

На правах рукописи



**НГУЕН Ван Тин**

**МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПРОЦЕССОВ  
ИНТЕЛЛЕКТУАЛЬНОЙ МАРШРУТИЗАЦИИ И БАЛАНСИРОВКИ  
ПОТОКОВ ДАННЫХ В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ  
НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ И РОЕВЫХ АЛГОРИТМОВ**

Специальность 2.3.5. «Математическое и программное обеспечение  
вычислительных систем, комплексов и компьютерных сетей»

**АВТОРЕФЕРАТ**

диссертации на соискание ученой степени  
кандидата технических наук

Рязань – 2025

Работа выполнена в ФГБОУ ВО «Рязанский государственный радиотехнический университет имени В.Ф. Уткина» на кафедре «Систем автоматизированного проектирования вычислительных средств».

**Научный руководитель:** **Перепелкин Дмитрий Александрович**, доктор технических наук, профессор, декан факультета вычислительной техники ФГБОУ ВО «Рязанский государственный радиотехнический университет им. В.Ф. Уткина», г. Рязань

**Официальные оппоненты:** **Буранова Марина Анатольевна**, проректор по учебной работе, доктор технических наук, профессор кафедры информационной безопасности ФГБОУ ВО «Поволжский государственный университет телекоммуникаций и информатики» (ФГБОУ ВО «ПГУТИ»), г. Самара

**Кравец Олег Яковлевич**, доктор технических наук, профессор, профессор кафедры автоматизированных и вычислительных систем ФГБОУ ВО «Воронежский государственный технический университет» (ФГБОУ ВО «ВГТУ»), г. Воронеж

**Ведущая организация:** ФГБОУ ВО «Пензенский государственный университет», г. Пенза

Защита диссертации состоится «04» июня 2025 г. в 12:00 часов на заседании диссертационного совета 24.2.375.01, созданного на базе ФГБОУ ВО «Рязанский государственный радиотехнический университет им. В.Ф. Уткина» по адресу: 390005, г. Рязань, ул. Гагарина, д. 59/1.

С диссертацией можно ознакомиться в научно-технической библиотеке ФГБОУ ВО «Рязанский государственный радиотехнический университет им. В.Ф. Уткина» и на сайте [www.rsreu.ru](http://www.rsreu.ru).

Автореферат разослан «\_\_» \_\_\_\_\_ 2025 г.

Ученый секретарь  
диссертационного совета  
доктор технических наук, доцент



А.В. Пруцков

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность темы.** Развитие технологии программно-конфигурируемых сетей (ПКС) совершило революцию в управлении и эксплуатации компьютерными сетями за счет разделения плоскости управления и передачи данных. Такое разделение позволяет централизованно управлять сетью, обеспечивая ее гибкость и адаптацию к изменяющимся условиям. Благодаря своим преимуществам перед традиционными сетями, ПКС стали широко использоваться в современных центрах обработки данных и облачных сервисах.

ПКС имеют большой потенциал для дальнейшего развития с быстрым прогрессом таких технологий, как Интернет вещей (Internet of Things, IoT) и искусственный интеллект (ИИ). В настоящее время ПКС развиваются в направлении интеллектуальной и эффективной автоматизации процессов управления и эксплуатации сетей с использованием решений на базе ИИ, а также в сторону более тесной интеграции с облачными сервисами. С быстрым распространением технологии 5G, ПКС стали занимать ключевую роль в управлении компьютерными сетями, для передачи данных которых требуется высокая пропускная способность и минимальная задержка.

Развитие технологии искусственного интеллекта привнесло значительные улучшения в процессы управления и оптимизации ПКС. ИИ-алгоритмы способны анализировать большие объемы данных, что позволяет с высокой точностью прогнозировать будущие состояния сети. В ПКС, технологии ИИ могут применяться для прогнозирования и классификации трафика, балансировки нагрузки, предсказания DDoS-атак, а также для принятия решений по маршрутизации. Постоянный рост объема данных открывает новые возможности для развития ИИ, способствующего созданию более эффективных решений для эксплуатации сетей ПКС.

По сравнению с традиционными сетями, маршрутизация в программно-конфигурируемых сетях претерпела значительные изменения. Благодаря возможностям обновления информации в режиме реального времени через контроллер и протокол OpenFlow, процессы маршрутизации становятся более гибкими и адаптивными, что позволяет ПКС быстро реагировать на изменения в сети. ПКС обеспечивают программируемость, позволяя применять сложные алгоритмы маршрутизации на основе собранных данных, что способствует оптимизации трафика, повышению производительности и надежности сети.

Вопросам ПКС и протокола OpenFlow посвящены работы Р. Л. Смелянского, Ю. Л. Леохина и др. Методы и алгоритмы маршрутизации в ПКС подробно рассмотрены в работах Д. В. Куракина, В. Н. Тарасова, Д. А. Перепелкина и др. Задачу нахождения кратчайших путей рассматривали в своих трудах ученые Е. W. Dijkstra, R. Bellman, В. А. Евстигнеева. Заметный вклад в разработку методов и алгоритмов управления многопоточным трафиком в компьютерных сетях внесли В. П. Корячко, С. И. Макаренко, О. Я. Кравец. Развитие методов и алгоритмов управления потоками данных с обеспечением качества сервиса в ПКС подробно рассматривается в работах В. Г. Карташевского, М. А. Бурановой, П. Н. Полежаева, С. В. Малахова и др.

Анализ и исследование существующих методов маршрутизации и балансировки потоков данных в ПКС показал, что большинство имеющихся

подходов имеет ряд ограничений, которые значительно снижают эффективность их применения. Во-первых, многие из них не способны адаптироваться к динамическим изменениям в сети, таким как перегрузка трафика или задержки, что может приводить к субоптимальным решениям. Во-вторых, существующие алгоритмы маршрутизации зачастую не обладают достаточной масштабируемостью, что затрудняет их использование в крупных и сложных сетях. Кроме того, высокие вычислительные затраты и время, необходимые для поиска оптимальных маршрутов, делают их менее подходящими для задач, решаемых в режиме реального времени. Наконец, недостаточная устойчивость к отказам и неспособность быстро адаптироваться к изменениям топологии сети также представляют собой значительные проблемы для многих существующих методов.

