

Казахский национальный исследовательский технический университет имени  
К.И. Сатпаева

УДК: 004.94

На правах рукописи

**АХМЕДИЯРОВА АЙНУР ТАНАТАРОВНА**

**Математическое моделирование транспортных систем мегаполиса с  
применением теории S-гиперсетей**

6D070300 – Информационные системы

Диссертация на соискание степени  
доктора философии (PhD)

Научные консультанты  
доктор физико-математических наук,  
профессор  
Калимолдаев М.Н.

Doctor of Science, профессор  
Keylan Alimhan

доктор физико-математических наук,  
профессор  
Попков В.К.

Республика Казахстан  
Алматы, 2018

## СОДЕРЖАНИЕ

<b>НОРМАТИВНЫЕ ССЫЛКИ</b>	4
<b>ОПРЕДЕЛЕНИЯ</b>	5
<b>ВВЕДЕНИЕ</b>	7
<b>1 СОСТОЯНИЕ И ПРОБЛЕМЫ ЗАДАЧ УПРАВЛЕНИЯ ТРАНСПОРТНЫМИ СИСТЕМАМИ МЕГАПОЛИСА</b>	12
1.1 Транспортная инфраструктура мегаполиса и уровни дорожно-транспортной сети(ДТС)	12
1.2 Моделирование транспортной сети в виде S-гиперсети	14
1.3 Постановка задачи исследования	21
Выводы по 1 разделу	24
<b>2 МЕТОДЫ ИССЛЕДОВАНИЯ И РЕШЕНИЯ ЗАДАЧ ОПТИМИЗАЦИИ ДВИЖЕНИЯ ГОРОДСКИМ ТРАНСПОРТОМ</b>	26
2.1 Исследование методов моделирования и управления движением транспорта через перекресток	26
2.2 Задачи управления светофорами на перекрестках любой конфигурации	38
2.3 Задачи анализа и оптимизации транспортных развязок и размещения городских остановок	49
Выводы по 2 разделу	60
<b>3 МЕТОДЫ И РЕШЕНИЯ ЗАДАЧ УПРАВЛЕНИЯ ПОТОКАМИ ТРАНСПОРТА В МЕГАПОЛИСЕ С ИСПОЛЬЗОВАНИЕМ ТЕОРИИ S-гиперсети</b>	61
3.1 Задача нахождения максимального потока и методы решения	61
3.1.1 Решение задачи нахождения максимального потока в транспортной сети с использованием алгоритма Форда-Фалкерсона	61
3.1.2 Определения интегрального максимального потока для участка региональной сети	65
3.2. Исследование влияния одностороннего движения на величину транспортного потока	68
3.3 Задача устранения пробок на маршруте при управлении транспортными потоками на сети городских дорог	75
3.4 Задача маршрутизации транспортных потоков города с применением теории S-гиперсети	84
3.4.1 Описание и анализ методов маршрутизации	86
3.4.2 Решение задачи маршрутизации транспорта с применением теории S-гиперсети	89
3.5 Задача о размещении минимального количества видеокамер на заданной транспортной сети	94
3.6. Практическая реализация результатов исследования и разработка программы	102
Выводы по 3 разделу	106
<b>ЗАКЛЮЧЕНИЕ</b>	108



## **НОРМАТИВНЫЕ ССЫЛКИ**

В настоящей диссертации использованы ссылки на следующие стандарты

ГОСТ 19.101-77 – Описание программы

СТ РК 34.014-2002 – Информационные технологии

ГОСТ 19.201-78 – Единая система программной документации

ГОСТ 28195-99 – Оценка качества программных средств

## ОПРЕДЕЛЕНИЯ

В настоящей диссертации использованы следующие термины с соответствующими определениями

**Абсолютно плоская гиперсеть** - гиперсеть  $H = (PS, WS)$ , если  $WS$  и  $PS$  плоские графы.

**Алгоритм Дейкстры** – это классический алгоритм для поиска кратчайших путей от одной вершины графа до всех остальных.

**Гиперсеть** - шестерка, состоящая из трех множеств и трех отображений  $S = (X, V, R; P, F, W)$ , если  $\forall v \in V |P(v)| = 2, \forall r \in R |W(r)| = 2, \forall r \in R$  множество  $F(r) \subseteq V$  составляет маршрут в графе  $PS = (X, V)$ .

**Дерево** – ациклический граф.

**Длина ребра** - число ветвей, инцидентных этому ребру (части ребра).

**Дорожная сеть города** – дороги, перекрестки (обычные), перекрестки со светофорами, кольца.

**Затор** – это состояние транспортного потока, когда длительность задержки транспортных средств на пересечении составляет более длительности одного цикла светофора.

**Квазимаршрут** в гиперсети  $S = (X, V, R)$  - конечная последовательность  $\mu$ , в которой пара соседних элементов  $(x_i, r_i)$  инцидентны, а  $(r_i, x_{i+1})$  слабо инцидентны.

**Квазипланарная S- гиперсеть** - нестационарная гиперсеть, который в любой стационарный промежуток времени является квазипланарной.

**Маршрут** в гиперсети  $S = (X, V, R)$  - конечная последовательность  $\mu = (x_1, r_1, x_2, \dots, x_{r-1}, r_{k-1}, x_k)$ , составленная из элементов  $X, R$  таким образом, что вершины и рёбра чередуются, а всякие два соседних элемента инцидентны.

**Мультиграф** – граф, в котором допускается более одного ребра между вершинами.

**Орграф** – граф, рёбрам которого присвоено направление.

**Псевдогиперсеть** - частично упорядоченная последовательность  $\eta = \{PS, WS_1, \dots, WS_k\}$ , если неопределены отображения  $\psi(WS, PS), \psi(WS_j, WS_i)$ , где  $j < i$ , для всех соседних пар вторичных сетей из  $\eta$ .

**Путь** - последовательность вершин  $v_1, \dots, v_n$  (\*) для которых существуют дуги  $v_i \rightarrow v_{i+1}$ .

**Ранг слоя** – число сетей на данном слое (значение соответствующего элемента в спектре).

**Регулярный город** - город, который обладает симметрией, по крайней мере приблизительно.

**Сеть Петри** – двудольный ориентированный мультиграф, состоящий из вершин двух типов — позиций и переходов, соединённых между собой дугами.

**Слабый маршрут** в гиперсети  $S = (X, V, R)$  - конечная последовательность  $\mu = (x_1, r_1, x_2, \dots, x_{r-1}, r_{k-1}, x_k)$ , составленная из элементов  $X, R$  таким образом, что вершины и рёбра чередуются, а всякие два соседних элемента слабо инцидентны.

**Транспортная развязка** - комплекс дорожных сооружений (мостов, туннелей, дорог), предназначенный для минимизации пересечений транспортных потоков и, как следствие, для увеличения пропускной способности дорог.

**Устойчивая S-гиперсеть** - S-гиперсеть, который при заданном воздействии на элементы, структурные и числовые характеристики данной S-гиперсети остаются в определенных интервалах.

## ВВЕДЕНИЕ

**Актуальность темы.** Концепция вхождения Казахстана в число 30 самых развитых стран мира предполагает динамичное развитие транспортной инфраструктуры. «Транспортная инфраструктура - это кровеносная система нашей индустриальной экономики и общества», - подчеркнул Президент Нурсултан Назарбаев, представляя Стратегию развития Казахстана до 2050 года.

Транспорт имеет жизненно важное значение для эффективного экономического развития и является ключом к обеспечению социального благосостояния населения, обеспечивая повседневную мобильность людей и необходимое распределение товаров. Адекватная инфраструктура является основным предварительным условием для эффективности транспортных систем. Урбанизация и автомобилизация – основные причины, по которым не могут быть удовлетворены требования городской мобильности.

Повышение уровня автомобилизации в мегаполисах редко сопровождалось соответствующим обновлением дорожной сети, и эти увеличения, вероятно, продолжатся и в XXI веке, что еще более усугубит проблему. В менее развитых странах владение автомобилем находится на гораздо более низком уровне. По уровню автомобилизации Казахстан занимает лидирующую позицию в Центральной Азии. По состоянию на 1 января 2018 года в Республике Казахстан их количество составило 3851,6 тыс. единиц (stat.gov.kz). Показатель по уровню насыщения владения автомобилями в нескольких городах США превышает 80 на 100, тогда как в городах Юго-Восточной Азии уровень редко превышает 10 на 100. В данное время на 100 человек населения Казахстана приходится 21,4 автомобиля [1].

Пробки на дорогах не дешевы. По оценкам INRIX, такие расходы американских водителей составляют около 305 млрд. долларов за 2017 год или 1445 долларов на одного водителя. Эти затраты поступают прямо и косвенно из источников, таких как потерянное время, топливо и производительность. Прямые потери каждого водителя в Алматы 178 тысяч тенге, а ежедневные потери экономики Алматы от пробок составляет более 600 млн тенге [1].

Автомобилизация также является причиной того, что почти половина всех жителей нашей планеты живут в районах, которые не соответствуют стандартам качества воздуха. Риски для здоровья, связанные с загрязнением воздуха, чрезвычайно серьезны. Плохое качество воздуха увеличивает респираторные заболевания, такие как астма и бронхит, повышает риск опасных для жизни состояний, таких как рак, и обременяет нашу систему здравоохранения значительными медицинскими расходами. Частичное вещество в составе автомобильных выбросов несет ответственность за приблизительно 30000 преждевременных смертей каждый год.

Светофор с фиксированным режимом переключения в настоящее время не обеспечивает регулирование движения пешеходов и транспортных средств, плавного движения автомобилей на транспортных маршрутах. Синхронизация нескольких систем светофора на соседних перекрестках является сложной проблемой, учитывая различные параметры. Обычные системы не

обрабатывают переменные потоки, приближающиеся к перекресткам. Кроме того, взаимные помехи между соседними системами светофора, несоответствие автомобилей с течением времени, несчастные случаи, прохождение аварийных транспортных средств и пешеходный переход не применяются в существующей системе движения. Это приводит к возникновению пробок и заторов.

Для решения проблем пробок и заторов мегаполиса мировая практика идет по пути разработки и внедрения информационных технологий и кибер-технических систем для планирования и организация движения улично-дорожной сети.

Современные программные комплексы PTV VisionVissim, Aimsun, AnyLogic, ISSOP и т.д. позволяют прогнозировать возникновение заторов и дают возможность управления светофорами и трафиком. В данное время предлагают системы на базе микроконтроллера PIC, которая оценивает плотность трафика с помощью датчиков и выполняет динамические временные интервалы с различными уровнями. Программа компании A+S выполняет весь комплекс работ по транспортному планированию, моделированию, проектированию объектов дорожной инфраструктуры и комплексной визуализации.

Программный комплекс «Автоматизированная система контроля транспортных потоков» отечественной компании ТОО «Галамдык технологиялар» формирует и управляет в автоматизированном режиме базой данных по учету транспортных средств, проезжающих через дорожные посты, а также выдает отчеты и анализ данных.

Многие журналы, как IEEEITSM, Transportation Research, Transportation Science и т.д., публикуют рецензируемые статьи, которые предоставляют инновационные идеи исследований и результаты их применения, повышают осведомленность о неотложных исследованиях и проблемах приложений во всех областях интеллектуальных транспортных систем[1-12].

Инженерный подход к анализу проблем моделирования и управления транспортными потоками и инфраструктурой основан на эмпирическом анализе, которые отражены в работах Д. Дрю [13], Т. Хамада[14], В. Сильянова, В.А.Черепанова[15], М.С.Фишельсона[16], Г.И.Клишковштейна[17], И.Н.Пугачева[18], В.Т.Капитанова[19], М.В.Яшиной, Ф. Хейта, В.В.Зырянова, М.П.Печерского, Л.С.Абрамова [20-21] и др.

Научно-исследовательский институт транспорта и коммуникаций (НИИ ТК) является крупнейшей компанией в Казахстане. НИИ ТК занимается проектированием объектов разных видов транспорта, коммуникаций и связи, исследованием развития транспортно-коммуникационного комплекса Республики Казахстан. В НИИ ТК особое внимание уделяется изучению теоретических и практических основ комбинированных и мультимодальных перевозок, транспортного законодательства, безопасности дорожного движения.

Отечественные исследователи В.В.Яворский, А.Н.Хамитов и другие рассматривали разработку моделей функционирования и развития маршрутизированных систем городского пассажирского транспорта, а также

интеллектуальную транспортную систему. Также Н.К.Исингарин, М.А.Кобдинов, С.М.Биттибаев, А.С.Сабетов, А.Д.Мустапаева, С.Ш.Сарбаев и другие вынесли большой вклад в развитие исследований эффективной и качественной работы транспорта, технологии процесса перевозок, исследования взаимодействия звеньев транспортного комплекса.

Малыхина Е.А. защитила кандидатскую диссертацию на тему «Разработка методов и моделей управления перемещениями населения в городской пассажирской системе». Результаты данного исследования могут быть внедрены в предприятия городского пассажирского транспорта для обоснования показателей работы автобусного транспорта на маршрутах.

Проведен анализ литературных источников по теории транспортных потоков следующих авторов, которые внесли значительный вклад в разработку теоретических аспектов развития транспортной инфраструктуры: Сильянов В.В., Дрю Д., Лобанов Е.М., Фишельсон М.С., Черепанов В.А., Капитанов В.Т., Хилажев Е.Б., Таташев А.Г., Новиков А.В., Трибунский В.М., Буслаев А.П., Приходько В.М., Яшина М.В. и т.д.

По результатам анализа выяснилось, что для повышения эффективности управления транспортной инфраструктурой мегаполиса при принятии управленческих решений надо уделить особое внимание системным связям. Определяющей характеристикой ее сложности является многоаспектность представления транспортной инфраструктуры и неопределенность их поведения. Решение данной проблемы современной интеллектуальной транспортной системой сдерживается из-за дефицита методов и инструментов приобретения, представления и обработки разнородных знаний для построения адекватных моделей и решения всех видов задач комплексного управления транспортной инфраструктурой.

Для осуществления всех фаз комплексной обработки информации может служить теория S-гиперсетей, позволяющая разработать современное методологическое и инструментальное обеспечение и технологию обработки информации на ее основе. Математические модели связности, модели и методы структурирования сложных систем, некоторые аспекты теории S-гиперсетей моделирование городских транспортных систем гиперсетями рассмотрены в работах В.К.Попкова [22-28], являвшимся одним из научных консультантов автора диссертации, что дает основание продолжить и развить это научное направление.

Таким образом, математическое моделирование транспортных систем мегаполиса с применением теории S-гиперсетей является актуальной научной задачей.

**Цель работы.** Целью работы является повышение эффективности управления транспортными потоками на основе исследования и разработки информационных технологий для комплексного математического моделирования транспортных систем города с применением теории S-гиперсетей.

Основной результат исследования заключается в возможности повышения оперативности управления транспортными потоками города, на основе

разработки комплексных решений с использованием кибер-физических систем мониторинга, моделирования, прогнозирования и оптимизации с применением теории S-гиперсети.

#### **Задачи исследования**

Для достижения цели в работе поставлены следующие задачи:

1. Анализ структурных особенностей транспортной сети мегаполиса с целью совершенствования моделей и алгоритмов для эффективного управления транспортным потоком в сети.

2. Разработать методику планирования транспортного потока и маршрутов, адаптированных на существующие инфраструктуры транспортной развязки и перекресток на основе теории S-гиперсетей.

3. Разработать предметно-ориентированную среду анализа, моделирования и поддержки принятия решений для эффективной организации дорожного движения.

4. Провести вычислительный эксперимент и имитационное моделирование на основе обработки разнородных информационных потоков и разработать методику эффективного управления транспортным потоком через транспортные сети мегаполиса.

**Объекты исследования.** Объектом исследования в диссертационной работе являются системы управления транспортной инфраструктурой города, включающей в себя улично-дорожную сеть, технические средства организации дорожного движения и транспортные потоки.

**Предмет и методы исследования.** В качестве методологической основы работы использовались методы управления транспортными потоками, теории графов, методы системного анализа и теории S-гиперсетей.

#### **Научная новизна**

1 Разработан унифицированный подход к моделированию и управлению движением транспорта на основе современной теории S-гиперсетей и усовершенствованного метода Попкова В.К., обеспечивающий комплексное решение вопросов структурно-функциональной организации транспортной инфраструктуры с учетом разнородности ее компонентов.

2 Разработаны алгоритм и программа, минимизирующие количество точек размещения видеокамер на заданной территории при условии полного её прослеживания.

3 Предложен подход с использованием разгонных светофоров, увеличивающий пропускную способность перекрестков, эффективность которого доказана имитационным моделированием, основанном на нечеткой логике.

#### **Положения, выносимые на защиту**

Предложенные математические и информационные модели и алгоритмы управления уличным движением через перекрестные участки транспортной сети на основе S-гиперсетей позволяют повысить транспортные потоки и эффективность загрузки транспортных сетей в крупных городах.

#### **Апробация результатов исследования**

Основные положения и результаты исследования докладывались на: Восьмой международной азиатской школы-семинара «Проблемы оптимизации сложных систем», (г. Омск, 2012); II Международной научно-практической конференции «Информационно-инновационные технологии, интеграция науки, образование и бизнеса», посвященной 20-летию Независимости Республики Казахстан (КазНТУ, 2012); Международной научно-практической конференции «Актуальные проблемы вычислительной и прикладной математики» (Новосибирск, 2014); V Международная научно-практическая конференция «Инновационная наука и современное общество» (г. Уфа, 2015), XXVII международная научно-практическая конференция «Естественные и математические науки в современном мире» (г. Новосибирск, 2015), II Международной научно-практической конференции «Информационные и телекоммуникационные технологии: образование, наука, практика» (Алматы, КазНТУ имени К.Сатпаева, 3-4 декабря, 2015), «International Conference on Information Science and Management Engineering» (Пхукет, Таиланд, 20-21 декабря, 2015), Международной научно-практической конференции «Казахстанский путь – 2050: единая цель, единые интересы, единое будущее» (г. Алматы, Каспийский университет, 17-18 марта 2016)

**Публикации.** По теме диссертации опубликовано 30 работ, из которых 8 опубликованы в изданиях, рекомендованных Комитетом по контролю в сфере образования и науки МОН РК, 1 статья опубликована в издании, входящем в базу данных Scopus, 1 статья опубликована журнале, входящем в базу Thomson Reuters, 15 статей опубликованы в сборниках международных научно-практических конференциях.

**Структура и объем диссертации.** Диссертация состоит из введения, трех разделов, заключения, списка использованных источников и приложений и список использованных источников из 76 наименований источников.

# 1 СОСТОЯНИЕ И ПРОБЛЕМЫ ЗАДАЧ УПРАВЛЕНИЯ ТРАНСПОРТНЫМИ СИСТЕМАМИ МЕГАПОЛИСА

## 1.1 Транспортная инфраструктура мегаполиса и уровни дорожно-транспортной сети(ДТС)

Транспортная инфраструктура мегаполиса как единая иерархическая нестационарная система состоит из элементов внешнего и внутри городского транспорта, взаимодействующих между собой. Особое внимание при проектировании транспортной системы следует уделять уровню автоматизации транспортных операций, который выбирают, исходя из экономических соображений. Необходимо учитывать, что при незначительных капитальных вложениях в транспортную систему высвобождается значительное число вспомогательных рабочих. Транспортная инфраструктура является интегральной взаимоувязанной системой сетей различного вида принадлежащих различным собственникам и функционирует в результате постоянного совершенствования управления со стороны муниципальных и государственных органов власти.

Основными элементами транспортной системы (ТС) мегаполиса являются перекрестки. Различают следующие элементы перекрестков [13, с.10]:

- с равнозначными дорогами и с разными дорогами;
- планарные, К – уровневые и распределенные;
- с пешеходными переходами (на земле, надземные, подземные) и без пешеходных переходов;
- регулируемые и нерегулируемые;
- размеченные и не размеченные;
- однородные по транспорту и неоднородные.

Транспортные единицы различаются на автомобильные и электрические. Автомобильными транспортными единицами являются: автобусы, микроавтобусы, индивидуальные автомобили, грузовой транспорт, служебный, скоростной (с сиреной и с мигалкой), специальный транспорт. Электрическими транспортными единицами являются: трамваи, троллейбусы, монорельсовый надземный, электропоезда на эстакаде, метрополитен, фуникулер.

Элементы транспортной инфраструктуры включают в себя: улично-дорожную сеть; внеуличную транспортную сеть (надземную, надземную и подземную); сети внешнего (междугородного) транспорта, проложенные через городские планировочные структуры; сооружения по обслуживанию транспортного хозяйства (парки и депо, грузовые терминалы или станции, энергетическое хозяйство, вокзалы).

Средствами управления движением на перекрестке являются: светофоры; информационные щиты и электронные табло; дорожные знаки и указатели; дорожная разметка; полосы разгона и торможения при перекрестке; правила дорожного движения; индивидуальные информационные и навигационные системы.

Для городской улицы дорога включает поперечное сечение между внешними границами тротуаров, а также проезды, рельсы (трамвая, метро,

электропоезда и др.), эстакады, мосты и т.д. Проезжая часть – элемент дороги или дорожного сооружения, предназначенный для движения безрельсовых транспортных средств.

Под полосой движения понимают продольный участок проезжей части, шириной достаточной для движения автомобилей в один ряд. Оконечными узлами полос являются: участки проезжих частей между перекрестками, разветвлениями и другими узлами дорожной сети: рельсовые пути, полосы движения на дорогах, полосы движения с контактными проводами для троллейбусов, монорельсы.

Существуют следующие модельные уровни ДТС [25, с.180]:

а) Локусы или локальные участки сети на полосах движения соответствующих местам занимаемым транспортными единицами в определенный интервал времени.

б) Маршруты движения транспорта - путь следования объекта, учитывающий направление движения относительно географических ориентиров или координат, с указанием начальной, конечной и промежуточных точек, в случае их наличия.

в) Графы аллелей. Графы аллелей представляют собой множество вершин соответствующих пересечению стоп-линий с полосами движения на перекрестках, разветвлениях и других узлах дорожной сети. Ребрам сопоставляются участки полос на проезжей части или аллели. Аллель является непрерывной полосой движения между узлами, а остановка необходима в силу правил дорожного движения.

г) Граф локусов. Граф локусов является самым большим по количеству вершин и ребер среди вторичных сетей S-гиперсети транспортной системы города. Для построения графа аллели разбиваются на участки, в каждом из которых может находиться не более одной транспортной единицы. Вершинами являются: возможные остановки транспорта или границы между локусами. Две вершины будут смежны, если транспортная единица перемещена за один такт в процессе движения.

Основными задачами моделирования внутри города транспортных потоков и их моделирования являются [25, с.182]:

- Математические модели потоков основных транспортных единиц внутри города;

- Задачи анализа транспортных потоков;

- Задачи распределения различных городских транспортных подсистем;

- Классификация, моделирование и оптимизация транспортных развязок.

Существуют следующие модели и задачи передвижения по городу пассажиров и товаров [25, с.182; 27, с.18]:

- Математические модели пассажиропотоков на городском транспорте;

- Задачи анализа и оптимизации маршрутов для пассажиров на городском личном транспорте;

- Математические модели грузопотоков и их оптимизация;

- Задачи организации транспортных потоков для обеспечения событийных перевозов.

Задачами управления транспортными системами являются [25, с.182; 27, с.18]:

- Задачи управления потоками транспортных средств посредством расстановки разметок и развязок;

- Задачи и методы управления потоками посредством светофоров и другими сигнальными средствами;

- Задачи оптимального управления городскими транспортными системами и алгоритмы маршрутизации транспортных единиц;

- Мониторинг движения транспорта и пешеходов.

Основными математическими моделями для задач размещения пунктов обслуживания транспортных средств и населения являются [25, с.183; 27, с.19]:

- Задачи размещения бензо-заправок и СТО;

- Оптимальное размещение гаражей и других мест хранения транспортных единиц;

- Задачи оптимизации размещения остановок, временных стоянок транспортных единиц;

- Размещение мест обслуживания населения с использованием городского транспорта и другие задачи.

## 1.2 Моделирование транспортной сети в виде S-гиперсети

Математическая модель должна объединять по возможности все факторы, влияющие на облик и функционирование транспортной системы. Сначала приведем определение простой гиперсети, в котором элементы в виде узла, однотипны, а линейные элементы (ветви, ребра) имеют различную природу.

Шестерка, состоящая из трех множеств и трех отображений  $S=(X,V,R; P,F,W)$  называется гиперсетью [27, с.17], если

$$\forall v \in V \mid P(v) \mid = 2,$$

$$\forall r \in R \mid W(r) \mid = 2,$$

$\forall r \in R$  множество  $F(r) \subseteq V$  составляет маршрут в графе  $PS=(X,V)$ .

Первичная PS и вторичная WS сети гиперсети S являются графами, а F отображает ребра  $WS=(X,R)$  в маршруты графа  $PS=(X,V)$ . Множество  $F(r)$  является маршрутом, поэтому отображение F единственным образом определяет отображение W. Концевые вершины маршрута  $F(r)$  являются одновременно концами ребра r, то есть гиперсеть S можно задать пятеркой  $(X,V,R; P,F)$ .

В гиперсети вида  $S=(Y,V,R)$  узел  $y \in Y$  заменяется на граф вида  $y = \{x_j^i, E_j\}$  – граф узла y структурированной гиперсети, где  $x_j^i$  – j-я вершина вторичной сети  $WS_i$ , отображенная в узел y структурированной гиперсети  $SA=(Y,V,G(X_i,R_i))$ . Таким образом, в отличие от гиперсетей, вершины вторичных сетей помещаются в узлы первичной сети независимо друг от друга, без ограничения на характер отображения, то есть возможны варианты отображения нескольких вершин одной вторичной сети в один узел первичной сети.

На рисунке 1.1 представлена S-гиперсеть с тремя видами транспортных систем (метро, автобус, трамвай) [27].

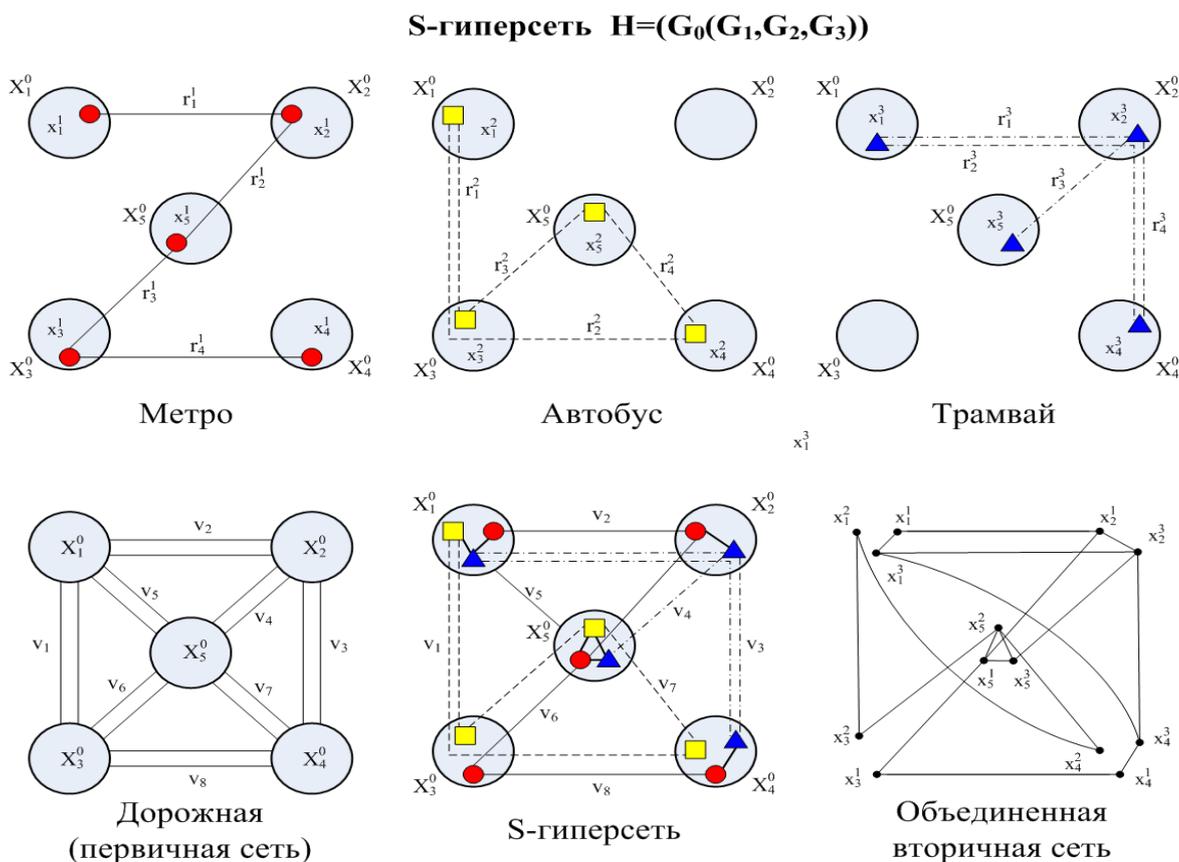


Рисунок 1.1- Пример объединения (суммирования) всех вторичных сетей гиперсети H

Дадим формальное определение S – гиперсети [27, с.18]. Пусть задано множество графов (гиперграфов)  $G_0=(X^0,V), G_1=(X^1,U^1), \dots, G_k=(X^k,U^k)$  и корневое дерево  $T_0=(Z,R)$ , где  $Z=z_0, z_1, \dots, z_k, R=r_1, \dots, r_k$  определяющее вложение графов  $G_j$  в  $G_i$  ( $i < j$ ) аналогично вложениям определяемым в гиперсетях за тем лишь исключением, что вершины  $x_k^i$  и  $x_l^j$  графов  $G_i$  и  $G_j$  не тождественны, а инцидентны. Очевидно, что одной и той же вершине  $x_k^i$  может быть инцидентны несколько вершин  $X_k^j = \{x_{k_1}^j, x_{k_2}^j, \dots, x_{k_l}^j\}$  из  $l$  графов  $\{G_{j_s}\}, s=1, \dots, l$ . На множестве вершин  $X_k^j$  определяется  $L^j = (X_k^j, E)$ . Вершины  $x_{k_j}^j$  и  $x_{k_s}^j$  смежны в  $L^j$ , если соответствующие графы  $G_{j_i}$  и  $G_{j_s}$  в вершине  $x_k^i$  имеют некоторую системообразующую связь  $l(x^{j_i}, x^{j_s})$ . В противном случае эти вершины не связаны. Также как в гиперсетях ребру  $u_i^j \in G_j$  в графе  $G_i$  сопоставляется цепь или некоторая связная часть между соответствующими вершинами из  $G_i$ . Здесь необходимо отметить, что системообразующие связи типа  $\{l(x,y)\}$ , вообще говоря, могут иметь разную природу и, как правило, существенно зависят от времени. В некоторых случаях, например, в системе транспортных

сетейного типа (метро, автобус, трамвай), такими связями в транспортных мульти узлах будут тротуарные линии(пешие переходы). В этом случае, имеет смысл рассматривать объединение всех вторичных сетей. Однако для некоторых задач имеет смысл рассматривать сумму всех графов гиперсети  $H$ , включая и первичную сеть  $PS$ , т.е.  $\underline{G} = G_0 + G_1 + \dots + G_n + \{L^j\}$ . На рисунке 1.2 приведен пример объединения всех графов, входящих в  $S$ -гиперсеть  $H$ .

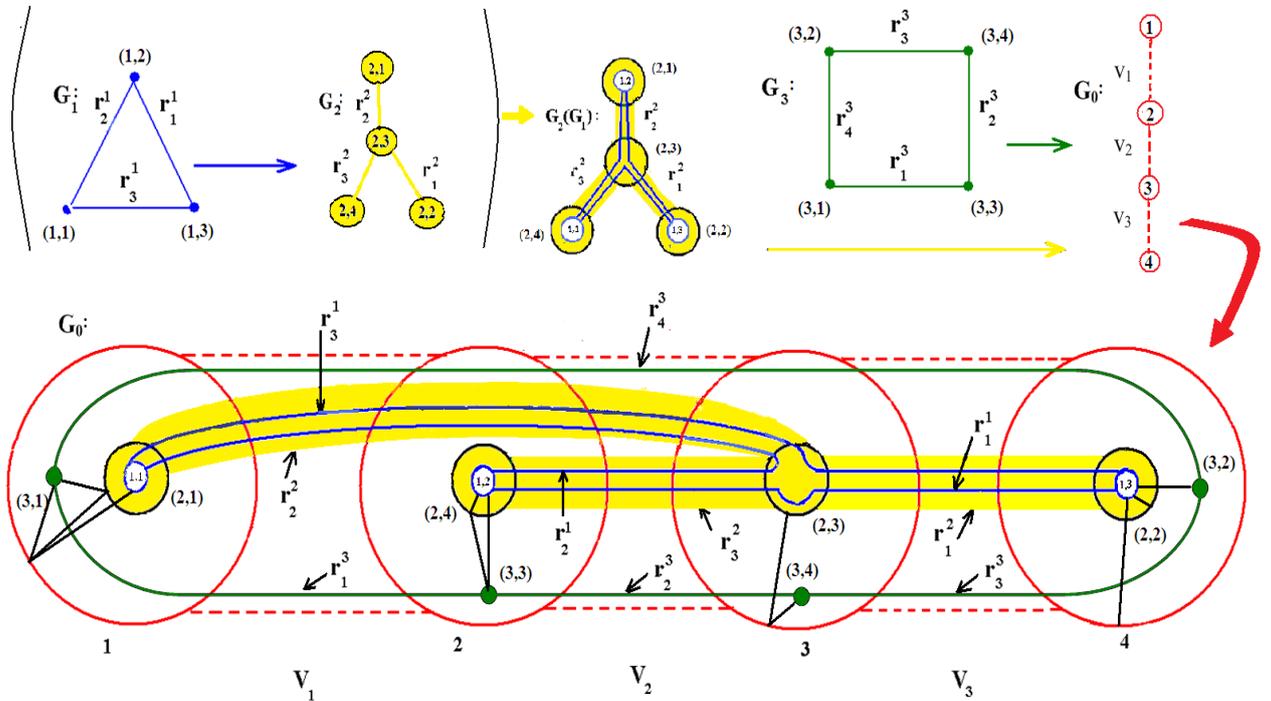


Рисунок 1.2 -Пример гиперсети [27]

Определенную таким образом  $S$ -гиперсеть с помощью нижеприведенных матриц можно задать с точностью до изоморфизма и даже с точностью до нумерации вершин и ребер [22, с.19].

Учитывая, что через один (пару) полюс могут проходить несколько ребер вторичных сетей, можно предложить данное свойство использовать для транспортного моделирования. Фрагмент  $S$ - гиперсети для транспортных моделей показано на рисунке 1.3.

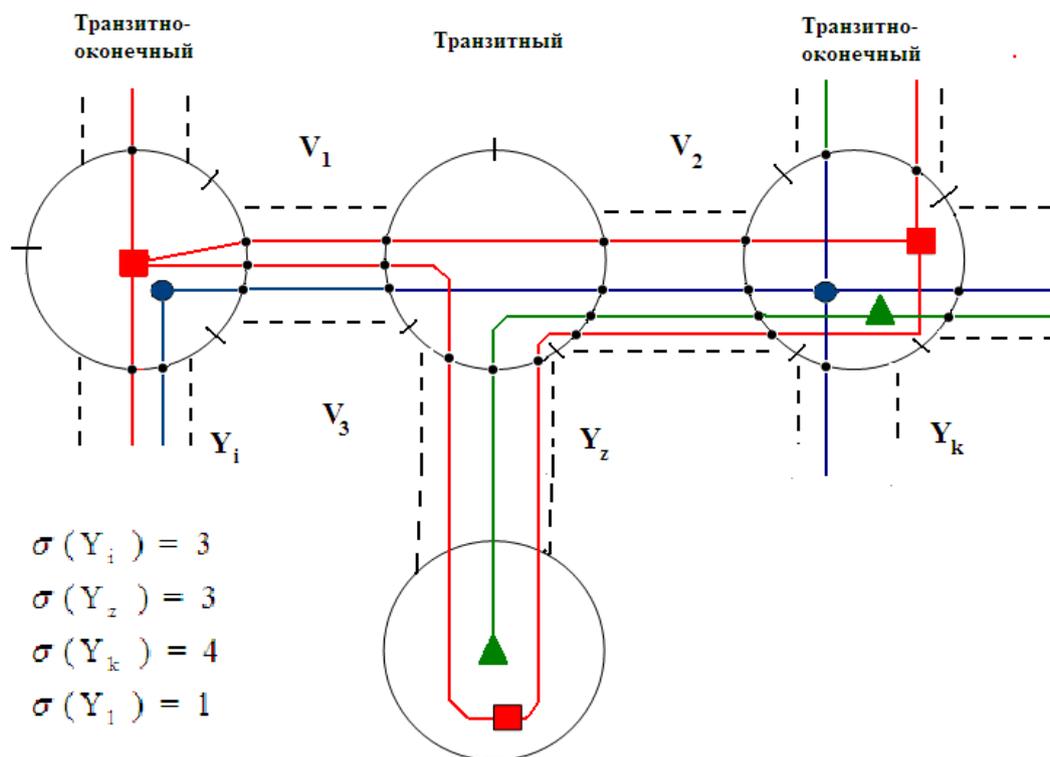


Рисунок 1.3 - Фрагмент  $S$ - гиперсети для транспортных моделей

Узлы  $Y_i$  и  $Y_k$  соответствуют транзитно-оконечным узлам, а  $Y_z$  целиком транзитный узел, а узел  $Y_1$  – оконечный  $\sigma(Y)$  - степень вершины  $Y$  в графе  $PS$ .

Отношения между вторичной сетью с первичной, а также между вторичными сетями, имеют характер типа симбиотических связей. Иначе говоря, они могут быть полезными, нейтральными или вредными. Исследование таких, взаимоотношений позволит анализировать системы различных взаимоувязанных сетей на предмет их устойчивости, развития и других характеристик. В таблице 1.1 приведена такая классификация симбиотических связей[27, с.23].

Таблица 1.1 - Классификация симбиотических связей

Типы симбиоза $S$ -гиперсетей $PS$ , $WS$		Первичная сеть $S$ -гиперсети, $PS=(Y,V)$		
		Полезная(П)	Нейтральная(Н)	Вредная(В)
Вторичная сеть $S$ -гиперсети, $WS=(X,R)$	П	Мутуализм Mutualism	Комменсализм Commensalism	Паразитизм Parasitism
	Н	Комменсализм Commensalism	Нейтрализм Neutralism	Аменсализм Amensalism
	В	Паразитизм Parasitism	Аменсализм Amensalism	Конкуренция Competition

Для лучшего понимания этих взаимоотношений и с учетом отображения элементов гиперсети на поверхности и разных взаимосвязей инцидентным ветвям ребер вторичных сетей, рассмотрим четыре таблицы с примерами

различных симбиотических связей и с учетом следующих отображений ребер в ветви первичной сети [27, с.24]:

- Ребра вторичной сети  $WS$  не отображаются в ребра первичной - экс-отображение  $WS \xrightarrow{экс} PS$
- Ребра вторичной сети  $WS = (X^1, R)$  идут рядом с ветвями первичной - пара-отображение  $WS \xrightarrow{p} PS$
- Ребра  $WS$  располагаются на «плоских» ветвях первичной сети - экто-отображение  $WS \xrightarrow{эк} PS$
- Ребра вторичных сетей располагаются внутри ветвей первичной сети - эндо-отображение  $WS \xrightarrow{эн} PS$ .

Приведённые примеры достаточно убедительно показывают важность данных связей для исследования различных и важных характеристик комплексов, состоящих из разнотипных систем сетевой структуры. Примеры экс-симбиоза, пара-симбиоза, экто-симбиоза и эндо-симбиоза в S-гиперсети  $H=(WS,PS)$  для различных CCC показаны в таблицах 1.2-1.5 [27, с.23].

