5':3}
tls['2']={0':30,'1':7,'2':3,'3':30,'4':7,'5':3}
tls['3']={0':30,'1':7,'2':3,'3':30,'4':7,'5':3}
tls['4']={0':20,'1':7,'2':3,'3':20,'4':7,'5':3,'6':20}

edgControll=['-22','1','-1','-14','-6','2','-2','-15','-7','3','-3','-16','-8','4','-17','-9']
edgFlowRestrictions=['1','-1','2','-2','3','-3','17','9']

decodeTlsData={}
decodeTlsData['1']=['0','3']
decodeTlsData['2']=['0','3']
decodeTlsData['3']=['0','3']
decodeTlsData['4']=['0','3']

x0=np.array([90,0.5,0.5,0.5,0.6])
lb=np.ones(x0.size)*0.1
ub=np.ones(x0.size)*0.9
ub[0]=120
lb[0]=60
"""

```



```

----- Объявление функций -----
"""
def decodeSolver(x0):
    global tls, decodeTlsData
    Tp=getTpTime(tls,decodeTlsData)
    key=tl.keys()
    cycleTime=x0[0]
    x=x0[1:]
    key.sort()
    for n in range(x.size):
        to=[]
        to1=round((cycleTime-Tp[key[n]])*x[n])
        to2=round(cycleTime-Tp[key[n]]-to1)
        if to1<7:
            to1=7
            to2=cycleTime-Tp[key[n]]-to1
        elif to2<7:
            to2=7
            to1=cycleTime-Tp[key[n]]-to2
        to=[to1,to2]
        for m in range(len(decodeTlsData[key[n]])):
            tls[key[n]][decodeTlsData[key[n]][m]]=to[m]
    return tls

def parseTurn(doc):
    turnsFlows={}
    interval=doc.getElementsByTagName('interval')
    Edg=interval[0].getElementsByTagName('fromEdge')
    for n in Edg:
        temp=n.getElementsByTagName('toEdge')
        out={}
        for m in temp:
            out[m.getAttribute('id')]=float(m.getAttribute('probability'))
        turnsFlows[n.getAttribute('id')]=out
    return turnsFlows

def Conformity(turnsFlows,tableOfConformity):
    key=tableOfConformity.keys()
    for n in key:
        temp=turnsFlows.pop(n)
        turnsFlows[tableOfConformity[n]]=temp
    return turnsFlows

def parseFlow(doc):
    flows={}
    flowdefs=doc.getElementsByTagName('flowdefs')
    flowRead=flowdefs[0].getElementsByTagName('flow')

```

```

for n in flowRead:
    begin=float(n.getAttribute('begin'))
    end=float(n.getAttribute('end'))
    number=float(n.getAttribute('number'))
    flows[n.getAttribute('from')]=number*3600.0/(end-begin)
return flows

def parseInputFlowToEdg(turn):
    out={}
    key=turn.keys()
    for n in key:
        key2=turn[n].keys()
        for m in key2:
            if out.has_key(m):
                out[m].append(n)
            else:
                out[m]=[n]
    return out

def parseEdgControll(edgDefs):
    edg=edgDefs.getElementsByTagName('edg')
    out={}
    for n in edg:
        out[n.getAttribute('id')]={'capacity':float(n.getAttribute('capacity')), 'tls':n.getAttribute('tls'), 'faza':n.getAttribute('faza')}
    return out

def getCycleTime(tls):
    out={}
    key=tls.keys()
    for n in key:
        faza=tls[n].keys()
        time=0
        for m in faza:
            time+=tls[n][m]
        out[n]=time
    return out

def getTpTime(tls, decodeTlsData):
    out={}
    key=tls.keys()
    for n in key:
        faza=tls[n].keys()
        time=0
        for m in faza:
            if not((decodeTlsData[n][0]==m) or (decodeTlsData[n][1]==m)):

```

```

        time+=tls[n][m]
    out[n]=time
    return out

def getMaxCycleTime(tls):
    cycleTime=getCycleTime(tls)
    key=cycleTime.keys()
    out=cycleTime[key[0]]
    KEY=key[0]
    for n in key:
        if cycleTime[n]>out:
            out=cycleTime[n]
            KEY=n
    return (out,KEY)

def getEdgTlsFlow(edg,tls):
    CycleTime=getCycleTime(tls)
    out={}
    key=edg.keys()
    for n in key:
        out[n]=edg[n]['capacity']*tls[edg[n]['tls']][edg[n]['faza']]/CycleTime[edg[n]['tls']]
    return out

def ObjFunction(x0):
    global net,edg,edgControll
    tls=decodeSolver(x0)
    net._setTlsFlow(edg,tls)
    net.setStateFlow()
    out=0
    for n in edgControll:
        out+=net.Flow[n]
    return out

def flowRestrictions(x0):
    global net,edg,edgFlowRestrictions
    tls=decodeSolver(x0)
    net._setTlsFlow(edg,tls)
    net.setStateFlow()
    out=[]
    for n in edgFlowRestrictions:
        out.append(net.Flow[n]-net.TlsFlow[n])
    return out

```

```

"""
----- Объявление классов -----
"""

class flowNet():
    def __init__(self,turnDef,flowDef):
        self.TurnFlows=turnDef
        self.FlowDef=flowDef
        self.InputFlowToEdg=parseInputFlowToEdg(self.TurnFlows)
    def _setTlsFlow(self,edg,tls):
        self.TlsFlow=getEdgTlsFlow(edg,tls)
    def clearGoList(self):
        out={}
        key=self.InputFlowToEdg.keys()
        for n in key:
            temp={}
            for m in self.InputFlowToEdg[n]:
                temp[m]='None'
            out[n]={'flow':'not success','flow_from':temp}
        self.GoList=out
    def setStatusFlowEdg(self):
        try:
            key=self.GoList.keys()
        except:
            self.clearGoList()
            key=self.GoList.keys()
        for n in key:
            if self.GoList[n]['flow']=='not success':
                flag=0
                key2=self.GoList[n]['flow_from'].keys()
                for m in key2:
                    if self.GoList[n]['flow_from'][m]=='None':
                        flag+=1
                if flag==0:
                    sumFlow=0.0
                    for m in key2:
                        sumFlow+=self.GoList[n]['flow_from'][m]
                    self.GoList[n]['flow']=sumFlow
    def _init_flow(self):
        out={}
        key=self.FlowDef.keys()
        for n in key:
            out[n]=self.FlowDef[n]
        self.Flow=out
        self._FlowList=out
    def _setFlow(self):

```

```

flow=self._FlowList.keys()
for n in flow:
    try:
        toEdg=self.TurnFlows[n]
        for m in toEdg:
            try:
                tempFlow=min(self._FlowList[n],self.TlsFlow[n])
                temp=tempFlow*self.TurnFlows[n][m]
            except:
                temp=self._FlowList[n]*self.TurnFlows[n][m]
            self.GoList[m]['flow_from'][n]=temp
        except:
            out2=n
def _sortGoList(self):
    flag=0
    FlowList={}
    key=self.GoList.keys()
    for n in key:
        if self.GoList[n]['flow']=='not success':
            flag+=1
        else:
            flow=self.GoList.pop(n)
            self.Flow[n]=flow['flow']
            FlowList[n]=flow['flow']
            #print 'edg %s = %s FlowList=%s' % (n,flow['flow'],FlowList)
    self._FlowList=FlowList
    return flag
def setStateFlow(self):
    self._init_flow()
    self.clearGoList()
    while not(self.GoList.keys()==[]):
        self._setFlow()
        self.setStatusFlowEdg()
        self._sortGoList()

"""
----- Расчетрежимаработы -----
"""

Turn=minidom.parse('turns.turns.xml')
Turn=parseTurn(Turn)
TurnFlows = Conformity(Turn,tableOfConformity)
InputFlowToEdg=parseInputFlowToEdg(TurnFlows)

Flow=minidom.parse('flows.flows.xml')
Flow=parseFlow(Flow)

```

## Окончание прил. 10

```
EDG=minidom.parse('C:\\sumo-0.18.0\\bin\\GM\\edgdefs.xml')  
edg=parseEdgControll(EDG)
```