Таким образом, в настоящее время актуальной является задача разработки математического и программного обеспечения процессов интеллектуальной маршрутизации и балансировки потоков данных в ПКС на основе нейронных сетей и алгоритмов роевого интеллекта с целью создания более адаптивных и эффективных решений. Алгоритмы роевого интеллекта, такие как оптимизация муравьиной колонии, алгоритм стаи птиц, алгоритм искусственной пчелиной колонии и другие, предлагают более гибкие и устойчивые подходы к маршрутизации за счет их способности адаптироваться к изменениям в сети. Искусственные нейронные сети, в свою очередь, могут обеспечить высокую производительность в условиях реального времени благодаря своей способности обучаться и принимать решения на основе большого объема данных. Комбинирование этих методов имеет высокий потенциал для создания новых, более эффективных систем маршрутизации, способных справляться с современными вызовами и требованиями сетевой инфраструктуры.

Работа выполнена в ФГБОУ ВО «Рязанский государственных радиотехнический университет имени В.Ф. Уткина» в рамках научного направления «Автоматизация проектирования и программное обеспечение высокопроизводительных систем и компьютерных сетей».

**Цель и задачи исследования.** Цель работы заключается в повышении эффективности процессов передачи и обработки данных в ПКС за счет разработки новых математических моделей, методов, алгоритмов и программных средств интеллектуальной маршрутизации и балансировки потоков данных на основе искусственных нейронных сетей и методов роевого интеллекта, обеспечивающих высокую точность определения оптимальных маршрутов и равномерное распределение нагрузки в условиях динамически изменяющейся сети.

Для достижения поставленной цели необходимо решить следующие основные задачи:

- провести сравнительный анализ традиционных и программно-конфигурируемых сетей, выделив ключевые преимущества и недостатки каждого подхода;
- провести анализ существующих методов и алгоритмов динамической многопутевой маршрутизации в ПКС;
- разработать математическую модель и метод интеллектуальной маршрутизации в ПКС на основе алгоритмов роевого интеллекта;

- спроектировать модель искусственной нейронной сети для решения задачи многопутевой маршрутизации в ПКС;
- оптимизировать гиперпараметры нейронной сети с помощью алгоритмов роевого интеллекта;
- разработать математическую модель и метод балансировки потоков данных для равномерного распределения нагрузки в ПКС;
- разработать программную систему интеллектуальной маршрутизации и балансировки потоков данных в ПКС;
- провести эксперименты и сравнительный анализ предложенных алгоритмов интеллектуальной маршрутизации и балансировки потоков данных в ПКС и оценить их эффективность.

**Объект исследования:** программно-конфигурируемые сети и методы интеллектуальной маршрутизации и балансировки потоков данных на основе алгоритмов роевого интеллекта и искусственных нейронных сетей.

**Предмет исследования:** средства математического и программного обеспечения интеллектуальной маршрутизации и балансировки потоков данных в ПКС.

**Методы исследования.** Для достижения поставленных целей в работе используются методы теории графов для исследования топологии сети, теория алгоритмов для разработки и анализа алгоритмов маршрутизации на основе роевого интеллекта и искусственных нейронных сетей, а также методы теории матриц для оптимизации маршрутов и оценки их эффективности. Кроме того, для проведения экспериментов и проверки предложенных решений применяются методы компьютерного моделирования, технологии объектно-ориентированного программирования, а также методы статистического анализа.

**Тематика работы** соответствует следующим пунктам паспорта специальности 2.3.5. «Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей»: п.3 «Модели, методы, архитектуры, алгоритмы, языки и программные инструменты организации взаимодействия программ и программных систем»; п.9 «Модели, методы, алгоритмы, облачные технологии и программная инфраструктура организации глобально распределенной обработки данных».

**Научная новизна.** В работе получены следующие результаты, отличающиеся научной новизной:

- математическая модель и метод интеллектуальной маршрутизации в ПКС, отличающиеся использованием алгоритмов роевого интеллекта и их адаптацией для условий динамически изменяющейся сети;
- нейросетевая модель многопутевой маршрутизации в ПКС на основе рекуррентной нейронной сети, позволяющая принимать решения о маршрутизации в режиме реального времени;
- математическая модель и метод оптимизации гиперпараметров нейросетевой модели многопутевой маршрутизации в ПКС на основе алгоритмов роевого интеллекта, обеспечивающие высокую точность прогнозирования маршрутов и снижение вычислительных затрат;
- модель и алгоритм динамической балансировки потоков данных в ПКС, обеспечивающие равномерное распределение нагрузки в сети и адаптацию к

изменяющимся условиям трафика для увеличения пропускной способности и минимизации потерь пакетов;

- архитектура библиотеки программных компонентов интеллектуальной маршрутизации и балансировки потоков данных в ПКС, отличающаяся наличием программных интерфейсов для взаимодействия с сетевыми приложениями и обеспечивающая эффективное управление потоками данных на основе нейронных сетей и роевых алгоритмов;

- структура программной системы для организации распределенной обработки данных, отличающаяся использованием микросервисной архитектуры и возможностью гибкого конфигурирования параметров и структуры сети.

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

1 Математическая модель и метод интеллектуальной маршрутизации в ПКС на основе алгоритмов роевого интеллекта, обеспечивающие гибкую адаптацию к изменениям параметров сети.

2 Нейросетевая модель многопутевой маршрутизации в ПКС с использованием рекуррентных нейронных сетей, позволяющая принимать решения о маршрутизации в режиме реального времени.

3 Математическая модель и метод оптимизации гиперпараметров нейросетевой модели многопутевой маршрутизации в ПКС на основе алгоритмов роевого интеллекта, обеспечивающие высокую точность прогнозирования маршрутов и снижение вычислительных затрат.

4 Модель и алгоритм динамической балансировки потоков данных в ПКС, позволяющие равномерно распределять нагрузку в сети, устраняя перегрузки на отдельных каналах и увеличивая общую пропускную способность.

5 Архитектура библиотеки программных компонентов интеллектуальной маршрутизации и балансировки потоков данных в ПКС, отличающаяся наличием программных интерфейсов для взаимодействия с сетевыми приложениями и обеспечивающая эффективное управление потоками данных в сети на основе нейронных сетей и роевых алгоритмов.