Таблица 1.2 - Примеры экс-симбиоза  $WS$  и  $PS$  в S-гиперсети  $H=(WS,PS)$  для различных CCC

Экс-симбиоз		Первичная сеть S-гиперсети- $PS=(Y,V)$		
		П	Н	В
Вторичная сеть S-гиперсети - $WS=(X,R)$	П	Команды одного клуба(Мутуализм-эс/пп)	Элита и общество (Комменсализм - эс/пн)	Финансовая пирамида (Паразитизм - эс/пв)
	Н	Социальное обслуживание населения (Комменсализм - эс/пп)	Независимые клубы поинтересам(Нейтрализм - эс/нн)	Инфекционные больные и общество (Аменсализм - эс/пв)
	В	Система жертва - хищник (Паразитизм - эс/вп)	Существа в агрессивной среде (Аменсализм -эс/вн)	Соперники (Конкуренция - эс/вв)

Таблица 1.3 - Примеры пара-симбиоза  $WS$  и  $PS$  в S-гиперсети  $H=(WS,PS)$  для различных CCC

Пара-симбиоз		Первичная сеть S-гиперсети- $PS=(Y,V)$		
		П	Н	В
Вторичная сеть S-гиперсети - $WS=(X,R)$	П	Композиционные материалы(пс/пп)	Линии электропередач (ЛЭП) и шоссе(пс/пн)	Занятие телефонных пар в кабеле для передачи данных (пс/пв)
	Н	Система освящения дороги и шоссе (пс/пп)	Линии и инженерные сети в одном коллекторе(пс/нн)	ЛЭП и низкочастотные каналы в кабеле тональной частоты (пс/пв)
		Зимнее горячее и холодное водоснабжение в одном	Тротуары и проезжая часть (пс/вн)	Трамвайные пути и дорога с проезжей частью (пс/вв)

	В	коллекторе (пс/вп)		
--	---	--------------------	--	--

Таблица 1.4 - Примеры экто-симбиоза WS и PS в S-гиперсетях H=(WS,PS) для различных CCC

Экто-симбиоз		Первичная сеть S-гиперсети- PS=(Y,V)		
		П	Н	В
Вторичная сеть S-гиперсети - WS=(X,R)	П	Асфальтовое покрытие бетонных дорог(эк.с/пп)	Маршруты общественного транспорта на дорогах (эк.с/пн)	Потоки грузового транспорта на дорогах(эк.с/пв)
	Н	Специальный транспорт при ремонте дорог (эк.с/нп)	Полосы движения на проезжей части (эк.с/нн)	Гололеда на дорогах(эк.с/нв)
	В	Защитные покрытия дорог, труб и т.д. (эк.с/вп)	Потоки транспорта на разбитых дорогах (эк.с/вн)	Рельсовый транспорт на автодорогах(эк.с/вв)

Таблица 1.5 - Примеры эндо-симбиоза WS и PS в S-гиперсетях H=(WS,PS) для различных CCC

Эндо-симбиоз		Первичная сеть S-гиперсети-PS=(Y,V)		
		П	Н	В
Вторичная сеть S-гиперсети - WS=(X,R)	П	Сети электросвязи города(эн.с/пп)	Поток пассажиров в метро (эн.с/пн)	Потоки мин. воды в стальных трубах(эн.с/пв)
	Н	Поток агрессивной жидкости и в трубах (эн.с/нп)	Кабели связи в тоннелях метро(эн.с/нн)	Ливневый поток в кабельной канализации (эн.с/нв)
	В	Система питания в автотранспорте(эн.с/вп)	Обледенение ЛЭП (эн.с/вн)	Виртуальный компьютер(ПК в ПК)(эн.с/вв)

Приведенные примеры систем сетевой структуры показывают, что их взаимодействие может также описываться симбиотическими связями.

Системные топологии показаны в таблицах 1.6-1.8 [27, с.28].

Таблица 1.6 - Системная типология первичных и вторичных сетей

Классификационные признаки	Характеристики первичной и вторичных сетей					
	Тип математического объекта	Граф		Гиперграф		Гиперсеть
Ориентация	Ориентированный		Смешанный		Неориентированный	
Сложность	Тривиальный	Пустой	Дерево	Планарный	Полный	Произвольный
Однородность	Однородный			Неоднородный		
Весовые характеристики	Взвешенный			Невзвешенный		

Атрибуты	Детерминированная	Случайная
Стабильность	Стационарная	Нестационарная

Таблица 1.7 - Системная типология элементов

Классификационные признаки	Характеристики элементов		
Типы элементов	Узлы (вершины)	Ветви (ребра)	Полюса (полусвязь)
Весовые характеристики	Взвешенный(ая,ое)		Невзвешенный(ая,ое)
Ориентация	Ориентированный		Неориентированный
Время существования	Постоянно	По расписанию	Случайное
Представление	Абстрактное		Геометрическое
Число полюсов	Все полюса		Свободные полюса

Таблица 1.8 - Системная типология связей

Классификационные признаки	Характеристики связей			
Способ соединения	Слияние	Инцидентность	Слабая инцидентность	Смежность
Ориентация	Ориентированный		Неориентированный	
Время существования связи	Постоянно	По расписанию	Случайное	
Надежность соединения	С резервированием		Без резервирования	
Связь точечных элементов	Абстрактная (функциональная)		Физическая (геометрическая)	
Связь элементов	Экс-вложение	Пара-вложение	Экто-вложение	Эндо-вложение

Топологические типологии показаны в таблицах 1.9-1.12 [27, с.34].

Таблица 1.9 - Топологическая типология S-гиперсетей

Классификационные признаки	Топологические характеристики S-гиперсетей			
Размерность пространства	Одномерное	Двумерное		Трёхмерное
Род поверхности первичной сети	Плоская		K-города	
Число уровней в иерархии	Обычная гиперсеть (два слоя)		Куровней (слоев)	
Число сетей на K-ом уровне	0 уровень – первичная сеть		N(K) число сетей на уровне K	
Размерность	Конечная		Бесконечная	
Тип укладки элементов вторичных сетей в первичной	Не укладываются	Укладываются рядом с другими	Укладываются на элементе	Укладываются внутри элемента

Таблица 1.10 - Топологическая типология вторичных сетей

Классификационные признаки	Топологические характеристики вторичных сетей			
Тип математического объекта	Граф	Гиперграф	Ультраграф	Гиперсеть
Наличие циклов	Циклический		Ациклический	
Ориентация	Ориентированный		Неориентированный	
Род графа	Плоский		К-го рода	
Связанность графов вторичных сетей	Связанная (сеть связана с другими вторичными сетями)		Свободная (сеть связана только с первичной сетью)	

Таблица 1.11 - Топологическая типология элементов

Классификационные признаки	Топологические характеристики элементов			
Типы элементов	Узел (вершина)	Ветвь (ребро)	Полюс (полусвязь)	
Размерность	Без размера	Линейный	Плоский	Объемный
Число полюсов	Без полюсов		К-полюсный	
Представление	Абстрактное		Геометрическое	

Таблица 1.12 - Топологическая типология связей

Классификационные признаки	Топологические характеристики связей		
Ориентация полюсов	Неориентированные		Ориентированные
Число полюсов в элементах	Неоднородные		Однородные
Числотипов полюсов	Однотипные		Разнотипные
Занятость полюсов	Занятые		Свободные
Тип связи полюсов	Слияние	Инцидентность	Полуинцидентность

### 1.3 Постановка задачи исследования

Требуется осуществить прохождение потока  $M$  из пункта  $S$  в пункт  $T$  за время, не превышающее  $Time$ . Если этого сделать невозможно из-за того, что на дорогах пробки, то выявить эти места и передать на выход. Скорость всех машин усреднена.

Для решения задачи используются следующие данные:

1) Дорожная сеть города – дороги, перекрестки (обычные), перекрестки со светофорами, кольца.

2) Параметры дороги:

а)  $P_{max}(i, j) \in \mathbb{R}^+$  – максимальная пропускная способность дуги  $(i, j)$ .

б)  $P_{real}(i, j) \in \mathbb{R}^+$  – реальный усредненный за период времени  $Time$  поток.

в)  $zerotime(i, j) \in \mathbb{R}^+$  – время прохождения дуги, если на ней нет ни одной машины.

3)  $\delta \in (0,1)$  – светофорный параметр (для каждого перекрестка свой), обозначает, какую часть времени какой свет горит (соответственно другой свет горит  $1-\delta$  времени).

4) Параметры кольца:

а)  $\text{Circle\_number} \in \mathbb{Z}^+$  – число входящих (исходящих) из кольца дорог.

б)  $\text{Circlemax} \in \mathbb{R}^+$  – максимальная пропускная способность кольца.

в)  $\text{Circlereal}.1, \dots, \text{Circlereal.Circle\_number} \in \mathbb{R}^+$  – реальные, усреднённые за период времени  $\text{Time}$ , потоки между 1-ой исходящей дорогой и второй, между второй и третьей, ..., между  $\text{Circle\_number}$  и 1-ой.

г)  $\text{Circletime}.1, \dots, \text{Circletime.Circle\_number} \in \mathbb{R}^+$  – по аналогии с  $\text{zerotime}(i, j)$  и  $\text{Circlereal}$ .

5) Поточковые данные:

а)  $S$  – точка отправки потока.

б)  $T$  – точка принятия потока.

в)  $M \in \mathbb{Z}^+$  – величина потока (сколько машин подается в точку  $S$ ).

г)  $\text{Time} \in \mathbb{R}^+$  – время, которое не должен превысить поток из  $S$  в  $T$ .

В качестве модели используется ориентированный, взвешенный мультиграф. Каждая дорога – это дуга графа, каждый перекрёсток без светофора – вершина. Перекрёсток со светофором и кольцо – это более сложные конструкции [28, с.88; 30, с.35; 31, с.195], которые, тем не менее, приводятся к первым двум (дорога, перекрёсток без светофора) следующим образом:

Изначально перекрёсток без светофора – обычная вершина (за исключением того, что ей приписан параметр  $\delta$ ), которая преобразуется в полный граф  $K_4$ , как показано на рисунке 1.4.

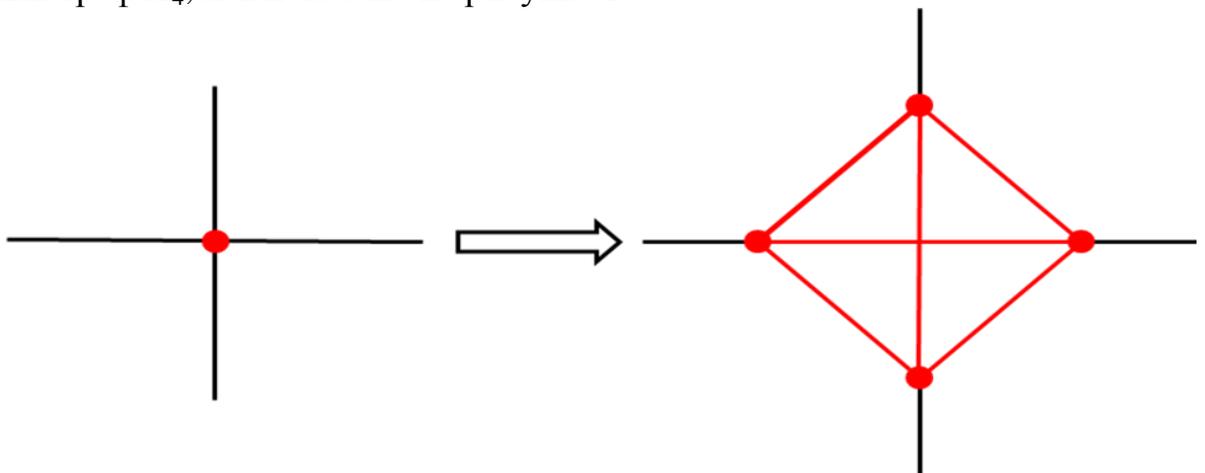


Рисунок 1.4- Преобразование перекрёстка со светофором

Пропускная способность  $P_{\max}(i, j)$  каждой из этих шести дуг равна минимуму пропускных способностей инцидентных основных дуг (не красного цвета). Так как светофор горит  $\delta$  времени в одну сторону и  $(1-\delta)$  в другую, то, соответственно, реальные пропускные способности будут  $\delta * P_{\max}$  и  $(1-\delta) * P_{\max}$ . Пояснение к преобразованию перекрёстка со светофором показано на рисунке 1.5.

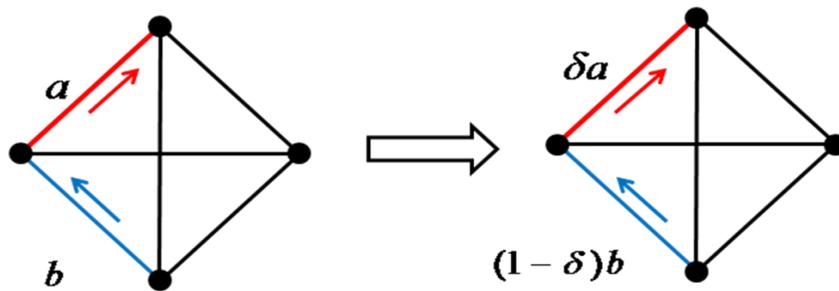


Рисунок 1.5 - Пояснение к преобразованию перекрёстка со светофором

Усредненный поток  $P_{real}(i, j)$  каждой дуги равен 0 (будем считать, что машины проходят перекресток мгновенно). Соответственно  $zerotime(i, j)$  из предыдущего предположения тоже равен 0.

Пусть дано кольцо. Тогда преобразуем его под нашу модель как на рисунке 1.6.

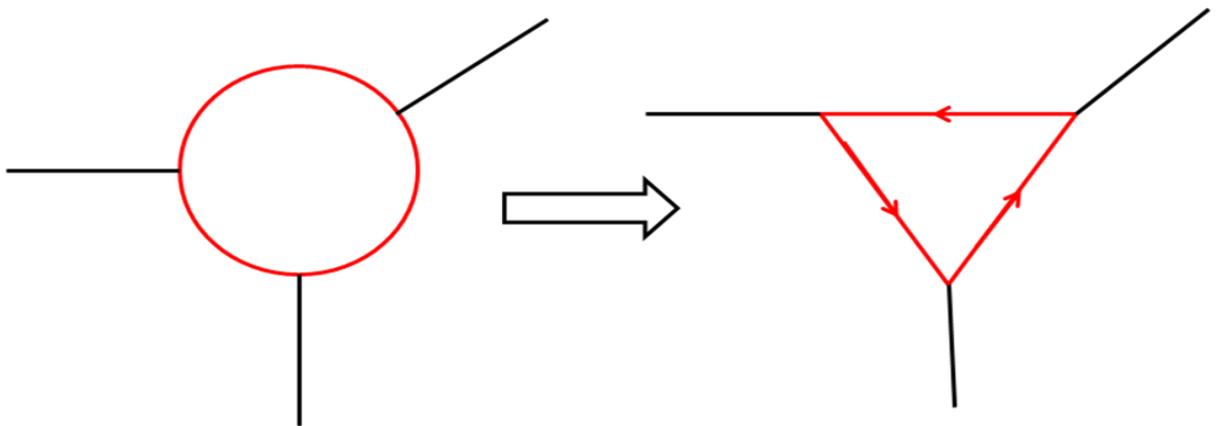


Рисунок 1.6 - Преобразование кольца

В результате получим структуру в рамках простых вершин и дуг. В качестве  $P_{max}(i, j)$  берем  $Circle_{max}$ . В качестве  $P_{real}(i, i+1)$  и  $zerotime(i, i+1)$  берем  $Circle_{real.i}$  и  $Circle_{time.i}$ , где рассматривается путь между  $i$ -ой и  $i+1$ -ой дорогами.

Для того чтобы анализировать движение, нам еще потребуется специальная функция, которая будет показывать время, потраченное для проезда единицы потока по данной дуге.

Вводится функция  $h(i, j) = h(P_{max}(i, j), P_{real}(i, j), zerotime(i, j)) \in \mathbb{R}^+$  – время прохождения единицы потока данного участка (дуги). Она получает на вход пропускную способность дуги, усредненный поток и время, которое бы потребовалось единице потока, чтобы проехать по этой дуге, будь она пуста (не было бы других машин). Эта функция носит более эмпирический характер и может задаваться многими способами. При реализации алгоритма используется следующая формула:

$$h(i, j) = \left(1 + \frac{P_{real}(i, j)}{P_{max}(i, j)}\right) * zerotime(i, j)$$

Для осуществления всех фаз комплексной обработки информации при решении поставленной и других задач моделирования и управления

транспортной инфраструктурой мегаполиса может служить теория S-гиперсетей, позволяющая разработать современное методологическое и инструментальное обеспечение и технологию обработки информации на ее основе. Математические модели связности, модели и методы структурирования сложных систем, некоторые аспекты теории S-гиперсетей и моделирование городских транспортных систем гиперсетями рассмотрены в работах В.К.Попкова [22-28], являвшимся одним из научных консультантов автора диссертации, что дает основание продолжить и развить это научное направление.

**Цель работы.** Целью работы является повышение эффективности управления транспортными потоками на основе исследования и разработки информационных технологий для комплексного математического моделирования транспортных систем города с применением теории S-гиперсетей.

Основной результат исследования заключается в возможности повышения оперативности управления транспортными потоками города, на основе разработки комплексных решений с использованием кибер-физических систем мониторинга, моделирования, прогнозирования и оптимизации с применением теории S-гиперсети.

#### **Задачи исследования**

Для достижения цели в работе поставлены следующие задачи:

1. Анализ структурных особенностей транспортной сети мегаполиса с целью совершенствования моделей и алгоритмов для эффективного управления транспортным потоком в сети.

2. Разработать методы и модели управления движениями транспортных средств на основе организации эффективного движения через перекресток мегаполиса и предложить методику планирования транспортного потока и маршрутов, адаптированных на существующие инфраструктуры транспортной развязки и перекресток на основе теории S-гиперсетей.

3. Разработать на основе полученных методов и моделей алгоритмы и предметно-ориентированную среду анализа, моделирования и поддержки принятия решений для эффективной организации дорожного движения.

4. Провести вычислительный эксперимент и имитационное моделирование на основе обработки разнородных информационных потоков и разработать методику эффективного управления транспортным потоком через транспортные сети мегаполиса.

#### **Выводы к разделу 1**

Проведен анализ инфраструктуры мегаполиса, ее математическое моделирование в виде S-гиперсети для основных видов транспорта.

В исследованиях транспортных потоков применяются идеи, методы и алгоритмы нелинейной динамики и возможностей управления. Их целесообразность обоснована наличием в транспортном потоке устойчивых и неустойчивых режимов движения, потерь устойчивости при изменении условий движения, нелинейных обратных связей, необходимости в большом числе

переменных для адекватного описания системы.

Рассмотрены актуальность моделирования транспортных потоков, суть основных существующих моделей и их возможности.

Выявлено и показано, что теория S-гиперсетей применима для анализа и синтеза многих систем сетевой структуры. В том числе для задач анализа межсетевых структурных взаимодействий сложных систем транспортной инфраструктуры.

Дана математическая формулировка маршрутов и метрик в S-гиперсетях, позволяющая вычислить отдаленность и расстояния с помощью известных методов на специально построенных графах, орграфах, гиперграфах и ультраграфах.

На основе проведенного анализа сформулирована цель исследования и задачи для реализации поставленной цели.

Основные результаты исследований по данному разделу содержится в следующих публикациях [31-35].

## 2 МЕТОДЫ ИССЛЕДОВАНИЯ И РЕШЕНИЯ ЗАДАЧ ОПТИМИЗАЦИИ ДВИЖЕНИЯ ГОРОДСКИМ ТРАНСПОРТОМ

### 2.1 Исследование методов моделирования и управления движением транспорта через перекресток

Управление транспортной инфраструктурой крупных городов с применением технологий интеллектуальных транспортных систем все активнее используется в мировой практике организации дорожного движения. Оптимальное регулирование движения транспортных потоков через перекресток является важной стратегической задачей в городском планировании. Естественным способом управления движением является регулирование работы светофора таким образом, чтобы образующиеся очереди автомобилей были как можно меньше, и чтобы стоящие в очереди автомобили как можно быстрее покидали перекресток.

Проанализированы основные характеристики транспортных потоков с точки зрения условия образования заторов. Приводится определение затора на регулируемых пересечениях: затор – это состояние транспортного потока, когда длительность задержки транспортных средств на пересечении составляет более длительности одного цикла светофора [22, 27].

Проанализировано решение проблем состояния транспортных потоков на регулируемых пересечениях УДС [15, 16, 5-9]. Существующая методика расчета задержек транспортных средств на регулируемых пересечениях, разработанная М.С. Фишельсоном [16], в современных условиях имеет недостатки, т.к. рассчитывает задержки только от однократной остановки транспортных средств и не учитывает существование заторов.

*Управление перекрестком в регулярных городах с помощью сетей Петри.*

«Регулярным» городом называется город, который обладает симметрией, по крайней мере приблизительно. Для таких симметрических городов строится сеть Петри, описывающая синхронизацию между всеми светофорами города, основанная на виртуальной циркуляции автомобилей на данной скорости и простого описания потоков этих автомобилей.

Для оптимизации проведения времени в системе автомобилем нужно разрабатывать "зеленые волны". Если правильно выбрать продолжительность цикла, то можно разработать четыре системы совместимых зеленых волн такие, что автомобиль может проехать между двумя точками города с предписанной скоростью, встретив максимум один красный свет. Этот результат является действительным только тогда, когда нет насыщения, когда потоки на всех улицах меньше, чем виртуальный поток машин. Предположение о геометрических закономерностях в городе не столь строго. Это может быть достигнуто зачастую путем адаптации скорости на каждой части улицы таким образом, что время, необходимое для прохождения каждого блока оставалось бы одинаковым.

*Макс-Плюс моделирование сетей Петри*

Поясним способ для расчета пропускной способности достаточно общего

класса сетей Петри, которые могут быть интерпретированы в терминах стохастического контроля.

Непрерывная сеть Петри определяется

$$N = (P, Q, M, \rho, m, \tau),$$

где:

$P$  – конечное множество элементов, называемых местами;

$Q$  представляет собой конечное множество элементов, которые называются переходами;

$M \in (\mathbb{R})^{P \times Q \cup Q \times P}$  являются множителями дуг, что  $M_{pq}$  (соответственно  $M_{qp}$ ) обозначает число дуг от перехода  $q$  к месту  $p$  (соответственно от места  $p$  к переходу  $q$ );

$\rho: Q \times P \rightarrow \mathbb{R}^+$  определенная уравнением:

$$\sum_{q \in p^{int}} \rho_{qp} = 1, \forall p \in P$$

является правилом маршрутизации, которое дает пропорцию жидкостей, идущих от места  $p$  до перехода  $q$ . по отношению к количеству жидкости, приходящей на место  $p$ ;

$m \in (\mathbb{R}^+)^P$  – первоначальные маркировки, а именно:  $m_p$  это количество жидкости на месте  $p$  в начальный момент времени;

$\tau \in (\mathbb{R}^+)^P$  является временем задержки, то есть временем, которое молекулы жидкости должны провести в месте перед отъездом.

Сеть Петри показано на рисунке 2.1.

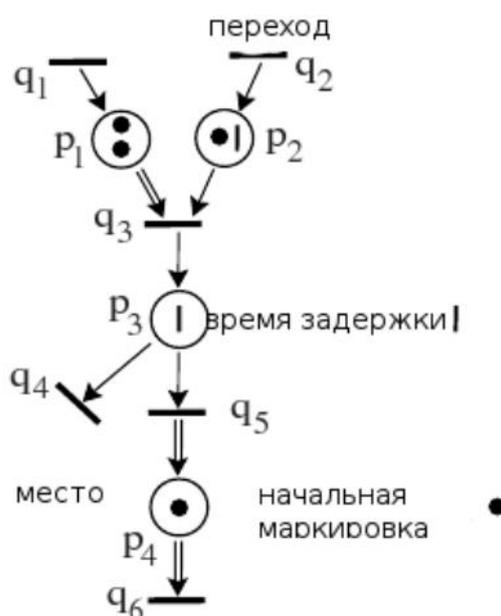


Рисунок 2.1 - Сеть Петри

Динамика системы определяется переходами. Переход происходит сразу, если есть достаточное количество текучей среды, доступной во всех

местах  $p \in q^{in}$  вверху перехода  $q$  (под "доступным" мы имеем в виду, что жидкость провела минимальное время стоя на месте). Общее количество проведенного  $q$  на месте рво время  $t$  есть  $Z_q(t)M_{qp}$  (где сумма срабатываний  $q$  обозначается  $Z_q$ ). Общий объем текущей среды, произведенной во время  $t \in q^{out}$ , является  $M_{pq}Z_q(t)$ . Процесс срабатываний идет как можно быстрее. Общий объем жидкости, который вступил на место рво время  $t$  обозначается  $Z_p(t)$ . Определяя:

$$\mu_{pq} = M_{pq}, \mu_{qp} = M_{qp}^{-1}, \tilde{\mu}_{qp} = \mu_{qp} \rho_{qp},$$

динамика системы полностью определится следующим образом:

Избавляясь от переменной  $Z_p$ , получаем уравнение динамического программирования, определяющее  $Z_q$ :

(2.1)

Это уравнение может быть истолковано как уравнение динамического программирования для задачи стохастического управления с дисконтируемой стоимостью.

При определенных условиях, описанных в следующей теореме эта задача (2.1) является недисконтируемой.

**Теорема 2.1.** Если существует  $v \in (\mathbb{R}^+)^Q$  что:

(2.2)

Уравнение (2.1) имеет интерпретацию недисконтируемого стохастического управления с функцией Беллмана:

$$W_q = \frac{z_q}{v_q}.$$

В частности, условие теоремы выполняется, когда:

,

то есть, когда для всех мест существует равное количество дуг, входящих и выходящих из места с общим правилом маршрутизации (то есть  $\rho_{qp} = 1/|p^{out}|$ , где  $|A|$  конечного множества  $A$  означает его мощность). В этом случае мы имеем  $v = \mathbf{1}$  ( $\mathbf{1}$  - вектор с координатами, равными 1).

Используя это замечание, мы можем определить пропускную способность этих сетей Петри, которые являются единственными живыми и стабильными сетями Петри (остальные либо взорвутся, либо умирают через конечное время).

**Теорема 2.2.** Обозначая

$$P_{qq'}^p = v_q^{-1} \widetilde{\mu}_{qp} \mu_{p q'} v_{q'}, v_q^p = m_p \widetilde{\mu}_{qp},$$

пропускная способность  $\lambda = \lim_{t \rightarrow \infty} Z_q(t)/t$  сильно связанной сети Петри, удовлетворяющей (2.2), существует, является независимой от  $q$  и является решением уравнения динамического программирования:

$$w_q = \min_{p \in q^{\text{in}}} (v^p - \lambda \tau_p + P^p w_q), \forall q \in Q.$$

На основе этой теоремы можно вычислить пропускную таких сетей Петри с помощью алгоритма Говарда, сложность которого экспериментально почти линейна по отношению к числу дуг в сети Петри.

*Моделирование синхронизации светофора с помощью сети Петри.* Сначала необходимо сделать предположение о том, что автомобили являются виртуальными, то есть реальные автомобили не обязаны двигаться, как виртуальные. Виртуальные автомобили полезны при исследовании идеального координирования. Потоки реальных машин не могут быть больше, чем потоки виртуальных.

*Моделирование перекрестка*

Сеть Петри, связанная с перекрестком, приведена на рисунке 2.2. Обозначим через  $x_0(t)$  и  $x_1(t)$  общее число зеленых фаз, которые произошли на каждом из двух светофорах до момента  $t$ .

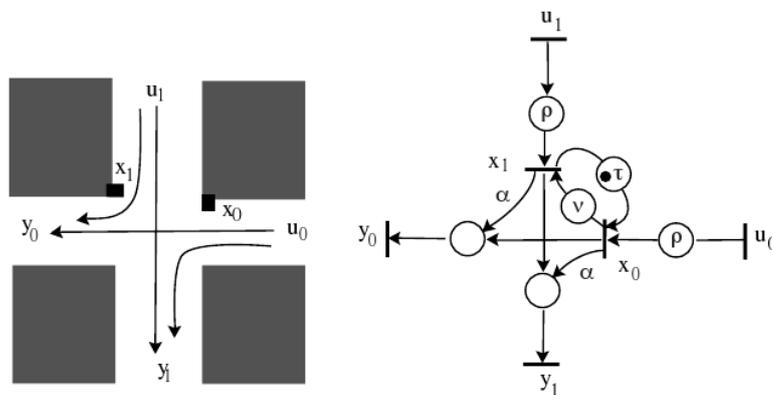


Рисунок 2.2 - Сеть Петри на перекрестке

Продолжительности зеленой фазы двух светофоров обозначаются соответственно  $\tau$  и  $v$ . Мы предполагаем, что количество автомобилей, которые могут пересекать перекресток, пропорционально длине соответствующей фазы зеленого с коэффициентом, который мы выбираем равным единице. Мы предполагаем, что на каждом перекрестке доля транспортных средств, равная  $\alpha$ , поворачивает только в одну сторону. Мы обозначим через  $u_0(t)$  и  $u_1(t)$  общее число легковых автомобилей, прибывших на перекресток до времени  $t$ , и  $y_0(t)$  и  $y_1(t)$  общее число автомобилей, которые покинули перекресток до

времени  $t$ .

Соотношение между входами  $u$  и выходами  $y$  является уравнением стохастического динамического программирования, где функция Беллмана есть  $x$ :

$$x = a \otimes x \oplus b \otimes u, \quad y = cx, \quad (2.3)$$

где

$$a = \begin{bmatrix} \varepsilon & \gamma\delta^\tau \\ \delta^\nu & \varepsilon \end{bmatrix}, \quad b = \begin{bmatrix} \delta^\rho/\nu & \varepsilon \\ \varepsilon & \delta^\rho/\tau \end{bmatrix}, \quad c = \begin{bmatrix} (1-\alpha)\nu & \alpha\tau \\ \alpha\tau & (1-\alpha)\tau \end{bmatrix},$$

где  $\oplus$  обозначает minplus сложение матриц,  $\otimes$  - minplus умножение матриц (замена сложения на минимум и умножения на сложение в обычном матричном умножении,  $\varepsilon = \infty, e = 0$ ),  $\delta$  - это единичный сдвиг во времени ( $\delta v(t) = v(t-1)$ ) и является единичным сдвигом в нумерации ( $\gamma v(t) = 1 + v(t)$ ).

С помощью этих обозначений в первом уравнении (2.3) получаем:  $x_0(t) = \min\{1 + x_1(t-\tau), u(t-\rho)/\nu\}$ ,  $y_0(t) = (1-\alpha)\nu x_0(t) + \alpha\tau x_1(t)$ .

Важно обратить внимание, что произведение матриц  $y = cx$  является стандартным. Очевидно эти уравнения динамического программирования (2.3), не линейны ни в minplus алгебре, ни в стандартной. Использование minplus матричного умножения является лишь удобным и компактным способом написания векторных уравнений.

#### *Моделирование блока перекрестков*

Рассмотрим регулярный город, такие, как, с одной изображен на рисунке 2.5., состоящий из разделенных улицами квадратов (с противоположным направлением движения для последовательных улиц).

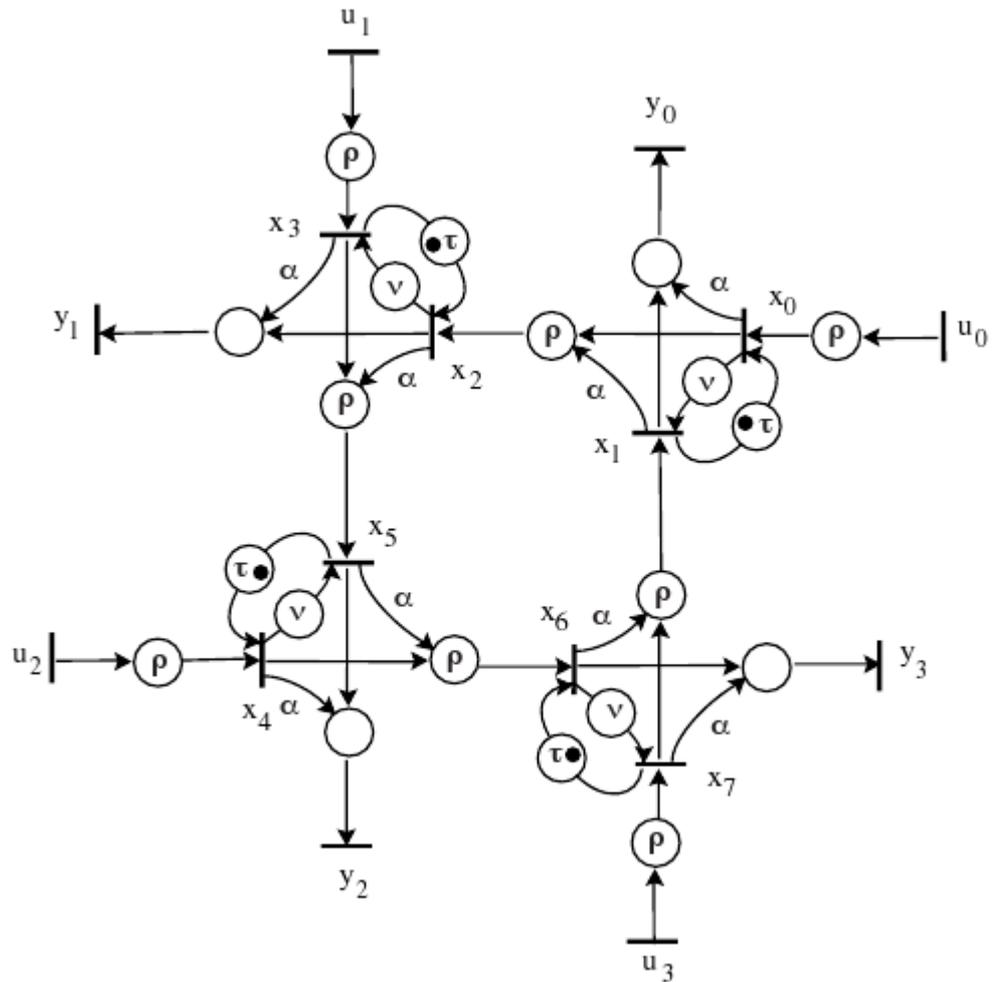


Рисунок 2.3 - Сеть Петри для блока из четырех перекрестков

Для определения динамики этой системы, было бы полезно сначала определить динамику блока, состоящего из четырех перекрестков.

Динамика блока определяется сетью Петри, приведенной на рисунке 2.3. Соответствующие уравнения:

$$\chi_i = a \otimes \chi_i \oplus b \otimes \pi^i \otimes c \chi_{i-1} \oplus b \otimes \pi^{i-1} \otimes E \otimes u_i, \quad y_i = E' \otimes \pi^i \otimes c \chi_i, \quad i = 0, 1, 2, 3,$$

где

$$\chi_i = \begin{bmatrix} x_{2i} \\ x_{2i+1} \end{bmatrix}, \quad \pi^1 = \pi^3 = \begin{bmatrix} e & \epsilon \\ \epsilon & \epsilon \end{bmatrix}, \quad \pi^0 = \pi^2 = \begin{bmatrix} \epsilon & \epsilon \\ \epsilon & e \end{bmatrix}, \quad E = \begin{bmatrix} e \\ e \end{bmatrix},$$

и расчет для индекса  $i$  был сделан по модулю 4.

Это система с 8 состояниями, 4 входами и 4 выходами системы, что можно формально записать, как:

$$x = A \otimes x \oplus B \otimes u, \quad y = Cx,$$

где  $A$  - это  $8 \times 8$  нелинейный оператор:

$$A = \begin{bmatrix} a & \epsilon & \epsilon & b\pi^0 c \\ b\pi^1 c & a & \epsilon & \epsilon \\ \epsilon & b\pi^0 c & a & \epsilon \\ \epsilon & \epsilon & b\pi^1 c & a \end{bmatrix}.$$

Наличие стационарного режима сразу гарантирует существование недисконтируемой интерпретации стохастического контроля этого уравнения динамического программирования. Для этого достаточно, чтобы

$$(1 - \alpha)v + \alpha\tau = v, (1 - \alpha)\tau + \alpha v = \tau,$$

где положили  $v = \tau$ .

В самом деле,  $\alpha, v, \tau$  могут зависеть от перекрестка, и в данном случае достаточным условием становится:

$$\begin{aligned} \alpha_{q-1}\tau_{q-1} + (1 - \alpha_{q-2})v_{q-2} &= v_q, q \text{ чётно,} \\ \alpha_{q-3}v_{q-3} + (1 - \alpha_{q-2})\tau_{q-2} &= \tau_q, q \text{ нечётно.} \end{aligned}$$

#### *Моделирование регулярного города*

Регулярный город (рисунок 2.4) состоит из блоков, которые мы можем пронумеровать парой  $(I, J)$ , где  $I$  это координата “запад-восток” (W-E) блока и  $J$  координата “юг-север” (S-N).

Тогда в динамике полного города можно написать:

$$x_{IJ} = Ax_{IJ} \oplus \mathcal{A}_0 x_{I+1, J} \oplus \mathcal{A}_1 x_{I, J+1} \oplus \mathcal{A}_2 x_{I-1, J} \oplus \mathcal{A}_3 x_{I, J-1},$$

где

$$\begin{aligned} \mathcal{A}_0 &= \begin{bmatrix} \epsilon & b\pi^1 c & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \mathcal{A}_1 = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & b\pi^0 c & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}, \\ \mathcal{A}_2 &= \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & b\pi^1 c \\ \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}, \quad \mathcal{A}_3 = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ b\pi^0 c & \epsilon & \epsilon & \epsilon \end{bmatrix}. \end{aligned}$$

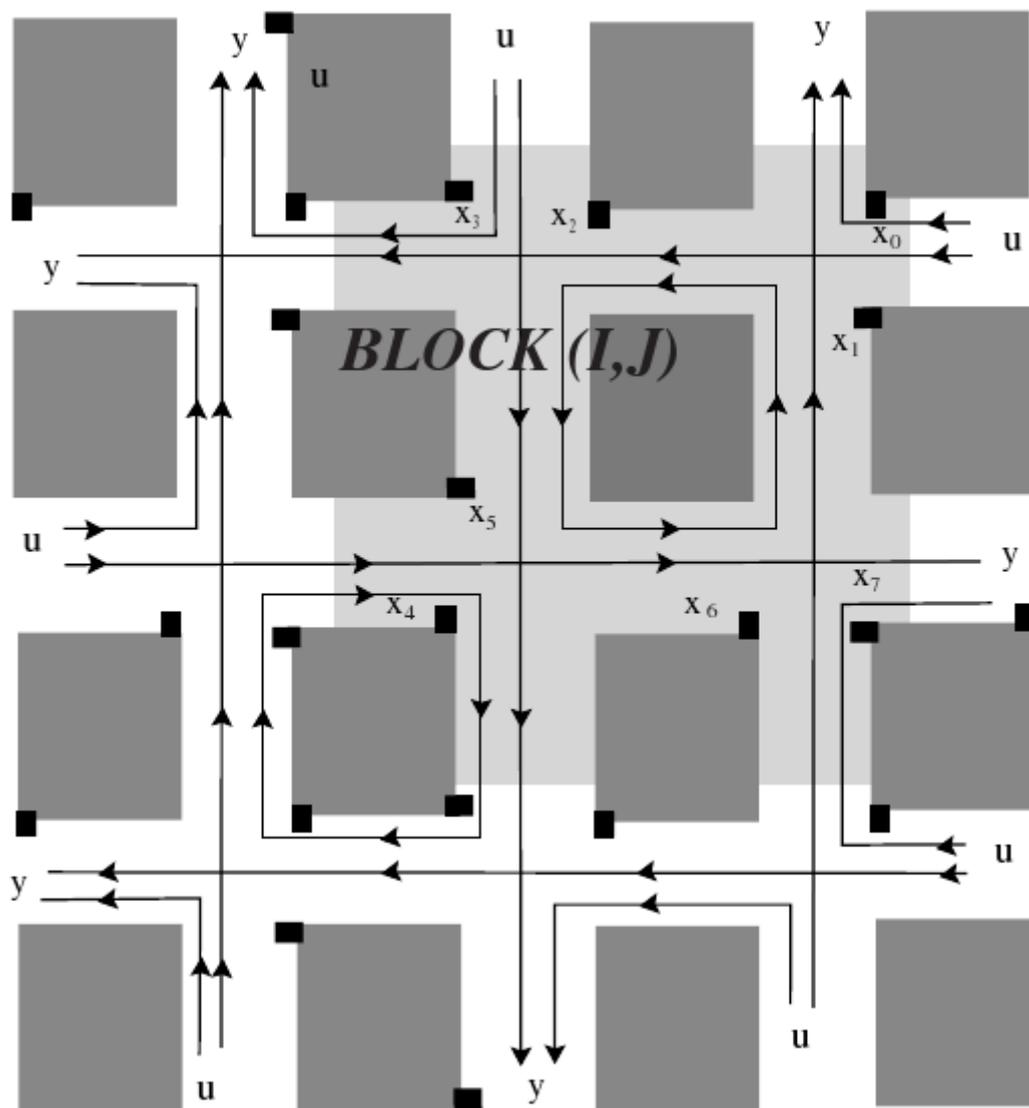


Рисунок 2.4 - Регулярный город

Как было показано выше, предыдущий тип системы мог быть решен эффективно. Но для этой системы придется оптимизировать некоторые параметры. Например, мы должны провести первоначальное обозначение в местах, соответствующих улице, время горения зеленого и красного света. Когда нет насыщения, можно подобрать довольно хорошие параметры и получить систему зеленых волн, которая позволяет проезд между двумя точками в городе с максимум одним красным светом.

