

## Введение

В связи с увеличением общего количества личного транспорта дорожное движение растет во всем мире, особенно в крупных городах. В часы пик на дорогах резко повышается интенсивность движения автотранспорта из-за того, что большинство людей едут на работу практически в одно и то же время. Особенно остро они проявляются в узловых пунктах улично-дорожной сети (УДС) - *перекрестках*. Здесь возникают пробки - длинные очереди автомобилей, которые замедляют движение транспорта и, конечно же увеличивает время передвижения. Поэтому возникает необходимость в моделировании и оптимизации алгоритмов управления движением.

Начиная с 1990-х годов в развитых странах (США, Западная Европа, Япония и др.) начали разрабатывать полностью автоматизированные системы дистанционного централизованного оперативного управления дорожным движением в режиме реального времени, получившие название *интеллектуальные транспортные системы* (ИТС). *Интеллектуальные транспортные системы* (англ. *Intelligent Transport Systems*) – это системная интеграция современных информационных технологий и средств автоматизации с транспортной инфраструктурой, транспортными средствами и пользователями, направленная на повышение эффективности управления дорожным движением и обеспечение безопасности и комфортности для участников.

ИТС – это одно из наиболее перспективных способов решения проблем дорожного движения. Сейчас интеллектуальный контроль движением важная проблема.

Таким образом, актуальность исследования обусловлена необходимостью изыскание нового научного подхода к задаче повышения обоснованности и эффективности решений, принимаемых в процессе управления транспортными потоками городов на основе светофорных объектов и обеспечивающих снижение временных затрат при существующем управленческом персонале для минимизации задержек транспорта, является актуальным в научном и практическом плане.

### *Научная новизна*

Все результаты, полученные в диссертационной работе, являются новыми и актуальными. В частности, предложена модель интеллектуального агента перекрестка и алгоритм работы «зеленой волны». Также выполнена программная реализация модели агента, пригодная для исследования транспортных систем. В настоящее время отсутствует алгоритмически четкое и полное описание модели поведения агента. Многие технологии составляют коммерческую тайну их создателей и поэтому предложенный алгоритм является новым.

### *Цель и задачи работы*

Целью работы является создание имитационной модели интеллектуальной транспортной системы, обеспечивающая применение многоагентного подхода. Для достижения цели поставлены следующие задачи:

- Изучить существующие подходы к моделированию комплексных многоагентных систем;
- Изучить пакет имитационного моделирования Anylogic;
- Провести исследование и сформулировать правила поведения участников дорожного движения;
- Спроектировать и реализовать модель интеллектуальной транспортной системы и провести ряд сравнительных экспериментов.

В данной работе содержится подробное аналитическое описание модели интеллектуальной транспортной системы.

## 1.1 Транспортная система. Основные понятия

*Транспортная система* – это участок дорожной сети и сопутствующей инфраструктуры с движущимися по ней транспортными средствами – участниками движения.

*Транспортный поток (ТП)* – совокупность транспортных средств, одновременно участвующих в движении на определенном участке улично-дорожной сети (УДС).

Основные параметры состояния ТП следующие:

- скорость потока в текущий момент;
- плотность потока – количество автомобилей на единицу длины дороги;
- интенсивность потока – количество автомобилей пересекающих определенный участок дороги за единицу времени;
- состав ТП – процентное соотношение в нем транспортных средств различного типа.

*Перекрёсток* — пересечение дорог или полос движения на одном уровне.

*Дорожный затор или автомобильная пробка* — скопление на дороге транспортных средств, движущихся со средней скоростью, значительно меньшей, чем нормальная скорость для данного участка дороги. При образовании затора значительно (до 20 раз и более) снижается пропускная способность участка дороги. Если прибывающий поток транспорта превышает пропускную способность участка дороги, затор растёт лавинообразно.

*Система управления дорожным движением* - предназначена для адаптивного и координированного управления светофорами. Именно здесь собирается, обрабатывается и анализируется вся информация о трафике, на основе которой ведется управление светофорами и перекрестками.

*Задачи, которые должна решать реальная система управления городским дорожным движением:*

- *Безопасность*: не допускать аварийных ситуаций при регулировке движения
- *Пропускная способность*: повысить интенсивность движения в дорожной сети
- *Комфорт для водителей*: снижение времени ожидания на светофорах, «справедливое» распределение времени ожидания, приемлемая скорость движения

- *Особые ситуации:* переключение в режим «зеленой волны» по заданному направлению, чтобы пропустить машины пожарных служб, скорой помощи и правительственные кортежи
- *Приоритетный режим для общественного транспорта:* чтобы обеспечить движение общественного транспорта по расписанию (трамваи, троллейбусы, автобусы с вместительностью более 100 человек), требуется особый режим переключения
- *Адаптивность (адаптируемость):* Иметь режим переключения доступных полос движения. Число доступных полос движения дороги может меняться в результате погодных условий (заснеженные дороги), проведения дорожных ремонтно-строительных работ, возникновения дорожно-транспортных происшествий, парковки в неразрешенном месте и пр.
- *Надежность:* при выходе из строя управляющего центра (обрыве коммуникаций, замыкании и т.д.) должен быть предусмотрен автономный режим работы каждого элемента.
- *Диагностика:* ключевые элементы системы должны регулярно предоставлять данные о техническом состоянии. При сбое элемента должно прийти соответствующее сообщение в управляющий центр и записано в журнал лога.
- *Мониторинг:* сбор данных о текущей дорожной ситуации (интенсивность движения машин, типы автотранспорта, средняя скорость, места аварий и пр.)

## **1.2 Агентное моделирование**

Агентное моделирование - метод имитационного моделирования, представляет изучаемую систему в виде множества агентов, которые проявляют независимое индивидуальное поведение. Агентное моделирование – относительно новое (1990е-2000е гг.) направление в имитационном моделировании, используется для исследования децентрализованных систем, динамика функционирования которых определяется не глобальными правилами и законами (как в других парадигмах моделирования), а наоборот, когда эти глобальные правила и законы являются результатом индивидуальной активности членов группы. Цель агентных моделей – получить представление об этих глобальных правилах, общем поведении системы, исходя из предположений об индивидуальном, частном поведении ее отдельных активных объектов и взаимодействии этих объектов в системе. Агент – некая сущность, обладающая активностью, автономным поведением, которая может принимать решения в соответствии с некоторым набором

правил, взаимодействовать с окружением, а также самостоятельно изменяться. Автономность – одно из важных свойств агента, также отмечают обучаемость (способность к адаптации). Агент действует в некоторой среде и обладает способностью получать информацию об объектах, которые в ней находятся. Отдельно выделяют характеристики интеллектуальных агентов

1. Реактивность — способность своевременно реагировать на воспринятые изменения среды.
2. Проактивность — способность проявлять инициативу для достижения своих целей.
3. Социальные навыки — способность к взаимодействию с другими агентами «ради дела».