6 Структура программной системы для организации распределенной обработки данных, отличающаяся применением микросервисной архитектуры и возможностью гибкого конфигурирования параметров и структуры сети.

**Степень достоверности.** Обоснованность полученных результатов определяется корректным использованием теории алгоритмов, теории графов, теории множеств, методов компьютерного моделирования и объектно-ориентированного программирования.

**Практическая значимость.** Предложенные алгоритмы интеллектуальной маршрутизации и балансировки потоков данных в ПКС на основе роевого интеллекта и искусственных нейронных сетей способствуют дальнейшему развитию программных средств управления сетевым трафиком в современных компьютерных сетях. Алгоритмы реализованы в составе визуальной программной системы SDNLoadBalancer, включающей графический редактор, эмулятор сети, генератор трафика и комплексную систему мониторинга и интеллектуального управления параметрами ПКС. Разработанная программная система может быть использована для проектирования и оптимизации сетевых инфраструктур, обеспечивая высокую производительность и адаптивность сетей. Программные

компоненты, созданные в рамках данной работы, могут быть интегрированы в существующие системы управления сетью и использованы для динамического распределения нагрузки и повышения надежности сетевых соединений. На элементы разработанных программных средств получены свидетельства о государственной регистрации в реестре Федеральной службы по интеллектуальной собственности.

**Реализация и внедрение результатов работы.** Разработанные в диссертационной работе модели, алгоритмы и программная система интеллектуальной маршрутизации и балансировки потоков данных в ПКС внедрены в учебном процессе на кафедре САПР ВС Рязанского государственного радиотехнического университета имени В.Ф. Уткина (РГРТУ), а также в инженерной практике компании ООО «Технологии НТР» (Вьетнам).

**Апробация результатов диссертации.** Основные результаты диссертационной работы докладывались и обсуждались на следующих всероссийских и международных конференциях: Международный научно-технический форум «Современные технологии в науке и образовании – СТНО» (г. Рязань, 2022 и 2023), Mediterranean Conference on Embedded Computing «MECO» (Budva, Montenegro, 2022), ELEKTRO «ELEKTRO» (Krakow, Poland, 2022), International Russian Automation Conference «RusAutoCon» (Sochi, 2022), International Russian Smart Industry Conference «SmartIndustryCon» (Sochi, 2023), International Symposium Problems of Redundancy in Information and Control Systems «REDUNDANCY» (Moscow, 2023).

**Публикации.** По результатам диссертационного исследования опубликованы 22 научные работы, из них: 7 статей в изданиях из Перечня ведущих рецензируемых научных журналов и изданий ВАК по специальности 2.3.5 (3 – К1, 4 – К2); 5 статей в изданиях, входящих в международные базы научного цитирования Web of Science и Scopus; 3 статьи в других изданиях и материалах конференций; 7 авторских свидетельства о регистрации программ для ЭВМ в ФГБУ «Федеральный институт промышленной собственности» (РОСПАТЕНТ). Все результаты диссертационной работы, включая постановку задач, разработку и исследование методов и алгоритмов, создание программной системы для многопутевой маршрутизации и балансировки потоков данных в программно-конфигурируемых сетях, а также основные научные результаты и выводы, являются результатом самостоятельной работы автора. Направление исследования и концептуальные подходы были определены автором совместно с научным руководителем, что позволило сформулировать актуальные научные задачи и уточнить методологии их решения.

**Структура и объем диссертации.** Диссертационная работа состоит из введения, 5 глав, заключения, списка литературы, 7 приложений, изложенных на 258 страницах (включая 80 рисунков и 18 таблиц). Список литературы содержит 136 наименования.

## СОДЕРЖАНИЕ РАБОТЫ

**Во введении** обосновывается актуальность темы диссертации, формулируются цель и задачи исследований, раскрыта научная новизна работы и ее теоретическая и практическая значимость, представлены основные положения, выносимые на защиту.

**В первой главе** проводится сравнительный анализ традиционных и программно-конфигурируемых сетей. Традиционные сети с распределенной архитектурой имеют ряд ограничений, таких как сложная конфигурация, недостаточная гибкость и слабая масштабируемость. В отличие от них, ПКС с разделением плоскостей управления и передачи данных предлагают новый подход, который позволяет централизованное управление и программируемость, повышая эффективность работы сети. Контроллер ПКС, играющий роль центрального элемента сети, может отслеживать всю сетевую информацию в реальном времени, что позволяет гибко и оперативно применять сетевые политики, адаптируясь к изменениям в условиях сети. Благодаря этому были исследованы известные методы маршрутизации, среди которых особенно выделяются методы многопутевой маршрутизации. Эти механизмы позволяют распределять трафик по нескольким маршрутам в зависимости от текущего состояния сети, вместо использования только одного маршрута, что помогает уменьшить перегрузку и повысить надежность всей сети.

На основе проведенных исследований установлено, что актуальным направлением исследований является разработка математического и программного обеспечения интеллектуальной маршрутизации и балансировки потоков данных в ПКС на основе нейронных сетей и роевых алгоритмов.

**Во второй главе** подробно рассматриваются разработанные математические модели и алгоритмы многопутевой маршрутизации в ПКС на основе нейронных сетей и роевых алгоритмов.

Для решения задачи многопутевой маршрутизации, топология ПКС представляется в виде графа  $G = (V, E)$ , где  $V$  обозначает множество узлов сети, а  $E$  – множество каналов между узлами. Каждый канал  $(v_i, v_j) \in E, v_i, v_j \in V$ , имеет связанный с ним вес  $w_{ij}$ . Вес канала рассчитывается на основе коэффициента загруженности канала следующим простым методом:

$$w_{ij} = \frac{1}{1 - u_{ij}},$$

$$u_{ij} = \frac{bw_{ij}^{used}}{bw_{ij}},$$

где  $u_{ij}$  – коэффициент загруженности канала,  $bw_{ij}^{used}$  – используемая пропускная способность и  $bw_{ij}$  – общая пропускная способность канала.