```
CycleTime=getCycleTime(tls)
```

```
net=flowNet(Turn,Flow)  
net._setTlsFlow(edg,tls)  
net.setStateFlow()
```

```
p=GLP(f=ObjFunction,c=flowRestrictions,x0=x0,lb=lb,ub=ub, maxCPUtime = 30)  
r=p.maximize('de')  
tls=decodeSolver(r.xf)
```

## Приложение 11

### Результаты моделирования управления светофорными объектами в сети по поиску разрыва в транспортном потоке

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
300	-1	955	14,65	10,96	2,74	0	19,83	70
300	-1,3	1230,02	25,01	45,33	7,55	581	3,62	55
300	-2	1035,2	14,28	11,88	2,97	0	20,34	72
300	-2,3	949,63	13,69	35	5,83	495	6,61	68
300	-3	978,59	13,96	17,13	4,28	29	13,65	70
300	-3,2	1861,59	26,68	68,6	11,43	944	3,39	67
300	-4	1860,13	23,13	20,88	5,22	138	12,84	0
300	-4,3	3404,79	44,02	125,48	20,91	2462	2,05	79
300	1	1867,28	23,02	21,43	5,36	379	12,62	82
300	1,3	3636,08	44,85	134	22,33	2541	2,02	79
300	2	1764,94	22,27	20,26	5,06	425	13,04	85
300	2,3	2732,4	38,44	100,7	16,78	1955	2,35	70
300	22	2009,62	21,03	22,55	5,64	35	14,12	0
300	22,3	2834,25	32,46	104,45	17,41	1880	2,79	92
300	3	772,16	12,95	13,51	3,38	13	14,7	65
300	3,2	1707,54	50,99	62,93	10,49	1213	1,77	44
600	-1	2067,78	20,44	23,73	5,93	33	14,21	95
600	-1,3	2810,46	25,38	103,57	17,26	1395	3,56	100
600	-2	1752,99	15,19	20,12	5,03	24	19,12	113
600	-2,3	2621,59	23,56	96,61	16,1	1577	3,84	114
600	-3	1602,15	15,9	28,04	7,01	65	11,97	97
600	-3,2	2265	21,49	83,47	13,91	1182	4,21	99
600	-4	2279,94	27,34	25,59	6,4	301	10,86	0
600	-4,3	4097,9	44,85	151,02	25,17	3057	2,02	83
600	1	2998,63	27,8	34,41	8,6	919	10,45	110
600	1,3	4268,39	40,87	157,3	26,22	3011	2,21	102
600	2	3283,76	29,11	37,69	9,42	130	9,98	105
600	2,3	4794,69	40,64	176,7	29,45	3234	2,23	114
600	22	2157,18	21,81	24,21	6,05	59	13,61	0
600	22,3	3869,27	35,25	142,59	23,77	2555	2,57	99
600	3	6916,67	62,53	121,06	30,26	4148	3,05	103
600	3,2	5427,05	45,2	200	33,33	4099	2	113
900	-1	1968,32	20,95	22,59	5,65	192	13,86	91
900	-1,3	2757,11	25,3	101,61	16,93	1554	3,57	100
900	-2	1332,65	18,43	15,29	3,82	148	15,76	72

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
900	-2,3	3815,49	44,71	140,61	23,44	2809	2,02	73
900	-3	961,13	10,65	16,82	4,21	7	17,89	88
900	-3,2	2949,08	40,12	108,68	18,11	2100	2,25	82
900	-4	2059,31	24,75	23,11	5,78	179	12	0
900	-4,3	4327,76	49,52	159,49	26,58	3295	1,83	83
900	1	3574,66	31,29	41,02	10,26	1093	9,28	109
900	1,3	4290,43	41,94	158,11	26,35	2906	2,16	104
900	2	6021,35	56,75	69,1	17,28	2301	5,12	104
900	2,3	5470,75	54,81	201,61	33,6	4102	1,65	102
900	22	2324,22	23,3	26,09	6,52	122	12,75	0
900	22,3	3475,85	31,61	128,09	21,35	2331	2,86	101
900	3	5787,13	60,44	101,29	25,32	2854	3,15	100
900	3,2	5377,25	48,57	198,17	33,03	4309	1,86	96
1200	-1	1522,45	17,5	17,47	4,37	76	16,59	93
1200	-1,3	1964,06	24,8	72,38	12,06	1015	3,65	79
1200	-2	3293,18	30,2	37,79	9,45	1156	9,62	112
1200	-2,3	4070,63	44,88	150,01	25	2848	2,02	88
1200	-3	2442,36	31,62	42,75	10,69	1137	6,02	78
1200	-3,2	3851,95	39,71	141,95	23,66	2858	2,28	81
1200	-4	2293,35	27,89	25,74	6,43	434	10,65	0
1200	-4,3	4437,42	67,35	163,53	27,26	3621	1,34	71
1200	1	4214,32	42,49	48,37	12,09	1961	6,84	98
1200	1,3	4431,5	41,08	163,31	27,22	3041	2,2	95
1200	2	6097,66	51,7	69,98	17,49	2184	5,62	122
1200	2,3	5113,86	42,18	188,46	31,41	3813	2,14	114
1200	22	2257,09	22,57	25,33	6,33	97	13,16	0
1200	22,3	3278,72	32,45	120,83	20,14	2114	2,79	100
1200	3	6460,04	61,37	113,07	28,27	4035	3,1	116
1200	3,2	5755,85	73,3	212,12	35,35	4801	1,23	83
1500	-1	5791,29	41,58	66,46	16,62	1974	6,98	142
1500	-1,3	4789,26	41,31	176,5	29,42	3237	2,19	118
1500	-2	4352	38,62	49,95	12,49	1300	7,52	107
1500	-2,3	3487,69	24,69	128,53	21,42	2059	3,66	130
1500	-3	1695,67	15,89	29,68	7,42	160	11,99	103
1500	-3,2	2693,28	24,7	99,25	16,54	1602	3,66	104
1500	-4	2421,13	28,71	27,17	6,79	364	10,34	0
1500	-4,3	3645,9	36,22	134,36	22,39	2613	2,5	95
1500	1	2813,71	28,64	32,29	8,07	645	10,14	95
1500	1,3	3456,64	28,07	127,39	21,23	2123	3,22	112
1500	2	10991,68	108,54	126,15	31,54	5192	2,68	121
1500	2,3	5849,22	76,06	215,56	35,93	4130	1,19	73



end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
1500	22	2302,27	23,12	25,84	6,46	114	12,85	0
1500	22,3	3765,37	38,79	138,76	23,13	2640	2,33	98
1500	3	6992,99	68,97	122,39	30,6	4114	2,76	79
1500	3,2	5738,27	46,51	211,47	35,25	4456	1,94	118
1800	-1	4430,59	46,26	50,85	12,71	1881	6,28	85
1800	-1,3	4312,59	38,52	158,93	26,49	2985	2,35	101
1800	-2	1850,51	18,32	21,24	5,31	186	15,86	101
1800	-2,3	2782,24	31,77	102,53	17,09	1665	2,85	91
1800	-3	2203,62	23,02	38,57	9,64	822	8,27	91
1800	-3,2	3998,31	39,8	147,35	24,56	2788	2,27	94
1800	-4	2180,58	26,16	24,47	6,12	244	11,35	0
1800	-4,3	3725,06	39,29	137,28	22,88	2681	2,3	84
1800	1	3401,22	30,47	39,03	9,76	830	9,53	115
1800	1,3	3720,79	34,13	137,12	22,85	2138	2,65	106
1800	2	13960,31	127,14	160,21	40,05	7404	2,28	102
1800	2,3	5136,37	37,09	189,29	31,55	3323	2,44	123
1800	22	2418,97	24,33	27,15	6,79	187	12,2	0
1800	22,3	3905,84	38,12	143,94	23,99	2652	2,37	93
1800	3	5596,36	42,19	97,95	24,49	2331	4,51	137
1800	3,2	4969,62	40,25	183,14	30,52	3644	2,25	117
2100	-1	2850,44	26,77	32,71	8,18	786	10,85	107
2100	-1,3	4808,42	36,88	177,2	29,53	3402	2,45	119
2100	-2	3379,26	30,2	38,78	9,7	653	9,62	111
2100	-2,3	3517,13	32,64	129,62	21,6	2114	2,77	98
2100	-3	1240,31	12,78	21,71	5,43	0	14,91	96
2100	-3,2	1794,07	17,93	66,12	11,02	820	5,04	94
2100	-4	2317,96	27,84	26,02	6,5	346	10,67	0
2100	-4,3	3836,23	42,13	141,38	23,56	2855	2,15	84
2100	1	5358,41	46,42	61,5	15,37	1726	6,26	124
2100	1,3	5566,23	64,11	205,13	34,19	3581	1,41	85
2100	2	14520,43	148,45	166,64	41,66	6647	1,96	93
2100	2,3	5740,36	61,8	211,55	35,26	3795	1,46	95
2100	22	2298,7	22,94	25,8	6,45	85	12,95	0
2100	22,3	3233,1	26,73	119,15	19,86	2019	3,38	108
2100	3	8811,02	86,5	154,21	38,55	5671	2,2	90
2100	3,2	5144,75	42,25	189,6	31,6	3895	2,14	113
2400	-1	3452,01	29,69	39,62	9,9	1197	9,78	111
2400	-1,3	4925,3	44,02	181,51	30,25	3557	2,05	113
2400	-2	2306	26,02	26,46	6,62	386	11,16	83
2400	-2,3	3041,28	25,95	112,08	18,68	1799	3,49	106

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
2400	-3	1085,79	11,21	19	4,75	50	17	99
2400	-3,2	2784,53	33,73	102,62	17,1	1793	2,68	88
2400	-4	2245,39	26,95	25,2	6,3	313	11,02	0
2400	-4,3	4958,78	57,95	182,74	30,46	3921	1,56	82
2400	1	12456,56	139	142,96	35,74	5995	2,09	99
2400	1,3	5733,32	55,53	211,29	35,21	3439	1,63	96
2400	2	14559,91	133,21	167,1	41,77	7171	2,18	104
2400	2,3	5398,71	52,78	198,96	33,16	4066	1,71	99
2400	22	2264,91	22,97	25,42	6,35	108	12,93	0
2400	22,3	3209,09	31,35	118,26	19,71	2059	2,89	99
2400	3	5769,94	64,08	100,99	25,25	2740	2,97	102
2400	3,2	5681,65	61,77	209,38	34,9	4672	1,46	80
2700	-1	2928,31	27,72	33,61	8,4	313	10,48	102
2700	-1,3	4977,88	43,34	183,45	30,57	3704	2,09	109
2700	-2	4051,36	36,41	46,5	11,62	742	7,98	109
2700	-2,3	4200,03	39,7	154,78	25,8	2770	2,28	110
2700	-3	2307,46	27,6	40,39	10,1	736	6,9	80
2700	-3,2	3362,12	34,27	123,9	20,65	2270	2,64	85
2700	-4	2061,16	24,86	23,13	5,78	232	11,95	0
2700	-4,3	4705,58	63,58	173,41	28,9	3772	1,42	76
2700	1	12288,73	99,93	141,03	35,26	4678	2,91	104
2700	1,3	5565,17	49,49	205,09	34,18	3560	1,83	115
2700	2	12084,59	109	138,69	34,67	6258	2,66	101
2700	2,3	5269,84	43,77	194,21	32,37	3885	2,07	114
2700	22	2748,89	27,6	30,85	7,71	354	10,76	0
2700	22,3	4252,84	42,32	156,73	26,12	2958	2,14	99
2700	3	8859,33	77,31	155,06	38,76	5769	2,46	116
2700	3,2	5556,94	56,63	204,79	34,13	4428	1,6	99
3000	-1	5441,41	45,08	62,45	15,61	1355	6,44	118
3000	-1,3	5339,63	43,1	196,78	32,8	4027	2,1	119
3000	-2	2959,28	26,91	33,96	8,49	137	10,79	112
3000	-2,3	4175,09	35,41	153,86	25,64	3011	2,55	107
3000	-3	1280,57	12,46	22,41	5,6	11	15,28	96
3000	-3,2	2710,34	25,94	99,88	16,65	1684	3,49	103
3000	-4	2260,09	26,98	25,37	6,34	292	11,01	0
3000	-4,3	3746,39	40,38	138,06	23,01	2777	2,24	91
3000	1	12388,44	154,34	142,18	35,54	6483	1,88	89
3000	1,3	6105,16	79,01	224,99	37,5	4642,81	1,14	73
3000	2	14847,57	189,44	170,4	42,6	7128	1,53	90
3000	2,3	5792,12	80,77	213,46	35,58	3912	1,12	66

<b>end</b>	<b>id2</b>	<b>Sampled Seconds</b>	<b>Travel time</b>	<b>Density</b>	<b>Occupancy</b>	<b>Waiting Time</b>	<b>Speed</b>	<b>Entered</b>
<b>3000</b>	22	3890,43	39,75	43,66	10,92	1131	7,47	0
<b>3000</b>	22,3	5788,1	57,33	213,31	35,55	4300	1,58	98
<b>3000</b>	3	5412,27	56,68	94,73	23,68	2729	3,36	76
<b>3000</b>	3,2	5351,19	43	197,21	32,87	4135	2,1	118
<b>3300</b>	-1	3058,72	28,98	35,1	8,78	558	10,02	111
<b>3300</b>	-1,3	3584,2	33,9	132,09	22,01	2466	2,67	96
<b>3300</b>	-2	1951,47	19,58	22,4	5,6	183	14,83	102
<b>3300</b>	-2,3	3374,9	36,38	124,37	20,73	2319	2,49	87
<b>3300</b>	-3	3212,36	34,48	56,22	14,06	1600	5,52	93
<b>3300</b>	-3,2	4251,75	48,12	156,69	26,11	3081	1,88	82
<b>3300</b>	-4	2287,34	27,46	25,67	6,42	328	10,82	0
<b>3300</b>	-4,3	4608,42	47,8	169,83	28,31	3509	1,89	84
<b>3300</b>	1	16480,19	216,76	189,13	47,28	9327	1,34	77
<b>3300</b>	1,3	5717,91	61,11	210,72	35,12	4035	1,48	80
<b>3300</b>	2	11642,08	103,13	133,61	33,4	5676	2,82	105
<b>3300</b>	2,3	5321,61	39,35	196,12	32,69	3669	2,3	123
<b>3300</b>	22	7719,97	85,77	86,64	21,66	2932	3,46	0
<b>3300</b>	22,3	6162,06	96,35	227,09	37,85	4084	0,94	64
<b>3300</b>	3	4149,45	33,31	72,63	18,16	1450	5,72	124
<b>3300</b>	3,2	3760,51	30,57	138,59	23,1	2572	2,96	116
<b>3600</b>	-1	4738,47	36,59	54,38	13,6	1155	7,94	121
<b>3600</b>	-1,3	4419,48	34,99	162,87	27,14	3045	2,59	125
<b>3600</b>	-2	2747,69	20,54	31,53	7,88	308	14,14	131
<b>3600</b>	-2,3	4028,34	28,97	148,46	24,74	2522	3,12	136
<b>3600</b>	-3	2249,44	20,26	39,37	9,84	388	9,4	107
<b>3600</b>	-3,2	2603,08	20,75	95,93	15,99	1489	4,36	116
<b>3600</b>	-4	2398,05	28,78	26,91	6,73	395	10,32	0
<b>3600</b>	-4,3	4802,62	52,77	176,99	29,5	3779	1,71	83
<b>3600</b>	1	7209,01	60,88	82,73	20,68	2792	4,77	95
<b>3600</b>	1,3	4176,76	26,74	153,93	25,65	2578	3,38	147
