

Учреждение Российской академии наук
Институт систем информатики им. А.П. Ершова
Сибирского отделения РАН

На правах рукописи

УДК 004.048/.421/.89;
330.322.7; 51-77

Зайцев Иван Дмитриевич

**МНОГОАГЕНТНЫЕ СИСТЕМЫ В МОДЕЛИРОВАНИИ
СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ ОТНОШЕНИЙ:
ИССЛЕДОВАНИЕ ПОВЕДЕНИЯ И ВЕРИФИКАЦИЯ
СВОЙСТВ С ПОМОЩЬЮ ЦЕПЕЙ МАРКОВА**

Специальность 05.13.10 – Управление в социальных и экономических
системах

Диссертация на соискание ученой степени
кандидата технических наук

Научные руководители:

ведущий научный сотрудник ИЭОПП СО РАН,
кандидат экономических наук
Есикова Татьяна Николаевна,
заместитель директора ИСИ СО РАН,
кандидат физико-математических наук
Мурзин Фёдор Александрович

Новосибирск–2014

Содержание

Введение.....	4
1 Мультиагентные системы. Обзор. Формализация	11
1.1 Основные термины и принципы построения	11
1.2 Области применения и примеры	15
1.3 Методы теоретического анализа	23
1.4 Программное обеспечение. Агентные платформы.....	25
1.5 Агентное моделирование. Системы с роевым интеллектом	32
1.6 Формальное описание.....	36
2 Исследование семейства МАС с помощью цепей Маркова.....	43
2.1 Проведение аналогии с цепями Маркова	43
2.2 Нахождение эргодического распределения и анализ поведения МАС.....	46
2.3 Описание специализированной агентной платформы MASwarm.....	52
2.4 Примеры использования агентной платформы MASwarm	58
2.5 Сравнение с другими агентными платформами	64
3 Модель развития международной транспортной сети.....	68
3.1 Постановка задачи	68
3.2 Описание модели	71
3.3 Описание программного комплекса.....	81
3.4 Программный комплекс для анализа и визуализации результатов моделирования	87
3.5 Моделирование развития Северного морского пути	96

3.6 Моделирование развития Международных Транспортных коридоров Евразии	104
Заключение	111
Литература	114
Приложение А. Доказательства утверждений о поведении некоторых МАС	123
1 Модель Шеллинга	123
2 Игра с логлинейным правилом выбора стратегии игроками	131
3 Локальный поиск. Алгоритм имитации отжига.....	137
Приложение Б. Свидетельство о регистрации программы для ЭВМ..	140
Приложение В. Акты о внедрении	141

Введение

Актуальность работы

Многоагентные системы (МАС) в последнее время являются одной из важных и перспективных областей развития информационных и коммуникационных технологий. Это обусловлено всё возрастающей сложностью, пространственной распределённостью современных информационных систем, организаций, исследуемых объектов. Агентный подход находит применение всюду, где монолитное, строго иерархическое представление сталкивается с теми или иными проблемами. Например, всё более актуальным становится построение многоагентных ERP-систем для управления распределёнными производственными процессами (например, в области логистики примечательны работы Самарской научной школы мультиагентных систем, П.О. Скобелева, Г.А. Ржевского). Применяются МАС и при решении задач искусственного интеллекта (ИИ), поскольку позволяют разделить большую проблему на составные части и решить её, используя коллективный интеллект.

В экономическом моделировании ранее преобладало оптимизационное моделирование, предполагавшее жёсткое подчинение отдельных экономических субъектов общей воле, выраженной неким центром, но сейчас всё большее распространение получают децентрализованные модели, учитывающие разнонаправленность интересов экономических субъектов и эгоистичность их поведения. Это связано как с процессами либерализации экономики ряда стран, так и децентрализацией международных связей, распадом системы двух противоборствующих блоков. Построению МАС в описанных областях посвящены работы А.Б. Шабунина, Н.А. Кузнецова, П.О. Скобелева, В. А. Виттиха, Katia Sycara, Uri Wilensky, Petter Holme, Andreas Grönlund.

Параллельно с практическим внедрением идёт теоретический анализ принципов многоагентных систем. Поскольку МАС применяются во многих различных областях, существуют различные подходы к исследованию МАС в зависимости от вопросов, стоящих перед исследователями. Большой популярностью пользуется логический подход к описанию и исследованию МАС, заключающийся в описании их работы с помощью различных логических конструкций, введения специальных операторов для описания работы агентов. Также применяется вероятностный подход, при котором некоторые элементарные шаги работы системы рассматриваются как подверженные действию случайных факторов (например, задержка при передаче информации, или индивидуальные решения агентов).

Перспективным методом изучения является выделение некоторого семейства МАС и дальнейшее исследование его свойств. В ряде частных случаев удаётся построить математическую модель многоагентной системы, исследовать её свойства и обосновать эффективность или оценить результаты её работы. Теоретические исследования МАС, их семейств, отдельных систем представлены в работах П.Р.Коэна, Г.Дж.Левескью, В.Б. Тарасова, М.К. Валиева, М.И. Дехтяря, А.Я. Диковского, Nikos Vlassis, Junfu Zhang.

Однако многие модели, построенные с помощью агентного подхода, являются чисто имитационными, а алгоритмы (если речь идёт о системах ИИ) – чисто эвристическими. Задача теоретического обоснования их эффективности и оценки их поведения остаётся нерешённой. Всё это делает актуальным дальнейшее теоретическое изучение МАС, создание новых подходов, выделение новых семейств, а также развития средств их описания, исследования и реализации, выведения приёмов и методов, как обосновывающих использование существующих систем, так и задающих направление их улучшения и создания новых систем, обладающих определёнными свойствами.

Цель работы.

Целью работы являлось исследование МАС, применяемых при моделировании социально-экономических отношений, развитие математического аппарата их описания и оценки поведения, а также программных средств их реализации. Для этого были поставлены следующие задачи.

1. Исследование различных многоагентных систем на примере ряда экономических моделей и мультиагентных реализаций оптимизационных алгоритмов. Выделение общих методов построения и вопросов, касающихся их поведения, критериев качества работы систем.

2. Определение закономерностей, взаимосвязей между интересующими нас свойствами и параметрами, используемыми при задании МАС и их доказательство; применение полученных закономерностей для доказательства утверждений о свойствах существующих систем и построения новых.

3. Разработка специализированной программной библиотеки, предоставляющей средства построения, управления процессом работы и анализа многоагентных систем предложенного семейства.

4. Проектирование новых моделей для макроэкономических исследований с использованием агентного подхода, создание модели развития транспортной сети, учитывающей современные принципы международных отношений, конкуренцию между странами и транснациональными операторами перевозок.

5. Создание основанного на предложенной модели и разработанной программной библиотеке программного комплекса для прогнозирования и оценки крупных международных грузоперевозок, развития транспортной сети, а также для анализа, визуализации и публикации результатов моделирования.

Методы исследования

При исследовании и построении новых моделей применены общие методы агентного моделирования, основанные на дискретном подходе к описанию работы многоагентных систем, а также методы имитационного компьютерного моделирования.

Для основной теоретической части работы, построения математической модели семейства мультиагентных систем и доказательства её свойств, использовались методы теории вероятности, случайные процессы и цепи Маркова, в частности, эргодическая теорема.

Кроме этого, при исследовании различных систем применялись элементы теории игр (принципы выбора решения, потенциальная функция, оптимальность по Парето и Нэшу), теории кодирования (покрывающий код), эвристические оптимизационные алгоритмы (алгоритм имитации отжига, генетический алгоритм).

Научная новизна

1. Предложено семейство МАС, включающее в себя различные важные системы управления и моделирования социально-экономических процессов. Проведена формализация основных действий, доказаны взаимосвязи между различными характеристиками систем.

2. Предложена агентная модель развития международной транспортной сети, учитывающая разнонаправленность интересов и конкуренцию участников и позволяющая прогнозировать загруженность различных международных транспортных коридоров.

3. Построен ряд альтернативных, более наглядных доказательств известных утверждений о свойствах работы различных систем и алгоритмов (например, модели Шеллинга и алгоритма имитации отжига) с использованием выведенных в работе общих утверждений.

Основные результаты, выносимые на защиту.

1. Математическая модель поведения семейства МАС с роевым интеллектом, основанная на цепях Маркова.
2. Утверждение о взаимосвязи стационарного распределения цепи Маркова со значением характеристической функции системы.
3. Точное выражение и приближённые оценки распределения вероятности значений характеристической функции системы.
4. Альтернативные доказательства свойств различных мультиагентных моделей социально-экономических отношений и оптимизационных алгоритмов.
5. Агентная платформа, воплощающая выведенную вычислительную модель поведения МАС и предоставляющая средства для их построения и исследования.
6. Агентная модель развития международной транспортной сети с учётом разнонаправленности интересов экономических субъектов и основанный на ней программный комплекс.

Практическая значимость

В процессе исследования был спроектирован и реализован программный комплекс для имитационного моделирования процесса развития транспортной сети в условиях изменяющихся грузопотоков и конкуренции между операторами перевозок, анализа и визуализации результатов моделирования. В настоящее время он используется в Институте экономики и организации промышленного производства (ИЭиОПП) СО РАН для решения макроэкономических задач анализа и прогнозирования.

Реализована библиотека программ для исследования мультиагентных систем с роевым интеллектом и построения новых систем с заданными свойствами, содержащая инструменты управления процессом и мониторинга значения важных показателей её работы. На примере реализации модели

Шеллинга показано её преимущество в данной области по сравнению с общими, неспециализированными агентными платформами NetLogo и Jade.

Доказанные в работе утверждения могут использоваться при обосновании утверждений о поведении различных МАС, что проиллюстрировано в главе 4. В частности, речь идёт о доказательстве эффективности методов оптимизации и верификации моделей общественно-экономических отношений.

Апробация работы

Основные результаты работы докладывались автором на ряде конференций: Международная научная конференция «Управление развитием крупномасштабных систем» (г. Москва, 2010 г., 2011 г., 2013 г.); Международная научная студенческая конференция «Студент и научно-технический прогресс» (г. Новосибирск, 2009 г., 2013 г.); Всероссийская научная конференция «Россия 2030 глазами молодых ученых» (Москва, 2011 г., 2013 г.)

Публикации

По теме диссертации автором опубликовано 13 работ, из них 3 статьи опубликованы в журналах из списка ВАК, 10 работ – в трудах и материалах международных конференций. Получено свидетельство о государственной регистрации программы для ЭВМ.

Личный вклад автора

Основные результаты, математические доказательства и программные библиотеки построены автором лично, создание экономической модели и отладка программы осуществлялась на основе консультаций с сотрудниками ИЭиОПП СО РАН.

Структура и объем диссертации

Диссертация состоит из оглавления, введения, 3 глав, заключения и приложения. Объем – 142 страницы. Работа содержит 18 рисунков. Общее число использованных источников: 73.

1 Мультиагентные системы. Обзор. Формализация

В данной главе даётся обзор области исследований, описываются основные понятия, связанные с мультиагентными системами, их свойства, примеры различных систем. Кратко описываются связанные с МАС теории и средства программной реализации МАС, выделяется семейство МАС, изучение свойств которого предлагается в данной работе. Обосновывается его значимость и общность схемы построения для многих важных МАС. Формализуется схема организации МАС.

1.1 Основные термины и принципы построения

Мультиагентные системы — это системы, состоящие из автономных интеллектуальных агентов, взаимодействующих друг с другом и пассивной среды, в которой агенты существуют и на которую также могут влиять. В первую очередь речь идёт о программных системах или моделях, описывающих процесс их работы, их поведение. Тогда агентом является программный агент (то есть, исполняемая программа с особыми характеристиками) или абстрактный интеллектуальный агент, например, экономический агент (то есть, модель, формализованное описание действующего лица реальной системы). Впрочем, вообще говоря, в качестве агентов могут выступать и другие объекты, например, роботы.

Пожалуй, наиболее широкое определение агента (как «любой сущности, которая находится в некоторой среде, воспринимает ее посредством сенсоров, получая данные, которые отражают события, происходящие в среде, интерпретирует эти данные и действует на среду посредством эффекторов») дано в одной из классических книг по агентам и агентным системам – «Искусственный Интеллект: современный подход» П. Рассела и С. Норвига [34]. Тем не менее, на протяжении практически всей книги под агентам подразумевается некоторая интеллектуальная система.

Лишь мельком отмечено, что простые реактивные агенты находили применение как модель для психологических бихевиористов.

Однако существует довольно широкий класс систем, составленных из многих агентов, в которых агенты обладают весьма ограниченной интеллектуальностью. Для того, чтобы охватить и такие системы и агенты мы будем использовать концепцию, предложенную В.Б. Тарасовым в [42] (и описанную подробно в [38]). В этой статье рассмотрен целый ряд различных определений агентов, произведено их разделение на «слабые» (в которых агент практически равен любой вычислительной единице с некоторой программой, обладающей теми или иными средствами коммуникации с другими агентами) и «сильные» (согласно которым агент представляет собой сложную интеллектуальную систему, обладающую такими антропоморфными свойствами как «убеждения», «настойчивость» и т.п.[67]).

В нашей работе рассматривается наиболее широкий круг МАС, то есть, модели, формализованные описания процессов и их программные реализации особого вида. Под особым видом понимается их представление как системы, состоящей из агентов и среды. При этом поведение системы в целом нигде не детерминируется, а определяется в процессе работы из поведения составляющих её агентов. Иными словами, агенты в процессе работы МАС организуются в систему и формируют её поведение не под воздействием директив «сверху», а основываясь на собственных интересах и правилах поведения. При этом мы не ограничиваем заранее понятие агента. Таким образом, система выстраивается не «сверху вниз», а «снизу вверх», не довлеет над интересами составляющих её агентов, но формируется с их обязательным учётом.

Данный подход к построению программных систем и моделей в последнее время приобретает всё большую популярность. В программировании он является естественным развитием объектного подхода с использованием идей параллельных и распределённых вычислений.

Действительно, реализация программных агентов во многих системах заключается в том, что агенты представляют собой объекты, с которыми связаны отдельные процессы или нити (threads) исполнения. Основным отличием агентного подхода от объектного и объектно-ориентированного является наличие у агентов активного начала. Среду же, в которой существуют агенты, можно представить в виде системы пассивных сущностей (например, объектов) подвергающихся влиянию агентов.

Поскольку мультиагентные системы находят применение в самых различных областях, то связанная с ними терминология довольно абстрактна и конкретное значение ключевых терминов может изменяться в зависимости от рассматриваемой области. В информационных технологиях под агентом чаще всего понимают программный агент или интеллектуальный агент – компонент программной системы, обладающий определёнными признаками автономности и интеллектуальности. В общественных науках (в первую очередь, экономике) под агентом подразумевают экономический или рациональный агент – абстрактное представление участника экономической системы, описание его целей, свойств и т.п. Таким образом, может произойти смешение понятий, относящихся к абстрактному описанию, модели некоторого процесса и понятий, относящихся к конкретной реализации, собственно к программной системе. Поэтому мы будем стараться там, где возможна путаница, оговаривать особо, какую группу понятий мы рассматриваем в данный момент.

Мультиагентную систему можно представить как набор активных, действующих агентов и пассивную среду. Каждый агент имеет свои представления о внешнем мире, текущее состояние, цели и логику, определяющие его поведение. Также агент может общаться с другими агентами в процессе работы. Деятельность агента заключается в сборе и обработке информации, принятии решения и соответствующем ему

воздействию на среду. Части агента, отвечающие за сбор информации, называются сенсорами, за воздействие на среду – эффекторами.

Схема, представляющая примерное внутреннее устройство агента, а также его положение в среде и взаимодействие с другими агентами, приведена на Рисунке 1.1.

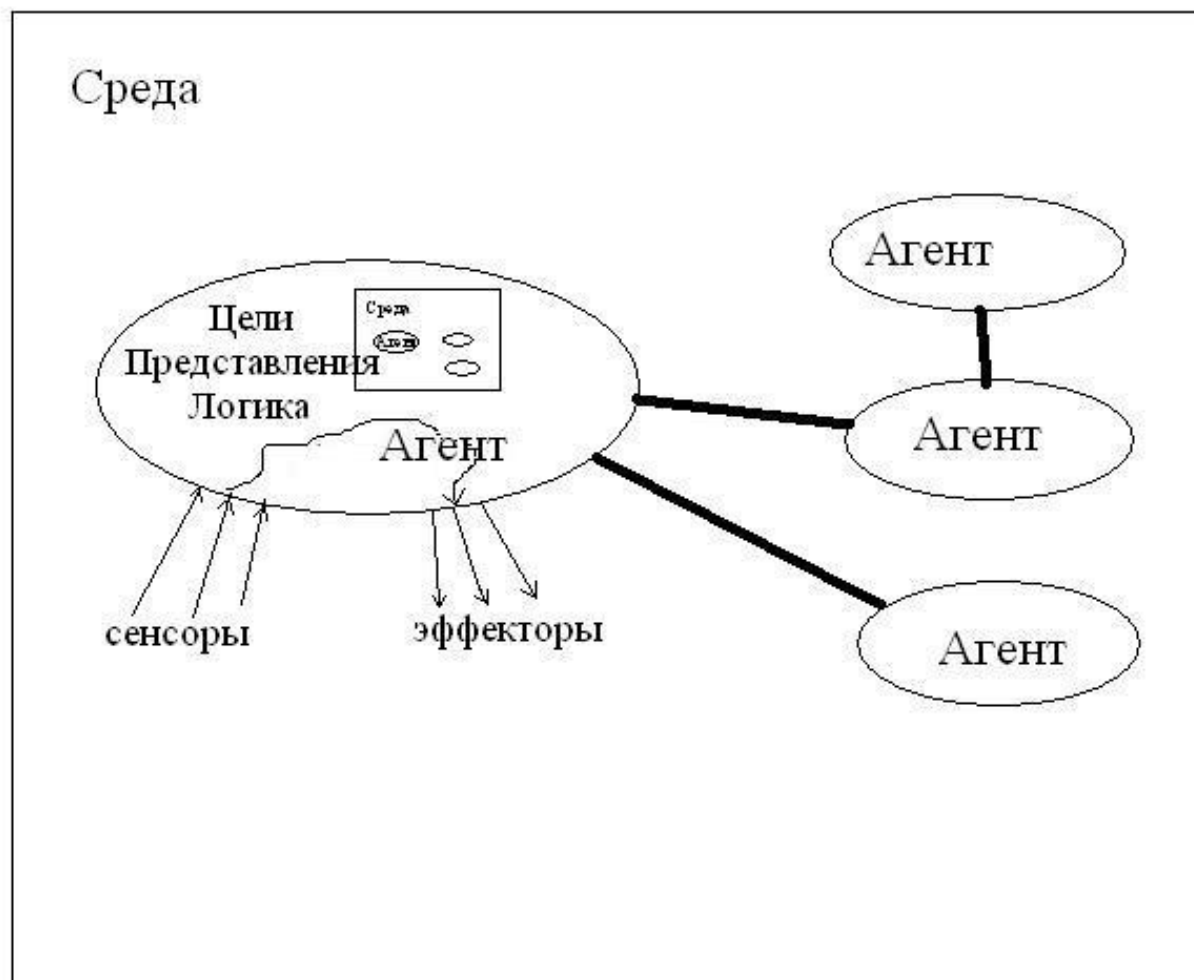


Рисунок 1.1 – Внутреннее устройство агента

Однако описанные понятия весьма отличаются в зависимости от рассматриваемой МАС. В распределённых системах может уделяться большое значение коммуникациям между агентами, а в моделях зачастую коммуникация может быть сведена к минимуму и главным аспектом деятельности агентов может быть взаимодействие со средой. Агенты могут как действовать независимо, так и конфликтовать за ресурсы, используя

общение для разрешения конфликта. Сенсоры и эффекторы могут быть определены явно (например, если агент представлен роботом), а могут только подразумеваться. То же самое касается и внутреннего устройства агента. Представление, цели и логика могут быть заданы как непосредственно, так и опосредованно, например, через алгоритм поведения агента в зависимости от получаемых сенсорами и анализируемых параметров среды, а также сведений от других агентов.

В общем, на иллюстрации представлена лишь абстрактная схема МАС, поскольку на данном этапе мы рассматриваем определение мультиагентной системы в наиболее общем виде. Перейдём теперь к более конкретным примерам, чтобы в дальнейшем подробнее проанализировать и точнее определить интересующие нас термины.

1.2 Области применения и примеры

Круг применения мультиагентных систем очень широк. Программные системы, составленные с использованием агентного подхода, применяются в автоматизации многих распределённых процессов. Наиболее распространены мультиагентные системы в логистике, хотя и другие процессы функционирования предприятия могут быть автоматизированы с помощью МАС. В первую очередь, конечно, речь идет об управлении процессами с несколькими разнесёнными в пространстве участниками, установлении координации в их работе. Например, известно применение МАС в качестве систем регулирования городского движения. Также МАС широко применяются в различных системах дистанционного обучения и интеграции знаний.

Ещё одной важной областью применения МАС является моделирование сложных процессов со многими участниками в различных науках, в первую очередь, социальных, где построить строгую математическую модель исследуемого процесса удаётся далеко не всегда.

Известны мультиагентные системы моделирования международных отношений, торговли, военных действий.

Интересным перспективным направлением применения МАС является робототехника. Многие общие элементы, методы МАС (например, протокол взаимодействия FIPA ACL) изначально нацелены на использование физических агентов, роботов. Современные автоматизированные бытовые системы всё ещё сильно зависят от человека и покрывают лишь небольшую определённую область человеческой деятельности. Дальнейшее их усложнение чревато, как правило, понижением стабильности и надёжности их работы. В этом свете интересным представляется путь к усложнению автоматизированных систем через взаимодействие, общение составляющих их автономных участников.

Резюмируя, можно сказать, что МАС находят применение везде, где монолитное представление системы по той или иной причине сталкивается с серьёзными проблемами. А такие проблемы возникают в последнее время всё чаще. Примером могут служить всё усложняющиеся системы управления производством, операционные системы вычислительных устройств (где аналогом агентов являются службы (services) в Windows или процессы-демоны (daemons) в Linux), области исследования общественных наук.

Важным для нас примером является развитие архитектуры вычислительных устройств. Ещё в 1980-х оно столкнулось с проблемой увеличения вычислительной мощности в рамках монолитного вычислителя, в связи с чем высокопроизводительные системы начали строиться с использованием параллелизма и распределённости. Со временем эти принципы всё больше стали проникать из специальных высокопроизводительных вычислительных устройств в производственные и простые пользовательские вычислительные устройства. Наглядной иллюстрацией этого распространения может служить ставшее в последнее

время уже обычным использование многоядерных процессоров в таких массовых устройствах, как мобильные телефоны.

Такой путь развития аппаратного обеспечения вычислений ставит соответствующие условия и для обеспечения программного. В программных системах всё чаще используется разделение на параллельные процессы и распределённые вычисления. Ярким примером этого могут служить ставшие популярными в последнее время облачные сервисы. В таком случае МАС можно рассматривать как дальнейшее развитие объектно-ориентированной парадигмы в условиях параллельности и распределённости вычислительных процессов.

Как мы показали выше, существует очень широкий спектр различных систем, называемых мультиагентными. Но для дальнейшего анализа этого многообразия полезно ввести на нём классификацию, разделяющую его по определённым признакам. Основным отличием между системами мы будем считать интеллектуальность составляющих её агентов, развитость их представлений о внешнем мире. По этому признаку в [42] все агенты разделяются на **когнитивные** (или интеллектуальные, обладающие развитым представлением о внешнем мире и действующие на основе его анализа) и **реактивные** (действующие на основе заложенных в них правил). Два этих типа агентов порождают два существенно отличающихся друг от друга типа МАС, два ключевых направления в исследовании агентов и составленных из них систем. Это **Распределённый искусственный интеллект (РИИ)** и **Искусственная жизнь (ИЖ)**. Системы РИИ составлены, как правило, из небольшого числа интеллектуальных агентов. Системы ИЖ, напротив, составлены из большого числа реактивных агентов.

Итак, можно заметить, что интеллектуальность каждого отдельного агента находится в обратной зависимости от количества агентов, составляющих систему и разделить по этому признаку мультиагентные системы на две большие группы, как показано на Рисунке 1.2.



Рисунок 1.2 – Принцип разбиения МАС на группы

Первую группу составляют системы, построенные с использованием небольшого (не более десятков) количества высокоинтеллектуальных агентов. Агенты в таких системах, как правило, сами представляют собой сложные программные системы или даже системы искусственного интеллекта. Так, каждый агент может содержать в себе экспертную систему, или обучающуюся нейронную сеть и т.п. Внешние характеристики агента в этих системах лучше всего описывать такими абстрактными понятиями как цели, верования («beliefs»), представления об окружающем мире и т.п. Разумеется, в конкретной реализации эти понятия принимают чёткое значение. Основой интеллектуальности системы в данном случае, является

интеллектуальность составляющих её агентов, сама же система главным образом отвечает за распределение задачи по исполнителям и обеспечивает общение между агентами и их взаимодействие со средой. Такие системы возникают в рамках построения и исследования Распределённого искусственного интеллекта (РИИ). Важнейшей отраслью применения систем РИИ является распределённое решение сложных задач. При этом интеллектуальность системы базируется на интеллектуальности (способности решать сложные задачи) агентов в отдельности.

Вторую группу составляют системы, построенные с использованием относительно большого количества (от десятков до тысяч) агентов, имеющих простую внутреннюю структуру и не обладающие сложным поведением. Как правило, их поведение определяется простыми реактивными правилами, то есть, агенты следят за изменением определённых внешних параметров и под влиянием их изменения изменяют некоторые другие параметры среды и своё состояние. Эта реакция может иметь вероятностный характер, имитирующий контринтуитивность и сложное поведение действительно интеллектуальных агентов. Несмотря на то, что отдельные агенты не представляют собой систем искусственного интеллекта, составленная из десятков-сотен таких агентов система в целом может проявлять так называемый «роевой интеллект». Это достигается благодаря большому числу агентов и вероятностной природе их поведения.

В [38] Направление теории МАС, изучающее такие системы, называется Искусственной Жизнью (в широком смысле этого слова). Однако, как отмечается, такие системы могут находить применение не только в моделировании естественных процессов, но и в других областях, например, в эвристических алгоритмах. При этом их ключевой особенностью остаётся проявление сложного поведения в результате совместных действий большого числа простых реактивных агентов, так называемого «роевого интеллекта».

Поэтому мы также будем называть такие системы мультиагентными системами с роевым интеллектом.

Соответственно, их применение имеет большое и разностороннее значение в управлении социально-экономическими системами. С одной стороны, возможно применение подобных МАС в качестве моделей сложных систем, одной из отличительных особенностей которых является самоорганизация. [6,43] С другой стороны, способность таких систем решать сложные задачи может быть использована при частных задачах нахождения оптимума сложных функций, возникающих во время поиска оптимального решения управленческих задач.

Необходимо заметить, что выделение двух групп МАС не означает, что между этими группами имеется чёткая, единственно верным образом определённая граница. Однако указанный принцип разделения помогает отнести однозначным образом к той или иной группе многие системы и определить круг проблем и методов их решения, связанных с той или иной группой.

Системы как первой, так и второй группы сейчас играют всё более важную роль в развитии информационных технологий и, в частности, искусственного интеллекта. Например, системы первой группы используются для решения производственных, в частности, логистических задач [50], при реализации систем дистанционного обучения [29], моделировании военных действий [31], второй – для задач оптимизации, например в мета-эвристических алгоритмах муравьиной колонии или роя частиц, при построении различных имитационных моделей в общественных науках (социологии, психологии, экономике) [53,55], естественных науках (экологии, биологии).

Рассмотрим подробнее такую важную область применения МАС, как моделирование. В моделировании системы первого типа применяются там, где надо рассмотреть процесс взаимодействия небольшого количества

игроков, обладающих сложным интеллектуальным поведением, например, в моделировании военных конфликтов нескольких стран.

Системы второго типа применяются при моделировании общественных (например, экономических) процессов, когда нужно учитывать поведение большого числа участников (людей, продавцов, покупателей), которые формируют своей деятельностью общую картину. Ещё одна развивающаяся область применения систем второго типа – исследование операций и эвристические алгоритмы оптимизации. Действительно, различные модели естественных процессов (биологических, физических, экологических, социальных, экономических) в ходе своей работы иллюстрируют способность моделируемых систем находить всевозможные оптимальные решения, природные «ниши» и т.п. Зачастую становится возможным применение этих способностей при решении сложных оптимизационных задач, для которых не существует эффективных классических алгоритмов.

Например, предположим, что у нас есть относительно достоверная модель поведения колонии муравьёв (базирующаяся на сравнительно простой модели поведения каждого отдельного муравья). Известно, что, несмотря на ограниченность картины мира у каждого муравья в отдельности, колония в целом справляется с такой сложной задачей, как поиск оптимального пути в сложных условиях. Следовательно, и наша модель справляется с той же задачей. В качестве среды мы можем использовать, например, граф некоторой задачи коммивояжёра (модифицированный применительно к условиям нашей модели). Тогда мы сможем найти оптимальное решение (или приблизиться к нему), используя нашу модель. Так агентная модель превращается в интеллектуальную систему, определяет эвристический алгоритм поиска оптимального решения сложной задачи.

Модель муравьиной колонии иллюстрирует способность муравьёв находить кратчайший путь к цели и может быть использована для определения кратчайшего гамильтонова цикла во взвешенном графе для

задачи коммивояжёра. Модель развития во времени генофонда популяции используется для нахождения оптимума сложной функции в генетическом алгоритме. Этот перечень аналогий можно продолжить и далее [64]. Таким образом, многим моделям можно поставить в соответствие эвристический оптимизационный алгоритм, основанный на тех же принципах. Примеры таких сопоставлений приведены в Таблице 1.1.

Таблица 1.1 – Примеры сопоставления эвристических алгоритмов и агентных моделей

Модель	Алгоритм
Движение атомов при охлаждении металла	Алгоритм имитации отжига
Движение муравьёв в колонии	Муравьиный алгоритм (алгоритм оптимизации подражанием муравьиной колонии, англ. ant colony optimization, ACO)
Приспособление генофонда популяции к условиям среды	Генетический алгоритм
Социальное поведение	Метод роя частиц

Полученные МАС, в свою очередь, могут быть использованы при построении интеллектуальных агентов для систем РИИ (например, в [34] генетический алгоритм рассматривается не как МАС, а как возможная составная часть интеллектуального агента)

Теоретическое обоснование эффективности таких алгоритмов, как правило, является отдельным и сложным вопросом, и не всегда возможно. А там, где возможно, имеет вероятностный характер, исчезающий лишь в пределе (например, доказательство эффективности алгоритма имитации отжига в [22]). Несомненно, такое обоснование является одной из важнейших

проблем, которую можно представить в рамках теории мультиагентных систем.