Сх.2 Интеллектуальный агент

### Отличия агентного моделирования от традиционного моделирования

Агентная модель	Аналитическая модель
<b>Фундаментальные отличия</b>	
Состояние агентов характеризуется с помощью аналитических переменных (отдельно взятых либо агрегированных). Агенты обладают заданным поведением. Агенты оказывают влияние на значения этих переменных посредством выполнения действий (т.е. через свое поведение).	Аналитические переменные получают значения в ходе решения системы уравнений и неравенств. Лицо, выполняющее моделирование, строит суждение о характере изменений в системе по значениям ключевых переменных, неявно предполагая соответствующее

	поведение, как целой системы, так и её составляющих. Нет явного выделения «поведения».
Агенты взаимодействуют друг с другом посредством своего поведения (т.е. оказывают влияние друг на друга либо явно, например «волк съел зайца», либо неявно через разделяемую среду, например «кони и коровы конкурируют за траву»).	Аналитические переменные связаны друг с другом с помощью уравнений и неравенств.
При агентном моделировании определяются поведения агентов с использованием аналитических переменных, характеризующих скорее отдельную часть системы, чем систему целиком. Переменные системного уровня образуются как характеристики эмерджентности системы в целом, и, как правило, не имеют управляющего воздействия.	При аналитическом моделировании аналитические переменные фокусируются, как правило, на общем описании системы (но могут отражать и характеристики отдельных частей). При таком подходе обычно довольно просто получить лаконичную замкнутую систему уравнений и неравенств.
<b>Структура модели</b>	
Единица модели – агент, обладающий поведением. Реализация поведения агента может быть скрыта (например, чтобы гарантировать коммерческую тайну компании – разработчика торгового агента). Известно лишь внешнее состояние и результат воздействий агента на среду.	Единица модели – уравнение или неравенство с переменными. Все уравнения и неравенства, как правило, открыты и известны.
Каждый агент – это отдельный процесс. Взаимодействие агентов – это взаимодействие между процессами. Поэтому на базе многоагентной модели естественным образом можно построить	Для построения распределенной системы на базе аналитической модели требуется дополнительная работа (определить связи по переменным между отдельными блоками, распараллелить участки

распределенную систему.	выполнения и пр.).
Агентные модели содержат понятие изменчивой среды и, как правило, адаптивны по построению. Поэтому задача построения автоматизированной адаптивной модели упрощается.	Для построения автоматизированной адаптивной модели как правило требуется дополнительная работа по исследованию особых случаев.
<b>Уровень системного представления</b>	
Агентные модели обычно включают модель дискретно-событийных взаимодействий. Поэтому они лучше подходят для моделирования бизнес-процессов.	Аналитические модели обычно лучше подходят для моделирования физических процессов (описание в терминах производных, частично-дифференцируемых или непрерывно-дифференцируемых функций).
Агентные модели содержат четкое разделение физической среды и пространства связей между элементами. Есть возможность задать произвольную топологию элементов системы.	Аналитические модели, как правило, сосредотачиваются на описании свойств и характеристик физической среды, при этом топология элементов учитывается неявно на системном уровне либо вовсе не учитывается.
Контроль корректности результатов происходит как на уровне отдельных агентов, так и на системном уровне.	Контроль корректности результатов происходит обычно на системном уровне.
<b>Адекватность и достоверность</b>	
При моделировании сложных нелинейных систем можно отказаться от усредненных характеристик и наделить каждую отдельную сущность характеристиками, наиболее близкими к реальным. В результате такого построения итоговая агентная модель более адекватно отражает процессы в системе по сравнению с аналитической, использующей усредненные показатели (даже с учетом дисперсии).	В некоторых (например, пространственно-временных) моделях при построении интегрально-дифференциальных уравнений часть значимых локальных свойств заменяется усредненными показателями. Таким образом, игнорируются локальные отклонения значений в подобластях, что для нелинейных систем приводит к значительному искажению получаемых результатов.
В агентных моделях возможно	В аналитических моделях, как правило,

<p>использование ограниченной рациональности агентов (т.е. когда агент не обладает полной информацией для принятия решения и ограничен по времени в принятии решения). При этом полноценная рациональность также поддерживается. На моделях с ограниченной рациональностью при моделировании социально-экономических процессов часто получаются более близкие к реальности результаты, чем на моделях с полноценной рациональностью.</p>	<p>фигурирует такая концепция, как «полная рациональность». Т.е. предполагается, что все элементы системы знают всю необходимую для принятия решений информацию и время принятия решения неограниченно. Бывает так, что равновесное решение на такой модели достигается, а на практике нет, т.к. не учитывается то, что элементы системы обладают ограниченной рациональностью, т.е. ограниченным сроком принятия решений и неполнотой информации.</p>
<p><b>Результат (получаемое решение)</b></p>	
<p>Агентные модели предлагают конструктивный способ получения решения. По сути, они предлагают историю взаимодействий агентов со средой, которая поэтапно приводит от заданных начальных условий к решению.</p>	<p>Бывает, что аналитические модели декларируют лишь существование решения, либо дают вычислительно неэффективный (экспоненциальный) алгоритм нахождения решения.</p>
<p>Нельзя достоверно судить об устойчивости получаемого решения. Однако можно косвенно исследовать устойчивость через варьирование начальных параметров и начальных условий модели.</p>	<p>Аналитические модели содержат аппарат дифференциальных уравнений, используют теорему о неявной функции для определения устойчивости получившегося решения.</p>
<p>На агентных моделях возможно получение неустойчивых неравновесных решений, и их наблюдение в динамике (например, исследование турбулентности или переходных форм экономики). При этом такие решения могут быть интересными с практической точки зрения. Например, при ламинарном течении в потоке на единицу площади приходится больше энергии, чем при турбулентом (неравновесном) течении.</p>	<p>В аналитических моделях при решении экономических задач, как правило, решения равновесны и устойчивы. Исследование неравновесных и неустойчивых состояний системы представляется проблемой.</p>



<p>Агентные модели позволяют проследить путь следования к конечной ситуации. Часто при моделировании социально-экономических процессов важно не столько конечное равновесное состояние, сколько сама системная динамика и характер изменения системных показателей на пути к решению.</p>	<p>Аналитические модели позволяют быстро оценить конечную ситуацию.</p>
<p><b>Другое</b></p>	
<p>Методы Монте-Карло и стохастические модели гармонично сочетаются с агентными моделями.</p>	<p>Есть класс задач, которые не решаются чисто аналитически. Приходится использовать стохастические модели и методы Монте-Карло.</p>
<p>Есть модельные ситуации, которые формализованы лишь частично. И порой очень сложно построить функцию зависимости глобального параметра системы от локальных взаимодействий элементов системы. Хотя интуиция подсказывает, что зависимость есть. В этом случае для исследования зависимости также могут применяться агентные модели.</p>	

Множество взаимодействующих агентов совместно со средой образует многоагентную систему (МАС).