<b>3600</b>	2	12424,92	101,64	142,59	35,65	5159	2,86	140
<b>3600</b>	2,3	5408,66	48,04	199,32	33,22	3682	1,88	111
<b>3600</b>	22	14225,77	168,4	159,66	39,92	7894	1,76	0
<b>3600</b>	22,3	5793,11	65,54	213,49	35,58	4160	1,38	84
<b>3600</b>	3	9830,89	98,21	172,06	43,02	5958	1,94	105
<b>3600</b>	3,2	5569,72	51,59	205,26	34,21	4345	1,75	97
<b>3900</b>	-1	3422,1	31,7	39,27	9,82	975	9,16	108
<b>3900</b>	-1,3	4076,07	32,85	150,21	25,04	2551,73	2,75	109
<b>3900</b>	-2	3165,29	32,47	36,33	9,08	1203	8,94	88
<b>3900</b>	-2,3	5181,37	48,13	190,95	31,82	3879	1,88	105

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
3900	-3	1283,59	16,8	22,47	5,62	349	11,34	78
3900	-3,2	3162,43	39,28	116,54	19,42	2261	2,3	75
3900	-4	2195,18	26,56	24,64	6,16	346	11,18	0
3900	-4,3	4409,12	64,16	162,49	27,08	3541	1,41	74
3900	1	10797,89	97,48	123,92	30,98	5516	2,98	123
3900	1,3	5701,83	71,84	210,13	35,02	4315	1,26	85
3900	2	16732,64	212,75	192,03	48,01	9160,58	1,37	65
3900	2,3	5478,49	50,57	201,9	33,65	3957	1,79	99
3900	22	6070,41	57,24	68,13	17,03	2379	5,19	0
3900	22,3	4806,17	35,15	177,12	29,52	2988	2,57	126
3900	3	5545,9	54,48	97,07	24,27	3092	3,5	105
3900	3,2	5971,47	77,18	220,07	36,68	5051	1,17	81
4200	-1	2266,58	25,08	26,01	6,5	457	11,58	87
4200	-1,3	3496,13	32,88	128,84	21,47	2258	2,75	97
4200	-2	2416,53	23,46	27,73	6,93	563	12,38	104
4200	-2,3	4834,94	54,51	178,18	29,7	3725	1,66	85
4200	-3	2642,65	24,22	46,25	11,56	849	7,86	103
4200	-3,2	3832,91	35,16	141,25	23,54	2552	2,57	105
4200	-4	2165,23	25,81	24,3	6,08	223	11,51	0
4200	-4,3	3646,5	38,35	134,38	22,4	2628	2,36	91
4200	1	12979,83	111,62	148,96	37,24	6988	2,6	119
4200	1,3	5439,35	49,64	200,45	33,41	4048	1,82	104
4200	2	9527,63	77,02	109,34	27,34	4107	3,77	100
4200	2,3	6139,7	48,17	226,26	37,71	4590	1,88	128
4200	22	6508,87	66,18	73,05	18,26	2808	4,49	0
4200	22,3	5492,24	48,77	202,4	33,73	3668,89	1,85	100
4200	3	7411,62	61,11	129,72	32,43	4256	3,12	117
4200	3,2	5008,35	40,8	184,57	30,76	3818	2,22	118
4500	-1	3601,51	32,11	41,33	10,33	865	9,05	112
4500	-1,3	2953,98	25,44	108,86	18,14	1692	3,56	110
4500	-2	5580,57	46,76	64,05	16,01	2634	6,21	119
4500	-2,3	5187,97	43,12	191,19	31,87	3847	2,1	114
4500	-3	1156,17	11,99	20,24	5,06	0	15,88	94
4500	-3,2	2134,08	21,07	78,65	13,11	1293	4,29	96
4500	-4	2278,45	27,33	25,57	6,39	318	10,87	0
4500	-4,3	3926,35	41,08	144,7	24,12	2849	2,2	85
4500	1	14389,58	149,43	165,14	41,29	9063	1,94	104
4500	1,3	5929,29	61,64	218,51	36,42	4701	1,47	91
4500	2	13460,03	147,8	154,47	38,62	6348	1,97	103
4500	2,3	6580,26	98,88	242,5	40,42	4882	0,91	63

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
4500	22	2560,7	25,42	28,74	7,18	273	11,68	0
4500	22,3	4359,53	39,03	160,66	26,78	2925	2,32	106
4500	3	6768,53	70,11	118,47	29,62	3988	2,72	77
4500	3,2	5222,78	42,6	192,47	32,08	3966	2,12	116
4800	-1	3682,06	30,36	42,26	10,56	528	9,57	124
4800	-1,3	4439,7	38,99	163,62	27,27	2848	2,32	114
4800	-2	2711,1	28,89	31,11	7,78	798	10,05	89
4800	-2,3	4802,4	40,02	176,98	29,5	3339	2,26	106
4800	-3	1946,15	19,07	34,06	8,52	607	9,99	100
4800	-3,2	4072,56	49,87	150,09	25,01	3094	1,81	88
4800	-4	2333,27	27,99	26,19	6,55	352	10,61	0
4800	-4,3	3996,31	43,17	147,28	24,55	2983	2,1	84
4800	1	15526,39	194,33	178,19	44,55	10406	1,49	59
4800	1,3	5504,42	53,31	202,85	33,81	4082	1,7	90
4800	2	12745,36	115,66	146,27	36,57	7641	2,51	109
4800	2,3	5526,58	47,34	203,67	33,94	4202	1,91	110
4800	22	5202,02	57,6	58,38	14,6	1914	5,16	0
4800	22,3	6105,48	109,23	225	37,5	4406	0,83	63
4800	3	5906,39	52,46	103,38	25,84	2726	3,63	111
4800	3,2	5067,71	41,04	186,76	31,13	3808	2,2	112
5100	-1	3261,59	24,7	37,43	9,36	302	11,76	128
5100	-1,3	3249,4	27,27	119,75	19,96	1966	3,32	119
5100	-2	3136,38	26,58	35,99	9	877	10,93	121
5100	-2,3	4563,96	36,31	168,19	28,03	3257	2,49	116
5100	-3	2055,05	20,86	35,97	8,99	440	9,13	98
5100	-3,2	2949,43	26,77	108,69	18,12	1579	3,38	96
5100	-4	2151,25	25,85	24,14	6,04	227	11,49	0
5100	-4,3	4284,03	48,23	157,88	26,31	3284	1,88	83
5100	1	10285,52	83,8	118,04	29,51	4951	3,47	124
5100	1,3	5082,44	36,43	187,3	31,22	3587	2,48	131
5100	2	12859,64	105,89	147,58	36,9	6919	2,74	135
5100	2,3	5440,89	46,39	200,51	33,42	4058	1,95	111
5100	22	8676,14	84,36	97,38	24,34	4567	3,52	0
5100	22,3	5112,47	41,17	188,41	31,4	3851	2,2	116
5100	3	10390,41	100,5	181,86	45,46	7046	1,9	114
5100	3,2	5719,48	56,2	210,78	35,13	4647	1,61	92
5400	-1	3753,35	32,48	43,08	10,77	810	8,94	112
5400	-1,3	2911,74	23,41	107,31	17,88	1479	3,86	111
5400	-2	2969,71	28,58	34,08	8,52	1011	10,16	96
5400	-2,3	4399,77	38,21	162,14	27,02	3143	2,37	108

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
5400	-3	791,04	11,2	13,85	3,46	0	17,01	70
5400	-3,2	1549,57	18,37	57,11	9,52	928	4,92	76
5400	-4	2047,28	24,89	22,98	5,74	277	11,93	0
5400	-4,3	4377,38	64,67	161,32	26,89	3467	1,4	71
5400	1	16040,61	176,45	184,09	46,02	10045	1,65	98
5400	1,3	6106,45	85,14	225,04	37,51	4849	1,06	77
5400	2	18049,52	259,15	207,14	51,79	11184	1,12	69
5400	2,3	5900,69	57,14	217,46	36,24	4005	1,58	89
5400	22	3331,31	33,54	37,39	9,35	732	8,86	0
5400	22,3	4834,1	45,99	178,15	29,69	3340	1,97	100
5400	3	5824,13	53,26	101,94	25,48	3102	3,58	112
5400	3,2	5719,67	63,12	210,79	35,13	4638	1,43	91
5700	-1	2118,56	23,03	24,31	6,08	57	12,61	90
5700	-1,3	2545,79	22,91	93,82	15,64	1341	3,95	103
5700	-2	2297,58	22,14	26,37	6,59	608	13,12	102
5700	-2,3	3769,66	36,78	138,92	23,15	2643	2,46	102
5700	-3	3168,8	31,7	55,46	13,87	1509	6,01	92
5700	-3,2	4496,39	42,13	165,7	27,62	3252	2,15	102
5700	-4	2144,97	25,44	24,07	6,02	163	11,67	0
5700	-4,3	3751,8	40,62	138,26	23,04	2795	2,23	92
5700	1	16101,35	178,34	184,79	46,2	10641	1,63	81
5700	1,3	5519,48	51,28	203,41	33,9	4235	1,76	102
5700	2	13129,13	129,09	150,68	37,67	7720	2,25	99
5700	2,3	5906,42	64,11	217,67	36,28	4717	1,41	92
5700	22	8923,49	101,9	100,15	25,04	4273	2,91	0
5700	22,3	6028,16	69,26	222,15	37,03	3922	1,31	80
5700	3	5793,37	51,38	101,4	25,35	2930	3,71	93
5700	3,2	5302,17	43,53	195,4	32,57	4116	2,08	120
6000	-1	1964,07	19,03	22,54	5,64	37	15,26	98
6000	-1,3	3072,44	27,06	113,23	18,87	1789	3,34	107
6000	-2	1844,57	18,24	21,17	5,29	197	15,92	100
6000	-2,3	4412,68	47,02	162,62	27,1	3267	1,92	96
6000	-3	2244,62	21,87	39,29	9,82	545	8,71	100
6000	-3,2	3068,34	29,29	113,08	18,85	1859	3,09	100
6000	-4	2288,14	27,43	25,68	6,42	325	10,83	0
6000	-4,3	3709,82	38,89	136,72	22,79	2635	2,33	87
6000	1	13578,65	141,78	155,83	38,96	8250	2,05	89
6000	1,3	5762,76	63,49	212,37	35,4	4558	1,42	86
6000	2	13653,22	124,65	156,69	39,17	7713	2,33	89
6000	2,3	5029,19	40,17	185,34	30,89	3570	2,25	122

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
6000	22	14855,46	194,96	166,73	41,68	8418	1,52	0
6000	22,3	6052,09	82,79	223,04	37,17	4167	1,09	71
6000	3	5714,4	49,02	100,02	25	3176	3,89	121
6000	3,2	4925,78	44,04	181,53	30,25	3645	2,05	106
6300	-1	2083,31	24,01	23,91	5,98	269	12,1	87
6300	-1,3	3407,52	36,55	125,58	20,93	2263	2,47	86
6300	-2	2119,78	23,61	24,33	6,08	562	12,3	89
6300	-2,3	4570,57	52,23	168,44	28,07	3508	1,73	83
6300	-3	854,8	9,43	14,96	3,74	0	20,19	90
6300	-3,2	2578,44	28,78	95,02	15,84	1799	3,14	88
6300	-4	2370,18	28,48	26,6	6,65	379	10,43	0
6300	-4,3	4784,53	50,38	176,32	29,39	3699	1,8	83
6300	1	13204,15	151,97	151,54	37,88	7573	1,91	110
6300	1,3	5583,14	74,52	205,75	34,29	4299	1,21	71
6300	2	15674,91	185,58	179,89	44,97	8182	1,57	82
6300	2,3	6212,76	89,06	228,96	38,16	4671	1,02	67
6300	22	16011,93	190,77	179,71	44,93	8492	1,56	0
6300	22,3	5544,52	58,08	204,33	34,06	4083	1,56	92
6300	3	6702,53	73,31	117,31	29,33	3882	2,6	79
6300	3,2	5482,75	46,05	202,05	33,68	4133	1,96	111
6600	-1	2211,58	23,55	25,38	6,35	466	12,33	97
6600	-1,3	3543,23	42,28	130,58	21,76	2361	2,14	85
6600	-2	2997,39	33,73	34,4	8,6	1104	8,61	91
6600	-2,3	4281,09	44,83	157,77	26,3	2976	2,02	82
6600	-3	2861,16	26,63	50,08	12,52	1175	7,15	105
6600	-3,2	4463,75	47,92	164,5	27,42	3214	1,89	97
6600	-4	2209,95	26,52	24,8	6,2	259	11,2	0
6600	-4,3	4345,83	48,79	160,16	26,69	3325	1,85	84
6600	1	15910,48	162,01	182,6	45,65	9338	1,79	92
6600	1,3	5215,38	43,67	192,2	32,03	3712	2,07	105
6600	2	14808,62	134,48	169,95	42,49	9174	2,16	119
6600	2,3	5388,01	49,57	198,56	33,09	4067	1,82	103
6600	22	14804,7	178,65	166,16	41,54	8235	1,66	0
6600	22,3	5465,66	57,86	201,42	33,57	3892	1,56	90
6600	3	6585,1	68,37	115,26	28,81	3602	2,79	104
6600	3,2	6078,83	61,85	224,02	37,34	4908	1,46	88
6900	-1	3517,73	34,37	40,37	10,09	1320	8,45	98
6900	-1,3	3849,59	41,24	141,87	23,64	2663	2,19	95
6900	-2	7098,5	70,27	81,47	20,37	3375	4,13	109
6900	-2,3	5041,48	49,38	185,79	30,97	3671	1,83	97

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
6900	-3	2072,99	27,85	36,28	9,07	646	6,84	75
6900	-3,2	3749,85	39,43	138,19	23,03	2686	2,29	78
6900	-4	2363,61	28,57	26,53	6,63	442	10,4	0
6900	-4,3	4311,46	61,82	158,89	26,48	3445	1,46	74
6900	1	15277,7	148,76	175,33	43,83	8556	1,95	99
6900	1,3	5739,3	55,17	211,51	35,25	4479	1,64	102
6900	2	13255,94	127,87	152,13	38,03	8134	2,27	88
6900	2,3	5532,34	47,66	203,88	33,98	4045	1,9	110
6900	22	11909,04	120,39	133,66	33,41	5935	2,47	0
6900	22,3	5655,68	61,6	208,43	34,74	4078	1,47	87
6900	3	9298,28	86,69	162,74	40,69	6533	2,2	117
6900	3,2	5938,78	69,01	218,86	36,48	4886	1,31	85
7200	-1	5741,56	48,31	65,89	16,47	2201	6,01	122
7200	-1,3	4451,64	41,38	164,06	27,34	3123	2,19	102
7200	-2	11791,18	100,61	135,32	33,83	5956	2,89	108
7200	-2,3	4890,02	37,76	180,21	30,04	3437	2,4	125
7200	-3	1823,44	18,11	31,91	7,98	569	10,52	97
7200	-3,2	3393,13	35,71	125,05	20,84	2326	2,53	93
7200	-4	2136,73	25,5	23,98	6	204	11,65	0
7200	-4,3	3629,86	37,86	133,77	22,3	2629	2,39	92
7200	1	16004,94	160,12	183,68	45,92	9197	1,81	88
7200	1,3	5632,08	51,72	207,56	34,59	4049	1,75	108
7200	2	15128,85	193,42	173,63	43,41	9817	1,5	106
7200	2,3	6205,41	76,32	228,69	38,11	4704	1,19	69
7200	22	14761,69	193,54	165,68	41,42	9429	1,53	0
7200	22,3	5606,25	70,82	206,61	34,43	4483	1,28	75