Минусы этой модели очевидны:

Она адекватна только на симметричных городах, при небольших плотностях потока. Кроме того, требуется вручную регулировать множество параметров. Что делает ее пока неподходящей к регулированию перекрестка в общем случае.

#### *Использование нечеткой логики в управлении светофором*

В системах, основанных на нечеткой логике, есть контроллер, который управляет временем горения светофора с тем, чтобы обеспечить движение

трафика с минимальной задержкой. В этих системах также используются датчики, размещенные на перекрестках, сигнализируют контроллеру о прибытии транспорта, а также о покидании транспортом перекрестка.

Работа светофоров основана на нескольких принципах. Этап, когда трафик по некоторому набору полос покидает перекресток. На рисунке 2.5 показаны четыре типичные фазы для перекрестка. Завершение всех фазы называется циклом. Длина цикл определяется длиной фаз цикла. Обычно длина цикла predetermined заранее и является фиксированной величиной. Однако, в динамических методах длина цикла может изменяться в зависимости от состояния движения. Цвет светофора показывает начало или конец фазы. Зеленый свет обозначает активную фазу, желтый свет указывает окончание фазы и красный цвет означает завершения текущей фазы и начало следующей фазы. Обычно фазы меняются одна за другой на каждом цикле.

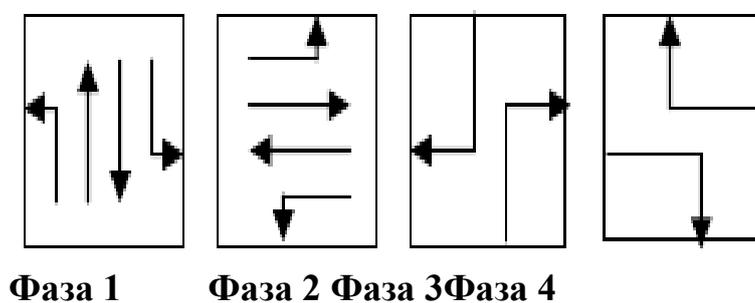


Рисунок 2.5 - Фазы изолированного перекрестка с четырьмя дорогами

Диспетчер Нечеткой Логике смотрит на состояния при других фазах светофора. Нечеткая логика разрешает использование подобны человеческому подход, чтобы принять решения, которые невозможно принять системе, основанной на количественных данных. Например, "ЕСЛИ МНОГО автомобилей прибыло, И МАЛО ЗЕЛЕНОГО СВЕТА, то ПРОДЛИТЬ ЗЕЛЕНЫЙ" - лингвистический подход, используемый в нечеткой логике.

*Модель нечеткой логики для изолированного перекрестка*

Формулировка нечетких входных параметров:

Рассмотрим изолированный перекресток с четырьмя подъездными путями, как показано на рисунке 2.6. Каждый подход имеет 3 полосы входящих и 3 исходящих. В этой модели четыре фазы.

1. Автомобили движутся с севера на юг (прямо), севера на восток (поворот), с юга на север (прямо) и с юга на запад (поворот)
2. С запада на восток (прямо), с запада на север (поворот), с востока на запад (прямо) и с востока на юг (поворот)
3. С севера на запад (поворот) и с юга на восток (поворот).
4. С запада на юг (поворот) и с востока на север (поворот).

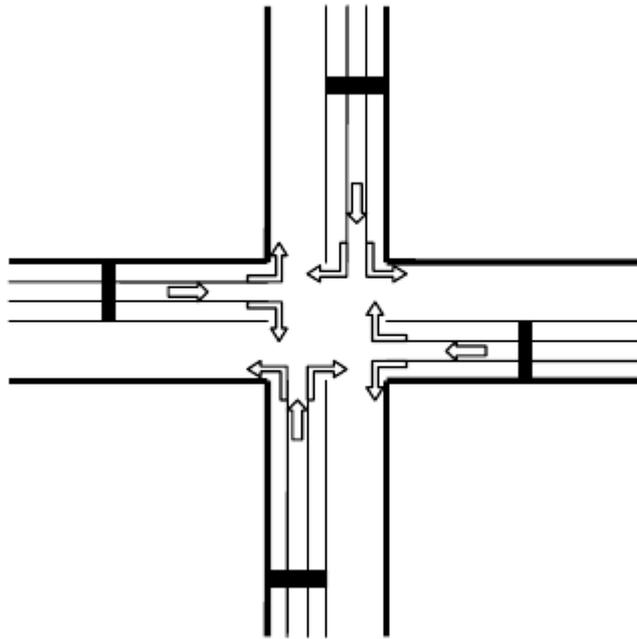


Рисунок 2.6 - Изолированный перекресток

В качестве входного вектора к диспетчеру нечеткой логики используется четыре параметра. Первый – длина очереди на красной фазе (RQL), которая является остатком транспортных средств начиная с последнего зеленого сигнала плюс прибытие в течение текущего красного сигнала:

$$RQL = Q_{gr} + \sum Vr(i), \quad (2.4)$$

где  $Q_{gr}$  - число транспортных средств, которые не покинули перекресток за фазу зеленого сигнала, и  $\sum Vr(i)$  - сумма прибывших транспортных средств в  $i$ -ю секунду красной фазы.

Второй параметр (RGT) - отношение оставшегося времени горения зеленого сигнала к общей продолжительности горения зеленого сигнала.

$$RGT = g(t)/T_g, \quad (2.5)$$

где  $T_g$  - полное время для зеленой фазы и  $g(t)$  - оставшееся время в момент  $t$  от начала зеленой фазы.

Третий параметр - прибытие транспортных средств в течение горения зеленого сигнала (AG), где  $\sum Vg(i)$  - число прибывших транспортных средств в  $i$ -ю зеленую фазу.

$$AG = \sum Vg(i). \quad (2.6)$$

Наконец, четвертый критический параметр для этой модели – среднее время разгрузки зеленого света (AGDT). Он указывает, покидают ли транспортные средства в пределах нормального потока или нет.

$$AGDT(n) = \alpha(CDT) + (1 - \alpha)AGDT(n - 1), \quad (2.7)$$

где CDT - текущее разгрузочное время покидания перекрестка автомобилями, AGDT(n-1) – предыдущее высчитанное время.

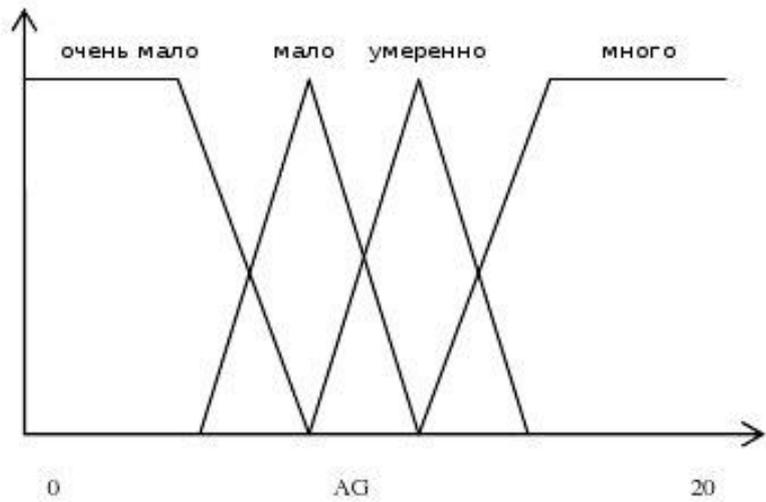
Задержка = момент покидания перекрестка транспортным средством – момент прибытия на перекресток (2.8)

*Нечеткие правила нормального диспетчера.* Диспетчер берет три входных параметра. Первые два - AG и RGT, которые берутся из текущей зеленой фазы. Третий параметр - RQL, который берется с дорожных полос на красной фазе. Он сравнивает эти параметры, чтобы решить, расширить ли или уменьшить время горения зеленого сигнала текущей фазы. Выходными данными является степень изменения во времени горения зеленого сигнала. Нечеткие наборы показываються в рисунке 2.7 (а, b, с).

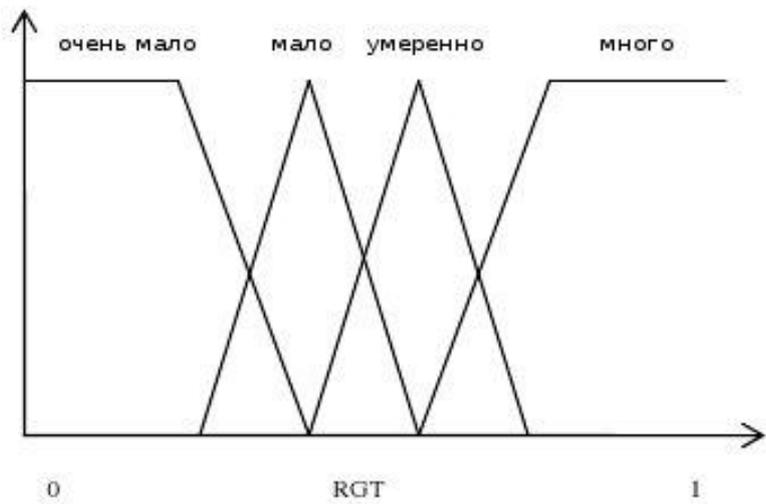
Если очередь на красной фазе является длинной и прибытие автомобилей на зеленой фазе невелико, то логически текущую зеленую фазу нужно закончить таким образом, чтобы диспетчер может перейти на следующую фазу. Например, нечеткое правило может быть заявлено подобно этому:

если **RQL** много и **AG** очень мало и **RGT** много, то *Расчет времени Уменьшен*.

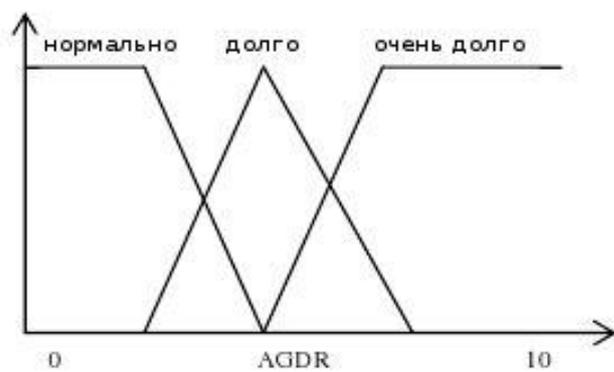
Рисунок 2.8 показывает нечеткие наборы выбора длительности сигнала.



(b) AG



(c) RGT



(d) AGDT

Рисунок 2.7 - Нечеткие наборы

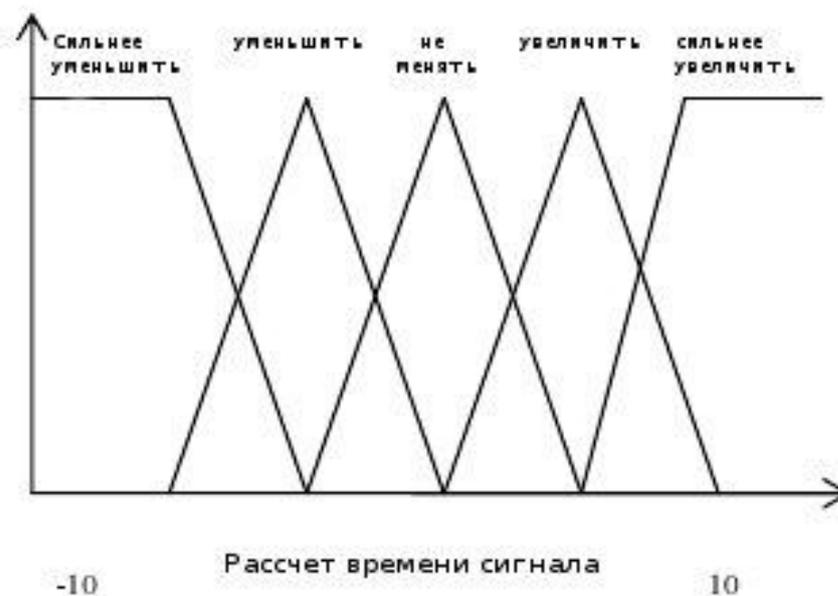


Рисунок 2.8 - Нечеткий расчет времени сигнала

*Нечеткие правила аномального диспетчера.* Этот диспетчер подобен нормальному, но он имеет дополнительный параметр AGDT. AGDT указывает, сколько времени в среднем автомобилю нужно, чтобы покинуть перекресток за зеленую фазу. Если AGDT возрастает, то есть возможность, что время горения зеленого сигнала будет увеличено. Например, нечеткое правило для этого модуля может быть заявлено так: если **RQL** мало и **AG** умеренно и **RGT** много и **AGDT** очень долгое, то расчет времени уменьшен.

Рисунок 2.7(d) иллюстрирует нечеткие множества для AGDT.

Эта система может быть расширена на группу перекрестков. Для этого диспетчеры дополнительно проводят синхронизацию со своими соседями.

Результаты сравнения автоматических контроллеров, основанных на нечеткой логике, показали выигрыш во времени прохождения автомобилями перекрестка по сравнению с обычными методами, в которых время горения сигналов светофора было фиксировано. Однако для реализации такой модели в городе необходимо понести солидные денежные убытки на установку контроллеров, датчиков и их соединение в сеть. Однако, это не является преградой для компьютерного моделирования.

## 2.2 Задача управления светофорами на перекрестках любой конфигурации

Способ относится к управлению движением транспортных потоков и может быть использован в различных транспортных системах. Технический результат заключается в повышении пропускной способности в местах пересечения потоков (в том числе транспорт - пешеходы), оборудованных светофорами. Способ заключается в том, что перед светофорами, осуществляющими регулирование транспортных потоков, устанавливают так называемые разгонные светофоры - РС. С помощью РС транспортный поток

разгоняется и стоп – линию светофора, разрешающего выезд на перекрёсток, проходит уже со значительной скоростью. Это позволяет увеличить объём транспортного потока, проходящего через светофорные объекты.

Для достижения указанного технического результата добавим к управляющему светофору(УС)дополнительно разгонный светофор(РС), который установим на расстоянии  $L_p$  передним. На разрешающий сигнал РС транспортное средство начинает ускоряться и когда оно преодолевает расстояние  $L_p$ , через время  $T_p$  подаётся разрешающий сигнал на УС. Транспортное средство со скоростью пересекает стоп – линию УС и совершает дальнейшее движение. В свою очередь запрещающий сигнал на УС подаётся через  $T_z$  после его поступления на РС. При этом  $T_z < T_p$ . Поэтому при одинаковой длительности переходных сигналов на РС и УС время разрешающего сигнала на РС больше, чем на УС. Для того, чтобы при необходимости транспортное средство могло предотвратить столкновение с транспортным средством своевременно не освободившем полосу движения или не совершить наезд на пешехода оно должно иметь возможность затормозить. В силу этого УС должен быть расположен на расстоянии  $L_t$  от перекрёстка или пешеходного перехода для светофорного пешеходного перехода вне перекрёстка. Величины  $L_p$ ,  $L_t$ ,  $T_z$  и  $T_p$  определяются для каждого транспортного потока. На их величину влияет состав транспортного потока или невозможность сделать величины  $L_p$ ,  $L_t$  более какой-то величины в силу малого расстояния между соседними светофорными объектами или иные причины.

Полученные путем моделирования значения: $L_p = L_t = 8\text{м}$ , $T_p = 3,5\text{ сек.}$ ,  $T_z = 0,6\text{ сек.}$  Откуда видно, что зелёный свет на РС горит больше, чем на УС на  $T_p - T_z = 3,5\text{ сек.} - 0,6\text{ сек.} = 2,9\text{ сек.}$

$L_p = L_t$  из соображения того, что транспортное средство должно иметь возможность затормозить перед перекрёстком, если при пересечении стоп – линии УС водитель транспортного средства видит, что перекрёсток занят, а участок разгона обычно длиннее тормозного пути.

$T_z$  чуть больше интервала времени, за которое транспортное средство преодолеет расстояние от стоп – линии РС до стоп – линии УС.

Это сделано для того чтобы транспортное средство, проскочившее РС пересекало стоп – линии УС«в более ранней стадии перед загоранием красного» и успевало пересечь перекрёсток.

Объектная модель разгонного светофора показано на рисунке 2.9.

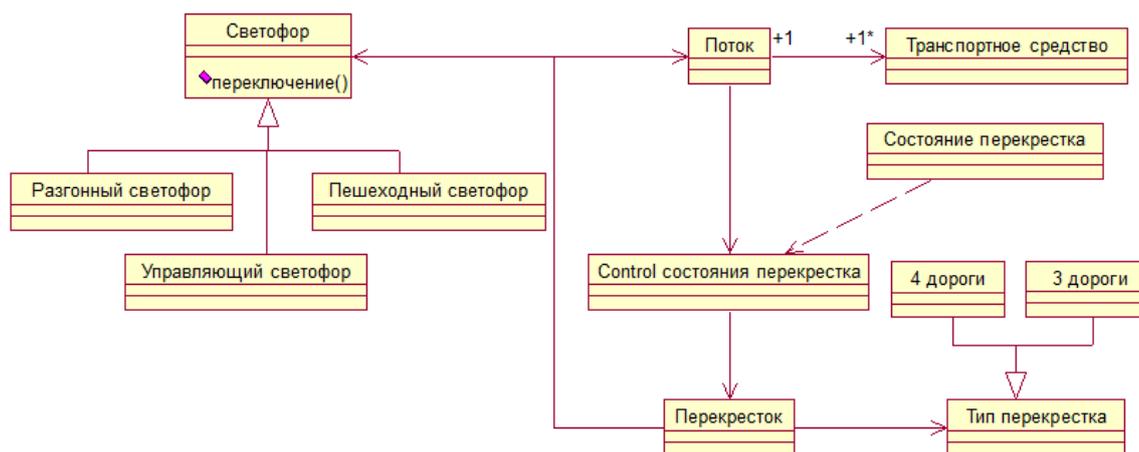


Рисунок 2.9 -Объектная модель разгонного светофора

На рисунке 2.10 представлена структура светофорного цикла с двумя фазами с одним промежуточным тактом в каждой фазе, поясняющая принцип работы предлагаемого способа управления пар светофоров РС и УС для равнозначного перекрёстка.

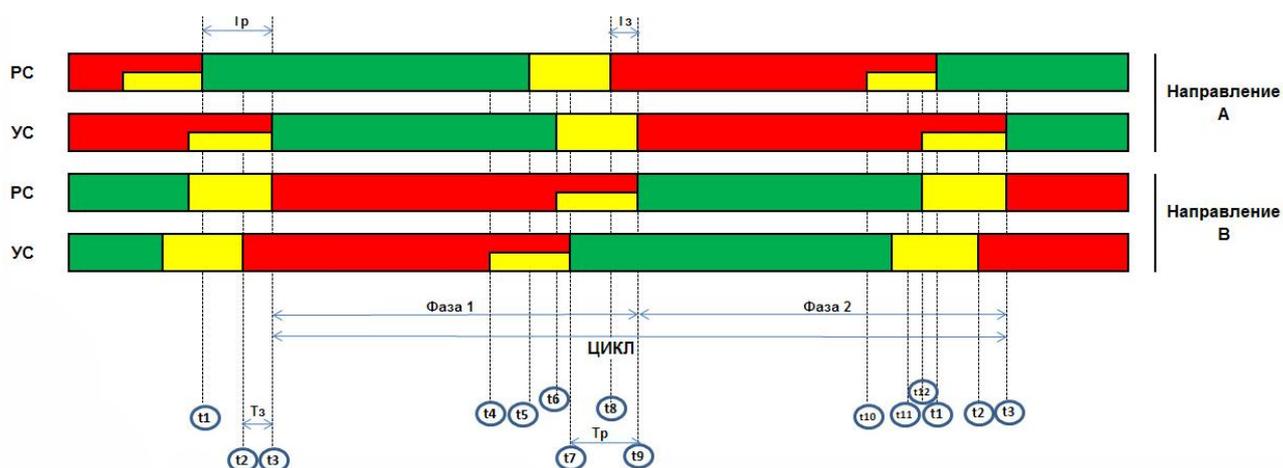


Рисунок 2.10 - Структура светофорного цикла

На рисунке 2.10 приняты следующие обозначения:

$t_1, t_2, t_3, \dots, t_{12}$  моменты времени, между которыми сигналы РС и УС постоянны,  $T_p$  – время разгона транспортного средства перед пересечением им стоп – линии управляющего светофора,  $T_3$  - время задержки поступления запрещающего сигнала на управляющий светофор после его поступления на разгонный светофор.

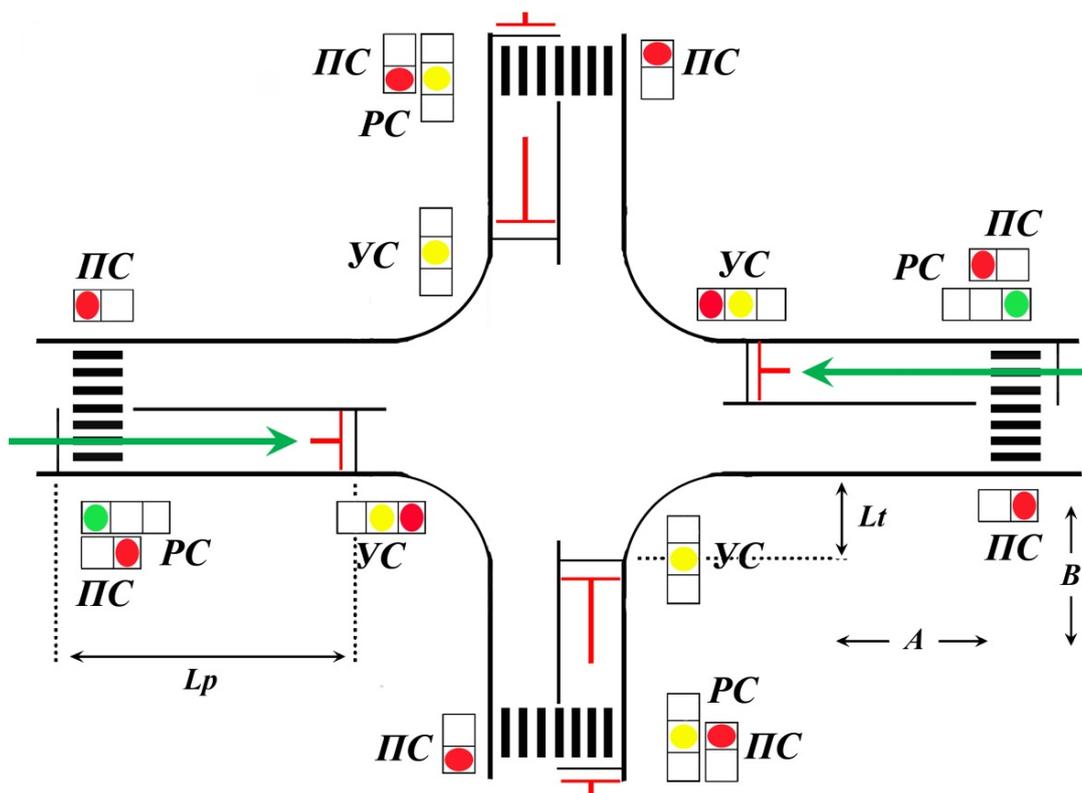


Рисунок 2.11- Равнозначный перекрёсток с установленными на нём светофорами

На рисунке 2.11 нарисован равнозначный перекрёсток с установленными на нём светофорами и приняты следующие обозначения:

РС – разгонный светофор,

УС – управляющий светофор,

ПС – пешеходный светофор,

A, B – направления движения на перекрёстке,

$L_p$  – расстояние между стоп – линиями разгонного и управляющего светофоров,

$L_t$  – расстояние от стоп – линии управляющего светофора до места пересечения транспортных потоков.

На следующих рисунках изображены перекрёстки в интервалах времени:  $t_1 - t_2, t_2 - t_3, t_3 - t_4, \dots, t_{12} - t_1$ .

Приведенные чертежи поясняют принцип работы предлагаемого способа управления транспортными потоками на светофорных объектах с помощью разгонных светофоров (Рисунки 2.12-2.23).

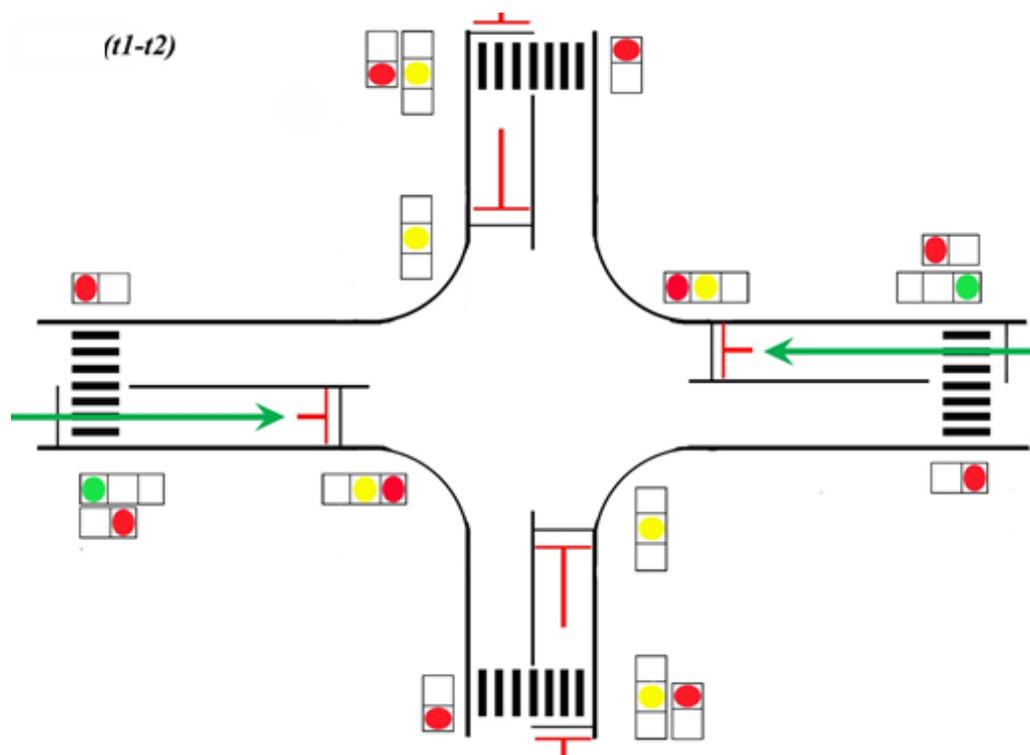


Рисунок 2.12 - В интервале времени  $t_1-t_2$

В интервале времени  $t_1-t_2$  транспортные средства потоков А ускоряются. На РС направлений А загорелся зелёный. Пешеходный переход везде запрещён.

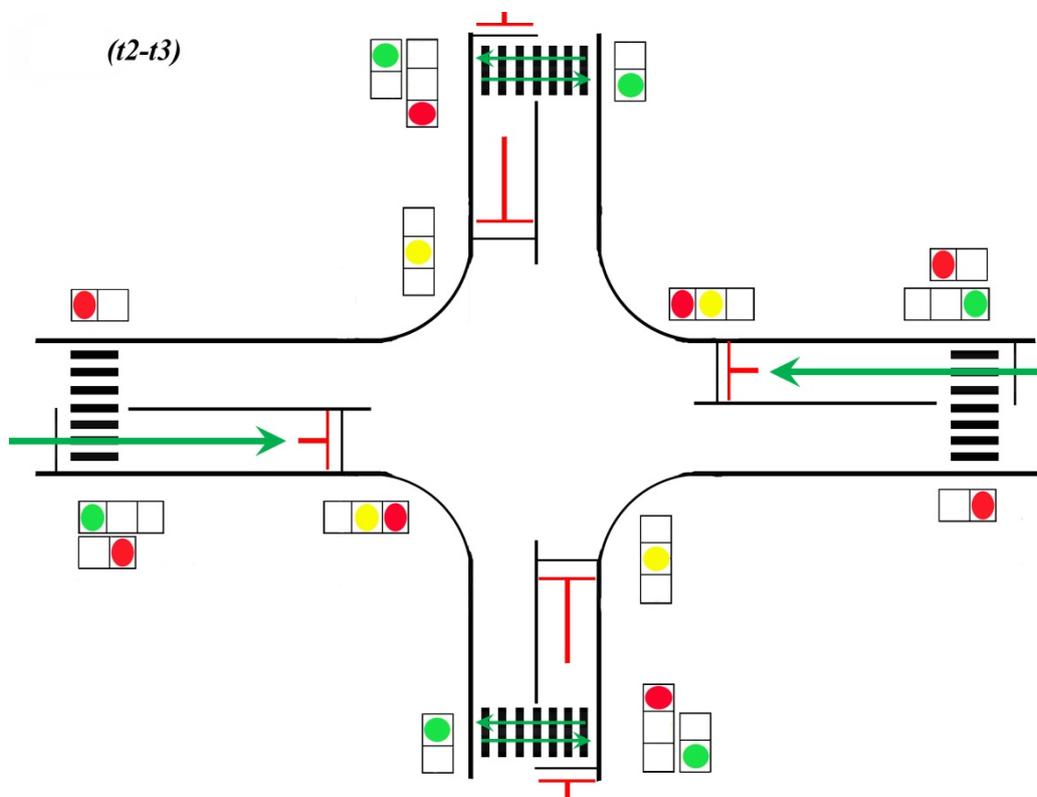


Рисунок 2.13 - В интервале времени  $t_2-t_3$

В интервале времени  $t_2 - t_3$  транспортные средства потоков А ускоряются.  
На УС направлений А красный – желтый.  
Пешеходы переходят проезжую часть В.

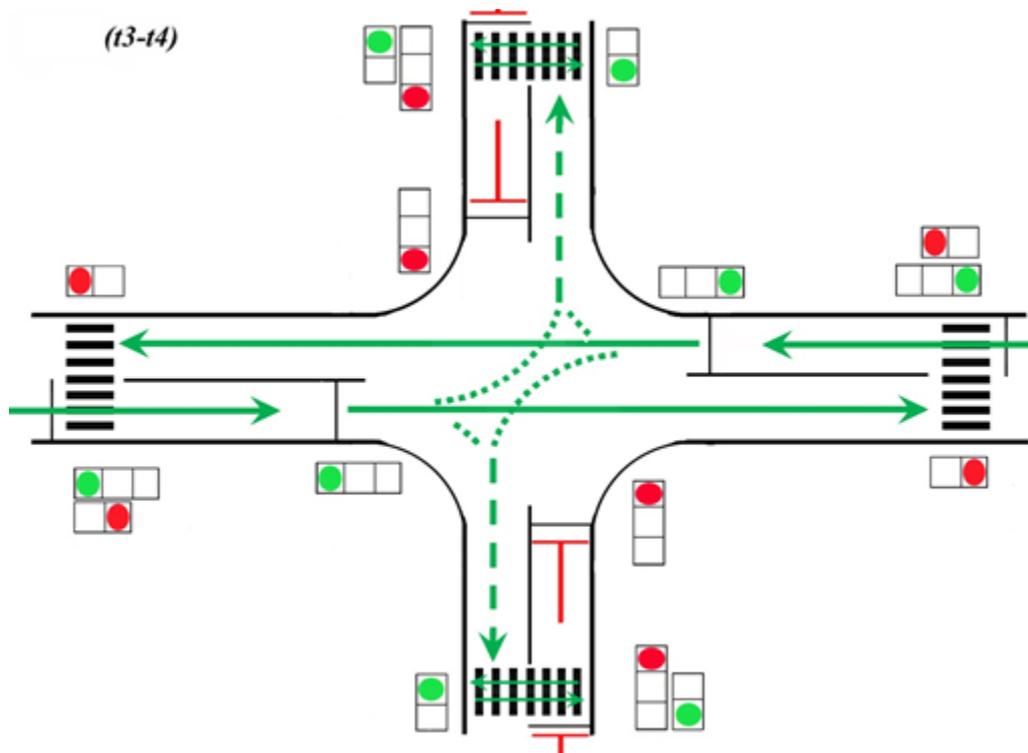


Рисунок 2.14 - В интервале времени  $t_3 - t_4$

В интервале времени  $t_3 - t_4$  транспортные средства потоков А пересекают стоп - линию УС направлений А и выезжают на перекрёсток, пересекают его, осуществляют повороты налево и направо. Повороты осуществлять легче, чем на обычном перекрёстке т.к. имеются значительные расстояния от транспортных потоков А до пешеходных переходов на потоках В. На этих участках перекрёстка транспортные средства могут ожидать возможности пересечь пешеходные переходы. На УС направлений А зелёный.

Пешеходы переходят проезжую часть В.

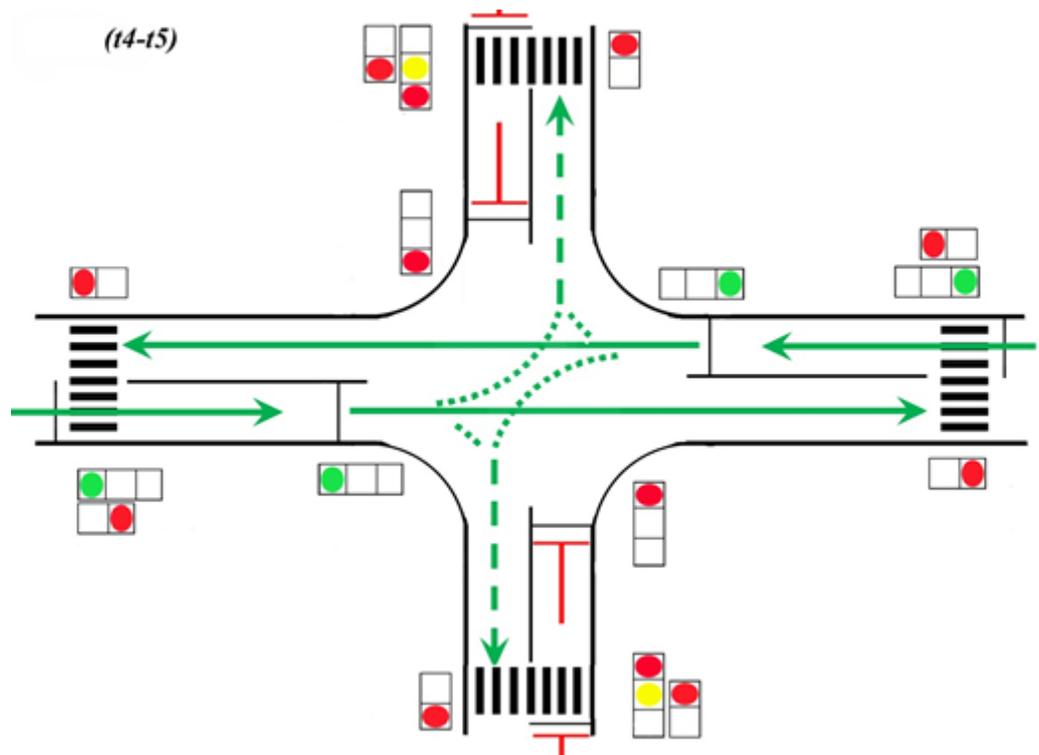


Рисунок 2.15 - В интервале времени  $t_4-t_5$

В интервале времени  $t_4-t_5$  транспортные средства потоков - А выезжают на перекрёсток, пересекают его, осуществляют повороты налево и направо. Поворотам пешеходы не мешают. Пешеходный переход везде запрещён.

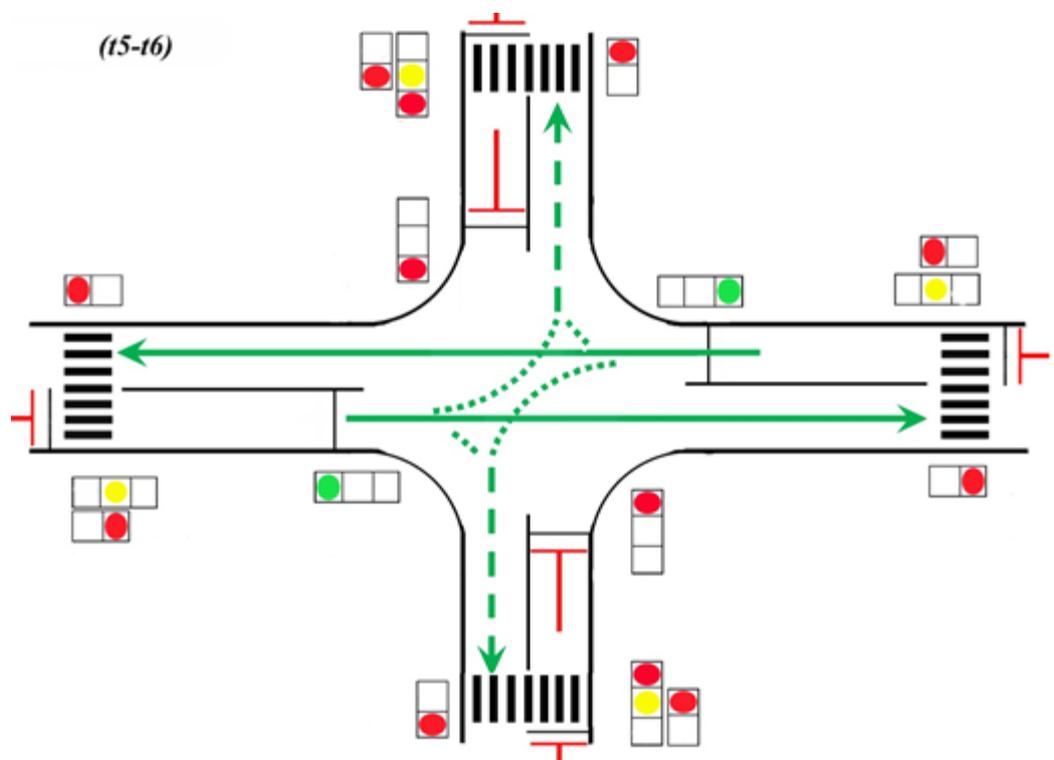


Рисунок 2.16 - В интервале времени  $t_5-t_6$

В интервале времени  $t_5$ –  $t_6$  транспортные средства потоков А выезжают на перекрёсток, пересекают его, осуществляют повороты налево и направо. Поворотам пешеходы не мешают. На РС потоков А желтый. Пешеходный переход везде запрещён.