Многоагентная система (МАС, англ. Multi-agent system) — или мультиагентные системы – это направление искусственного интеллекта, которое для решения сложной задачи или проблемы использует системы, состоящие из множества взаимодействующих агентов. МАС относится к самоорганизующимся системам [5], так как в них ищется оптимальное решение задачи без внешнего вмешательства. Главное достоинство МАС — это гибкость. Многоагентная система может быть дополнена и модифицирована без переписывания значительной части программы.

В многоагентных системах предполагается, что отдельный агент может иметь лишь частичное представление об общей задаче и способен решить лишь некоторую ее подзадачу. Поэтому для решения сколько-нибудь сложной проблемы, как правило, требуется взаимодействие агентов, которое неотделимо от организации многоагентной системы. Идея многоагентности предполагает кооперацию агентов при коллективном решении задач. В многоагентной системе агент, который не способен решить некоторую задачу самостоятельно, может обратиться к другим агентам. Другой вариант, когда необходима кооперация – это использование коллектива агентов для решения одной общей трудной задачи. При этом агенты могут строить планы действий, основываясь уже не только на своих возможностях, но и —думать о планах и намерениях других агентов.

Интеллектуальный агент — разумная сущность, наблюдающая за окружающей средой и действующая в ней, при этом поведение агента рационально в том смысле, что он способен к пониманию и его действия всегда направлены на достижение какой-либо цели [4,5]. Об интеллектуальности агента можно говорить, если он взаимодействует с окружающей средой примерно так же, как действовал бы человек.

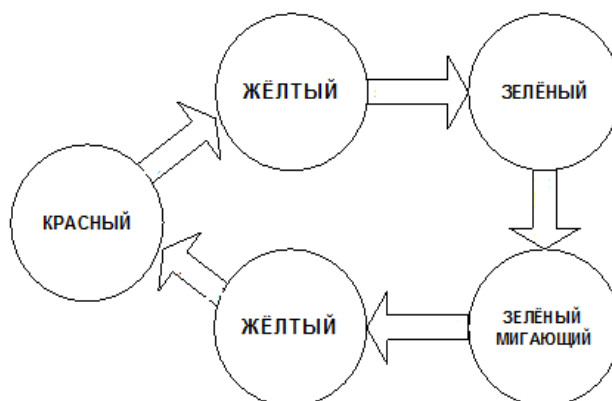
В данной работе в качестве интеллектуальных агентов рассматриваются перекрестки. Общая внешняя среда агента – проезжая часть дороги и все расположенные на ней средства организации дорожного движения.

### **1.3 Транспортная система: с обычными перекрестками и с обучающимися перекрестками (Q-Learning).**

Перекресток, являясь пересечением дорог, представляет из себя одно из самых небезопасных мест на дороге. В виду возросшего количества автотранспорта, во многих странах осталось очень мало безсветофорных перекрестков — даже самые, на первый взгляд, безобидные пересечения снабжают светофорами, дублируют приоритеты знаками и рисуют разного рода дорожную разметку. Однако, перекресток это также еще одна причина горячо "любимых" всеми пробок и затрудненного движения в городах.

Светофор представляет собой светотехническое устройство, предназначенное для последовательного включения на заданное время определенных сигналов: движение разрешено - зеленый сигнал, движение запрещено - красный и между ними промежуточный - желтый, предупреждающий о смене сигналов. Работой светофора управляет контроллер либо вычислительное устройство. Обычные светофоры могут создавать огромные пробки на дорогах, а могут, наоборот, помогать их избегать — всё зависит от того, как их настроить и как ими управлять. Обычный светофор "не глядя"

управляет движением машин, не обращая внимания на обстановку на перекрестке. Вот алгоритм работы светофора:



Сх.1 Граф состояний светофора

Наиболее распространенный способ светофорного регулирования – использование фиксированных по длительности фаз, рассчитанных на основе статистических данных. Однако данный подход испытывает значительные трудности из-за нестабильности транспортных потоков, которые меняются не только в течение дня, но и в более короткие промежутки времени.

Поэтому сейчас активно развиваются различные системы, которые позволяют адаптивно учитывать изменения в транспортных потоках. То есть, системы, которые адаптируются к изменяющимся условиям движения.

Управление светофором – сложная задача и для оптимизации работы светофора уже использовались много алгоритмов, такие как эволюционные алгоритмы, интеллектуальные алгоритмы, алгоритмы нечеткой логики и обучение с подкреплением (Q-Learning).

Q-Learning (Q-обучение) — метод, применяемый в искусственном интеллекте при агентном подходе. Относится к экспериментам вида обучение с подкреплением. Агент обучается посредством взаимодействия с окружающей средой. Это взаимодействие происходит через поощрения (наказания), получаемые агентом в ответ на предпринятые им действия. При этом агент пытается выработать стратегию поведения, которая приносит максимальную прибыль не только здесь и сейчас, но и в долгосрочной перспективе [11,12].

На основе получаемого от среды вознаграждения агент формирует функцию полезности Q, что впоследствии дает ему возможность уже не случайно выбирать стратегию поведения, а учитывать опыт предыдущего взаимодействия со средой. Одно из

преимуществ Q-обучения — то, что оно в состоянии сравнить ожидаемую полезность доступных действий, не формируя модели окружающей среды[11].

При работе алгоритма Q-Learning происходит построение функции оценки пар состояние-действие. Q-значений является многократное опробование всех возможных пар состояние-действие. Задача обучения с подкреплением в общем виде формулируется следующим образом. Для каждого переключения сигнала назначается некоторое скалярное значение, “награда”. Целью системы является нахождение политики управления, которая максимизирует сумму награды. Каждый агент обучается самостоятельно, старается минимизировать время ожидания машин на перекрестках. Каждое его действие и награда, полученная в ответ на действие, сохраняется в памяти и в следующем выборе действия учитывает опыт предыдущего взаимодействия со средой.

На каждом шаге агент распознаёт текущее состояние  $s_t \in S$  и выбирает действие  $a_t \in A$ . После этого окружающая среда возвращает агенту поощрение  $r_t = r(s_t, a_t)$  и переносит его в новое состояние  $s_{t+1} = \delta(s_t, a_t)$ . Функции  $\delta$  и  $r$  являются частью окружающей среды и не известны агенту.