1.3 Методы теоретического анализа

Поскольку понятия, связанные с мультиагентными системами, могут рассматриваться с весьма различных сторон в зависимости от предметной области и типа системы, то и теории, связанные с мультиагентными системами отличает разнообразие, различие как в подходах, так и в исследуемых свойствах МАС.

Довольно развит логический подход к описанию деятельности отдельных агентов и системы агентов в целом. В этой области можно выделить интенционистскую логику Коэна-Левескю, описывающую поведение агента через его внутренние свойства, такие как цели, знания и т.п. Вообще для логического подхода к анализу МАС свойственно использование логики первого порядка дополненной средствами, характеризующими изменчивость и релятивизм определения истины. Например, в работе [47] рассматривается процесс договоров агентов между собой с логической точки зрения и используются следующие выражения:

$$\begin{aligned}\Diamond \varphi(t) &\equiv \exists t' | t' > t \wedge \varphi(t') \\ _ \varphi(t) &\equiv \forall t' | t' > t \Rightarrow \varphi(t').\end{aligned}\tag{1.1}$$

То есть, формализованы понятия того, что формула истинна хотя бы когда-нибудь и всегда. В дальнейшем вводятся операторы, характеризующие то, что агенты или группа агентов стремится к достижению истинности определённого выражения. С использованием этой логики в работе [47] выражено несколько условий и доказано несколько теорем.

Однако логический подход позволяет приходить лишь к очень общим результатам, касающихся возможностей МАС. Что же касается верификации системы, проверки и вывода утверждений, касающихся её поведения в процессе работы, то и тут существуют различные подходы. Как правило,

сначала выделяется определённое семейство МАС, использующихся в той или иной области, описываются и формализуются его свойства.

Так, в работе [8] выделяется семейство систем, в которых поведение агентов описывается набором логических выражений (например, задано с помощью языка логического программирования), а процесс работы системы заключается в обмене агентов отдельными логическими выражениями. При этом процесс пересылки каждого отдельного сообщения подвержен случайной задержке, причём её продолжительность обладает известным распределением вероятности. Рассмотрение МАС в данном аспекте может быть полезно, например, с точки зрения анализа каналов связи и их влияния на работу распределённой системы. Опишем вкратце использованную при этом формализацию.

МАС представляется в виде $A = \{A_1 \dots A_n\}$. У каждого агента A есть БД I_A состоящая из «базисных атомов» – логических выражений вида $p(c_1 \dots c_k)$, где p – предикатный символ, а $c_1 \dots c_k$ – константы, а также почтовый ящик $MsgBox_A$. Текущее состояние агента это состояние его БД и почтового ящика $IM_A = \langle I_A, MsgBox_A \rangle$. Для каждой пары агентов определён канал связи CH_{AB} . Агенты кладут сообщения в канал и по прошествии определённого времени они попадают агентам-адресатам в почтовый ящик. Вероятность того, что сообщение будет идти по каналу связи в течение времени t определяется как $p_{AB}(t)$. В том числе, существует вероятность того, что сообщение вообще не дойдёт до агента-адресата. Время в данной модели считается дискретным.

Текущее состояние канала связи между агентами определяется как множество пар сообщение – продолжительность времени, в течение которого

сообщение находится в канале. Иными словами, $CH_{AB} = \{(Msg, t)\}$. Каждое сообщение представляет собой базисный атом.

С каждым агентом связана некая программа – последовательность параметризованных действий: добавление/удаление атомов из БД агента или пересылка сообщений другому агенту. На этой последовательности определено распределение вероятности и каждый следующий шаг выбирается в связи с текущим состоянием агента. В дальнейшем для определённого таким образом семейства проводится аналогия с Марковскими процессами и доказывается ряд свойств и оценок сложности верификации.

Исследование деятельности отдельных МАС с помощью Марковских процессов является весьма распространённым, однако, как правило, применяется лишь для конкретной системы, когда все правила деятельности агентов уже определены и конкретизированы.

Итак, мы видим, что в области теоретического описания и исследования свойств МАС существует значительное количество различных подходов, более или менее развитых. Однако математическое обеспечение мультиагентных систем как отдельного класса программных систем находится всё ещё в стадии развития. Поэтому как теоретический, так и практический интерес представляет формализация отдельных семейств мультиагентных систем и оценка их свойств.

1.4 Программное обеспечение. Агентные платформы

Для программной реализации мультиагентных систем в наше время, как правило, используются специализированные программные системы, содержащие набор средств как для программного описания деятельности агентов и состояния среды, так и для контроля и управления процессом их взаимодействия и работы. Такие системы называются агентными платформами.

Существует довольно большой набор платформ, подходящих для создания мультиагентных систем, и этот набор постоянно пополняется. В него входят NetLogo, StarLogo, Repast Symphony, Eclipse AMP, JADE, Jason. Сами эти платформы реализованы весьма по-разному: от отдельных сред разработки до встраиваемых плагинов и подключаемых библиотек. Они могут использовать как уже существующие языки различных парадигм, так и языки, специально разработанные для построения программных агентов, например, AgentSpeak в системе разработки Jason.

По степени формализованности, жёстко заданного способа взаимодействия с разработчиком системы можно разделить на промышленные (более жёсткие, но предоставляющие больше встроенных возможностей) и научные (дающие разработчику больше свободы).

По объёму инструментария платформы можно разделить на простые и сложные. Простые (NetLogo, StarLogo) имеют маленький, но мощный инструментарий, что позволяет быстро писать довольно сложные программы, однако при написании большой системы этого инструментария может не хватить. Возможность расширить его, дополнить собственными разработками, как правило, не предоставляется. Так что, если для формализации и настройки модели ещё подходят простые системы, то для реализации лучше выбрать сложную, предоставляющую больше возможностей, хотя ими и труднее пользоваться. Рассмотрим некоторые агентные платформы. Разделим их на классы в зависимости от способа реализации и опишем их характерных представителей.

Самостоятельные среды разработки. NetLogo. Данная система является типичным примером автономной среды разработки мультиагентных систем. С первого взгляда бросается довольно большой объём предоставляемых средств взаимодействия с пользователем – различных кнопок и регуляторов для ввода и корректировки информации и рисунков и диаграмм для упорядоченного вывода. Всё это, вкупе со свободной

распространяемостью системы, несомненно, составляет её основные плюсы. Необходимо также отметить очень широкий набор примеров моделей и систем, предоставляемых вместе со стандартной сборкой NetLogo. Эти примеры действительно помогают лучше понять устройство данной агентной платформы и её возможности.

Однако за богатством предоставляемых средств скрывается бедность непосредственно технических возможностей системы. Во-первых, набор предоставляемых средств визуализации, хотя и широк, но строго фиксирован и регламентирован, кроме того, невозможно добавление каких-то своих приёмов. А во-вторых, система страдает именно как средство разработки программ, фактически принуждая программиста писать всю программу одним файлом, без какой-либо чёткой структуры. Кроме того, язык программирования весьма специфичен, поскольку он основан на учебном языке Logo, и вдобавок содержит многочисленные методы, предназначенные уже специально для моделирования. Язык программирования NetLogo является скриптовым языком и написанные на нём программы исполняются ощутимо медленно. В настоящее время ведутся работы по созданию его транслятора в байт-код, что обещает в перспективе существенное улучшение времени исполнения [65].

Далее, система в принципе не предоставляет инструменты для распределённого, независимого исполнения. Функционирующие в ней субъекты лишены структуры агента, и являются лишь хранителями собственного состояния. Осуществление же их поведения реализуется исполнением функций на классе объектов, то есть, по сути, в рамках объектной парадигмы программирования.

С другой стороны, такая простота взаимодействия находится в гармонии со слабыми возможностями структурирования программы и вообще организации процесса разработки. Именно она позволяет писать небольшие, не громоздкие и не засорённые лишними и ненужными

практически (хотя и важными теоретически и структурно) элементами, но довольно мощные программы. Этому также способствует и отмеченная выше простота работы с широким, хотя и жёстко ограниченным набором средств визуализации.

Реализация большого научного проекта на подобной системе невозможна, но она и предназначена для другого – а именно, для подбора параметров, для тестирования тех или иных сторон модели, её калибровки. В этом плане такая простая во взаимодействии, хотя и бедная в плане предоставляемых возможностей система может оказаться незаменимой. На Рисунке 1.3 представлен внешний вид среды разработки NetLogo.

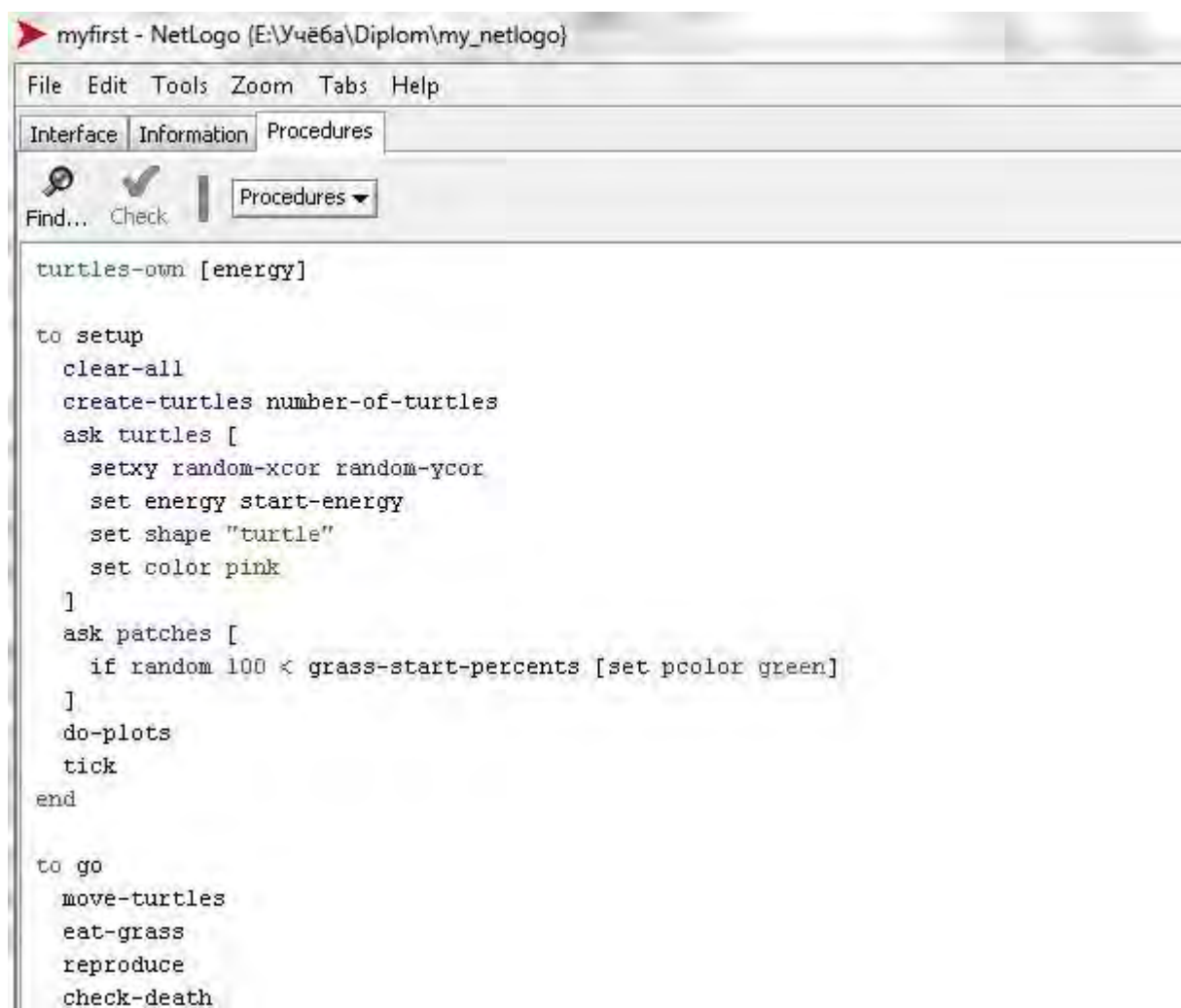


Рисунок 1.3 – Среда разработки NetLogo

Итак, подытожим. Как правило, самостоятельные среды разработки характеризуются высокой ценой использования (AnyLogic), либо же

неудобством интерфейса. Однако они позволяют сравнительно просто создавать простые имитационные модели, снабжены учебными примерами.

Отдельные подключаемые модули для разработки мультиагентных систем. Repast Symphony. Отдельные модули не представляют ещё собой собственно библиотеки, наборы исключительно разработческих средств для написания системы, поскольку они предоставляют широкий набор методов взаимодействия с пользователем. Однако они уже и не являются фактически автономными системами, а воплощаются в виде подключаемых модулей к существующим средствам разработки. Это довольно хорошо, поскольку позволяет совмещать богатство стандартных средств разработки со специализированными и предназначенными исключительно для агентного моделирования средствами. К данному классу инструментов относятся системы Jason, Eclipse AMP (2009).

Ярким представителем такого класса систем является Repast Symphony – подключаемый модуль к системе разработки (IDE) Eclipse. Во-первых, такая реализация позволяет использовать существующие возможности среды, да и новых языков программирования учить не приходится. А во-вторых, предоставляемые методы разработки агентных систем позволяют сконцентрироваться именно на написании агентов, причём использовать при этом весьма наглядные и интересные средства.

Главным плюсом, безусловно, является непосредственная работа с агентами и системами их взаимодействия. Так, существуют простые методы создания агента, задания его параметров, описание его поведения даже с использованием блок-схемы. Всё это делает работу с системой довольно удобной и понятной.

Однако, не всё так хорошо, как может показаться изначально. Кроме чисто случайных минусов, недочётов в среде взаимодействия с пользователем (невозможность отката при совершении некоторых важных

операций, сложности при сохранении и восстановлении и т.п.) и частичной недоработанности системы (временами появляются ошибки, препятствующие работе системы, отследить которые может только разработчик), есть ещё и трудности, видимо, присущие такому способу решения в целом. А именно – громоздкость и зачастую излишний объём проделываемой работы. Кроме того, несмотря на кажущуюся лёгкость создания программ в такой среде, сам процесс создания является строго регламентированным, и если что-то делается не так, как изначально задумано создателями системы, это может привести к ошибке на более низком уровне, т.е. на уровне уже собственно среды разработки и языка программирования. На Рисунке 1.4 представлен внешний вид среды разработки RepastSymphony.

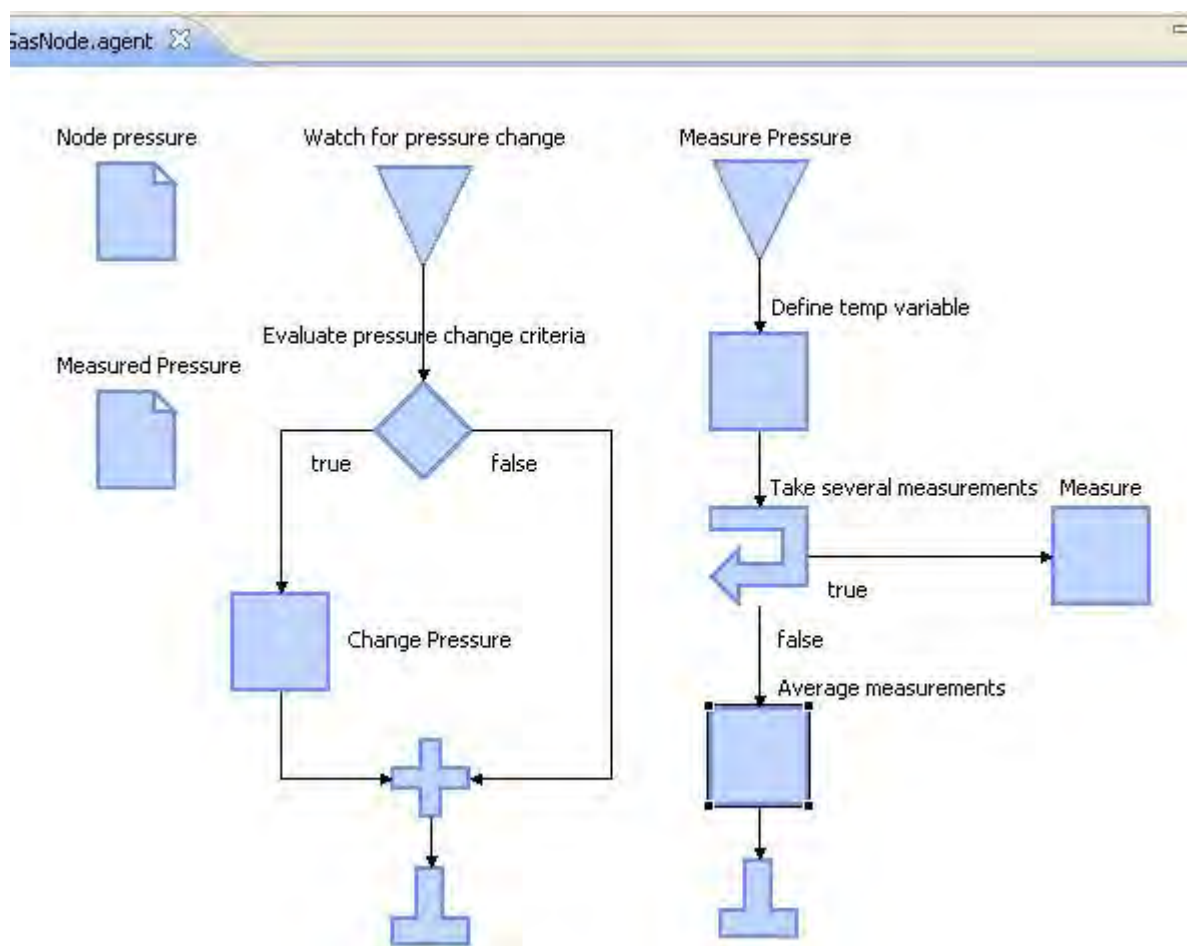


Рисунок 1.4 – Среда разработки RepastSymphony

Всё это делает необходимым либо периодическое использование низкоуровневых средств разработки, либо строгое следование схемам построения систем. Такое положение частично перечёркивает ту огромную выгоду, которую приносит совмещение стандартных и мощных средств разработки со специализированными методами.

Отдельные программные библиотеки. JADE. Вообще говоря, JADE предоставляет больше, чем просто библиотеку классов. Она также включает в себя средства визуализации и управления работой системы. Однако они реализованы просто как библиотечные функции, так что отнести всю систему следует именно к программным библиотекам.

Итак, преимущества и недостатки такой системы ясны сразу – богатая функциональность, с одной стороны, и необходимость детального изучения и использования дополнительных усилий на создание визуализации, с другой. Кроме того, непосредственно про JADE следует сказать, что она основывается на существующем стандарте мультиагентных систем, что, безусловно, делает работу с ней более упорядоченной, хотя, возможно, и загромождает ненужными действиями.

Также достоинством данной системы является возможность её распределённого исполнения, что позволяет решать задачи большой сложности, складывая мощности многих машин. Такой подход является особенно актуальным именно для мультиагентных систем, поскольку в нём модули, занятые собственно исполнением кода изначально автономны и слабо связаны между собой. В общем, система довольно большая и солидная, хотя её детальное изучение и займёт, видимо, много времени, но предоставляемые ею возможности незаменимы при решении сложных задач.

Итак, что касается отдельных библиотек, то они, во-первых, как правило, распространяются с исходным кодом, так что при желании в них даже можно внести изменения. Во-вторых, их использование не ограничивает нас в выборе подходов к решению задачи, заданию различных

параметров. Что же касается отсутствия механизма визуализации – то тут надо отметить, что нужных нам средств визуализации нет и у решений, выполненных в виде отдельных сред разработки. Так что, вероятнее всего, всё равно придётся использовать, какие-то другие, сторонние средства. К тому же такие библиотеки весьма свободно распространяются.

Единственным существенным минусом при таком подходе остаётся, пожалуй, лишь необходимость детального изучения библиотеки. Однако использование самостоятельных систем не даст нам в этом особого преимущества, поскольку тоже требует детального изучения особого языка программирования, притом довольно небогатого по предоставляемой функциональности и расширяемости. На Рисунке 1.5 представлен внешний вид средства управления агентами в платформе JADE.

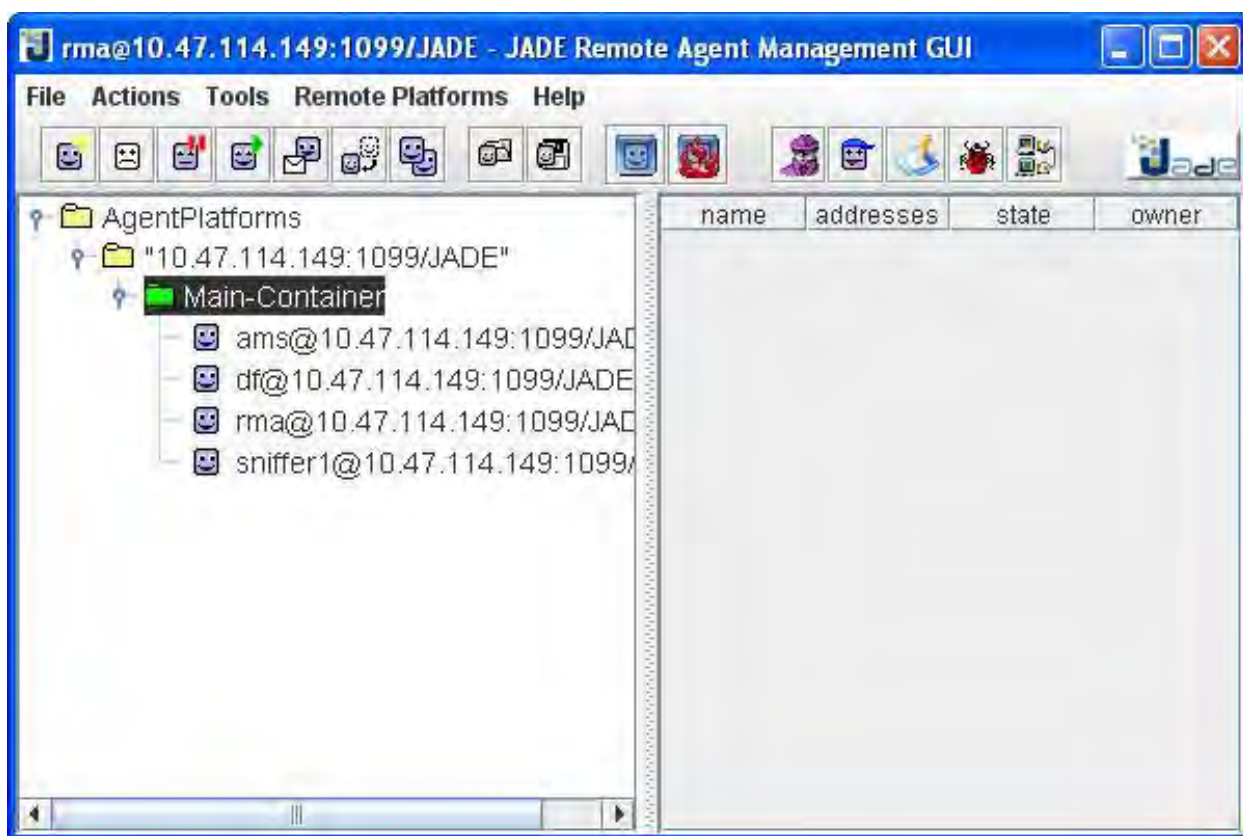


Рисунок 1.5 – Средство управления агентами в платформе JADE

1.5 Агентное моделирование. Системы с роевым интеллектом

Ранее в текущей главе было выделено две большие группы MAC, весьма отличающиеся друг от друга. Соответственно, при их изучении будет

существенно отличаться подход к ним, как в области вопросов и целей исследования, так и в области применяемых формализмов и математических методов. Поэтому, чтобы начать исследования, предварительно надо определиться, какую группу мы рассматриваем. В дальнейшем мы выделим семейство МАС, в которое включена и составленная нами система, обоснуем важность этого семейства на примере входящих в него систем, формализуем его и докажем его некоторые свойства при заданных условиях.

В данной работе рассматриваются главным образом системы второй группы. Действительно, поскольку интеллектуальность систем первой группы обеспечивается не столько их организацией, сколько самими агентами, постольку и оценка их свойств опирается в первую очередь на исследованных свойствах тех методов, благодаря которым достигается интеллектуальность отдельных агентов. Для второй группы как раз характерно то, что интеллектуальность системы имеет особую, мультиагентную природу и исследование её свойств в первую очередь способствует развитию теоретического обеспечения именно мультиагентных систем.

Вторую группу, как уже было сказано, образуют системы, которые состоят из большого числа (до нескольких тысяч или даже десятков тысяч) простых агентов, взаимодействующих главным образом со средой, а не друг с другом, и обладающих простым, но не строго детерминированным поведением. Такие системы, как известно, способны демонстрировать «роевой интеллект» (swarm intelligence), приспосабливаться к меняющимся условиям, находить «ниши» с оптимальным значением тех или иных показателей. Опишем вкратце важные примеры систем, попадающих во вторую группу.

Агентные модели. Первым, наиболее важным в данной работе представителем рассматриваемого семейства являются имитационные агентные модели (agent-based models), наиболее распространённые в

общественных науках – социологии, экономике и т.п. Встречаются они и в естественных науках (физике, биологии), в общем, во всех случаях, когда нужно промоделировать деятельность системы, состоящей из большого количества субъектов, чья деятельность или вероятностна по своей природе, или подвержена влиянию необозримого количества различных случайных воздействий. При этом поведение системы в целом определяется не сверху, а снизу, т.е. в результате совокупности действий составляющих её агентов. В число моделируемых таким образом процессов входят, например, процесс распространения вируса, или установления цены на рынке, или распространения пожара в лесу.

Популярность этого подхода в моделировании социально-экономических отношений объясняется ещё и тем, что в экономике и общественных науках создать продуманную и сложную модель действующего субъекта (то есть построить высокоинтеллектуальный агент) подчас бывает довольно сложно. Это требует наличия высокоразвитой теории психологии субъектов, чёткого знания алгоритмов, используемых ими для принятия тех или иных решений. Эффективнее оказывается принять простую схему действий агента, а отклонения от неё считать случайной величиной с заданным распределением.

Конечная цель агента в таких системах (и направленная к её достижению деятельность) определяется поиском более или менее удачного приближения к наилучшему решению какой-то определённой задачи. Например, поиск глобального максимума некоторой функции $\max(f)$ с помощью приближённых методов или эвристик, характерных для задач искусственного интеллекта. В экономике это, как правило, максимизация некоторой заданной изначально функции полезности при определённых ограничениях. Решение задачи производится поэтапно. Сначала агенты ищут приближённое решение (каждый исходя из собственных представлений о функции и собственного алгоритма, и используя доступные сведения из

среды). При этом возможно как взаимодействие агентов, согласованное определение решений, так и совершенно независимая деятельность агентов. Затем агенты изменяют своё состояние и устанавливают некоторые значения среды согласно найденному решению. После этого снова ищут приближение, исходя из уже изменившихся значений переменных среды. В последовательном повторении этих двух этапов и заключается процесс моделирования.

Эвристические мета-алгоритмы для задач локального поиска. Вторым, не менее важным примером МАС рассматриваемого типа являются мультиагентные реализации многих видов эвристических алгоритмов оптимизации. В последние десятилетия всё большую популярность набирают такие эвристические мета-алгоритмы как метод муравьиной колонии, роя частиц и т.п. Кроме этого, эволюционные алгоритмы могут быть представлены в мультиагентном виде, где каждая особь представляет собой агент, целью которого является оптимизация целевой функции, а деятельность заключается в выборе наилучшего «партнёра» и размножении.

Мы будем рассматривать определение задачи локального поиска данное в [23]. Для задач такого типа существуют различные общие алгоритмы (или схемы алгоритмов, называемые также мета-алгоритмами). Возьмём, к примеру, генетический алгоритм (ГА). Описание схемы генетического алгоритма широко известно, его можно взять, например, в [25].

Будем считать каждую особь отдельным агентом, кроме того, введём агента, отвечающего за селекцию, поскольку, как правило, процесс селекции производится централизованно. Тогда деятельность агентов заключается в вычислении целевой функции и оповещении агента селекции, а также в размножении, то есть в определении свойств агента-представителя следующего поколения. Как определение целевой функции, так и размножение может происходить независимо от остальных агентов.

Поскольку рассматриваемая функция, как правило, имеет довольно сложный вид (для простых функций чаще всего имеются специализированные алгоритмы, работающие гораздо эффективней, чем ГА), то её вычисление занимает значительное время по сравнению с другими действиями, такими как сбор информации, определение особей для размножения и собственно размножение (формирование характеристик новой особи). Таким образом, нагрузка на систему в целом равномерно распределяется по составляющим её агентам.

Деятельность агента селекции заключается в сборе информации о функции полезности каждой особи определении пар особей для размножения. Впрочем, как это определение, так и количество особей могут варьироваться в зависимости от реализации ГА.

Итак, мы обрисовали круг интересующих нас задач, входящих в рассматриваемое семейство МАС. Основой интеллектуальности, разделения на агенты и, в перспективе, распараллеливания процесса исполнения их программных реализаций всех рассмотренных систем является стохастическая деятельность нескольких агентов на каждом шаге.

Перейдём теперь к формализации интересующего нас семейства МАС, с учётом поставленного перед нами круга задач и определённого выше понятия верификации для систем данного семейства.