Каждый маршрут представляет собой последовательность узлов от источника до назначения, не содержащую циклов:  $P_r = [v_s, \dots, v_d]$ , где  $v_s$  и  $v_d$  – исходный и целевой узлы соответственно. Стоимость маршрута представляет собой сумму его весов каналов:

$$w_{P_r} = \sum_{(v_i, v_j) \in P_r} w_{ij}.$$

Задача многопутевой маршрутизации состоит в том, чтобы найти  $K$  различных маршрутов от исходного узла  $v_s$  до целевого узла  $v_d$  так, чтобы



стоимость каждого маршрута была минимальной.

Пусть  $R_M(v_s, v_d)$  обозначается как набор всех различных маршрутов, отсортированных в порядке возрастания по значению стоимости.

$$R_M(v_s, v_d) = \{P_1, P_2, \dots, P_M | w_{P_1} \leq w_{P_2} \leq \dots \leq w_{P_M}\}.$$

Тогда задача многопутевой маршрутизации может быть определена как поиск подмножества  $R_K(v_s, v_d) \subset R_M(v_s, v_d)$  следующим образом:

$$R_K(v_s, v_d) = \{P_1, P_2, \dots, P_K | w_{P_1} \leq w_{P_2} \leq \dots \leq w_{P_K}\}. \quad (*)$$

Может существовать несколько маршрутов одинаковой длины, что означает, что количество маршрутов, удовлетворяющих условию (\*), может превышать значение  $K$ . Следовательно, все эти маршруты могут быть включены в подмножества  $R_K$ . Значение  $K$  выбирается администратором сети с учетом контекста системы. В небольших компьютерных сетях обычно используются все возможные маршруты, а в крупных – значение  $K$  часто ограничивается диапазоном от 2 до 5 для оптимизации производительности и затрат. В процессе исследования выбрано  $K = 4$  для соответствия практическим требованиям.

Для решения задачи многопутевой маршрутизации, алгоритмы роевого интеллекта применяли соответствующие методы представления решения. Для генетического алгоритма и оптимизации муравьиной колонии было предложено использовать метод прямого кодирования маршрута, т. е. каждое решение представлялось как последовательность узлов от источника к пункту назначения, через которые прокладывался маршрут. Для алгоритма стаи птиц, алгоритма искусственной пчелиной колонии и алгоритма светлячков было предложено использовать метод приоритетного кодирования. Каждому узлу в сети назначалось значение приоритета в диапазоне от  $-1.0$  до  $1.0$ . Среди соседних узлов выбирался узел с наивысшим значением приоритета в качестве следующего перехода.

$$\gamma_i \in [-1.0, 1.0], i \in (1, 2, \dots, n),$$

$$next = \underset{j \in A_{cr}}{argmax} \gamma_j,$$

где  $\gamma_i$  – приоритет узла  $i$ ,  $n$  – количество узлов в топологии,  $next$  – следующий узел и  $A_{cr}$  – набор узлов, смежных с текущим узлом, не входящих в маршрут.

Общая структура работы алгоритмов роевого интеллекта для решения поставленной задачи имеет следующий вид.

### **Шаг 1. Инициализация начальных параметров**

Задаются основные параметры алгоритма, включая: размер популяции, максимальное число итераций и характерные параметры для каждого алгоритма.

### **Шаг 2. Создание начальной популяции**

Начальные позиции или состояния агентов генерируются случайным образом в пределах пространства поиска, обеспечивая начальное распределение решений.

### **Шаг 3. Оценка приспособленности**

Рассчитывается значение функции цели для каждого агента, что позволяет оценить качество текущего решения каждого агента. Некорректные решения получают максимальное значение приспособленности, чтобы сделать их менее привлекательными для алгоритма.

### **Шаг 4. Обновление решений и атрибутов агентов**

Позиции и атрибуты агентов обновляются с учетом: информации от соседних

агентов, что отражает принципы взаимодействия в популяции, локально и глобально наилучших найденных решений, случайных факторов, уникальных для каждого алгоритма.

### Шаг 5. Сравнение и обновление набора $K$ лучших решений

В процессе каждой итерации поддерживается список из  $K$  лучших решений, достигнутых агентами, что позволяет отслеживать несколько потенциально оптимальных решений.