$S$  – множество состояний, в которых может находиться агент;

$A$  – множество доступных агенту действий;

$t$  – дискретный шаг по времени;

Задача агента состоит в том, чтобы найти оптимальную стратегию выбора следующего действия по текущему состоянию  $\pi : S \rightarrow A$ . оптимальной называется стратегия, максимизирующая прибыль агента в соответствии со следующей формулы:

$$V^\pi(S_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$$

где  $0 \leq \gamma \leq 1$  – некоторая константа (в англоязычной литературе называемая discount factor [11]), которая позволяет в большей степени учитывать награду, полученную от окружающей среды здесь и сейчас, нежели в ближайшем будущем.

## 2. Модель с кооперативными интеллектуальными перекрестками

### 2.1. Интеллектуальный агент перекресток

*Интеллектуальная Транспортная Система*, далее ИТС (от англ. Intelligent Transport System and Services) – это системная интеграция современных информационных и коммуникационных технологий и средств автоматизации с транспортной инфраструктурой, транспортными средствами и пользователями, направленная на повышение эффективности управления дорожным движением и обеспечения безопасности и комфорта для его участников.

В интеллектуальной транспортной системе есть несколько составляющих: это то оборудование, которое устанавливается по городу (светофоры, камеры наблюдения, умные знаки, информационное табло и даже мобильные метеостанции), и многоуровневая система управления этими объектами. Некоторые представляют, что ИТС – это набор высокотехнологичных примочек, расставленных по городу. Это не совсем так. Обычные светофоры могут создавать огромные пробки на дорогах, а могут, наоборот, помогать их избежать – всё зависит от того, как их настроить и как ими управлять.

В данной работе интеллектуальная транспортная система состоит из трех компонент:

- Сенсоры - собирают информацию о дорожной ситуации;
- Интеллектуальные агенты перекрестки, которые получают информацию о дорожной ситуации с сенсоров, а также обмениваются данными с другими агентами и принимают решение на переключение светофора;
- Светофоры – используются для регулирования движения на перекрестках.



Сх.3 Компоненты интеллектуальной транспортной системы

Интеллектуальный агент-перекресток выполняет следующие действия: переключает светофоры на перекрестке, которые не создают потенциально опасных конфликтных дорожных ситуаций. Основные характеристики интеллектуального агента-перекрестка.

#### *Целеустремленное поведение*

- Агент стремится не допустить роста суммарного времени ожидания машин перед светофорами (либо сделать так, чтобы оно не достигало критических отметок);

#### *Социальные навыки*

- Агенты обмениваются друг с другом информацией о времени ожидания на светофорах и о предполагаемых действиях;
- Агенты договариваются, принимают либо отменяют взятые на себя обязательства участия в «зеленой» волне;
- Агент знает только про себя и про своих непосредственных соседей;

#### *Реактивное поведение*

- Агент безусловно пропускает ожидающие машины, если их время ожидания превысило некоторый допустимый порог (при этом неважно стоит одна машина или много машин);

#### *Любознательность (мнения агента)*

- Агент строит вероятностную модель распределения трафика по каждому направлению движения;
- Агент считает усредненные показатели ожидания по позициям, чтобы «предсказывать» свои переключения в будущем (необходимо при коллективном поведении);
- Агент имеет временную модель переключений своих соседей в некоторый момент времени  $t$ . У агента есть информация о расстояниях до соседних агентов-перекрестков.

## 2.2. Алгоритм работы «зеленой волны»

Зеленая волна – это система светофорного регулирования, в котором переключение сигналов светофоров на смежных перекрестках происходит так, чтобы водитель, подъезжая к очередному перекрестку, видел перед собой зелёный сигнал светофора. Такой метод управления светофорами широко используется во многих странах. Опыт применения «зелёной волны» показал, что её использование на магистралях более чем на 30 % повышает среднюю скорость движения, примерно настолько же снижаются задержки транспортных средств у перекрестков и на 20 % уменьшается количество наездов на пешеходов. Система получает информацию о дорожной ситуации через специальные сенсоры («радары», датчики индуктивности, видеокамеры). Но, к сожалению, отсутствует алгоритмически четкое и полное описание модели поведения агента в «зеленой волне». Многие технологии составляют коммерческую тайну их создателей.

Рассмотрим подробнее работу «зеленой волны». Для каждой конфигурации переключений перекрестка выполняется предварительная оценка организации «зеленых волн» по направлениям движения:

1. Агент-перекресток рассылает запросы на «зеленую волну» по предполагаемым направлениям продолжения движения (влево, вправо, прямо) для полос, на которых ожидается включить зеленый свет. И ждет ответа (положительного или отрицательного).

Агент «знает» сколько машин поедет по каждому направлению, благодаря статистике распределения трафика, собираемой с помощью сенсоров.

Пусть  $P_n$ ,  $P_l$ ,  $P_{\text{прямо}}$  - вероятности поехать направо, налево и прямо соответственно, вычисляются так:

$$P_n = \frac{c_n}{c}, \quad P_l = \frac{c_l}{c}, \quad P_{\text{прямо}} = \frac{c_{\text{прямо}}}{c}, \text{ где}$$

$c$  - общее количество машин, проехавшие за данный период времени  $T$ ,

$c_n$  - количество машин, повернувших направо (из количества  $c$ ),

$c_l$  - количество машин, повернувших налево (из количества  $c$ ),

$c_{\text{прямо}}$  - количество машин, проехавших прямо (из количества  $c$ ).

Агент «знает» диапазон времени, когда следует ожидать машины на следующем перекрестке. Среднее время  $t$ , которое машины затратят на движение от одного перекрестка к другому, вычисляется как отношение расстояния к средней скорости на

участке. Если агент-перекресток, получивший запрос, принимает обязательство на переключение, то он будет «держателем» зеленого света в интервале  $(t + t_0 - \varepsilon, t + t_0 + \varepsilon)$ , где  $\varepsilon$  - это некоторая заданная константа доверительного интервала,  $t_0$  - текущее время, точка отсчета. Агент «знает» общее (полное, на текущий момент) время ожидания пропускаемых машин на перекрестках. Время вычисляется как: *сумма времени ожидания на текущем перекрестке и времени ожидания на предыдущих перекрестках (информацию об этом предоставляет соседний агент-перекресток, с которого пришел трафик).*

В запросе на «на зеленую волну» содержится:

- $t + t_0$  - усредненное время, когда ожидается прибытие машин,
- $n'$  - количество прибывающих машин,  $n' = P * n$ , где  $P$  соответствующая вероятность из распределения трафика,  $n$  - общее количество движущихся машин с исходного направления,
- $wt$  - удельное общее время ожидания на перекрестках ( $wt * n = swt$  - суммарное общее время ожидания для  $n'$  машин),
- $(l_{*,e} = g)$  - требование зеленой волны (далее будет более подробно).