1.6 Формальное описание

Итак, для основных представителей рассматриваемого нами семейства МАС характерным оказывается следующее. Процесс работы системы представляется в виде итераций, последовательно проходимых однотипных шагов. Но на каждом из этих шагов в действие вступают, как правило, несколько агентов. Далее, поведение агентов случайно, но распределение их решений зависит от вычисляемого значения некоторой определённой функции. Кроме этого, после принятия всеми агентами решения на

конкретном шаге в процесс может включаться некоторый центральный агент, определяющий параметры для следующего шага. В общем случае все агенты делятся с ним информацией о принятых решениях – то есть о значении своих свойств и целевой функции. Но в конкретных моделях такого централизованного обмена информацией может и не быть. Например, каждого агента могут интересовать значения характеристик лишь выбранных агентов, скажем, его соседей (отношение соседства определяется на множестве агентов заранее или может меняться в процессе работы МАС). Например, в случае размножения в мультиагентной реализации ГА — это значения характеристик особи, выбранной в качестве партнёра. На основании этих функций, вычисленных централизованно и/или каждым агентом отдельно, определяются исходные значения для следующей итерации.

В общем можно выделить два основных случая. В первом агенты действуют в течение шага совершенно независимо и разрешение конфликтов (если таковые имеются) происходит во время синхронизации. Это позволяет полностью реализовать преимущества параллельного исполнения, например, для ГА. Во втором случае агенты принимают решения согласованно. Например, это решение о переезде агентов в рассмотренной выше модели Шеллинга. Это накладывает ограничения на использование параллельного исполнения, но всё равно не исключает выгоды, достигаемой им.

Рассмотрим сначала общий случай, при котором мы не знаем, зависят ли действия агентов друг от друга на протяжении одного шага и опишем процесс моделирования для этого случая. Процесс представляет собой пошаговый переход от одного состояния всех агентов и среды к другому. При этом правила перехода определяются правилами перехода, то есть алгоритмами поведения агентов, вообще говоря, вероятностными. Определим состояние системы как совокупность состояния всех её агентов и среды. Предположим, что правила перехода агента из одного состояния в другое зависят только от его текущего состояния (и не зависят от

предыдущих, уже пройденных агентом состояний). Если это не так, то в набор состояний агента можно добавить «память» агента о предыдущих состояниях, и привести правила к указанному виду. В таком случае и переход всей системы из одного состояния в другое представляется как случайная величина, чьё распределение зависит только от текущего состояния.

Система, в которой деятельность агентов не зависит друг от друга, является частным случаем описанной выше системы. Здесь, кроме описания состояний системы в целом, мы можем описать состояния каждого агента. Предположим, все они принадлежат одному общему для всех агентов пространству состояний. Для многих случаев это действительно так (например, методы оптимизации), остальные можно привести к этому виду, взяв для каждого агента пространство состояний равное произведению пространств состояний всех агентов. Далее, деятельность агентов можно определить как вычисление значения некоторой характеристической функции, зависящей от значения этого состояния и переход к другому состоянию в зависимости от состояний и значений функции всех агентов. В рамки приведённого описания укладываются многие важные примеры МАС, например, мультиагентная реализация ГА. Опишем деятельность агентов пошагово и проиллюстрируем её на этом примере.

Пусть нам даны n агентов $(A_1, \dots, A_n)$. Целью каждого агента является нахождение максимума заданной функции $f(x)$ (будем называть её целевой функцией) на заданной области D . Действия агентов определяются следующим алгоритмом. Изначально каждый агент получает начальное решение $x_{0,j}$ ($j = 1, \dots, n$). Для краткости будем обозначать совокупность значений переменной x у всего набора агентов как $X = \langle x_1, \dots, x_n \rangle$. Итак, агенты получают (или находят случайным образом) начальный набор X_0 и начинают пошагово искать максимум целевой функции. Опишем алгоритм их деятельности (в общем виде) на i -м шаге.

У каждого агента имеется значение x_{ij} (короче говоря, задан набор X_i). Сначала каждый агент вычисляет значение целевой функции на данном ему значении x ($f(x_{ij})$), обозначим для краткости f_{ij} , а набор значений f у всех агентов - F_i). Затем агенты обмениваются информацией о найденных решениях (каждый агент сообщает каждому, или передаёт в общую память, значения x_{ij} , f_{ij}). Далее, исходя из полученных решений, каждый агент определяет следующее значение $x_{i+1,j} = g_{ij}(X_i, F_i)$. Обозначим это кратко так: $X_{i+1} = G_i(X_i, F_i)$, где функция $G_i(X_i, F_i)$ определяет набор $\langle g_{i,1}(X_i, F_i), \dots, g_{i,n}(X_i, F_i) \rangle$. Функции $g_{ij}(X_i, F_i)$ в общем случае могут не быть строго детерминированы, и для многих мультиагентных моделей это действительно так.

Поскольку вычисление значений функций f и g происходит агентами независимо, постольку можно ожидать пропорционального количеству агентов прироста производительности работы системы, в случае, если деятельность каждого агента реализуется отдельным вычислителем (например, процессором или одним ядром многоядерного процессора), а обращение вычислителей к общей памяти происходит за пренебрежимо малое время (в сравнении со временем, необходимым на вычисление значений функций f и g).

Представление генетических алгоритмов в рамках полученной модели. Деятельность довольно большого круга интересующих нас МАС можно представить в предложенном нами общем виде. Интересным примером является мультиагентная реализация генетического алгоритма (ГА). Возьмём за основу описание ГА, данное в [25]. Предположим, что агенты – это селекционеры, выращивающие, отбирающие и скрещивающие лучших особей. Количество селекционеров равно количеству особей в одной популяции. Сначала селекционеры получают начальную популяцию X_0 . Затем они вычисляют значение функции полезности каждый для своей особи ($F(X_0) = \langle f(x_{0,1}), \dots, f(x_{0,n}) \rangle$). Затем каждый агент-селекционер выбирает

сообразно значениям функции полезности случайным образом 2 особи-решения и скрещивает их (алгоритм выбора и алгоритм скрещивания могут быть разными в разных реализация ГА). Таким образом определяется значение функции $g_{i,j}(X_i, F_j)$. Формально функцию g можно определить через умножение случайным образом составленных матриц на векторы, что например, проделано в [3].

ГА применяются в основном для задач, в которых функция полезности имеет достаточно сложный вид, что препятствует решению её точными методами. Это означает, что зачастую вычисление функции полезности представляет собой весьма трудную задачу и основное время в реализации ГА занимает вычисление функции полезности для одной особи.

Представление ГА в виде мультиагентной системы, где каждому агенту соответствует особь в популяции позволяет существенно сократить время работы ГА. Однако ещё одна выгода такого представления состоит в том, что оно позволяет применить математические методы для оценки свойств описываемой им МАС.

МАС со сложными зависимостями между агентами. Мы рассмотрели наиболее удобный для построения параллельных программных систем, когда на каждом отдельном шаге агенты действуют совершенно независимо друг от друга. Однако в более общем случае независимость действий агентов на протяжении шага нельзя гарантировать. Напротив, для более развитых моделей и методов агентам необходима коммуникация, хотя она, возможно, будет и не так сложно организована, как в МАС первой группы. Тем не менее, в действиях агентов сохраняется некоторая независимость, а круг агентов, с которыми общается данный конкретный агент для принятия решения на определённом шаге, зачастую может быть ограничен (например, заданием на множестве агентов отношения соседства). Всё это ещё оставляет возможность для повышения эффективности работы системы по времени благодаря параллельному исполнению.

В любом случае представляется полезным рассмотреть более общую формализацию, при которой нам не известно заранее, общаются ли агенты между собой на протяжении одного шага работы системы или нет. В таком случае мы будем, не углубляясь в работу агентов для каждой конкретной системы, рассматривать состояние системы в целом и рассматривать законы его изменения. И лишь для конкретных случаев мы будем опускаться на уровень деятельности отдельных агентов, и смотреть, удовлетворяет ли она нужным нам условиям.

Итак, мы будем рассматривать состояние всей системы X . Множество всех состояний обозначим S . Поскольку в общем виде у нас может не быть заданной для всех агентов функции f , то мы будем рассматривать функцию перехода G_i как зависящую только от текущего состояния X (её можно представить в таком виде и для рассмотренного ранее более конкретного случая с независимой деятельностью агентов). Иными словами, мы получаем $X_{i+1} = G_i(X_i)$. Функция f у нас характеризовала состояние отдельных агентов. Теперь нам нужна функция, аналогичная ей, но определённая уже на множестве состояний всей системы. Обозначим её $h(X)$. В случае, когда нам дана f , можно, например, задать $h(X_i) = \max(f(x_{i,1}), \dots, f(x_{i,n}))$, где $X_i = \langle x_{i,1}, \dots, x_{i,n} \rangle$.

Нетрудно заметить, что мы получили Марковский процесс с дискретным временем. Марковские процессы часто используются для теоретического изучения свойств конкретных процессов случайного поиска или работы отдельных моделей. Мы же в данной работе применим связанный с ними математический аппарат для более общей задачи верификации выделенного и формализованного нами семейства МАС. В следующей главе мы углубим аналогию между МАС и Марковскими процессами и докажем ряд свойств систем. Позднее мы покажем, как на основе доказанных нами свойств значительно облегчается процесс доказательства свойств конкретных

МАС и становится возможным построение системы с заранее заданными свойствами.

2 Исследование семейства МАС с помощью цепей Маркова

В данной главе описаны основные теоретические результаты исследования. Для полученной формализации семейства МАС проведена аналогия с цепями Маркова, с помощью этой аналогии доказан ряд свойств исследуемого семейства МАС. Также приведён результат работы по созданию агентной платформы – программной системы (в данном случае реализованной как библиотека классов), которая позволяет упростить процесс построения МАС исследуемого семейства, а также содержит средства управления работой системы и наблюдения за её ключевыми свойствами, исследуемыми в данной работе.

2.1 Проведение аналогии с цепями Маркова

Для начала введём формальное определение Марковского процесса и ряда интересующих нас свойств. Точное и полное определение Марковского процесса можно найти, например, в [5]. Говоря неформально, Марковским является такой случайный процесс, у которого при фиксированном настоящем состоянии все будущие зависят только от него (а не от прошлых). В общем случае не существует ограничений ни на непрерывность времени, ни на пространство состояний.

Однако, исходя из формализации нашего семейства МАС, нас будут интересовать лишь процессы с дискретным временем (поскольку МАС нашего семейства работают пошагово).

Далее, пространство состояний мы также примем дискретным и конечным, поскольку для основных представителей нашего семейства это действительно так и поскольку теория Марковских процессов развита в первую очередь для дискретных пространств состояний. Такие процессы называются Цепями Маркова.

Итак, представим себе последовательность случайных величин $X_0, X_1, \dots, X_i, \dots$, $\forall i: X_i \in S$, где S – дискретное множество состояний. Эта последовательность называется цепью Маркова (ЦМ) если

$$P(X_n = y | X_{n-1} = y_{n-1}, \dots, X_0 = y_0) = P(X_n = y | X_{n-1} = y_{n-1}), \quad (2.1)$$

иными словами, вероятность попасть в определённое состояние зависит лишь от предыдущего состояния.

Уже из определения можно увидеть способ задания ЦМ через вероятность попадания из одного состояния в другое за единицу времени, заданную для всех состояний.

$$P_{xy}(n) = P(X_{n+1} = y | X_n = x). \quad (2.2)$$

Эти элементы $P_{xy}(n)$ составляют так называемую матрицу переходов.

Очевидно они быть неотрицательны и удовлетворять условию

$$\forall x, n: \sum_{y \in S} P_{xy}(n) = 1. \quad (2.3)$$

Также видно, что необходимо задать распределение для нулевого момента времени, поскольку его определять не из чего.

$$p(x) = P(X_0 = x) \quad (2.4)$$

Эти элементы составляют вектор, который называют начальным распределением. Как можно заметить, при заданных матрице перехода и начальном распределении можно получить распределение вероятности состояний на заданном шаге путём перемножения нужного количество матриц перехода и вектора начального распределения. Если матрица перехода постоянна и не зависит от рассматриваемого момента времени, то такая ЦМ называется однородной. В дальнейшем будем рассматривать только такие цепи. Обозначим вероятность перехода из состояния x в

состояние y за n шагов как $P_{xy}^{(n)}$. Как видно, эти элементы так же составляют матрицу. Тогда для однородной цепи очевидно следующее соотношение для матриц перехода.

$$P^{(n)} = P^n \quad (2.5)$$

Основным интересующим нас фактом из теории Цепей Маркова является эргодическая теорема. Приведём её формулировку по [5]. При условии, что

$$\exists x_0, h, \delta | \forall x: P_{xx_0}^{(h)} \geq \delta > 0 \quad (2.6)$$

существует для всех y и не зависит от начального состояния x предел

$$\lim_{t \rightarrow \infty} P_{xy}^{(t)} = \mu(y). \quad (2.7)$$

Вектор $\mu(y)$ называется стационарным распределением. Очевидно, что, если текущее распределение стационарно, то оно уже не изменяется после применения матрицы перехода. Иными словами,

$$\mu P = \mu. \quad (2.8)$$

На практике проще бывает воспользоваться эквивалентным условием эргодичности. ЦМ должна быть неразложима, возвратна и апериодична. Неразложимость означает возможность перейти через определённое время из любого состояния в любое. Возвратность – вероятность в дальнейшем вернуться в текущее состояние (сумма вероятностей вернуться в состояние через любой промежуток времени, от 1 до бесконечности, равна 1). Апериодичность означает отсутствие периодических состояний, то есть таких состояний, в которые система может попасть через кратные определённому числу периоду.

Для наглядности этих условий можно использовать граф переходов. Определим для ЦМ граф переходов по заданной матрице переходов

$$G = \langle S, E \rangle; (x, y) \in E \Leftrightarrow P_{xy} > 0. \quad (2.9)$$

Тогда все возможные переходы из одного состояния в другое мы можем совершить, проходя по рёбрам графа.

Теперь вернёмся к введённому выше нами описанию работы МАС. Оно в нашем описании представляет Марковский процесс с дискретным временем, где множество состояний — это S , а матрица перехода определяется функцией $G(X)$. В дальнейшем для описания состояний МАС будем употреблять строчные буквы, а если мы рассматриваем поведение и состояние отдельных агентов, оговаривать это.

$$P_{xy}(n) = P(G_n(x) = y). \quad (2.10)$$

Наиболее часто встречается случай, при котором правила, определяющие действия агента, не меняются на протяжении работы МАС, следовательно, матрица перехода не зависит от номера шага. Это означает, что рассматриваемый нами Марковский процесс однороден.

Как уже было сказано, нас интересует поведение МАС после количества шагов, стремящегося к бесконечности. В этом плане для нас интересна такая характеристика ЦМ как стационарное распределение. Для МАС это означает, что мы получаем вероятность нахождения системы в том или ином состоянии через количество шагов, стремящееся к бесконечности. Таким образом, задача верификации свойств системы, проверки и установлению утверждений о её характеристиках сводится к задаче нахождения стационарного распределения Марковской цепи, соответствующей данной МАС.

2.2 Нахождение эргодического распределения и анализ поведения МАС

Связь стационарного распределения с характеристической функцией. Итак, мы используем представление работы МАС в виде Цепи

Маркова. Для такого вида положим ряд условий и докажем следующую теорему выражении её стационарного распределения.

Теорема 1. Обозначим множество состояний цепи Маркова S . Предположим, что S конечно. Пусть на S определена некоторая функция (будем называть её характеристической) $h: S \rightarrow \mathbb{R}$, а также дана матрица $\Gamma: S \times S \rightarrow \mathbb{R}$. Предположим, что:

- Цепь Маркова неразложимая и непериодическая;
- вероятность перехода из одного состояния в другое может быть представлена в виде

$$P_{xy} = \Gamma_{xy} h(y) \quad (2.11)$$

- при этом

$$\Gamma_{xy} = \Gamma_{yx} \text{ для всех } x, y \in S. \quad (2.12)$$

Тогда стационарное распределение имеет вид

$$\mu(x) = kh(x), \text{ где } k = \frac{1}{\sum_{x \in S} h(x)}. \quad (2.13)$$

Прежде чем доказывать теоремы, рассмотрим её важное для теории МАС следствие, которое, как мы покажем в 5й главе, часто используется при доказательстве свойств поведения отдельных систем.

Следствие 1. Пусть $h(x)$ – характеристическая функция, определённая на множестве состояний МАС. При этом вероятность перехода МАС из состояния x в состояние y равна $\Gamma_{xy} h(y)$, причём $\Gamma_{xy} = \Gamma_{yx}$ для всех $x, y \in S$. Определим граф окрестностей $G = \langle V, E \rangle$, где $V = S$ а $(x, y) \in E \Leftrightarrow \Gamma_{xy} \neq 0$.

Пусть на данном графе выполнены условия эргодичности (то есть для вершин графа выполняются условия, сформулированные выше для состояний ЦМ и графа переходов). Тогда вероятность попадания МАС в то или иное состояние с увеличением времени работы стремиться к величине, пропорциональной значению характеристической функции на данном

состоянии. Иными словами, для вероятности попадания системы из любого состояния x в состояние y выполняется следующая формула:

$$\forall x: \lim_{t \rightarrow \infty} P_{xy}^{(t)} = kh(y); k = \text{const.} \quad (2.14)$$

Доказательство Теоремы 1. Из условий эргодичности мы получаем по эргодической теореме факт существования стационарного распределения. Осталось лишь доказать, что оно представимо в нужном нам виде. Как известно, для стационарного распределения необходимым и достаточным условием является следующее утверждение: $\mu P = \mu$. Покажем, что для выбранного нами μ оно выполняется. Для произвольного состояния y рассмотрим выражение $\sum_{x \in S} \mu(x) P_{xy}$.

$$\sum_{x \in S} \mu(x) P_{xy} = \sum_{x \in S} kh(x) \Gamma_{xy} h(y) = kh(y) \sum_{x \in S} \Gamma_{xy} h(x) = \mu(y) \sum_{x \in S} \Gamma_{yx} h(x)$$

Последнее равенство мы получили в силу исходных положений теоремы

$$\Gamma_{xy} = \Gamma_{yx}$$

$$kh(y) = \mu(y).$$

Далее, поскольку $P_{xy} = \Gamma_{xy} h(y)$ – элементы матрицы перехода ЦМ, то для них выполняется утверждение

$$\sum_{y \in S} \Gamma_{xy} h(y) = 1. \quad (2.15)$$

Заменой y на x получаем

$$\sum_{x \in S} \Gamma_{yx} h(x) = 1. \quad (2.16)$$

Подставив это выражение, получаем

$$\forall y \sum_{x \in S} \mu(x) P_{xy} = \mu(y), \quad (2.17)$$

иными словами, в векторном виде $\mu P = \mu$. Это значит, что определённое в теореме $\mu(x)$ действительно является стационарным распределением для данной ЦМ, что и следовало доказать.

Доказательство Следствия 1. Следствие 1 доказывается тривиальным образом с использованием приведённой выше аналогии между понятиями, связанными с МАС и терминами ЦМ. Действительно, с течением времени вероятность попадания системы в то или иное состояние стремится к величине стационарного распределения для этого состояния. Поскольку в Теореме 1 доказано, что стационарное распределение имеет определённый вид, то утверждение следствия становится очевидным.

Тот факт, что $h(y) > 0$, позволяет нам переформулировать условие эргодичности для графа окрестностей. Действительно, переход из состояния x в состояние y возможен (иными словами, x и y – соседние состояния) тогда, когда $P_{xy} \neq 0 \Leftrightarrow \Gamma_{xy} \neq 0$. Заметим кстати, что граф G оказывается неориентированным, поскольку $\Gamma_{xy} = \Gamma_{yx}$, следовательно, $(y, x) \in E \Leftrightarrow (x, y) \in E$. Иными словами, отношение соседства симметрично.

Оценка распределения вероятности характеристической функции.

Тот факт, что вероятность нахождения в каждом состоянии в пределе пропорциональна характеристической функции на этом состоянии, на первый взгляд кажется весьма полезным. Например, если МАС представляет собой систему, решающую задачу оптимизации, то мы получаем, что вероятность попадания в то или иное состояние тем больше, чем больше

значение функции. Однако, если нас интересует приближение к оптимуму, а количество состояний, в котором значение исследуемой функции близко к оптимальному мало, то вероятность приблизиться к оптимуму будет по-прежнему малой.

Проиллюстрируем это на предельном случае – пусть в 1000 состояний значение функции равно 1, а в одном (оптимальном) – 10. Тогда вероятность попадания в оптимальное состояние относится к вероятности непадения в него как 10:1000, то есть, вероятность попасть в оптимальное состояние равна 1%.

Таким образом, о свойстве, доказанном в Теореме 1 можно сказать то, что сказано для его частного случая в [74].

Для исследования поведения МАС полезно было бы иметь функцию, оценивающую вероятность приближения к оптимальному состоянию. Очевидно следующее утверждение

$$P(h(x) \geq \rho) = \sum_{x|h(x) \geq \rho} \mu(x). \quad (2.18)$$

Однако его вычисление на практике едва ли возможно. Попробуем его оценить, используя приближённые функции.

Теорема 2. Пусть выполнены условия Теоремы 1, то есть $\mu(x) = kh(x)$. Также предположим, что нам дана оценочная функция

$$\tilde{N}(\rho) \leq N(\rho) = |\{x|h(x) \geq \rho\}|. \quad (2.19)$$

Тогда для вероятности того, что характеристическая функция превзойдёт заранее данное значение ρ_0 , верна формула

$$P(h(x) \geq \rho_0) \geq k(\rho_0 \tilde{N}(\rho_0) + \int_{\rho_0}^{\infty} \tilde{N}(\rho) d\rho). \quad (2.20)$$

Доказательство Теоремы 2. Рассмотрим введённую в условии теореме функцию $N(\rho) = |\{x|h(x) \geq \rho\}|$. Из её определения видно, что, поскольку у нас пространство состояний дискретно, то эта функция имеет ступенчатый вид и убывает до 0. Её производную можно определить через дельта-функцию и рассматривать её дальше как непрерывную.

Рассмотрим малое приращение аргумента $d\rho$ в точке ρ . Количество состояний на данном промежутке равно $-dN(\rho)$. При этом вероятность попадания в эти состояния равна (по условию Теоремы 1) $kh(x) = k\rho$. Осталось проинтегрировать полученное выражение от заданного значения $h(x) = \rho$ до максимального. Обозначим для краткости искомую вероятность через P . Получаем

$$P = P(h(x) \geq \rho_0) = - \int_{\rho}^{\rho_{max}+0} k\rho dN(\rho). \quad (2.21)$$

Поскольку не существует значений функции, превосходящих ρ_{max} , а само это значение может быть неизвестно, удобнее перейти к интегрированию до бесконечности. Проинтегрируем полученное выражение по частям и получим

$$P = -k\rho N(\rho)|_{\rho_0}^{\infty} + \int_{\rho_0}^{\infty} kN(\rho) d\rho = k(\rho_0 N(\rho_0) + \int_{\rho_0}^{\infty} N(\rho) d\rho),$$

поскольку при $\rho > \rho_{max}$ и, следовательно, на бесконечности $N(\rho) = 0$.

Теперь перейдём к введённой в условии теоремы функции $\tilde{N}(\rho)$.

$$\tilde{N}(\rho) \leq N(\rho) \Rightarrow$$

$$P \geq k(\rho_0 \tilde{N}(\rho_0) + \int_{\rho_0}^{\infty} \tilde{N}(\rho) d\rho),$$

что и требовалось доказать. Поскольку вычислить k зачастую также сложно, введём оценку для этой величины. Для этого будем использовать, функцию, ограничивающую сверху количество состояний с данным значением характеристической функции.

$$\tilde{N}(\rho) \leq N(\rho) = |\{x|h(x) \geq \rho\}| \leq \bar{N}(\rho)$$

С использованием этих функций мы можем получить оценку распределения вероятности не только снизу, но и, по аналогии, сверху:

$$\frac{\rho_0 \bar{N}(\rho_0) + \int_{\rho_0}^{+\infty} \bar{N}(\rho) d\rho}{\int_{-\infty}^{+\infty} \tilde{N}(\rho) d\rho} \geq P(h(x) \geq \rho_0) \geq \frac{\rho_0 \tilde{N}(\rho_0) + \int_{\rho_0}^{+\infty} \tilde{N}(\rho) d\rho}{\int_{-\infty}^{+\infty} \bar{N}(\rho) d\rho}$$

2.3 Описание специализированной агентной платформы MASwarm

В главе 1 нами были рассмотрены некоторые агентные платформы. Теперь, приступая к практической работе с агентными моделями (и другими системами с роевым интеллектом) мы должны определиться с тем инструментом, который будем при этом использовать.

Для создания небольших МАС с роевым интеллектом лучше всего подходит платформа NetLogo. Однако она имеет ряд существенных минусов. Главные минусы связаны со средой разработки. Это специфический язык, который обуславливает медленное исполнение программ и вынуждает программиста применять довольно замысловатые приёмы для реализации привычных вещей. Это среда разработки, жёстко ограничивающая программиста в выборе средств визуализации и стиле написания программы (а также исключая возможность модульности программы). Это

закрытость системы в целом, не позволяющая использовать написанные программы из других программ.

Подытоживая, можно сказать, что система NetLogo является плохо масштабируемой, она замечательно подходит для реализации небольших и нетребовательных ко времени исполнения МАС, однако её использование для создания сложных, больших или просто требовательных ко времени МАС сопряжено с неразрешимыми трудностями.

Использование стороннего языка и среды разработки (например, C++ или Java) сняло бы все эти проблемы. Однако агентные платформы, использующие такое решение, как правило, излишне тяжеловесны для нашей задачи. Они изначально ориентированы скорее на реализацию МАС 1го типа (систем РИ, см. главу 1). Всё это делает привлекательным идею создания собственной агентной платформы, которая смогла бы соединить положительные черты как одних, так и других платформ. В общем она должна предоставлять примерно ту же функциональность, что и NetLogo, однако средой разработки должна стать некоторая независимая и достаточно хорошо развитая среда (в этом качестве была выбрана Java).

Ещё более привлекательным перспективу создания собственной платформы делает тот факт, что нами уже проведена формализация систем рассматриваемого семейства, и она может быть непосредственно использована при реализации нашей платформы и работе с ней. Действительно, проведённая в главе 1 формализация семейства МАС позволяет полностью задать работу системы с точностью до определения ключевых функций – h , G и (в случае независимости действий агентов на каждом шаге) f . Понятен простейший набор средств управления, они включают в себя рычаги запуска и остановки процесса моделирования. Ясны и значения, нуждающиеся в мониторинге – это номер текущего шага и значение характеристической функции h . Также необходимы чисто технические средства обеспечения ввода/изменения параметров системы,

визуализации в различных формах (в том числе встраиваемой), выкладывания созданных моделей в интернет, сохранение результатов моделирования.

Итак, платформа должна предоставлять все эти возможности, однако остаётся неопределёнными и зависит от конкретной системы целый ряд фактов. Во-первых, это значения перечисленных выше функций. Во-вторых, это свойственные конкретной модели особенные средства визуализации, контроля и управления её характеристиками. Кроме этого, хотелось бы избавиться от замкнутости системы, сделать возможным её взаимодействие с другими программными системами, библиотеками.

В связи с этим было принято решение создать программную библиотеку, предоставляющую средства для разработки и исполнения МАС исследуемого семейства. В библиотеке реализован процесс работы агентов, предусмотрено их распределённое исполнение. Также предоставлены средства управления и контроля над общими для всего семейства свойствами.

Пользователю же библиотеки предоставляется возможность определить вид визуализации, целиком определить связанную с ней структуру объектов и определить функции. После этого, с использованием стандартных средств библиотеки он получает МАС с заданными им свойствами и средства для работы с нею и её изучения.

Для реализации библиотеки было принято решение использовать Java. Этот язык обладает рядом преимуществ при реализации многоагентных и вообще распределенных систем. Во-первых, ни он, ни написанные на нём программы не привязаны к конкретной архитектуре и среде вычислительной системы, что позволяет использовать их практически на любых устройствах. Во-вторых, он имеет мощные встроенные в сам язык средства обеспечения параллельного исполнения и устранения конфликтов. В-третьих, на Java существует множество библиотек, которые мы можем применить при

построении MAC, в частности, агентных платформ, которые можем использовать при совершенствовании нашей библиотеки.

Опишем реализованный нами набор классов, предоставляемые ими возможности и методы, нуждающиеся в конкретизации при создании конкретной реализации MAC на базе нашей платформы. Библиотека классов представляет собой пакет ASwarm с двумя подпакетами ASwarm.grid и ASwarm.parallel.

Основными классами для создания на их основе конкретной системы являются классы: MAC (ASwarm, «рой»), агент (Agent), отношение (Relationship). При этом класс ASwarm имеет абстрактные методы, которые необходимо определить при реализации системы. Абстрактные методы MAC – выбрать агентов для следующего шага, сделать шаг (вычислить G), вычислить характеристическую функцию (h). Библиотеки содержат: отношение типа «шахматная доска» (grid) и соответствующее ему отображение(абстрактный метод – нарисовать 1 агента в клетке) и параллельные (независимые в течение одного шага) агенты и MAC (абстрактные методы – вычислить f (у агента) и вычислить g' (у MAC)). Рассмотрим классы подробнее.

Пакет ASwarm.

ASwarm – абстрактный класс «роя» (MAC). Абстрактные методы – выбрать агентов для следующего шага, сделать шаг (вычислить g), вычислить характеристическую функцию (h). Можно определить свою панель для отображения.

ASwarmFactory – вспомогательный интерфейс фабрики MAC для передачи основной панели MAC.

Agent – агент.

Globals – класс глобальных характеристик MAC, изменяемых в процессе исполнения (время, значение характеристической функции).

LineChart – вспомогательный класс (наследник JPanel) для рисования графика характеристической функции в режиме реального времени

SwarmPanel – главная панель с визуализацией МАС и средствами управления ею. Включена в стандартные формы программы, содержащиеся в библиотеке (окно и апплет). Может быть включена в программу пользователя как независимый модуль.

SwarmFrame – окно, содержащее основную панель визуализации. Может быть создано и вызвано в программе пользователя.

SwarmApplet – абстрактный класс, Java-апплет, содержащий основную панель визуализации. Может быть использован путем создания пользователем класса-наследника и определения им инициализирующего абстрактного метода `init`.

Params – monostate-класс с глобальными параметрами МАС.

Relationship – базовый класс для задания отношения между агентами.

Пакет ASwarm.parallel (для случая независимой работы агентов на каждом шаге, агенты-потoki, исполняемые параллельно)

ParAgent – абстрактный агент, содержащий свой частный поток управления и средства синхронизации. Абстрактный метод – вычислить f .

ParSwarm – абстрактная МАС, содержащая средства синхронизации параллельных агентов в конце каждого шага. Абстрактный метод – вычислить g' (функцию, обеспечивающую переход агентов к новому состоянию с учётом известных полученных значений функции f).