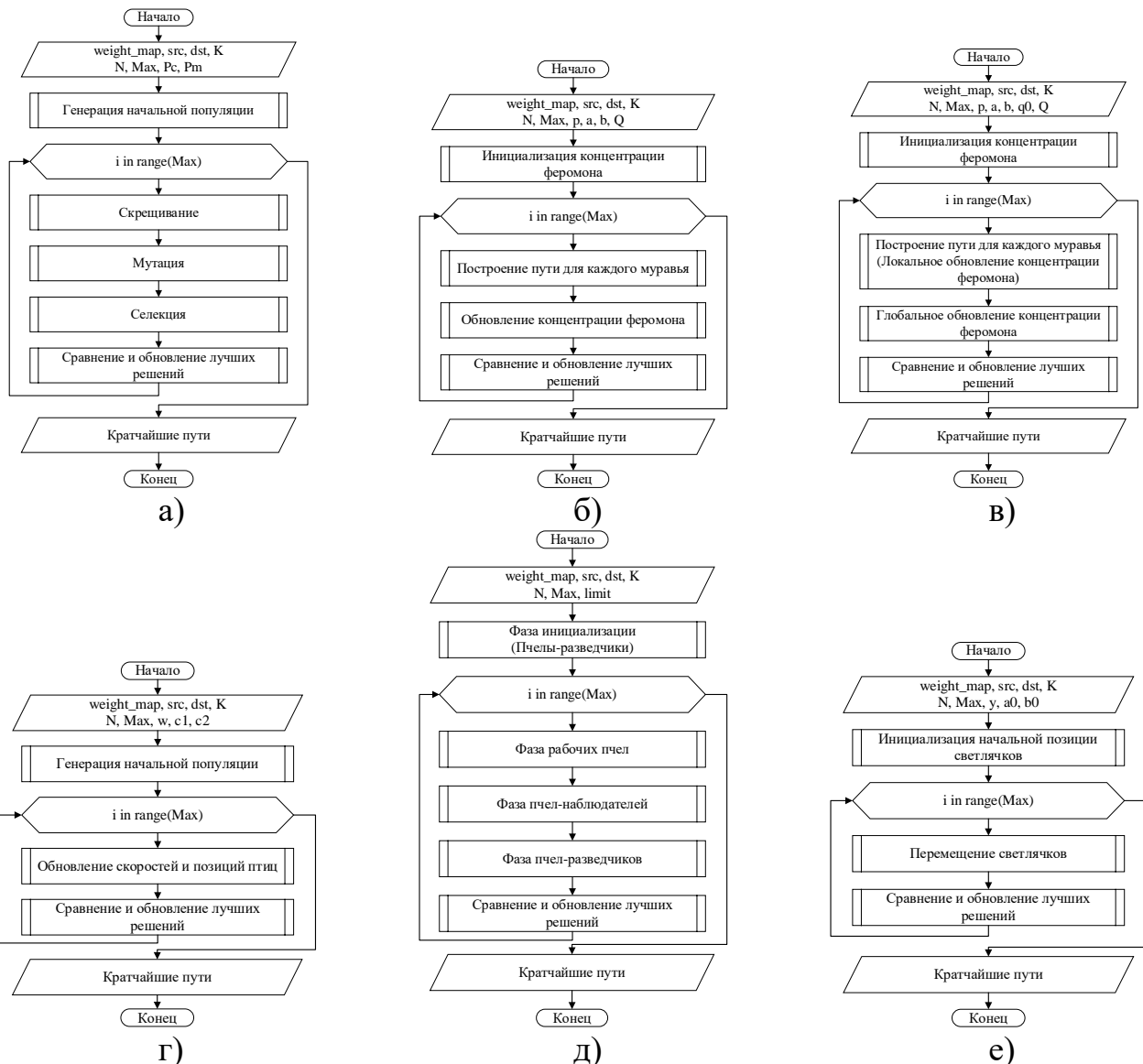
### Шаг 6. Проверка условия останова

Если достигнуто максимальное число итераций, алгоритм завершается.

### Шаг 7. Вывод результатов

По завершении каждого алгоритма выводятся лучшие найденные решения и соответствующие значения приспособленности.

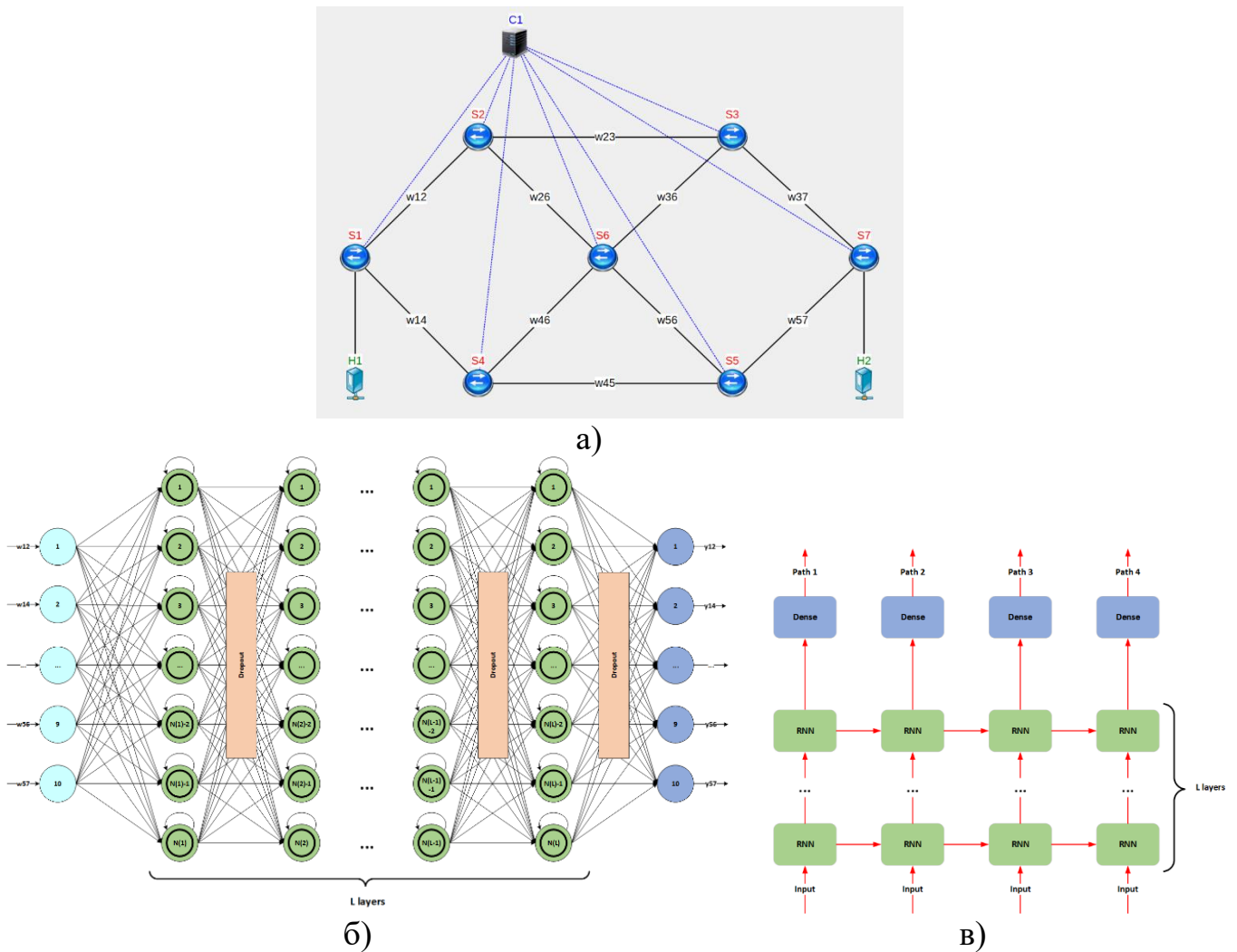
Укрупненные схемы алгоритмов роевого интеллекта для многопутевой маршрутизации в ПКС приведены на рисунке 1.



**Рисунок 1** – Укрупненные схемы алгоритмов роевого интеллекта для многопутевой маршрутизации в ПКС: (а) – генетический алгоритм, (б) – муравьиная система, (в) – система муравьиной колонии, (г) – алгоритм стаи птиц, (д) – алгоритм искусственной пчелиной колонии, (е) – алгоритм светлячков

Помимо роевых алгоритмов, еще одним перспективным подходом к решению проблемы интеллектуальной маршрутизации является использование рекуррентных нейронных сетей (RNN). В отличие от алгоритмов поиска на основе популяции, RNN – это особый тип искусственной нейронной сети, предназначенный для решения задач, связанных с временными рядами или последовательными данными. Это делает их особенно подходящими для решения таких задач, как многопутевая маршрутизация, где последующие маршруты зависят от информации о предыдущих маршрутах (аналогично алгоритму Йена).