7200	3	6228,6	58,45	109,02	27,25	3457	3,26	87
7200	3,2	5063,31	42,62	186,6	31,1	3844	2,12	116
<b>Итого</b>		1970998,18	20261,42			1120182,01		32886



## Приложение 12

### Результаты моделирования работы светофорными объектами в сети по режиму работы, рассчитанным методом балансировки объемов движения

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
300	-1	1025,99	14,1	11,77	2,94	0	20,6	76
300	-1,3	1341,36	19,02	49,43	8,24	769	4,76	68
300	-2	1049,03	14,09	12,04	3,01	0	20,61	74
300	-2,3	1066,78	14,03	39,31	6,55	517	6,45	73
300	-3	699,51	9,48	12,24	3,06	0	20,09	73
300	-3,2	737,28	9,96	27,17	4,53	317	9,08	71
300	-4	1535,29	19,09	17,23	4,31	0	15,56	0
300	-4,3	1978,79	24,21	72,92	12,15	1201	3,74	79
300	1	1310,87	14,05	15,04	3,76	0	20,67	92
300	1,3	1670,47	17,83	61,56	10,26	1048	5,07	90
300	2	1447,37	14,39	16,61	4,15	0	20,18	100
300	2,3	2636,39	27,28	97,16	16,19	1624	3,32	95
300	22	1880,49	19,6	21,11	5,28	0	15,16	0
300	22,3	2081,72	21,27	76,72	12,79	1084	4,25	94
300	3	807,77	9,56	14,14	3,53	0	19,93	87
300	3,2	963,79	11,48	35,52	5,92	482	7,88	80
600	-1	1297,68	14,1	14,89	3,72	0	20,6	87
600	-1,3	2699,28	28,02	99,48	16,58	1846	3,23	94
600	-2	1332,19	14,18	15,29	3,82	0	20,48	92
600	-2,3	1415,23	14,9	52,16	8,69	686	6,07	92
600	-3	763,55	9,65	13,36	3,34	0	19,74	76
600	-3,2	1332,2	16,1	49,1	8,18	730	5,62	78
600	-4	1585,23	19,02	17,79	4,45	0	15,61	0
600	-4,3	2303,94	28,06	84,91	14,15	1361	3,22	83
600	1	1326,37	14,09	15,22	3,81	0	20,62	93
600	1,3	1710,25	17,48	63,03	10,5	947	5,18	94
600	2	1458,62	14,12	16,74	4,18	0	20,58	103
600	2,3	2507,14	22,92	92,4	15,4	1594	3,95	104
600	22	1912,2	19,26	21,46	5,37	3	15,42	0
600	22,3	2939,13	30,43	108,32	18,05	1799	2,97	99
600	3	1029,26	9,64	18,01	4,5	0	19,76	100
600	3,2	3168,81	30,24	116,78	19,46	2103	2,99	106
900	-1	1541,66	14,22	17,69	4,42	0	20,43	110
900	-1,3	2178,66	20,02	80,29	13,38	1238	4,52	102
900	-2	1403,07	13,95	16,1	4,03	0	20,82	99
900	-2,3	1258,87	11,71	46,39	7,73	496	7,72	99
900	-3	853,08	9,68	14,93	3,73	0	19,67	86
900	-3,2	942,98	10,48	34,75	5,79	425	8,63	85
900	-4	1592,18	19,11	17,87	4,47	0	15,54	0
900	-4,3	2456,62	26,57	90,53	15,09	1399	3,4	83
900	1	1625,18	14,07	18,65	4,66	0	20,64	115
900	1,3	1646,89	13,68	60,69	10,12	854	6,61	112

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
900	2	1719,23	14,03	19,73	4,93	0	20,7	120
900	2,3	2071,16	16,53	76,33	12,72	1160	5,47	118
900	22	1947,92	19,56	21,86	5,47	2	15,19	0
900	22,3	3049,74	26,91	112,39	18,73	1774	3,36	99
900	3	1206,01	9,88	21,11	5,28	0	19,28	119
900	3,2	2949,11	22,76	108,68	18,11	1692	3,97	118
1200	-1	1148,71	14,02	13,18	3,3	0	20,71	78
1200	-1,3	2292,83	26,87	84,5	14,08	1555	3,37	84
1200	-2	1038,1	13,93	11,91	2,98	0	20,85	73
1200	-2,3	1314,03	17,51	48,43	8,07	729	5,17	73
1200	-3	592,86	9,61	10,38	2,59	0	19,83	60
1200	-3,2	821,4	12,86	30,27	5,05	436	7,03	61
1200	-4	1584,34	19,02	17,78	4,45	0	15,61	0
1200	-4,3	3097,26	37,36	114,14	19,02	2098	2,42	84
1200	1	1351,81	14,08	15,51	3,88	0	20,62	94
1200	1,3	1267,71	12,91	46,72	7,79	657	7,01	96
1200	2	1479,37	14,04	16,98	4,24	0	20,68	104
1200	2,3	2125,62	18,79	78,34	13,06	1242	4,81	107
1200	22	1919,75	19,34	21,55	5,39	2	15,35	0
1200	22,3	2608,54	27,03	96,13	16,02	1504	3,35	100
1200	3	1074,98	9,69	18,81	4,7	0	19,65	109
1200	3,2	3452,4	31,68	127,23	21,21	2361	2,85	109
1500	-1	1383,45	14,12	15,88	3,97	0	20,57	97
1500	-1,3	1996,18	19,62	73,56	12,26	1120	4,61	93
1500	-2	1237,09	13,97	14,2	3,55	0	20,79	89
1500	-2,3	1267,66	13,79	46,72	7,79	615	6,56	86
1500	-3	870,69	9,6	15,24	3,81	0	19,85	89
1500	-3,2	1100,83	11,93	40,57	6,76	579	7,58	88
1500	-4	1588,55	19,06	17,83	4,46	0	15,58	0
1500	-4,3	2281,68	24,13	84,09	14,01	1279	3,75	83
1500	1	1666,77	14,07	19,13	4,78	0	20,64	116
1500	1,3	2018,64	16,36	74,39	12,4	1170	5,53	115
1500	2	1704,96	14,03	19,57	4,89	0	20,71	119
1500	2,3	1363,4	11,04	50,24	8,37	587	8,19	117
1500	22	1943,45	19,43	21,81	5,45	0	15,29	0
1500	22,3	2477,48	21,75	91,3	15,22	1243	4,16	100
1500	3	1105,19	9,62	19,34	4,84	0	19,79	113
1500	3,2	3164,67	25,91	116,63	19,44	2044	3,49	110
1800	-1	1210,75	14,19	13,9	3,47	0	20,46	82
1800	-1,3	2058,49	23,22	75,86	12,64	1295	3,89	88
1800	-2	1088,07	13,96	12,49	3,12	0	20,81	75
1800	-2,3	1438,49	17,96	53,01	8,84	815	5,04	78
1800	-3	641,49	9,53	11,23	2,81	0	19,98	65
1800	-3,2	1507,14	21,33	55,54	9,26	1029	4,24	66
1800	-4	1578,29	18,96	17,71	4,43	0	15,67	0
1800	-4,3	2542,56	30,71	93,7	15,62	1602	2,95	83
1800	1	1415,95	14,23	16,25	4,06	0	20,41	99
1800	1,3	1912,47	18,58	70,48	11,75	1143	4,87	99
1800	2	1467,01	14,25	16,84	4,21	0	20,38	102

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
1800	2,3	2119,21	19,87	78,1	13,02	1370	4,55	103
1800	22	1904,27	19,37	21,37	5,34	3	15,34	0
1800	22,3	2516,35	26,22	92,73	15,46	1458	3,45	98
1800	3	1137,94	10,06	19,92	4,98	0	18,93	109
1800	3,2	3626,81	32,06	133,66	22,28	2361	2,82	112
2100	-1	1398	14,14	16,04	4,01	0	20,54	103
2100	-1,3	2382,87	23,07	87,82	14,64	1490	3,92	92
2100	-2	1423,37	14,09	16,34	4,08	0	20,62	100
2100	-2,3	1361,01	12,68	50,16	8,36	626	7,14	99
2100	-3	949,35	9,57	16,62	4,15	0	19,91	97
2100	-3,2	1490,48	14,86	54,93	9,15	846	6,09	96
2100	-4	1590,89	19,1	17,86	4,46	0	15,55	0
2100	-4,3	2127,33	22,83	78,4	13,07	1124	3,96	84
2100	1	1737,07	14,13	19,94	4,98	0	20,55	121
2100	1,3	1827,07	14,34	67,33	11,22	976	6,31	119
2100	2	1870,54	14,04	21,47	5,37	0	20,68	130
2100	2,3	1910	13,71	70,39	11,73	914	6,6	130
2100	22	1935,41	19,36	21,72	5,43	0	15,34	0
2100	22,3	2543,95	22,49	93,75	15,63	1328	4,02	100
2100	3	1493,64	10,3	26,14	6,54	0	18,49	144
2100	3,2	3032,73	20,21	111,76	18,63	1671	4,48	138
2400	-1	1265,05	14,33	14,52	3,63	0	20,27	82
2400	-1,3	2738,5	29,83	100,92	16,82	1781	3,03	92
2400	-2	1093,27	14,16	12,55	3,14	0	20,51	77
2400	-2,3	1257,51	15,86	46,34	7,72	659	5,7	76
2400	-3	680,03	9,7	11,9	2,98	0	19,63	68
2400	-3,2	1117,67	15,43	41,19	6,86	632	5,86	69
2400	-4	1584,15	19,02	17,78	4,44	0	15,62	0
2400	-4,3	2602,93	32,22	95,93	15,99	1727	2,81	83
2400	1	1495,92	14,4	17,17	4,29	0	20,18	101
2400	1,3	1693,56	15,89	62,41	10,4	886	5,69	104
2400	2	1410,89	14,11	16,19	4,05	0	20,58	98
2400	2,3	1829,19	17,75	67,41	11,24	1104	5,09	100
2400	22	1925,08	19,45	21,61	5,4	1	15,27	0
2400	22,3	2590,87	26,55	95,48	15,91	1498	3,41	99
2400	3	959	10,21	16,78	4,2	19	18,64	88
2400	3,2	3090,12	30,63	113,88	18,98	2048	2,95	95
2700	-1	1609,36	14,23	18,47	4,62	0	20,42	114
2700	-1,3	2784,27	23,62	102,61	17,1	1696	3,83	107
2700	-2	1476,26	14,03	16,94	4,24	0	20,7	102
2700	-2,3	1297,57	12	47,82	7,97	618	7,54	102
2700	-3	891,97	9,59	15,61	3,9	0	19,87	92
2700	-3,2	1186,43	12,37	43,72	7,29	591	7,31	90
2700	-4	1580,07	18,96	17,73	4,43	0	15,67	0
2700	-4,3	3044,14	32,4	112,18	18,7	1985	2,79	83
2700	1	1701,39	14,19	19,53	4,88	0	20,47	118
2700	1,3	1677,96	13,23	61,84	10,31	777	6,84	117
2700	2	1736,61	14,06	19,93	4,98	0	20,65	122
2700	2,3	1629,87	12,72	60,07	10,01	724	7,11	119

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
2700	22	1948,91	19,57	21,87	5,47	2	15,17	0
2700	22,3	3007,15	26,94	110,82	18,47	1746	3,36	100
2700	3	1347,47	10,11	23,58	5,9	0	18,85	132
2700	3,2	3719,49	28,42	137,07	22,85	2490	3,18	126
3000	-1	1193,34	14,17	13,7	3,42	0	20,49	79
3000	-1,3	2420,28	27,52	89,19	14,87	1541	3,29	87
3000	-2	1088,51	13,97	12,49	3,12	0	20,79	78
3000	-2,3	971,26	12,27	35,79	5,97	493	7,37	77
3000	-3	696,94	9,52	12,2	3,05	0	20,01	70
3000	-3,2	966,53	12,72	35,62	5,94	503	7,11	72
3000	-4	1585,98	19,04	17,8	4,45	0	15,6	0
3000	-4,3	2771,49	33,61	102,14	17,02	1798	2,69	84
3000	1	1149,81	14,14	13,2	3,3	0	20,54	80
3000	1,3	1563,38	18,72	57,61	9,6	907	4,83	81
3000	2	1391,45	14,15	15,97	3,99	0	20,52	97
3000	2,3	2721,77	26,57	100,3	16,72	1798	3,4	98
3000	22	1895,33	19,12	21,27	5,32	1	15,54	0
3000	22,3	4303,48	45,41	158,6	26,43	3080	1,99	99
3000	3	810,04	9,46	14,18	3,54	0	20,14	83
3000	3,2	3969,8	43,39	146,3	24,38	2997	2,08	87
3300	-1	1517,64	14,09	17,42	4,35	0	20,61	108
3300	-1,3	2080,09	18,94	76,66	12,78	1127	4,78	101
3300	-2	1290,17	13,94	14,81	3,7	0	20,84	90
3300	-2,3	1078,64	10,89	39,75	6,63	423	8,31	92
3300	-3	841,66	9,52	14,73	3,68	0	20	87
3300	-3,2	935,37	10,44	34,47	5,75	435	8,67	85
3300	-4	1578,27	18,94	17,71	4,43	0	15,68	0
3300	-4,3	2350,84	24,99	86,64	14,44	1370	3,62	83
3300	1	1771,74	14,21	20,33	5,08	0	20,45	123
3300	1,3	1493,42	11,54	55,04	9,17	656	7,84	121
3300	2	1858,17	14,04	21,33	5,33	0	20,69	130
3300	2,3	3174,04	23,67	116,97	19,5	2141	3,82	128
3300	22	1954,74	19,6	21,94	5,48	0	15,16	0
3300	22,3	4600,05	40,42	169,52	28,25	3198	2,24	100
3300	3	1191,35	9,47	20,85	5,21	0	20,12	122
3300	3,2	3339,01	25,8	123,05	20,51	2170	3,51	121
3600	-1	1188,44	14,37	13,64	3,41	0	20,22	81
3600	-1,3	2081,21	22,84	76,7	12,78	1250	3,96	86
3600	-2	1187,35	14,18	13,63	3,41	0	20,48	83
3600	-2,3	1762,65	22,02	64,96	10,83	1082	4,11	80
3600	-3	700,04	9,76	12,25	3,06	0	19,5	69
3600	-3,2	1166,87	15,64	43	7,17	664	5,78	71
3600	-4	1585,14	19,02	17,79	4,45	0	15,61	0
3600	-4,3	2656,11	32,55	97,88	16,31	1715	2,78	83
3600	1	1317,47	14,3	15,12	3,78	0	20,32	91
3600	1,3	2443,3	26,78	90,04	15,01	1616	3,38	92
3600	2	1240,53	14,21	14,24	3,56	0	20,44	85
3600	2,3	3051,76	32,39	112,47	18,74	2125	2,79	89
3600	22	1925,49	19,4	21,61	5,4	7	15,31	0

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
3600	22,3	3578,66	35,92	131,88	21,98	2407	2,52	99
3600	3	1033,49	10,56	18,09	4,52	0	18,03	96
3600	3,2	3203,89	31,27	118,07	19,68	2129	2,89	97
3900	-1	1858,79	14,28	21,33	5,33	0	20,34	129
3900	-1,3	3931,61	31,01	144,89	24,15	2757	2,92	124
3900	-2	1559,34	14,03	17,9	4,47	0	20,7	108
3900	-2,3	1769,73	14,52	65,22	10,87	848	6,23	111
3900	-3	928,72	9,55	16,25	4,06	0	19,95	96
3900	-3,2	1166,27	11,82	42,98	7,16	564	7,65	93
3900	-4	1584,45	19,01	17,78	4,45	0	15,62	0
3900	-4,3	3066,86	32,64	113,02	18,84	1990	2,77	84
3900	1	1617,16	14,14	18,56	4,64	0	20,54	112
3900	1,3	1676,55	13,65	61,79	10,3	830	6,62	111