Пакет ASwarm.grid (для задания на агентах отношения соседства на основе окрестности по Муру на «шахматной доске», имеющей топологию тора).

GridPanel – абстрактный класс для отображения агентов в виде множества клеток. Абстрактный метод – нарисовать 1 агент на клетке.

GridRelationship – класс, организующий отношение между агентами. Предоставляет возможность сопоставления агенту места, определения всех соседних агентов и является ли определённая пара агентов соседними.

Рассмотрим составные части основной панели, представленные на Рисунке 2.1.

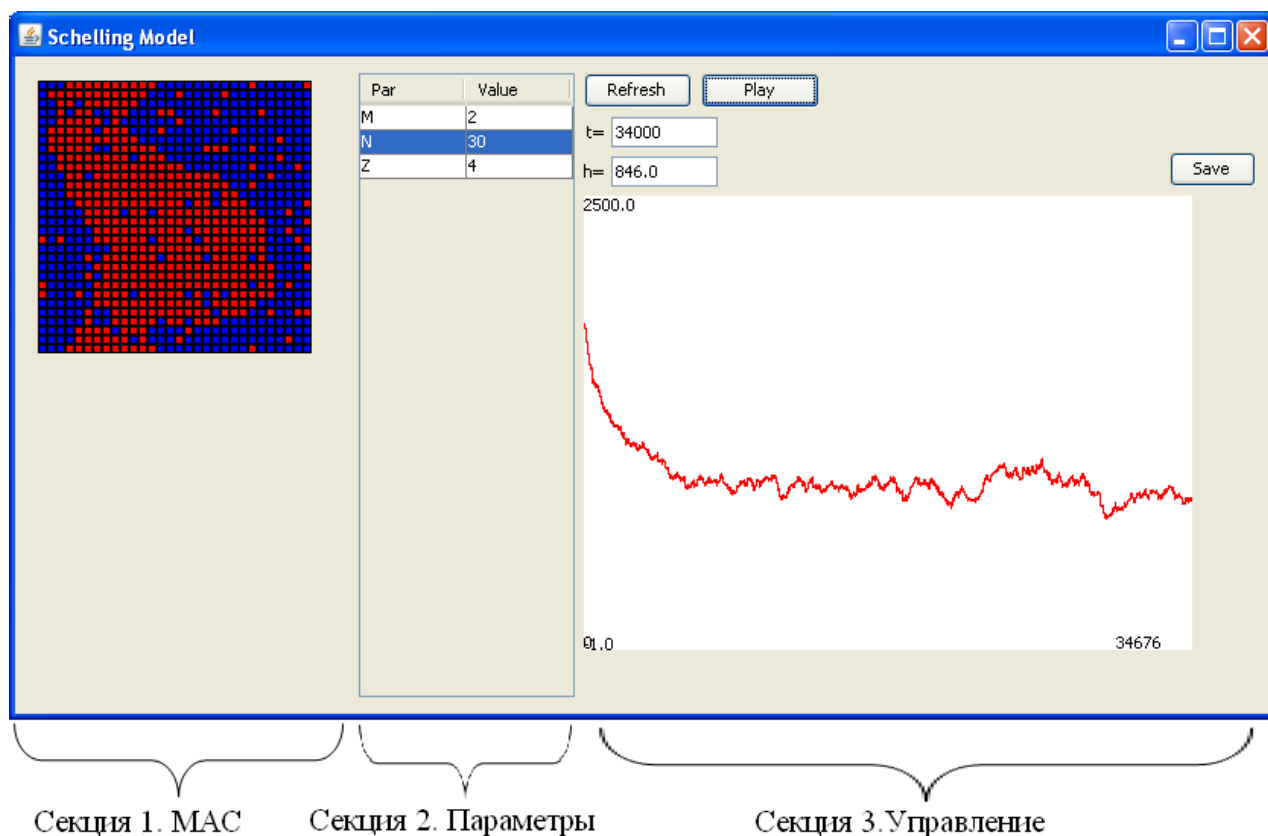


Рисунок 2.1 – Элементы основной панели MASwarm

Секция 1 содержит панель, на которой отображается состояние системы. Поскольку библиотека заранее ничего не знает о структуре системы, то наполнение этой панели происходит при описании каждой конкретной модели. Библиотека содержит как опциональное часто встречающееся отображение модели в виде набора клеток, заполненных агентами.

Секция 2 содержит глобальные параметры системы. При создании конкретной системы пользователь задаёт их количество, обозначение и значение по умолчанию.

Секция 3 содержит основные средства управления процессом работы системы и является общей для всех систем, пользователь может лишь изменить частоту обновления информации на ней. Кнопка Refresh инициирует сброс и создание новой системы. Кнопка Play/Pause инициирует запуск/приостановку работы системы. Панели t и h выводят текущие номер шага и значение характеристической функции. График изображает изменение характеристической функции в зависимости от номера шага. Для подробного изучения работы системы можно сохранить значения характеристической функции с помощью кнопки Save.

Опишем в общем процесс создания модели (позже будут рассмотрены конкретные примеры). Во-первых, необходимо задать систему, определив свои классы МАС и агента, наследованные от базовых. Во-вторых, определить визуализацию МАС для того или иного состояния. В-третьих, воспользоваться тем или иным приёмом для вызова основной панели (библиотека предоставляет 3 возможных способа: окно, апплет, внедрение панели в сторонний интерфейс).

Библиотеку и примеры реализованных с её помощью систем можно найти на интернет-странице проекта [58].

2.4 Примеры использования агентной платформы MASwarm

Модель Шеллинга. В качестве первого примера реализации МАС с роевым интеллектом была выбрана модель Шеллинга в интерпретации Ж.Чжанга (описанная в приложении).

Поскольку средой в данной системе является доска, на которой расставлены фишки-агенты, было принято решение использовать введённые в библиотеку классы, отражающие такое устройство среды (GridPanel и GridRelationship). Зависимость между агентами на каждом шаге, к сожалению, не позволяет использовать параллелизм и связанные с ним классы библиотеки при реализации модели.

У агента были заведены поля для хранения цвета агента и параметров функции полезности (Z и M). Хотя в начальной реализации функция полезности у всех агентов одинакова, их поведение при различных предпочтениях, различной функции полезности, представляет несомненный интерес. Также у класса агентов были заведены методы для вычисления количества «чужих» соседей и текущего значения функции полезности.

Для класса моделей необходимо было определить 4 метода: конструктор, функция выбора агентов для каждого шага, вычисление функции перехода G и характеристической функции h . Конструктор просто создаёт агентов по числу клеток и расставляет их Парето-оптимальным способом (в шахматном порядке). Функция выбора агентов заключается просто в выборе пары не соседних агентов. Вычисление характеристической функции заключается в делении пополам суммы количества «чужих» соседей у каждого агента.

Функция перехода является более сложной. Она состоит из вычисления суммы функций полезности для агентов в случае их текущего расположения и в случае обмена местами, вычисления вероятности осуществления переезда, генерации случайного числа и принятия на его основе решения о переезде.

Кроме этого в классе модели определяется способ её визуализации. Поскольку мы выбрали для этой цели класс `GridPanel`, нам лишь осталось определить визуализацию для каждого отдельного агента. Она заключается в закрашивании клетки красным или синим цветом в зависимости от цвета агента.

Все эти действия определяют модель Шеллинга. Остаётся лишь ввести главный класс (или класс апплета), который будет инициировать визуальное представление модели. Созданный с помощью библиотеки апплет можно увидеть на интернет-странице проекта [67].

Пример процесса моделирования представлен на Рисунке 2.2, это начальное состояние модели.

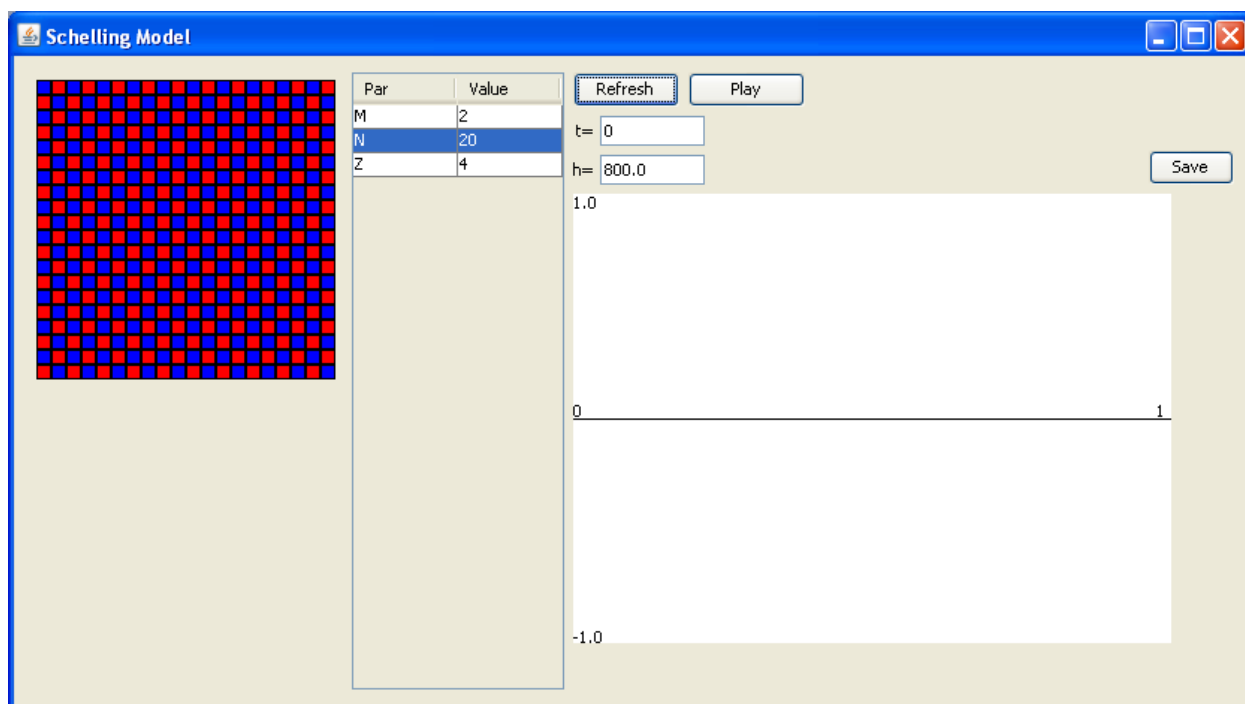


Рисунок 2.2 – Начальное состояние модели

Генетический алгоритм. Нахождение покрывающего кода.

Покажем, что с помощью нашей платформы можно легко создавать не только модели, но и системы, реализующие эвристические алгоритмы решения сложных задач. В качестве алгоритма выберем генетический алгоритм (ГА), а в качестве задачи – задачу нахождения оптимального покрывающего кода из [22].

Это весьма важная задача теории кодирования и заключается она в следующем (сначала опишем её неформально, затем формально). Нам дан многомерный куб. В его вершины надо расставить центры шаров определённого радиуса так, чтобы они покрыли все вершины куба. При этом нас интересует минимальный по мощности набор таких шаров. Нахождение количества шаров, необходимого для покрытия куба с заранее заданными характеристиками является важной и технически сложной теоретической и практической задачей.

$B^n = \{0,1\}^n$ - n -мерный куб. R – радиус шаров.

$X = \{x \in B^n \mid \forall y \in B^n \exists x \in X : d_H(x, y) \leq R\}$ - множество центров покрывающих шаров (искомый набор), где $d_H(x, y)$ - расстояние Хемминга между точками.

$K_2(n, R) = |X|$ - искомое число. Общая формула для нахождения такого числа пока что отсутствует, и может быть вообще не найдена. Некоторые наборы и их мощности найдены теоретически, некоторые практически. На Рисунке 2.3 представлены значения $K_2(n, R)$ для различных n и R , взятые из [22].

n	$R = 1$	$R = 2$	$R = 3$	$R = 4$
1	1			
2	2	1		
3	2	2	1	
4	4	2	2	1
5	7	2	2	2
6	12	4	2	2
7	16	7	2	2
8	32	11 – 12	4	2
9	55 – 62*	15 – 16	7	2
10	105 – 120*	23 – 30*	9-12	4
11	178 – 192	36 – 44	12 – 16	7
12	342 – 380*	61 – 78*	18 – 28*	8 – 12
13	598 – 736*	97 – 128	28 – 42*	11 – 16
14	1172 – 1408*	157 – 256	44 – 64	15 – 28
15	2^{11}	309 – 384*	70 – 112	22 – 40*
16	2^{12}	512 – 768	114 – 192*	33 – 64

Рисунок 2.3 – Значения $K_2(n, R)$

Границы, помеченные звёздочкой, получены с помощью эвристических алгоритмов. Для целого ряда значений характеристик существуют лишь оценочные предположения о покрывающих наборах шаров и их мощности (их текущее состояние можно найти, например, в [50]). Всё это делает рассматриваемую задачу превосходным примером для использования эвристических алгоритмов.

Она может быть представлена как задача минимизации функции [22]

$$f(C) = \left| \{x \in B^n \mid \forall y \in C : d_H(x, y) > R\} \right|. \quad (2.22)$$

Её значение равно количеству непокрытых шарами точек, где S – рассматриваемый набор шаров. Перейдём к терминам генетического алгоритма. Одной отдельной особью будем считать набор шаров. Количество шаров считаем заданным изначально. Геном особи будем считать выписанные подряд двоичные представления центров составляющих их шаров. Поскольку нас интересует не максимум, а минимум функции, видоизменим её соответствующим образом (например, возьмём обратную).

Итак, осталось лишь реализовать алгоритм с помощью нашей платформы. Поскольку агенты могут действовать независимо (вычислять значение целевой функции), то тут целесообразно использовать реализующие параллелизм классы ParAgent и ParSwarm.

Сначала был реализован вспомогательный класс Code, содержащий информацию об одной особи (её генетический код – набор центров шаров) и базовый набор операций с ними (генерация, скрещивание, мутация, преобразование формата).

Затем был введён класс агента. Каждый агент имеет поле, содержащее соответствующий ему набор шаров (объект Code). Осталось определить для агента функцию f – она равна количеству точек, не покрываемых данным набором шаров.

Теперь определим класс, соответствующий МАС («рой»). Для него необходимо определить 4 метода: конструктор, функция выбора агентов для каждого шага, вычисление модифицированной функции перехода G' и характеристической функции h . Конструктор генерирует набор случайных особей. Функция выбора агентов возвращает всех агентов, так как в естественном отборе участвуют все особи (в принципе, это может быть изменено). Характеристическая функция возвращает минимальное значение функции f среди особей текущего поколения.

Модифицированная функция перехода, опираясь на набор вычисленных агентами значений целевой функции, выбирает особей для скрещивания и производит само скрещивание.

Осталось также определить визуализацию текущего состояния модели. Было решено выводить на экран самое минимальное значение, найденное на текущий момент. Кроме этого, при достижении целевой функции нуля, реализована автоматическая остановка работы системы.

Итак, МАС задана, осталось лишь описать главный класс (или апплет). Пример работы алгоритма показан на Рисунке 2.4.

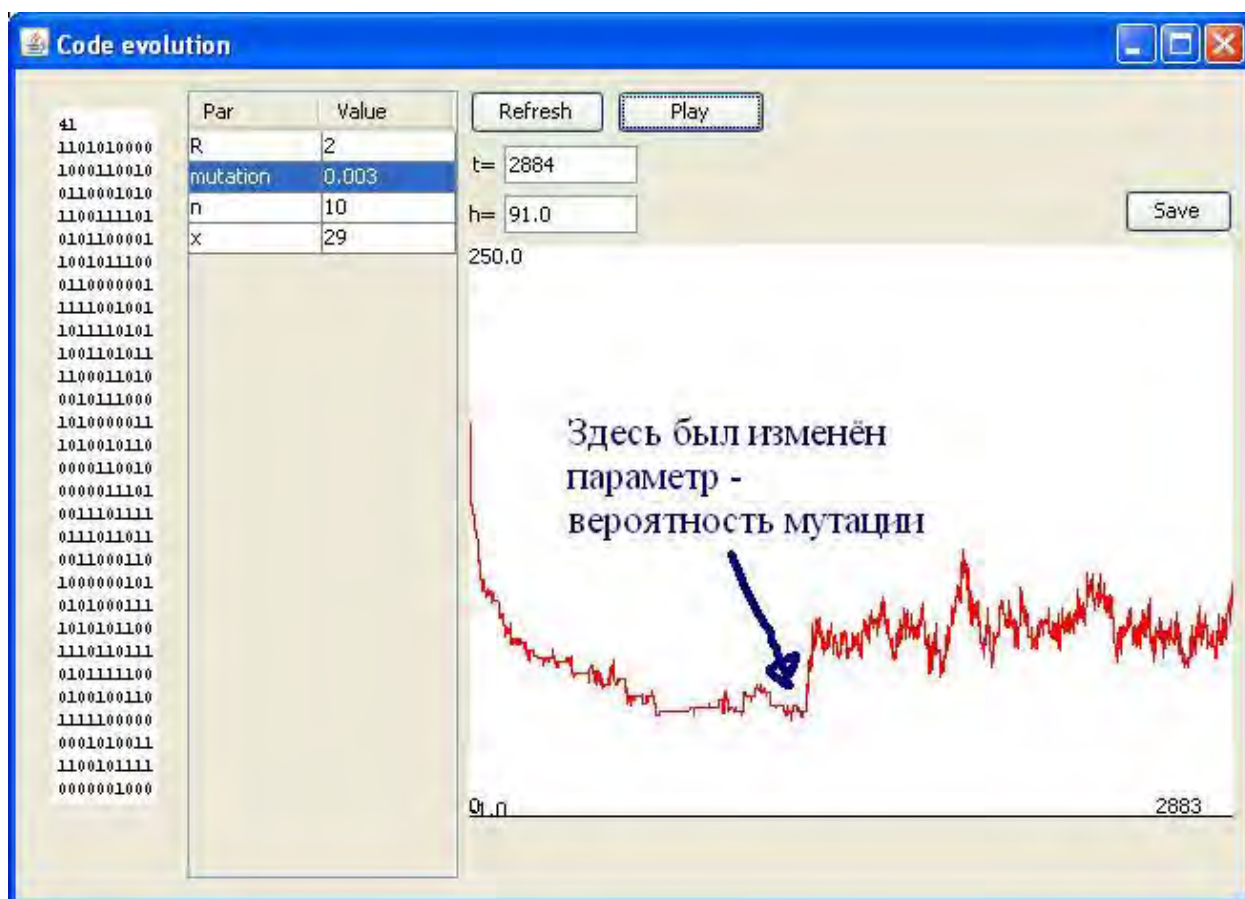


Рисунок 2.4 – Визуализация процесса работы МАС

В процессе работы система была остановлена и изменён параметр, отвечающий за вероятность осуществления мутации. Момент, в который это произошло можно явно отследить на графике характеристической функции системы.

2.5 Сравнение с другими агентными платформами

Как отмечалось в первой главе, в настоящее время уже существует довольно большое количество средств разработки мультиагентных систем (агентных платформ). Однако понятия МАС и агента имеет в настоящее время очень широкую трактовку в общем случае, поэтому каждая конкретная агентная платформа может ориентироваться на то или иное определение, применение, сферу использования МАС. Это неизбежно приводит к тому, что к классу агентных платформ относятся во многом различные системы. Здесь также нашло своё отражение фундаментальное разделение исследования МАС на два ключевых направления – Распределенный искусственный интеллект (РИИ) и Искусственная жизнь (ИЖ)[42]

Одни платформы предполагают, главным образом, организацию распределённого исполнения агентов и предоставляют средства их коммуникации. Другие нацелены на создание большого числа реактивных агентов, построение моделей, предоставляют средства визуализации, задания простых алгоритмов поведения агентов (например, с помощью блок-схем). Ситуация осложняется тем, что многие серьёзные, развитые платформы трудно (или вовсе невозможно) использовать вне рамок их основного направления. Например, с помощью платформы NetLogo, одной из лучших для построения и изучения простых агентных моделей и вообще МАС с роевым интеллектом, принципиально невозможно организовать распределённое или параллельное исполнение программ агентов.

Наша платформа, хотя и относится изначально также к направлению ИЖ, не накладывает существенных ограничений на дальнейшее развитие моделей в большие программные комплексы, поскольку не содержит никаких специфических технических средств и, по сути, использует объектно-ориентированную парадигму. С этим связаны и все особенности нашей системы, отличающие её от других как в лучшую, так и в худшую сторону.

Для сравнения были отобраны развитые и популярные свободно распространяемые агентные платформы Repast Symphony, Jade и NetLogo. Наша система во многом занимает промежуточное положение между ними. Вначале проведём общее сравнение по характеристикам платформ и предоставляемой ими функциональности, а затем перейдём к анализу конкретной системы, реализованной с помощью этих платформ, чтобы сравнить их производительность. Результаты этого сравнения представлены в Таблице 2.1.

Таблица 2.1 – сравнение агентных платформ по функциональности

	Jade	RepastS	NetLogo	MASwarm
Использование распространённого языка программирования (Java)	+	+	-	+
Средства для моделирования	-	+	+	+
Средства для публикации МАС в интернете (в виде апплета)	-	-	+	+
Специализированные средства визуализации	+	+	+	+
Возможность подключать к стороннему проекту в качестве библиотеки	+	-	-	+
Средства создания параллельных программ	+	-	-	+
Средства визуализации алгоритма деятельности агентов	-	+	-	-
Средства создания распределённых программ	+	-	-	-

Как видно, наша система (MASwarm) изначально сочетает многие плюсы отдельных платформ. Не предоставляемая в настоящий момент ею функциональность может быть реализована позднее в виде дополнительных

подключаемых библиотек, в том числе отдельными пользователями. Это выгодно отличает её от таких «замкнутых» систем, как NetLogo.

Перейдём теперь к конкретному примеру МАС. Возьмём рассмотренную ранее модель Шеллинга в интерпретации J.Zhang. Реализуем её с помощью других платформ и измерим время, необходимое на моделирование определённого количества шагов для различной размерности модели.

Библиотека моделей NetLogo содержит модель Шеллинга, однако в несколько другой интерпретации. Для исследования она была модифицирована. Её внешнее представление изображено на Рисунке 2.5.

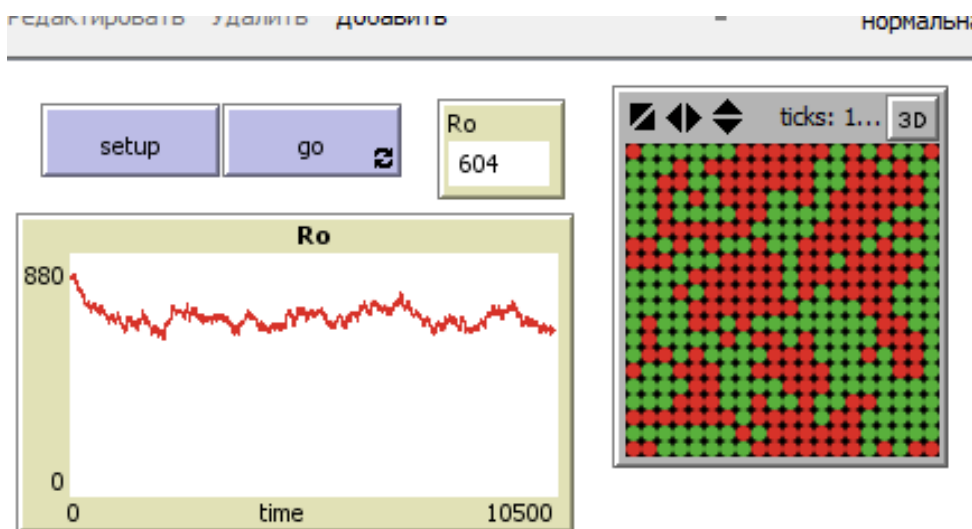


Рисунок 2.5 – Модель Шеллинга в системе NetLogo

В Таблице 2.2 приведено среднее время моделирования с использованием различных средств разработки моделей.

Таблица 2.2 – Среднее время моделирования

Размер поля	Кол-во агентов	Время моделирования в секундах		
		NetLogo	Jade	MASwarm (наша платформа)
10×10	100	10,9	0,8	0,6
14×14	196	14,9	0,9	0,8
20×20	400	28,7	1,8	1,4
30×30	900	61,9	4,5	3,3
40×40	1600	112,7	8	6,2

Из сравнения видно, что наша платформа по времени исполнения существенно опережает NetLogo, и даже несколько быстрее Jade (поскольку последняя по умолчанию ориентирована на распределённое исполнение и расходует часть времени на обеспечение его поддержки). Впрочем, основным преимуществом нашей платформы по сравнению с Jade является не время исполнения, а скорость написания программы, поскольку наша платформа предоставляет средства для создания именно агентных моделей (специализированный пользовательский интерфейс и реализующие общую функциональность абстрактные классы).

Отметим, что для реализации модели на платформе Jade оказалось возможным использовать уже составленный с помощью нашей платформы интерфейс, лишь изменив внутреннюю логику исполнения. В этом также видно преимущество платформ, реализованных в виде подключаемых библиотек и использующих распространённые языки программирования.

3 Модель развития международной транспортной сети

3.1 Постановка задачи

Теперь перейдём к созданию и реализации собственной модели. Здесь представлена модель, разработанная для нужд Института экономики и организации промышленного производства СО РАН по прогнозированию грузопотока в условиях конкуренции операторов перевозок и изменяющегося грузопотока между различными странами.

Экономическая картина мира в последнее время стремительно меняется. Формируются новые экономические центры, возникают совершенно новые экономические связи. Ещё 20 лет назад пришедшее на смену биполярного мира и казавшееся довольно прочным главенство Соединённых Штатов сменяется принципиально новыми схемами устройства мировой хозяйственной системы. Быстрыми темпами развиваются экономики крупных азиатских стран. По валовому внутреннему продукту, рассчитанному по паритету покупательной способности (в постоянных ценах), Китай, обогнав Японию в 2001 году, вышел на второе место в мире, Индия, обогнав Германию в 2006 – на четвёртое, а в 2012 Японию – на третье.[74]

Вместе с увеличением объёмов производства идёт рост грузоперевозок. Список портов по грузообороту на 2008 год возглавляют Сингапур, Шанхай и Гонконг. Развивающиеся порты восточной и юго-восточной Азии в последние годы всё дальше оттесняют такие старинные центры морских перевозок как Роттердам, Гамбург (на 2004 год – 7е и 9е места, на 2008 – 9е и 11е соответственно). [45]

Развитие транспортной сети происходит в тесном взаимодействии с компаниями-операторами, обслуживающими определённые направления грузоперевозок. Всё это делает группы операторов перевозок, действующих на различных участках национальных или международных транспортных

сетей, ключевыми фигурами при развитии того или иного маршрута взаимодействия стран. Также политика операторов в отношении обслуживания участков транспортной сети и установлении тарифов на перевозки может существенным образом повлиять на привлекательность того или иного маршрута, обеспечивающего доставку заданного количества груза в заданном направлении.

Кроме этого, обслуживающие международные грузоперевозки операторы неизбежно ориентируются на обслуживание какой-то определённой страны, связанной с ними, или, возможно, группы стран, связанных международными организациями. Это отражается в ценовой политике, в приоритетности обслуживания грузопотоков между различными странами.

Итак, операторы перевозок на международном уровне функционируют как отдельные конкурирующие экономические субъекты, способные существенным образом повлиять на ход грузопотока по тому или иному маршруту согласно достижению своих собственных целей. Поэтому при определении направлений транспортной сети, могущих стать в будущем существенно задействованными, необходимо учитывать интересы крупных компаний, обеспечивающих перевозки как по этим путям, так и по множеству альтернативных маршрутов, учитывать также национальные интересы операторов при обслуживании грузопотоков между различными странами.

В связи со всем вышесказанным открываются многочисленные проблемы, связанные с развитием транспортной сети страны с учётом использования её в крупных международных грузоперевозках. Поскольку такой ход вещей является одним из возможных сценариев дальнейшего развития транспортной инфраструктуры нашей страны [44], этот круг проблем становится весьма актуальным.

Эти проблемы можно разделить на уровни абстракции, при рассмотрении которых они возникают. Самый низкий уровень абстракции занимают вопросы размещения запланированных грузопотоков с использованием имеющихся ограниченных транспортных средств и путей сообщения, планирование расписания движения поездов, кораблей и пр., отслеживание перемещающихся грузов, динамическое составление расписания.

На высшем уровне рассматриваются вопросы взаимодействия стран, их объединения в коалиции, принятие решений о партнёрстве с той или иной страной. Этот круг проблем характеризуется высокой сложностью, большими трудностями при формализации деятельности даже отдельных участников, не говоря о системе в целом.

Рассматриваемая же проблема относится к промежуточному уровню. Во-первых, мы абстрагируемся от перемещения дискретных единиц груза, рассматривая грузооборот как непрерывный поток в течение определённого промежутка времени. Во-вторых, мы не пытаемся построить модель развития группы стран. На рассматриваемом нами уровне абстракции существуют транспортные сети, их основные направления, операторы, эти направления обслуживающие и решения о направлении определённого грузопотока от одной страны к другой. Данный уровень в настоящее время исследуется всё более подробно, создаются модели для его анализа [29].

Проблема состоит в том, чтобы определить, какие направления окажутся наиболее задействованными, если речь идёт о развитии транспортной сети в условиях конкуренции между операторами перевозок, в условиях изменяющихся отношений между странами, роста (или падения) грузооборота по определённым направлениям. Также важно определить, что нужно сделать, чтобы заданные направления, маршруты (например, включающие в себя элементы российской транспортной инфраструктуры) оказались востребованными в таких условиях. На основании полученных

результатов можно будет определить, какие решения нужно принимать на уровне государства и отдельных операторов, например, какие направления нуждаются в модернизации, какие в скидках на тарифы, какие операторы нуждаются в государственной поддержке в первую очередь.

Для решения такой проблемы необходимо понять, как происходит развитие транспортной сети, а значит, необходимо разработать модель развития транспортной сети с учётом конкуренции операторов, их национальной принадлежности и с учётом динамически изменяющейся картины мира и взаимоотношений между странами.