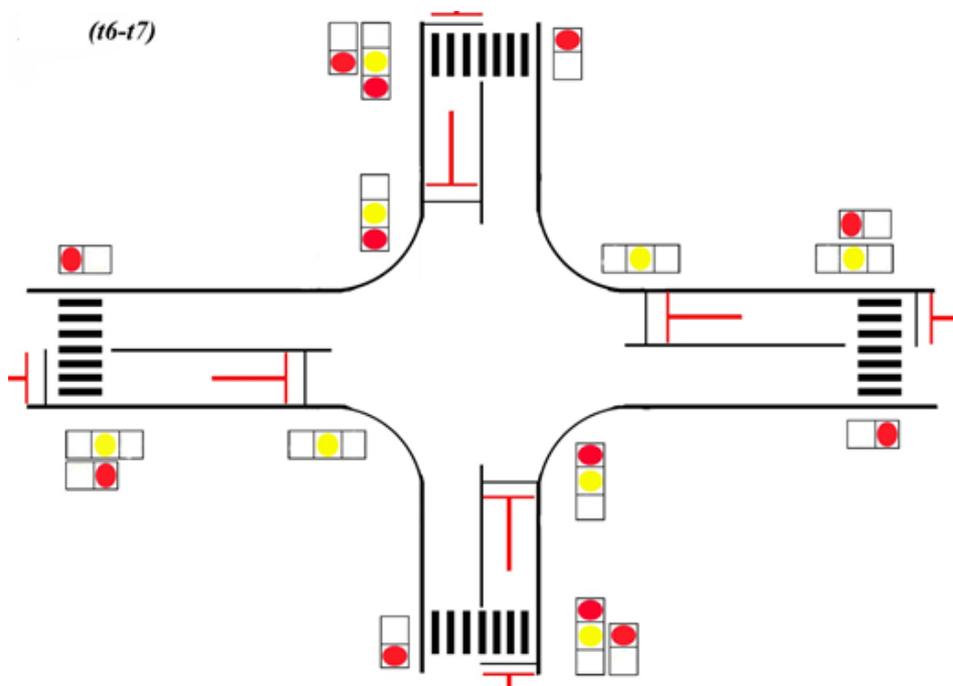


Рисунок 2.17 - В интервале времени  $t_6$ – $t_7$

В интервале времени  $t_6$ – $t_7$  на всех РС и УС направлений А желтый. На всех РС и УС направлений В красный – желтый. Транспортные средства освобождают перекрёсток. Пешеходный переход везде запрещён.

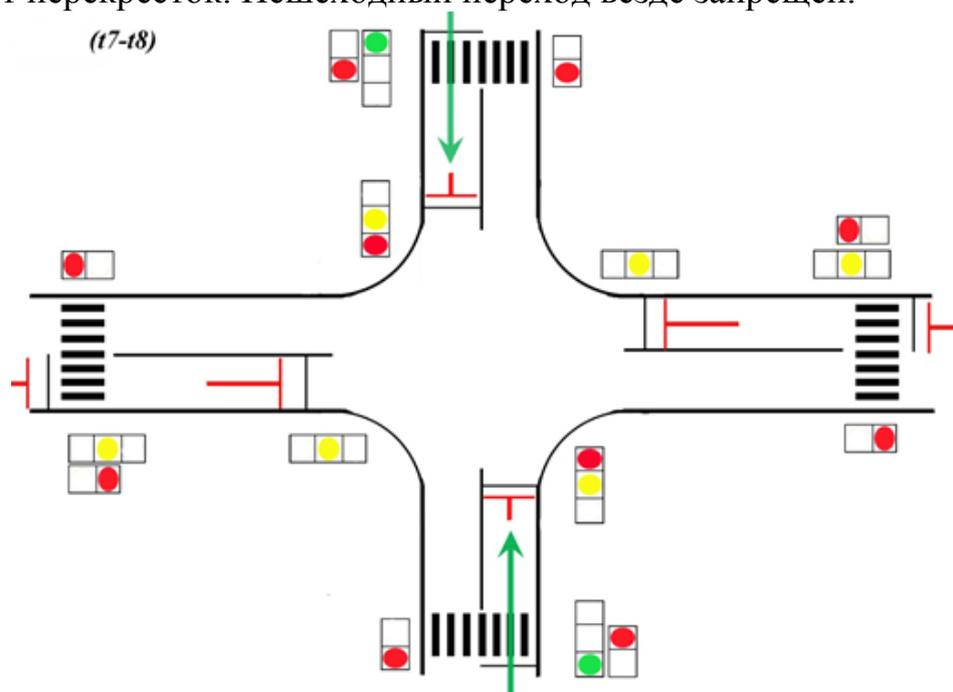


Рисунок 2.18 - В интервале времени  $t_7$ – $t_8$

В интервале времени  $t_7-t_8$  транспортные средства потоков В ускоряются. На РС направлений В зелёный. Пешеходный переход везде запрещён.

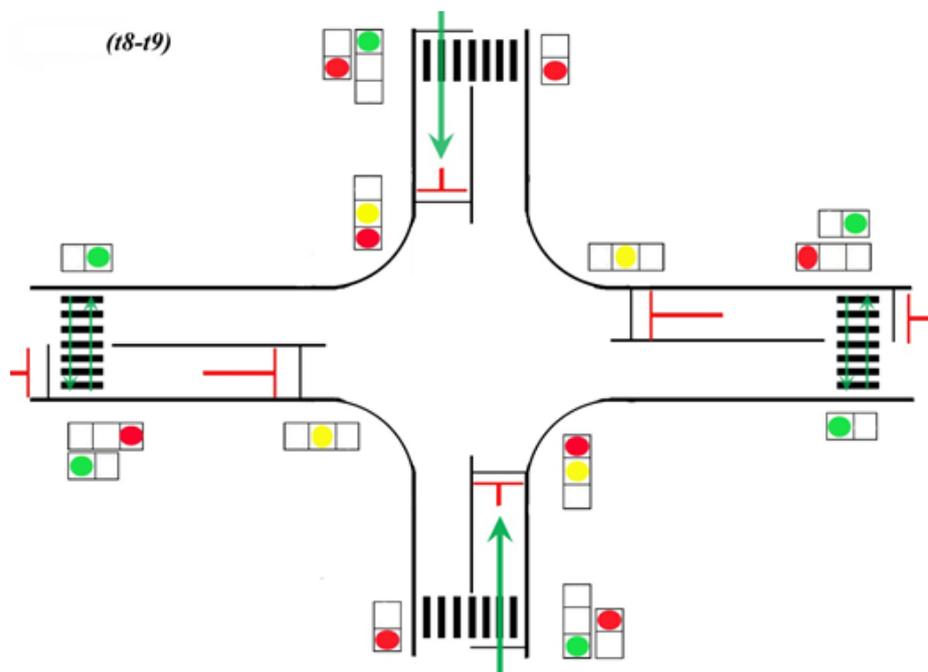


Рисунок 2.19 - В интервале времени  $t_8-t_9$

В интервале времени  $t_8-t_9$  транспортные средства потоков В ускоряются. На УС направлений В красный - желтый. Пешеходы переходят проезжую часть потоков А.

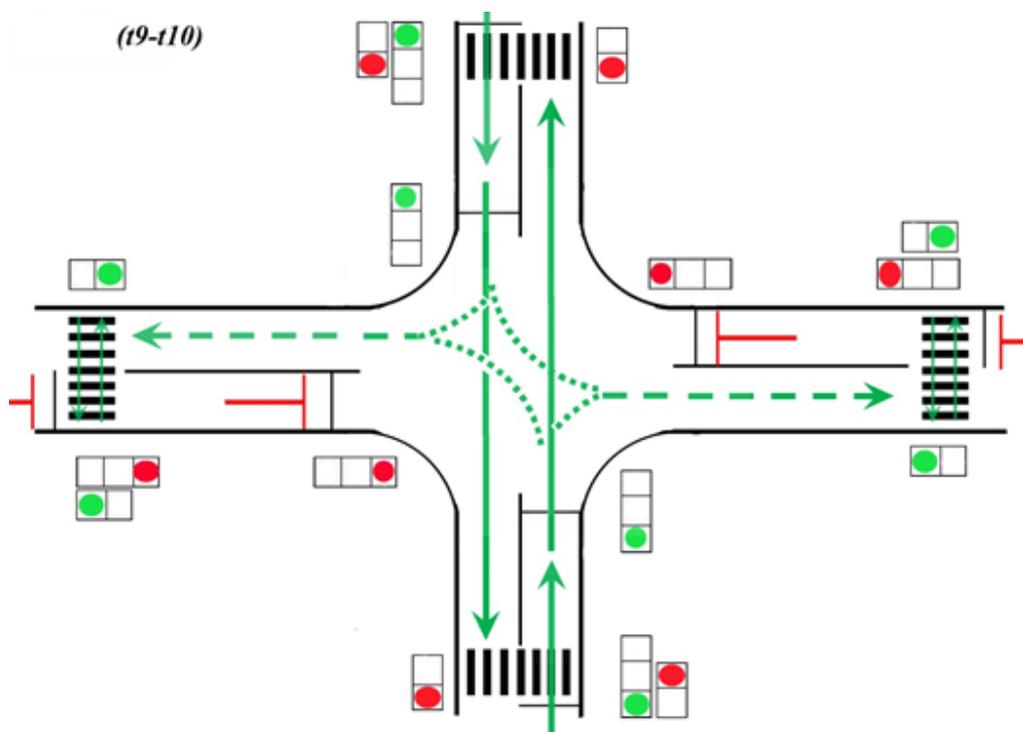


Рисунок 2.20 - В интервале времени  $t_9-t_{10}$

В интервале времени  $t_9-t_{10}$  транспортные средства потоков В пересекают стоп - линию УС направлений В и выезжают на перекрёсток, пересекают его и осуществляют повороты налево и направо. Повороты осуществлять легче, чем на обычном перекрёстке т.к. имеются значительные расстояния от транспортных потоков В до пешеходных переходов на потоках А. На этих участках перекрёстка транспортные средства могут ожидать возможности пересечь пешеходные переходы. На УС направлений В зелёный.

Пешеходы переходят проезжую часть потоков А.

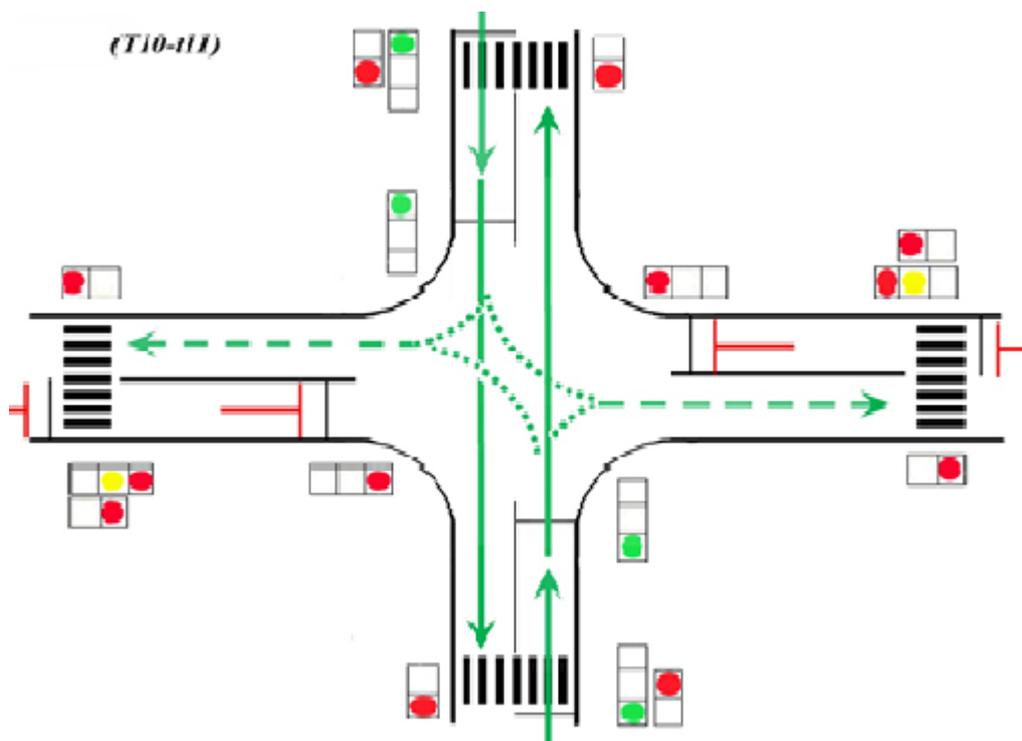


Рисунок 2.21 - В интервале времени  $t_{10}-t_{11}$

В интервале времени  $t_{10}-t_{11}$  транспортные средства потоков В выезжают на перекрёсток, пересекают его, осуществляют повороты налево и направо. Поворотам пешеходы не мешают.

Пешеходный переход везде запрещён.

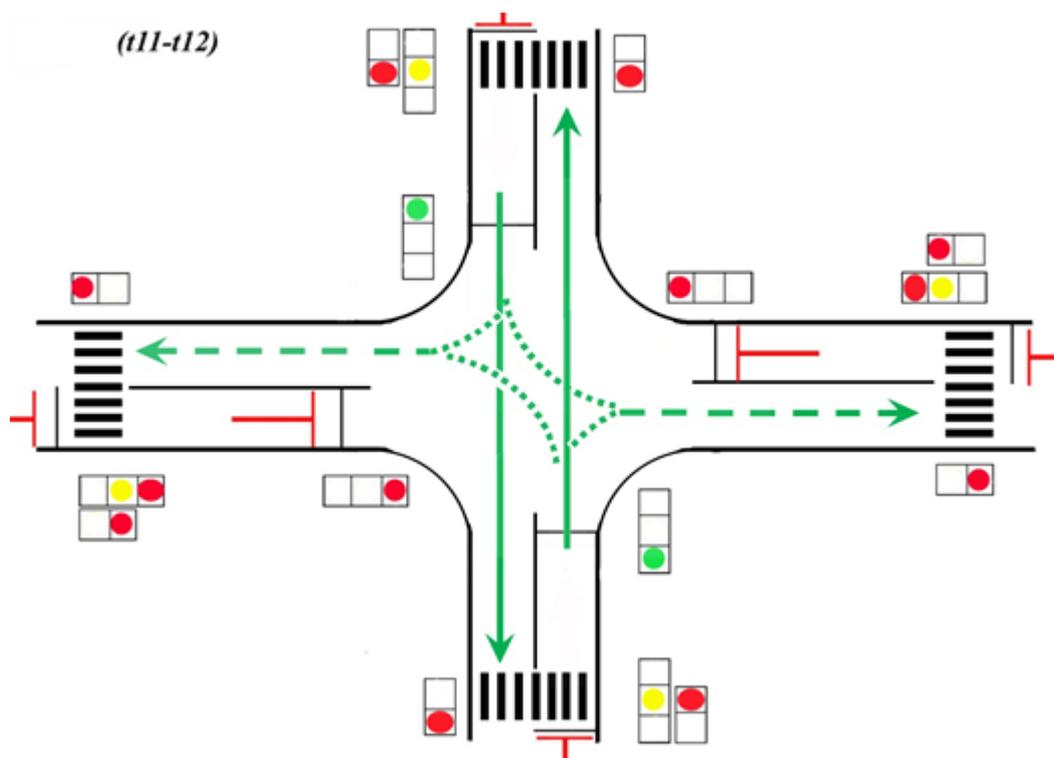


Рисунок 2.22 - В интервале времени  $t_{11}-t_{12}$

В интервале времени  $t_{11}-t_{12}$  транспортные средства потоков В выезжают на перекрёсток, пересекают его, осуществляют повороты налево и направо. Поворотам пешеходы не мешают. На РС потоков В желтый.

Пешеходный переход везде запрещён.

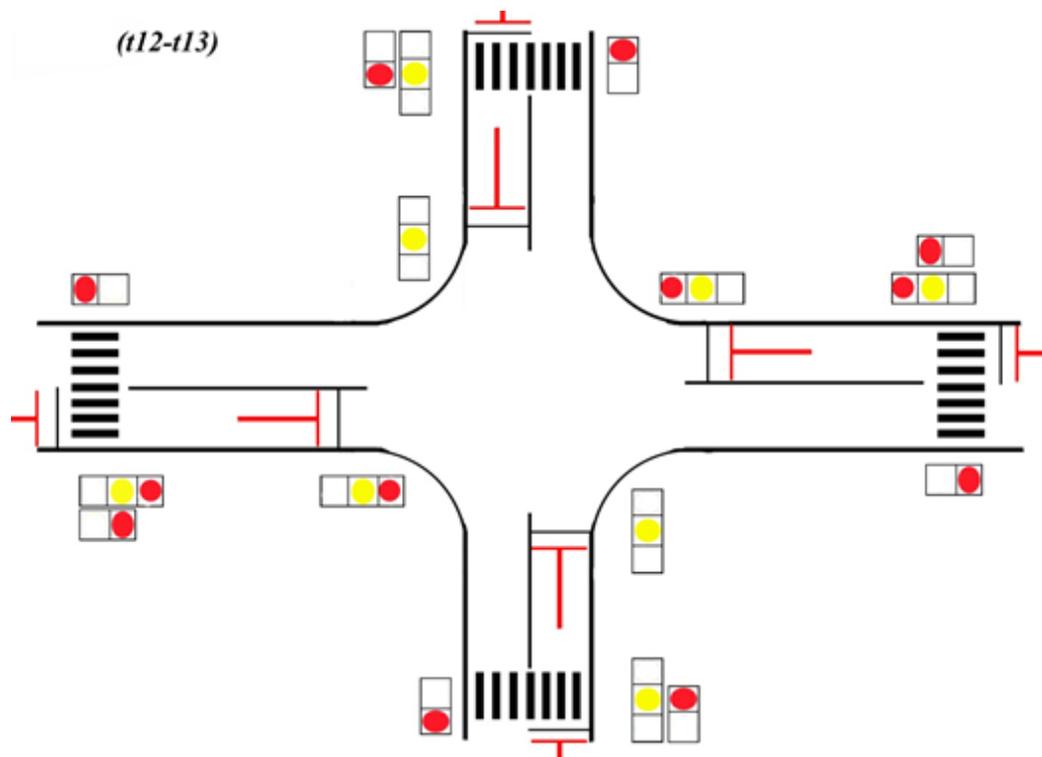


Рисунок 2.23 - В интервале времени  $t_{12}-t_{13}$

В интервале времени  $t_2 - t_1$  на всех РС и УС направлений В желтый. На всех РС и УС направлений А красно – желтый сигналы.

Пешеходный переход везде запрещён.

Следует отметить, что:

- Разгонные светофоры могут применяться перед всеми управляющими светофорами на светофорном объекте, либо их частью;

- При применении разгонных светофоров транспортные средства и пешеходы руководствуются существующими правилами дорожного движения.

Положительные особенности применения разгонных светофоров:

а) Способ управления транспортными потоками на светофорных объектах с помощью разгонных светофоров, заключающийся в разгоне транспортного потока перед прохождением его на разрешающий сигнал управляющего светофора повышает среднюю скорость транспортного потока, проходящего через управляющий светофор и как следствие этого более высокая пропускная способность светофора, перед которым стоит разгонный светофор, перед обычным светофором.

б) Пешеходные переходы на перекрёстках, оборудованных разгонными светофорами более удалены от мест пересечения транспортных потоков, что позволяет аккумулировать на этих участках перекрёстка транспортные средства, поворачивающие налево и направо, и этим создавать меньше помех другим участникам движения на перекрёстке, что увеличивает пропускную способность перекрёстков.

в) Часть времени пересечения пешеходных переходов транспортными потоками, поворачивающими налево и направо на перекрёстках, оборудованных разгонными светофорами, происходит, когда движение пешеходов через пешеходные переходы запрещено, что позволяет создавать меньше помех другим участникам движения на перекрёстке и увеличивает пропускную способность перекрёстков. Это реализуется без увеличения светофорного цикла.

г) Для увеличения пропускной способности существующих перекрёстков не требуется уширение проезжей части.

д) Предлагаемый способ управления транспортными потоками обеспечивает повышение безопасности дорожного движения при высокой эффективности.

### **2.3 Задачи анализа и оптимизации транспортных развязок и размещения городских остановок**

Алматы - один из крупнейших мегаполисов Казахстана. Естественно, что он, как и другие крупные города развитых стран, сталкивается с необходимостью решать проблему дорожных развязок. Сегодня при проектировании автомобильных дорог предпочтение отдают современным технологиям и методам производства изысканий, основанным, прежде всего, на использовании высокопроизводительных методов сбора информации о местности:

- использованию ГИС - технологий при изысканиях автомобильных дорог и сооружений на них;
- методам наземной и аэрокосмической цифровой фотограмметрии;
- системам спутниковой навигации «GPS»;
- методам электронной тахеометрии,
- наземного лазерного сканирования местности и геофизическим методам инженерно - геологических изысканий.

Транспортная развязка - комплекс дорожных сооружений (мостов, туннелей, дорог), предназначенный для минимизации пересечений транспортных потоков и, как следствие, для увеличения пропускной способности дорог. Преимущественно под транспортными развязками понимаются транспортные пересечения в разных уровнях, но термин используется и для специальных случаев транспортных пересечений в одном уровне. На сегодняшний день при строительстве используются новейшие современные технологии при строительстве автотранспортных развязок для улучшения качества и безопасности развязок.

В нашем городе чаще используют такие приборы, как Leica TC 407 производство Швейцария, а также электронные рулетки и системы GPS. При строительстве развязок используются новейшие программы ГИС, такие как Credomix и AutoCAD. Эти программы предназначены для решения задач разных видов и сложностей при проектировании строительства УДС.

Вначале определим типы развязок. Транспортные пересечения в разных уровнях по начертанию их в плане подразделяются на следующие группы [29, с.145]:

- а) клеверообразные;
- б) кольцевые;
- в) петлеобразные;
- г) сложные пересечения с обособленными левоповоротными съездами;
- д) линейные, ромбовидные и комбинированные пересечения в разных уровнях с сочетанием элементов различных видов пересечений, преимущественно таких, как клеверные листья, левоповоротные обособленные съезды, петли и участки перестроений.

Наибольшее распространение получили транспортные пересечения в двух уровнях, которые подразделяются на следующие типы [29, с.146]:

- а) полный клеверный лист;
- б) полный клеверный лист с объездом вокруг прилегающих кварталов;
- в) неполный клеверный лист;
- г) сплюснутый клеверный лист;
- д) с пятью путепроводами типа «крест»;
- е) с отнесенными левыми поворотами;
- ж) с регулируемым движением по второстепенному направлению и т.д.

Каждый из приведенных типов пересечений имеет свои недостатки и преимущества. Определение целесообразности применения того или иного пересечения в разных уровнях в конкретных условиях требует подробного изучения всех перечисленных типов.

### *Транспортное пересечение в двух уровнях – “полный клеверный лист”*

Транспортное пересечение типа “полный клеверный лист” по своему очертанию в плане напоминает контур листа клевера. К преимуществам данной развязки относится то, что для ее устройства требуется не так много места по сравнению с другими видами многоуровневых развязок. Он обеспечивает непрерывной движения транспорта по всем направлениям при пересечении между собой двух магистралей (рисунок 2.24).

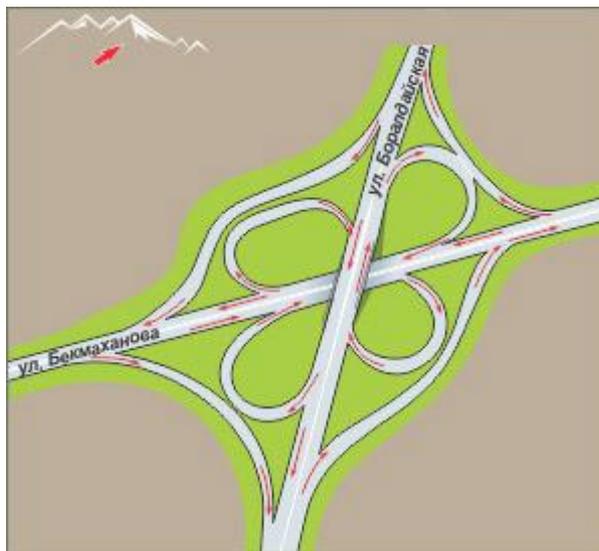


Рисунок 2.24 - Пересечение типа “полный клеверный лист”

Принцип его проектирования заключается в обеспечении такой организации движения, при которой пересекающиеся между собой прямые направления движения осуществляются в двух уровнях, а левоповоротные направления заменяются поворотами вправо по специальным четырем съездам [29, с.148].

### *Пересечение в двух уровнях – “неполный клеверный лист”*

При выборе схемы неполного клеверного листа важно учитывать приоритетные направления съездов. Особенность такого пересечения в том, что некоторые потоки совершают исключительно правые повороты, а некоторые – левые в районе примыкания в одном уровне. Если приоритетные направления совпадают с правоповоротными потоками, развязка будет хорошо работать.

### *Пересечение в двух уровнях типа “сплюснутый клеверный лист”*

В городских условиях для уменьшения занимаемой территории применяется так называемый сплюснутый клеверный лист с левоповоротными съездами, описанными минимальными радиусами и расположенными параллельно рампам тоннеля или эстакады (рисунок 2.26). Сооружение такого пересечения возможно при ширине основной магистрали непрерывного движения не менее 80 – 100 м и связано с затруднениями в пропуске левоповоротных потоков. Преимущество заключается в том, что при относительно малой площади, занимаемой пересечением, обеспечивается непрерывность движения по всем направлениям.



Рисунок 2.26 - Пересечение по типу “сплюснутый клеверный лист”

*Улучшенное пересечение в двух уровнях по типу “распределительное кольцо с пятью путепроводами”*

Пересечение “распределительное кольцо с пятью путепроводами” можно применить при скрещивании двух равнозначущих магистралей с большими сквозными потоками транспорта для полной развязки движения по всем направлениям непосредственно на площади перекрестка (рисунок 2.27).



Рисунок 2.27 - Схема “распределительного кольца” с пятью путепроводами

*Пересечение в двух уровнях на примыкании магистралей по типу “труба”*

Пересечение по типу “труба” применяется на примыкании одной магистрали к другой при Т-образных или У-образных перекрестках.

В городских условиях развязка «труба» так же, как «клеверный лист», сохраняет схему движения, но изменяет планировку: малая ширина улиц прижимает съезды к основному транспортному сооружению. Радиусы левоповоротных съездов при этом уменьшаются до 10-15м.

*Пересечение в двух уровнях по типу “двойная петля”*

Этот тип пересечения можно сооружать при скрещении магистральной улицы, требующей большой пропускной способности, с улицей второстепенного значения. Здесь сквозные потоки транспорта на магистральной улице направляются по прямой, а на второстепенном проезде по разветвленной кривой - через два путепровода одностороннего движения (рисунок 2.28). Все поворотное движение (в том числе и правые повороты) пропускается через путепроводы.

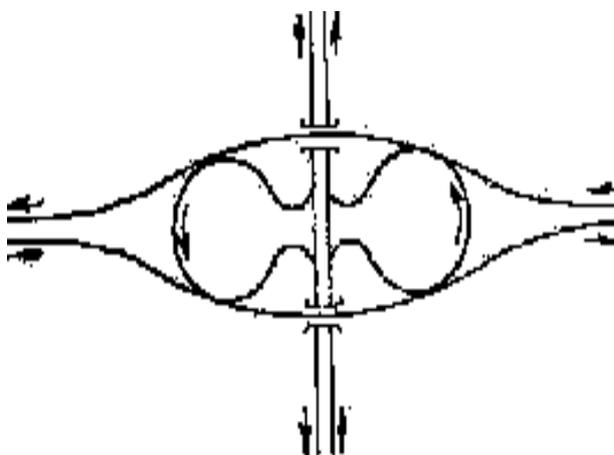


Рисунок 2.28 - Схема пересечения типа "двойная петля"

При строительстве моста Кавацу-Нанадару в Японии инженеры использовали развязку типа двойная петля. Он находится в городе Кавацу. Эта двойная спираль позволяет автомобилям подниматься и спускаться на 45 м, причем, они как будто подвешены между двумя сторонами горы. В диаметре петля равна 80 м, а вся наклонная часть составляет 1,1 км в длину. Мост на шоссе 414 между Токио и полуостровом Ицу был завершён в 1982 году и стал местной достопримечательностью.

*Пересечение в двух уровнях с пятью путепроводами типа «крест»*

Преимущество данного пересечения заключается в отсутствии лишних переходных кривых, в том, что все повороты идут прямо, без отклонений, исключая перепробеги при левоповоротном движении (характерные для пересечений «клеверного» типа). Длина пути каждого левого поворота на 500-600 м короче, чем в «клеверном листе». В данном случае лучше организуется и движение, так как машины, сворачивающие налево, придерживаются левой стороны, направо - правой стороны, а прямой сквозной поток идет посередине.

При выборе типа транспортной развязки в разных уровнях необходимо, прежде всего, учитывать интенсивность и характер движения на узле. Интенсивность движения на узле характеризуется размерами транспортных потоков, проходящих через узел в определенный отрезок времени. В

транспортных расчетах принимаются максимальные размеры движения по всем направлениям за один час.

При выборе типа пересечения в разных уровнях наряду с перечисленными выше факторами, прежде всего, следует учитывать стоимость его строительства и дальнейшей эксплуатации. Проектирование пересечений в разных уровнях должно учитывать также удобство реконструкции и стадийности строительства по мере роста размеров движения.

Надо предложить математическую модель транспортной развязки с использованием теории S-гиперсетей и в зависимости от полученного результата построить конструкцию развязки со следующими свойствами (рисунок 2.34):

- поток шел непрерывно. Под непрерывностью подразумевается следующее, что поток автомобилей должен не останавливаться при выезде или въезде на другую дорогу, т.е. не должны создаваться задержки в движении автомобилей;

- конструкция была оптимальной. Под оптимальностью понимается следующее: построенное сооружение не содержало бы лишних (не используемых) дорог.

Начальные данные (рисунок 2.34):

- место, где нужно построить транспортную развязку и дороги, проходящие через данное место, т.е. первичная сеть PS с её пропускной способностью дороги  $D_k$ , где  $k$ -количество дорог.

- матрица потоков для данных дорог, т.е. вторичная сеть WS (взвешенные ориентированные ребра  $P_{ij}$ ).

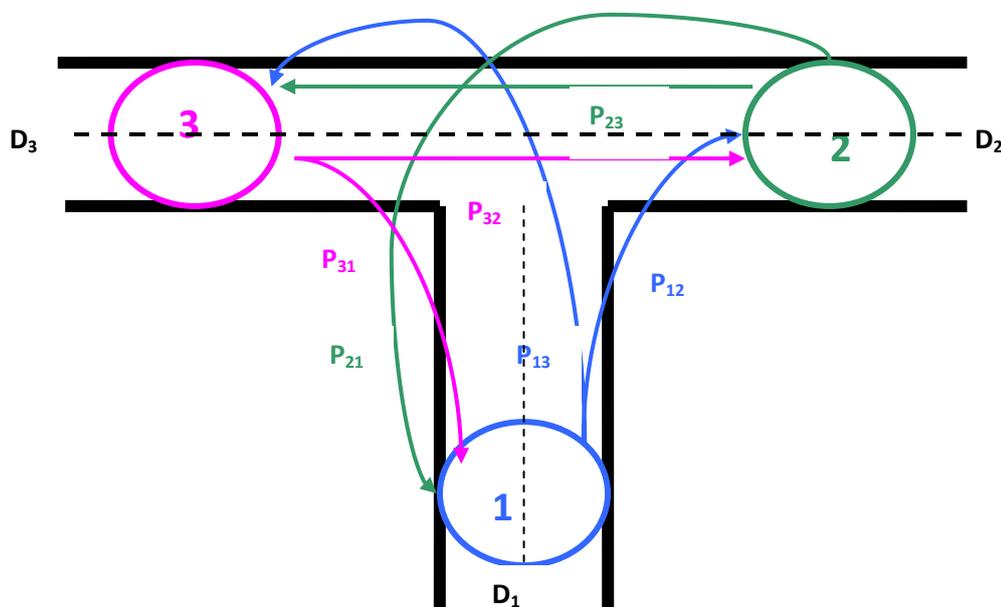


Рисунок 2.34 - Начальные данные

*Алгоритм построения транспортных развязок с использованием теории S-гиперсетей*

Алгоритм:

- выбирается “главную” дорогу;
- по матрице потоков выбирается пригодная развязка;
- поиск дополнительных возможностей.

*Шаг 1.* “Главная” дорога – это дорога, у которой поток максимален; если потоки равнозначны, то можно взять за главную одну из них и обозначить ее “1”. Перед подъездами к перекрестку поставить полосу замедления, чтоб автотранспорт смог перестроиться на нужную полосу.

*Шаг 2.* Имеется матрица потоков на перекрестке  $P_{ij}$ . По данной матрице предлагается выбрать необходимую развязку следующим образом:

а) для 4 дорог имеем:

Если потоки  $P_{ij}$  удовлетворяют следующей системе (2.4.1), тогда выбираем развязку типа “полный клеверный лист” или “сплюснутый клеверный лист”.

$$\begin{cases} P_{i,i+1} + P_{i,i+2} + P_{i,i+3} < D_i \\ P_{i,i+3} > \frac{1}{4} D_i \\ P_{i,i+1} > \frac{1}{4} D_i \end{cases} \quad (2.4.1)$$

где  $\forall i = \overline{1,4}$  и  $i+4 \equiv i$ .

Если потоки  $P_{ij}$  удовлетворяют следующей системе (2.4.2), тогда выбираем развязку типа “неполный клеверный лист” с четырьмя односторонними проездами.

$$\begin{cases} P_{i,i+1} + P_{i,i+2} + P_{i,i+3} < D_i \\ P_{i,i+1} + P_{i,i+3} > \frac{1}{4} D_i \\ P_{i,i+3} < \frac{1}{4} D_i \end{cases} \quad (2.4.2)$$

где  $\forall i = \overline{1,4}$  и  $i+4 \equiv i$ .

Если потоки  $P_{ij}$  удовлетворяют следующей системе (2.4.3), тогда выбираем развязку типа “неполный клеверный лист” с двумя съездами и проездами двустороннего движения, расположенными в соседних четвертях перекрестка.

$$\begin{cases} P_{i,i+2} + P_{i,i+3} < D_i \\ P_{i,i+1} = 0 \\ P_{i+1,i} + P_{i+1,i+2} + P_{i+1,i+3} < D_i \\ P_{i,i+3} > \frac{1}{4} D_i \\ P_{i+2,i+3} + P_{i+2,i} < D_i \\ P_{i+2,i+1} = 0 \\ P_{i+3,i+2} = 0 \\ P_{i+3,i} = 0 \end{cases} \quad (2.4.3)$$

где  $\exists i = \overline{1,4}$  и  $i+4 \equiv i$ .

Если потоки  $P_{ij}$  удовлетворяют следующей системе (2.4.4), тогда выбираем развязку типа "неполный клеверный лист" с двумя съездами и проездами двустороннего движения, расположенными в накрест лежащих четвертях перекрестка.

$$\left\{ \begin{array}{l} P_{i,i+2} + P_{i,i+3} < D_i \\ P_{i,i+1} = 0 \\ P_{i+1,i} + P_{i+1,i+2} + P_{i+1,i+3} < D_i \\ P_{i+1,i} < \frac{1}{4} D_i \\ P_{i+2,i} + P_{i+2,i+1} < D_i \\ P_{i+3,i} + P_{i+3,i+1} + P_{i+3,i+2} < D_i \\ P_{i+3,i+2} < \frac{1}{4} D_i \end{array} \right. \quad (2.4.4)$$

где  $\exists i = \overline{1,4}$  и  $i+4 \equiv i$ .

Если потоки  $P_{ij}$  удовлетворяют следующей системе (2.4.5), тогда выбираем развязку типа "неполный клеверный лист" с тремя съездами и проездами двустороннего движения, расположенными в трех четвертях перекрестка.

$$\left\{ \begin{array}{l} P_{i,i+1} + P_{i,i+2} + P_{i,i+3} < D_i \\ P_{i,i+3} > \frac{1}{4} D_i \\ P_{i+1,i} + P_{i+1,i+2} + P_{i+1,i+3} < D_i \\ P_{i+1,i} > \frac{1}{4} D_i \\ P_{i+2,i} + P_{i+2,i+1} + P_{i+2,i+3} < D_i \\ P_{i+2,i+1} < \frac{1}{4} D_i \\ P_{i+3,i+1} + P_{i+3,i+2} < D_i \\ P_{i+3,i} = 0 \end{array} \right. , \quad (2.4.5)$$

где  $\exists i = \overline{1,4}$  и  $i+4 \equiv i$ .

Если потоки  $P_{ij}$  удовлетворяют следующей системе (2.4.6), тогда выбираем развязку типа "распределительное кольцо с пятью путепроводами".

$$\left\{ \begin{array}{l} P_{i,i+1} + P_{i,i+2} + P_{i,i+3} < D_i \\ P_{i,i+1} + P_{i,i+3} + P_{i,i} + P_{i+2,i+1} + P_{i+2,i+2} + P_{i+3,i+3} + P_{i+3,i+2} < \max\{D_i\} \end{array} \right. , \quad (2.4.6)$$

где  $\forall i = \overline{1,4}$  и  $i+4 \equiv i$ .

б) для 3 дорог имеем:

Если потоки  $P_{ij}$  удовлетворяют следующей системе (2.4.7), Тогда выбираем развязку типа Т-образного примыкания.

$$\begin{cases} P_{i,i+1} + P_{i,i+2} < D_i \\ P_{i,i+2} > \frac{1}{4} D_i \end{cases} \quad (2.4.7)$$

где  $\forall i = \overline{1,4}$  и  $i+4 \equiv i$ .

Если к (2.4.7) добавить еще  $\exists j P_{j,j} \neq 0$  (2.4.8), тогда можно выбрать развязку полуклеверного типа примыкания или кольцевого типа примыкания.

Диаграмма деятельности для определения типа развязки показано на рисунке 2.35.

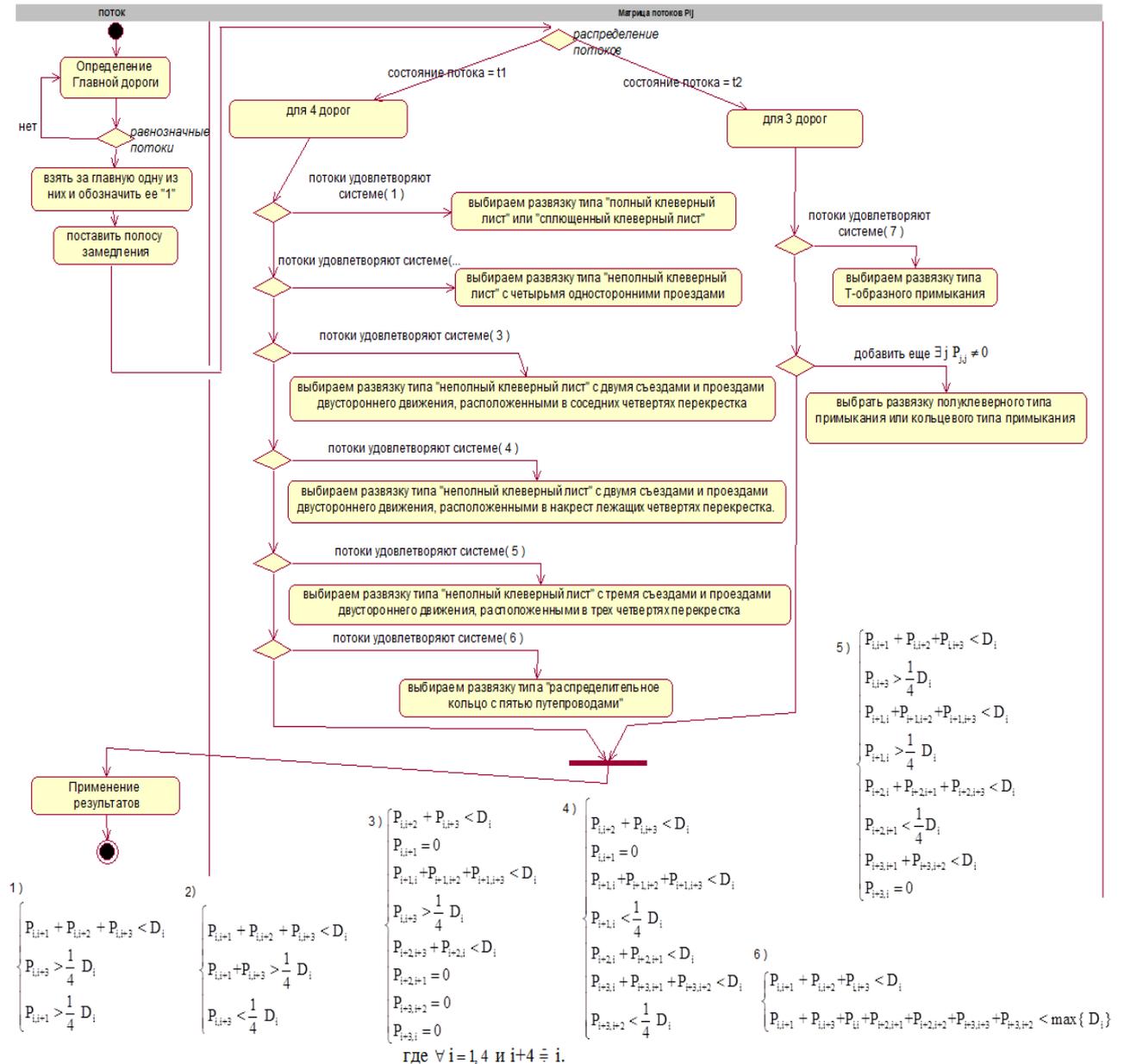


Рисунок 2.35 - Диаграмма деятельности

**Шаг 3.** Как уже отмечалось, при построении развязки нужно учитывать территорию, прилегающую к перекрестку, например, архитектурные

сооружения и т.д. Многие развязки занимают значительно большую территорию, чем занимаемая территория перекрестка. Но мы показали, что для данного перекрестка необходима такая то развязка, согласно шагу 2.