В данной работе предложена рекуррентная искусственная нейронная модель для прогнозирования маршрутов, представленная на рисунке 2.



**Рисунок 2** – Модель нейронной сети для многопутевой маршрутизации в ПКС: (а) – топология ПКС из 7 узлов, (б) – полносвязное представление, (в) – свернутое представление

На каждом шаге в качестве входных данных используются веса всех каналов (*Input*), тогда как выходные данные представляют собой кратчайшие маршруты, определенные с применением метода бинарного кодирования (*Output*). Скрытые слои функционируют на основе рекуррентной нейронной сети. Для данной модели используется функция потерь *BinaryCrossentropy*. После RNN-слоев включается полносвязный слой с активационной функцией сигмоида. Для предотвращения переобучения после каждого RNN-слоя применяются слои Dropout. Входные данные нормализуются с использованием метода *StandardScaler*.

Пространство поиска гиперпараметров для модели рекуррентной нейронной сети, предназначенной для поиска кратчайших маршрутов, определялось в соответствии с таблицей 1.

**Таблица 1** – Пространство поиска гиперпараметров

Гиперпараметры	Множество значений
Количество скрытых слоев	[1, 2, 3, 4, 5]
Количество нейронов в скрытом слое	[10, 20, ..., 190, 200]
Dropout	[0.0, 0.1, 0.2, 0.3, 0.4, 0.5]
Тип ячейки	[SimpleRNN, GRU, LSTM]
Алгоритм оптимизации	[SGD, RMSprop, Adam]
Скорость обучения	[0.0001, 0.001, 0.01, 0.1]
Размер батча	[16, 32, 64, 128, 256, 512, 1024]

Чтобы применить роевые алгоритмы к задаче оптимизации гиперпараметров, использовался метод представления набора гиперпараметров в соответствии с таблицей 2.

**Таблица 2** – Представление набора гиперпараметров

Слой 1	Dropout 1	...	Тип ячейки	Алгоритм оптимизации	Скорость обучения	Размер батча
$N_1$	$d_1$	...	$T$	$A$	$lr$	$b$

В методе представления решений, значения представляли собой позиции значений гиперпараметров в наборе значений таблицы 1. Например,  $b = 2$  означало, что размер батча равен 64, а аналогичные принципы применялись и к другим гиперпараметрам. При выполнении алгоритмов роевого интеллекта для оптимизации гиперпараметров результаты вычислений округлялись до ближайшего целого числа.

Для сокращения времени и затрат на оптимизацию гиперпараметров нейронных сетей применялись следующие подходы: использование подмножества данных вместо полного набора, остановка обучения при отсутствии улучшений точности на проверочном наборе в течение 10 эпох, а также сохранение результатов обучения во время оптимизации. В случае равной точности на тестовом наборе предпочтение отдавалось модели с меньшим количеством параметров.

Для решения поставленной задачи оптимизации гиперпараметров, алгоритмы роевого интеллекта были реализованы следующим образом.

- **Инициализация популяции:** В контексте оптимизации гиперпараметров нейронной сети, каждый агент в популяции представляет собой отдельный набор гиперпараметров модели. Первоначальные решения генерируются случайным образом в пределах предопределенных границ для каждого параметра, как указано в таблице 1.

- **Оценка приспособленности:** Приспособленность каждого решения определяется его производительностью, измеряемой как точность на тестовом наборе данных. Более высокая точность указывает на лучший набор гиперпараметров.

- **Итеративная оптимизация:** Поиск оптимальных гиперпараметров

осуществляется посредством повторяющихся циклов оценки и обновления. Каждая итерация обновляет популяцию на основе оценок приспособленности и конкретных правил алгоритма.

- **Критерии остановки:** Процесс оптимизации останавливается, когда достигается максимальное количество итераций.

- **Оптимальные гиперпараметры:** Решение с самой высокой точностью на проверочном наборе данных выбирается в качестве оптимального набора гиперпараметров. Затем этот оптимальный набор используется для обучения окончательной модели на полном обучающем наборе, чтобы подготовить ее к реальному тестированию или развертыванию.

**В третьей главе** разработаны математическая модель и алгоритм динамической балансировки потоков данных в ПКС (Dynamic Adaptive Multipath Load Balancing, *DAMLB*).

Пусть в ПКС имеется  $L$  потоков, которые необходимо маршрутизировать:  $Flows = \{flow_1, flow_2, \dots, flow_L\}$ . Каждый поток  $flow_i$  описывается следующим образом:  $flow_i = \{v_{src}(flow_i), v_{dst}(flow_i), \lambda(flow_i)\}$ , где  $v_{src}(flow_i)$  и  $v_{dst}(flow_i)$  – исходный и конечный коммутатор,  $\lambda(flow_i)$  – требуемая пропускная способность. Для каждого потока  $flow_i$  общий трафик  $\lambda(flow_i)$  распределен по набору  $K$  маршрутов  $P_1, P_2, \dots, P_K$  от  $v_{src}(flow_i)$  до  $v_{dst}(flow_i)$ :

$$\sum_{k=1}^K \lambda_k(flow_i) = \lambda(flow_i),$$

где  $\lambda_k(flow_i)$  – это часть трафика, распределенная по маршруту  $P_k$  для потока  $flow_i$ .