3900	2	1777,96	14	20,4	5,1	0	20,75	125
3900	2,3	3142,69	24,3	115,82	19,3	2080	3,72	122
3900	22	1939,79	19,44	21,77	5,44	2	15,27	0
3900	22,3	2464,93	21,88	90,84	15,14	1251	4,13	100
3900	3	1237,04	9,86	21,65	5,41	0	19,32	123
3900	3,2	2549,38	20,19	93,95	15,66	1458	4,48	120
4200	-1	1142,82	14,06	13,12	3,28	0	20,66	77
4200	-1,3	3799,02	43,98	140	23,33	2830	2,06	84
4200	-2	1056,47	14,01	12,12	3,03	0	20,73	76
4200	-2,3	957,99	12,48	35,3	5,88	426	7,25	74
4200	-3	681,73	9,6	11,93	2,98	0	19,84	68
4200	-3,2	1527,94	20,45	56,31	9,38	993	4,42	71
4200	-4	1583,01	19,02	17,77	4,44	0	15,62	0
4200	-4,3	2784,01	34,05	102,6	17,1	1833	2,66	83
4200	1	1354,71	14,11	15,55	3,89	0	20,59	94
4200	1,3	2053	20,81	75,66	12,61	1220	4,35	95
4200	2	1367,66	13,98	15,7	3,92	0	20,78	95
4200	2,3	2414,51	23,29	88,98	14,83	1615	3,88	98
4200	22	1918,5	19,27	21,53	5,38	1	15,41	0
4200	22,3	2804,22	28,32	103,34	17,22	1693	3,19	99
4200	3	1046,61	9,87	18,32	4,58	0	19,29	101
4200	3,2	3321,23	30,54	122,4	20,4	2185	2,96	105
4500	-1	1456,62	14,04	16,72	4,18	0	20,69	106
4500	-1,3	2576,94	24,65	94,97	15,83	1685	3,67	97
4500	-2	1448,71	14,04	16,63	4,16	0	20,68	100
4500	-2,3	1177,88	10,99	43,41	7,23	513	8,23	101
4500	-3	933,85	9,62	16,34	4,09	0	19,8	96
4500	-3,2	1154,48	11,73	42,55	7,09	595	7,71	93
4500	-4	1584,08	19,01	17,78	4,44	0	15,62	0
4500	-4,3	2596,66	27,48	95,69	15,95	1530	3,29	83
4500	1	1603,97	14,17	18,41	4,6	0	20,5	112
4500	1,3	1481,04	12,49	54,58	9,1	678	7,24	110
4500	2	1757,25	14,04	20,17	5,04	0	20,69	124
4500	2,3	1928,56	14,75	71,07	11,85	977	6,13	122
4500	22	1907,5	19,27	21,41	5,35	0	15,42	0
4500	22,3	3193,47	29,21	117,69	19,61	1982	3,1	100

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
4500	3	1416,99	10,35	24,8	6,2	0	18,4	136
4500	3,2	2976,59	20,96	109,7	18,28	1621	4,32	131
4800	-1	1308,9	14,15	15,02	3,76	0	20,52	88
4800	-1,3	3926,26	39,54	144,69	24,12	2828	2,29	95
4800	-2	1268,4	14,2	14,56	3,64	0	20,46	89
4800	-2,3	1531,99	16,98	56,46	9,41	893	5,33	88
4800	-3	670,44	9,55	11,73	2,93	0	19,94	67
4800	-3,2	1033,55	14,08	38,09	6,35	538	6,42	70
4800	-4	1587,29	19,06	17,81	4,45	0	15,58	0
4800	-4,3	2462,63	29,63	90,75	15,13	1516	3,05	84
4800	1	1336,88	14,22	15,34	3,84	0	20,42	91
4800	1,3	1686,48	17,29	62,15	10,36	992	5,23	94
4800	2	1296,97	13,89	14,88	3,72	0	20,91	92
4800	2,3	1618,06	16,66	59,63	9,94	943	5,43	93
4800	22	1911,66	19,24	21,46	5,36	0	15,44	0
4800	22,3	3270,43	33,34	120,52	20,09	2131	2,71	99
4800	3	913,79	9,58	15,99	4	0	19,88	91
4800	3,2	3071,38	31,71	113,19	18,86	2090	2,85	96
5100	-1	1590,32	14,08	18,25	4,56	0	20,63	114
5100	-1,3	2657,66	22,98	97,94	16,32	1663	3,94	107
5100	-2	1462,8	13,99	16,79	4,2	0	20,76	101
5100	-2,3	1306,9	11,72	48,16	8,03	481	7,72	103
5100	-3	868,38	9,59	15,2	3,8	0	19,85	89
5100	-3,2	760,77	8,21	28,04	4,67	252	11,02	87
5100	-4	1587,27	19,07	17,81	4,45	0	15,57	0
5100	-4,3	2256,72	24,21	83,17	13,86	1238	3,74	83
5100	1	1640,67	14,1	18,83	4,71	0	20,6	116
5100	1,3	1528,36	12,69	56,32	9,39	669	7,13	113
5100	2	1780,19	14,03	20,43	5,11	0	20,71	125
5100	2,3	2642,52	20,22	97,38	16,23	1673	4,47	124
5100	22	1908,71	19,44	21,42	5,36	2	15,28	0
5100	22,3	2457,51	22	90,57	15,09	1235	4,11	98
5100	3	1211,35	9,5	21,2	5,3	0	20,04	127
5100	3,2	2717,45	20,45	100,15	16,69	1566	4,42	121
5400	-1	1268,68	14,02	14,56	3,64	0	20,72	87
5400	-1,3	2720,13	28,03	100,24	16,71	1871	3,23	93
5400	-2	1152,02	14,12	13,22	3,31	0	20,57	82
5400	-2,3	1160,26	13,96	42,76	7,13	639	6,48	80
5400	-3	689,89	9,63	12,07	3,02	0	19,78	69
5400	-3,2	1048,78	14,03	38,65	6,44	537	6,45	71
5400	-4	1578,09	18,94	17,71	4,43	0	15,69	0
5400	-4,3	2566,6	31,53	94,59	15,76	1673	2,87	83
5400	1	1398,42	14,33	16,05	4,01	0	20,26	94
5400	1,3	2720,21	26,8	100,25	16,71	1843	3,37	98
5400	2	1437,59	14,13	16,5	4,12	0	20,56	101
5400	2,3	2415,48	22,98	89,02	14,84	1578	3,94	101
5400	22	1906,71	19,19	21,4	5,35	1	15,48	0
5400	22,3	2671,72	27,64	98,46	16,41	1615	3,27	99
5400	3	1041,9	9,68	18,24	4,56	0	19,67	103

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
5400	3,2	3660,82	33,44	134,91	22,49	2539	2,71	108
5700	-1	1436,46	14,04	16,49	4,12	0	20,68	105
5700	-1,3	2158,66	20,84	79,55	13,26	1274	4,34	97
5700	-2	1455,73	14,02	16,71	4,18	0	20,72	100
5700	-2,3	2095,95	19,2	77,24	12,87	1196	4,71	102
5700	-3	914,63	9,6	16,01	4	0	19,84	94
5700	-3,2	980,59	10,22	36,14	6,02	455	8,85	92
5700	-4	1597,83	19,19	17,93	4,48	0	15,48	0
5700	-4,3	2267,76	24,07	83,57	13,93	1216	3,76	83
5700	1	1725,68	14,17	19,8	4,95	0	20,5	121
5700	1,3	2202,53	17,54	81,17	13,53	1344	5,16	118
5700	2	1755,56	14,05	20,15	5,04	0	20,68	121
5700	2,3	2167,18	16,35	79,87	13,31	1152	5,53	123
5700	22	1918,95	19,32	21,54	5,38	1	15,37	0
5700	22,3	2376,33	21,08	87,57	14,6	1170	4,29	100
5700	3	1358,4	10,05	23,78	5,94	0	18,95	133
5700	3,2	2935,74	20,8	108,19	18,03	1691	4,35	130
6000	-1	1264,34	14,34	14,51	3,63	0	20,26	82
6000	-1,3	3284,34	35,73	121,04	20,17	2262	2,53	91
6000	-2	1101,74	14,14	12,64	3,16	0	20,54	79
6000	-2,3	920,51	11,77	33,92	5,65	425	7,68	76
6000	-3	648,45	9,7	11,35	2,84	0	19,64	64
6000	-3,2	831,33	11,72	30,64	5,11	389	7,72	66
6000	-4	1593,83	19,12	17,89	4,47	0	15,53	0
6000	-4,3	2364,72	29,08	87,15	14,52	1417	3,11	84
6000	1	1463,82	14,22	16,8	4,2	0	20,42	101
6000	1,3	2307,16	21,43	85,03	14,17	1377	4,22	103
6000	2	1611,51	14,48	18,49	4,62	0	20,06	111
6000	2,3	2327,28	20,62	85,77	14,29	1383	4,39	109
6000	22	1896,02	19,11	21,28	5,32	0	15,54	0
6000	22,3	2974,11	30,27	109,6	18,27	1880	2,99	99
6000	3	1059,29	9,45	18,54	4,64	0	20,15	107
6000	3,2	3146,68	28,59	115,96	19,33	2183	3,16	111
6300	-1	1516,87	14,12	17,41	4,35	0	20,57	108
6300	-1,3	2106,36	18,71	77,63	12,94	1228	4,83	102
6300	-2	1355,99	14	15,56	3,89	0	20,75	94
6300	-2,3	1103	10,68	40,65	6,77	422	8,47	96
6300	-3	887,99	9,62	15,54	3,89	0	19,8	91
6300	-3,2	1524,16	16,32	56,17	9,36	895	5,54	89
6300	-4	1589,46	19,1	17,84	4,46	0	15,55	0
6300	-4,3	2609,18	27,68	96,16	16,03	1535	3,27	83
6300	1	1788,96	14,29	20,53	5,13	0	20,33	124
6300	1,3	2306,13	19,06	84,99	14,16	1276	4,75	120
6300	2	1627,05	14,2	18,67	4,67	0	20,45	110
6300	2,3	1254,38	10,18	46,23	7,7	474	8,89	114
6300	22	1917,32	19,35	21,52	5,38	2	15,35	0
6300	22,3	2788,82	25,05	102,78	17,13	1563	3,61	99
6300	3	1229,8	9,7	21,52	5,38	0	19,63	123
6300	3,2	2894,03	21,36	106,65	17,78	1684	4,23	124

end	id2	Sampled Seconds	Travel time	Density	Occupancy	Waiting Time	Speed	Entered
6600	-1	1074,45	14,02	12,33	3,08	0	20,71	72
6600	-1,3	2150,58	26,27	79,25	13,21	1411	3,44	79
6600	-2	911,53	13,88	10,46	2,62	0	20,93	63
6600	-2,3	898,47	12,83	33,11	5,52	444	7,05	66
6600	-3	711,21	9,62	12,45	3,11	0	19,79	71
6600	-3,2	1089,57	14,22	40,15	6,69	581	6,36	73
6600	-4	1587,76	19,07	17,82	4,45	0	15,57	0
6600	-4,3	2403,38	29,37	88,57	14,76	1485	3,08	83
6600	1	13887,69	337,95	159,38	39,85	12687	0,86	75
6600	1,3	6844,75	2069,43	252,25	42,04	6619	0,04	5
6600	2	291,27	14,64	3,34	0,84	0	19,84	22
6600	2,3	333,84	17,7	12,3	2,05	141	5,11	19
6600	22	2109,46	21,44	23,68	5,92	119	13,86	0
6600	22,3	3572,52	43,81	131,66	21,94	2513	2,06	89
6600	3	361,06	9,99	6,32	1,58	0	19,06	35
6600	3,2	1952,14	50,47	71,94	11,99	1465	1,79	35
6900	-1	1305,66	14,04	14,98	3,75	0	20,68	94
6900	-1,3	2037,36	21,83	75,08	12,51	1165	4,14	87
6900	-2	1343,82	13,96	15,42	3,86	0	20,81	97
6900	-2,3	1214,22	12,44	44,75	7,46	587	7,27	93
6900	-3	883,72	9,54	15,47	3,87	0	19,95	91
6900	-3,2	866,63	9,19	31,94	5,32	369	9,84	89
6900	-4	1592,99	19,1	17,88	4,47	0	15,55	0
6900	-4,3	2080,01	22,03	76,65	12,78	1039	4,1	84
6900	1	10336,56	67,91	118,63	29,66	2851	4,28	131
6900	1,3	4051,41	20,86	149,31	24,88	2125	4,34	177
6900	2	2582,64	15,19	29,64	7,41	6	19,12	173
6900	2,3	3294,87	20,99	121,43	20,24	1761	4,31	154
6900	22	9709,56	97,24	108,97	27,24	4613	3,05	0
6900	22,3	5413,36	44,03	199,5	33,25	3739	2,05	108
6900	3	1450,47	10,11	25,39	6,35	0	18,83	141
6900	3,2	4135,96	28,97	152,42	25,4	2820	3,12	138
7200	-1	1306,59	14,18	15	3,75	0	20,48	88
7200	-1,3	2372,89	24,4	87,45	14,57	1465	3,71	95
7200	-2	1216,17	14,22	13,96	3,49	0	20,42	82
7200	-2,3	1435,66	16,18	52,91	8,82	795	5,59	85
7200	-3	773,15	9,69	13,53	3,38	0	19,66	77
7200	-3,2	1219,97	14,82	44,96	7,49	638	6,1	79
7200	-4	1584,6	19,01	17,78	4,45	0	15,62	0
7200	-4,3	2450,46	30,29	90,31	15,05	1543	2,99	83
7200	1	3659,87	28,77	42	10,5	470	10,09	108
7200	1,3	4689,25	32,91	172,81	28,8	3033	2,75	136
7200	2	8155,43	56,47	93,6	23,4	2330	5,14	137
7200	2,3	4755,92	33,24	175,27	29,21	3197	2,72	141
7200	22	2022,67	20,3	22,7	5,68	11	14,63	0
7200	22,3	3069,37	30,09	113,11	18,85	1878	3,01	101
7200	3	1260,12	9,52	22,06	5,51	0	20	128
7200	3,2	4954,18	37,72	182,58	30,43	3583	2,4	130
<b>Итого</b>		741722,5	9587,36			289580		32494



## Спецификация модулей TraCI

### Модуль `simulation.py`

Функция: `getCurrentTime()`

Назначение: получить текущее время симуляции

Параметры: отсутствуют

Результат: время, мс

Функция: `getLoadedNumber()`

Назначение: получить число машин, которые скоро появятся в сети (загружаются)

Параметры: отсутствуют

Результат: число машин

Функция: `getLoadedIDList()`

Назначение: получить список идентификаторов машин, которые скоро появятся в сети (загружаются)

Параметры: отсутствуют

Результат: список идентификаторов машин

Функция: `getDepartedNumber()`

Назначение: получить число машин, которые появились в сети на последнем шаге

Параметры: отсутствуют

Результат: число машин

Функция: `getDepartedIDList()`

Назначение: получить список идентификаторов машин, которые появились в сети на последнем шаге

Параметры: отсутствуют

Результат: список идентификаторов машин

Функция: `getArrivedNumber()`

Назначение: получить число машин, которые покинули сеть на последнем шаге

Параметры: отсутствуют

Результат: число машин

Функция: `getArrivedIDList()`

Назначение: получить список идентификаторов машин, которые покинули сеть на последнем шаге

Параметры: отсутствуют

Результат: список идентификаторов машин