Предлагаются следующие методы или модернизации для сокращения территории занимаемой развязкой, при этом поток автотранспорта будет так же идти непрерывно:

а) для поворота направо создать полосу разгона.

Если прилегающая территория к перекрестку не позволяет построить поворот направо, то можно осуществить поворот при помощи полосы разгона, т.е. подъезжая к перекрестку повернуть на полосу разгона (рисунок 2.36). Полоса разгона нужна для того, чтоб автотранспорт, осуществивший поворот, смог вписаться в транспортный поток.

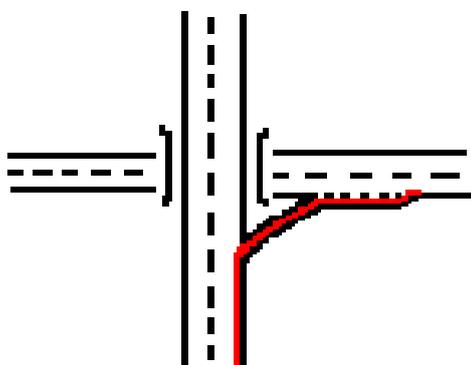


Рисунок 2.36 - Поворот направо при помощи полосы разгона

б) поворот налево осуществить на следующем перекрестке.

Если прилегающая территория к перекрестку не позволяет построить поворот налево, то можно осуществить поворот при помощи ближайшего перекрестка справа или сверху, на котором можно будет развернуться (Рисунок 2.37)

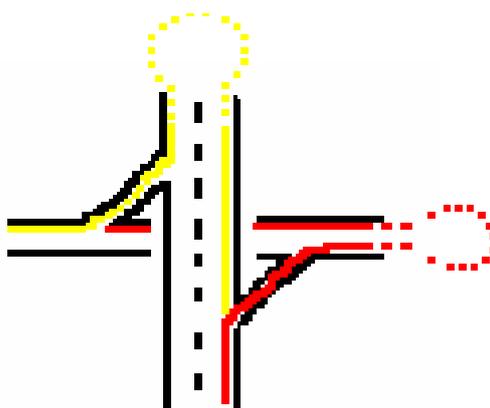


Рисунок 2.37 - Поворот налево при помощи другого перекрестка

Можно выбрать этот перекресток, если выполняется следующее неравенство:

$$t_{\text{left}} \geq \frac{2 * \min\{L_1, L_2\}}{v}$$

Где:  $t_{\text{left}}$  – среднее время для осуществления поворота налево;

$L_1$  – расстояние для ближнего правого перекрестка;

$L_2$  – расстояние для ближнего верхнего перекрестка;

$v$  – средняя скорость движения.

*Замечание:*

а) Использовалось следующее утверждение:

Если число поворотного транспорта превышает четверть от всего потока, то для непрерывного движения без светофорного регулирования необходимо построить специальную поворотную дорогу.

б) Алгоритм работает для  $k=3,4$ . При  $k>4$  имеем:

Если все потоки будут существенны, то будет развязка огромных размеров и не пригодная к городским условиям. И для построения развязки предлагается следующее:

а) “распределительное кольцо” с  $k$  входящими дорогами (рисунок 2.38);

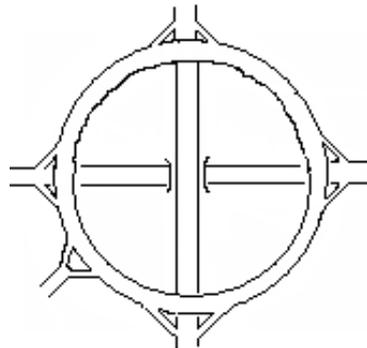


Рисунок 2.38 - “Распределительное кольцо” с 5 дорогами

б) развязка со светофорным регулированием.

Таким образом, алгоритм строить оптимальную транспортную развязку для данного перекрестка.

Рассмотрим теперь преимущества данного сооружения:

а) Существенно уменьшается время прохождения данного узла автотранспортом.

Время прохождения узла при развязке по всем направлениям существенно быстрее, чем время прохождения узла при светофорном регулировании за счет того, что поток начинает движение только по зеленому сигналу.

б) Существенно снижаются затраты.

При светофорном регулировании на перекрестке при насыщенном потоке неизбежно возникают пробки. По подсчетам экспертов каждый автомобиль, застрявший в столичной пробке, стоит государству около 1500тенге в час. Затор в пробке означает рост расхода бензина, дополнительную амортизацию транспорта, помехи в рабочем графике людей, которые застряли на дороге. Моделирование с использованием с использованием теории S-гиперсетей при построении развязки показывает, что на данном перекрестке эти показатели существенно снизятся.

## **Выводы к разделу 2**

Проведен анализ методов моделирования и управления движением транспорта через перекресток. По результатам сравнения автоматических контроллеров, основанных по нечеткой логике, показали выигрыш по времени прохождения автомобилями перекрестка по сравнению с обычными методами, в которых время горения сигналов светофора было фиксировано.

Рассмотрены и доказаны возможности определения пропускной способности УДС с помощью сетей Петри.

Рассмотрено моделирование перекрестка и блока перекрестков с использованием нечеткой логики в управлении светофором. Результаты сравнения автоматических контроллеров, основанных на нечеткой логике, показали выигрыш во времени прохождения автомобилями перекрестка по сравнению с обычными методами.

Рассмотрены способы управления транспортными потоками на светофорных объектах с помощью разгонных светофоров. На основе проведенного анализа выявлены положительные особенности применения разгонных светофоров.

Проведен анализ основных типов развязок. Построена математическая модель для каждого типа развязок с использованием с использованием теории S-гиперсетей.

Предложен алгоритм построения оптимальной транспортной развязки, при котором существенно уменьшается время прохождения данного узла автотранспортом, и снижаются затраты.

Основные результаты исследований по данному разделу содержится в следующих публикациях [39-43].

### 3 МЕТОДЫ И РЕШЕНИЯ ЗАДАЧ УПРАВЛЕНИЯ ПОТОКАМИ ТРАНСПОРТА В МЕГАПОЛИСЕ С ИСПОЛЬЗОВАНИЕМ ТЕОРИИ S-гиперсети

#### 3.1 Задача нахождения максимального потока и методы решения

Для того чтобы рационально организовать движение транспортных потоков необходимо: оценить максимальный поток в сети, найти наиболее эффективное распределение потока, выявить узкие места и своевременно их ликвидировать. Одновременно с этими задачами нужно оценить суммарные затраты времени транспортных средств при их движении из начального пункта в конечный.

Вопросам нахождения интегрального максимального потока посвящены работы исследователей: Жогаль С.И., Максимей И.В. [44], Зайченко Ю.П. [45], Поляков К.Ю. [46], Соболев И.М. [47], Сукач Е.И. [48] и др. Проведен анализ ряда литературных источников [44-48] по нахождению максимального потока, рассмотрены аналитические модели исследования операций и теории автоматического управления. Также проанализировано применение имитационного моделирования для исследования динамики транспортных потоков региона. На основе данного анализа была выявлена актуальность разработки алгоритма суперпозиции потоков в соответствии с теоремой Форда-Фалкерсона, позволяющего определить интегральный максимальный поток для участка региональной сети.

##### 3.1.1 Решение задачи нахождения максимального потока в транспортной сети с использованием алгоритма Форда-Фалкерсона

Пусть  $G = (N, A)$  - ориентированная сеть с одним источником  $s \in N$  и одним стоком  $t \in N$ , и пусть дуги  $(i, j) \in A$  имеют ограниченную пропускную способность. Задача о максимальном потоке заключается в поиске таких потоков по дугам, принадлежащим множеству  $A$ , что результирующий поток, протекающий из источника  $s$  в сток  $t$ , является максимальным. Предполагается, что в источник может поступать неограниченный поток, для каждого промежуточного узла сети выполняется условие сохранения потока, а пропускная способность  $U_{ij}$  каждой дуги представляет собой конечную верхнюю границу потока  $f_{ij}$  по этой дуге.

Задача о максимальном потоке может быть сформулирована в виде следующей задачи:

$$\text{максимизировать } \sum_{i \neq n} f_{in} \quad (3.1)$$

$$\text{при условии, что } \sum_j f_{ij} - \sum_j f_{ji} = 0, \quad i \neq 1, i \neq n, \quad (3.2)$$

$$0 \leq f_{ij} \leq U_{ij}, \quad (i, j) \in A. \quad (3.3)$$

Для ее решения можно воспользоваться обычным симплексным методом. Однако существует более эффективная процедура поиска решения данной задачи. Алгоритм начинает работу с некоторого допустимого решения. Затем

выполняется процедура расстановки пометок, разработанная Фордом и Фалкерсоном [44, с.85], с помощью которой определяется другой допустимый поток большей величины. В данном алгоритме узлы рассматриваются как промежуточные пункты передачи потока, а дуги - как распределительные каналы. Для формального описания алгоритма необходимо ввести два основных понятия - пометки и аугментальные пути потока [45, с.176; 46, с.8].

Пометка узла используется для указания как величины потока, так и источника потока, вызывающего изменение текущей величины потока по дуге, соединяющей этот источник с рассматриваемым узлом. Если  $q_j$  единиц потока посылается из узла  $i$  в узел  $j$  и вызывает увеличение потока по этой дуге, то узел  $j$  помечается из узла  $i$  символом  $+ q_j$ .

В данном случае узлу  $j$  приписывается пометка  $[+ q_j, i]$ . Аналогично если посылка  $q_j$  единиц потока вызывает уменьшение потока по дуге, то узел  $j$  помечается из узла  $i$  символом  $- q_j$ . В данном случае узлу  $j$  приписывается пометка  $[- q_j, i]$ .

Текущий поток из узла  $i$  в узел  $j$  увеличивается, когда  $q_j$  единиц дополнительного потока посылается в узел  $j$  по ориентированной дуге  $(i, j)$  в направлении, совпадающем с ее ориентацией. В данном случае дуга  $(i, j)$  называется *прямой*.

Текущий поток из  $i$  в  $j$  уменьшается, когда  $q_j$  единиц потока посылается в узел  $j$  по ориентированной дуге  $(i, j)$  в направлении, противоположном ее ориентации. В этом случае дуга  $(i, j)$  называется *обратной*.

Если узел  $j$  помечается из узла  $i$  и дуга  $(i, j)$  прямая, то поток по данной дуге увеличивается и величина, соответствующая оставшейся неиспользованной пропускной способности дуги, должна быть нужным образом скорректирована. Данную величину называют остаточной пропускной способностью дуги. Если некоторому узлу приписывается пометка и при этом используется прямая ветвь, то она может иметь только положительную «остаточную пропускную способность». Кроме того, узел  $j$  может быть помечен из узла  $i$  только после того, как узлу  $i$  приписана пометка.

Аугментальный путь потока из  $s$  в  $t$  определяется как связанная последовательность прямых и обратных дуг, по которым из  $s$  в  $t$  можно послать несколько единиц потока. Поток по каждой прямой дуге увеличивается, не превышая при этом ее пропускной способности, а поток по каждой обратной дуге уменьшается, оставаясь при этом неотрицательным.

Аугментальный путь потока используется для выбора такого способа изменения потока, при котором поток в узле  $t$  увеличивается и при этом для каждого внутреннего узла сети не будет нарушено условие сохранения потока.

Алгоритм Форда-Фалкерсона имеет ряд ограничений, которые на практике не выполняются:

Во-первых, пропускные способности ветвей сети должны быть целыми неотрицательными детерминированными числами. В реальной сети пропускная способность не является постоянной и зависит от таких вероятностных факторов, как загруженность, состояние дороги, параметры внешней среды. Загруженность на различных участках дороги бывает различной и зависит от

наличия внутренних транспортных потоков на данном участке, которые могут рассматриваться как помехи при передвижении транспортной единицы из начального пункта сети в конечный. Состояние дороги определяется её изношенностью, условиями эксплуатации, влиянием погодных условий. Параметры внешней среды изменяются в зависимости от времени года и времени суток. Как правило, в транспортной сети перечисленные факторы являются взаимосвязанными.

Во-вторых, выполнение алгоритма Форда-Фалкерсона предлагается начинать с некоторого, уже существующего в сети, начального потока, в то время как выбор начального потока не изменяет величину максимального потока, а изменяет распределение максимального потока в сети [49, с.78; 50, с.683].

В-третьих, задача о поиске максимального потока решается для единственного потока через сеть, т.е. в самой сети должен быть только один вход – узел сети, через который транспортные средства попадают в сеть, и только один выход – узел сети, через который они выходят из сети.

Исходя из вышеизложенного, в качестве выхода из положения можно прибегнуть к имитационному моделированию транспортных потоков в сети, позволяющему учесть большое число внешних факторов и как можно ближе привязать программную модель к её реальному представлению. Метод позволяет определить максимальный поток и его распределение по ветвям сети, обладающее наилучшей эффективностью среди остальных вариантов распределения потоков в сети, пропускные способности ветвей которой изменяются во времени случайным образом.

Далее рассматривается задача нахождения интегрального максимального потока в сети, имеющей заданное количество входов и выходов, определяющих некоторое направление движения транспорта в сети. Предлагается метод определения максимального потока в региональной сети для заданного направления, снимающий ограничения алгоритма Форда-Фалкерсона.

Для нахождения интегрального максимального потока транспортной сети в заданном направлении (ZV) для каждого временного интервала составляются матрицы величин максимальных потоков и эффективностей этих потоков, элементами которых являются значения максимальных потоков и эффективностей потоков по каждому сочетанию входа и выхода. Матрицу величин потоков обозначим  $\varphi = \|\varphi_{ij}^{\max}\|, i=\overline{1,m}, j=\overline{1,n}$ . Матрицу эффективностей потоков обозначим  $\Phi = \|\Phi_{ij}\|, i=\overline{1,m}, j=\overline{1,n}$ . Полученные матрицы нормируются по максимальному элементу:

$$\varphi_{ij}^* = \frac{\varphi_{ij}^{\max}}{\varphi_{ij}^{\max}}, \Phi_{ij}^* = \frac{\Phi_{ij}}{\Phi_{ij}^{\max}}, i=\overline{1,m}, j=\overline{1,n} \quad (3.4)$$

В результате все элементы матриц  $\varphi^*, \Phi^*$  удовлетворяют неравенствам  $0 < \varphi_{ij}^* < 1, 0 < \Phi_{ij}^* < 1$ . В прямоугольной системе координат  $\varphi^* \times \Phi^*$  отмечаем точки  $(\varphi_{ij}^*; \Phi_{ij}^*)$ . Причем, в силу того, что элементы матриц

нормированы по максимальному элементу, все точки будут находиться в пределах единичного квадрата, левый нижний угол которого совмещен с началом координат.

Все точки единичного квадрата сравниваются при помощи некоторой метрики. Составляется список потоков, в котором все элементы ранжируются от «наихудшего» по эффективности к «наилучшему» в соответствии с данными единичного квадрата. Одновременно составляется матрица достижимости  $D = \|d_{ij}\|$ ,  $i = \overline{1, m}$ ,  $j = \overline{1, n}$ , где  $d_{ij} = 1$ , если есть поток из  $i$ -того входа в  $j$ -й выход, и  $d_{ij} = 0$ , если нет потока из  $i$ -того входа в  $j$ -й выход.

Список просматривается, начиная с потока, наихудшего по эффективности. Текущий поток исключается из списка, если его исключение не оставляет ни одну начальную вершину без исходящего потока и не оставляет ни одну конечную вершину без входящего потока. При исключении из списка потока, который идет из  $i$ -того входа в  $i$ -тый выход, модифицируется матрица достижимости  $D$ , в которой на пересечении  $i$ -той строки и  $i$ -того столбца единица заменяется нулем.

Таким образом, текущий поток из  $i$ -того входа в  $j$ -тый выход исключается из списка в том случае, если после его исключения и модификации матрицы достижимости  $D$  в её  $i$ -той строке будет хотя бы одна единица и в  $j$ -том столбце также будет хотя бы одна единица. Если же исключение потока ведет к тому, что в матрице достижимости  $i$ -тый столбец либо  $j$ -тая строка будет состоять из нулей, то поток из списка потоков не исключается, и в списке переходим к следующему потоку.

В результате отбраковки потоков получаем множество наиболее эффективных потоков  $ZV^3 = \{Z_i, V_j\}$ , таких, что каждый из входов связан транзитным потоком, по крайней мере, с одним выходом. То есть каждый из входов транспортной сети имеет, по крайней мере, один исходящий из него поток, а каждый из выходов сети имеет хотя бы один входящий в него поток. Для множества таких потоков  $ZV^3$  применяется принцип суперпозиции, когда соответствующие им матрицы распределения потоков  $X^{ij} = \|x_{ki}^{ij}\|$  суммируются, образуя матрицу интегрального транзитного потока по выбранному направлению  $\|\Sigma X\|$ . Причём, задача суперпозиции потоков решается таким образом, чтобы в сети могли одновременно существовать все оставшиеся потоки из множества  $ZV^3$ .

С этой целью составляется общее множество дуг сети  $DN = \{<d_{ij}, n>\}$ . Элементами этого множества являются пары  $<d_{ij}, n$ , состоящие из указателя насыщенной дуги  $d_{ij}$  из  $i$ -того узла в  $j$ -тый, и числа  $n$ , показывающего, насколько интегральный поток для этой дуги превышает ее пропускную способность. Из множества насыщенных дуг  $DN$  выбирается элемент, у которого величина  $n$  максимальная. Этот элемент списка  $<d_{ij}, n$  описывает дугу, для которой величина  $n$  суммарного потока, построенного из оставшихся транзитных потоков, больше всего превышает пропускную способность дуги. Поэтому в каждом из транзитных потоков множества  $DN$ , где встречается эта

выбранная дуга, необходимо уменьшить поток в  $nI$  раз для того, чтобы после выполнения суперпозиции оставшихся потоков дуга  $d_{ij}$  оказалась насыщенной. Уменьшение производится для каждого потока из множества  $\{Z_i V_j\}$  по всем путям, которые насыщали рассматриваемую дугу в ходе выполнения алгоритма Форда-Фалкерсона, пропорционально их вкладу в насыщение дуги. После этого множество  $DN$  перестраивается в силу того, что было произведено уменьшение каждого из транзитных потоков, что повлекло изменение величин  $n$  для ветвей сети, которые были задействованы при уменьшении потока.

Описанные действия выполняются до тех пор, пока во множестве  $DN$  находится хотя бы один элемент, у которого  $n$  больше нуля. Как только у всех элементов множества  $DN$  величины  $n$  станут отрицательными либо равными нулю, процесс уменьшения транзитных потоков заканчивается и в качестве решения задачи находится результирующий интегральный поток. При этом потоки распределены таким образом, что после применения к ним принципа суперпозиции величины результирующего суммарного потока на дугах не превышают их пропускных способностей.

В результате выполнения алгоритма суперпозиции потоков в соответствии с теоремой Форда-Фалкерсона величина потока на любом разрезе сети будет максимальной, а сам суммарный поток будет состоять из уменьшенных транзитных потоков.

### 3.1.2 Определение интегрального максимального потока для участка региональной сети

Для иллюстрации работы алгоритма нахождения интегрального потока рассмотрим участок региональной транспортной сети, представленный в виде графа на рисунке 3.1.

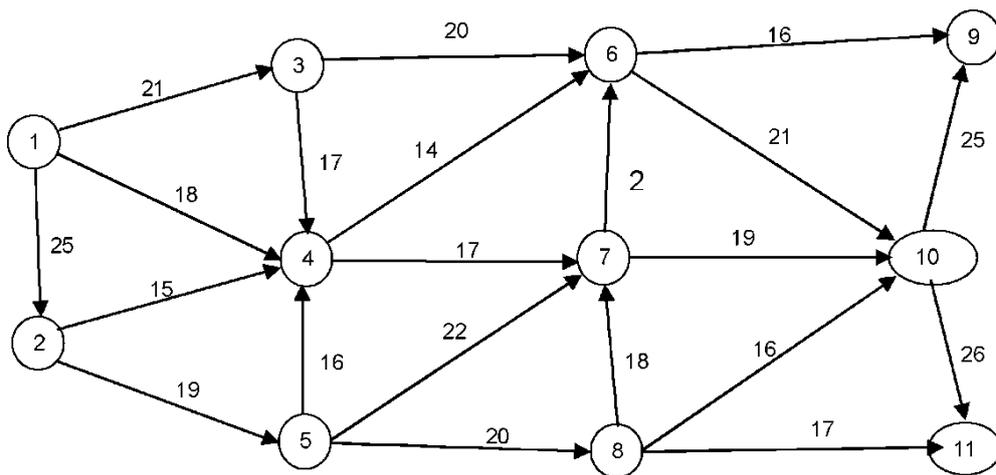


Рисунок 3.1 - Графическое представление участка региональной транспортной сети

Множество входов в сети представлено вершинами 1 и 2. Множество выходов задаётся вершинами 9 и 11. В сети рассматриваются следующие транзитные потоки:  $(1, \dots, 9)$ ,  $(1, \dots, 11)$ ,  $(2, \dots, 9)$  и  $(2, \dots, 11)$ .

Для каждого из указанных транзитов решается задача о максимальном потоке по отдельности, в результате чего получаем четыре матрицы максимальных потоков для этих транзитов  $(\varphi_{1,9}^{\max}, \varphi_{1,11}^{\max}, \varphi_{2,9}^{\max}, \varphi_{2,11}^{\max})$ , соответственно четыре матрицы эффективностей  $(\Phi_{1,9}, \Phi_{1,11}, \Phi_{2,9}, \Phi_{2,11})$ , а также соответствующие распределения величин потоков по ветвям сети  $(X^{1,9}, X^{1,11}, X^{2,9}, X^{2,11})$ .

Для формирования списка наиболее эффективных потоков отбрасываем наименее эффективные потоки таким образом, чтобы не оставить ни один вход без хотя бы одного выходящего потока и ни один выход хотя бы без одного входящего потока. Для рассматриваемого примера наиболее эффективными потоками оказались потоки  $1 \rightarrow 9$  с величиной потока 41 единица,  $1 \rightarrow 11$  с величиной потока 43 единицы и  $2 \rightarrow 9$  с величиной потока 34 единицы.

Суммируя матрицы максимальных потоков  $\|X^{1,9}\|$ ,  $\|X^{1,11}\|$  и  $\|X^{2,9}\|$ , получаем матрицу интегрального транзитного потока  $\|\Sigma X\|$ . Путем поэлементного вычитания матрицы  $\|\Sigma X\|$  из матрицы пропускных способностей сети  $\|C\|$  получаем матрицу  $\|C - \Sigma X\|$ , которая имеет следующий вид:

$$\|C - \Sigma X\| = \begin{pmatrix} 0 & 5 & -19 & -6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3 & -17 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 17 & 0 & -20 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -15 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 16 & 0 & 0 & 3 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -16 & -18 & 0 \\ 0 & 0 & 0 & 0 & 0 & 19 & 0 & 0 & 0 & -11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -18 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Отрицательные значения элементов матрицы показывают на недостаточную пропускную способность соответствующих ветвей сети в случае движения через сеть одновременно всех оставленных потоков. Выбираем ветвь, суммарный поток на которой наиболее превышает её пропускную способность. Для рассматриваемого примера эта ветвь (3,6) - наименьший элемент матрицы  $\|C - \Sigma X\|$ .

Выбираем все пути, которые насыщали ветвь (3,6) в ходе решения задачи о максимальном потоке для каждого из транзитных направлений и величины  $\Delta$ , на которые увеличивался поток по этим путям.

Для транзитного направления  $1 \rightarrow 9$  это будут пути:

- $(1,3) \rightarrow (3,6) \rightarrow (6,9), \Delta = 16;$
- $(1,3) \rightarrow (3,6) \rightarrow (6,10) \rightarrow (10,9), \Delta = 4.$

Для транзитного направления  $1 \rightarrow 11$ :

- $(1,3) \rightarrow (3,6) \rightarrow (6,10) \rightarrow (10,11), \Delta = 20.$

В транзитном направлении  $2 \rightarrow 9$  ни один из путей в ходе решения задачи о максимальном потоке не проходил через вершину  $(3,6)$ . Для того, чтобы суммарный поток смог пройти через ветвь  $(3,6)$ , уменьшаем потоки транзитных направлений по выбранным путям так, чтобы весь суммарный поток уменьшился на 20 единиц, причем по каждому пути поток уменьшается пропорционально величине  $\Delta$ . Для транзитного направления  $1 \rightarrow 9$  поток по пути 1 уменьшаем на 8 единиц, а поток по пути 2 уменьшаем на 2 единицы. Для транзитного направления  $1 \rightarrow 11$  поток по пути 1 уменьшаем на 10 единиц. С этой целью отнимаем от элементов матриц  $\|X^{1-9}\|, \|X^{1-11}\|$ , соответствующих узлам путей величины, на которые уменьшается поток по пути.

Далее пересчитываются значения элементов матрицы  $\|C - \sum X\|$ . В результате получаем новую матрицу, у которой элемент  $(3,6)$  равен 0, так как поток через ветвь сети  $(3,6)$  был уменьшен.

Процесс уменьшения рассматриваемых потоков повторяется до тех пор, пока в матрице  $\|C - \sum X\|$  будут оставаться отрицательные элементы. Когда все элементы матрицы  $\|C - \sum X\|$  окажутся неотрицательными, это будет означать, что пропускных способностей ветвей сети достаточно для того, чтобы все уменьшенные транзитные потоки смогли существовать в сети одновременно. В этом случае алгоритм уменьшения транзитных потоков заканчивается и решением задачи о максимальном потоке будет суммарная матрица  $\|\sum X\|$ .

Для рассматриваемого графа путём последовательных уменьшений была получена следующая матрица  $\|\sum X\|$ :

$$\|\sum X\| = \begin{pmatrix} 0 & 12 & 20 & 17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 17 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 14 & 13 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 & 9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 19 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 23 & 0 & 16 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Величина потока после уменьшения по транзиту  $1 \rightarrow 9$  составила 24 единицы, по транзиту  $1 \rightarrow 11$  – 25 единиц, по транзиту  $2 \rightarrow 9$  – 15 единиц. Суммарная величина трех потоков составила 64 единицы.

Как видно из примера, суммарная величина интегрального максимального потока больше, чем любая величина максимального потока для рассматриваемых потоков, найденная для случая одного потока в сети, что свидетельствует о наиболее полном использовании ресурсов сети.

### **3.2 Исследование влияния одностороннего движения на величину транспортного потока**

Основной показатель загруженности улиц – средняя скорость движения транспортных средств, которая, с годами, резко снижается. Причем эта проблема стала затрагивать не только крупные города, но и более мелкие. Возрастающие скопления транспортных средств на дорожно-транспортной сети городов не только повышают затраты за счет непродуктивных потерь времени, но также увеличивают вероятность происшествий и оказывают негативное влияние на окружающую среду и качество жизни людей.

Управление транспортными потоками дорожно-транспортных сетей тесно связано с их пропускной способностью. Для наиболее эффективной работы сети, необходимо обеспечить возможность прохождения через нее максимального потока.

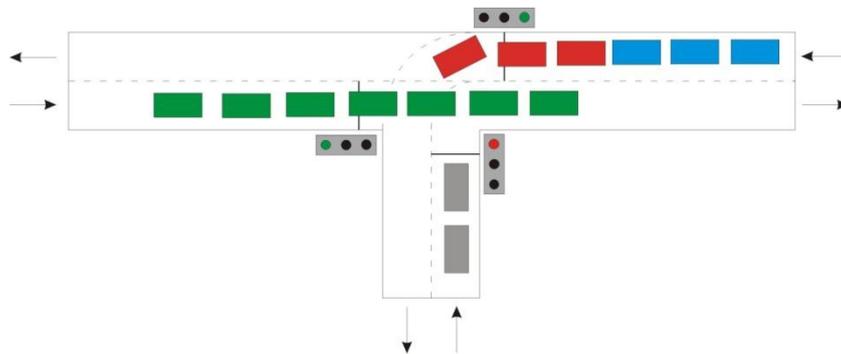
Анализ литературы показал, что для построения моделей дорожно-транспортной сети применяют теорию графов или гиперграфов [20, 21, 51, 52]. Однако у построенных таким образом моделей есть недостатки, например, сильная перегруженность в узлах (перекрестки, развязки и т.п.). При добавлении в модель потоков разного сорта, граф превращается в абсолютно иной объект, с которым, порой, неясно как работать.

Гиперсети [22,24, 25], в отличие от графов, позволяют адекватно описывать системы сетевой структуры послойной иерархии.

Транспортная инфраструктура мегаполиса представляет собой сложную иерархическую нестационарную систему сетевой структуры. Выбранная теория, позволяет близко к реальности моделировать транспортные сети и решать комплексно ряд связанных с ними задач:

- моделирование транспортных потоков дорожно-транспортной сети города;
- управление транспортными системами;
- оптимальное размещение пунктов обслуживания и др.

Рассмотрим участок дорожно-транспортной сети с двусторонним движением (одна полоса движения в одну сторону, другая – в противоположную). В данном случае, на перекрестке, возникает так называемая проблема левого поворота (рисунок 3.2).



Риунок 3.2 - Проблема левого поворота

На рисунке 3.2, красные машины, по правилам дорожного движения, вынуждены ожидать, пока проедут зеленые, и только после этого они могут повернуть налево. Однако синим машинам нужно двигаться прямо, но они ожидают пока проедут красные. Вследствие чего возникает затор и снижается пропускная способность данного участка.

Ввод одностороннего движения (обе полосы движения в одну сторону) может полностью или частично решить данную проблему, разгрузить участок дороги и увеличить пропускную способность, а как следствие и величину потока (рисунки 3.3-3.4).

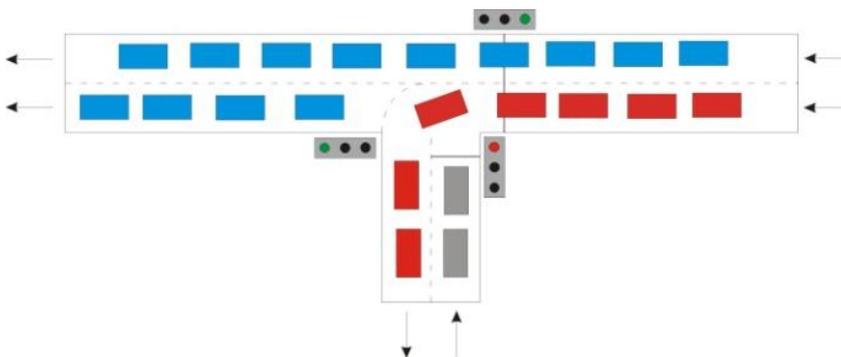


Рисунок 3.3 - Введение одностороннего движения на всей горизонтальной проезжей части

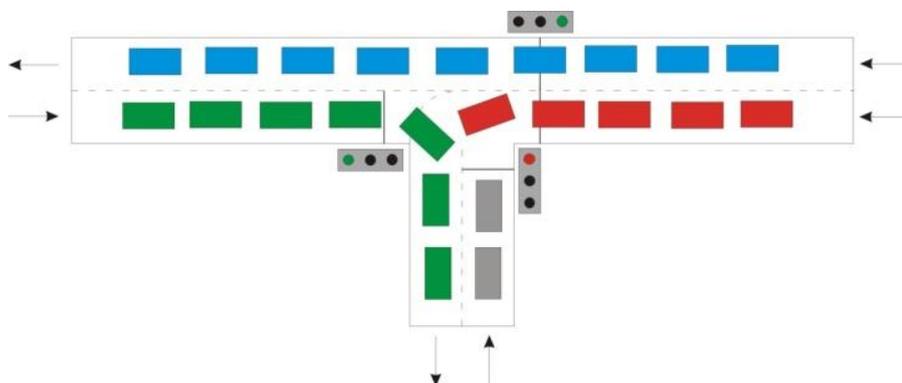


Рисунок 3.4 - Ввод одностороннего движения справа от перекрестка

На рисунке 3.3 видно, что введение одностороннего движения на всей горизонтальной проезжей части полностью решает проблему левого поворота. Ввод одностороннего движения справа от перекрестка (рисунок 3.4) решает проблему частично, позволяет синим машинам беспрепятственно проезжать перекресток, но красным машинам по-прежнему необходимо ждать, пока проедут зеленые.

Здесь следует заметить, что теория нестационарных s-гиперсетей позволяет учесть практически все дорожные коллизии связанные: со структурой сети; потоками машин, зависимиыми от времени суток и/или дней недели; системой управления транспортными потоками (светофоры, инверсные полосы, дорожные знаки как постоянные, так и изменяемыми в зависимости от ситуации на дорогах); разметкой дорог и т.д.

Локусы, это участки сети занимаемые транспортными единицами. Количество локусов, помещающихся на один аллель вычисляется по формуле:

$$S = \frac{L_a}{L_l} \quad (3.5)$$

где  $S$  – количество локусов,  $L_a$  – длина аллеля,  $L_l$  – длина локуса. Однако длина локуса не может быть слишком маленькой, например, если  $L_l$  равна длине автомобиля. При таком раскладе, обеспечить безопасную скорость движения (отличную от нуля) невозможно.

Предполагается, что для безопасного движения по участку дороги, необходимо соблюдать дистанцию между транспортными средствами. Причем, дистанция зависит от скорости автомобиля, за исключением мест обозначенных знаком ее определяющим.

Ориентировочно безопасной дистанцией между транспортными средствами, движущимися в транспортном потоке с одинаковой скоростью, принято считать расстояние в метрах, равное трети величины скорости (при скорости 60 км/ч – безопасная дистанция 20 метров). Следовательно в формуле (3.5) длина локуса  $L_l$  равна  $\frac{v}{3}$  – где  $v$  – скорость транспортного средства. В этом случае получим ситуацию представленную на рисунке 3.6.

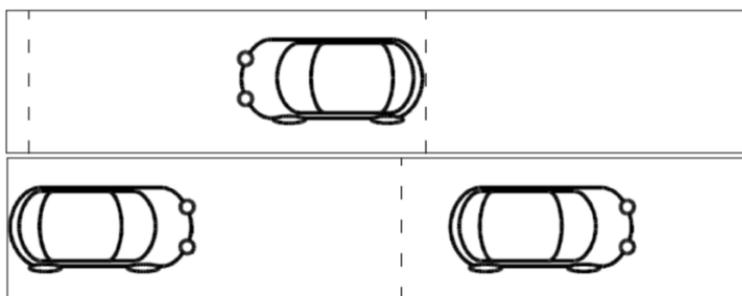


Рисунок 3.6 - Ситуация

Локус активизируется и начинает работать тогда, когда в нем появляется транспортная единица (автомобиль). Вариант его работы зависит от того,

какого типа автомобиль в него попал, в зависимости от этого выбирается «программа» работы. Например: локусы могут работать по расписанию; или случайно: с вероятностью 0.01 – автомобиль поедет назад, 0.02 – остановится, 0.97 – поедет вперед. Это позволяет рассматривать разнотипные потоки и их поведение.

Рассмотрим участок дорожно-транспортной сети изображенный на рисунке 3.7.

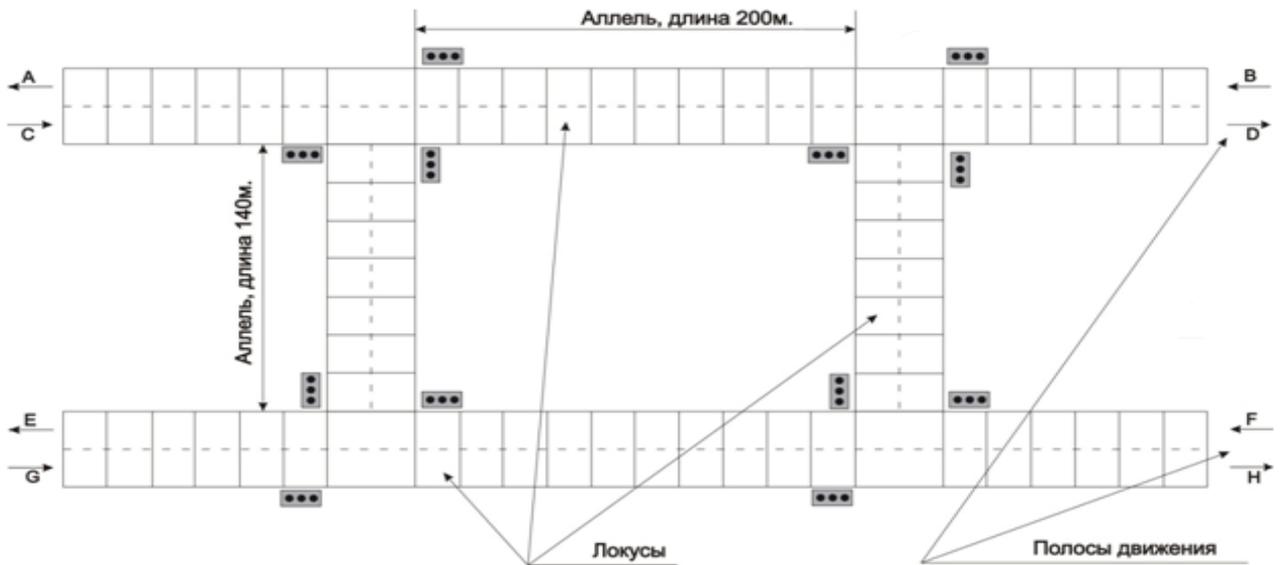


Рисунок 3.7 - Участок дорожно-транспортной сети

Пусть из С едут машины чередуясь: 5 – в D, 5 – в E; из F едут машины чередуясь: 5 – в E, 5 – в D; из В едут в А, но на каждые 7 машин приходится 3, которые едут в H; из G едут в H, но на каждые 7 машин приходится 3, которые едут в А. Скорость машин составляет 60 км/ч. Светофоры работают с периодичностью в 20 секунд (20 секунд горит зеленый, 20 – красный), за это время по одной полосе перекресток успевают преодолеть 15 машин.

Например, надо посчитать пропускную способность данного участка.

Рассмотрим, по тактово, что происходит с потоками на данном участке. На рисунке 3.8. показано состояние в момент времени  $t = 0$  сек, на рисунке 3.9 показано состояние в момент времени  $t = 20$  сек.