Все потоки данных должны быть распределены по маршрутам таким образом, чтобы общий трафик через каждый канал  $e_{ij}$  на маршруте не превышал его общую пропускную способность  $C_{ij}$ :

$$\sum_{\{flow_i | e_{ij} \in P_k(flow_i)\}} \lambda_k(flow_i) \leq C_{ij} \quad \forall e_{ij} \in E.$$

Стоимость маршрута  $P_k$  рассчитывается как сумма стоимостей  $w_{ij}$  каналов, принадлежащих этому маршруту:

$$w_{P_k} = \sum_{e_{ij} \in P_k} w_{ij}.$$

Задача балансировки потоков данных состоит в том, чтобы найти маршруты  $P_1, P_2, \dots, P_K$  и распределить трафик  $\lambda(flow_i)$  по этим маршрутам так, чтобы минимизировать общую стоимость сети и избежать перегрузок:

$$\min \sum_{flow_i \in Flows} \sum_{k=1}^K w_{P_k} \times \lambda_k(flow_i).$$

Для оценки эффективности распределения трафика между каналами, используется индекс Джейна (Load Balancing Index, *LBI*), который измеряет, насколько равномерно распределена загрузка между всеми каналами:

$$LBI = \frac{(\sum_{e_{ij} \in E} u_{ij})^2}{|E| \times \sum_{e_{ij} \in E} u_{ij}^2},$$

где  $|E|$  – количество каналов и индекс принимает значение от 0 до 1:  $LBI = 1$  означает полную равномерность в распределении трафика между каналами и  $LBI = 0$  указывает на крайнее неравенство.

Предложенный в работе подход динамической адаптивной балансировки потоков данных в ПКС построен на тесной координации четырех основных компонентов.

**Компонент мониторинга топологии** постоянно обновляет информацию о состоянии сети, позволяя построить полный граф  $G = (V, E)$ , точно отражающий текущую топологию сети. Каждый канал  $e_{ij} \in E$  имеет следующие атрибуты:  $e_{ij} = \{port_{v_i}, port_{v_j}, w_{ij}\}$ , где  $port_{v_i}$  и  $port_{v_j}$  – порты на коммутаторах  $v_i$  и  $v_j$ ,  $w_{ij}$  – вес канала, первоначально инициализируется как  $w_{ij} = 1$ . Когда в сетевой топологии происходят изменения, такие как добавление или удаление коммутатора или изменение состояния канала связи, граф  $G = (V, E)$  обновляется.

**Компонент мониторинга портов** отслеживает трафик на портах и периодически рассчитывает пропускную способность соединений, определяя стоимость канала и индекс балансировки нагрузки на основе коэффициента загруженности канала.

$$LBI = \frac{(\sum_{e_{ij} \in E} u_{ij})^2}{|E| \times \sum_{e_{ij} \in E} u_{ij}^2},$$

$$w_{ij} = \begin{cases} 100, & \text{если } u_{ij} = 1 \\ \frac{1}{1-u_{ij}}, & \text{если } 0 \leq u_{ij} < 1 \end{cases}$$

**Компонент мониторинга потоков** непрерывно обновляет информацию о потоках, проходящих через каждый коммутатор. Множество потоков между двумя коммутаторами  $v_i$  и  $v_j$ , и их пропускная способность, записываются в виде:  $switch\_to\_switch\_flows\_speed_{ij} = \{speed_{ij}^{f_1}, speed_{ij}^{f_2}, \dots, speed_{ij}^{f_k}\}$ , где  $speed_{ij}^{f_k}$  – это пропускная способность потока  $f_k$ , проходящего между коммутаторами  $v_i$  и  $v_j$ .

Вся информация о стоимости и пропускной способности, собранная компонентами мониторинга, передается в **компонент многопутевой маршрутизации и балансировки потоков данных**, что позволяет системе динамически маршрутизировать потоки трафика по оптимальным маршрутам. При изменении топологии сети или увеличении трафика система автоматически адаптируется и перенаправляет потоки по мере необходимости для поддержания эффективной балансировки нагрузки и обеспечения качества обслуживания.

### Этап 1. Динамическая многопутевая маршрутизация

**Шаг 1.** Извлечение информации из заголовков IPv4 и TCP/UDP пакетов

Информации о клиенте и сервере представляются в виде двух массивов:

$$Source = \{IP_{host\_src}, v_{src}, port_{host\_src}, port_{v_{src}}\},$$

$$Destination = \{IP_{host\_dst}, v_{dst}, port_{host\_dst}, port_{v_{dst}}\},$$

где  $IP_{host\_src}$  и  $IP_{host\_dst}$  – IP-адреса клиента и сервера,  $v_{src}$  и  $v_{dst}$  – коммутаторы, подключенные к клиенту и серверу,  $port_{host\_src}$  и  $port_{host\_dst}$  – TCP/UDP порты клиента и сервера.

**Шаг 2.** Расчет и установка оптимального маршрута

Разработанные алгоритмы роевого интеллекта или проектированная модель нейронной сети используются для вычисления  $K$  маршрутов, с наименьшей стоимостью между  $v_{src}$  и  $v_{dst}$ :

$$P_1, P_2, \dots, P_K = \underset{P_k}{argmin}(w_{P_k}), k \in (1, 2, \dots, K).$$

### Шаг 3. Распределение трафика

После того как оптимальные маршруты  $P_1, P_2, \dots, P_K$  между исходным коммутатором  $v_{src}$  и конечным коммутатором  $v_{dst}$  найдены, трафик между клиентом и сервером разбивается на несколько отдельных потоков. Определение этих потоков зависит от используемой стратегии балансировки нагрузки. В данной работе, Iperf3 используется для инициализации параллельных потоков между каждой парой клиента и сервера, поэтому потоки имеют одинаковые IP-адреса источника и назначения ( $IP_{host\_src}$  и  $IP_{host\_dst}$ ), но отличаются друг от друга портами источника и назначения ( $port_{host\_src}$  и  $port_{host\_dst}$ ).

Для каждого маршрута  $P_k$ , рассчитывается вес  $W_k$ , который обратно пропорционален стоимости маршрута  $w_{P_k}$ :  $W_k = \frac{1}{w_{P_k}}$ . Затем каждый вес  $W_k$  нормализуются и преобразуются в целые числа:

$$W_k^{normalized} = \text{round}\left(\frac{10 \times W_k}{\sum_{i=1}^K W_i}\right).$$

Для распределения потоков между доступными маршрутами используется метод взвешенного циклического распределения. Цикл  $C$  строится таким образом, чтобы маршруты  $P_1, P_2, \dots, P_K$  появлялись в строго упорядоченной последовательности. При этом маршрут  $P_k$  должен появляться ровно  $W_k^{normalized}$  раз. Потоки данных распределяются последовательно в соответствии с порядком в цикле  $C$ . После завершения одного цикла процесс продолжается с начала цикла до тех пор, пока все потоки не будут распределены по маршрутам.