Функция: `getStartingTeleportNumber()`

Назначение: получить число машин, которые начали телепортироваться

Параметры: отсутствуют

Результат: число машин

Функция: `getStartingTeleportIDList()`

Назначение: получить список идентификаторов машин, которые начали телепортироваться

Параметры: отсутствуют

Результат: список идентификаторов машин

Функция: `getEndingTeleportNumber()`

Назначение: получить число машин, которые завершили телепортирование

Параметры: отсутствуют

Результат: число машин

Функция: `getEndingTeleportIDList()`

Назначение: получить список идентификаторов машин, которые завершили телепортирование

Параметры: отсутствуют

Результат: список идентификаторов машин

Функция: `getDeltaT()`

Назначение: получение времени изменения шага (?) (исходный код `trac1` вернул ошибку)

Параметры: отсутствуют

Результат: время изменения шага

Функция: `getNetBoundary()`

Назначение: получение границ сети (исходный код `trac1` вернул ошибку)

Параметры: отсутствуют

Результат: границы сети

### Модуль edge.py

Функция: getIDList()

Назначение: получить список идентификаторов дорог

Параметры: отсутствуют

Возвращаемое значение: список дорог

Функция: getAdaptedTraveltime(edgeID, time)

Назначение: получить адаптированное время проезда по дороге

Параметры:

edgeID: идентификатор дороги

time: время

Возвращаемое значение: при разных значениях времени выходило -1: возможно, т.к. необходим был некий составной (сложный) объект (compound object)

Функция: getEffort(edgeID, time)

Назначение: получить объем работы

Параметры:

edgeID: идентификатор дороги

time: время

Возвращаемое значение: при разных значениях времени выходило -1: возможно, т.к. необходим был некий составной (сложный) объект (compound object)

Функция: getCO2Emission(edgeID)

Назначение: посчитать объем выделенного CO2

Параметры:

edgeID: идентификатор дороги

Возвращаемое значение: объем CO2

Функция: getCOEmission(edgeID)

Назначение: посчитать объем выделенного CO

Параметры:

edgeID: идентификатор дороги

Возвращаемое значение: объем CO

Функция: getHCEmission(edgeID)

Назначение: посчитать объем выделенного HC

Параметры:

edgeID: идентификатор дороги

Возвращаемое значение: объем НС в единицу времени

Функция: getPMxEmission(edgeID)

Назначение: посчитать объем выделенного PMx

Параметры:

edgeID: идентификатор дороги

Возвращаемое значение: объем PMx в единицу времени

Функция: getNOxEmission(edgeID)

Назначение: посчитать объем выделенного NOx

Параметры:

edgeID: идентификатор дороги

Возвращаемое значение: объем NOx в единицу времени

Функция: getFuelConsumption(edgeID)

Назначение: посчитать потребление топлива

Параметры:

edgeID: идентификатор дороги

Возвращаемое значение: объем топлива(л) в единицу времени

Функция: getNoiseEmission(edgeID)

Назначение: посчитать шум

Параметры:

edgeID: идентификатор дороги

Возвращаемое значение: уровень шума, дБ

Функция: getLastStepMeanSpeed(edgeID)

Назначение: получить среднюю скорость на последнем шаге

Параметры:

edgeID: идентификатор дороги

Возвращаемое значение: средняя скорость, м/с

Функция: getLastStepOccupancy(edgeID)

Назначение: получить загруженность дороги на последнем шаге

Параметры:

edgeID: идентификатор дороги

Возвращаемое значение: загруженность, %

Функция: getLastStepLength(edgeID)

Назначение: получить длину участка дороги, занятого машинами

Параметры:

edgeID: идентификатор дороги

Возвращаемое значение: длина участка дороги, который занимают машины, м

Функция: getTraveltime(edgeID)

Назначение: узнать время в пути по дороге

Параметры:

edgeID: идентификатор дороги

Возвращаемое значение: время, с

Функция: getLastStepVehicleNumber(edgeID)

Назначение: получить количество машин на последнем шаге на дороге

Параметры:

edgeID: идентификатор дороги

Возвращаемое значение: количество машин

Функция: getLastStepHaltingNumber(edgeID)

Назначение: получить количество машин на последнем шаге на дороге, пребывающих в состоянии покоя

Параметры:

edgeID: идентификатор дороги

Возвращаемое значение: количество машин

Функция: getLastStepVehicleIDs(edgeID)

Назначение: получить список идентификаторов машин на последнем шаге

Параметры:

edgeID: идентификатор дороги

Возвращаемое значение: список идентификаторов

Функция: adaptTraveltime(edgeID, time)

Назначение: установить адаптивное время движения

Параметры:

edgeID: идентификатор дороги

time: время

Возвращаемое значение: отсутствует

Функция: `setEffort(edgeID, effort)`

Назначение: установка объема работы (усилий)

Параметры:

`edgeID`: идентификатор дороги

`effort`: объем работы (усилия)

Возвращаемое значение: отсутствует

Функция: `setMaxSpeed(edgeID, speed)`

Назначение: установка максимальной скорости

Параметры:

`edgeID`: идентификатор дороги

`speed`: скорость

Возвращаемое значение: отсутствует

### Модуль `gui.py`

Функция: `getIDList()`

Назначение: получить список идентификаторов точек обзора дорожной сети

Параметры: отсутствуют

Возвращаемое значение: список идентификаторов точек обзора

Функция: `getZoom(viewID=DEFAULT_VIEW)`

Назначение: получить коэффициент увеличения изображения

Параметры:

`viewID`: идентификатор точки обзора (необязательный)

Возвращаемое значение: коэффициент увеличения изображения (по умолчанию - 100)

Функция: `getOffset(viewID=DEFAULT_VIEW)`

Назначение: получить смещение изображения от точки обзора

Параметры:

`viewID`: идентификатор точки обзора

Возвращаемое значение: список (смещение по X, смещение по Y)

Функция: `getSchema(viewID=DEFAULT_VIEW)`

Назначение: получить схему (о какой схеме говорится - неясно)

Параметры:

`viewID`: идентификатор точки обзора

Возвращаемое значение: схема

Функция: `getBoundary(viewID=DEFAULT_VIEW)`

Назначение: узнать границы

Параметры:

`viewID`: идентификатор точки обзора

Возвращаемое значение: границы (не получилось узнать: ошибка в исходном python коде)

Функция: `setZoom(viewID, zoom)`

Назначение: установить новый коэффициент увеличения изображения

Параметры:

`viewID`: идентификатор точки обзора

`zoom`: коэффициент увеличения изображения

Возвращаемое значение: отсутствует

Функция: `setOffset(viewID, x, y)`

Назначение: установить смещение относительно точки обзора

Параметры:

`viewID`: идентификатор точки обзора

`x`: смещение по x

`y`: смещение по y

Возвращаемое значение: отсутствует

Функция: `setSchema(viewID, schemeName)`

Назначение: установить схему

Параметры:

`viewID`: идентификатор точки обзора

`schemeName`: название схемы

Возвращаемое значение: отсутствует

Функция: `setBoundary(viewID, xmin, ymin, xmax, ymax)`

Назначение: установить границы (не получилось установить: ошибка в исходном python коде)

Параметры:

`viewID`: идентификатор точки обзора

`xmin`: минимальный x

`ymin`: минимальный y

`xmax`: максимальный x

`ymax`: максимальный y

Возвращаемое значение: отсутствует

Функция: screenshot(viewID, filename)

Назначение: сделать снимок экрана и сохранить в файл (не получилось сделать: ошибка в исходном python коде)

Параметры:

viewID: идентификатор точки обзора

filename: файл для снимка

Возвращаемое значение: отсутствует

Функция: trackVehicle(viewID, vehID)

Назначение: проследить за перемещением машины (не получилось проследить: ошибка в исходном python коде)

Параметры:

viewID: идентификатор точки обзора

vehID: идентификатор объекта наблюдения

Возвращаемое значение: отсутствует

### **Модуль inductionloop.py**

Функция: getIDList()

Назначение: получить список индикаторов

Параметры: отсутствуют

Возвращаемое значение: список идентификаторов детекторов

Функция: getLastStepVehicleNumber(loopID)

Назначение: получить число машин, которые проехали за последний шаг мимо детектора

Параметры:

loopID: идентификатор детектора

Возвращаемое значение: число машин

Функция: getLastStepMeanSpeed(loopID)

Назначение: получить среднюю скорость машин, которые проехали за последний шаг мимо детектора

Параметры:

loopID: идентификатор детектора

Возвращаемое значение: средняя скорость, м/с

Функция: getLastStepVehicleIDs(loopID)

Назначение: получить список идентификаторов машин, которые проехали за последний шаг мимо детектора



Параметры:

loopID: идентификатор детектора

Возвращаемое значение: список идентификаторов машин

Функция: getLastStepOccupancy(loopID)

Назначение: узнать загруженность для машин, которые проехали за последний шаг мимо детектора

Параметры:

loopID: идентификатор детектора

Возвращаемое значение: показатель загруженности

Функция: getLastStepMeanLength(loopID)

Назначение: узнать среднюю длину очереди машин, которые проехали за последний шаг мимо детектора

Параметры:

loopID: идентификатор детектора

Возвращаемое значение: средняя длина

Функция: getTimeSinceDetection(loopID)

Назначение: получить время, прошедшее с начала цикла детектора

Параметры:

loopID: идентификатор детектора

Возвращаемое значение: время, с

Функция: getVehicleData(loopID)

Назначение: получить данные о машинах, которые проехали за последний шаг мимо детектора (ошибка в исходных кодах trací)

Параметры:

loopID: идентификатор детектора

Возвращаемое значение: данные о машинах

### **Модуль junction.py**

Функция: getIDList()

Назначение: получить список идентификаторов переходов

Параметры: отсутствуют

Возвращаемое значение: список идентификаторов переходов

Функция: getPosition(junctionID)

Назначение: узнать позицию перехода

Параметры:

junctionID: идентификатор перехода

Возвращаемое значение: позиция в виде списка (x,y)

### Модуль lane.py

Функция: getIDList()

Назначение: получить список идентификаторов полос

Параметры: отсутствуют

Возвращаемое значение: список идентификаторов полос

Функция: getLength(laneID)

Назначение: получить длину полосы

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: длина, м

Функция: getMaxSpeed(laneID)