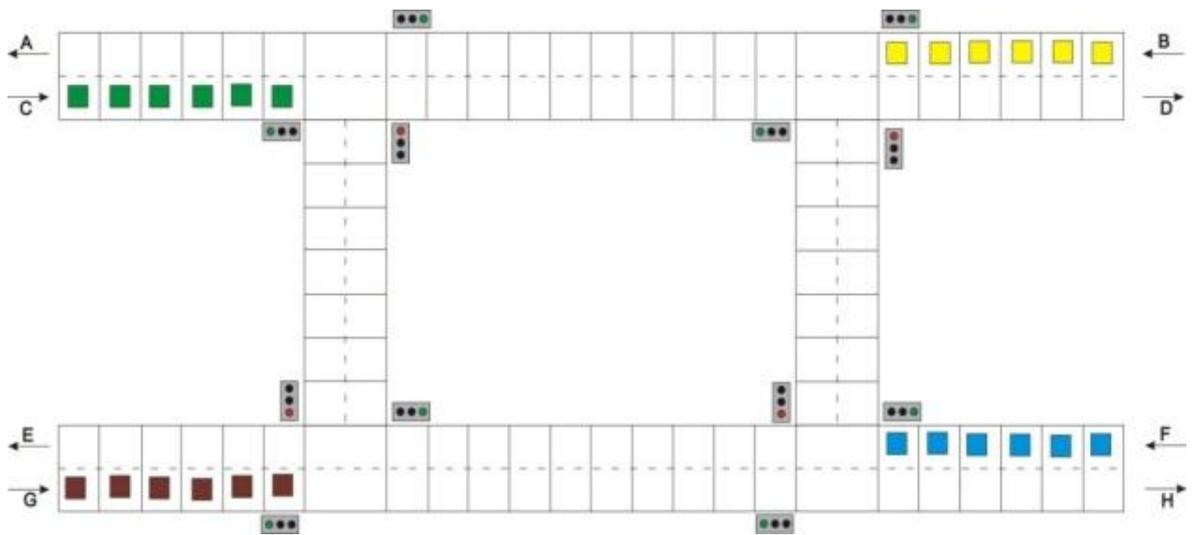


Рисунок 3.8 - Состояние в момент времени  $t = 0$  сек

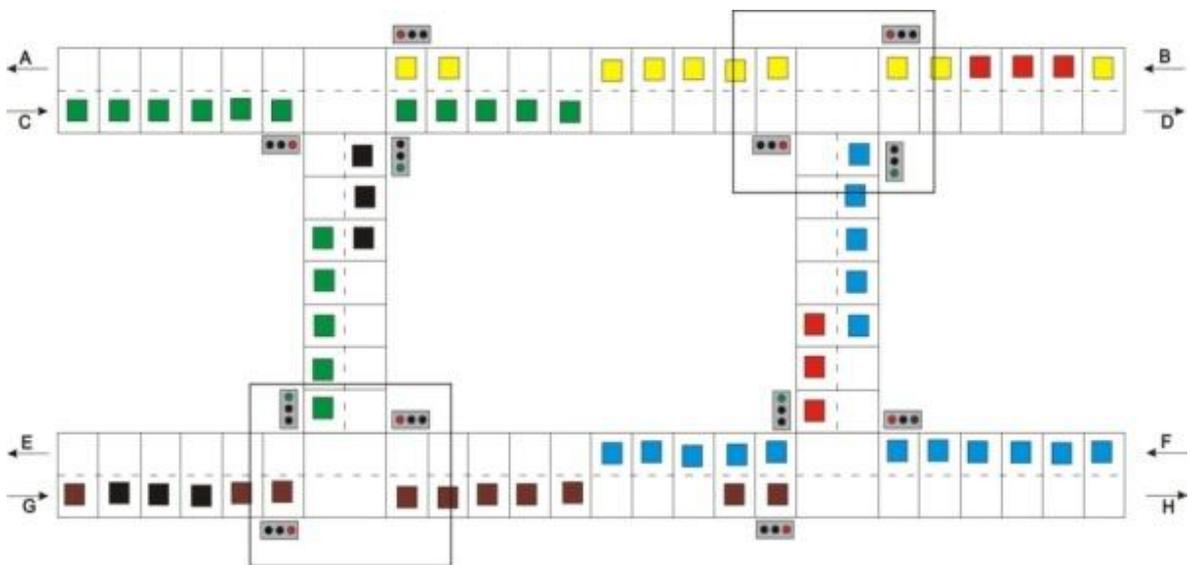


Рисунок 3.9 - Состояние в момент времени  $t = 20$  сек

Так же на рисунке 3.9 выделены два участка, на которых при следующем такте светофора возникает проблема левого поворота, когда красные и черные машины, ожидая поворота, будут мешать проехать желтым и коричневым соответственно.

Моделируя несколько тактов светофора, получаем, что пропускная способность данного участка дорожно-транспортной сети равна  $\sim 2600$  машин в час в одну сторону.

Рассмотрим аналогичную задачу, но теперь не будет расписания сколько машин куда едет. Из C в E машины едут с вероятностью 0.33, остальные едут в D; из F в D машины едут с вероятностью 0.33, остальные едут в E; из B в A машины едут с вероятностью 0.8, остальные едут в H; из G в H машины едут с вероятностью 0.8, остальные едут в A. Скорость машин составляет 60 км/ч. Светофоры работают с периодичностью в 20 секунд (20 секунд горит зеленый,

20 – красный), за это время по одной полосе перекресток успевают преодолеть 15 машин.

Теперь посчитаем пропускную способность данного участка.

Аналогично прошлому случаю, рассмотрим, по тактов, что же происходит на участке. На рисунке 3.10 показано состояние в момент времени  $t = 0$ .

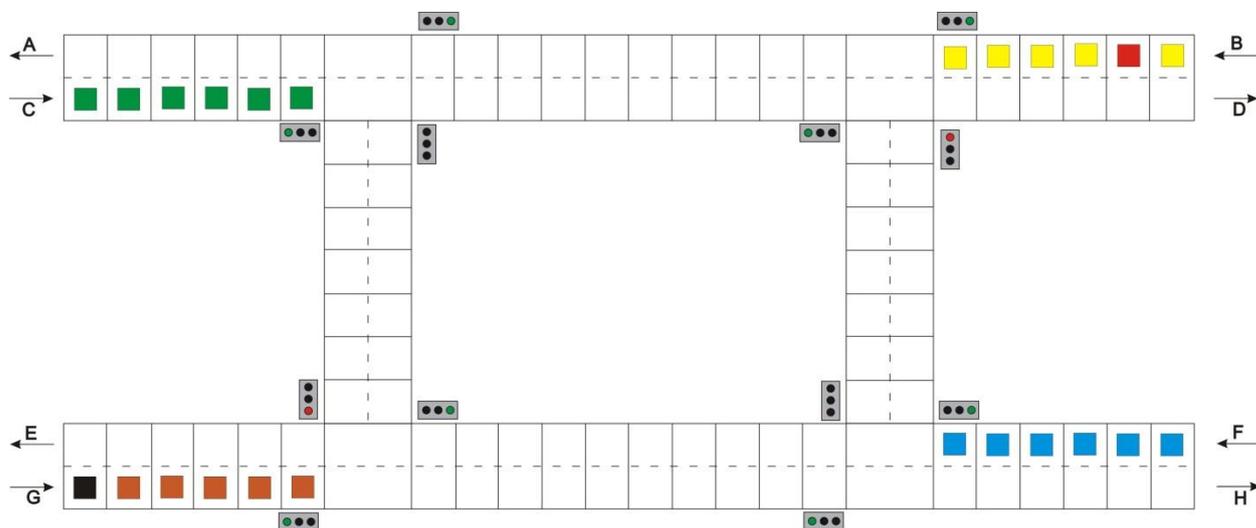


Рисунок 3.10 - Состояние в момент времени  $t = 0$

На рисунке 3.11 показано состояние в момент времени  $t=60$ . Выделенные участки указывают на места возникновения проблемы левого поворота. Так, например, за время с 40 до 60 секунд, по самой верхней полосе перекресток покунило всего 3 машины (1 красная и 2 желтых).

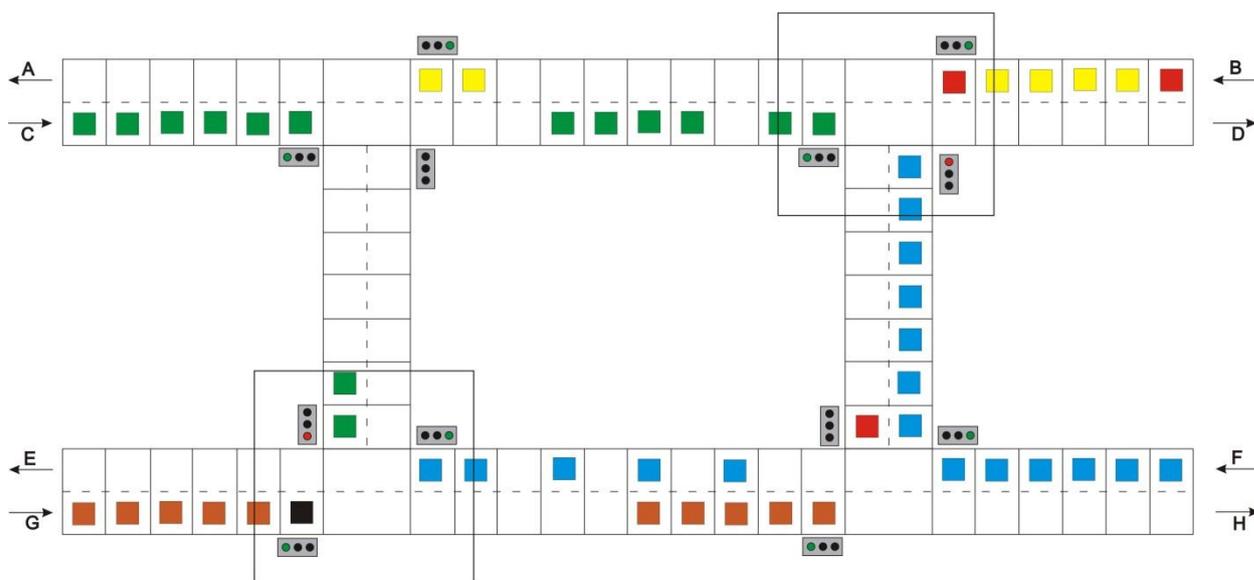


Рисунок 3.11 - Состояние в момент времени  $t=60$

Моделируя несколько тактов светофора, получаем, что пропускная способность данного участка дорожно-транспортной сети равна ~2220 машин в час в одну сторону.

Теперь рассмотрим ситуацию, когда двустороннее движение заменили на одностороннее так, как это показано на рисунке 3.12.

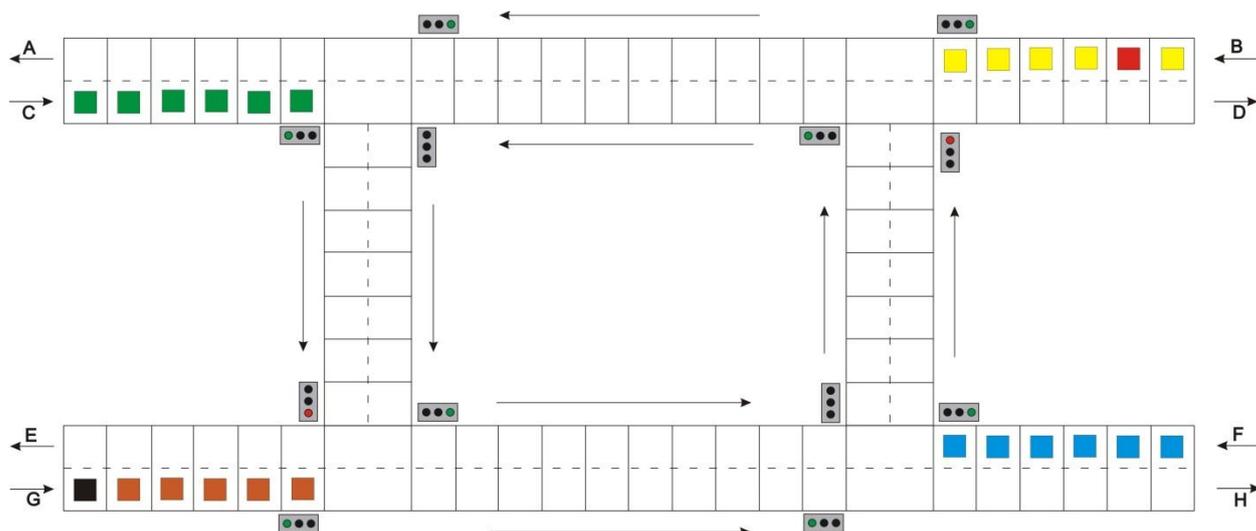


Рисунок 3.12 - Состояние в момент  $t=0$

Исходные данные остаются неизменными, за исключением того, что из B машины едут в E, а не в H; из G машины едут в D, а не в A.

Посчитаем пропускную способность данного участка.

На рисунке 3.13 показано состояние в момент времени  $t=60$ . В данном случае, заторы возникают везде. Происходит это из за того, что слишком большой поток машин поступает на вертикальные участки имеющие небольшую вместимость.

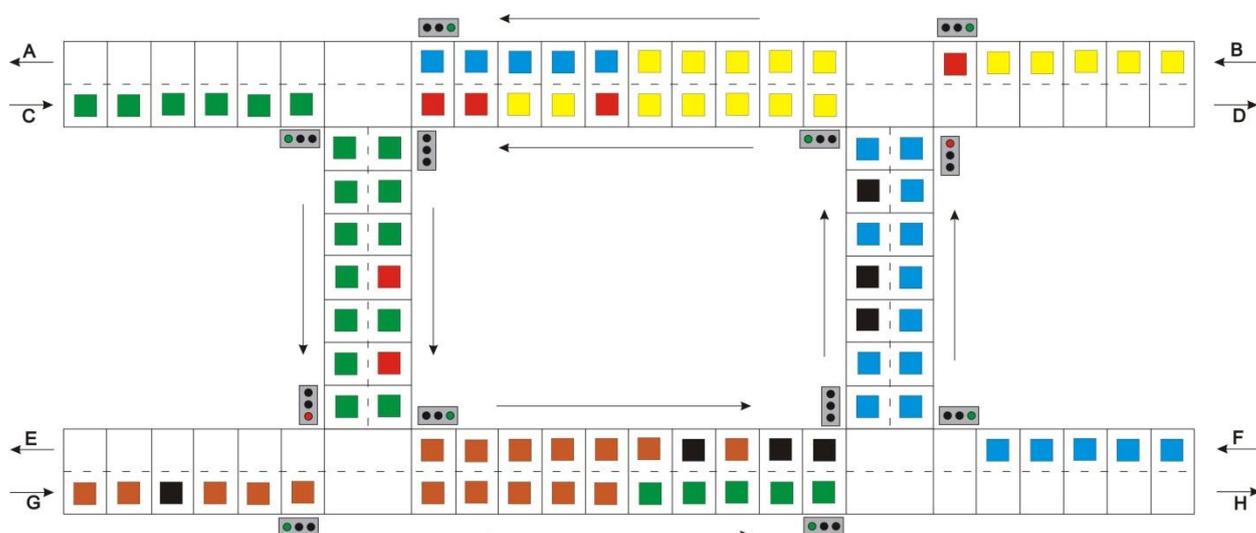


Рисунок 3.13 - Состояние в момент времени  $t = 60$

Моделируя несколько тактов светофора, получаем, что пропускная способность данного участка дорожно-транспортной сети равна ~1860 машин в час в одну сторону.

Теперь рассмотрим ситуацию, когда двустороннее движение заменили на одностороннее, как показано на рисунке 3.14. Тогда условие задачи переписывается следующим образом: машины едут из В в А, но на каждые 15 машин встречаются 5, которые едут в D; машины едут из С в D, но на каждые 15 машин встречаются 5, которые едут в А.

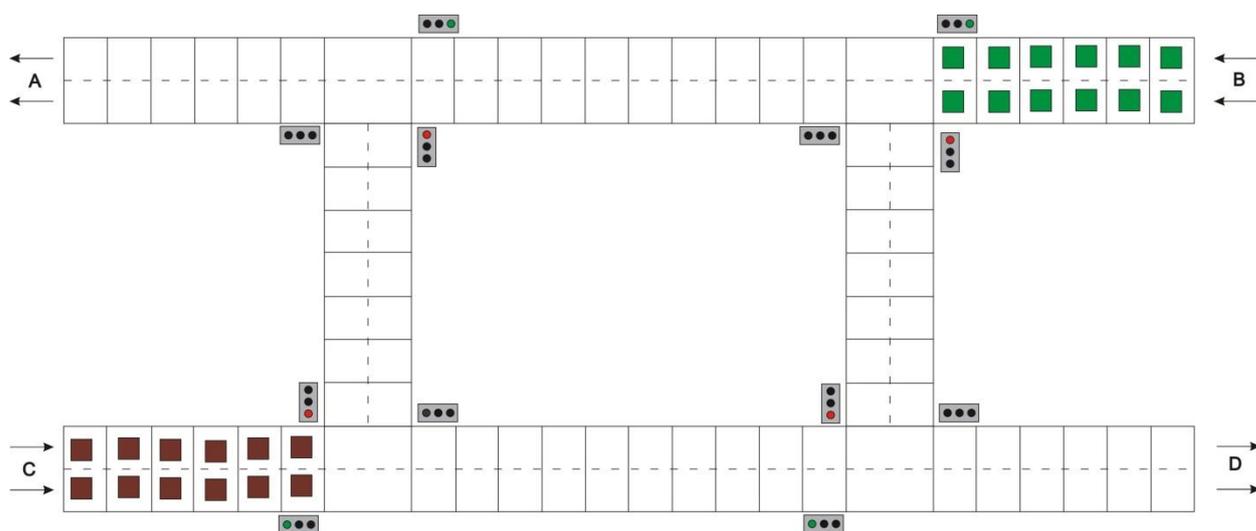


Рисунок 3.14 - Состояние в момент времени  $t = 0$

Моделируя несколько тактов светофора, получаем, что пропускная способность данного участка дорожно-транспортной сети равна ~3600 машин в час в одну сторону.

Ввод одностороннего движения не только решает проблему левого поворота, но может как увеличить так и уменьшить пропускную способность участка дорожно-транспортной сети.

Также, стоит отметить, что при построении модели стоит учитывать не только градостроительные факторы дорожно-транспортной сети, поведение потоков и характеристики видов транспортных средств их составляющих. Разработана концепция имитационной модели транспортных потоков и управляющих транспортных систем. В рамках имитационной модели проведено моделирование потоков на участке дорожно-транспортной сети [66] и выявлено, в целом, положительное влияние внедрения одностороннего движения.

### 3.3 Задача устранения пробок на маршруте при управлении транспортными потоками на сети городских дорог

С каждым годом машин становится всё больше и больше. Соответственно, растёт и число пробок. Эта проблема приобретает характер эпидемии – все большие города мира сталкиваются с ней в той или иной степени. Потрясают цифры статистики – к примеру, в Алматы считается нормальным, если время

проезда от работы до дома составляет порядка полутора-двух часов. Это только в одну сторону. Несложно посчитать, сколько времени уходит в неделю/месяц/год во время простоя. Это потеря времени, как для отдельного человека, так и для экономики страны в целом. Кроме этого, можно отметить, и «пустую» трату бензина, а значит, и лишнее загрязнение окружающей среды.

Управление дорожным движением весьма нетривиальная задача. С одной стороны, управление транспортным потоком, состоящим из технических средств (автомобилей), с другой – управлять приходится живыми людьми, каждый из которых преследует свои цели.

Полное решение проблемы возникновения пробок может быть только комплексным, учитывающим не только непосредственное регулирование на перекрёстках, но нечто большее. Для наглядных примеров обратимся к зарубежному опыту [53, с.14]:

- Можно реализовать поочерёдное движение, в зависимости от каких-то свойств автомобилей. Например, во многих странах Евросоюза (таких, как Греция) реализован такой механизм: по чётным дням разрешён проезд автомобилей с чётными номерами, и по нечётным с нечётными номерами. Сокращение числа автомобилей налицо.

- Создание в городе зон с платным проездом. Для примера тут можно привести Лондон, где уже около 5 лет действует тарификация машин для проезда через центр города, предоставляя, между тем, льготы инвалидам и такси. Применительно к Казахстанским реалиям – для некоторых крупных городов (таких, как Алматы и Астана), её ввести в принципе можно. И, хотя по самым оптимистичным прогнозам, это снизит транспортный поток не более, чем на треть, это может стать дополнительным источником финансирования дорожного строительства – новых развязок, ремонта дорог и т.д.

- По возможности, убрать «пешеходные» светофоры. В Казахстане это отчасти уже происходит – им на смену приходят подземные/надземные переходы (чаще вторые).

- Решение проблемы с парковками – нередко можно наблюдать такую картину – машины припаркованы у обочины (а зачастую, и не в один ряд), что конечно же создаёт проблемы движению.

- Применительно к нашим реалиям, стоит добавить и обязательное повышение культуры вождения, без которой, это так и останется «рекомендациями», которые большинство будут игнорировать.

Пусть  $V$  непустое множество вершин, и  $E \subseteq V \times V$  – множество рёбер, с условием симметричности: пара  $(u, w) \in E$  тогда и только тогда, когда  $(w, u) \in E$ . Тогда совокупность  $G=(V, E)$  называется *неориентированным графом* [25, с.182].

Орграф – неориентированный граф без условия симметричности. Вершины графа обычно называют *узлами*, а дуги – *ориентированными рёбрами*.

*Мультиграф* – граф, в котором допускается более одного ребра между вершинами.

*Путь* в орграфе называется последовательность вершин  $v_1, \dots, v_n(*)$  для которых существуют дуги  $v_i \rightarrow v_{i+1}$ . Длиной пути (\*) является  $n-1$ . Путь называется простым, если в нём все вершины, за исключением, быть может, начальной и конечной, различны. *Цикл* – простой путь длины больше 1. *Дерево* – это ациклический граф. Существует несколько способов обхода всех узлов дерева (обход дерева равнозначен упорядочиванию по какому-либо правилу узлов). Выделим наиболее часто встречающиеся [24, с.9]:

Если дерево  $T$  – пустое, то в список обхода заносится пустая запись

Если дерево  $T$  состоит из одного узла, то в список обхода заносится эта вершина.

Далее, пусть  $T$  дерево с корнем  $n$  и под деревьями  $T_1, \dots, T_k$

1. При *прямом обходе* вначале посещается корень  $n$ , затем все узлы поддерева  $T_1$ , затем все узлы поддерева  $T_2$  и т.д. Последними посещаются узлы поддерева  $T_k$

2. При *симметричном обходе* вначале посещаются в симметричном порядке все узлы поддерева  $T_1$ , затем корень  $n$ , затем последовательно в симметричном порядке все узлы поддеревьев  $T_2, \dots, T_k$

3. Во время *обхода в обратном порядке* сначала посещаются в обратном порядке все узлы поддерева  $T_1$ , затем  $T_2, \dots, T_k$ , последним посещается корень  $n$ .

Например, в городе, для выделенного маршрута  $S-T$  ( $S=X_1, \dots, X_k=T$ ), требуется устранить все пробки путём изменения режимов работы светофоров, не создавая дополнительных пробок на боковых, к этому маршруту, дорогах.

Остановимся на формулировке чуть подробнее. Во-первых, под пробкой мы будем понимать превышение входящего потока на перекрёстке над его пропускной способностью. Кроме того, обратим внимание на вторую часть постановки – о «не создании дополнительных пробок на боковых направлениях», без неё задача решалась бы тривиально – достаточно было бы установить зелёный свет на всех перекрёстках маршрута  $S-T$ .

Входные данные:

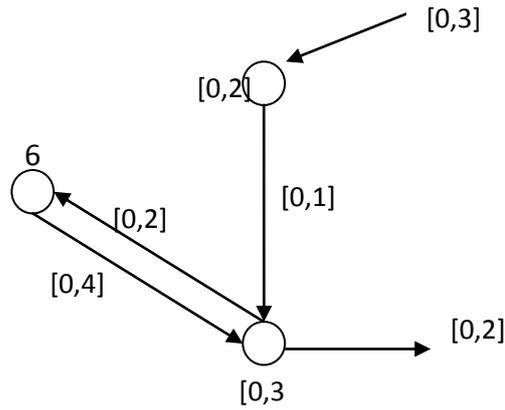
- а) город со светофорами
- б)  $S-T$  маршрут
- в)  $PR$  – множество пробок маршрута
- г)  $K$  – глубина анализа боковых дорог
- д)  $I$  – число итераций алгоритма
- е)  $T$  – максимальное время ожидания на перекрёстке

Распишем входные данные поподробнее.

Город представляет собой ориентированный мультиграф  $M(U, V)$ , где в качестве вершин  $U$  мы рассматриваем перекрёстки, а в качестве рёбер  $V$  – возможные пути проезда от одного перекрёстка к другому. Т.е. ребро  $(X_i; X_j) \in M$  тогда и только тогда, когда разрешено движение от перекрёстка  $X_i$  к перекрёстку  $X_j$ .

Для каждого ребра задаются веса:

$M_{X_j, X_{j+1}}$  соответствует максимальной (физической) пропускной способности улицы за единицу времени  $t$  в направлении от  $X_j$  к  $X_{j+1}$



Пример части города из трёх перекрёстков ( $X_1, X_2, X_3$ ), с заданными максимальными пропускными способностями улиц.

$R_{X_j, X_{j+1}}$  соответствует реальным данным о количестве машин за единицу времени  $t$  в направлении от  $X_j$  к  $X_{j+1}$

Перейдём к перекрёсткам.

Рассмотрим общий случай пересечения двух дорог (рисунок 3.15).

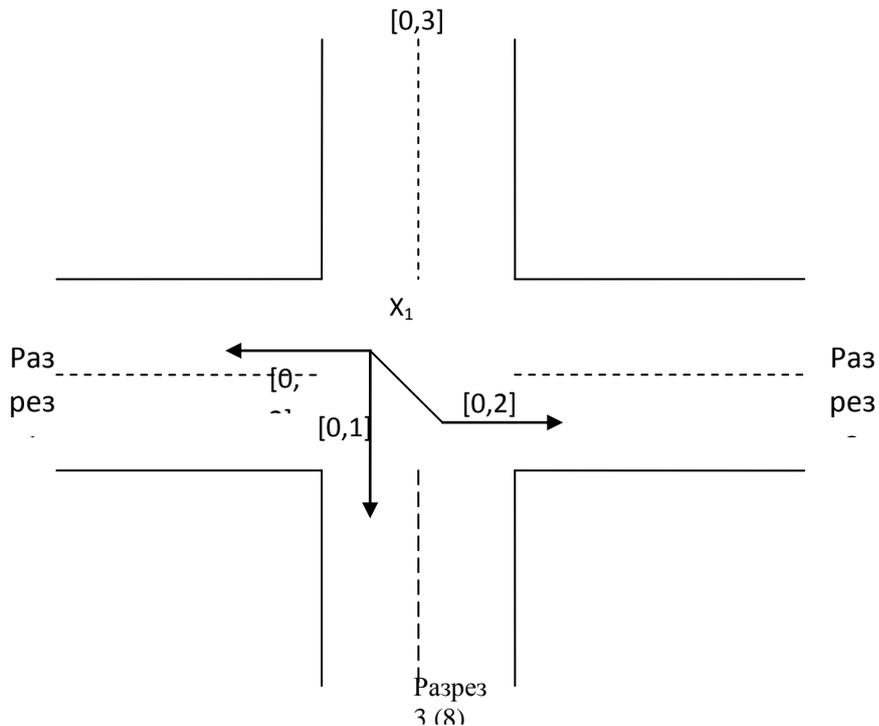


Рисунок 3.15 - Общий случай пересечения двух дорог

Для примера рассмотрим какое-то одно направление. Пусть на перекрёстке можно двигаться в направлениях, отмеченных на рисунке I, II и III. Обозначим потоки, проходящие перекрёсток за время горения зелёного за  $P_i$ . Тогда время горения зелёного и красного для каждого из направлений обозначим  $G_i$  и  $R_i$  единиц времени  $t$  соответственно. Отсюда, вычислим «среднюю пропускную способность» перекрёстка:

$$S_i = \left[ \frac{P_i}{G_i + R_i} \right], \text{ для } i = I, II, III$$

Для совместимости с обозначениями максимальной и реальной пропускной способности улиц, примем следующее обозначение для перекрёстков:

$J_{X_j, X_n}^{X_{j+1}}$ , где  $X_{j+1}$  – текущий перекрёсток,  $X_j$  – «предшествующий», а  $X_n$  – «последующий» перекрёстки, и  $J$  литера, обозначающая максимальную ( $M$ ), либо реальную ( $R$ ) пропускную способность. Теперь, переходя от перекрёстка к графу, получаем такую картинку 3.16.

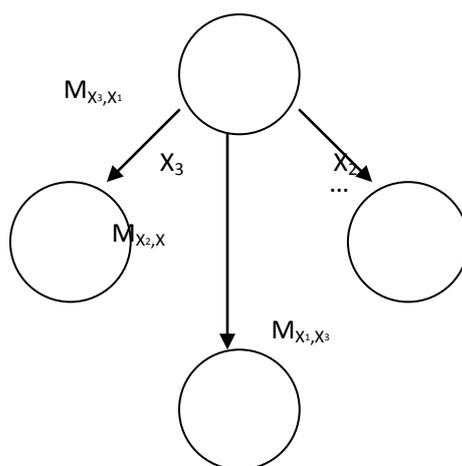


Рисунок 3.16 - Граф

Аналогично определяем максимальную (физическую) пропускную способность направлений.

Остальные входные данные:

- S-T маршрут – последовательность вершин  $X_1 \dots X_k$ , где существует путь длины 1 из  $X_i$  в  $X_{i+1}$

- PR – подмножество вершин маршрута, на перекрёстках которых есть пробки

- Глубина анализа  $K$  – высота дерева боковых дорог (потребуется для алгоритма)

- I – Число итераций алгоритма

### Алгоритм

I. Начальные данные алгоритма:

-  $J$  – «внутренний» итератор, хранящий текущую вершину, в которой мы сейчас находимся, первоначально  $J = X_{k-1}$ . Условимся:

$J+1 = X_{i+1}$ , если  $J = X_i$  при  $k-1 \geq i \geq 1$

$J-1 = X_{i-1}$ , если  $J = X_i$ , при  $k-1 \geq i \geq 1$

- COUNT – текущее число пробок. Первоначально  $COUNT = |PR|$

- ITERATOR – «глобальный» итератор, хранящий номер текущей итерации алгоритма, первоначально равен 1.

-  $C$  – величина текущей разницы потока, нужна для передачи изменений величины потока между соседними вершинами, первоначально равна 0

II. Ход алгоритма:

*Случай 1.*  $J = X_{k-1}$  Т.е. мы делаем первый шаг нашего алгоритма.

Шаг 1. Уменьшаем боковые для нашего маршрута потоки на этом перекрёстке

Шаг 2. Пересчитываем режим работы светофора таким образом, чтобы выходной поток был наибольшим, не создавая при этом пробок на боковых дорогах, и он не превосходил  $M_{J, J+1}$ .

Обозначим полученные боковые потоки  $P_I$  и  $P_{II}$ , а максимальную пропускную способность  $M_I$  и  $M_{II}$  соответственно. Тогда  $M_{\max} = \max\{M_I, M_{II}\}$ ,  $P_{\max} = \max\{P_I, P_{II}\}$ . Теперь высчитаем новый режим светофора (для боковых дорог):

$$G = \frac{P_{\max} T}{M_{\max} - P_{\max}}$$

$$R = T$$

В случае  $M_{\max} = P_{\max}$ , пропускаем этот перекрёсток (т.е. не меняем режим работы). Кроме этого, может получиться, что  $G > T$ , в этом случае:

$$G = \frac{T(M_{\max} - P_{\max})}{P_{\max}}$$

$$R = G,$$

Где  $R$  – время горения красного, а  $G$  – зелёного света.

Шаг 3. В  $C$  заносим разницу между реальным входящим потоком и тем потоком, который теперь может принять перекрёсток, не создавая пробки,  $C \leq 0$ . Заметим, что если в вершине  $J$  была пробка, то мы от неё избавились, т.е. в этом случае  $COUNT = COUNT - 1$

Шаг 4. Переходим к следующей вершине ( $J = J - 1$ )

*Случай 2.* Теперь пусть мы стоим в  $J = X_i$  и  $J$  не совпадает с вершиной  $X_{k-1}$

Шаг 1. Уменьшаем боковые потоки

Шаг 2. Изменяем режим работы светофора таким образом, чтобы выходной поток был  $N_p = \min\{M_{J, J+1}; R_{J, J+1} + C\}$ . Для этого:

Пересчитываем светофор с помощью формул из предыдущего случая,

$$\text{Далее } R_{zl}^{cv} = \frac{R_{J-1, J} - N_p T'}{N_p}, \text{ где } T' = \min\left\{T, \frac{T(M_{\max} - P_{\max})}{P_{\max}}, \frac{R_{J-1, J}}{2N_p}\right\}$$

$$G_{zl}^{cv} = T'$$

Шаг 3. Далее, опять вычисляем разность между реальным входящим потоком и получившимся исходящим и заносим в  $C$ . При этом, если в текущей вершине была пробка, то уменьшаем  $COUNT$  на 1

Примечание. В том случае, если перекрёсток нерегулируемый, то пропускаем шаг изменения режима работы светофора и переходим к следующей вершине.

III Минимизация боковых потоков

Пусть мы стоим в вершине  $J$ . В общем случае, наш поток формируется из трёх потоков, обозначенных как I, II, III (рисунок 3.17).

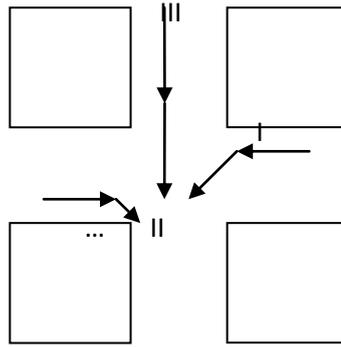


Рисунок 3.17 - Потоки

Из них к боковым потокам относятся только I и II. Эти подслучаи одинаковые, поэтому рассмотрим только один, I. Заметим, что как поток, I так же формируется (в общем случае) из трёх: 1, 2, 3, и т.д. Будем строить дерево из этих потоков таким образом: в роли корня выступает поток I. Припишем этой вершине вес, равный величине этого потока.

Теперь пусть у нас есть дерево T, если его высота меньше K, то для каждого листа этого дерева зададим потомков, формирующих каждый лист, как потоки. Будем продолжать до тех пор, пока высота дерева не станет равной K.

Обозначим это дерево  $Tree'_j$ .

Теперь наша цель – обойти дерево, уменьшая веса узлов, для того, чтобы уменьшить суммарное воздействие бокового потока (в нашем случае, I) на маршрут S-T. Но заметим, что отдельные узлы дерева могут участвовать и при анализе соседних перекрёстков. Кроме того, веса какой-то части вершин мы поменять не сможем, так как они уже были пересчитаны предыдущими итерациями алгоритма. Для того, чтобы установить, какие вершины подлежат изменению, а какие нет, зададим раскраску дерева по следующим правилам:

Для начала зададим цвета – Красный, Чёрный и Синий. Красный цвет будет означать, что вершина подлежит пересчёту при обходе, Чёрный, что не подлежит, а Синий – что вершина уже была пересчитана предыдущими итерациями (рисунок 3.18).

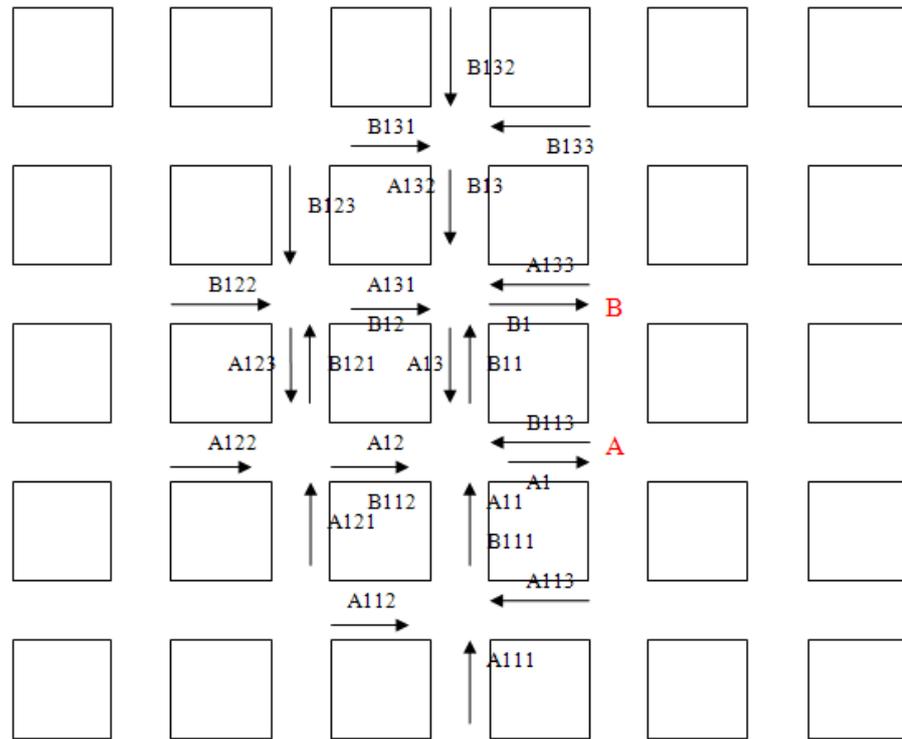


Рисунок 3.18 - Начальный вариант

Начнём раскраску дерева с корня. Корень всегда будет красным. Сделаем замечание – чем больше путь от корня к вершине, тем «дальше» она находится от анализируемого перекрёстка. Уместно заметить, что в общем случае, это означает, что такая вершина участвует в формировании бокового потока в меньшей степени, чем вершина того же дерева с меньшим путём. Рассмотрим два соседних перекрёстка. Из рисунка видно, что часть элементов принадлежат сразу дереву А и дереву В (как например, A131/B12), поэтому каждая из этих вершин будет пересчитана как минимум два раза. Учитывая замечание, отсюда можно вывести ещё одно правило: при совпадении вершин деревьев соседних перекрёстков, вершина с меньшим номером красится в красный цвет, а с большим – в синий.

Кроме того, будем красить вершину в чёрный цвет, если она попадает на уже пересчитанные значения нашего маршрута. Исходя из этого, заметим, что если в дереве Т, есть поддерево с Синим/Чёрным корнем, то и всё поддерево должно быть покрашено в чёрный цвет (рисунок 3.19).

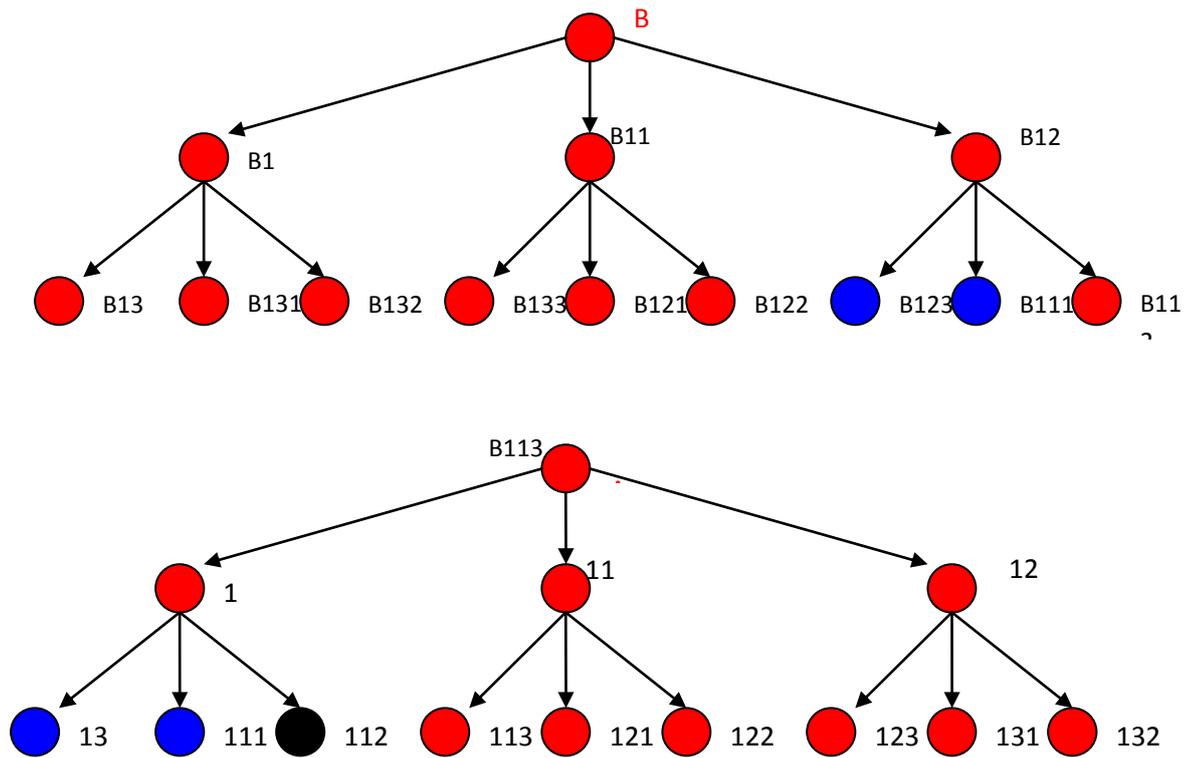


Рисунок 3.19 - Пример раскраски деревьев для двух соседних перекрёстков при  $k=3$

Далее, зададим функцию  $f$ , которая для каждого перекрёстка выдаёт допустимое изменение потока –  $f(g, p, \text{ITERATOR})$ , где  $g$  – текущее расстояние от нашего перекрёстка,  $p$  – поток на этом перекрёстке и  $\text{ITERATOR}$  – номер итерации алгоритма.