Для разработки такой модели было принято решение использовать имитационное моделирование, а именно, агентный подход. Во-первых, это весьма современное и перспективное направление в моделировании, во-вторых, оно адекватно решаемой проблеме, поскольку в нём тоже основными элементами являются автономные активные сущности, находящиеся в сложной среде и взаимодействующие с ней и друг с другом каждый для достижения собственных целей.

Кроме этого, в нашу задачу входит формализация и реализация разработанной модели с тем, чтобы отладить её, проверить её адекватность указанной проблеме на множестве реальных примеров.

3.2 Описание модели

Агентное моделирование применяется для решения целого ряда проблем, связанных с прогнозированием и оптимизацией транспортных потоков. Известны модели, специализирующиеся на логистических задачах [2]. Также существуют модели более высокого уровня абстракции, целью которых является прогнозирование будущего грузопотока в заданном направлении [29]. Рассматриваемая же нами модель относится к промежуточному уровню абстракции. Приблизительный прогноз грузопотоков по отдельным направлениям (составленный с помощью

экспертных оценок или модели более высокого уровня) входит в набор её исходных данных, на выходе же мы получаем объём грузопотока, проходящего по тому или иному конкретному маршруту.

Обоснование модели и её общее описание дано в работах [15-17], модель применяется в Институте экономики и организации промышленного производства Сибирского отделения Российской академии наук. Здесь мы приведём краткое математическое описание модели.

Рассмотрим множество предприятий P . Обозначим грузопоток между предприятиями $s_t(a, b)$, где $a, b \in P$. Предприятия распределены в пространстве, однако существуют центры притяжения, отдельные экономические районы, территориально-производственные комплексы (ТПК), в которых группируются предприятия. В контексте государственной или международной транспортной сети они являются центрами формирования грузопотока. Введём множество таких центров C .

Определим отношение принадлежности того или иного предприятия тому или иному центру формирования грузопотока $R \subseteq P \times C$. Перемещения грузов между предприятиями и домохозяйствами одного центра считаются в рамках нашей модели пренебрежимо малыми. Таким образом, нас интересует товарооборот между предприятиями различных центров.

Определим агрегированный грузопоток между различными центрами как сумму грузопотоков отдельных предприятий этих центров $s_t(c_1, c_2) = \sum_{a|aRc_1, b|bRc_2} s_t(a, b)$. Изначальной целью каждого предприятия является минимизация стоимости (или некоторого взвешенного показателя, учитывающего стоимость, скорость, качество и т.д.) перевозок. Агрегированной целью в таком случае будет минимизация стоимости обеспечения агрегированного грузопотока.

Для осуществления перевозки грузов центры должны быть связаны между собой транспортной сетью. Сеть состоит из узлов (транспортно-логистических центров, ТЛЦ) и путей сообщений между ними. Элементы сети обладают множеством параметров, главные из которых для нас – стоимость перевозки единицы груза и пропускная способность. За формирование, установление значений этих параметров отвечают крупные компании, обслуживающие транспортные сети. Здесь, как и в случае с грузопотоком, мы будем рассматривать агрегированные агенты – операторы перевозок, обслуживающие отдельные участки, звенья транспортной сети.

Мы будем использовать обычное для многих задач представление транспортной сети в виде графа. Однако у этого графа будет ряд особенностей. Будет два различных типа вершин, представляющие ТПК как источники формирования грузопотока и ТЛЦ как перевалочные центры, не формирующие грузопоток, но обеспечивающие его перевалку и направления на те или иные пути сообщения. Рёбра, соединяющие ТЛЦ между собой или с источниками грузов будут представлять те или иные пути сообщения. Рёбра же, соединяющие источники грузов друг с другом будут символизировать грузопоток, который необходимо перевезти из одного района в другой с использованием заданной транспортной сети, поскольку мы рассматриваем случай, когда центры производства разнесены в пространстве и связываются не непосредственно, а с участием различных крупных логистических центров – портов, перевалочных пунктов и т.п.

За основу была взята следующая схема. Во-первых, дана сеть дорог между странами. На дорогах находятся узловые пункты, представляющие собой развязки и транспортно-логистические центры. Кроме этого, дороги и центры обслуживаются различными операторами перевозок. На уровне стран принимается решение относительно объёма планируемых перевозок между ними. В связи с этим возникает задача распределение объёма всех перевозок между всеми странами по реально существующей транспортной сети.

При этом объектом рассмотрения становится поведение операторов в процессе распределения перевозок по транспортной сети. Они получают право управлять различными параметрами дорог и узлов, стараясь создать, таким образом, более выгодные для себя маршруты перевозок между странами. Такими параметрами могут быть стоимости перевозок и пропускные способности дорог.

При распределении потоков перевозок между странами по транспортной сети учитываются, во-первых, характеристики дорог, во-вторых, скидки, предоставляемые операторами при использовании их маршрутов, в-третьих, принадлежность стран и операторов различным наднациональным организациям. На основании этих параметров выбираются оптимальные маршруты среди возможных.

Сами потоки перевозок между странами не остаются неизменными. Но поскольку их планирование и изменение происходит не на уровне транспортной сети, а на уровне стран и международных организаций, что не относится к моделированию именно с позиции транспортной сети, решено было не моделировать это изменение, а либо загружать данные об объёме перевозок в каждый момент времени из внешней среды, либо генерировать их случайно, либо же использовать смешанный вариант.

Изначально строится вариант для одного типа грузов, затем возможно его расширение на несколько типов грузов, при котором все характеристики транспортной сети и объёмы перевозок между странами задаются для каждого типа грузов в отдельности и между ними возможна взаимозависимость.

Поскольку в процессе моделирования нам необходимо создать программные агенты, постольку надо, во-первых, определить основные виды агентов, во-вторых, указать их цели и, в-третьих, выделить и формализовать их основные способы воздействия на среду. Назовём эти программируемые способы воздействия или элементы деятельности агентов механизмами

агентов. С одной стороны, агентами являются компании-операторы. С другой стороны, агентами являются планируемые грузопотоки между странами.

Цель потоков – распределить оптимальным образом планируемый объём груза по транспортной сети. При этом активность потока заключается в таких двух элементах (механизмах): **Изменение объёма планируемого потока между странами** и **Распределение этих потока по сети**.

Цель каждого оператора – добиться увеличения показателя, соответствующего его благосостоянию (денежного запаса). Для этого у него есть следующие механизмы: **Изменение пропускных способностей дорог** и **Изменение стоимостей перевозок**. Кроме этого, вспомогательными (техническими) механизмами являются: **Вычисление текущего показателя благосостояния** и **Принятие особых решений в связи с текущим показателем благосостояния** (например, прекращение деятельности оператора при опускании показателя ниже нуля).

Регулируемыми параметрами дорог являются *Цена* (стоимость перевозки единицы груза) и *Объём* (он же – *пропускная способность*). Изначально задаваемым параметром является *Себестоимость* дорог (себестоимость перевозки единицы груза). Параметры потоков – *Цена* (приоритет при распределении по сети) и *Объём*. Параметр оператора – *денежный запас*.

Процесс моделирования в целом заключается в следующем. Каждый планируемый грузопоток собирает информацию о структуре транспортной сети, а также о стоимостях перевозок, остаточных пропускных способностях и операторах перевозок для каждой дороги. На основании этих знаний поток планирует своё размещение по транспортной сети оптимальным образом. В соответствии с этим размещением поток увеличивает значение текущего потока для каждой дороги и уменьшает её остаточную пропускную способность. После этого поток изменяет свою величину случайным образом согласно заданной тенденции.

Операторы собирают данные о потоке, пришедшемся на обслуживаемые ими дороги, вычисляют для каждой своей дороги получившуюся относительную загруженность. Затем каждый оператор включает все свои указанные механизмы, преследуя цель увеличить в будущем свой денежный запас. Принятые решения отражаются на пропускных способностях и стоимостях перевозок для различных дорог, а также на внутреннем состоянии операторов.

Представим основные действия агентов в строгом математическом виде. Сначала определим основные объекты и термины.

- C – страны (источники и приёмники потоков грузов в транспортной сети)
- N – узлы (остальные вершины в графе транспортной сети)
- $V = C \cup N$ – вершины графа транспортной сети
- Op – операторы
- $E \subseteq V \times V \setminus C \times C$ – дороги (любые пути сообщения, соединяющие узлы со странами или между собой)
- $S \subseteq C \times C$ – потоки (планируемые грузопотоки между странами, которые надо пустить по транспортной сети)
- $T = N_0$ – моменты времени

Определим, далее, на них основные функции:

- $p: T \times E \rightarrow \mathbb{R}$ – стоимость (стоимость провозки единицы груза по дороге)
- $c: E \rightarrow \mathbb{R}$ – себестоимость (затраты оператора на провозку единицы груза по дороге)
- $v: T \times E \rightarrow \mathbb{R}$ – объём (пропускная способность дороги в единицах груза)
- $cv: T \times E \rightarrow \mathbb{R}$ – пущенный поток (грузопоток, проходящий по дороге в данный промежуток времени)

- $op: E \rightarrow Op$ – оператор, обслуживающий данную дорогу
- $m: T \times Op \rightarrow \mathbb{R}$ – денежный запас оператора
- $sp: S \rightarrow \mathbb{R}$ – цена потока (приоритет при его распределении по транспортной сети)
- $sv: T \times S \rightarrow \mathbb{R}$ – объём потока (величина планируемого грузопотока между странами в данный промежуток времени в единицах груза)
- $\tau: T \times S \rightarrow \mathbb{R}$ – тенденция потока (среднее увеличение потока за единицу времени, рассчитанное на основе прогнозных данных значения объёма потока через определённое время)

Введём вспомогательные функции, основанные на предыдущих:

- $z: T \times E \rightarrow \mathbb{R}$ – относительная загруженность дороги

$$z_t(e) = \frac{cv_t(e)}{v_t(e)} \quad (3.1)$$

- $E(op) = \{e | op(e) = op\}$ – множество всех дорог, обслуживаемых данным оператором

Теперь определим в этих терминах основные действия агентов.

Агент оператор, изменение пропускных способностей дорог:

- $Echv(t, op) \subseteq \{e | op(e) = e \ \& \ z_t(e) > zchv\}$ – множество дорог, пропускную способность которых надо изменить
- $\forall t \forall op, \forall e \in Echv(t, op) \ v_{t+1}(e) = v_t(e) + \Delta_v$ – изменение объёма
- стоимость изменения объёма дороги

$$Mchv(t, op) = \sum_{e \in E(op)} c(e) \times \Delta_v \quad (3.2)$$

- $m_t(op) - Mchv(t, op) \geq 0$ – ограничение по затратам, затраты не должны превышать денежный запас
- $zchv, \Delta_v$ – параметры (*Критическая величина загруженности, при превышении которой рекомендуется изменять пропускную способность и Величина, на которую увеличивается пропускная способность дороги*)

Агент оператор, изменение стоимостей перевозок:

- $EchprL(t, op) = \{e | z_t(e) < zL\}$ – недогруженные дороги
- $EchprH(t, op) = \{e | z_t(e) > zH\}$ – сильно загруженные дороги
- $\forall t \forall op, \forall e \in EchprL(t, op) \ p_t(e) = \max(c(e), p_t(e) - \Delta_p)$ – уменьшение стоимости не ниже себестоимости
- $\forall t \forall op, \forall e \in EchprH(t, op) \ p_t(e) = p_t(e) + \Delta_p$ – увеличение стоимости
- zL, zH, Δ_p – параметры (*Высшая критическая величина загруженности для изменения цены, Низшая критическая величина загруженности для изменения цены и Величина, на которую увеличивается цена дороги*)

Агент оператор, вычисление прибыли

- величина прибыли:

$$\forall t \forall op \ Mpr(t, op) = X \times \sum_{e \in E(op)} (p_t(e) - c(e)) cv_t(e) \quad (3.3)$$

- $X \sim N(1, \sigma_{pr})$ – случайная величина с нормальным распределением с заданными параметрами
- σ_{pr} – параметр (*Стандартное отклонение прибыли*)

Агент оператор, вычисление текущего денежного запаса

- $\forall t \forall op \ m_{t+1}(op) = m_t(op) + Mpr(t, op) - Mchv(t, op) - cons$ – формула для расчёта денежного запаса
- $m_t(op) \geq 0$ – ограничение по запасу. Обеспечивается так:
- $m_t(op) + Mpr(t, op) - Mchv(t, op) - cons \Rightarrow m_{t+1}(op) = subs$
- $cons, subs$ – параметры (*Величина потребления* и *Величина субсидии*)

Агент поток, изменение объёма

- $\forall t \forall s \ sv_{t+1}(s) = sv_t(s) + Y$
- $Y \sim N(\tau_t(s), \sigma_{chv})$ – случайная величина с нормальным распределением, определяемая тенденцией изменения потока
- σ_{chv} – параметр (*Стандартное отклонение запланированного изменения объёма потока*)

Агент поток, размещение по транспортной сети

- σ_{chv} – параметр (*Стандартное отклонение запланированного изменения объёма потока*)
- $f: T \times S \times E \rightarrow \mathbb{R}$ – вспомогательная функция, величина пущенного потока по данной дороге для данного грузопотока между странами
- $\forall t \forall s \ f_{t,s}(u, v) = -f_{t,s}(v, u)$ – прямой и обратный потоки взаимно противоположны
- $s = (v_1, v_2)$ – обозначение вершин грузопотока
- Требование сохранения потока

$$\forall v \in V \setminus \{v_1, v_2\} \sum_{u \in V | (u,v) \in E} f_{t,s}(u, v) = 0 \quad (3.4)$$

- Объём потока

$$\sum_{u \in V | (v_1, u) \in E} f_{t,s}(v_1, u) = sv(s); \quad \sum_{u \in V | (u, v_2) \in E} f_{t,s}(u, v_2) = sv(s) \quad (3.5)$$

- Недопустимо превышение пропускной способности

$$\forall t \forall e \sum_{s \in S} |f_{t,s}(e)| \leq v_t(e) \quad (3.6)$$

- Суммарная стоимость перевозок для потока

$$L(t, s, f) = \sum_{e \in E} \max(f_{t,s}(e), 0) \quad (3.7)$$

- Оптимальные пути для каждого потока, согласно приоритету

$$\forall t \forall s \quad f | L(t, s, f) = \min_{\forall f_{t,s}; \forall f_{t,s'}, sp(s') < sp(s); \widetilde{f}_{t,s}, sp(s') > sp(s)} L(t, s, f) \quad (3.8)$$

- Для потоков, равных по приоритету удовлетворяется требование равновесия Нэша (невозможно достичь более дешёвого решения без изменения решений для других потоков).

Для реализации модели с помощью нашей платформы необходимо, как и для рассмотренных ранее примеров, определить свои классы агентов, МАС и задать визуализацию. Агенты делятся на 2 типа которым соответствуют 2 различных класса (операторы и грузопотоки). Отношение между ними имеет вид взвешенного графа, где каждому ребру сопоставлен ряд характеристик (пропускная способность, грузопоток и т.п.). Конструктор МАС создаёт набор агентов и связей между ними, метод выбора агентов выбирает поочерёдно то всех операторов, то все грузопотоки, характеристической функции в явном виде нет, однако мы можем рассматривать в качестве неё состояние интересующего нас оператора.

Функция перехода заключается в поэтапном осуществлении деятельности составляющих модель агентов – планировании грузопотока, формировании инвестиционной политики и т.п. В следующей главе рассматривается использование данной модели в рамках программного

комплекса, и там более подробно описан её программный интерфейс и нюансы устройства.

3.3 Описание программного комплекса

Мы предложили и реализовали модель развития транспортной сети, пользуясь нашей агентной платформой. Однако модель для её практического использования в качестве экономического инструмента требует доработки, превращения в полноценный удобный программный комплекс, помимо собственно модели содержащий средства управления ею и дополнительные обслуживающие модель (собирающие для неё данные и обрабатывающие результаты моделирования) программные модули.

Это потребует, во-первых, создание полноценной программной системы, использующей описанные в предыдущей главе классы, имеющей удобный пользовательский интерфейс. Средства визуализации, предоставляемые нашей библиотекой, подходят для калибровки и оттачивания модели, но становятся явно недостаточными при переходе к её практическому повседневному использованию исследователями. Однако наша платформа изначально предполагает такое развитие событий и предоставляет возможность легко перейти к использованию одного только внутреннего механизма МАС, отказавшись от стандартной визуализации. Так же без особого труда при необходимости дальнейшей калибровки модели можно снова перейти к стандартной визуализации, менее привязанной к структуре модели.

Во-вторых, для создания программного комплекса необходимо реализовать отдельную систему обработки данных, как результатов моделирования, так и исходных параметров модели. Моделирование само по себе составляет лишь часть исследования. Другая важная и ответственная часть состоит из анализа данных, относящихся к политико-географическим единицам, их обработки, нормировки и визуализации. Необходимость построения такой системы обоснована в параграфе 3.4. Пока что лишь

заметим, что разработка собственной системы, кроме всего прочего, позволит нам осуществить её программное взаимодействие с моделью.

Для реализации каркаса модели и последующего написания рабочей версии модели было принято решение использовать язык программирования Java и связанные с ним технологии. Кроме того, что это обеспечивает проект с точки зрения поддержки и развития, унифицирует его компоненты, делает возможным составление его схемы с помощью наглядной диаграммы классов, выбор Java имеет также большое значение в контексте агентного моделирования. Дело в том, что, ввиду открытости платформы Java, большое количество агентных платформ, как промышленных, так и научных, построено на ней и для неё, например, JADE, Repast, Swarm, Eclipse AMP, AnyLogic. Это позволяет на этапе написания каркаса не привязываться жёстко к механизмам реализации моделирования, и в будущем даже переходить по мере необходимости и расширения модели от одного к другому.

Благодаря использованию технологии Java становится возможным построение модульного каркаса модели, независимого как от средств визуализации, так и от конкретных механизмов, использующихся при моделировании. Это достигается путём отделения визуализации от модели и разделения самой модели на часть, отвечающую за состояние и структуру мира, и часть, отвечающую за деятельность, направленную на изменение мира. Миром модели является транспортная сеть определённой структуры.

Вначале была реализована функциональность, предоставлявшаяся ещё прототипом. Однако был внесён ряд небольших изменений в интерфейс программы, поскольку применение низкоуровневых (по сравнению с прототипом) средств программирования пользовательского интерфейса сделало возможным повышение удобства его использования и его настройку в зависимости от предпочтений пользователя. Внешний вид системы представлен на Рисунке 3.1.

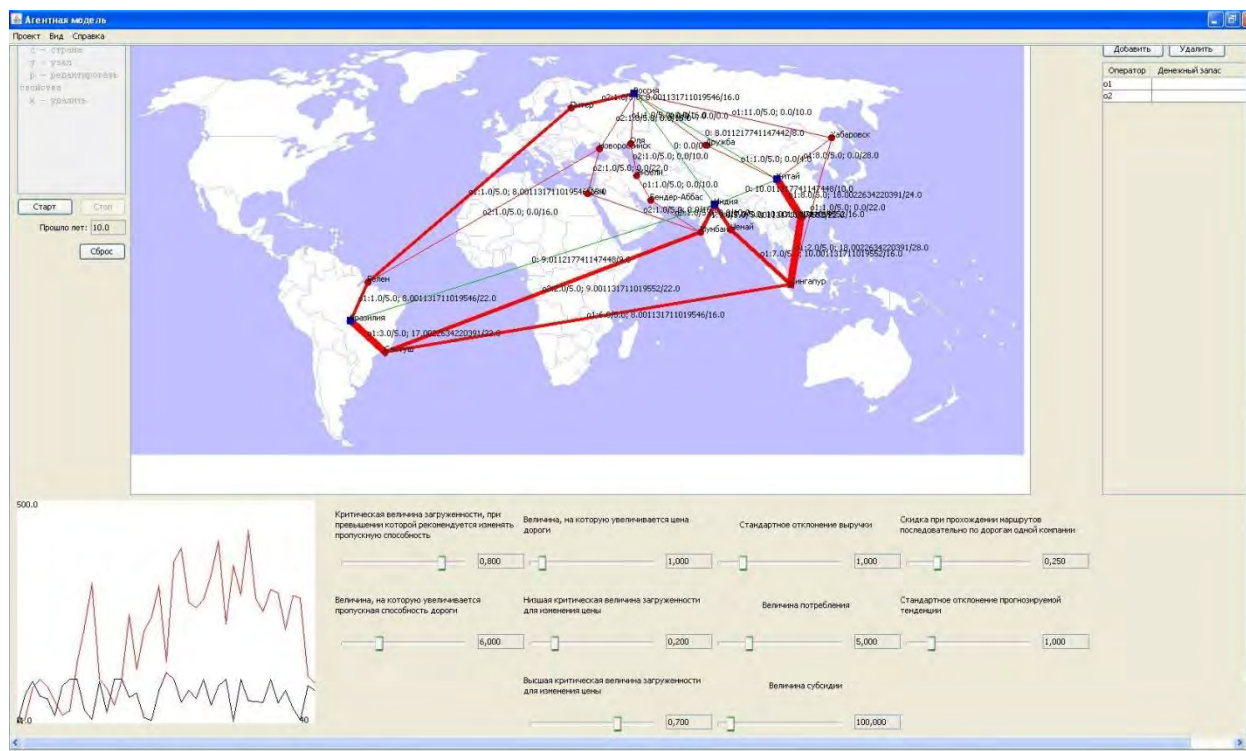


Рисунок 3.1 – Внешний вид системы моделирования

Ключевым понятием для программного комплекса является Проект. Проект состоит из географической карты, на которой задаётся модель, определения базовых множеств модели, элементы которой привязаны к определённым местам на карте, а также базовых функций на этих множествах. Работа с программой разделяется на два процесса, которые могут чередоваться: задание/изменение модели и её исследование. Рассмотрим эти процессы подробнее.

Для задания модели нам необходимо создать новый проект, установить используемую карту и задать модель. Для создания/сохранения проекта и установки карты используются соответствующие пункты меню «Проект». Для задания модели используется сочетание управления с помощью мыши и клавиатуры. Чтобы описать его, определим сначала некоторые понятия, связанные с моделью и её составными частями.

Базовые множества модели представляют собой объекты двух типов: точечные (ТЛЦ и центры формирования грузопотока) и линейные (пути сообщения и направления). Центры формирования грузопотока

обозначаются с помощью синих квадратов, ТЛЦ – с помощью красных кружков. Пути сообщения обозначаются с помощью красных линий, грузопотоки – зелёных.

Управление мышью заключается в следующем.

- 1) При нажатии определённой клавиши на клавиатуре в месте нахождения курсора создаётся (или удаляется) тот или иной объект:
 - клавиша “с” – «страна» (центр образования грузопотока, ТПК);
 - клавиша “у” – «узел» (перевалочный узел, ТЛЦ).
- 2) Все дальнейшие действия представляют собой либо редактирование/удаление этих базовых объектов и их свойств, либо создание новых объектов на их основе; при этом всякий раз при взаимодействии с объектом берётся ближайший к курсору;
- 3) Чтобы соединить два объекта, необходимо зажать левую клавишу мыши вблизи одного объекта, при этом появится соединительная линия, которую надо дотянуть до другого объекта и после соединения с ним отпустить левую клавишу мыши. При этом, если оба объекта страны, то создастся объект-поток; иначе – объект-путь сообщения.
- 4) Чтобы переместить точечный объект, необходимо зажать правую клавишу мыши вблизи объекта и перетянуть его на нужное место, после чего отпустить правую клавишу мыши.
- 5) Чтобы удалить объект, необходимо подвести к нему указатель и нажать клавишу “х”.
- 6) Чтобы редактировать свойства объекта, необходимо два раза кликнуть вблизи него левой клавишей мыши.

Свойства точечных объектов – название (оно отображается рядом с ними)

Свойства путей сообщения – обслуживающий оператор, пропускная способность, себестоимость перевозок. Они обозначаются на карте над объектом следующим образом:

“оператор”: “цена”/”себестоимость провозки единицы груза”;
“грузопоток, проходящий в данный момент”/”предельная пропускная способность”.

Свойства потоков – нынешний и планируемый грузопоток, приоритет направления. Они обозначаются на карте над объектом следующим образом:

“приоритет”: “нынешняя величина грузопотока”/”планируемая через 10 лет величина грузопотока в данном направлении”

Для удобства предусмотрены 3 режима просмотра с различной фильтрацией отображаемой информации (полный режим, «страны и потоки», «дороги»), управление ими происходит с помощью меню «Вид». В полном режиме отображаются все элементы модели. В режиме «страны и потоки» отображаются центры формирования грузопотоков («страны») и грузопотоки между ними. В режиме «дороги» отображаются все пути сообщений и связываемые ими ТЛЦ и ТПК (то есть, скрытыми оказываются грузопотоки).

В правой верхней части экрана находится список операторов. Его можно редактировать:

- добавлять/удалять операторов грузоперевозок с помощью соответствующих кнопок, расположенных над списком
- изменять их названия и денежный запас в начальный момент, для перехода в раздел редактирования соответствующих полей используется двойной клик левой клавиши мыши.

Внизу в левой части окна расположена автоматически обновляющаяся линейная диаграмма, иллюстрирующая изменение денежного запаса операторов в процессе моделирования. Каждому оператору соответствует линия определённого цвета.

Внизу в центральной части окна расположены бегунки (слайдеры), использующиеся для изменения параметров модели. Текущее значение параметра отображается в окошке слева от бегунка, название изменяемого параметра располагается над бегунком.

Посередине в левой части окна располагаются инструменты управления и мониторинга процесса моделирования. Они включают в себя кнопки запуска/остановки моделирования, кнопку сброса к начальным условиям и поле вывода, отображающее текущее время модели.

В процессе моделирования толщина линий, соответствующих путям сообщения изменяется пропорционально объёму проходящего по ним грузопотока. Пути, загрузка которых приближается к предельной, окрашиваются в жёлтый цвет. Также линии, соответствующие грузопотокам между странами, загрузка путей при осуществлении перевозок которых подошла к критической, утолщаются. Все изменения параметров и значений функций модели логируются в выходном потоке программы и могут быть сохранены в файл стандартными средствами.

Итак, мы описали предоставляемые программным комплексом возможности по построению и исследованию конкретных моделей. Последовательность действий (не строгая, но примерная) при этом такова: создание проекта, задание карты, элементов и параметров модели. Затем запуск и остановка процесса моделирования, сброс и повторное возвращение к изменению элементов и параметров модели.

В последующих параграфах мы покажем два примера использования программного комплекса для моделирования развития транспортных сетей в условиях конкуренции различных маршрутов. Вначале рассмотрим более простой пример (с небольшим количеством агентов и маршрутов), затем перейдём к более сложному.

3.4 Программный комплекс для анализа и визуализации результатов моделирования

Второй составляющей программного комплекса является система анализа и визуализации результатов моделирования. Она может быть использована также и для подготовки новых входных данных для модели. Модель имеет дело с данными, характеризующими экономические и социальные показатели крупных географических и политических объектов – государств и их крупных административно-территориальных единиц, участвующих в формировании грузопотока. В связи с этим для наглядной интерпретации и статистического анализа как результатов модели для их последующей визуализации, так и данных государственной и международной статистики для определения параметров модели, прогнозных характеристик, тенденций, экономистам понадобился отдельный инструмент.

Как правило, для подобных систем существуют 2 крайности. Статистические данные предоставляются либо в сыром виде (например, в таблице в файле Word), не подготовленном для автоматической обработки и анализируются вручную, либо предоставляются системами визуализации в виде уже готовых диаграмм, таблиц, карт. Многие такие системы расположены на сайтах крупных государственных и международных организаций (Всемирный Банк, МВФ, Росстат), а также журналов.

Однако готовые решения не обладают необходимой для исследователей гибкостью. Они хороши для общего ознакомления с геостатистической информацией или как источник данных, но анализ данных и визуализацию изменённых или произвольных пользовательских данных с их помощью провести нельзя. Это вынуждает либо пользоваться дорогими специализированными статистическими программными пакетами (которые могут также не содержать нужных средств анализа данных), либо для каждой задачи строить отдельную программу (макрос, набор формул), обрабатывающую исходные данные.

Ещё одна проблема – отсутствие отдельно работающего, удобного, автономного механизма визуализации географических данных. Как правило, системы создания картографических диаграмм (map charts) являются составной частью более сложных систем. Ранее такую возможность предоставлял Microsoft Excel, однако в настоящее время она не включается в стандартную сборку. Другие системы предоставляют возможность лишь уже содержащихся в них, а не введённых пользователем данных. Третьи требуют низкоуровневого взаимодействия (GoogleMaps).

Итак, всё это обуславливает необходимость создания программной системы, объединяющей 3 основные задачи: извлечение данных из файлов пользователя, их гибкий анализ с возможностью использования специализированных статистических функций и нормировок, визуализация как результатов анализа, так и произвольных исходных данных. Опишем сначала процесс работы с программой, определим основные термины, а затем перейдём к описанию её внутреннего устройства и расскажем, с помощью каких технологий она построена.

Описание работы с программой. Работа с программой осуществляется в рамках проекта. Проект подразумевает совокупность исходных данных, алгоритма их обработки и анализа и описание визуализации. Проект идентифицируется названием и может содержать подробное описание. Проект может быть сохранён в файл и впоследствии прочитан.

В работе с программой можно выделить два блока связанных операций: 1) подготовка данных, формирование выборки; 2) настройка визуализации и её сохранение. Для визуализации изначально должна быть сформирована выборка, однако в дальнейшем можно снова возвращаться к работе с ней, изменять её. Рассмотрим эти блоки в отдельности.

Выборка как сущность представляет собой трёхмерный куб данных, осями которого являются параметры, регионы и годы. Сами данные являются

числами, характеризующими значение определённого параметра для определённого региона за определённый год. Рассмотрим оси в отдельности.

Параметры (например, численность населения, среднемесячная зарплата и т.п.). Параметры разбиваются на две группы. Параметры, содержащиеся в БД выбираются из множества параметров, хранящихся в БД. Расчётные параметры задаются пользователем и вычисляются на основе выбранных из БД с помощью алгебраических и статистических формул. Каждый параметр имеет название (описание параметра на русском языке, использующееся при визуализации) и обозначение (краткое латинское обозначение параметра, используемое в формулах для расчётных параметров). Обозначение можно изменять как у расчётных параметров, так и у параметров из БД. Название можно изменять только у расчётных параметров, у параметров из БД оно фиксировано и задаётся в процессе непосредственной работы с БД.