Назначение: получить максимально допустимую скорость на полосе

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: максимальная скорость, м/с

Функция: getAllowed(laneID)

Назначение: список типов машин, которым разрешается ехать по полосе

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: список типов машин

Функция: getDisallowed(laneID)

Назначение: список типов машин, которым запрещается ехать по полосе

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: список типов машин

Функция: getLinkNumber(laneID)

Назначение: узнать число выездов с полосы

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: число выездов с полосы

Функция: getLinks(laneID)

Назначение:

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: список объектов с полями: идентификатор выезда с полосы, приоритет, открытость, наличие бокового движения (hasFoe)

Функция: getShape(laneID)

Назначение: узнать форму(координаты) полосы

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: список с координатами: [(x,y),(x,y)]

Функция: getEdgeID(laneID)

Назначение: узнать идентификатор дороги, на котором расположена полоса

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: идентификатор дороги

Функция: getCO2Emission(laneID)

Назначение: узнать выделение CO2 на последнем шаге у машин

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: выделение CO2

Функция: getCOEmission(laneID)

Назначение: узнать выделение CO на последнем шаге у машин

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: выделение CO

Функция: getHCEmission(laneID)

Назначение: узнать выделение HC на последнем шаге у машин

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: выделение НС

Функция: getPMxEmission(laneID)

Назначение: узнать выделение PMx на последнем шаге у машин

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: выделение PMx

Функция: getNOxEmission(laneID)

Назначение: узнать выделение NOx на последнем шаге у машин

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: выделение NOx

Функция: getFuelConsumption(laneID)

Назначение: узнать потребление топлива на последнем шаге у машин

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: потребление топлива

Функция: getNoiseEmission(laneID)

Назначение: узнать шум на последнем шаге у машин

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: уровень шума, дБ

Функция: getLastStepMeanSpeed(laneID)

Назначение: узнать среднюю скорость для машин на последнем шаге

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: средняя скорость

Функция: getLastStepOccupancy(laneID)

Назначение: узнать среднюю загрузенность для машин на последнем шаге

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: загрузенность

Функция: getLastStepLength(laneID)

Назначение: узнать среднюю длину пространства, занятого машинами на последнем шаге

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: длина

Функция: getTraveltime(laneID)

Назначение: узнать время движения машин по полосе

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: время

Функция: getLastStepVehicleNumber(laneID)

Назначение: узнать число машин на полосе на последнем шаге

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: число машин

Функция: getLastStepHaltingNumber(laneID)

Назначение: узнать число не движущихся машин на полосе на последнем шаге

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: число не движущихся машин

Функция: getLastStepVehicleIDs(laneID)

Назначение: получить список идентификаторов машин на полосе на последнем шаге

Параметры:

laneID: идентификатор полосы

Возвращаемое значение: список идентификаторов машин

Функция: setAallowed(laneID, disallowedClasses)

Назначение: задать идентификаторы классов машин, которые не могут двигаться по полосе

Параметры:

laneID: идентификатор полосы

disallowedClasses: список идентификаторов классов машин

Возвращаемое значение: отсутствует

Функция: `setDisallowed(laneID, disallowedClasses)`

Назначение: задать идентификаторы классов машин, которые не могут двигаться по полосе

Параметры:

`laneID`: идентификатор полосы

`disallowedClasses`: список идентификаторов классов машин

Возвращаемое значение: отсутствует

Функция: `setMaxSpeed(laneID, speed)`

Назначение: установить максимальную скорость на полосе

Параметры:

`laneID`: идентификатор полосы

`speed`: скорость, м/с

Возвращаемое значение: отсутствует

Функция: `setLength(laneID, length)`

Назначение: установить длину полосы

Параметры:

`laneID`: идентификатор полосы

`length`: длина полосы

Возвращаемое значение: отсутствует

### Модуль `multientryexit.py`

Функция: `getIDList()`

Назначение: получить список идентификаторов детекторов с несколькими выходами

Параметры: отсутствуют

Результат: список идентификаторов детекторов

Функция: `getLastStepVehicleNumber(detID)`

Назначение: узнать число машин, которые проехали мимо детектора на последнем шаге

Параметры:

`detID`: идентификатор детектора

Результат: число машин

Функция: `getLastStepMeanSpeed(detID)`

Назначение: узнать среднюю скорость машин, которые проехали мимо детектора на последнем шаге

Параметры:

detID: идентификатор детектора

Результат: средняя скорость

Функция: getLastStepVehicleIDs(detID)

Назначение: узнать идентификаторы машин, которые проехали мимо детектора на последнем шаге

Параметры:

detID: идентификатор детектора

Результат: список идентификаторов машин

Функция: getLastStepHaltingNumber(detID)

Назначение: узнать число машин, которые остановились у детектора на последнем шаге

Параметры:

detID: идентификатор детектора

Результат: число не движущихся машин

### **Модуль poi.py**

Функция: getIDList()

Назначение: получить список идентификаторов достопримечательностей (point of interest(poi))

Параметры: отсутствуют

Возвращаемое значение: список идентификаторов poi

Функция: getType(poiID)

Назначение: узнать тип poi

Параметры:

poiID: идентификатор poi

Возвращаемое значение: тип poi

Функция: getPosition(poiID)

Назначение: позиция poi

Параметры:

poiID: идентификатор poi

Возвращаемое значение: позиция poi в виде списка: (x,y)

Функция: getColor(poiID)

Назначение: узнать цвет poi

Параметры:

poiID: идентификатор poi

Возвращаемое значение: цвет в виде списка RGB(R,G,B,AlphaMask)

Функция: setType(poiID, poiType)

Назначение: задать тип poi

Параметры:

poiID: идентификатор poi

poiType: тип poi ("source", "sink", "between", "discarded")

Возвращаемое значение: отсутствует

Функция: setPosition(poiID, x, y)

Назначение: задать позицию poi

Параметры:

poiID: идентификатор poi

x: x

y: y

Возвращаемое значение: отсутствует

Функция: setColor(poiID, color)

Назначение: установить цвет для poi

Параметры:

poiID: идентификатор poi

color: цвет в виде списка RGB(R,G,B,AlphaMask)

Возвращаемое значение: отсутствует

Функция: add(poiID, x, y, color, poiType="", layer=0)

Назначение: добавить poi

Параметры:

poiID: идентификатор poi

x: x

y: y

color: цвет в виде списка RGB(R,G,B,AlphaMask)

poiType: тип poi ("source", "sink", "between", "discarded")

layer: слой

Возвращаемое значение: отсутствует

Функция: remove(poiID, layer=0)

Назначение: удаление poi из слоя



Параметры:

roiID: идентификатор roi

layer: слой, с которого удалять

Возвращаемое значение: отсутствует

**Модуль polygon.py**

Функция: getIDList()

Назначение: вернуть список идентификаторов полигонов

Параметры: отсутствуют

Результат: список идентификаторов полигонов

Функция: getType(polygonID)

Назначение: получить тип полигона

Параметры:

polygonID: идентификатор полигона

Результат: тип полигона

Функция: getShape(polygonID)

Назначение: получить форму полигона

Параметры:

polygonID: идентификатор полигона

Результат: форма полигона в виде списка точек ((X1,Y1),(X2,Y2),..)

Функция: getColor(polygonID)

Назначение: получить цвет полигона

Параметры:

polygonID: идентификатор полигона

Результат: цвет полигона в виде списка RGB(R,G,B,AlphaMask)

Функция: setType(polygonID, polygonType)

Назначение: установить тип полигона

Параметры:

polygonID: идентификатор полигона

polygonType: тип полигона("waterway","natural.water", etc.: стр 46

руководства)

layer:

Результат: отсутствует

Функция: `setShape(polygonID, shape)`

Назначение: задать форму полигона

Параметры:

`polygonID`: идентификатор полигона

`shape`: форма полигона в виде списка точек `((X1,Y1),(X2,Y2),..)`

Результат: отсутствует

Функция: `setColor(polygonID, color)`

Назначение: задать цвет полигона

Параметры:

`polygonID`: идентификатор полигона

`color`: цвет полигона в виде списка `RGB(R,G,B,AlphaMask)`

Результат: отсутствует

Функция: `add(polygonID, shape, color, fill=False, polygonType="", layer=0)`

Назначение: добавить полигон

Параметры:

`polygonID`: идентификатор полигона

`shape`: форма полигона в виде списка точек `((X1,Y1),(X2,Y2),..)`

`color`: цвет полигона в виде списка `RGB(R,G,B,AlphaMask)`

`fill`: необходимость закрасить полигон

`polygonType`: тип ("waterway", "natural.water", etc.: стр 46 руководства)

`layer`: слой

Результат: отсутствует

Функция: `remove(polygonID, layer=0)`

Назначение: удалить полигон

Параметры:

`polygonID`: идентификатор полигона

`layer`: слой

Результат: отсутствует

## Модуль `route.py`

Функция: `getIDList()`

Назначение: получить список идентификаторов маршрутов

Параметры: отсутствуют

Возвращаемое значение: список идентификаторов маршрутов

Функция: `getEdges(routeID)`

Назначение: получить дороги маршрута

Параметры:

`routeID`: идентификатор пути

Возвращаемое значение: список дорог маршрута

Функция: `add(routeID, edges)`

Назначение: добавить маршрут

Параметры:

`routeID`: идентификатор маршрута

`edges`: список дорог

Возвращаемое значение: отсутствует

### **Модуль `trafficlights.py`**

Функция: `getIDList()`

Назначение: получить список идентификаторов светофоров