2. Соседний агент-перекресток, получив запрос, строит оценку функции полезности, для того чтобы определить, выгодно ли ему переключиться в период времени  $(t + t_0 - \varepsilon, t + t_0 + \varepsilon)$  на зеленый свет по требуемому направлению, или нет. Но используется для этого не фактические данные, а статистические усредненные данные, собранные за последнее время, в надежде, что, когда зеленая волна придет, дорожная ситуация будет аналогичной. Если выгодно, то он «проталкивает» запрос дальше по направлению движения и ждет ответа от соответствующих соседних перекрестков.

Более формально: Пусть  $\bar{l}_1, \bar{l}_2, \bar{l}_3, \dots, \bar{l}_k \in \bar{L}$  - возможные конфигурации неконфликтных переключений.  $\bar{l}_i = \{l_{i,1}, l_{i,2}, \dots, l_{i,m}\}$ , где  $m$  - количество светофоров на перекрестке, а  $l_{i,j}$  принимает значения из двухэлементного множества  $\{g, r\}$ . Тогда требование зеленой волны в момент времени  $t + t_0$  задается элементом  $(l_{*,e} = g)$ , где  $e$  - некоторое заданное число из диапазона  $1 \dots m$ . Конфигурация удовлетворяет требованию, если на соответствующей позиции  $e$  содержит элемент  $g$ .  $S_e(\bar{L})$  - множество конфигурации, удовлетворяющих требованию  $(l_{*,e} = g)$ .  $f': \bar{L} \rightarrow \mathbf{R}$  - предварительная оценка функции пользы (на основе статистических данных), Пример:  $f'(\bar{l}_1) > f'(\bar{l}_2)$  - по предварительным оценка конфигурация  $(\bar{l}_1)$  принесет агенту-перекрестку больше пользы, чем конфигурация  $(\bar{l}_2)$ . Тогда конфигурации-претенденты на переключение



задаются так:  $C(\bar{L}) = \arg \max_{\bar{l} \in \bar{L}} f(\bar{l})$ . Если  $C(\bar{L}) \cup S_e(\bar{L}) \neq \emptyset$ , то требование зеленой волны может быть удовлетворено (иначе возвращается false отправителю запроса). В случае непустого множества конфигураций агент-перекресток формирует запросы по исходящим направлениям конфигураций (аналогично предыдущему агенту, по указанной ранее схеме) и пересылает дальше.

Агент-перекресток ждет ответы. Виды ответов: false, true N, где N – число машин, участвующих в «зеленой волне» на последующих перекрестках (т.е. одна и та же машина считается два и более раз, если будет «зеленое» переключение на последующих перекрестках). Агент выбирает такую конфигурацию, в которой участвует максимальное количество машин. Как оценивать такую конфигурацию? Ведь n машин на соседнем перекрестке «надежнее», чем n машин на дальних перекрестках. Пусть p – вероятность того, что агент-перекресток не возьмет обязательство назад и поддержит «зеленую волну». Тогда среднее взвешенное равно:

$$n' = p * n + (1 - p) * 0 = p * n$$

Таким образом, каждый агент должен собирать статистику о поведении соседей. Но ради простоты p можно взять инерциальный коэффициент  $\gamma < 1$ .

Пусть агент получил от соседей числа  $M_1, M_2, M_3$  на итоговой конфигурации  $\bar{l}^*$ . Тогда агент возвращает **True**  $\gamma(M_1 + M_2 + M_3)$  в ответ на запрос. Пусть функция  $coop(\bar{l})$  возвращает число машин (N), участвующих в кооперативном поведении для конфигурации  $\bar{l}$ . Тогда итоговая конфигурация на предварительной оценке (для возврата результата) находится так:  $l^* = \arg \max_{\bar{l} \in C(\bar{L}) \cup S_e(\bar{L})} coop(\bar{l})$ .

### *Предварительная оценка конфигурации агентом-перекрестком*

Суммарное время ожидания для двигающихся дорог вычисляется следующим образом:

$$stmw = \sum_{\text{по всем дорогам, где } l=i'g'} \sum_{\text{по всем } i} \sum_{\text{по всем } p} \widehat{w}_{i,p}, \text{ где}$$

$\widehat{w}_{i,p}$ - усредненная статистика общего времени ожидания на полосе i и на позиции p за последнее время.

Суммарное время ожидания для ожидающих дорог:

$$sww = \sum_{\text{по всем дорогам, где } l=i'r'} \sum_{\text{по всем } i} \sum_{\text{по всем } p} \widehat{w}_{i,p}$$

Тогда предварительная эффективность переключения конфигурации  $\bar{l}$  равна:

$$f'(\bar{l}) = smw - sww.$$

Но оценка была бы неполной без запроса на «зеленую волну». Поэтому для  $l_{j=e}$  сумма считается так:

$$\max \left[ \sum_{\text{по всем } i} \sum_{\text{по всем } p} \widehat{w}_{i,p}, n' * wt \right]$$

Составляющие оценки конфигурации для принятия решения «здесь и сейчас»:

- Фактические данные о количестве машин и их времени ожидания,
- Данные о времени ожидания на предыдущих перекрестках, полученные от соседей,
- Прогнозные данные «на шаг вперед»,
- Кооперативность конфигурации (зеленые волны),
- Возможность применения конфигурации (свободно ли пространство за перекрестком?).

#### *Оценка конфигурации агентом-перекрестком*

Суммарное время ожидания для двигающихся дорог вычисляется следующим образом:

$$smw = \sum_{\text{по всем дорогам, где } l=lg'} \sum_{\text{по всем } i} \sum_{\text{по всем } p} w_{i,p}, \text{ где}$$

$w_{i,p}$  – фактическое общее время ожидания машины на полосе  $i$  и на позиции  $p$  за последнее время.

Суммарное время ожидания для ожидающих дорог:

$$sww = \sum_{\text{по всем дорогам, где } l=rr'} \sum_{\text{по всем } i} \sum_{\text{по всем } p} w_{i,p}$$

Если же в текущем интервале  $(t_0 - \varepsilon, t_0 + \varepsilon)$  возможно «зеленая волна», то формула для суммы меняется аналогично предварительной оценке. Для всех  $l$ , для которых ожидается прибытие «зеленой волны» сумма считается как:

$$\max \left[ \sum_{\text{по всем } i} \sum_{\text{по всем } p} w_{i,p}, n' * wt \right]$$

#### *Оценка на шаг вперед*

Для ожидающего направления также берется в рассмотрение время до следующего переключения:

$$sww = \sum_{\substack{\text{по всем дорогам,} \\ \text{где } l=r'}} \sum_{\text{по всем } i} \sum_{\substack{\text{по всем} \\ \text{занятым } p}} (w_{i,p} + w_0), \text{ где}$$

$w_0$  - это время до следующего момента принятия решения о переключении светофоров

#### *Отмена/Подтверждение обязательств*

При устойчивом распределении дорожного трафика выбранная конфигурация переключений на этапе предварительной оценки и выбранная конфигурация в текущий момент времени совпадают. Но если распределение трафика меняется, то текущая выбранная конфигурация может быть отличной от предварительной. В этом случае агент-перекресток «отменяет» взятые на себя обязательства перед предыдущими агентами-перекрестками: рассылает по направлению «продолжения отмененных обязательств» специальный запрос. В то же время для выбранной конфигурации он посылает подтверждающий запрос на принятие обязательств для последующих перекрестков, которые в свою очередь берут обязательства, сформированные на этапе запросов «зеленой волны».