Примечание – в общем случае, эта функция эмпирическая, так как ситуация на каждом перекрёстке разная и заранее указать какой конкретно она будет иметь вид практически невозможно. Более того, уместно это функцию рассматривать как своеобразного Оракула. Остановимся чуть подробнее. Под Оракулом понимается некий внешний механизм воздействия на ситуацию – им, к примеру, может быть Суперкомпьютер, который высчитывает данные на основе статистики. Далее, совершая обратный обход дерева, применяем к вершинам функцию  $f$ .

#### IV. Условия завершения

Одна итерация нашего алгоритма может завершиться по двум причинам:

1. В какой-то момент времени  $C \leq 0$  и  $\text{COUNT} = 0$ . В этом случае можно сказать, что наш алгоритм справился с задачей, мы возвращаем обработанный маршрут

2.  $C > 0$  И мы находимся в вершине  $S$ , т.е. исправить ситуацию полностью нам не удалось. В этом случае мы увеличиваем  $\text{ITERATOR}$  на 1 и запускаем следующую итерацию, ужесточая условия. Здесь опять же возможны два варианта –

а) либо мы попадём в случай 1 (т.е. решим задачу) после какого-то числа итераций,

б) либо в какой-то момент времени мы превысим входные данные на число итераций. В этом случае мы возвращаем, что ситуация не разрешима в нашей модели.

Результаты показаны в виде таблицы (таблица 3.1).

Таблица 3.1 - Таблица результатов

Число итераций	Вид эмпирической функции	Глубина анализа	Результат
2	F=числу итераций	2	Решение найдено
1	F=числу итераций	3	Решение найдено
5	F=1	2	Решение не найдено
1	F=1	3	Решение найдено
1	F=высоте поддерева для каждой вершины	2	Решение найдено
1	F=высоте поддерева для каждой вершины	3	Решение найдено

### 3.4 Задача маршрутизации транспортными потоками города с применением теории S-гиперсети

Задача маршрутизации транспорта на городских транспортных сетях очень похожа на маршрутизацию компьютерного трафика в сетях информационных. Главные отличия состоят в том, что в первом случае в качестве пакета рассматривается транспортное средство, а также существуют правила дорожного движения, ограничивающие передвижение таких пакетов. Таким образом, задача маршрутизации транспорта состоит все в той же проблеме нахождения кратчайшего пути между двумя узлами [54-57].

Маршрутизация может быть представлена в следующем виде. Пусть дан направленный взвешенный граф  $G=(V,E)$ , в котором каждый узел из множества  $V$  представляет собой устройство, обрабатывающее и передающее данные, а каждое ребро из множества  $E$  является линией связи. Основной задачей алгоритмов маршрутизации является передача данных из узла источника в узел приемник, максимизируя при этом производительность сети. Здесь подразумевается, например, передача максимального числа пакетов за минимальное время.

Алгоритмы маршрутизации должны выполнять следующие функции [58]:

- сбор, организация и распределение информации о созданном пользователем трафике и состояниях сети;
- использование собранной информации для создания подходящих маршрутов, максимизирующих производительность объектов;
- направление трафика пользователя по выбранному маршруту.

Способ реализации описанных трех функции сильно зависит от технологии передачи и коммутации пакетов, положенной в основу сети, и от

особенностей других взаимодействующих уровней приложений. Отправка трафика пользователя может происходить с использованием двух базовых операций сети: коммутация каналов и коммутация пакетов (которые также связанные с понятиями ориентированный и неориентированный на соединение). При коммутации каналов на стадии установки соединения ищутся и резервируются ресурсы сети, которые впоследствии будут предоставлены каждой новой сессии. В этом случае все пакеты данных, принадлежащие одной и той же сессии, будут направлены по одному и тому же пути. От маршрутизаторов требуется хранение информации об активной сессии. При коммутации пакетов нет стадии резервирования, информация о состоянии не хранится на маршрутизаторах, и пакеты данных могут отправляться по разным путям. В каждом промежуточном узле принимается самостоятельное решение о выборе выходной линии, по которой будет отправлен пакет данных в узел приемник.

Общим параметром для всех видов алгоритмов маршрутизации является таблица маршрутизации. Таблица маршрутизации располагается в каждом узле сети, и содержит всю информацию о ней. Эта информация, в свою очередь используется маршрутизаторами для создания маршрутов отправки пакетов данных. Тип информации, содержащейся в маршрутных таблицах, зависит исключительно от алгоритма маршрутизации.

Алгоритмы маршрутизации можно классифицировать следующим образом [59-62]:

- Централизованные и распределенные;
- Статические и адаптивные.

В централизованных алгоритмах главное управляющее устройство отвечает за обновление таблиц маршрутизации всех узлов и/или принимает каждое решение о маршрутизации. Централизованные алгоритмы могут быть использованы только в частных случаях и для малых сетей. В общем, задержки необходимые для сбора информации о состоянии сети и для трансляции запроса на обновление данных делает такие алгоритмы неприменимыми на практике. Более того, централизованные системы не являются отказоустойчивыми. В статических системах маршрутизации путь, который проходит пакет, определяется только на основе его источника и приемника, без рассмотрения текущего состояния сети. Этот путь обычно выбирается как кратчайший относительно выбранного стоимостного критерия, и может быть изменен только за счет поврежденных линий связи или узлов.

Адаптивные маршруты, в принципе, более привлекательны, так как они могут адаптировать способ маршрутизации к временным и к пространственным изменениям трафика. Как недостаток такого подхода выделяют то, что слишком частые изменения в сети могут стать причиной колебаний в выбранных путях. Это обстоятельство, в свою очередь, может привести к созданию циклических путей, а также к большим отклонениям в выполнении алгоритма. К тому же адаптивная маршрутизация может привести к противоречивым ситуациям, которые могут возникнуть при выходе из строя узлов, линий связи или при изменении локальной топологии. Однако, все эти

проблемы устойчивости более характерны для сетей неориентированных на соединение.

Другой способ классификации алгоритмов маршрутизации имеет вид [59-62]:

- Минимальная маршрутизация и неминимальная маршрутизация;
- Оптимальная маршрутизация и маршрутизация, определяющая кратчайший путь.

Минимальные маршруты позволяют пакетам выбирать только пути с минимальной стоимостью, в то время как неминимальные алгоритмы позволяют делать выбор между всеми доступными путями, используя при этом некоторые эвристические стратегии.

Оптимальная маршрутизация имеет доступ ко всей сети и ко всем ее объектам для оптимизации функции всех отдельных потоков линий связи (обычно эта функция является суммой стоимостей путей, назначенных в соответствии со средними задержками пакетов).

Маршрутизация, определяющая кратчайшие пути, объективно определяет кратчайший путь (минимальную стоимость) между двумя узлами. Учитывая содержание в таблицах маршрутизации различной информации, алгоритмы нахождения кратчайших путей могут быть поделены на два класса: дистанционно-векторные и алгоритмы состояния связи.

Оптимальная маршрутизация является статической и требует знания всех характеристик трафика. Алгоритмы нахождения кратчайших путей более гибкие, они не требуют априорных знаний о моделях трафика, и, на данный момент, наиболее распространены среди алгоритмов маршрутизации

### 3.4.1 Описание и анализ методов маршрутизации

Прежде чем приступить к рассмотрению конкретных алгоритмов маршрутизации, сформулируем принцип оптимальности. Этот принцип утверждает, что если узел коммутации (УК) J находится на оптимальном пути между УК I и K, то оптимальный маршрут между J и K принадлежит этому оптимальному пути. Это так, поскольку существование между J и K оптимального маршрута отличного от части маршрута между I и K противоречило бы утверждению об оптимальности маршрута между I и K. Дело в том, что если рассмотреть (Рисунок 3.4.1) маршрут от I до K как от I до J (назовем его S1) и от J до K (назовем его S2), то если между J и K есть маршрут лучше, чем S2, например S3, то маршрут S1S2 не может быть лучшим. Взяв конкатенацию маршрутов S1S3, мы получим лучший маршрут, чем маршрут S1S2.

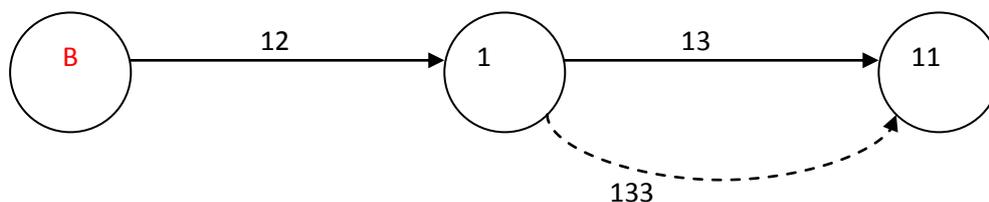


Рисунок 3.4.1 - Маршрут

Следствием из принципа оптимальности является утверждение, что все маршруты к заданной точке сети образуют дерево с корнем в этой точке. Это дерево называется деревом захода.

Поскольку дерево захода - это дерево, то там нет циклов так, что каждый пакет будет доставлен за конечное число скачков. На практике все может оказаться сложнее. Маршрутизаторы могут выходить из строя и наоборот появляться новые, каналы могут выходить из строя, разные маршрутизаторы могут узнавать об этих изменениях в разное время и т.д. и т.п.

#### *Маршрутизация по наикратчайшему пути*

Идея этого алгоритма состоит в построении графа транспортной среды, где вершины - маршрутизаторы, а дуги - линии связи. Алгоритм находит для любой пары маршрутизаторов, а точнее абонентов, подключенных к этим маршрутизаторам, наикратчайший маршрут в этом графе.

Расстояние можно измерять в скачках, а можно в километрах. Возможны и другие меры. Например, дуги графа могут быть размечены весами, величина которых равна средней задержке для пакетов в соответствующем канале. В графе с такой разметкой наикратчайший путь - наиболе быстрая путь, хотя он не обязательно имеет минимальное число скачков или километров. В общем случае веса на дугах могут быть функциями от расстояния, пропускной способности канала, среднего трафика, стоимости передачи, средней длины очереди в буфере и других факторов. Изменяя весовую функцию, алгоритм будет вычислять наикратчайший путь в смысле разных мер.

Известно несколько алгоритмов вычисления наикратчайшего пути в графе. Один из них предложил Дейкстра (Dijkstra). Идею этого алгоритма можно описать так. Все вершины в графе помечаются расстоянием (оно указано в скобках) до исходной вершины вдоль наилучшего, известного пути. Изначально никаких путей не известно и все вершины помечены бесконечностью. По мере работы алгоритма и нахождения путей, метки могут меняться. Метки могут быть двух видов либо пробными, либо постоянными. Изначально все метки пробные. Когда обнаруживается, что метка представляет наикратчайший путь до исходной вершины, она превращается в постоянную и никогда более не меняется.

#### *Маршрутизация лавиной*

Другим примером статического алгоритма может служить алгоритм переполнения. В этом алгоритме, каждый поступающий пакет отправляют по всем имеющимся линиям, за исключением той, по которой он поступил. Ясно, что если ничем не ограничить число повторно генерируемых пакетов, то их число может расти неограниченно. Для этого в заголовке каждого изначально генерируемого пакета устанавливается счетчик скачков. При каждой пересылке этот счетчик уменьшается на единицу. Когда он достигает нуля, пакет сбрасывается и далее не посылается. В качестве начального значения счетчика выбирают наихудший случай, например, диаметр транспортной подсети.

Другим приемом, ограничивающим рост дублируемых пакетов, является отслеживание на каждом маршрутизаторе тех пакетов, которые через него

однажды проходили. Такие пакеты сбрасываются и больше не пересылаются. Для этого каждый маршрутизатор, получая пакет непосредственно от абонентской машины, помечает его надлежащим числом. В свою очередь каждый маршрутизатор ведет список номеров, сгенерированных другим маршрутизатором. Если поступивший пакет уже есть в списке, то этот пакет сбрасывается. Для предотвращения безграничного роста списка, вводят ограничительную константу  $k$ . Считается что все номера, начиная с  $k$  и далее, уже встречались.

#### *Маршрутизация на основе потока*

Алгоритмы, которые были рассмотрены до сих пор, принимали в расчет только топологию транспортной среды и никак не учитывали ее загрузку. Хотя, например, в том случае, когда наикратчайший маршрут перегружен, очевидно, лучше воспользоваться пусть более длинным, но менее загруженным маршрутом. Здесь мы рассмотрим статический алгоритм маршрутизации на основе потока, который учитывает, как топологию, так и загрузку транспортной подсети.

В некоторых сетях трафик между каждой парой узлов известен заранее и относительно стабилен. Например, взаимодействие сети торгующих организаций со складом. Время подачи отчетов, размер и форма отчетов известны заранее. В этих условиях, зная пропускную способность каналов, можно с помощью теории массового обслуживания вычислить среднюю задержку пакета в канале. Тогда не трудно построить алгоритм, вычисляющий путь с минимальной задержкой пакета между двумя узлами.

Для реализации этой идеи нужно знать заранее о каждой транспортной среде следующее:

- топологию,
- матрицу трафика  $F_{ij}$ ,
- матрицу пропускных способностей  $C_{ij}$ ,
- алгоритм маршрутизации.

Имея эти данные, строится алгоритм вычисления наикратчайшего пути с точки зрения весов на дугах графа. Если трафик измениться по какой-либо причине, то достаточно перевычислить таблицу, не меняя алгоритма.

#### *Маршрутизация по вектору расстояния*

Все современные транспортные подсети используют динамическую маршрутизацию, а не статическую. Один из наиболее популярных алгоритмов - маршрутизация по вектору расстояния. Этот алгоритм построен на идеях алгоритмов Беллмана-Форда и Форда-Фолкерсона. Он изначально использовался в сети ARPANET и используется по сей день под названием RIP алгоритма (русский аналог названия - метод рельефов). Он используется в сетях Novell, AppleTalk, Cisco маршрутизаторах.

В основе его лежит идея, что у каждого маршрутизатора в транспортной подсети есть таблица расстояний до каждого маршрутизатора в подсети. Периодически маршрутизатор обменивается такой информацией со своими соседями и обновляет информацию в своей таблице. Каждый элемент таблицы состоит из двух полей: первое - номер линии, по которой надо отправлять

пакеты, чтобы достичь нужного места, второе - величина задержки до места назначения. Эта величина задержки может быть измерена в разных единицах: скачках, миллисекундах, длине очереди на линии и т.д.

Каждые  $T$  секунд маршрутизатор шлет своим соседям свой вектор задержек до всех маршрутизаторов в подсети. В свою очередь он получает такие же вектора от своих соседей. Кроме этого, он постоянно замеряет задержки до своих соседей. Поэтому, имея вектора расстояний от соседей и зная расстояние до соседей, маршрутизатор всегда может вычислить кратчайший маршрут.

Алгоритм маршрутизации по вектору расстояния теоретически работает хорошо, но у него есть один недостаток: он очень медленно сходится к правильному значению. Информация о появлении хорошего маршрута в подсети распространяется более или менее быстро, а вот данные о потере, разрушении какого-то маршрута распространяются не столь быстро.

#### *Маршрутизация по состоянию линии*

Алгоритм маршрутизации по вектору расстояний использовался в сети ARPANET до 1979 года, после чего он был заменен. Тому было две основных причины. Первая, поскольку основной мерой задержки была длина очереди, то пропускная способность канала никак не учитывалась. Пока основная масса каналов имела пропускную способность 56Кб/с, это было не страшно. Однако, когда появились каналы на 230 Кб/с и 1.5 Мб/с, это стало недостатком. Вторая проблема – медленная сходимости алгоритма при изменениях. По этим причинам был создан новый алгоритм маршрутизации по состоянию линий.

Основная идея построения этого алгоритма проста и состоит из пяти основных шагов [59-62]:

- а) Определить своих соседей и их сетевые адреса;
- б) Измерить задержку или оценить затраты до каждого соседа;
- в) Сформировать пакет, где указаны все данные, полученные на шаге 2;
- г) Послать этот пакет всем другим маршрутизаторам;
- д) Вычислить наикратчайший маршрут до каждого маршрутизатора.

Топология и все задержки действительно оцениваются экспериментально и сообщаются всем узлам. После этого можно использовать, например, алгоритм Дейкстры для вычисления наикратчайшего маршрута.

### 3.4.2 Решение задачи маршрутизации транспорта с применением теории S-гиперсети

В настоящее время с ростом населения крупных городов значительно увеличилось число автомобилей на дорогах. В результате загруженность городских автомагистралей усилилась, также неизбежно выросла вероятность совершения аварии, по вине которой движение замедляется или вообще останавливается. Вместе с тем человек, естественно, желает добраться куда-либо как можно быстрее. Или он просто ездит с работы домой и обратно и не хочет тратить свое время на «стояние» в пробке, или это курьер, или работник экстренной службы, для которого каждая минута на счету.

Одним из выходов может стать система управления маршрутизацией транспорта, владеющая самой свежей информацией о состоянии дорог и способная вычислить кратчайший путь через весь город. Таким образом, автомобилисты смогут избежать пробок и достигнуть места назначения в кратчайшие сроки [54-57]. Рассмотрим идею работы системы управления маршрутизацией транспорта (рисунок 3.4.2).

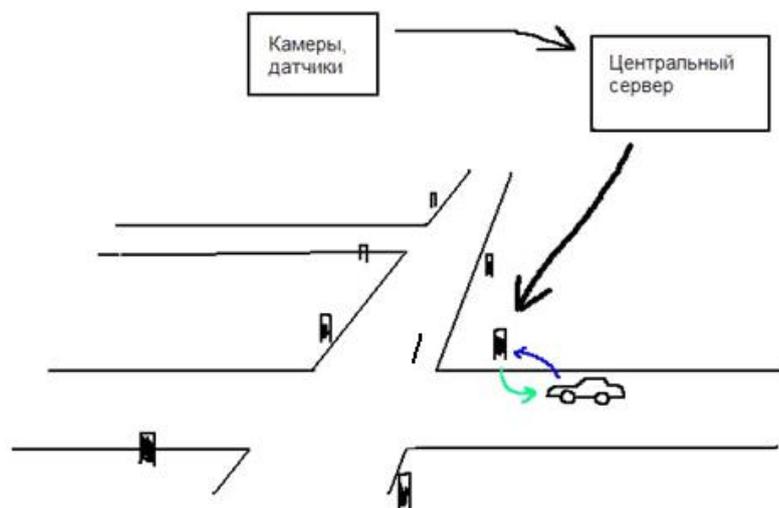


Рисунок 3.4.2 -Работа системы управления маршрутизацией транспорта

На подъездах к перекресткам установлены специальные приемо-передатчики, связанные с центральным сервером. На центральный сервер периодически поступает информация с камер/датчиков, расставленных на дорогах и следящих за состоянием дорожной сети. Эти камеры способны замерять скорость движения потока по участкам дороги. Таким образом, центральный сервер узнает об изменении условий движения на каком-то участке дороги и отправляет соответствующую информацию на приемо-передатчики. Автомобиль оборудован специальной меткой. При подъезде к перекрестку, который является фактически узлом коммутации, приемо-передатчик считывает информацию об автомобиле. Эта информация может быть зашифрована в виде <номер автомобиля, точка назначения>. Затем автомобиль получает направление до следующего узла и продолжает движение.

Таким образом, мы будем рассматривать централизованный адаптивный алгоритм управления маршрутизацией, так как это позволяет снизить затраты, поскольку нет необходимости оснащать приемо-передатчики процессорами большой мощности. Также необходимо учитывать задержки при обмене информацией между приемо-передатчиками и центральным сервером. Однако скорость движения автомобилей гораздо меньше скорости распространения электронного сигнала, так что неприятностей можно избежать, расставив приемо-передатчики соответствующим образом.

#### *Математическая постановка задачи маршрутизации*

Математически задача маршрутизации и транспорта в частности сводится к нахождению кратчайшего пути в (не)ориентированном графе [63-65].

Городская транспортная сеть  $G = (N, A)$  - взвешенный ориентированный граф, где  $N$  – множество вершин (перекрестков)  $A$  – множество дуг (участков дорог, соединяющих две вершины каждый).

Функция  $c: A \rightarrow \mathbb{R}^+$  определяет стоимость каждой дуги  $(i, j) \in A$ . Метрикой в нашем случае является время прохождения пакета по дуге.

Таким образом, целевая функция выглядит следующим образом:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij}, \quad x_{ij} \in \{0,1\} \quad (3.5)$$

$$\sum_{(j,i) \in BS(i)} x_{ji} - \sum_{(i,j) \in FS(i)} x_{ij} = \begin{cases} -n+1, & i=r, \\ 1, & i \in N \setminus \{r\}; \end{cases} \quad (3.6)$$

$BS(i)$  и  $FS(i)$ - это соответственно звезды исходящих и входящих путей в вершину  $i$ , то есть:

$$BS(i) = \{(u, v) \in A \mid v = i\},$$

$$FS(i) = \{(u, v) \in A \mid u = i\};$$

Таким образом, решается целочисленная задача булева линейного программирования – минимизация суммарного времени пути от узла  $r$  к узлу  $i$ .

Существуют разные алгоритмы решения подобного типа задач.

*Метод Дейкстры (Dijkstra)*

Алгоритм Дейкстры – это классический алгоритм для поиска кратчайших путей от одной вершины графа до всех остальных. Пусть  $G = (V, E)$  - взвешенный ориентированный граф с неотрицательными весами дуг,  $s \in V$  - источник,  $t$  - сток, и  $l(v, w)$  длина дуги  $(v, w) \in E$ . Тогда алгоритм выглядит следующим образом [62, с.154]:

*Алгоритм 1*

1) Пусть  $U$  пустое множество, и потенциалы  $p(v) = +\infty$  для каждой вершины  $v \in V$ , кроме  $p(s) = 0$ .

2) Добавляем к  $U$  вершину  $v_0$ , имеющую минимальный потенциал в  $V \setminus U$ . Если  $t \in U$ , стоп.

3) Для каждой вершины  $v \in V$  такой, что  $(v_0, v) \in E$ , если  $p(v_0) + l(v_0, v) < p(v)$ , положить  $p(v) := p(v_0) + l(v_0, v)$  и перемещаемся в вершину  $v (v := v_0)$ .

4) Перейти на шаг 2.

Минимальное время работы алгоритма есть  $O(|E| + |V| \log(|V|))$ .

*Алгоритм A\**

$A^*$  («А-звёздочка»)- это расширение метода Дейкстры, эвристический алгоритм для поиска кратчайшего пути между двумя вершинами. Он использует эвристическое приближение для длины кратчайшего пути от каждой вершины к вершине-назначению. Пусть  $h(u, v)$  - оценщик длины кратчайшего пути между  $u$  и  $v$ . И пусть  $t$  - точка назначения. Тогда алгоритм действует по следующей схеме:

Алгоритм 2

1) Пусть  $U$  пустое множество, и потенциалы  $p(v) = +\infty$  для каждой вершины  $v \in V$ , кроме  $p(s) = 0$ .

2) Добавляем к  $U$  вершину  $v_0$ , для которой  $p(v) + h(v_0, v)$  минимально в  $V \setminus U$ . Если  $v_0 = t$ , стоп.

3) Для каждой вершины  $v \in V$ , такой, что  $(v_0, v) \in E$ , если  $p(v_0) + l(v_0, v) < p(v)$ , положить  $p(v) := p(v_0) + l(v_0, v)$ , перемещаемся в вершину  $v$  ( $v := v_0$ ) и удаляем  $v$  из  $U$ , если  $v \in U$ .

4) Перейти на шаг 2.

Если  $h(v, t)$  удовлетворяет ограничению, что  $h(v, t)$  есть нижняя граница  $h^*(v, t)$ , то полученный путь обязательно будет оптимальным кратчайшим путем.

$$\forall v \in V \quad h(v, t) \leq h^*(v, t) \quad (3.7)$$

Следует отметить, что если  $h(v, t) = h^*(v, t) \quad \forall v \in V$ , алгоритм  $A^*$  сразу найдет только дуги на кратчайшем пути от источника к стоку. Более того, удаление вершины из  $U$  на шаге 3 может быть исключено из алгоритма, если приближение удовлетворяет следующему ограничению, называемому монотонным ограничением:

$$\forall (u, v) \in E \quad l(u, v) + h(v, t) \geq h(u, t) \quad (3.8)$$

В этом случае оценщик называется выполнимым для двойственной задачи (dualfeasibleestimator). Например, для евклидова расстояния на дорожной сети – оценщик для двойственной задачи выполним, то есть двойственная задача выполнима. Очевидно, что  $h^*(v, t)$  также удовлетворяет верхнему ограничению. Следует отметить, что число найденных вершин в этом случае всегда не превосходит числа найденных вершин методом Дейкстры. Фактически метод Дейкстры – это метод  $A^*$ , у которого  $h(v, t) = 0$  для всех вершин.

#### *Двунаправленный метод (The bidirectional Method)*

Двунаправленный метод также используется для поиска кратчайшего пути между двумя вершинами графа. Он не требует эвристическую оценку, но может уменьшить число найденных вершин в большинстве случаев. В этом алгоритме поиск ведется не только от источника к стоку, но и в обратную сторону. Алгоритм работает следующим образом [62, с.186]:

#### *Алгоритм 3*

1) Пусть  $U$  и  $W$  - пустые множества, и пусть потенциалы  $p_s(v)$  и  $p_t(v)$  равны  $+\infty$  для каждой вершины  $v \in V$ , за исключением  $p_s(s) = 0$  и  $p_t(t) = 0$ .

2) Добавляем к  $U$  вершину  $v_0$ , которая имеет наименьший потенциал  $p_s(v)$  в  $V \setminus U$ . Если  $v_0 \in W$ , перейти на шаг 7.

3) Для каждой вершины  $v \in V$ , такой, что  $(v_0, v) \in E$ , если  $p_s(v_0) + l(v_0, v) < p_s(v)$ , положим  $p_s(v) := p_s(v_0) + l(v_0, v)$  и рассматриваем предыдущую  $s$  вершину  $v$  в качестве  $v_0$ .

4) Добавляем к  $W$  вершину  $v_0$ , которая имеет наименьший потенциал  $p_t(v)$  в  $V \setminus W$ . Если  $v_0 \in U$ , перейти на шаг 7.

5) Для каждой вершины  $v \in V$ , такой, что  $(v, v_0) \in E$ , если  $p_t(v_0) + l(v, v_0) < p_t(v)$ , положим  $p_t(v) := p_t(v_0) + l(v, v_0)$  и рассматриваем предыдущую  $t$  вершину  $v$  в качестве  $v_0$ .

6) Перейти на шаг 2.

7) Найти дугу  $(u_0, w_0) \in E$ , которая имеет наименьшее значение  $p_s(u) + l(u, w) + p_t(w)$ . Кратчайший путь от  $s$  до  $t$  состоит из кратчайшего пути от  $s$  до  $u_0$ , дуги  $(u_0, w_0)$  и кратчайшего пути от  $w_0$  до  $t$ .

В целях моделирования процессов маршрутизации транспорта на городских транспортных сетях была разработана программа, использующая метод Дейкстры для поиска кратчайшего пути на графе между двумя вершинами.

Предлагается разбить задачу управления маршрутизацией транспорта на две части: статическую и динамическую.

Если рассматривать городскую транспортную сеть в виде графа, то появляется проблема, связанная с большой размерностью системы. Облегчить вычислительную задачу может разбиение городской сети на меньшие сегменты, связанные между собой через единственные узлы. То есть такие узлы, которые нельзя миновать при движении из одной точки города в другую. При таком разбиении кратчайшие пути не изменятся, так как здесь будет работать принцип оптимальности.

Далее, при первоначальной прокладке маршрута предлагается воспользоваться статистическими данными. Это средние скорости транспортного потока на всех участках дорог в зависимости от дня недели, времени суток, возможно погоды. По этим данным вычисляются веса ребер графа, и кратчайший маршрут вычисляется с помощью любого из алгоритмов поиска кратчайшего пути на графе.

В динамической части предполагается собственно управление маршрутизацией транспортных средств центральным сервером в режиме реального времени. При этом предлагается использовать аналог метода рельефов для информационных сетей.

Метод рельефов - это динамический децентрализованный адаптивный алгоритм. Его удобство заключается в том, что при нём пакету дается направление движения к следующему узлу, что очень удобно в случае маршрутизации автомобильного транспорта. Таким образом, водителю транспортного средства будет дана рекомендация по выбору дальнейшего направления. И при подъезде следующего автомобиля не потребуются никаких перевычислений, при условии сохранения тех же параметров пропускной способности дорог.

### **3.5 Задача о размещении минимального количества видеокамер на заданной транспортной сети**

Автомобильные пробки одна из самых насущных проблем современных мегаполисов, перенасыщенных автотранспортом. Затрудненное движение стало одной из неотъемлемых черт жизни большого города. Поэтому практически в каждом промышленно развитом городе представляет интерес внедрение интеллектуальных транспортных систем (ИТС). Такие системы предназначены для увеличения пропускной способности улиц и магистралей, повышения скорости сообщения и безопасности движения транспорта, сокращения задержек на перекрестках, снижения расхода горюче-смазочных материалов, оздоровления экологической обстановки. Основа всех интеллектуальных транспортных систем: непрерывный сбор информации о загруженности дорог, скорости потоков, авариях и условиях для движения машин, которая далее обрабатывается и удобным образом доводится до сведения водителей либо используется для регулирования движения.

Видеокамеры - одни из самых простых, привычных и действенных средств, используемых в мониторинге. Преимущество видеокамер перед другими системами слежения в том, что есть возможность непосредственно увидеть всё, что находится в обзоре камеры. Современное специализированное программное обеспечение обработки изображений от видеокамер позволяет измерять параметры движущегося потока (например, скорость движения автомобилей, плотность потока движения), определять государственные регистрационные знаки автомобилей, выявлять нарушения правил дорожного движения. В дальнейшем эта информация может быть использована также в самых различных целях.

С развитием цифровой техники и удешевлением видеокамер появилась возможность организовывать видеонаблюдение (с последующим развитием ИТС) в городах с применением минимального количества камер при условии наиболее полного наблюдения за городской транспортной сетью. Данная работа предполагает реализацию именно этого подхода.

Для построения адекватной модели требуется некоторая начальная информация о проблемных участках в городе, скоростные ограничения, средняя загруженность отдельных участков в зависимости от времени суток, времени года. Кроме этого, требуется карта города и знание расстояния от одной точки до другой [75].

На первом шаге территория города визуально делится на районы области с густой дорожной сетью, соединённые с другими районами магистральными дорогами. Размещение камер в одном районе не будет зависеть от размещения в другом. Таким образом, задача несколько упрощается, делится на несколько подзадач меньшей размерности.

По тому же принципу можно произвести разбиение района на микрорайоны. Следующим шагом является определение дорог в районе, на которых сосредоточена основная масса транспортного потока, а также аварийных участков. Из наблюдения следует исключить дороги, которые не дают существенного вклада в поток. К ним относятся дороги с малой загрузкой,

на которых возникновение аварийных ситуаций не меняет характера движения, внутренние дороги жилых зон. Не следует исключать участки, непосредственно примыкающие к магистралям.

Следующей задачей будет определение на полученной территории мест возможного расположения камер. Удобно располагать их так, чтобы в круге обзора находилось как можно больше участков, которые требуется прослеживать. Такими точками служат прежде всего перекрёстки, повороты, крыши или углы зданий, высотные сооружения. После того, как основные точки размещения определены, нужно разделить всю требуемую территорию на множества участков, просматриваемых каждой камерой, и при необходимости добавить или убрать камеры.

Рассмотрим возможные ситуации.

#### 1. Частичное прослеживание

Такая ситуация характерна для длинных промежутков между перекрёстками камеры покрывают участок частично. Если требуется прослеживать участок по всей длине, то нужно добавить точки размещения камер в зависимости от длины участка. Самым простым вариантом в таком случае является размещение через расстояние, как показано на рисунке 3.5.1., равное двум радиусам обзора камеры (или одному радиусу, если камера не может совершать круговое вращение).

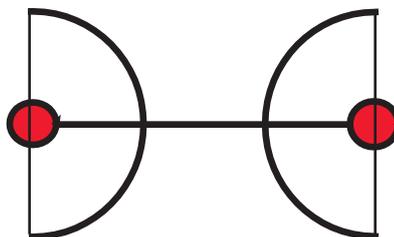


Рисунок 3.5.1 - Частичное прослеживание

Если данный участок не является аварийным или не требует прослеживания по всей длине, то двух камер будет достаточно, так как о возникшей аварийной ситуации внутри промежутка можно судить по второстепенным признакам (например, сравнивая плотность потока на концах промежутка).

На длинных неаварийных участках магистралей, к которым примыкают только второстепенные дороги, также не требуется полное прослеживание и камеры удобно размещать либо на поворотах, либо через определённое расстояние, которое может быть много больше, чем радиус обзора камеры.

#### 2. Частичное перекрытие

Один и тот же участок может в разной степени прослеживаться двумя и более камерами. При незначительном перекрытии (незначительность перекрытия может определяться, например, в процентном соотношении или в единицах длины и её конкретное значение зависит от рассматриваемого

участка) можно разделить покрываемый участок на нужное количество подучастков и указать для каждой камеры наблюдаемые ей территории. Если перекрытие существенное, то можно считать, что обе камеры прослеживают участок целиком. Примеры таких ситуаций показаны на рисунке 3.5.2.

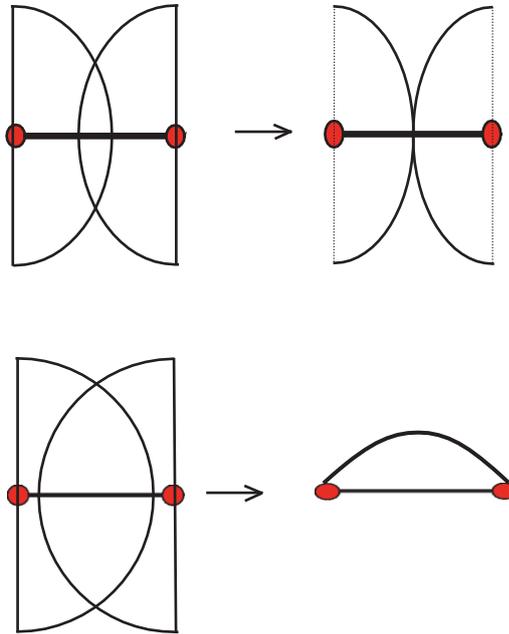


Рисунок 3.5.2 - Незначительное и существенное перекрытия

### 3. Закруглённые участки

Для закруглённых участков дороги применяется либо осцилляция, либо есть точка внутри закругления, с которой беспрепятственно прослеживается весь поворот (рисунок 3.5.3).

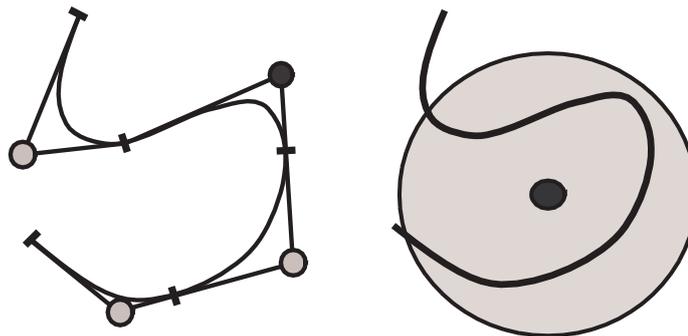


Рисунок 3.5.3 - Закругления

Также из всего множества возможных точек размещения нужно выделить те, в которых камеры должны находиться обязательно. Такие точки могут находиться, например, на мостах, в тоннелях, на особо аварийных участках.

После проделанного анализа задача формализуется следующим образом.

Пусть  $I = \{1, \dots, m\}$  - множество всех возможных мест размещения камер;

$J = \{1, \dots, n\}$  - множество всех наблюдаемых участков;

$$a_{ij} = \begin{cases} 1, & \text{если камера } i \text{ контролирует участок } j \\ 0, & \text{в противном случае} \end{cases}$$

Переменные задачи:

$$x_i = \begin{cases} 1, & \text{если на участке } i \text{ устанавливается камера} \\ 0, & \text{в противном случае} \end{cases}$$

Тогда математическая модель решаемой задачи принимает вид

$$\min \sum_{i \in I} x_i$$

при ограничениях:  $\sum_{i \in I} a_{ij} x_i \geq 1, j \in J;$

$$x_i \in \{0, 1\}, i \in I$$

Данная задача известна в целочисленном программировании как задача о минимальном покрытии (ЗМП).

Для решения ЗМП разработано большое число как точных, так и приближенных методов решения. Все существующие точные алгоритмы связаны с перебором на некотором конечном множестве, задаваемом исходными данными. По объему этого перебора как функции количества исходных данных точные алгоритмы решения ЗМП относятся к экспоненциальным, что делает невозможным их применение для задач с размерностью матрицы более 100. Для практического решения реальных задач используются, как правило, приближенные алгоритмы, среди которых ведущее место принадлежит, так называемым "жадным" алгоритмам [75]. Идея алгоритмов этого типа связана с последовательным построением допустимых вариантов. На каждом алгоритмическом этапе делается попытка достижения некоторой локальной вспомогательной цели (минимизация числа невыполненных ограничений, оптимизация исходной целевой функции и др.) с помощью выбираемых на данном этапе параметров. Локальный подход такого типа не предусматривает попыток оценки ни предыдущих, ни последующих шагов алгоритма.

В рассматриваемой задаче можно ожидать, что результаты предварительного анализа позволят так сконструировать входные данные, что задачу можно будет решать с помощью точных алгоритмов за приемлемое время. Используем для решения задачи два алгоритма: аддитивный алгоритм Балаша и жадный алгоритм, предложенный Д.С.Джонсоном.

Аддитивный алгоритм Балаша

Алгоритм представлен в [75], является методом неявного перебора, в общем случае предназначен для решения следующей задачи:

$$Z(x) = \sum_{j=1}^n c_j x_j \rightarrow \min_{x \in B^n}$$

$$\sum_{j=1}^n a_{ij} x_j \geq b_i, i = 1, \dots, m.$$

Решением назовём множество  $\{x_j : x_j = 1, j \in N_1; x_j = 0, j \in N \setminus N_1, N = \{1, \dots, n\}\}$ .

Решение является допустимым, если выполняются неравенства

$$\sum_{j=1}^n a_{ij} x_j \geq b_i .$$

Предположим, что выполнены неравенства  $0 \leq c_1 \leq c_2 \leq \dots \leq c_n$ .

Задача имеет  $2^n$  различных решений. Разобьём все множество решений на  $(n+1)$  подмножества с номерами  $k=0,1,\dots,n$  таким образом, что  $k$ -е подмножество содержит все решения с  $k$  переменными, равными единице, и  $n-k$  переменными, равными нулю.