Каждому расчётному параметру также поставлена в соответствие формула, на основании которой он вычисляется. Формула представляет собой математическое выражение, переменными в котором являются обозначения других параметров, а операторами – знаки «плюс», «минус», «умножить», «разделить», «возвести в степень» («^») и специальные функции. Функции имеют вид «\$имя_функции(список_параметров)». Список параметров представляет собой перечисление параметров, разделённых знаком «;».

Имя функции является одним из допустимого набора функций. Пока что функции не могут участвовать в выражениях. Каждая отдельная формула может содержать либо функцию, либо алгебраическое выражение. Совокупность параметров (то есть, имён и обозначений для всех параметров и формул для расчётных параметров) называется алгоритмом расчёта. Он может быть сохранён отдельно в виде файла с расширением «alg» в формате csv и после считан. Ниже, в Таблице 3.1 приведён перечень функций.

Таблица 3.1 – Обозначения и названия функций

Функции от одного параметра:	
\$rang	ранжирование (больше параметр - больше ранг)
\$rango	обратное ранжирование (больше параметр - меньше ранг)
\$sum	сумма
\$min	минимум
\$max	максимум
\$avg	среднее
\$otkl	стандартное отклонение
\$no_mr	нормировка Минрегиона
\$no_vb	нормировка Всемирного Банка
\$no_z	нормировка Z (вспомогательная, для метода Хельвига)
Функции от произвольного числа параметров:	
\$no_hel	нормировка Хельвига
\$no_m	нормировка М

1. **Регионы.** Регионы являются любыми политико-географическими единицами, для которых можно определить значение параметра за определённый год. Список регионов и их иерархия хранится в БД. По умолчанию БД содержит все субъекты федерации и федеральные округа РФ (по состоянию с 2010 года), а также саму РФ. Список регионов, выбранных из множества всех регионов для выборки данных по ним называется региональным срезом.

2. **Годы.** Ось годов всегда включает в себя фиксированный набор всех годов, содержащийся в БД. В настоящее время он включает в себя все годы с 1990 по 2013 включительно.

Итак, для формирования выборки необходимо задать её оси, а для этого нужно задать (или выбрать из сохранённых) алгоритм расчёта и региональный срез (ось годов задаётся автоматически). Эти действия могут быть выполнены в любом порядке. После задания осей происходит заполнение куба данных значениями, выбранными из БД и вычисленными на их основе.

Следующим этапом является визуализация. Доступны три формы визуализации – карта, таблица, диаграмма. Выбранные данные представляют собой куб, а для визуализации необходимо выбрать из этого куба некоторый набор значений, который можно наглядно визуализировать в виде таблицы,

карты или диаграммы. Таким набором является сечение куба какой-либо плоскостью. Для определения этой плоскости нам необходимо задать, какую ось (параметров, регионов или годов) и какое её значение мы фиксируем. Затем для двух оставшихся осей мы определяем, значения какой из них будут считаться столбцами, а какой – рядами, а также выбираем из всего выбранного нами набора значений для каждой оси список визуализируемых значений. Для визуализации в виде таблицы выбранным рядам и столбцам будут соответствовать строки и столбцы таблицы. Для визуализации в виде диаграммы каждый ряд будет представлять собой отдельный ряд (линию) диаграммы, а столбцы – значения оси абсцисс (оси Ох, области определения). Для визуализации в виде карты в качестве рядов должна быть определена ось регионов. Для одного значения из столбцов в один момент времени может быть построена карта, где каждый регион будет раскрашен в цвет, соответствующий значению параметра. Можно выбрать это конкретное значение столбца из списка всех значений в раскрывающемся списке над картой. Выбор того, какая ось является фиксированной, какая – осью рядов и какая – осью столбцов, фиксированное значение для фиксированной оси, а также набор визуализируемых значений для осей рядов и столбцов называется разрезом данных для визуализации.

Кроме определения разреза данных для визуализации можно также определить чисто графические настройки визуализации. В настоящее время они доступны лишь для картографической формы визуализации. Во-первых, это набор цветов, в которые раскрашиваются регионы в зависимости от соответствующих им значений. Во-вторых, это диапазоны значений, соответствующие каждому цвету (по умолчанию диапазоны формируются разбиением отрезка значений от минимума до максимума на количество отрезков, равное количеству цветов, однако опционально можно изменить это разбиение на диапазоны пользовательским). В-третьих, можно выбрать карту из списка доступных.

Результаты визуализации могут быть экспортированы в файлы различного формата в зависимости от формы визуализации. Таблица может быть сохранена в файл Excel (.xls). Карта – в виде растрового изображения формата jpg или векторного в формате svg. Диаграммы – в виде растрового изображения формата png.

Для работы с БД (добавление параметров в БД, определение и изменение их значений для различных регионов в различные годы) существует отдельный модуль. При добавлении параметра задаётся его название и краткое обозначение, являющееся его обозначением по умолчанию в алгоритме расчёта. Для задания/изменения значений параметров возможно как ручное редактирование, так и импорт таблицы значений из файла Excel.

Совокупность алгоритма расчёта, регионального среза, разреза данных и настроек визуализации называется проектом. Для каждого проекта возможно задание свойств: краткого названия (отображающееся в панели заголовка) и подробного описания. Проект сохраняется в файлах собственного формата с расширением «asy» (от “asymmetry”).

Описание внутреннего устройства программы и использованных технологий.

Для реализации системы был выбран язык Java. Во-первых, это поможет в перспективе объединить систему визуализации и анализа с системой моделирования, во-вторых, существуют различные свободно распространяемые Java-библиотеки, предоставляющие нужные нам средства визуализации.

Для хранения исходных статистических данных используется база данных, в настоящее время это файл Microsoft Access, но возможна перенастройка на произвольную БД с нужной нам структурой данных. Структура данных в общем виде представлена на Рисунке 3.2.

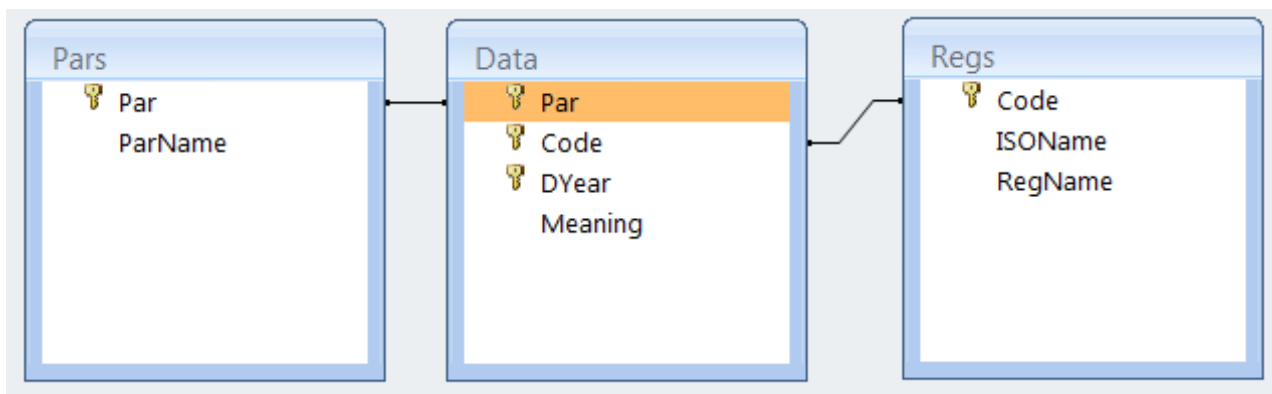


Рисунок 3.2 – Основные таблицы БД

Для сохранения файлов проекта используется технология Java Serialize, позволяющая легко сохранять и восстанавливать объекты сложной структуры. Для настроек файлов программы используется файл формата XML (settings.xml). С одной стороны, это облегчает его программную поддержку, поскольку стандартная библиотека Java предоставляет инструменты для работы с параметрами приложения и их сохранение/чтение из файла формата XML. С другой стороны, файл настроек, таким образом, имеет понятную пользователю структуру и может быть легко исправлен вручную. Некоторая его громоздкость не составляет проблемы, поскольку собственно настройки приложения задаются малым количеством параметров, основная тяжесть ложится на отдельные проекты по анализу и визуализации данных. Для хранения алгоритмов расчёта используются файлы формата CSV (с расширением “.alg”). Для картографической визуализации используются файлы формата SVG особого формата, позволяющего легко и быстро произвести раскрашивание определённых областей и дающего ряд других преимуществ, о которых мы расскажем ниже.

Все используемые программой ресурсы находятся в папке ares, расположенной в той же папке, что и сама программа. Файл настроек и файл базы данных лежат непосредственно в папке. Алгоритмы хранятся в папке algs (впрочем, это лишь папка по умолчанию, при сохранении и чтении алгоритма можно выбрать другую папку). Файлы проектов и файлы с результатами анализа и визуализации лежат в папках projects и results

соответственно (это также лишь папки по умолчанию). Файлы карт находятся в папке `maps`, там же лежит настроечный файл `maps.txt` формата CSV, содержащий перечень карт и дополнительную информацию о них.

Использование файловой системы для хранения многих настроек даёт нашей программе желаемую гибкость. Многие задачи по экстенсивному расширению её функциональности могут быть решены простым копированием, созданием, редактированием файлов.

Для создания интерфейса программы, визуализации данных в табличном формате использованы средства стандартной библиотеки Java Swing. Для создания визуализации в форме графика использовалась свободно распространяемая библиотека JFreeCharts.

Система картографической визуализации была разработана самостоятельно. Она опирается на формат векторной графики SVG, позволяющий описывать рисунок как совокупность фигур на плоскости (а не таблицу цветов отдельных точек, как используется в обычных растровых форматах изображения, таких как BMP, JPG, GIF). Кроме этого, он предоставляет возможность задания объектам или их группам различных свойств (таких как цвет, контур и т.п.). Формат SVG является подмножеством формата XML, что позволяет легко работать с ним, изменять его даже в обычном текстовом редакторе. Это всё делает данный формат незаменимым для решаемой задачи.

С помощью идентификатора мы можем сопоставить каждой политико-географической единице область на карте (или их группу, если эта единица содержит анклав, острова и т.п.). В качестве идентификатора используется код, определённый согласно стандарту ISO 3166 и однозначно идентифицирующий страны и их основные административно-территориальные единицы.

С помощью задания свойств области (или группы областей) мы можем раскрасить её в определённый цвет, или применить какой-либо другой приём визуализации данных.

Кроме основной версии программы, созданы также её демо-версии, реализующие часть функциональности и представляющие собой веб-приложения. Их отличительной особенностью является использование технологии HTML5, тогда как многие подобные системы используют для визуализации Flash. Однако Flash в настоящее время практически не поддерживается современными вычислительными устройствами – смартфонами, планшетными компьютерами. HTML5, наоборот, становится всё более поддерживаемым форматом. Итак, опишем созданные нами версии программ.

Во-первых, это система визуализации данных, подготовленных основной программой, но предназначенной для публикации результатов моделирования в интернете. Для картографической визуализации используются те же технологии, поскольку формат векторной графики SVG в настоящее время поддерживается всеми текущими версиями основных интернет-браузеров. Для визуализации в виде диаграммы используется библиотека HighCharts, также построенная на базе формата SVG, кроме этого, использующая формат VML, который позволяет просматривать графики даже в старых версиях браузера Internet Explorer. Приложение расположено на интернет-странице проекта [47].

Во-вторых, это система построения картографической диаграммы. Она построена исключительно на формате SVG, поддерживает загрузку данных пользователя в формате CSV, распознавание названий регионов, настройку диапазонов значений и соответствующих им цветов, сохранение картографической диаграммы в файл формата SVG на компьютере пользователя. Приложение расположено на интернет-странице проекта [77].

3.5 Моделирование развития Северного морского пути

В качестве примера использования созданного программного комплекса рассмотрим моделирование различных сценариев развития транспортной инфраструктуры Северного морского пути (СМП) и его использования для международных грузоперевозок, а также в качестве трассы для вывоза углеводородов с отдалённых месторождений.

Тема развития Арктики привлекает всё большее внимание со стороны правительства в последние годы и даже месяцы. Важным индикатором этого является Стратегия развития Арктической зоны Российской Федерации, утвержденная Президентом России 20 февраля 2013 [38], определяющая основные способы и средства достижения устойчивого развития Арктических районов России. Важное место при этом занимает «модернизация и развития инфраструктуры арктической транспортной системы, обеспечивающей сохранение Северного морского пути как единой национальной транспортной магистрали Российской Федерации», которой посвящён полностью пункт 14.

Разумеется, всё это требует масштабных капиталовложений и согласованной работы различных отраслей, обоснованного выбора инвестиционных проектов. В связи с этим становится необходимым «научное обоснование долгосрочных перспектив и основных направлений развития различных видов деятельности в Арктике» (пункт 14д [38]). В частности, интересен вопрос о том, насколько окажутся задействованными (и при каких условиях) вновь возводимые объекты инфраструктуры. Исследованию этого вопроса и посвящена данная глава.

В последние годы полным ходом идёт восстановление и развитие СМП, растёт количество перевозимых по нему грузов, в том числе транзитных. Хотя текущий объём перевозок в разы меньше достигнутого в 1987 году максимума (6,6 млн.т), ряд фактов говорит о перспективности данного маршрута [39]. Рассмотрим эти факты подробнее.

Первым таким фактом можно назвать развитие арктических областей Западной Сибири, связанное с освоением новых месторождений нефти и газа, расположенных вблизи с трассой СМП. Северные области Западной Сибири, в первую очередь Ямал, переживают в настоящее время период крупной, небывалой для этих мест экономической активности. В тяжёлых условиях, буквально на ровном месте строятся новые порты (Сабетта, Каменный Мыс), железные дороги (Паюта-Бованенково-Карская-Харасавэй с ответвлениями к строящимся портам), заводы (по производству сжиженного природного газа).

Хотя в своей истории Ямал пережил яркий период транспортного бума во время плаваний русских купцов и промышленников в Мангазею в конце XVI – начале XVII века [21], до XXI века он оставался всё ещё очень слабо освоенным районом. Планы по строительству на берегу Ямала (близ устья Оби) порта СМП, который стал бы крупным логистическим центром, связывавшим морской, речной и железнодорожный транспорт, начали активно разрабатываться ещё в конце 40-х годов XX века. Во время войны была достроена и введена в эксплуатацию железная дорога на Воркуту (к месторождению каменного угля), затем её ответвление Чум-Лабытнанги к низовьям Оби и планирующемуся к постройке новому порту.

Однако в ходе подготовительных работ в 1947-1949 годах выяснилось, что Обская губа очень неудобна для судоходства: чересчур мелководна для крупных морских судов [33]. В связи с этим строительство железной дороги продолжилось к низовьям Енисея, доступным для прохода крупных океанских кораблей [25] и Норильскому промышленному району. Хотя после 1953 года строительство дороги было заброшено, она послужила началом освоения северных областей ЯНАО, именно вдоль её трассы в 1960х годах были открыты крупнейшие в России и одни из крупнейших в мире месторождения природного газа (Уренгойское, Заполярное, Медвежье).

По мере приближения осваиваемых месторождений к Арктике становилась всё более явной возможность использования инфраструктуры СПМ, что было продемонстрировано при реализации проекта «Варандей». В настоящее время полным ходом идёт освоение ямальских месторождений нефти и газа.

Всё это снова сделало актуальным развитие транспортной инфраструктуры Ямала. Во 2й половине 2000х достроена железная дорога Обская-Бованенково, ставшая самой северной железной дорогой в мире. Ведутся работы по её продолжению с выходом к Карскому морю и Обской губе, строительству портов. Рядом с будущим портом в Обской губе Сабетта (прогнозируемый грузооборот к 2020 г. – 13-23 млн. тонн в год [39]) планируется построить завод по сжижению газа, а в порту обслуживать в основном суда-газовозы, вывозящие добытый газ.

Проблема мелководной губы решается теперь радикально: углубляется дно, в порту строится подъездной канал [34]. Примечательно, что в 2012 году также были проведены работы по разминированию Обской губы – мины были поставлены в 1943 год проникшей сюда немецкой подводной лодкой [40]. Уже этот факт сам по себе иллюстрирует стратегическую важность рассматриваемого района.

Второй предпосылкой развития СПМ в последние годы является связанная с таянием арктических льдов возросшая привлекательность его использования для транзитных грузоперевозок между Европой и Азией. Действительно, маршрут с использованием СПМ при транспортировке грузов из стран Восточной Азии в страны Западной и особенно Северной Европы оказывается несколько короче, чем традиционный маршрут через Суэцкий канал и значительно короче, чем маршрут вокруг Африки (его вынуждены преодолевать корабли, для которых прохождение канала невозможно или невыгодно).

Однако использование транзитного потенциала СМП сдерживается суровыми природными условиями, коротким периодом навигации и неразвитостью инфраструктуры.

На Рисунке 3.3 представлен фрагмент карты-приложения к Транспортной стратегии Российской Федерации на период до 2030 года, разработанной Министерством транспорта РФ и утверждённой распоряжением Правительства Российской Федерации от 22 ноября 2008 года № 1734-р [44].



Рисунок 3.3 – Фрагмент приложения к Транспортной стратегии РФ до 2030 г.

Зелёным цветом выделены строящиеся железные дороги. Как видно, планируемое восстановление участка заброшенной дороги Салехард-Надым

связет вновь создаваемые морские порты с центральными районами Тюменской области. Всё это создаёт предпосылки для заполнения грузопотока портов Ямала (а, следовательно, и СМП) не только нефтегазовыми, но и различными, в том числе импортными грузами.

Однако прогнозы, обещающие таяние арктических льдов, которое в последние годы стало, кажется, уже реальностью, заставляют страны, заинтересованные в перевозках между Азией и Европой, проявлять всё больший интерес к СМП. Особенно значимым является интерес со стороны Китая. Для сравнения можно отметить, что суммарный грузооборот всех морских портов России в 2011 году составил 535 млн. тонн (при этом свыше 55% его приходится на сырую нефть и продукты её переработки), а одного Шанхайского порта – 590 млн. тонн [45].

При этом Россия относится к категории стран с довольно крупным (от 500 млн. до 1 млрд. тонн) суммарным грузооборотом морских портов, к которой также относятся, например, Япония, Австралия и Великобритания.[39]. Поэтому переориентация хотя бы нескольких процентов китайского грузооборота на маршрут СМП окажет значительное влияние на его развитие.

В перспективе использование транзитными судами нашей арктической инфраструктуры (в первую очередь, атомных ледоколов для проводки караванов судов по опасным участкам) сулит немалые выгоды как северу России, так и стране в целом.

Моделирование различных вариантов развития. Указанные выше предпосылки ещё не означают автоматически, что СМП будет развиваться, что прогнозы об увеличении грузопотока по нему до 30-80 млн. т в год к 2030 году станут реальностью. Для исследования различных вариантов развития транспортной сети рассматриваемого района нами была составлена модель.

Итак, главным объектом исследования для нас являлись перспективы грузоперевозок по СМП. Им обслуживаются два основных направления.

Во-первых, это связь российских северных промышленных районов с «большой землёй» и европейскими потребителями их продукции (топливо-энергетического сырья).

Во-вторых, транзитные перевозки груза между странами Европы и Восточной Азией, в основном, Китаем. При этом основными конкурентами СМП являются пути через Суэцкий канал и вокруг Африки. Интересно было бы посмотреть, сможет ли СМП составить им достойную конкуренцию, и если да, то при каких условиях.

Исходя из этого были заданы начальные условия модели, выбраны основные участники (Россия, Китай, ЕС) и маршруты грузоперевозок. На Рисунке 3.4 представлено начальное состояние модели.

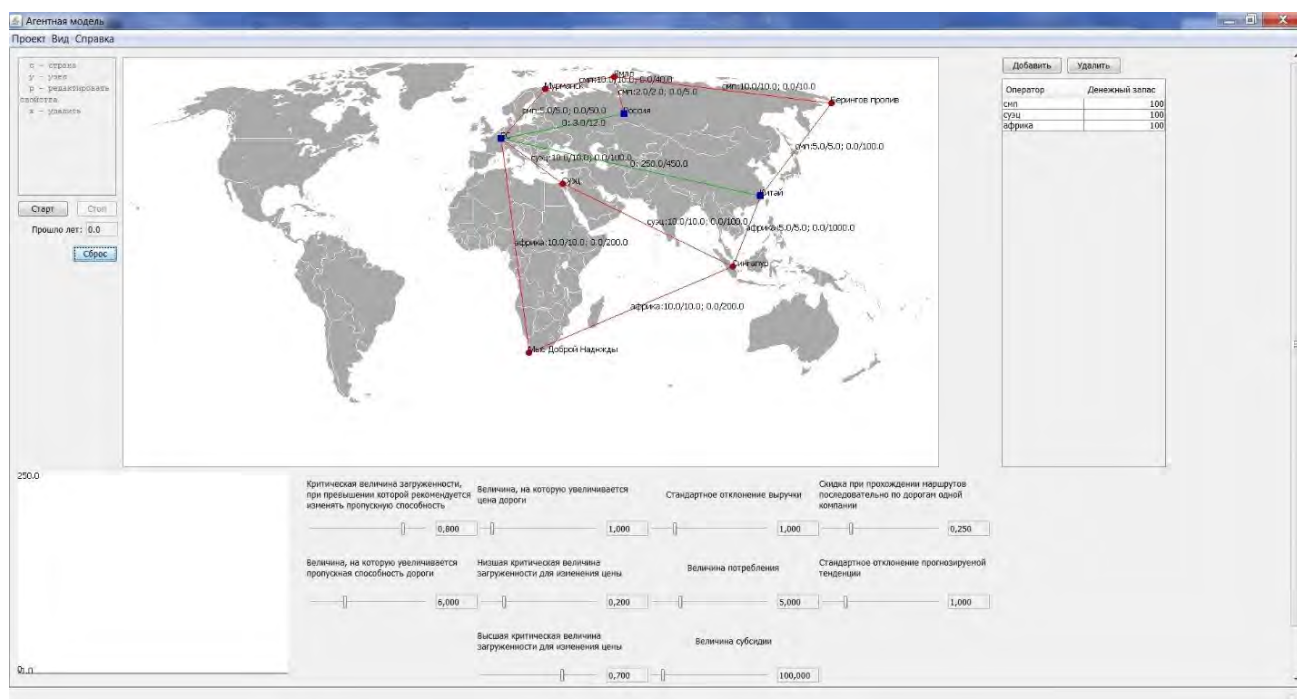


Рисунок 3.4 – Начальное состояние модели

Выбирая различные сценарии, мы исходили из следующего. Для СМП в ещё большей степени, чем для других магистралей, справедлив закон – чем **большие** вложения совершаются в транспортную инфраструктуру, тем меньше будет в итоге удельная стоимость перевозки единицы груза,

Таким образом, мы приняли два варианта развития. В первом удельная стоимость перевозок по СМП остаётся высокой, во втором она снижается благодаря комплексному развитию инфраструктуры.

Map showing shipping routes from the port of Murmansk (Мурманск) to various destinations:

- Мурманск** (Murmansk)
- Берингов пролив** (Bering Strait)
- Читай** (Chitai)
- Сингапур** (Singapore)
- Малайя** (Malaya)
- Судан** (Sudan)
- Египет** (Egypt)
- Италия** (Italy)
- Франция** (France)
- Германия** (Germany)
- Нидерланды** (Netherlands)
- Бельгия** (Belgium)
- Лихтенштейн** (Liechtenstein)
- Швейцария** (Switzerland)
- Австрия** (Austria)
- Чехия** (Czech Republic)
- Польша** (Poland)
- Венгрия** (Hungary)
- Словакия** (Slovakia)
- Украина** (Ukraine)
- Беларусь** (Belarus)
- Литва** (Lithuania)
- Латвия** (Latvia)
- Эстония** (Estonia)
- Финляндия** (Finland)
- Швеция** (Sweden)
- Норвегия** (Norway)
- Дания** (Denmark)
- Германия** (Germany)
- Франция** (France)
- Италия** (Italy)
- Греция** (Greece)
- Турция** (Turkey)
- Сирия** (Syria)
- Ливан** (Lebanon)
- Израиль** (Israel)
- Египет** (Egypt)
- Судан** (Sudan)
- Сомали** (Somalia)
- Кения** (Kenya)
- Уганда** (Uganda)
- Руанда** (Rwanda)
- Бурунди** (Burundi)
- Конго** (Congo)
- Замбия** (Zambia)
- Малави** (Malawi)
- Мозамбик** (Mozambique)
- Индонезия** (Indonesia)
- Малайзия** (Malaysia)
- Сингапур** (Singapore)
- Австралия** (Australia)
- Новая Зеландия** (New Zealand)

Shipping routes and their coordinates:

- Мурманск - Берингов пролив** (смп: 22.0/10.0; 12.552681668651491/160.0)
- Мурманск - Чинтай** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Сингапур** (смп: 8.0/5.0; 0.0/172.0)
- Мурманск - Малайя** (смп: 16.0/5.0; 12.552681668651491/170.0)
- Мурманск - Египет** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Италия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Франция** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Германия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Нидерланды** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Бельгия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Лихтенштейн** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Швейцария** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Австрия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Чехия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Польша** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Венгрия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Словакия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Украина** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Беларусь** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Литва** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Латвия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Эстония** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Финляндия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Швеция** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Норвегия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Дания** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Германия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Франция** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Италия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Греция** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Турция** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Сирия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Ливан** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Израиль** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Египет** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Судан** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Сомали** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Кения** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Уганда** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Руанда** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Бурунди** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Конго** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Замбия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Малави** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Мозамбик** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Индонезия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Малайзия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Сингапур** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Австралия** (смп: 1.0/2.0; 12.552681668651491/170.0)
- Мурманск - Новая Зеландия** (смп: 1.0/2.0; 12.552681668651491/170.0)

На Рисунке 3.6 приведён результат моделирования по II варианту.

более точным количественным оценкам необходимых для этого условий, и анализу окупаемости необходимых для их достижения капиталовложений.

3.6 Моделирование развития Международных Транспортных коридоров Евразии

Северный Морской путь представляет собой лишь часть (хотя и довольно перспективную) транзитного потенциала России. Важное значение России как возможного пути международных перевозок обусловлено её центральным положением на евразийском континенте. Грузопоток между странами Европы и Азии в последнее время всё увеличиваются.

Экономики крупнейших стран Азии (Китая, Индии) переживают в последнее время период бурного роста. В то же время они нуждаются, во-первых, в энергоносителях (поскольку не обладают достаточным количеством запасов топливно-энергетических полезных ископаемых), а во-вторых, в рынках сбыта (в качестве которых выступают, как правило, развитые страны Европы). Всё это обуславливает наличие мощного грузопотока между востоком, юго-востоком и западом, северо-западом Евразии.

Большинство сухопутных и некоторые морские транспортные пути, обслуживающие этот грузопоток, проходят по территории России. Можно отметить два ключевых направления, входящих в состав Международных Транспортных Коридоров «Запад-Восток» (из Европы через транссибирскую магистраль в страны восточной Азии, в первую очередь, Китай) и «Север-Юг» (из Северной Европы через Центральную Россию и Каспийское море или вокруг него в страны Южной Азии, в первую очередь, Индию).

У этих путей имеются как альтернативные маршруты, использующие их инфраструктуру лишь отчасти, так и прямые конкуренты, по тем или иным причинам обходящие территорию России. В развитие как одних, так и других путей вкладывается немало средств, происходит развитие

инфраструктуры, повышение пропускной способности. Немаловажную роль играет и ценовая политика, установление тех или иных пошлин и сборов. При этом возникает проблема оценки перспектив того или иного направления в зависимости от международной конъюнктуры в условиях конкуренции.

Традиционно большая часть всего грузопотока между Европой и Азией проходит по морским путям. Морской транспорт, несмотря на относительный проигрыш в скорости, имеет ряд неоспоримых преимуществ. Главной артерией между Европой и Азией без сомнения можно назвать Суэцкий канал. Так, проходящий через него в 2012 году грузопоток составил 740 млн. тонн [66].

Однако в последнее время ограничения его пропускной способности вынуждают компании искать альтернативные пути для транспортировки грузов. Немаловажное значение в этом плане имеет политическая нестабильность в районе Красного моря (в частности, Сомали) и Суэцкого канала (Египта).

Всё это делает актуальным исследование и развитие альтернативных транспортных путей между Европой и Азией, в частности, проходящих по территории России. Ключевыми центрами формирования грузопотока в данном случае становятся Европейский Союз, Китай и Индия. Также особое значение приобретает грузопоток, связанный непосредственно с Россией. Среди ключевых путей, кроме рассмотренных в предыдущем разделе морских, выделим и рассмотрим некоторые важнейшие сухопутные и мультимодальные маршруты, связывающие Китай и Индию с европейским рынком.

Приблизительно их можно оценить по карте азиатских железных дорог, изображённой на Рисунке 3.7 (разным цветом отмечены железные дороги с разной шириной колеи)



Рисунок 3.7 – Карта железных дорог Азии

Географическое положение Индии существенно снижает возможности использования сухопутных маршрутов для международной торговли. С севера и северо-востока территорию Индии ограничивают высочайшие горные системы Земли – Гималаи и Каракорум. С северо-запада почти на 3000 км тянется граница с Пакистаном, однако её пересечение затруднено в связи с длительным индо-пакистанским конфликтом. Тем не менее в 2005 году был открыт пункт для пересечения границы в селе Вагах, товарооборот через него активно растёт. В перспективе возможно соединение Индии через Пакистан с Европейским рынком. В частности, для этого служит создаваемый МТК «Север-Юг». Он проходит по территории центральной России и выходит к району Каспийского моря. Здесь возможны различные маршруты: по восточному берегу через Казахстан и Туркмению, по западному через Азербайджан, или напрямую по Каспийскому морю в Иран.

По Ирану грузы могут доставляться к портам в Индийском океане, либо, в перспективе, через Пакистан напрямую в Индию. По состоянию на 2013 год входит в завершающую стадию строительство огибающих Каспий железных дорог, связывающих российскую железнодорожную сеть с иранской через Азербайджан (Казвин – Решт – Астара, без захода в Армению и Нагорный Карабах) и Казахстан и Туркмению (Узень – Берекет – Горган, без захода в Узбекистан). Всё это создаёт предпосылки для создания нового маршрута грузоперевозок из Южной Азии в Европу.

Географическое положение Китая также (как и у Индии) накладывает существенные ограничения на грузоперевозки через сухопутные границы. С юга территорию Китая ограничивают Гималаи и Каракорум, с запада – Памир и Тянь-Шань. Перевозка товаров через эти районы или невозможна, или сопряжена с большими трудностями.