Параметры: отсутствуют

Результат: список идентификаторов светофоров

Функция: `getRedYellowGreenState(tlsID)`

Назначение: получить состояние светофора

Параметры:

`tlsID`: идентификатор светофора

Результат: состояние светофора

Функция: `getCompleteRedYellowGreenDefinition(tlsID)`

Назначение: получить полное описание светофора

Параметры:

`tlsID`: идентификатор светофора

Результат: полное описание светофора

Функция: `getControlledLanes(tlsID)`

Назначение: получить список идентификаторов полос, контролируемых светофором

Параметры:

`tlsID`: идентификатор светофора

Результат: список идентификаторов полос

Функция: `getControlledLinks(tlsID)`

Назначение: получить список связей, контролируемых светофором

Параметры:

`tlsID`: идентификатор светофора

Результат: список связей

Функция: `getProgram(tlsID)`

Назначение: узнать идентификатор программы (логики) светофора

Параметры:

`tlsID`: идентификатор светофора

Результат: идентификатор программы

Функция: `getPhase(tlsID)`

Назначение: определить номер фазы

Параметры:

`tlsID`: идентификатор светофора

Результат: номер фазы

Функция: `getNextSwitch(tlsID)`

Назначение: определить время до следующего переключения

Параметры:

`tlsID`: идентификатор светофора

Результат: время, мс

Функция: `setRedYellowGreenState(tlsID, state)`

Назначение:

Параметры:

`tlsID`: идентификатор светофора

`state`: состояние светофора

Результат: отсутствует

Функция: `setPhase(tlsID, index)`

Назначение: установить фазу с номером `index` на светофоре

Параметры:

`tlsID`: идентификатор светофора

`index`: номер фазы

Результат: отсутствует

Функция: setProgram(tlsID, programID)

Назначение: задать программу (логику) работы светофора

Параметры:

tlsID: идентификатор светофора

programID: идентификатор логики светофора

Результат: отсутствует

Функция: setPhaseDuration(tlsID, phaseDuration)

Назначение: установить длительность текущей фазы

Параметры:

tlsID: идентификатор светофора

phaseDuration: длительность фазы, мс

Результат: отсутствует

Функция: setCompleteRedYellowGreenDefinition(tlsID, tls)

Назначение: задать описание работы светофора (аналогичное тому, что возвращает getCompleteRedYellowGreenDefinition(tlsID))

Параметры:

tlsID: идентификатор светофора

tls: полное описание работы семафора

Результат: отсутствует

### **Модуль vehicle.py**

Функция: getIDList()

Назначение: получить список идентификаторов машин

Параметры: отсутствуют

Возвращаемое значение: список идентификаторов машин

Функция: getSpeed(vehID)

Назначение: получить скорость машины

Параметры:

vehID: идентификатор машины

Возвращаемое значение: скорость машины, м/с

Функция: getPosition(vehID)

Назначение:

Параметры: узнать положение машины

vehID: идентификатор машины

Возвращаемое значение: положение машины в виде списка (x,y)

Функция: `getAngle(vehID)`

Назначение: узнать угол направления движения машины

Параметры:

`vehID`: идентификатор машины

Возвращаемое значение: угол, в градусах

Функция: `getRoadID(vehID)`

Назначение: узнать идентификатор дороги, по которой движется машина

Параметры:

`vehID`: идентификатор машины

Возвращаемое значение: идентификатор дороги

Функция: `getLaneID(vehID)`

Назначение: узнать идентификатор полосы, по которой движется машина

Параметры:

`vehID`: идентификатор машины

Возвращаемое значение: идентификатор полосы

Функция: `getLaneIndex(vehID)`

Назначение: узнать индекс полосы

Параметры:

`vehID`: идентификатор машины

Возвращаемое значение: индекс полосы

Функция: `getTypeID(vehID)`

Назначение: узнать идентификатор типа машины

Параметры:

`vehID`: идентификатор машины

Возвращаемое значение: идентификатор типа машины

Функция: `getRouteID(vehID)`

Назначение: получить идентификатор пути

Параметры:

`vehID`: идентификатор машины

Возвращаемое значение: идентификатор маршрута

Функция: `getRoute(vehID)`

Назначение: узнать маршрут

Параметры:

`vehID`: идентификатор машины

Возвращаемое значение: маршрут

Функция: `getLanePosition(vehID)`

Назначение: узнать положение полосы по которой движется машина

Параметры:

`vehID`: идентификатор машины

Возвращаемое значение: список из 2-х точек, задающих полосу

Функция: `getColor(vehID)`

Назначение: узнать цвет машины

Параметры:

`vehID`: идентификатор машины

Возвращаемое значение: цвет машины RGB (R,G,B,AlphaMask)

Функция: `getBestLanes(vehID)`

Назначение: узнать лучший маршрут по полосам

Параметры:

`vehID`: идентификатор машины

Возвращаемое значение: список (текущая полоса, X нач.полосы, Y нач. полосы, X кон. полосы, Y кон. полосы, (оставшиеся полосы маршрута, включая текущую))

Функция: `setMaxSpeed(vehID, speed)`

Назначение: задать максимальную скорость для машины

Параметры:

`vehID`: идентификатор машины

`speed`: скорость, м/с

Возвращаемое значение: отсутствует

Функция: `setSpeed(vehID, speed)`

Назначение: задать скорость машины

Параметры:

`vehID`: идентификатор машины

`speed`: скорость, м/с

Возвращаемое значение: отсутствует

Функция: setLanePosition(vehID, position)

Назначение: задать положение машины на полосе

Параметры:

vehID: идентификатор машины

position: положение (x,y)

Возвращаемое значение: отсутствует

Функция: setStop(vehID, edgeID, pos=1., laneIndex=0, duration=2\*\*31-1)

Назначение: задать остановку машины

Параметры:

vehID: идентификатор машины

edgeID: идентификатор дороги

pos: место остановки

laneIndex: индекс полосы

duration: длительность остановки, мс

Возвращаемое значение: отсутствует

Функция: changeLane(vehID, laneIndex, duration)

Назначение: изменить полосу движения

Параметры:

vehID: идентификатор машины

laneIndex: индекс полосы

duration: длительность пребывания на полосе (?)

Возвращаемое значение: отсутствует

Функция: slowDown(vehID, speed, duration)

Назначение: затормозить

Параметры:

vehID: идентификатор машины

speed: скорость, м/с

duration: длительность торможения (?)

Возвращаемое значение: отсутствует

Функция: changeTarget(vehID, edgeID)

Назначение: изменить дорогу

Параметры:

vehID: идентификатор машины

edgeID: идентификатор дороги

Возвращаемое значение: отсутствует



Функция: `setRoute(vehID, edgeList)`

Назначение: задать маршрут

Параметры:

vehID: идентификатор машины

edgeList: список дорог

Возвращаемое значение: отсутствует

Функция: `moveTo(vehID, laneID, pos)`

Назначение: двигаться к определенной точке на полосе (? : ошибка в исходном коде `trac1`)

Параметры:

vehID: идентификатор машины

laneID: идентификатор полосы

pos: положение на полосе

Возвращаемое значение: отсутствует

Функция: `reroute(vehID)`

Назначение: пересчитать маршрут (?)

Параметры:

vehID: идентификатор машины

Возвращаемое значение: отсутствует

Функция: `setColor(vehID, color)`

Назначение: установить цвет

Параметры:

vehID: идентификатор машины

color: цвет машины RGB (R,G,B,AlphaMask)

Возвращаемое значение: отсутствует

Функция: `add(vehID, routeID, depart=-1, pos=0, speed=0, lane=0, typeID="DEFAULT_VEHICLE")`

Назначение: добавить машину

Параметры:

vehID: идентификатор машины

routeID: идентификатор маршрута

depart: время отправления

pos: положение с которого, происходит отправление

speed: скорость, м/с

lane: полоса

typeID: идентификатор типа транспортного средства

Возвращаемое значение: отсутствует

### Модуль `vehicletype.py`

Функция: `getIDList()`

Назначение: узнать список идентификаторов типов машин

Параметры: отсутствуют

Результат: список идентификаторов типов машин

Функция: `getLength(typeID)`

Назначение: узнать длину машины определенного типа

Параметры:

`typeID`: идентификатор типа машины

Результат: длина машины

Функция: `getMaxSpeed(typeID)`

Назначение: узнать максимальную скорость машины определенного типа

Параметры:

`typeID`: идентификатор типа машины

Результат: максимальная скорость, м/с

Функция: `getAccel(typeID)`

Назначение: узнать ускорение машины определенного типа

Параметры:

`typeID`: идентификатор типа машины

Результат: ускорение

Функция: `getDecel(typeID)`

Назначение: узнать торможение машины определенного типа

Параметры:

`typeID`: идентификатор типа машины

Результат: торможение

Функция: `getTau(typeID)`

Назначение: узнать показатель мастерства водителя машины определенного типа (?)

Параметры:

`typeID`: идентификатор типа машины

Результат: показатель мастерства

Функция: `getVehicleClass(typeID)`

Назначение: узнать класс, к которому принадлежат машины данного типа (bus,taxi,transport, etc.)

Параметры:

typeID: идентификатор типа машины

Результат: класс машины

Функция: `getEmissionClass(typeID)`

Назначение: узнать класс испускания (выброса) для машины данного типа

Параметры:

typeID: идентификатор типа машины

Результат: НВЕФА-класс испускания (выброса)

Функция: `getShapeClass(typeID)`

Назначение: узнать класс фигуры для машины данного типа

Параметры:

typeID: идентификатор типа машины

Результат: класс фигуры

Функция: `getGUIOffset(typeID)`

Назначение: узнать необходимый размер свободного места позади машины (empty space after leader)

Параметры:

typeID: идентификатор типа машины

Результат: свободное место

Функция: `getWidth(typeID)`

Назначение: узнать ширину машины данного типа

Параметры:

typeID: идентификатор типа машины

Результат: ширина

Функция: `getColor(typeID)`

Назначение: узнать цвет машины данного типа

Параметры:

typeID: идентификатор типа машины

Научное издание

Власов Алексей Александрович  
Орлов Никита Андреевич

УПРАВЛЕНИЕ НАСЫЩЕННЫМИ ТРАНСПОРТНЫМИ ПОТОКАМИ  
В ГОРОДАХ

Монография

В авторской редакции  
Верстка Н.А. Сазонова

---

Подписано в печать 16.09.14. Формат 60×84/16.

Бумага офисная «Снегурочка». Печать на ризографе.

Усл. печ.л. 10,92. Уч.-изд.л. 11,75. Тираж 500 экз. 1-й завод 100 экз.

Заказ № 302.

---

Издательство ПГУАС.  
440028, г.Пенза, ул. Германа Титова, 28.