Следовательно:

— при  $k=0$ , подмножество решений состоит из единственного решения  $x=0$ ;

— при  $k=1$ , подмножество решений включает  $C_n^1$  решений, в которых  $x_i=1; x_j=0, j \neq i, i=1,\dots,n$ ;

—  $k$ -е подмножество состоит из  $C_n^k$  решений.

Затем все решения упорядочиваются с помощью диаграммы, где каждая вершина, содержащая список  $N_1$ , представляет решение, в котором переменные с индексами из  $N_1$  равны единице, а остальные нулю.

Если в диаграмме существует путь из вершины  $u$  в вершину  $v$ , то вершина  $u$  называется предшествующей для вершины  $v$ , а вершина  $v$  называется следующей за  $u$ . Таким образом, решения частично упорядочены.

Для проверки допустимости решений будем проверять условие

$$\sum_{j=1}^n a_{ij} x_j \geq b_i .$$

Алгоритм начинает работу с вершины  $x=0$ , затем просматривает следующие за ней вершины. Если некоторая вершина соответствует допустимому решению, то не надо просматривать следующие за ней вершины, они исключаются.

На основе рассмотренного алгоритма Балаша разработаны 3 варианта: приближённый алгоритм C1, алгоритм Climb - при проходе диаграммы снизу и алгоритм Descend - при проходе диаграммы сверху, и C1. Программы для реализации алгоритмов написаны на языке С.

### 1) Приближённый алгоритм C1

Алгоритм использует теоретико-множественную постановку задачи.

Множества  $S_i$  строятся для нашей задачи следующим образом: для каждого номера  $i$  множество  $S_i$  представляет собой набор номеров тех участков, которые прослеживаются  $i$ -й камерой.

Пусть  $Sub$  содержит включенные в покрытие подмножества,  $Unsov$  содержит ещё не покрытые элементы.

Алгоритм C1 состоит в следующем:

1. полагаем  $Sub = \emptyset$ ,  $Uncov = \bigcup_{S \in F} S$ ;  $N = |F|$ ;  $Set(i) = S_i$ ,  $1 \leq i \leq N$ ;
2. если  $Uncov = \emptyset$ , алгоритм заканчивает работу и возвращает в качестве решения  $Sub$ ;
3. выбираем  $j \leq N$  такое, что  $|Set(j)|$  максимально;
4. полагаем  $Sub = Sub \cup \{S_j\}$ ,  $Uncov = Uncov - Set(j)$ ,  $Set(i) = Set(i) - Set(j)$ ,  $1 \leq i \leq N$ ;
5. вернуться на шаг 2.

Особенности реализации: из входной матрицы составляются множества  $S_i$ , каждое такое множество хранится в виде списка, есть счётчик количества элементов для каждого  $S_i$ . Функция `del_el` удаляет компоненты множества, включенного в покрытие, из множеств, не включенных в него (удаляет элементы списков), и уменьшает счётчики.

Если счётчик равен нулю, множество исключается из рассмотрения. Программа заканчивает работу, когда все счётчики равны нулю.

### 2) Алгоритм *Climb*

Учитывая структуру результатов, получаемых после предварительного анализа, можно предполагать, что решение найдётся быстрее, если просматривать диаграмму снизу вверх. При этом достаточно находить первое по очереди решение на уровне и перемещаться выше, так как все остальные возможные решения на уровне будут не лучше первого найденного. Алгоритм остановится, когда на уровне не окажется допустимых решений и оптимальным будет решение, полученное на предыдущем уровне. Данный подход реализован в программе *Climb*.

Особенности реализации: на каждом уровне программа последовательно генерирует перестановки (вектор длины `depth` с номерами камер) и одновременно проверяет неравенства  $\sum_{j=1}^n a_{ij}x_j \geq b_i$ . Для того, чтобы проверить эти неравенства, не нужно считать сумму, достаточно сравнить строку матрицы и перестановку. Если в строке есть хотя бы одна единица на месте, определяемом номерами в перестановке, то неравенство для данной строки выполняется. При нахождении первого допустимого решения проверка на текущем уровне заканчивается, полученное решение запоминается и совершается переход на уровень выше.

Результат программы *C1* удобно использовать в *Down*. Размер полученного вектора решения, уменьшенный на 1, нужно задать в качестве начального параметра `depth`. Программа либо найдёт решение на уровнях выше и оно будет лучше, либо не найдёт, тогда полученный вектор будет оптимальным.

### 3) *Descend*

В данной задаче будет удобно просматривать вершины в другом порядке последовательно по уровню в диаграмме, то есть просматривать решения с одинаковым количеством индексов в списке  $N_1$ . В этом случае первое же

допустимое решение будет одним из оптимальных. Данный метод реализован в программе Descend.

Реализовано аналогично Climb с тем отличием, что уровни диаграммы проходятся сверху. Кроме того, в программе также можно использовать результат C1: поскольку он отличается от точного решения незначительно, то размер вектора точного решения должен быть не меньше целой части  $1 + \ln N$ , поэтому в качестве начального значения depth нужно использовать именно это выражение.

Данный алгоритм лучше работает на мало разреженных матрицах.

#### 4. Вычисления и результаты

Для тестирования был произведен примерный анализ двух микрорайонов и составлены входные матрицы (Таблица 3.5.1). Среднее количество точек размещения камер 20-40. При  $N > 40$  эффективно работает только приближённый алгоритм, поскольку в точном может потребоваться полный просмотр срединного уровня (в котором  $CN=2 > 1010$  элементов).

Таблица 3.5.1. Входные матрицы двух микрорайонов районов

Входная матрица микрорайона А	Входная матрица микрорайона В
110000000000100000000000	1100000000000000000000000000
110000000000100000000000	0110000000000000000000000000
100000000000010000000000	010000000000000000010000000000
10000000000000000010000100	00000000000000000001100000000000
01000000000010000000000000	00000000000000000001000000000010
00100000000001000000000000	1000000000000000000000000000010
00110000000010000000000000	10000000000000000000000000000011
000100000000100100000000100	0000000000000000000110000000000000
000100000000100100000000100	0000000000000000000110000000000000
000100000000101100000000100	0000000000000000000110000000000000
000100000000001100000000100	000000000000000000011100000000000000
00001000000000100000000000	000000000000000000011100000000000000
0000100000000000000000100000	000000000000000000011000000000000000
0000100000000000000000100000	000000000000000000011100000000000000
000001000000000000000011000000	000000000000000000011000000000000000
1000000000000000000010000100	00000000011000000000000000000000
000001100000000000110000000	00000000011000000000000000000000
00000100000000001000000000	00000001100000000000000000000000
000001100000000000110000000	00000001100000000000000000000000
000100000000000000100000000	00000110000000000000000000000000
000001101000000000110000000	00011000000000000000000000000000
00000010000010000000000000	00110000000000000000000000000000
001000000000000000100000000	00100000000000000000100000000000
00000010100000001100000000	00000000000000000000110000000000
00000010100000001000000000	00000000000000000000110000000000
000000001000000000000000001	00000000000000000000001100000000

000000100000000000000000010	0000000100000000000000010000000
000010000000000000001001000	0000000000100000000000010000000
00000000010000000000100000	00000000000000000000000011000000
00000000001100000000000000	00000000000000000000000001100000
000000000010000000000000001	00000000000000000000000000110000
00000000001100000000000000	00000000000000000000000000011000
000000000001000000000000010	00000000000000000000000000001100
00000000000110000000000000	01000000000000000001000000011000
000000000010000000000100000	0000000000000000001000000000100
	000000000000001000000000010000000
	00000000000000000100000000110000
	00000000000000000110000000000001

Входными данными для всех программ является булева матрица  $A(i, j)$  размерности  $N \otimes M$ , формируемая следующим образом:  $A(i, j) = 1$ , если камера  $j$  отслеживает участок  $i$ , в противном случае  $A(i, j) = 0$ . Матрица записывается в текстовый файл построчно, строка может содержать или не содержать пробелы между элементами, после ввода матрицы должен быть осуществлён переход на новую строку.

При построении матрицы столбцы можно разместить в таком порядке, что первые  $k$  столбцов будут давать какое-то допустимое решение, программы будут работать быстрее (из-за особенности генерации перестановок). Для этого достаточно, чтобы в каждой строке матрицы была хотя бы одна единица.

В матрице не должно быть нулевых строк и столбцов. Нулевая строка говорит о том, что какой-то участок не просматривается и у задачи нет решения; нулевой столбец признак того, что какая-то камера не прослеживает ни одного участка и, следовательно, не нужна в покрытии.

Если в матрице есть строка, в которой единица встречается только один раз, значит участок прослеживается только одной камерой и эта камера обязательно будет включена в покрытие, следовательно, можно удалить строку и столбец из матрицы, на пересечении которых находится единица и таким образом уменьшить размерность (впоследствии к полученному покрытию нужно будет добавить эту камеру). Кроме того, нужно все остальные участки, просматриваемые этой камерой, удалить из наборов участков других камер (удаляются строки, содержавшие единичные элементы данной камеры).

Программы Descend и Climb считают размерность входной матрицы (параметры  $N$  и  $M$ ). Все программы в качестве результата возвращают строку с номерами камер, вошедших в покрытие.

Вычисления производились при частоте процессора 1ГГц (таблица 3.5.2).

Таблица 3.5.2. Продолжительности вычислений программами разных алгоритмов

Матрица	Программа	Время работы
---------	-----------	--------------

A	C1		менее 1 секунды
A	Climb		1 секунда
A	Descend		2 секунды
A	Climb	при depth=10	1 секунда
A	Descend	при depth=2	2 секунды
B	C1		менее 1 секунды
B	Climb		5 мин 52 сек
B	Descend		11 мин 23 сек
B	Climb	при depth=15	5 мин 31 сек
B	Descend	при depth=3	11 мин 23 сек

В данном разделе описаны мероприятия, проведение которых позволит так разделить существующую транспортную систему города, чтобы в каждой полученной части можно было организовать систему видеонаблюдения при минимальном количестве камер и просмотре всех требуемых участков, используя при этом в расчётах точные алгоритмы для задачи о покрытии.

В результате работы получен необходимый комплекс мероприятий и программы, реализующие для данной задачи приближённый и адаптированный точный алгоритмы [75]. Работу можно использовать для организации любых систем видеонаблюдения.

### 3.6 Практическая реализация результатов исследования и разработка программы

Предлагаемый алгоритм по входным данным будет проверять возможность пропуска данного потока пунктами (из вершины S в вершину T) за время, не превышающее данное. Алгоритм на выходе будет либо выдавать, что поток проходит, либо, если это невозможно, будет выявлять пути (Memory.стоимость) и конкретные перекрёстки (список TrafficWay, по структуре аналогичный Memory.путь), на которых возникают заторы, мешающие успешному проезду заданного потока в заданное время.

Алгоритм TrafficWay

1-шаг.Используем алгоритм Mflow, в котором:

- а) в качестве начального и конечного пунктов берем вершины S и T,
- б) в качестве  $c_{ij}$  берем  $P_{\max}(i, j)$ ,
- в) в качестве  $cost(i, j)$  берем  $h(P_{\max}(i, j), P_{\text{real}}(i, j), \text{zerotime}(i, j))$ ,
- г) в качестве M берем Time.

2-шаг. Преобразуем полученный в предыдущем шаге список в:

Memory.стоимость = round(Time - Memory.стоимость) + 1.Затем получаем F как сумму всех Memory.стоимость.

3-шаг. Полученный на втором шаге максимальный поток  $F$  сравниваем с  $M$ :

а) если  $M \leq F$ , то на выходе алгоритм завершает работу с выводом «Успех».

б) если  $M > F$ , то переходим к 4-ему шагу.

4-шаг. Берем первый путь из списка `Мемогу.путь` и начинаем по нему идти.

а) берем первую дугу – переходим к пункту б).

б) сравниваем  $P_{real}(i,j)$  текущей дуги с суммой  $P_{max}(j,k)$  всех исходящих из вершины дуг, в которую входит текущая. Если эта сумма больше чем  $P_{real}(i,j)$  (на рисунке  $3.6.1 a < b + c + d$ ), то переходим к следующей дуге, если такое существует, и повторяем пункт б). Если не существует, то переходим к пункту г). Если сумма меньше  $P_{real}(i,j)$ , то переходим к пункту в).

в) если сумма оказалась меньше, то в список `TrafficWay` (в строку с номером, аналогичным номеру записи текущего пути в списке `Мемогу.путь`) заносим номер вершины, в которую входила текущая дуга.

г) если следующей дуги в пути не существует, то переходим к шагу 5.

5-шаг. Берем следующий путь и переходим к шагу 4а. Если такого пути нет, то переходим к 6 шагу.

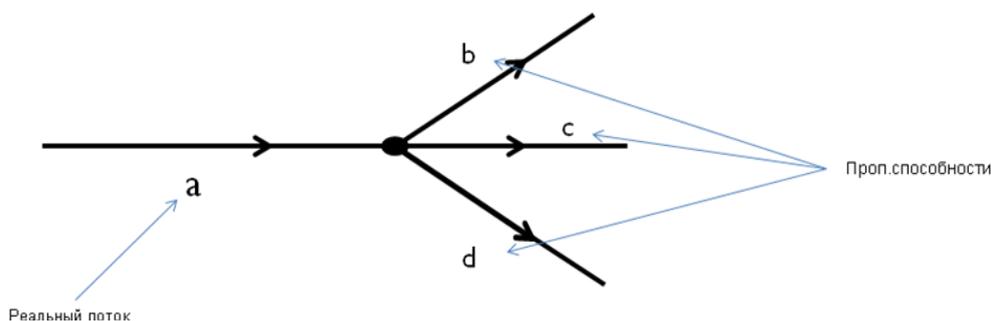


Рисунок 3.6.1 - Пояснение к шагу 4б

6-шаг. Если `TrafficWay` пуст, то алгоритм завершает свою работу с выводом «Некорректные входные данные». Если `TrafficWay` содержит записи, то алгоритм также завершает работу и на выходе выдает два списка: `Мемогу.путь` – хранит пути и `TrafficWay` – хранит пробки.

Полученные алгоритмы были реализованы на языке C++. Главное меню программы показано на рисунке 3.6.2. Входные данные берутся из текстового файла.

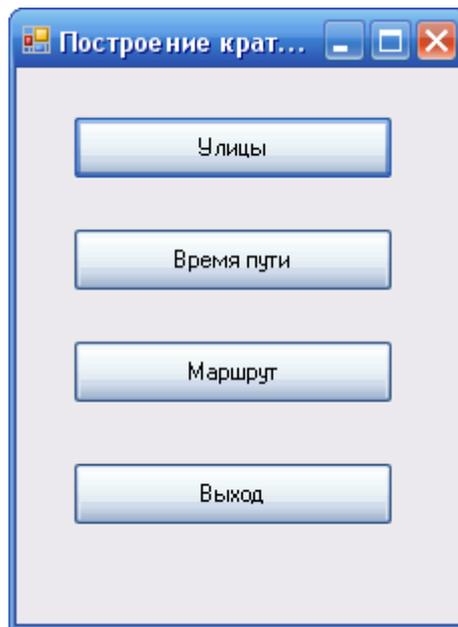


Рисунок 3.6.2 - Главное меню программы

Программа анализирует транспортную сеть на прохождение по ней потока (рисунок 3.6.3) и ищет пробки и предлагает короткий вариант маршрута (рисунок 3.6.4). Если поток не может пройти, то для получения результата, отражающего реальное состояние дорог, при решении связанных задач, этот модуль должен быть использован в итеративной связке с другими (устранение пробок, построение развязок – в этом случае нужно, чтобы модуль, строящий развязки, перед подачей модулю, предложенному в данной работе, преобразовал развязку в вид, который понимает модель – обычные дуги, вершины и веса на них).

*Доказательство корректности алгоритма*

Алгоритм TrafficWay корректен, его временная сложность  $O((n^2+m^2)mC)$ , где  $n$  – число вершин графа  $(V, E)$ ,  $m$  – число дуг,  $C = \max\{P_{\max_{ij}}\}$ .

Пункт А	Пункт В	Время
ул. Сатпаева	просп. Достык	20
просп. Абая	просп. Достык	15
ул. Курмангазы	просп. Достык	10
просп. Абая	ул. Фурманова	13
ул. Сатпаева	ул. Фурманова	7
ул. Курмангазы	ул. Фурманова	10
просп. Аль-Фараби	просп. Достык	23
ул. Шевченко	просп. Достык	6
просп. Абая	ул. Желтоксан	5
ул. Хаджи Мукана	просп. Достык	28
просп. Абая	просп. Сакена Сейфуллина	30
ул. Тимирязева	просп. Сакена Сейфуллина	23
просп. Абая	ул. Наурызбай батыра	17
просп. Абая	ул. Пушкина	5
просп. Абая	ул. Чайковского	7
просп. Абая	ул. Байтурсынова	20
ул. Шевченко	просп. Сакена Сейфуллина	5
ул. Сатпаева	ул. Желтоксан	3
ул. Сатпаева	просп. Сакена Сейфуллина	23
просп. Аль-Фараби	просп. Сакена Сейфуллина	10
ул. Сатпаева	ул. Байтурсынова	10
ул. Сатпаева	ул. Наурызбай батыра	15

Рисунок 3.6.3 - Окно «Время пути»

Вносятся время, затрачиваемое на каждом перекрестке. Впоследствии данные используются для построения матрицы смежности.

Маршрут

Пункт А: просп. Абая

Пункт В: ул. Сатпаева

Построить      Закрыть

Короткий маршрут между просп. Абая и ул. Сатпаева: ул. Сатпаева <- ул. Желтоксан <- просп. Абая. Потраченное время 8

Рисунок 3.6.4 - Построение короткого маршрута

Данный алгоритм работает по следующему принципу. Сначала он ищет поток, который можно в принципе пустить из пункта S в пункт T за время, не превосходящее Time (так как F – был максимальный поток за единицу времени,

то, чтобы понять сколько машин успеют пройти за время Time из S в T, нужно было взять Time минус время, потраченное на элементарном пути плюс еще одна единица, которая успела доехать до T в последний момент), и, после того, как нашли его, начинаем проверку. Если входной поток не превысил максимальный, то считаем, что он успевает благополучно пройти из S в T за время Time. Если же превышает, то тогда мы предполагаем, что на пути нашего потока встречаются заторы и, начиная с четвертого шага, мы их начинаем искать. Для этого берем поочередно пути из списка Memory.путь, который был сформирован еще при выполнении алгоритма Mflow и для каждого из этих путей проделываем следующие выкладки: Идём по нему и проверяем, чтобы реальный (усредненный) поток, входя в вершину, весь мог пройти сквозь нее, то есть пропускные способности всех исходящих дуг ему это позволяли. Если, пройдя по всем путям, мы не сможем найти пробок – TrafficWay будет пуст, то тогда, скорее всего, входные данные (поток M или время Time) были заданы некорректно. Если же такие найдены, то тогда мы запоминаем с помощью TrafficWay, в каком пути и на каком узле эта пробка возникла. После чего на выход передаем эти два списка.

Вычислим временную сложность данного алгоритма по шагам.

Выполнение алгоритма Mflow дает временную сложность  $O((n^2+m)mC)$ . Изменение Memory.стоимости затем суммирование занимает не более, чем  $O(mC)$  операций. Далее, когда начинаем проверять все пути (не более, чем  $mC$ ) и на каждом пути сравниваем каждую дугу (не более, чем  $m$ ) с суммой исходящих (не более  $m$ ), то получаем  $O(m^3C)$ . В итоге получаем  $O(n^2mC+m^2C+mC+m^3C) = O((n^2m+m^3+m+m^2)C) = O((n^2+m^2)mC)$ .

### **Выводы к разделу 3**

Рассмотрены особенности алгоритма Форда-Фалкерсона и процедура расстановки пометок для задачи о максимальном потоке.

Исследовано влияние одностороннего движения на величину транспортного потока, дано описание уровней дорожно-транспортной сети.

Разработана концепция имитационной модели транспортных потоков и управляющих транспортных систем на основе теории S-гиперсетей.

В рамках имитационной модели проведено моделирование потоков на участке дорожно-транспортной сети и выявлено, в целом, положительное влияние внедрения одностороннего движения.

Рассмотрены примеры расчета пропускной способности участка дорожно-транспортной сети.

Разработан алгоритм устранения пробок путем изменения режимов работы светофоров устранившие дополнительные пробки на боковых, к этому маршруту, дорогах.

Разработан алгоритм маршрутизации и составлена таблица маршрутизации.

Сформулирован принцип оптимальности. Математическая задача маршрутизации сводится к нахождению кратчайшего пути в неориентированном графе.

Рассмотрены разные алгоритмы решения задач маршрутизации с применением теории S-гиперсети.

Предложен алгоритм и программа, минимизирующие количество точек размещения видеокамер на заданной территории при условии полного её прослеживания.

В результате исследования были построены:

- Алгоритм для определения кратчайшего пути в ориентированном мультиграфе.
- Алгоритм Mflow для нахождения максимального потока, стоимость которого не превышает заданную.
- Алгоритм TrafficWay для анализа транспортной сети и выявления на ней узких мест.

Доказана корректность алгоритма TrafficWay. Полученные алгоритмы были реализованы на языке C++. Так как алгоритм TrafficWay только анализирует транспортную сеть на прохождение по ней потока и ищет пробки. Если поток не может пройти, то для получения результата, отражающего реальное состояние дорог, при решении связанных задач, этот модуль должен быть использован в итеративной связке с другими.

Основные результаты исследований по данному разделу содержится в следующих публикациях [66-76].

## ЗАКЛЮЧЕНИЕ

Диссертация является научной квалификационной работой, в которой были рассмотрены математическое моделирование транспортных систем города с применением теории S-гиперсетей. Основные научные результаты диссертации, практические выводы и рекомендации, полученные при выполнении исследований, заключаются в нижеизложенных предложениях.

Проведен анализ инфраструктуры мегаполиса, ее математическое моделирование в виде S-гиперсети для основных видов транспорта.

Выявлено и показано, что теория S-гиперсетей применима для анализа и синтеза многих систем сетевой структуры, в том числе, для задач анализа межсетевых структурных взаимодействий сложных систем различной природы.

Сформулирована и доказана теорема планарности гиперсети .

Дана математическая формулировка маршрутов и метрик в S-гиперсетях, позволяющая вычислить отдаленность и расстояния с помощью известных методов на специально построенных графах, орграфах, гиперграфах и ультраграфах.

В исследованиях транспортных потоков применяются идеи, методы и алгоритмы нелинейной динамики и возможностей управления. Их целесообразность обоснована наличием в транспортном потоке устойчивых и неустойчивых режимов движения, потерь устойчивости при изменении условий движения, нелинейных обратных связей, необходимости в большом числе переменных для адекватного описания системы. Рассмотрены особенности алгоритма Форда-Фалкерсона и процедура расстановки пометок для задачи о максимальном потоке. Исследовано влияние одностороннего движения на величину транспортного потока, дано описание уровней дорожно-транспортной сети.

Проведен анализ методов моделирования и управления движением транспорта через перекресток. По результатам сравнения автоматических контроллеров, основанных по нечеткой логике, показан выигрыш по времени прохождения автомобилями перекрестка по сравнению с обычными методами, в которых время горения сигналов светофора было фиксировано.

Рассмотрены и доказаны возможности определения пропускной способности УДС с помощью сетей Петри. Рассмотрены примеры расчета пропускной способности участка дорожно-транспортной сети.

Выполнено моделирование перекрестка и блока перекрестков с использованием нечеткой логики в управлении светофором. Результаты сравнения автоматических контроллеров, основанных на нечеткой логике, показали выигрыш во времени прохождения автомобилями перекрестка по сравнению с обычными методами.

Рассмотрены способы управления транспортными потоками на светофорных объектах с помощью разгонных светофоров. На основе проведенного анализа выявлены положительные особенности применения разгонных светофоров. Разработан алгоритм устранения пробок путем

изменения режимов работы светофоров, устранившие дополнительные пробки на боковых к этому маршруту дорогах.

Проведен анализ основных типов развязок. Построена математическая модель для каждого типа развязок. Предложен алгоритм построения оптимальной транспортной развязки, при котором существенно уменьшается время прохождения данного узла автотранспортом и снижаются затраты.

Рассмотрена задача размещения остановок на транспортной сети города и разработан алгоритм расчета числа остановок. Разработаны алгоритмы маршрутизации с применением теории S-гиперсети и составлена таблица маршрутизации. Сформулирован принцип оптимальности маршрутизации, который сводится к нахождению кратчайшего пути в неориентированном графе.

Предложены алгоритм и программа, минимизирующие количество точек размещения видеокамер на заданной территории при условии полного её прослеживания.

В результате исследования были построены алгоритмы:

- для определения кратчайшего пути в ориентированном мультиграфе.
- Mflow для нахождения максимального потока, стоимость которого не превышает заданную.
- TrafficWay для анализа транспортной сети и выявления на ней узких мест.

Доказана корректность алгоритма TrafficWay, который анализирует транспортную сеть на проходимость по ней потока и ищет пробки. Если поток не может пройти, то для получения результата, отражающего реальное состояние дорог, при решении связанных задач, этот модуль должен быть использован в итеративной связке с другими. Полученные алгоритмы доведены до программной реализации на языке C++.

Достоверности научных результатов, выводов и положений, сформулированных в диссертации, базируются на предложенных математических моделях и алгоритмах, характеризующих различные уровни организации транспортных систем.

Научные результаты позволили исследовать поведение транспортной системы мегаполиса в целом, которое не может быть выяснено рассмотрением отдельных ее звеньев. Разработанные в рамках проведенного исследования модели и алгоритмы обладают гибкостью и универсальностью.

Таким образом, проведенные в работе исследования показали возможность повышения эффективности и оперативности управления транспортными потоками города на основе разработки комплексных решений с использованием кибер-физических систем мониторинга, моделирования, прогнозирования и оптимизации с применением теории S-гиперсети.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 О количестве легковых автомобилей в Республике Казахстан. [Электронный ресурс]: <http://stat.gov.kz>
- 2 Алматы ежедневно теряет более 600 млн. тенге из-за пробок. [Электронный ресурс]: <https://kapital.kz/economic/67073/almaty-ezhednevno-teryayet-bolee-600-mln-tenge-iz-za-probok.html>
- 3 Бурлуцкий А.А., Киргисарова Е.Ю. Моделирование транспортных систем городов с использованием программного продукта VISUM. [Электронный ресурс]: <http://www.niigip.ru/info/stat/156-----visum>
- 4 «Умные светофоры»: Интеллектуальная система контроля дорожного движения. [Электронный ресурс]: <http://mer-voronezha.ru/articles/71>
- 5 Schadschneider A., Schreckenberg M. Cellular automata for traffic. [Электронный ресурс]: <http://arXiv.org/abs/cond-mat/9511037v1>
- 6 Turky A.M. The Use of Genetic Algorithm for Traffic Light and Pedestrian Crossing Control // IJCSNS International Journal of Computer Science and Network Security. -2009.-Vol.9, №2. -P.88-96.
- 7 Madhavan Nair B., Cai J. A fuzzy Logic Controller for Isolated Signalized Intersection with Traffic Abnormality Considered //Proc. 2007 IEEE Intelligent Vehicles Symposium Istanbul. -2007. -P.1229-1233.
- 8 Hartman D. Head leading algorithm for urban traffic modeling //Proc. 16th European Simulation Symposium György Lipovszki, IstvánMolnár © SCS Press. – 2004. -P.10-17.
- 9 Mancinelli E. On Traffic Light Control of Regular Towns //Rapport de recherché. Institut national de rechercheeninformatique et enautomatique. -2001. - P.12-18.
- 10 Gonzalez H. Adaptive Fastest Path Computation on a Road Network: A Traffic Mining Approach //VLDB 07.-2007. -P.794-805.
- 11 Reijmers J.J. Traffic guidance systems //Course ET4-024. Delft University of Technology. -2011. -P. 99-108.
- 12 Jee-Hyong LEE. Traffic Control of Intersection Group Based on Fuzzy Logic //Proc. 6th International Fuzzy Systems Association World Congress. -1995. -P. 465-468.
- 13 Дрю Д. Теория транспортных потоков и управление ими. - М.: Транспорт, 1972. – 424 с.
- 14 Иносэ Х., Хамада Т. Управление дорожным движением. - М.: Транспорт, 1983. - 248 с.
- 15 Черепанов В.А. Транспорт в планировке городов. - М.: Издательство литературы по строительству, 1970. – 304 с.
- 16 Фишельсон М.С. Городские пути сообщения. - М.: Высш.Школа, 1980. – 296 с.
- 17 Клинковштейн Г.И. Организация дорожного движения: учебник для вузов. – Изд. 5-е, перер. и доп. – М.: Транспорт, 2001 – 247 с.
- 18 Пугачёв И.Н. Теория транспортных потоков. – Хабаровск, 2002. – 40 с.
- 19 Капитанов В.Т. Управление транспортными потоками в городах. - М.:

Транспорт, 1985. - 94 с.

20 Абрамова Л.С., Чернобаев Н.С. Имитационная модель управления транспортными потоками // Вестник ХНАДУ. -2009. -№47. - С.93-96.

21 Абрамова Л.С., Ширин В.В. Моделирование заторовых ситуаций по улично-дорожной сети // Автомобильный транспорт. – 2009. -№25. - С.66-69.

22 Попков В.К. Математические модели связности. – Новосибирск: Изд. ИВМиМГ СО РАН, 2006. - 409 с.

23 Попков В.К. Модели и методы структурирования сложных систем // Материалы IV Международной конференции «Математика, ее приложения и математическое образование» МПМО`11. -Улан-Удэ, 2011. - С.62-66

24 Попков В.К. Некоторые аспекты теории S-гиперсетей // Доклады Седьмой международной азиатской школы-семинара Проблемы оптимизации сложных систем. -Ташкент, 2011. - С.7-13.

25 Попков В.К. О моделировании городских транспортных систем гиперсетями // Автоматика и телемеханика. -2011. -№ 6. - С.179-189.

26 Попков В.К., Гаврилов А.С. Минимальная реализация и другие операции над гиперсетями // Проблемы информатики. -2011. -№4. - С.4-16.

27 Попков В.К. Применение теории S-гиперсетей для моделирования систем сетевой структур // Проблемы информатики. -2010. -№ 4. - С.17-40.

28 Попков Г.В. Применение теории гиперсетей в задачах оптимизации систем сетевой структуры // Труды ИВМиМГ СО РАН. Сер. Информатика. - Новосибирск: Изд.ИВМиМГ СО РАН, 2001. - С.87-92.

29 Дубровин Е.Н. Пересечение в разных уровнях на городских магистралях. - М.: Транспорт, 1968. - 278 с.

30 Beckman M., McGuire C.B., Winsten C.B. Studies in the economics of transportation. - СТ:Yale University Press, 1956. – 145 p.

31 Ахмедиярова А.Т., Утепбергенов И.Т. Касымова Д.Т. Метод анализа транспортной сети для выявления узких мест // Интеллектуальные технологии на транспорте. – 2016. - №3. - С.9-14.

32 Утепбергенов И.Т., Попков В.К., Ахмедиярова А.Т. Применение теории S-гиперсетей для моделирования систем сетевой структуры // Вестник КазНТУ имени К.Сатпаева. -2016. -№2. -С.63-69.

33 Утепбергенов И.Т., Ахмедиярова А.Т. Классификация симбиотических связей в теории S-гиперсетей // Материалы международной научно-практической конференции «Современная экономика и образование: проблемы, возможности и перспективы развития». - Алматы, 2016. –Т.2. –С.511-513.

34 Ахмедиярова А.Т., Куандыкова Д.Р., БуранбаеваА.И. Стохастические модели транспортных сетей мегаполиса // Материалы международной научно-практической конференции «Современная экономика и образование: проблемы, возможности и перспективы развития». - Алматы, 2016. –Т.2. –С.425-428.

35 Ахмедиярова А.Т., Касымова Д.Т., БуранбаеваА.И. Структурные модели транспортных сетей мегаполиса // Материалы международной научно-практической конференции «Современная экономика и образование: проблемы, возможности и перспективы развития». - Алматы, 2016. –Т.2. –С.428-433.

- 36 Кочерга В.Г., Зырянов В.В. Оценка и прогнозирование параметров дорожного движения в интеллектуальных транспортных системах. - Ростов н/Д: Рост.гос.строит, ун-т, 2002. - 130 с.
- 37 Кременец Ю.А. Управление движением по отдельным направлениям. – Таллин: TrafficSafety, 1990. - С.137–139.
- 38 Левашев А.Г., Михайлов А.Ю. Состояние методов расчета регулируемых пересечений // Вестник ИрГТУ. -2003. -№3-4. -С.71–76.
- 39 Калимолдаев М.Н., Утепбергенов И.Т., Ахмедиярова А.Т. Исследование методов моделирования и управления движением транспорта через перекресток // Scienceandworld. -2015. - №12 (28). – С.8-16.
- 40 Ахмедиярова А.Т., Бижанова А.С., Касымова Д.Т. О задаче управления светофорами на перекрестках // Вестник КазАТК имени М.Тынышпаева. - 2016. - №1. - С.74-77.
- 41 Попков В.К., Ахмедиярова А.Т. Задача размещения остановок на транспортной сети города // Материалы 7-ой Международной школы-семинара, серия «Информатика». – Новосибирск, 2011. – С.178-189
- 42 Попков В.К., Ахмедиярова А.Т. Об одной задаче размещения остановок на транспортной сети города // Труды Международной научно-практической конференции «Информационно-инновационные технологии, интеграция науки, образование и бизнеса», посвященной 20-летию Независимости Республики Казахстан. – Алматы, 2011. – С.478-495.
- 43 Утепбергенов И.Т., Ахмедиярова А.Т., Касымова Д.Т. О задаче моделирования регулярного города с помощью сети Петри // ВестникКазАТК имени М.Тынышпаева. – 2016. – №1. – С.77-81.
- 44 Жогаль С.И., Максимей И.В. Задачи и модели исследования операций: учебное пособие. - Гомель: БелГУТ, 1999. - 109с.
- 45 Зайченко Ю.П. Исследование операций: учебное пособие. – Киев: Слово, 2002. - 320 с.
- 46 Поляков К.Ю. Теория автоматического управления. – СПб, 2008. – 80с.
- 47 Соболев И.М. Метод Монте-Карло. – М.: Наука, 1968. - 64 с.
- 48 Сукач Е.И. Применение имитационного моделирования для исследования динамики транспортных потоков региона // Известия ГГУ имени Ф. Скорины. - 2006. - № 4(37). - С.96-99.
- 49Кристофидес Н. Теория графов. Алгоритмический подход. - М.:Мир, 1978. – 432 с.
- 50 Томас Х. Кормен. Алгоритмы: построение и анализ. - М.: Вильямс, 2006. - 1296 с.
- 51 Розенберг И.Н., Старостина Т.А. Определение максимального потока от источника к стоку в транспортной сети с нечеткими пропускными способностями // Известия ТГРУ. – 2005. - №9. – С.101-108.
- 52Гецович Е.М., Засядько Д.В. Расчет потребной пропускной способности условно-кольцевых связей в транспортной сети города // Вестник ХНАДУ. – 2009. - №47. – С.116-118.
- 53 Семенов В.В. Математическое моделирование динамики транспортных потоков мегаполиса. – М., 2004. - 44 с.

54 Соколова О.Д., Юргенсон А. Н. Задачи анализа работы современных информационных сетей // Материалы Российской научной конференции с участием зарубежных исследователей «Моделирование систем информатики» МСИ-2011. - Новосибирск, 2011. – С.11-20.

55 Токтошов Г.Ы., Юргенсон А.Н. Гиперсетевой метод прокладки инженерных сетей // Труды всероссийской конференции «Математическое моделирование и вычислительно-информационные технологии в междисциплинарных научных исследованиях». - Иркутск, 2011. - С.114-115.

56 Токтошов Г.Ы., Юргенсон А.Н. Двухэтапный метод прокладки инженерных сетей // Материалы VII азиатской международной школы-семинара «Проблемы оптимизации сложных систем», серия Информатика. – Новосибирск, 2011. - С.164-169.

57 Поспелов Д.А. Нечеткие множества в моделях управления и искусственного интеллекта. - М.: Наука, 1986. – 312 с.

58 CiscoSystems. Руководство Cisco по междоменной многоадресатной маршрутизации. - М.: Вильямс, 2004. – 320 с.

59 Алексеев В.Е., Таланов В.А. Графы. Модели вычислений. Структуры данных. - Нижний Новгород: Издательство Нижегородского гос. университета, 2005. - 307 с.

60 Галкина В.А. Дискретная математика. Комбинаторная оптимизация на графах. – М.: Гелиос АРВ, 2003. - 232 с.

61 Берж К. Теория графов и её применения / под ред. И.А. Вайнштейна. - М.: Издательство иностранной литературы, 1962. - 320 с.

62 Ойстин Оре. Теория графов/ под ред. И.М.Овчинниковой. – М.: Наука, 1980. - 336 с.

63 Крылов Ю.Д. Вычислительные сети: учебное пособие. – СПб: ГУАП, 2006. – 124 с.

64 Benjaafar S. Cellular Automata for Traffic Flow Modeling // Final report, Center for Transportation Studies University of Minnesota, 1997. – P.28-36.

65 Shivaram Subramanian. Routing algorithms for dynamic, intelligent transportation networks // Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1997. – P.175-183.

66 Ахмедиярова А.Т., Иванов Н.И. О влиянии одностороннего движения на величину потока // Проблемы информатики. – 2014. - №4. – С.59-65.

67 Утепбергенов И.Т., Ахмедиярова А.Т., Шилибаева А.С. Нахождение интегрального максимального потока транспортной сети в заданном направлении // Вестник КазНПУ имени Абая. - 2015. - №4. – С.53-58.

68 Утепбергенов И.Т., Ахмедиярова А.Т., Бижанова А.С. Об одной задаче маршрутизации транспорта на городских транспортных сетях // Вестник КазАТК имени М.Тынышпаева. - 2015. - №5-6. - С.59-64.

69 Ахмедиярова А.Т., Куандыкова Д.Р., Нуркаманова М.А. Классификация алгоритмов маршрутизации // Труды Международной научно-практической конференции «Казахстанский путь – 2050: единая цель, единые интересы, единое будущее». - Алматы, 2016. – С.673-677.

70 Калимолдаев М.Н., Утепбергенов И.Т., Ахмедиярова А.Т. Об одной задаче маршрутизации транспорта в мегаполисе // Вестник КазНТУ имени К.Сатпаева. - 2016. - №1. - С.409-414.

71 Ахмедиярова А.Т., Шилибаева А.С., Буранбаева А.И. Задача анализа транспортной сети для выявления узких мест // Труды Международной научно-практической конференции «Информационные и телекоммуникационные технологии: образование, наука, практика». - Алматы, 2015. - Т.1. - С.127-131.

72 Akhmediyarova A.T., Utepbergenov I.T., Sagindykova Sh.N., Kuandykova J.R. Problem of Transport Network Analysis for Identifying Bottlenecks // Scholars Journal of Engineering and Technology (SJET). - 2016. - Vol.4, №5. - P.211-214.

73 Akhmediyarova A.T., Utepbergenov I.T. Objective of Modeling and Computation of City Electric Transportation Networks Properties // International Conference on Information Science and Management Engineering (ICISME 2015). - Thailand, 2015. – P.106-111.

74 Akhmediyarova A.T., Kassymova D.T., Utegenova A.O, Utepbergenov I.T. Development and research of the algorithm for determining the maximum flow at distribution in the network // OpenComputerScience. – 2016. – Vol.6, №1. – P. 213-218.

75 Waldemar Wójcik, Akhmediyarova A.T., Mamyrbayev O., Kassymova D.T., Utepbergenov I.T. Development and research of the algorithm for determining the maximum flow at distribution in the network // PrzeglądElektrotechniczny. – 2017. – Vol.93, Iss.6. – P.137-140

76 Ахмедиярова А.Т., Попков В.К., Буранбаева А.И. Управление перекрестком в регулярных городах с помощью сетей Петри // Сборник статей по материалам международной научно-практической конференции «Инновационная наука и современное общество». – Уфа, 2015. – Часть 2. С.3-8