К примеру, Каракорумское шоссе, построенное в исключительно тяжёлых условиях и соединяющее Китай с Пакистаном, находится в зоне частых обвалов, камнепадов, периодически затапливается, а на зимнее время вовсе закрывается. Кроме этого, стоит отметить, что оно проходит по территории, на которую претендует Индия, хотя фактически управляет ею Пакистан.

У железнодорожной сети Китая есть считанное число выходов, через которые возможна транспортировка грузов в Европу. Во-первых, это выходы в Россию (непосредственно или через Монголию). Во-вторых, это выходы в Казахстан. В последнее время Китай уделяет большое внимание строительству путей сообщения, связывающих его со Средней Азией. В 2012 году был открыт второй железнодорожный переход на китайско-казахстанской границе, планируется строительство железных дорог, связывающих Китай с Киргизией.

Налаживание сообщения Китая со странами Средней Азии позволяет ему использовать пути, связывающие его с Европой как по территории

России, так и минуя её, через Иран и Турцию. Оценить в такой сложной обстановке, какие маршруты и при каких условиях окажутся наиболее востребованными – довольно сложная проблема. Для её исследования была построена агентная модель. Исследования показали, что в краткосрочном прогнозе наиболее вероятная картина распределения имеет вид, представленный на Рисунке 3.8.

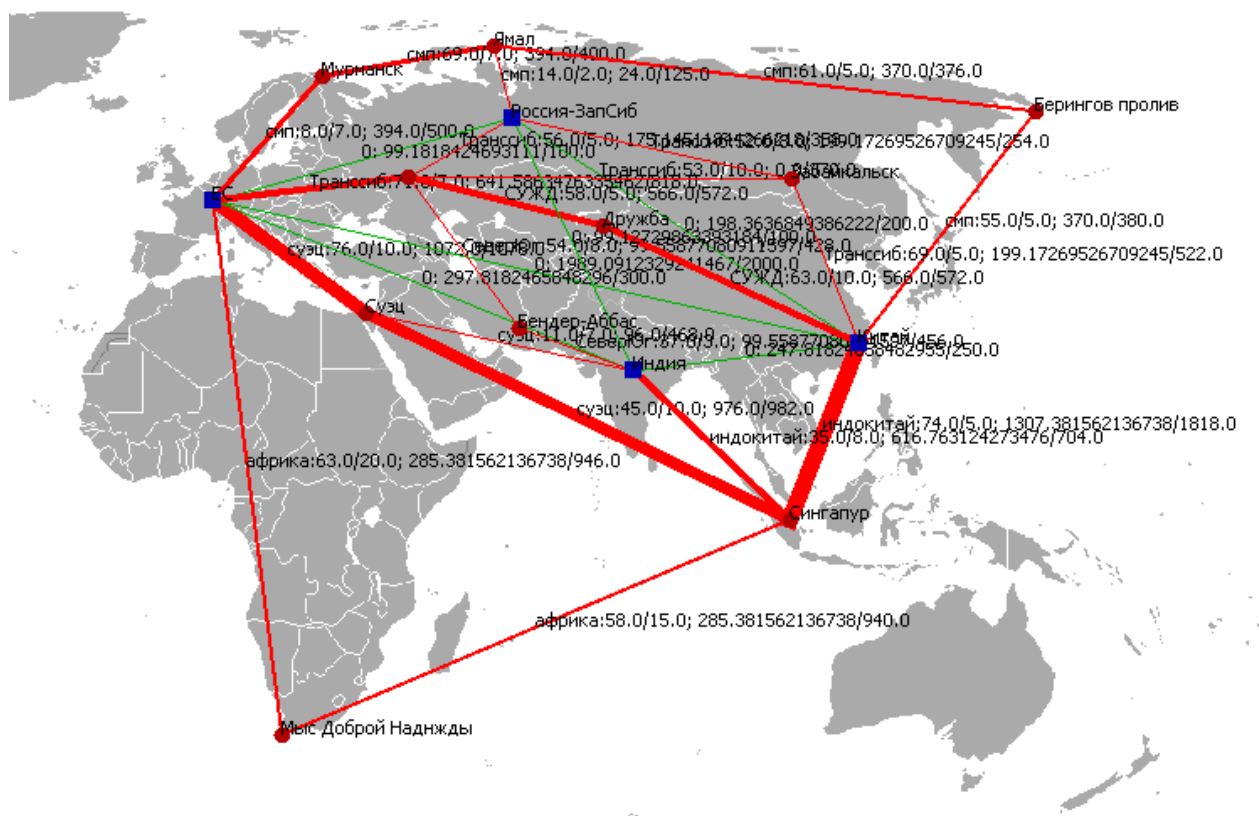


Рисунок 3.8 – Результат моделирования по I варианту

Иными словами, морской путь через Суэцкий канал остаётся основным, на маршрут СМП переходит около 10-20% китайского грузооборота, наиболее задействованным сухопутным маршрутом оказывается путь, проходящий через Казахстан.

В случае, если стоимость перевозок по Транссибу снижается, а пропускная способность увеличивается, в дальней перспективе становится вероятной картина, изображена на Рисунке 3.9.

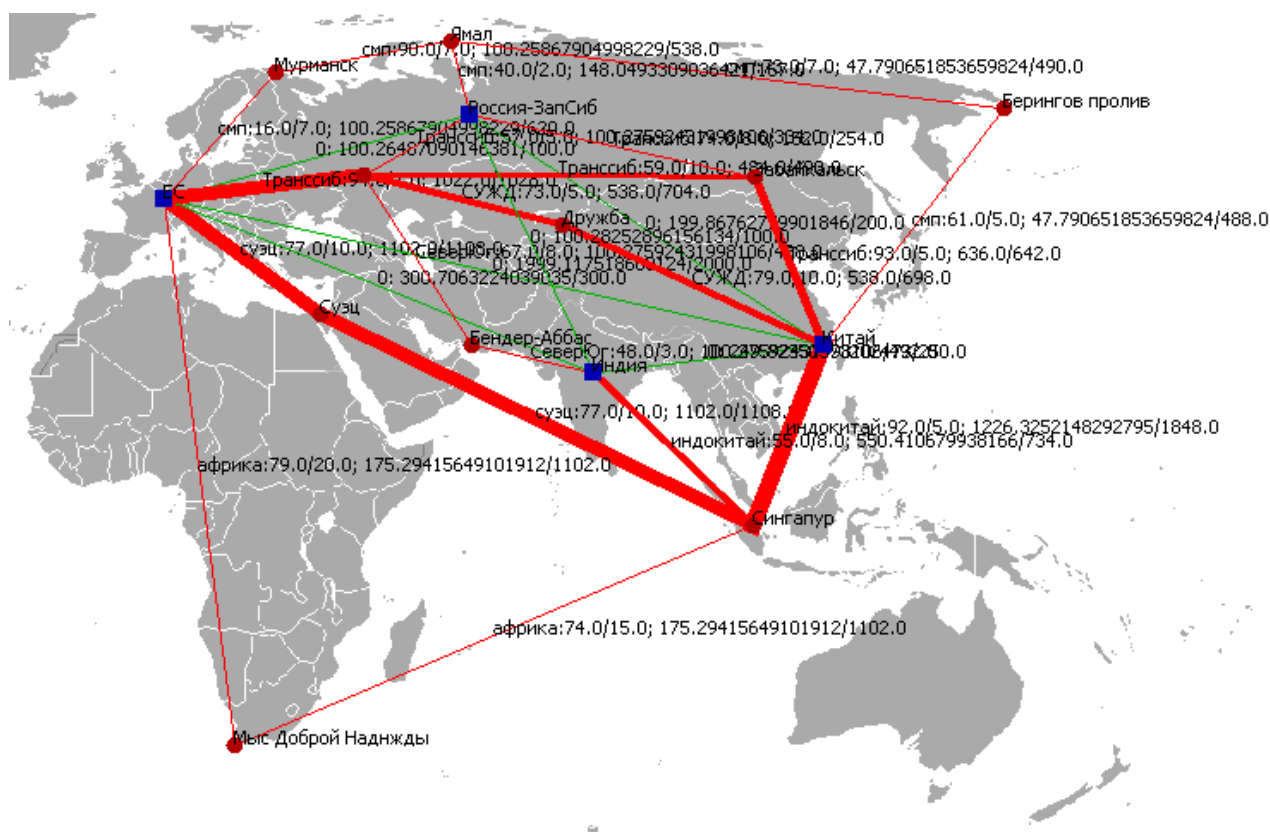


Рисунок 3.9 – Результат моделирования по II варианту

То есть, при таком развитии событий Транссиб становится таким же важным маршрутом, но, в свою очередь, начинает конкурировать с СМП. Между тем, значительно увеличивается роль центральной России, в которой встречаются как перевозки из Китая (через Казахстан и по транссибирской магистрали) и из Индии по МТК «Север-Юг».

Итак, мы показали, как построенный программный комплекс позволяет переходить от простых к более сложным моделям, исследовать их, получать прогнозы при различных заданных первоначальных условиях. Наличие прогнозов грузоперевозок по различным МТК, проходящим по территории России, позволяет сопоставить их с результатов моделирования. В Таблице 3.2 приведено такое сопоставление.

Таблица 3.2 – Транзитный грузопоток, млн. тонн

	Текущий	Прогноз на 2020 г. [39]		Результат моделирования	
		Реалистичный	Оптимистичный	Вариант 1	Вариант 2
МТК «Север-Юг»		25	40	26,9	41,3
Северный морской путь	3	30	50	47,8	25,4
Транссиб	33	60	90	57,4	103

Заключение

Итак, подытожим проделанную работу и выделим достигнутые результаты.

Проведён анализ различных МАС, на основании которого определены принципы из разделения на группы и семейства.

В качестве основного исследуемого примера построена экономическая мультиагентная модель развития транспортной сети в условиях изменяющихся грузопотоков и конкуренции между операторами перевозок.

На основании конкретной модели выделены общие принципы построения и работы ряда схожих МАС и определён круг интересующих нас вопросов, касающихся их работы. Это семейство МАС, состоящих из одинаковых, однородных агентов, поведение которых реактивно и определяется простыми стохастическими правилами.

Показано, что в рассматриваемое семейство входят важные модели, применяемые в естественных науках, и эвристические оптимизационные алгоритмы, обоснована эффективность такого их представления при программной реализации.

Для полученного семейства МАС проведена формализация процесса работы и интересующих нас свойств, определено понятие характеристической функции, проведена аналогия с Марковскими процессами.

На основании этой аналогии доказана взаимосвязь характеристической функции со стационарным распределением Цепи Маркова, проведена оценка распределения значений характеристической функции.

На ряде примеров конкретных моделей и алгоритмов показана возможность их представления в используемом нами виде, и применения к ним доказанных утверждений, что помогает существенно упростить процесс

исследования систем, выделить в них основные характеристики, определяющие их поведение.

Так, для реализации модели Шеллинга фактически построено собственное доказательство утверждений о её поведении, ориентированное на применение полученных выше общих утверждений. При этом становятся ясными многие закономерности, определяющие те или иные аспекты её поведения.

Реализована программная библиотека, позволяющая легко создавать МАС, описанные в терминах рассматриваемого семейства, и содержащая средства контроля и исследования процесса их работы.

Применение находят как практические, так и теоретические результаты работы. Полученные программные модели используются для прогнозирования развития международных перевозок в ИЭиОПП СО РАН. На базе программной библиотеки создаются и исследуются различные агентные модели, исследуются различные сценарии их работы. Выведенные в работе теоретические утверждения используются при доказательстве свойств различных систем.

Направления дальнейшего развития тоже можно условно развить на два направления – практическое и теоретическое. Возможно дальнейшее конструирование и исследование моделей и различных алгоритмов, доказательство их свойств. Также на базе работы становится возможным дальнейшее изучение выделенного семейства МАС, проведение более удобных для тех или иных случаев оценок.

Основные результаты работы

1. Предложено семейство многоагентных систем с однородными агентами, обладающими стохастическим реактивным поведением, в которое входят важные модели, применяемые в естественных науках; эвристические оптимизационные алгоритмы такие, как: генетические алгоритмы и

алгоритмы «роя частиц»; модели типа «хищник-жертва»; модели распространения вирусов и др. Показана связь систем данного семейства с Марковскими процессами, на основании которой доказаны утверждения, характеризующие поведение таких систем.

2. Исследован ряд конкретных моделей и алгоритмов: модель сегрегации Шеллинга, игра с логлинейным правилом принятия решения, алгоритм имитации отжига. Показана возможность их формального представления в виде однородной цепи Маркова, построены альтернативные доказательства вероятностных оценок их поведения, что позволило объяснить ряд закономерностей, определяющих различные аспекты поведения такого рода моделей.

3. Реализован программный комплекс, который предоставляет возможность легко создавать программные модели многоагентных систем, описанных в терминах рассматриваемого в диссертации семейства, содержит средства управления системами и средства исследования процессов их функционирования. Проведено сравнение с рядом известных многоагентных программных платформ.

4. Построена многоагентная модель развития транспортной сети в условиях изменяющихся грузопотоков и конкуренции между операторами перевозок. Реализован комплекс программ, предназначенный для моделирования возможных путей развития Мировой хозяйственной системы, анализа и визуализации исходных статистических данных и полученных результатов.

Литература

1. **Айвазян С.А., Мхитарян В.С.** Прикладная статистика и основы эконометрики. – М.: ЮНИТИ, 1998.
2. **Батищев С.В., Лахин О.И., Минаков И.А., Ржевский Г.А., Скобелев П.О.** Разработка мультиагентной системы для дистанционного обучения в Интернет-портале «Оптик-сити». // Известия Самар. научн. центра РАН. – 2003. – Т.5, №1. – С.91-95.
3. **Береснев А.Л., Бонко А.С., Зайцев И.Д.** Разработка ПО «Оценка асимметрии регионального развития» // Материалы XLVI международной научной студенческой конференции «Студент и научно-технический прогресс», Информационные технологии. Новосибирск, 2008, с. 133-134.
4. **Богатырев М. Ю.** Инварианты и симметрии в генетических алгоритмах // Искусственный интеллект. Одиннадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2008 (28 сентября -- 3 октября 2008 г., г. Дубна, Россия). Труды конференции. Т.1 Секция 3
5. **Булинский А.В. Ширяев А.Н.** Теория случайных процессов. – М.: ФИЗМАТЛИТ, 2005. – 408 с. – ISBN 5-9221-0335-0.
6. **Бурганова Л.А.** Теория управления: учебное пособие (с грифом УМО РФ) – М.: ИНФРА-М, 2005. – 139 с.
7. **Бурмистров М.Ю., Валиев М.К., Дехтярь М.И., Диковский А.Я.** О верификации динамических свойств систем взаимодействующих агентов. // Труды X национальной конференции по искусственному интеллекту с международным участием, Обнинск, Физматлит, Москва, 25-28 сентября 2006 года.
8. **Валиев М.К., Дехтярь М.И.** Вероятностные мультиагентные системы: семантика и верификация. // Вестник Тверского государственного университета, серия «Прикладная математика», 35 (95), 2008, С. 9-22.

9. **Валиев М.К., Дехтярь М.И.** О сложности верификации недетерминированных вероятностных мультиагентных систем // Журнал "Моделирование и анализ информационных систем", 2010, т. 17, № 4, с. 41-51.
10. **Валиев М.К., Дехтярь М.И.** О моделировании вероятностных МАС Марковскими системами // Труды 12-ой Национальной конференции по искусственному интеллекту с международным участием. М., Физматлит, 2010, с. 79-87.
11. **Есикова Т.Н., Зайцев И.Д.** Разработка агентной модели «Оценка стратегических направлений опорной транспортной сети России при разной геоэкономической архитектонике Мировой Хозяйственной Системы» // Управление развитием крупномасштабных систем (MLSD'2010): Труды Четвертой международной конференции (4-6 октября 2010 г., Москва, Россия). Москва, 2010 – Том I. С. 107-114
12. **Есикова Т.Н., Зайцев И.Д.** Разработка инструментария «Агентная модель прогнозирования изменений геоэкономических позиций России» // Управление развитием крупномасштабных систем (MLSD'2010): Материалы Четвертой международной конференции (4-6 октября 2010 г., Москва, Россия), Москва 2010 – Т. II. С. 59-61
13. **Зайцев И.Д.** Верификация мультиагентных систем с помощью цепей Маркова: оценка вероятности нахождения агентами оптимального решения // Программные продукты и системы, Тверь – том 4, 2013 г., стр. 89-93.
14. **Зайцев И.Д.** Вероятностная оценка результатов агентного моделирования на примере модели сегрегации Шеллинга. // Журнал "Проблемы управления", Москва – №2, 2014 г., стр. 39-43.
15. **Зайцев И.Д.** Опыт разработки прототипа агентной модели «оценка стратегических направлений опорной транспортной сети России» в среде NetLogo. // Управление развитием крупномасштабных систем

(MLSD'2010): Материалы Четвертой международной конференции (4-6 октября 2010 г., Москва, Россия), Москва 2010 – Т. II. С. 62-64

16. **Зайцев И.Д.** Разработка агентной модели «Оценка востребованности направлений опорной транспортной сети России при разных геоэкономических условиях»: построение каркаса модели с помощью технологии Java. Тезисы. // Управление развитием крупномасштабных систем MLSD'2011 : материалы Пятой междунар. конф. (3-5 окт. 2011 г., Москва, Россия) / общ. ред. С.Н. Васильев, А.Д. Цвиркун ; Ин-т проблем управления им. В.А. Трапезникова РАН. - М., 2011. - Т. II (секции 4.2 - 8). - С. 49-50.

17. **Зайцев И.Д.** Построение модели развития международной транспортной сети с использованием агентного подхода // Россия 2030 глазами молодых ученых. Материалы II Всероссийской научной конференции (Москва, 2011 г.) Москва Научный эксперт 2012, часть 1 стр 301-307

18. **Зайцев И.Д.** Верификация мультиагентных систем с помощью цепей Маркова // Материалы LI международной научной студенческой конференции «Студент и научно-технический прогресс», Информационные технологии. Новосибирск, 2013, с. 143.

19. **Зайцев И.Д.** Разработка ПО для оценки требований к транспортной сети с помощью агентного моделирования // Материалы XLVII международной научной студенческой конференции «Студент и научно-технический прогресс», Информационные технологии. Новосибирск, 2009, с. 143.

20. **Замятина Е.Б.** Современные теории имитационного моделирования: Специальный курс. - Пермь: ПГУ, 2007. - 119 с.

21. **История открытия и освоения Северного морского пути:** В 4 томах. Том 1. Арктическое мореплавание с древнейших времён до середины

XIX века / Белов М. И. Под редакцией: Я. Я. Гаккеля, А. П. Окладникова, М. Б. Черненко. — М.: Морской транспорт, 1956. — 592 с.

22. **Кочетов Ю.А.** Вероятностные методы локального поиска для задач дискретной оптимизации // Дискретная математика и ее приложения. Сборник лекций молодежных и научных школ по дискретной математике и ее приложениям. М: МГУ, 2001, с. 87-117.

23. **Кочетов Ю.,** Младенович Н., Хансен П. Локальный поиск с чередующимися окрестностями // Дискретный анализ и исследование операций. Серия 2. 2003. Том 10, № 1. С.11–43.

24. **Красильников. О.Ю.** Региональная асимметрия структурных сдвигов в экономике Общество и экономика. 2001. № 2. С. 150 – 155.

25. **Курейчик В.М., Лебедев Б.К..** Определения и основные понятия генетических алгоритмов. // Интернет-лекция в рамках научной школы Третьей Всероссийской межвузовской конференции молодых ученых СПбГУ ИТМО
(http://faculty.ifmo.ru/csd/files/kureichik_v_m_lebedev_b_k_internet_lecture.pdf).

26. **Лаппо С. Д.** Морские суда в Самоедской протоке Енисея в 1928 году // «Летопись Севера», в III, Издательство Мысль, 1962, С. 87-90

27. **Лавровский Б.Л.** Экономический рост и региональная асимметрия (эмпирический анализ) // Новосибирск: Сибирской Научное Издательство, 2005 – 216 с.

28. **Малин А.С., Мухин В.И.** Исследование систем управления / Учебник для вузов. — М.: ГУ, 2002. — 329 с.

29. **Малинецкий Г. Г., Степанцов М. Е.** Дискретная математическая модель динамического развития транспортной сети // Журнал вычислительной математики и математической физики, 2009, том 49, номер 9, страницы 1565–1570

30. **Минасов Ш.М., Тархов С.В.** Проект "Гефест" как вариант практической реализации технологий электронного обучения в вузе в условиях

интеграции традиционного и дистанционного обучения // Образовательные технологии и общество (Educational Technology & Society). Журнал ВосточноЕвропейской подгруппы Международного форума. ISSN 1436-4522. 2005. № 8(1). С. 134-147.

31. **Новиков Д. А.** Иерархические модели военных действий / Управление большими системами. Выпуск 37. М.: ИПУ РАН, 2012. С.25-62.

32. **Павский В.А.** Лекции по теории вероятностей и элементам математической статистики: учебное пособие. – Кемеровский технологический институт пищевой промышленности. – Кемерово, 2004. – 184 с.

33. **Пиманов А.С.** История строительства железной дороги «Чум - Салехард – Игарка» (1947-1955 гг.) – Тюмень, 1998. – 114с.

34. **Пономарев Вадим** «Пошли на углубление» // «Expert Online» 15.06.2012, официальный сайт журнала «Эксперт» (<http://expert.ru/2012/06/15/poshli-na-uglublenie/>)

35. **Поспелов И.Г.** Экономические агенты и системы балансов: / Препринт – М.: ГУ-ВШЭ, 2001. – 68 с.

36. **Рассел С., Норвиг П.** Искусственный интеллект. Современный подход (2-е издание) / Издательский дом "Вильямс", 2006 - 1408 с.

37. **Рассказова, М. Н.** Имитационное моделирование систем : учебное пособие / М. Н. Рассказова. – Омск : Омский государственный институт сервиса, 2010. – 80 с.

38. **Стратегия** развития Арктической зоны Российской Федерации. // Официальный сайт Правительства Российской Федерации (<http://government.ru/docs/22846/>)

39. **Стратегия** развития морской портовой инфраструктуры России до 2030 года // Официальный сайт ФГУП «Росморпорт» (<http://www.rosmorport.ru/seastrategy.html>)

40. **Судоходные** районы Обской губы разминировали от военных сюрпризов // ГТРК "Ямал", "Вести" интернет-газета" ("VESTI.RU") (<http://www.vesti.ru/doc.html?id=917102>)

41. **Тарасов В.Б.** От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – М.: Эдиториал УРСС, 2002. – 352 с.

42. **Тарасов В.Б.** Агенты, многоагентные системы, виртуальные сообщества: стратегическое направление в информатике и искусственном интеллекте // Новости искусственного интеллекта. – 1998. – №2. – С.5-63.

43. **Теория управления:** Учебник / Под общ. ред. А.Л. Гапоненко, А.П. Панкрухина. - М.: Изд-во РАГС, 2003. - 558 с.

44. **Транспортная стратегия** Российской Федерации на период до 2030 года. // Официальный сайт Министерства транспорта Российской Федерации (http://www.mintrans.ru/documents/detail.php?ELEMENT_ID=13008)

45. **Унтура Г.А., Есикова Т.Н., Зайцев И.Д., Морошкина О.Н.** Проблемы и инструменты аналитики инновационного развития субъектов РФ. // Вестник НГУ: серия социально-экономические науки, Новосибирск – 2014 г., Т.14 №1, С.81-100.

46. **American Association of Port Authorities** World port ranking <http://www.aapa-ports.org/>

47. **Asymmetry Server Demo** (http://navizv.github.io/asym_web/)

48. **Cohen P.R and Levesque H.J.** Intention is Choice with Commitment// Artificial Intelligence. – 1990. – Vol.42. – P.213-262

49. **Dorigo M. et Gambardella L.M.,** Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem, IEEE Transactions on Evolutionary Computation, volume 1, numero 1, pages 53-66, 1997.

50. **Gerzson Kéri.** Tables for Bounds on Covering Codes
<http://www.sztaki.hu/~keri/codes/>
51. **Himoff, Jonathan, Rzevski, George and Skobelev, Petr** Magenta technology multi-agent logistics i-Scheduler for road transportation // AAMAS '06 Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, Pages 1514 - 1521, New York, NY, USA 2006
52. **Hjorth-Jensen, Morten** Computational Physics / University of Oslo, Fall 2008
53. **Holme, Petter and Grönlund, Andreas** (2005). 'Modelling the Dynamics of Youth Subcultures'. // Journal of Artificial Societies and Social Simulation 8(3)3
54. **Hu, X., Zhang, J., and Li, Y.** (2008) Orthogonal methods based ant colony search for solving continuous optimization problems. Journal of Computer Science and Technology, 23 (1). pp. 2-18. ISSN 1000-9000 (doi:10.1007/s11390-008-9111-5)
55. **International Monetary Fund** <http://www.imf.org/>
56. **JADE - Java Agent DEvelopment Framework**
(<http://jade.tilab.com/>).
57. **Marsella, Stacy, Pynadath, David V. and Read, Stephen J.** PsychSim: Agent-based modeling of social interactions and influence., in Proceedings of the International Conference on Cognitive Modeling, Pittsburg, 2004.
58. **MASwarm Agent Platform** (<http://navizv.github.io/MASwarm/>)
59. **NetLogo Homepage** (<http://ccl.northwestern.edu/netlogo/>).
60. **Okada, Daijiro, Tercieux, Olivier** Log-linear dynamics and local potential // Journal of Economic Theory – May 2012 – Volume 147, Issue 3 – P. 1140–1164

61. **Pancs Romans, Vriend Nicolaas J.** Schelling's Spatial Proximity Model of Segregation Revisited // Journal of Public Economics, 2007, Vol. 91, p. 1-24.
62. **Panzarasa Pietro, Carley Kathleen M., Jennings, Nicholas R.** A logical approach to formalizing negotiation in multi-agent systems // Center for Computational Analysis of Social and Organizational Systems (CASOS), Carnegie Mellon University (Pittsburgh, PA.) 2003
63. **Pham D.T., Ghanbarzadeh A., Koç E., Otri S., Rahim S., and M.Zaidi** The Bees Algorithm – A Novel Tool for Complex Optimisation Problems, Proceedings of IPROMS 2006 Conference, pp.454–461
64. **Raudenska, Lenka.** Swarm-based optimization Kvalita Inovacia Prosperita / Quality Innovation Prosperity XIII/1 – 2009 p.45-51
65. **Repast - Recursive Porous Agent Simulation Toolkit**
(<http://repast.sourceforge.net/>).
66. **Review of Maritime Transport 2012** // United Nations Conference on Trade and Development ([http://unctad.org/en/pages/publications/Review-of-Maritime-Transport-\(Series\).aspx](http://unctad.org/en/pages/publications/Review-of-Maritime-Transport-(Series).aspx))
67. **Schelling** **Model** **Demo**
(<http://navizv.github.io/MASwarm/schelling.html>)
68. **Sondahl, F., Tisue, S., & Wilensky, U.** Breeding faster turtles: Progress towards a NetLogo compiler. //. Proceedings of the Agent 2006 conference on social agents, Chicago, IL, 2006.
69. **Sycara K. P.** Multiagent systems // AI Magazine 10(2): 79-93, 1998.
70. **Tesfatsion, Leigh** Agent-Based Computational Economics: A Constructive Approach to Economic Theory / Handbook of Computational Economics, Vol. 2: Agent-Based Computational Economics, North-Holland/Elsevier, Amsterdam, the Netherlands, (2006)

71. **Vlassis, N.** A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence. / San Rafael, CA: Morgan & Claypool Publishers. (2008)
72. **Wilensky, U.** NetLogo Segregation model. // Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1997. URL: (<http://ccl.northwestern.edu/netlogo/models/Segregation>).
73. **Wooldridge M., Jennings N.** Intelligent Agents: Theory and Practice// The Knowledge Engineering Review. – 1995. – Vol.10, №2. – P.115-152
74. **World Bank, The** (<http://www.worldbank.org/>)
75. **Zhang, Junfu.** Tipping and Residential Segregation: A Unified Schelling Model // Journal of Regional Science, 2011,51(1), pp. 167-193.
76. **Zhang, Junfu.** "A Dynamic Model of Residential Segregation," Journal of Mathematical Sociology, 2004, 28, pp. 147-170.
77. **ZiMapMaker** – csv data to svg map (<http://navizv.github.io/zimm/>)

Приложение А. Доказательства утверждений о поведении некоторых МАС

В приложении исследован ряд примеров интересующего нас семейства МАС, применены доказанные нами выше утверждения об их свойствах. Показано, что с помощью их применения удаётся унифицировать и значительно упростить процесс доказательства утверждений о свойствах конкретных систем.

1 Модель Шеллинга

Рассмотрим в качестве конкретного примера агентной модели так называемую модель расовой сегрегации Шеллинга. Поскольку эта модель считается одной из первых агентных моделей, а Шеллинг – пионером в области агентного моделирования в общественных науках, то для неё существует большое количество конкретных реализаций и доказанных теоретически свойств.

Модель Шеллинга описывает поведение нескольких различных типов агентов при выборе места нахождения с учётом типов своих соседей. Наиболее частый пример (для американской науки, где эта модель и родилась) – стихийное возникновение расовой сегрегации при расселении людей по кварталам. Множество всех агентов разбито на подмножества агентов одного типа. Тип агента символизирует собой, как правило, цвет кожи, однако может отвечать и за другие признаки, например, религиозные воззрения или уровень дохода.

Место нахождения в классической модели соответствует месту проживания человека (или семьи), однако в вариациях модели может приобретать и другой смысл (например, выбор места в зрительном зале или автобусе при свободной рассадке и т.п.). У каждого агента в зависимости от доли агентов своего типа среди окружения задаётся характеристика, олицетворяющая его комфортность при нахождении в данном месте или,

наоборот, желание поменять своё местонахождение. В зависимости от этой характеристики агенты в процессе работы модели могут изменять своё положение. В каждой конкретной реализации как характеристики, так и правила согласования решений и перемещения агентов определяются по-своему и могут существенно отличаться.