### **3. Программная реализация транспортной системы с кооперативными интеллектуальными перекрестками.**

#### **3.1. Среда имитационного моделирования – Anylogic**

Существует большой набор платформ, подходящих для моделирования транспортных систем, и этот набор постоянно пополняется.

- VISSIM Transmodeler SIDRA Intersection – разработка немецкой фирмы PTV Systems. Позволяет строить микроскопические модели транспортных систем, состоящих из участков дороги, перекрестков любой формы, многоуровневых развязок. Основной особенностью является отображение трехмерной анимации в ходе моделирования.
- Transmodeler – разработка фирмы Caliper (США). Позволяет интегрировать модели транспортных систем с данными геоинформационных систем и предназначена в основном для моделирования движения по шоссе и сложным развязкам. Ориентирована на особенности дорожного движения в США.
- SIDRA Intersection – разработка фирмы SidraSolutions (Австралия). Позволяет моделировать поведение участников дорожного движения на перекрестках и прилегающих к ним территориях, обладает мощными средствами сбора и анализа статистики состояния модели.
- AnyLogic – разработка отечественной компании ЭксДжей Текнолоджис. Главной особенностью является возможность сочетать модели транспортной системы с другими имитационными моделями. Обладает богатыми возможностями настройки анимации.

Моделирование в VISSIM и Transmodeler предоставляет широкий набор возможностей, но часто является неоправданно сложным и недоступным специалистам в предметной области, так как предполагает глубокое знание технологий имитационного моделирования. Некоторые (SIDRA Intersection и другие) средства позволяют моделировать только отдельные типы объектов транспортной инфраструктуры.

Использование AnyLogic дает возможность оценить эффект конструкторских решений в сложных системах реального мира. Инструментальная система AnyLogic не ограничивает пользователя одной единственной парадигмой моделирования, что является характерным для существующих на рынке инструментов моделирования. В AnyLogic разработчик может гибко использовать различные уровни абстрагирования и различные

стили и концепции и смешивать их при создании одной и той же модели. AnyLogic включает в себя графический язык моделирования, а также позволяет пользователю расширять созданные модели с помощью языка Java. Графическая среда моделирования AnyLogic включает в себя следующие элементы:

- Stock & Flow Diagrams (диаграмма потоков и накопителей) применяется при разработке моделей, используя метод системной динамики.
- Statecharts (карты состояний) в основном используется в агентных моделях для определения поведения агентов. Но также часто используется в дискретно-событийном моделировании, например для симуляции машинных сбоев.
- Action charts (блок-схемы) используется для построения алгоритмов. Применяется в дискретно-событийном моделировании (маршрутизация звонков) и агентном моделировании (для логики решений агента).
- Process flowcharts (процессные диаграммы) основная конструкция, используемая для определения процессов в дискретно-событийном моделировании.

Среду моделирования транспортной системы было решено создать на платформе AnyLogic 6, приняв за основу агентный подход. Богатые возможности визуализации AnyLogic 6 обеспечивают наглядность как на этапе создания модели, так и при проведении экспериментов. Разработчики моделей имеют возможность в визуальном режиме создавать модели и сохранять их в XML-файлы. Эксперименты проводятся с помощью исполняющего модуля, отображающего анимацию по ходу моделирования. Результаты экспериментов могут быть выведены в базы данных или в файлы Microsoft Excel. Программист-имитационщик имеет возможность дополнить функциональность среды, разработав необходимые компоненты в виде подключаемых библиотек AnyLogic.

### **3.2. Разработка интеллектуальной транспортной системы**

Описанная выше формальная математическая модель была реализована с применением инструментальных средств дискретно-событийного и агентного моделирования в среде AnyLogic ([www.anylogic.com](http://www.anylogic.com)). Дискретно-событийная модель описывала дорожную сеть и транспорт, многоагентная модель описывала перекрестки и коммуникации между ними.

Дискретно-событийное моделирование является разновидностью имитационного моделирования. В дискретно-событийном моделировании функционирование системы

представляется как хронологическая последовательность событий. Событие происходит в определенный момент времени и знаменует собой изменение состояния системы. В основе данного вида моделирования лежит концепция заявок (транзактов, entities), ресурсов и потоковых диаграмм (flowcharts), определяющих потоки заявок и использование ресурсов. В соответствии с этой парадигмой поток заявок (транзакций) продвигается через flowchart, стоя в очередях, обрабатываясь, захватывая и освобождая ресурсы, разделяясь, соединяясь и, в конце концов, покидают систему.

Моделирование проводилось с использованием встроенной в *AnyLogic дорожной библиотеки*, которая включает в себя:

- Классы Road и Lane, задающие дорогу и дорожную полосу соответственно;
- Класс Car – базовый класс для всех автомобилей, являющийся подклассом класса заявки Основной библиотеки Entity;
- Шесть активных объектов, с помощью которых можно задать поток машин (RoadNetwork, CarSource, CarDispose, CarMoveTo, CarEnter, CarExit);
- Логика выбора пути с учетом предупреждений по полосам;
- Стандартный водитель, управляющий машинами с учетом ограничений скорости и логики смены полос.

Топология сети дорог задается с помощью обычных фигур презентации AnyLogic - линий и дуг. Все фигуры, задающие дороги сети, должны быть помещены в группу, и группа должна быть указана в соответствующем параметре объекта RoadNetwork, который создаст на базе этой группы фигур дорожную сеть (набор объектов класса Road и Lane).

Объект RoadNetwork задает сеть дорог, основываясь на нарисованной пользователем графике, проверяет правильность сети и отображает сеть дорог на анимации во время выполнения модели.

С объекта *CarSource* начинается диаграмма процесса дорожного трафика. *CarSource* создает машины, помещает их на одну из дорог и вставляет заявку типа *Car* в диаграмму процесса, задающую автотрафик. Времена возникновения машин в модели, задаются аналогично временам появления заявок в объекте *Source*. Машина будет помещена на дорогу только в том случае, если на этой дороге перед ней будет требуемое расстояние, свободное от машин. Иначе же машина будет храниться во внутреннем буфере объекта *CarSource*, пока такое расстояние не появится. Если количество машин в буфере достигнет его максимальной вместимости, возникнет ошибка. В нашем случае

максимальная вместимость – 1000. Далее они следуют по маршруту, задаваемому объектом *CarMoveTo*. Объект *SelectOutput* направляет входящие заявки в один из двух выходных портов. То есть, если машина успеет на зеленый свет, то она следует по маршруту, задаваемому объектом *selectOutput5* (объект Основной Библиотеки), которая направляет машины направо, налево или прямо. Иначе строится очередь заявок с помощью объекта *Queue* (объект Основной Библиотеки). Объект *hold* (объект Основной Библиотеки) блокирует/разблокирует поток заявок.

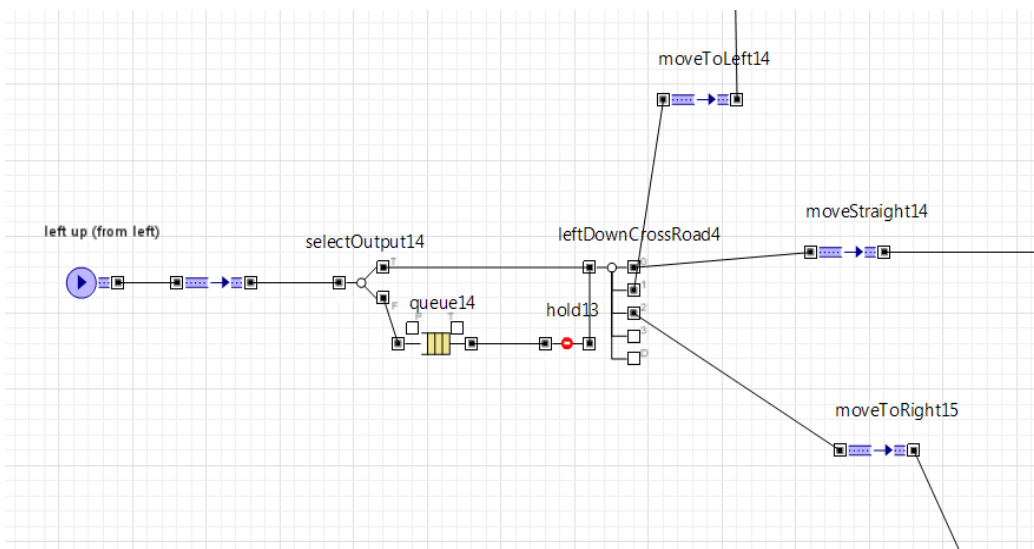


Рисунок 1. диаграмма процесса дорожного трафика

В текущей версии библиотеки еще не реализован светофор. Поэтому нами был реализован перекресток со светофорами самостоятельно. Состояние перекрестка (как набор цветов на соответствующих позициях) было представлено при помощи диаграмм состояний (statechart).

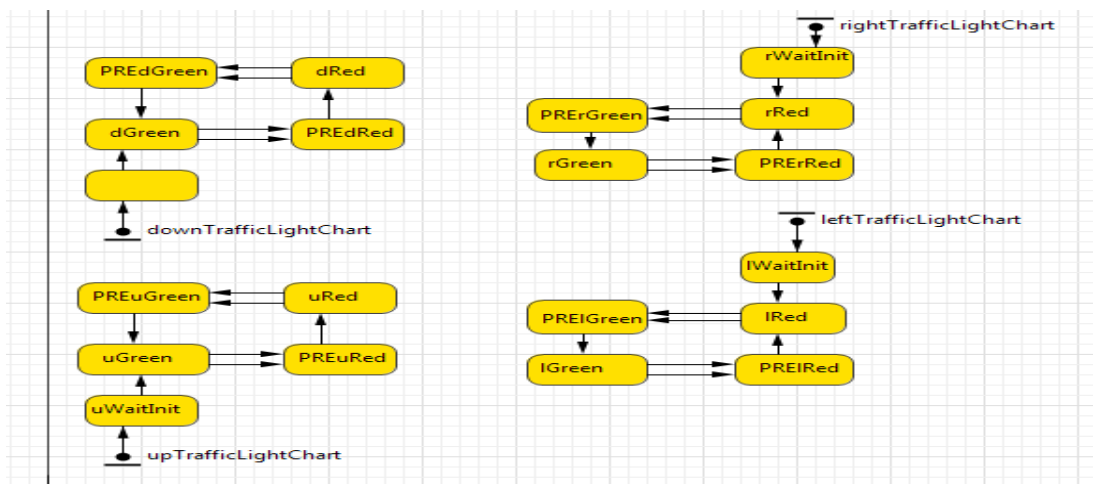


Рисунок 2. Диаграммы состояний, описывающие перекресток

Так как полноценной поддержки в Anylogic интеллектуальных агентов нет, были реализованы основные их свойства (целеустремленность, ментальная модель, реактивность и социальное поведение) с помощью функций и переменных. Такие как:

- *selectTLConfiguration* – выбор конфигурации перекрестка на основе полученной оценки;
- *evalConformedConfiguration* – оценка кооперативной конфигурации;
- *evalConfiguration* – оценка конфигурации на основе суммарного времени ожидания по полосам;
- *setTLConfiguration* - установка конфигурации для заданного перекрестка;
- *submitRequestsC* - отправить запросы на «зеленые волны»;
- *receiveRequest* – получить запрос на «зеленую волну»;
- *receiveCancel* – отказ в зеленой волне для посланного запроса;
- *receiveResponse* – получение ответа на запрос.

Относительные скорости перемещения транспортных средств и интервалы переключения светофоров были взяты на основе реальных данных (но пропорционально увеличены – для получения оперативных результатов работы модели).

Реальные перекрестки, как правило, имеют специальные режимы для пропуска пешеходов. Текущая модель их не имеет. И поэтому величина проходящего трафика через перекрестки была несколько выше (до 1,5 раз).

Для демонстрации работы алгоритмов был выбран пример с 4-мя перекрестками для сети дорог, имеющих 4 полосы (2 – в одну сторону, 2 – в другую). Перекрестки объединены в симметричную сеть (см. рис. 3).

Как показали эксперименты, наилучший эффект от многоагентного управления (по сравнению с обычным фиксированным переключением) достигается, когда в сети перекрестков присутствуют меняющиеся выделенные загруженные направления. За счет гибкого перестроения режима работы примерно в 4 раза удалось сократить среднее общее время ожидания на перекрестках по сравнению с обычными светофорными перекрестками с фиксированным временем переключения (обычные светофорные перекрестки в AnyLogic моделировались как автоматы с конечным числом состояний и фиксированным временем смены состояния). В то же время по сравнению с теми же перекрестками без кооперативного режима работы (без «зеленых волн») среднее общее время ожидания уменьшилось примерно в 3 раза.