Главным результатом модели (как правило, хотя это зависит от конкретной реализации) является тот факт, что, несмотря на выставленную изначально достаточную «толерантность» агентов по отношению к своему окружению, общая картина постепенно сдвигается к всеобщей сегрегации. Так наглядно показывается суть мультиагентной системы: поведение системы в целом выстраивается на основе поведения отдельных агентов, но, вообще говоря, может существенно от него отличаться.

Вообще говоря, изначально Шеллингом рассматривалось целое семейство моделей и поэтому в зависимости от реализации свойства модели могут существенно изменяться, как, например, это показано в [61]. В связи с этим мы для конкретики возьмём одну хорошо описанную и изученную реализацию, приведённую в работе [76]. Вкратце опишем её.

Описание модели Шеллинга по Жангу. Средой для агентов служит доска N на N клеток, представляющая собой город. Агенты представляют домовладельцев (людей или домохозяйства), которые занимают определённые места в среде, представляющие собой частные дома. Основным свойством объектов является цвет. В описываемой модели существуют агенты лишь двух цветов. Агенты располагаются так, что занимают все места, и каждый агент занимает ровно одно место. Окрестность каждого места задана так, что общее количество соседей у каждого агента одинаково и считается равным $2n$. Например, можно считать, что на доске задана топология тора, а окрестность определена по Муру, то есть $n=4$ для всех агентов.

Характеристикой, определяющей уровень комфортности объекта при нахождении в заданном месте является функция полезности $u: N \rightarrow \mathfrak{R}$, ставящая в соответствие количеству соседей своего типа некоторое действительное число.

Функция определяется следующим образом. При отсутствии соседей своего типа она равна нулю. Далее она линейно растёт и достигает максимума Z при n соседей своего типа, то есть, когда окружение агента по признаку «свой-чужой» делится в отношении 50 на 50. Затем, при повышении доли «своих» функция линейно убывает, и при значении, равном $2n$ (то есть, когда окружение состоит только из «своих»), принимает значение $M > 0$.

Таким образом, в функции полезности оказываются отражены следующие факты.

Во-первых, наиболее предпочтительным для каждого агента в отдельности является такое положение, когда «чужие» агенты составляют ровно половину его окружения. А с ростом отклонения от «золотой середины» как в сторону преобладания «чужих», так и «своих», комфортность пребывания агента в данном месте падает.

Во-вторых, комфортность пребывания в окружении, составленном исключительно из «своих» всё же больше, чем при окружении, составленном только из «чужих». Также и для любого другого соотношения, скажем 70% на 30%, комфортность при проживании среди 70% «своих» всё же превосходит комфортность проживания среди 70% чужих, хотя, возможно, и не намного (в зависимости от значения M).

Итак, можно заметить, что функция полезности соответствуют состоянию, когда агенты не просто достаточно толерантны друг к другу, но и предпочитают видеть среди соседей разнообразие. Изобразим функцию полезности на Рисунке А.1.

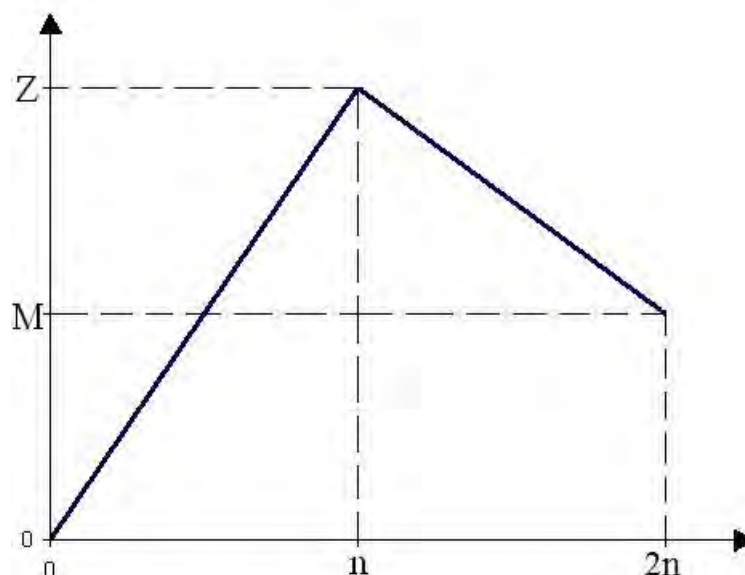


Рисунок А.1 – Функция полезности

Величиной, характеризующей сегрегацию в системе в целом, является количество пар агентов-соседей разного цвета. Чем меньше это количество, тем большей считается сегрегация. Разумеется, в общем случае его необходимо должным образом пронормировать, но в случае заданной изначально размерности модели его можно использовать и как абсолютную величину. Нетрудно заметить, что эта величина равняется сумме количеств всех соседей типа «чужой» для всех объектов, делённой пополам (поскольку все пары в такой сумме считаются дважды).

Правила выбора начального положения и поведения агентов в процессе моделирования определяются следующим образом. Количество агентов одного и другого цвета равны. Изначально они распределяются по клеткам случайным образом, хотя, если нас интересует какой-то конкретный случай, мы можем задать их начальное распределение по местам. Затем моделирование происходит пошагово. На каждом шаге случайным образом выбирается пара агентов разного цвета и решается вопрос – будут ли они меняться местами или нет. Принятие этого решения зависит от суммы функций полезности агентов до и после предполагаемой смены мест. При этом используется логлинейная модель принятия решения, то есть логарифм

отношения вероятностей равен разности сумм функций полезности для различных исходов.

$$\ln \frac{p_2}{p_1} = u_2 - u_1 \quad (\text{A.1})$$

Такая модель часто встречается при моделировании принятия решений.

Всё это можно представить в виде шахматной доски, на которой расставлены чёрные и белые шашки. В процессе моделирования они переставляются согласно определённым выше правилам. Процесс этот потенциально бесконечен, поскольку никак не задано правило для его остановки.

В работе [76] формализованная таким образом модель подвергается детальному анализу, определяются её свойства и поведение, стабильность тех или иных состояний. Главной частью анализа является теоретическое доказательство того, что заданная таким образом модель со временем лишь увеличивает показатель сегрегации, определённому выше. Мы покажем, как доказанные для этой конкретной модели свойства тривиальным образом выводятся из свойств, доказанных выше для всего рассматриваемого семейства МАС.

Функция полезности и характеристическая функция. Введём характеристическую функцию системы, равную принятой автором в [76] мере сегрегации, то есть, количеству пар соседей различных цветов. Свяжем значение изменения функции полезности при обмене выбранных агентов местами с изменением значения характеристической функции. По сути, это сделано в [76], однако мы это сделаем проще и в нужном нам виде и более наглядно.

Обмен агентов разных цветов местами можно представить как смену цвета каждым агентом из выбранной пары на противоположный. Рассмотрим значение функции полезности до и после смены цвета отдельным агентом.

Изобразим эту смену и соответствующее ей изменение значения функции полезности на Рисунке А.2.

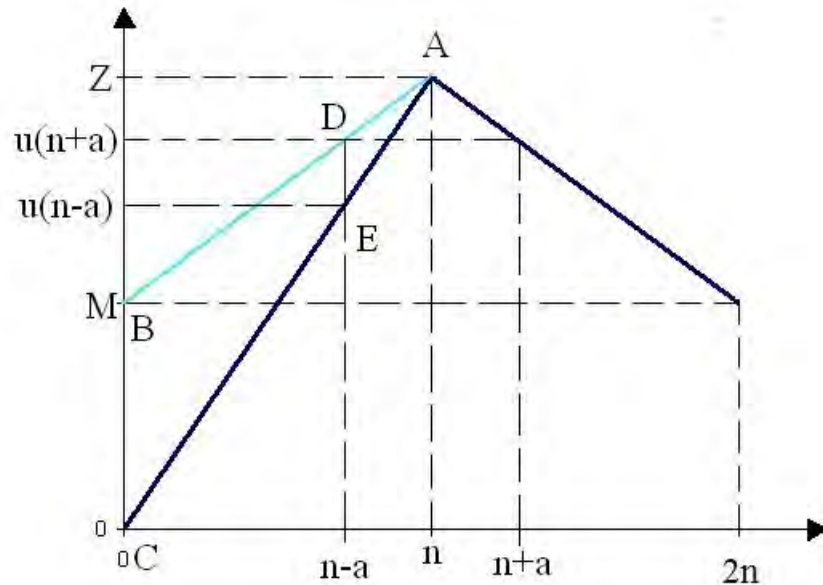


Рисунок А.2 – Изменение значения при смене цвета фишкой

Пусть до обмена количество своих агентов-соседей своего цвета равнялось $n - a$, а после обмена – $n + a$ (где a – произвольное целое, от 0 до n включительно). Соответственно, число «чужих» соседей до обмена равнялось $n + a$, а после обмена – $n - a$. Поскольку у других агентов цвет соседей не менялся, то изменение характеристической функции равно изменению количества «чужих» соседей у рассматриваемого агента.

$$\Delta \rho = (n - a) - (n + a) = -2a. \quad (\text{A.2})$$

Теперь рассмотрим изменение функции полезности. Для этого отобразим значение функции полезности при $x > n$ симметрично относительно $x = n$ и обозначим полученную функцию через $v(x)$. Очевидно, $u(x+a) = v(x-a)$. Тогда изменение функции полезности равно длине отрезка DE . Как видно из графика, треугольники ABC и ADE подобны. Значит, отношения их сторон равны между собой и равны отношению соответствующих высот. Конкретно,

$$\frac{a}{n} = \frac{\Delta u}{M} \Rightarrow \Delta \rho = -2a = -\frac{2n}{M} \Delta u = -\Delta u / \lambda, \quad (A.3)$$

$$\lambda = \frac{M}{2n}.$$

Поскольку при обмене местами цвета меняют два не соседних агента, а разность сумм функций полезности равна сумме разностей, то получаем, что выведенная формула верна и для элементарного шага модели – обмене двух не соседних агентов местами. Так мы наглядно получили тот факт, что *при определённых в [76] правилах действиях агента* изменение полезности обратно пропорционально величине характеристической функции, следовательно, процесс работы агентов более вероятно ведёт к увеличению сегрегации. Вообще говоря, при той же функции полезности, но при других правилах поведения агентов, вышесказанное неверно. Но не будем пока делать выводов, а попытаемся применить к данной модели Теорему 1.

Вероятности перехода и стационарное распределение. Как мы уже заметили выше, для модели принята логлинейная модель вероятности принятия решения. Это означает, что, при выбранной паре агентов, для отношения вероятности выбора решения из двух возможных (переезжать/не переезжать) верно

$$\ln \frac{p_2}{p_1} = u_2 - u_1 \quad (A.4)$$

$$\ln \frac{p_2}{p_1} = \Delta u = -\lambda \Delta \rho \quad (A.5)$$

$$\frac{p_2}{p_1} = \frac{e^{-\lambda \rho_2}}{e^{-\lambda \rho_1}}, p_1 + p_2 = 1 \quad (A.6)$$

Из приведённой выше системы получаем формулу для вероятности переезда

$$p_2 = \frac{e^{-\lambda \rho_2}}{e^{-\lambda \rho_1} + e^{-\lambda \rho_2}}. \quad (A.7)$$

Всё вышесказанное верно для уже выбранной пары агентов. Теперь рассмотрим вероятность перехода всей системы из состояния x в состояние y . Очевидно, что для $x \neq y$ она определяется следующим образом. Во-первых, если y не получается из x сменой цветов пары не соседних агентов, то она равна нулю. Если же это не так, то она равна вероятности выбрать эту пару агентов, умноженной на определённую выше вероятность переезда.

$$\begin{aligned} P_{xy} &= d_{xy} \frac{f(y)}{f(x) + f(y)}, \\ f(x) &= e^{-\lambda \rho_x}, \\ d_{xy} &= \frac{1}{|S(x)|} = \text{const} \end{aligned} \tag{A.8}$$

(зависит от общего числа агентов).

Для случая $x = y$ значение вероятности получаем из формулы

$$P_{xx} = 1 - \sum_{\forall y \in S \setminus \{x\}} P_{xy}. \tag{A.9}$$

Теперь перейдём к условиям Теоремы 1. Определим

$$\Gamma_{xy} = \frac{d_{xy}}{f(x) + f(y)}, h(x) = f(x) \tag{A.10}$$

Очевидно, что $\Gamma_{xy} = \Gamma_{yx}, \Gamma_{xy} \geq 0, h(y) > 0$. Последнее следует из того, что на ограниченном пространстве, где заняты все клетки, всегда будут существовать агенты с хотя бы одним соседом другого цвета. Выполнение условий эргодичности на графе окрестностей доказано в [76]. Таким образом, мы получили выполнение условий Теоремы 1 и её следствия, следовательно, для стационарного распределения верна формула

$$\mu(x) = kh(x); k = \frac{1}{\sum_{x \in S} h(x)}, h(x) = e^{-\lambda \rho x}. \quad (\text{A.11})$$

Так с помощью Теоремы 1 мы более простым способом получили факт, доказанный в [76]. Вероятность нахождения системы тем больше, чем больше в данном состоянии величина сегрегации. Кроме того, мы показали, благодаря чему модель ведёт себя именно таким образом, а именно благодаря определённым образом заданного поведения агентов, нацеленного на описанное выше изменение характеристической функции и принятой модели их деятельности.

2 Игра с логлинейным правилом выбора стратегии игроками

Итак, мы применили Теорему 1 к конкретной модели и доказали утверждение о её поведении, а также показали, что оно выполняется благодаря принятой в модели схемы принятия решения. Опишем теперь, как можно применить следствие Теоремы 1 для абстрактных МАС с различными схемами принятия решения об изменении состояния, не рассматривая их внутреннее устройство. Это позволит нам в дальнейшем упростить процедуру проверки условий Теоремы 1, очертить круг систем, для которых эти условия выполняются.

Модель с выбором из двух решений. Предположим, что, как и в предыдущем примере, переход системы из одного состояния в другое осуществляется поэтапно. Сначала случайным образом выбирается одно из соседних состояний. Затем, в зависимости от значения характеристической функции, принимается решение: переходить в это состояние или нет.

Таким образом, вероятность перехода системы из одного состояния в другое равна

$$P_{xy} = d_{xy}v(x, y), \quad (\text{A.12})$$

где $d_{xy} = \frac{1}{|S(x)|} = \text{const}$ для соседних состояний (положим, что мощность окрестности постоянна) и $d_{xy} = 0$ в противном случае, а $v(x,y)$ зависит от $h(y)$ и характеризует вероятность перехода системы в состояние y . Предположим, $v(x,y) = \frac{h(y)}{w(x,y)}$, то есть определённым образом нормированная характеристическая функция, причём $w(x,y) = w(y,x)$. Тогда можно перейти к условиям теоремы 1, определив

$$\Gamma_{xy} = \frac{d_{xy}}{w(x,y)}. \quad (\text{A.13})$$

Из вышеизложенного получаем $\Gamma_{xy} = \Gamma_{yx}$. Также для выполнения условий теоремы необходима положительность функций w и h и выполнение условий эргодичности на графе окрестностей. Заметим, что случай, доказанный для модели Шеллинга, является частным случаем этой схемы выбора, для которого $w(x,y) = h(x)+h(y)$. В дальнейшем мы рассмотрим другие важные конкретные МАС, для которых верна эта схема выбора и сформулированные выше условия, но функция $w(x,y)$ имеет другой вид.

Модель выбора из многих вариантов. Обобщим схему выбора, принятую в описанной выше модели Шеллинга, на случай выбора из многих параметров. В таком случае вероятность перехода системы из одного состояния в другое будет равна

$$P_{xy} = d_{xy} \frac{h(y)}{\sum_{z \in S(x)} h(z)}. \quad (\text{A.14})$$

В этом случае

$$\Gamma_{xy} = \frac{d_{xy}}{\sum_{z \in S(x)} h(z)} \quad (\text{A.15})$$

$$\Gamma_{yx} = \frac{d_{yx}}{\sum_{z \in S(y)} h(z)} \quad (\text{A.16})$$

и для выполнения условия Теоремы 1 кроме $d_{xy} = d_{yx}$ требуется также $\forall x \forall y: \sum_{z \in S(x)} h(z) = \sum_{z \in S(y)} h(z)$. Это довольно сложно обеспечить в общем случае. Однако это условие становится тривиальным в случае $S(x) = S(y) = S$, то есть, когда окрестность каждого состояния равна всему пространству состояний, то есть возможен переход из любого состояния в любое. Данная схема актуальна для логлиненных моделей принятия решения, применяемых в теории игр.

Обобщённая модель перехода. Представим функцию вероятности перехода из одного состояния в другое в наиболее общем виде (с учётом отношения соседства).

$$P_{xy} = d_{xy} f(x, y) \quad (\text{A.17})$$

При этом вероятность перехода тем больше, чем больше значение характеристической функции.

$$f(x, y_1) > f(x, y_2) \Leftrightarrow h(y_1) > h(y_2) \quad (\text{A.18})$$

Предположим, что для рассматриваемой системы выполняются условия эргодичности и существует стационарное распределение. Посмотрим, выполняется ли для стационарного распределения условие, верное для функции перехода, то есть, увеличивается ли вероятность попадания в то или иное состояния в зависимости от значения характеристической функции. Воспользуемся для этого известной формулой для стационарного распределения и матрицы перехода $\mu P = \mu$. Отсюда получаем следующее.

$$\mu(y_1) = \sum_{x \in S} \mu(x) f(x, y_1) d_{xy_1} \quad (\text{A.19})$$

$$\mu(y_2) = \sum_{x \in S} \mu(x) f(x, y_2) d_{xy_2} \quad (\text{A.20})$$

Таким образом, для выполнения условий на стационарном распределении необходимо также соблюдение условий на функции d_{xy} .

$$h(y_1) > h(y_2), d_{xy_1} \geq d_{xy_2} \Rightarrow \mu(y_1) > \mu(y_2) \quad (\text{A.21})$$

Это условие также получается тривиальным образом, когда $S(x) = S(y) = S$, то есть, когда возможен переход из любого состояния в любое.

Рассмотрим в качестве примера базовую модель конечной игры с n игроками с логлинейным правилом поведения из [58]. Для каждого игрока задано множество стратегий S_i и функция полезности $u_i: S \rightarrow \mathbb{R}$, где $S = S_1 \times \dots \times S_n$ множество состояний игры (то есть, множество всех возможных комбинаций выбранных игроками стратегий). На каждом шаге выбирается случайный игрок и меняет стратегию с учётом функции полезности согласно логлинейному стохастическому правилу. Это означает, что для каждого i -го агента, для каждой пары стратегий i -го игрока s'_i, s''_i и каждого набора статистик всех остальных игроков, который мы обозначим s_{-i} , верно

$$\ln \frac{p(s''_i)}{p(s'_i)} = \beta(u_i(s''_i, s_{-i}) - u_i(s'_i, s_{-i})) \quad (\text{A.22})$$

Кроме этого, по построению, сумма всех вероятностей выбрать ту или иную стратегию равна единице. Отсюда легко показать, что для произвольного агента вероятность выбрать определённую стратегию s_i равна

$$p(s_i) = \frac{e^{\beta u_i(s_i, s_{-i})}}{\sum_{s'_i \in S_i} e^{\beta u_i(s'_i, s_{-i})}} \quad (\text{A.23})$$

Основной проблемой при этом является то, что функции полезности могут существенно отличаться у разных агентов. Однако во многом случае для игры можно ввести потенциальную функцию, что значительно упрощает её анализ. Потенциальной функцией называется функция $v: S \rightarrow \mathbb{R}$, такая что $\forall i = 1 \dots n, \forall s'_i, s''_i \in S_i, \forall s_{-i} \in S_{-i}$, выполняется

$$u_i(s''_i, s_{-i}) - u_i(s'_i, s_{-i}) = v(s''_i, s_{-i}) - v(s'_i, s_{-i}). \quad (\text{A.24})$$

Обозначение S_{-i} используется для множества всех комбинаций стратегий всех агентов кроме i -го. Тогда вероятность i -го агента выбрать стратегию s_i равна

$$p(s_i) = \frac{e^{\beta v(s_i, s_{-i})}}{\sum_{s''_i \in S_i} e^{\beta v(s''_i, s_{-i})}}. \quad (\text{A.25})$$

А для всей игры вероятность перехода от одной комбинации стратегий s к другой s' может быть выражена в следующем виде [58]:

$$P_{ss'} = \sum_{i=1..n} I(s'_{-i} = s_{-i}) \rho_i p(s'_i), \quad (\text{A.26})$$

где I – индикаторная функция (равна единице тогда и только тогда, когда содержащееся в ней выражение верно, иначе равна нулю), ρ_i – вероятность выбрать i -го игрока для процедуры смены стратегии.

Перейдём к нашей терминологии. Набор стратегий всех игроков будет являться состоянием системы. Для произвольной пары состояний s и s' возможны 3 варианта значений индикаторной функции.

1) s и s' отличаются более чем одной стратегией некоторого агента. Тогда $\forall i: I(s'_{-i} = s_{-i}) = 0$, соответственно, $P_{ss'} = 0$.

2) $s = s'$. Тогда $\forall i: I(s'_{-i} = s_{-i}) = 1$. Впрочем, для применения Теоремы 1 этот случай не интересен, поскольку выполнимость условий для него доказывается тривиально ($\Gamma_{ss} = \Gamma_{ss}$).

3) s и s' отличаются ровно одной стратегией некоторого (i -го) агента. Этот случай является для нас самым интересным и значимым. Рассмотрим его подробнее.

Очевидно, что $\forall j = 1..n, j \neq i: I(s'_{-j} = s_{-j}) = 0$. При этом $I(s'_{-i} = s_{-i}) = 1$. Тогда с учётом формулы (A.26) мы можем записать

$$P_{ss'} = \rho_i p(s'_i) = \rho_i \frac{e^{\beta v(s'_i, s_{-i})}}{\sum_{s''_i \in S_i} e^{\beta v(s''_i, s_{-i})}}. \quad (\text{A.27})$$

Выражение в знаменателе симметрично относительно s и s' , поскольку $s'_{-i} = s_{-i}$. Числитель зависит лишь от состояния s' , в которое переходит модель. Значит, условия Теоремы 1 выполнены для

$$\Gamma_{ab} = \frac{\rho_i}{\sum_{\forall s_i \in S_i} e^{\beta v(s_i, b_{-i})}}, \quad (\text{A.28})$$

$$h(b) = e^{\beta v(b)}, \quad (\text{A.29})$$

, следовательно, стационарное распределение имеет вид.

$$\mu(x) = k e^{\beta v(b)}, k = \frac{1}{\sum_{\forall s \in S} e^{\beta v(s)}}. \quad (\text{A.30})$$

Таким образом, применяя следствие Теоремы 1 мы можем легко показать, что для такой игры с течением времени вероятность принятия того или иного набора стратегий тем больше, чем больше значение

потенциальной функции игры на этом наборе. А увеличивая показатель β мы можем добиться всё большего усиления этого эффекта, в пределе вероятность будет равна единице только для оптимума потенциальной функции.

3 Локальный поиск. Алгоритм имитации отжига

До сих пор мы рассматривали МАС, представляющие агентные модели в социально-экономических отношений. Однако, как мы уже отмечали, в рассматриваемое нами семейство МАС входят также мультиагентные реализации алгоритмов поиска оптимального значения функций. Рассмотрим такой широко известный метод случайного поиска как алгоритм имитации отжига (АИО).

АИО рассматривается, например, в работе [22], откуда мы и будем брать его формализацию. Отмечается, что он интересен тем, что одновременно приносит результаты на практике и его сходимость к оптимуму доказана математически (это, вообще говоря, редкость для алгоритмов случайного поиска, которые в общем случае являются эвристиками в чистом виде). Здесь мы покажем, что доказательство его сходимости легко выводится из Теоремы 1.

Процесс работы АИО заключается в следующем: на каждом шаге выбирается случайное значение аргумента из окрестности текущего, и, в зависимости от значения оптимизируемой функции на данном аргументе, принимается решение о переходе к новому аргументу. Вероятность перехода определяется по формуле

$$b_{xy} = \begin{cases} 1, \text{если } f(y) \leq f(x); \\ e^{\frac{f(x)-f(y)}{c}}, \text{если } f(y) > f(x) \end{cases}, \quad (\text{A.31})$$

где $f(x)$ – оптимизируемая (в данном случае, минимизируемая) функция. В классическом АИО значение c изменяется в процессе работы.

Однако, как показано, например, в [22], для однородности представляющей АИО Цепи Маркова полезно временно зафиксировать c . В общем вероятность перехода из одного состояния в другое будет равна

$$P_{xy} = d_{xy}b_{xy}, \quad (\text{A.32})$$

где d_{xy} – вероятность выбрать состояние y при текущем состоянии x .

$$d_{xy} = \begin{cases} 0, \text{если } y \notin S(x) \\ \frac{1}{|S(x)|} \text{иначе} \end{cases} \quad (\text{A.33})$$

Предположим

$$\forall x_1, x_2: |S(x_1)| = |S(x_2)| = \text{const}. \quad (\text{A.34})$$

Представим вероятность перехода в следующем виде

$$b_{xy} = e^{-\frac{f(y)}{c}} l_{xy}, l_{xy} = \begin{cases} e^{\frac{f(y)}{c}}, f(x) \leq f(y) \\ e^{\frac{f(x)}{c}}, f(y) \leq f(x) \end{cases}. \quad (\text{A.35})$$

Очевидно, что мы получили описанный выше случай модели с выбором из двух решений, для которого

$$h(x) = e^{-\frac{f(x)}{c}}, \quad (\text{A.36})$$

$$w(x, y) = l_{xy} = \begin{cases} 1/h(y), h(y) \leq h(x) \\ 1/h(x), h(x) \leq h(y) \end{cases}.$$

Очевидно $w(x, y) = w(y, x)$. Следовательно, если определение окрестности удовлетворяет условию эргодичности, мы получаем выполнение условий Теоремы 1. Таким образом, стационарное распределение для ЦМ, представляющей МАС, реализующую АИО, имеет вид

$$\mu(x) = kh(x) = ke^{-\frac{f(x)}{c}} \quad (\text{A.37})$$

Итак, мы получили, что вероятность нахождения того или иного решения тем больше, чем меньше значение функции в нём.

Приложение Б. Свидетельство о регистрации программы для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2014610632

Программный комплекс «ПАВИСЭР» для организации, хранения, обработки данных, Анализа и Визуализации индикаторов Социально-Экономического и инновационного развития административно-территориальных единиц России

Правообладатель: *Федеральное государственное бюджетное учреждение науки Институт экономики и организации промышленного производства Сибирского отделения Российской академии наук (ИЭОПП СО РАН) (RU)*

Авторы: *Унтура Галина Афанасьевна (RU), Зайцев Иван Дмитриевич (RU), Есикова Татьяна Николаевна (RU)*



Заявка № 2013619952

Дата поступления 31 октября 2013 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 15 января 2014 г.

Руководитель Федеральной службы
по интеллектуальной собственности

Б.П. Симонов

Приложение В. Акты о внедрении

ЕВРОАЗИАТСКИЙ
ТРАНСПОРТНЫЙ
ИННОВАЦИОННЫЙ
ЦЕНТР



EURO-ASIAN
TRANSPORT
INNOVATION
CENTRE

Т/ф. 8-496-762-91-61, т. 8-499-243-40-30, т. 8-916-478-09-95 E..mail: goncharenkoss@mail.ru
Исх. № 14 от 07.06.2014 г.

АКТ

о внедрении результатов
кандидатской диссертационной работы
Зайцева Ивана Дмитриевича

В Евроазиатском транспортном инновационном центре внедрены и используются следующие результаты диссертационной работы Зайцева И.Д. «Многоагентные системы в моделировании социально-экономических отношений: исследование поведения и верификация свойств с помощью цепей Маркова», представленной на соискание ученой степени кандидата технических наук:

1. агентная модели развития транспортной сети, позволяющей моделировать не только развитие экономического и, соответствующего ему, транспортно-экономического пространства, но и учитывающей разнонаправленность интересов экономических субъектов, задействованных в этих процессах, изменение их стратегических интересов с учетом ослабевающей или усиливающейся конкуренции между странами, между транснациональными операторами перевозок и другими акторами;
2. программный комплекс для прогнозирования и оценки крупных международных грузоперевозок, развития транспортной сети, а также для анализа, визуализации и публикации результатов моделирования.

Программный комплекс позволяет в интерактивном режиме реализовывать разные задачи по агентной модели развития транспортной сети, проводить сценарные расчеты и оперативный наглядный анализ получаемых результатов решений. Программный комплекс является полноценной программной системой, имеющей удобный для исследователя пользовательский интерфейс, развитые системы анализа и визуализации, как результатов моделирования, так и произвольных статистических величин, используемых при интерактивной постановки задачи.

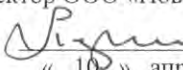
Результаты внедрены при выполнении поисковых исследовательских работ специалистами ЕТИЦ, в том числе: «Экономика и политика в каспийском регионе: ресурсный и транзитный потенциал», «Внутренние водные пути и порты России, как механизм экономической интеграции стран Каспийского региона».

Президент Евроазиатского транспортного инновационного
центра, кандидат экономических наук, профессор
Московской государственной академии водного транспорта

С.С.Гончаренко



«УТВЕРЖДАЮ»
Директор ООО «НовыйСофт-Н»

 М.И. Ларин
« 10 » апреля 2014 г.

Акт

об использовании результатов кандидатской диссертационной работы
Зайцева Ивана Дмитриевича в научно-производственном процессе

Комиссия в составе: председатель директор ООО «НовыйСофт-Н» Ларин М.И., члены: к.ф.-м.н. Семич Д.Ф. составили акт о том, что результаты диссертационной работы Зайцева Ивана Дмитриевича используются в научно-производственном процессе предприятия ООО «НовыйСофт-Н».

Многоагентные системы, алгоритмы, программные системы и методы их исследования на основе цепей Маркова, рассмотренные в диссертации, полезны для анализа макроэкономических процессов в мировой экономике.

Предложенные методики были применены для исследований динамики изменения показателей:

1. ВВП стран мира по ППС, рост (падение) ВВП стран мира.
2. Ставка рефинансирования центральных банков, объем внутренних кредитов.
3. Экспорт и импорт стран мира, сальдо внешней торговли.
4. Внешний государственный долг стран мира.
5. Иностранные инвестиции в странах мира.
6. Инвестиции в зарубежные страны (вывоз капитала).

Предполагается применение предложений автора в будущих разработках компании «НовыйСофт-Н».

Председатель комиссии

Члены комиссии



М.И. Ларин

Д.Ф. Семич