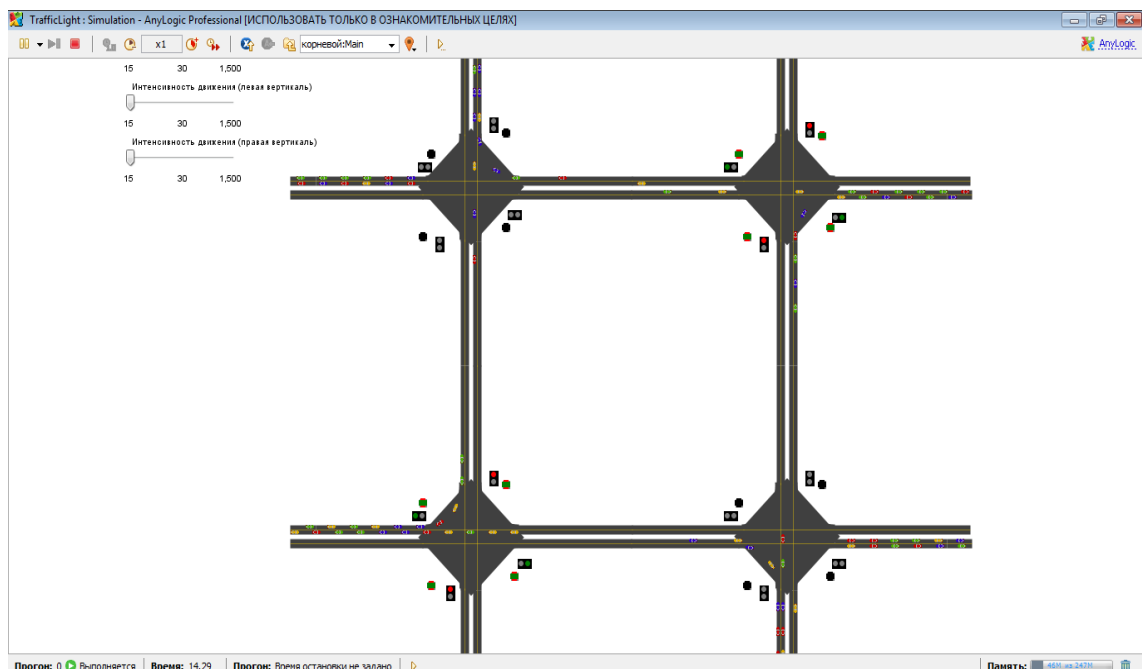


Рисунок 3. Общий вид перекрестков

При равномерной загрузке направлений выигрыш удалось получить за счет синхронизации работы светофорных объектов. Однако он не был таким значительным (максимальное сокращение общего времени ожидания до 2-3 раз, в среднем – в 1,5 раза).

Надо заметить, что на модели перекрестка с пешеходными переходами выигрыш уменьшится, но по предварительным оценкам останется существенным.

## Заключение

В результате проделанной работы была более детально изучена предметная область по транспортным системам. Был избран метод решения поставленной задачи – многоагентный подход. Исследованы основные принципы, парадигмы агентного моделирования, организации многоагентных систем. Формализованы интеллектуальные агенты и их механизмы принятия решений. В работе описана многоагентная система управления дорожным движением, обеспечивающая безопасное движение транспортных средств на перекрестках.

Рассмотрен довольно широкий круг средств разработки многоагентных систем: VISSIM, Transmodeler, SIDRA Intersection, AnyLogic. После детального анализа достоинств и недостатков этих продуктов была выбрана и изучена среда имитационного моделирования Anylogic. Модель ИТС была реализована с применением инструментальных средств дискретно-событийного и агентного моделирования в среде AnyLogic. Моделирование проводилось с использованием встроенной в AnyLogic *дорожной библиотеки*. Поскольку в текущей версии дорожной библиотеки Anylogic еще не реализован светофор, был реализован перекресток со светофорами самостоятельно. Основные свойства интеллектуальных агентов (целеустремленность, ментальная модель, реактивность и социальное поведение) также реализованы самостоятельно с помощью функций и переменных.

Внедрение кооперативного регулирования по системе «зеленая волна» создаёт ряд преимуществ по сравнению с индивидуальным регулированием на каждом перекрёстке: повышается скорость движения по магистрали, сокращаются остановки транспорта у перекрёстков. Результаты проведения экспериментов показали, что среднее общее время ожидания для одной машины снизилось в сети перекрестков, имеющих меняющиеся выделенные направления:

- В 3-4 раза для перекрестков с фиксированным строго заданным временем переключения зеленый/красный;
- В 2-3 раза для перекрестков с интеллектуальными некооперативными светофорами.

## Список литературы

1. <http://agents.felk.cvut.cz/> - Agent Technology Center
2. <http://www.anylogic.com> – AnyLogic Simulation Software
3. Рассел, Норвиг. Искусственный интеллект. Современный подход. 2-е изд. 2006 год.
4. В.Городецкий, О.Карсаев, В.Самойлов, С.Серебряков. Многоагентные системы и групповое управление. Журнал "Искусственный интеллект и принятие решений», № 2, 2009.
5. В.И. Городецкий. Самоорганизация и многоагентные системы. I. Модели многоагентной самоорганизации. Известия РАН "Теория и системы управления", 2012, № 2, с. 92–120
6. M. Wiering, J. van Veenen, J. Vreeken, and A. Koopman. Intelligent Traffic Light Control. Technical Report UU-CS-2004-029, University Utrecht, 2004.
7. M. Wiering, Multi-Agent Reinforcement Learning for Traffic Light Control, in Proc. 17th International Conf. on Machine Learning, pg. 1151-1158, 2000.
8. David Sislak and Premysl Volf and Michal Pechoucek: Agent-Based Cooperative Decentralized Airplane Collision Avoidance. IEEE Transactions on Intelligent Transportation Systems. 2011, vol. 12, p. 36-46.
9. Camurri M., Mamei M. and Zambonelli F. Urban Traffic Control with Co-Fields // Series Lecture Notes in Artificial Intelligence. V. 4389/ Eds. D.Weyns, H.V.D.Parunak and F.Michel Springer, 2007. P. 239–253.
10. Reinforcement Learning of Traffic Light Controllers under Partial Observability MSc Thesis of. Artificial Intelligence, (0010774). Kuyer, L., Whiteson, S., Bakker, B.
11. Mitchel T. Machine Learning. McGraw-Hill Education (ISE Editions). 1997.
12. Sutton R., Barto A. Reinforcement Learning: An Introduction. MIT Press. Cambridge, MA. 1998