### Шаг 4. Установка правил передачи

После определения маршрута для каждого потока компонент устанавливает правила передачи на каждом коммутаторе вдоль этого маршрута:

$$\begin{aligned} \text{Flow\_entry}_k(v_i) = \\ (\text{match fields} = \{IP_{host\_src}, IP_{host\_dst}, port_{host\_src}, port_{host\_dst}\}, \\ \text{actions} = \{\text{output} = \text{out\_port}_{v_i}\}), \end{aligned}$$

где *match fields* – это информация о пакете, которую коммутатор использует для идентификации потока, соответствующего предопределенным полям, *actions* – это действия, которые коммутатор выполняет, если пакет соответствует полям соответствия.

## Этап 2. Адаптивная многопутевая маршрутизация

### Шаг 1. Обнаружение перегруженных каналов

Для канала между двумя коммутаторами  $v_i$  и  $v_j$  перегрузка обнаруживается, если его стоимость превышает определенный порог  $\theta$ . Список перегруженных каналов определяется следующим образом:  $overloaded\_links = \{(v_i, v_j) : w_{ij} \geq \theta\}$ .

### Шаг 2. Определение перегруженных потоков

Для каждого перегруженного канала  $e_{ij}$ , извлекается потоки  $F_{ij} = \{f_1, f_2, \dots, f_n\}$ , которые проходят через этот канал. Все потоки  $F_{ij}$  объединяются в единый список  $F_{all}$ , содержащий уникальные потоки, проходящие через все перегруженные каналы:  $F_{all} = \cup_{e_{ij} \in overloaded\_links} F_{ij}$ .

Рассчитывается суммарная скорость каждого потока через перегруженные каналы:

$$speed_{total}(f) = \sum_{e_{ij} \in \text{overloaded\_links}} speed_{ij}^f.$$

Список потоков  $F_{all}$  сортируется по  $speed_{total}(f)$  в порядке убывания. Это гарантирует, что потоки, которые вносят наибольший вклад в перегрузку, будут рассмотрены в первую очередь.

Инициализируется множество потоков для перенаправления  $F_{redirect} = \emptyset$ . Итеративно просматривается список потоков, отсортированных по убыванию:

- Временно добавляется поток  $f$  в  $F_{redirect}$ .
- Проверяется, что после удаления всех потоков из  $F_{redirect}$ , оставшиеся потоки на каждом канале  $e_{ij} \in \text{overloaded\_links}$  не вызывают перегрузку:

$$F_{ij}^{remain} = F_{ij} \setminus F_{redirect},$$

$$speed(F_{ij}^{remain}) = \sum_{f \in F_{ij}^{remain}} speed_{ij}^f,$$

$$w_{ij}(F_{ij}^{remain}) = \frac{1}{1 - \frac{speed(F_{ij}^{remain})}{bw_{ij}}},$$

где  $F_{ij}^{remain}$  – список оставшихся потоков на канале  $e_{ij}$ ,  $speed(F_{ij}^{remain})$  – суммарная скорость оставшихся потоков и  $w_{ij}(F_{ij}^{remain})$  – стоимость канала после удаления подмножество потоков  $F_{redirect}$ .

- Если все каналы удовлетворяют  $w_{ij}(F_{ij}^{remain}) \leq \theta$ , процесс завершен.
- В противном случае переходим к следующему потоку.

В результате  $F_{redirect}$  является минимальным множеством потоков, которые нужно перенаправить.

### Шаг 3. Корректировка стоимости каналов

Когда необходимо перенаправить перегруженные потоки, важно учесть, что если при расчете маршрутов используется текущая стоимость каналов, включающая пропускную способность перегруженных потоков, то результат может быть не оптимальным. Чтобы избежать этого, стоимость каналов пересчитывается, исключив пропускную способность потоков  $f \in F_{redirect}$  на всех каналах, через которые проходят эти потоки.

$$u_{ij}^{new} = u_{ij}^{used} - \sum_{f \in F_{redirect}} speed_{ij}^f,$$

$$w_{ij}^{new} = \frac{1}{1 - \frac{u_{ij}^{new}}{bw_{ij}}}.$$

### Шаг 4. Перерасчет маршрутов для перегруженных потоков

Новые маршруты для потоков  $F_{redirect}$  рассчитываются на основе обновленных метрик сети.

$$P'_1, P'_2, \dots, P'_K = \underset{P'_k}{\operatorname{argmin}}(w_{P'_k}^{new}), k \in (1, 2, \dots, K).$$

### Шаг 5. Распределение перегруженных потоков по новым маршрутам

Перегруженные потоки распределяются также по маршрутам  $P'_1, P'_2, \dots, P'_K$  согласно описанному методу взвешенного циклического распределения.

### Шаг 6. Обновление правил передачи

После перерасчета новых маршрутов  $P'_k$ , компонент обновляет правила передачи на каждом коммутаторе вдоль нового маршрута  $Flow\_entry'_k(v_i)$ , заменяя старые правила  $Flow\_entry_k(v_i)$ .



В четвертой главе разработана визуальная программная система интеллектуальной маршрутизации и балансировки потоков данных в ПКС SDNLoadBalancer. Это передовой программный инструмент, разработанный на платформе PyQt5 и предназначенный для организации распределенной обработки потоков данных, проектирования и управления средами ПКС. Система предоставляет интуитивно понятный пользовательский интерфейс, позволяющий сетевым инженерам и исследователям легко создавать, настраивать и исследовать различные топологии ПКС.

Визуальная программная система SDNLoadBalancer доступна на платформе Linux и включает следующие компоненты: графический редактор, эмулятор ПКС, генератор трафика и комплексную программную систему мониторинга и интеллектуального управления ПКС.

- SDNLoadBalancer интегрирует эмулятор Mininet, который позволяет создавать виртуальные коммутаторы, маршрутизаторы и хосты для моделирования топологии ПКС, что помогает тестировать и экспериментировать с сетевыми структурами без необходимости использовать реальное оборудование.
- Контроллер Ryu отвечает за мониторинг и управление ПКС. Ryu предоставляет API для координации коммутаторов в Mininet с целью реализации сетевых политик, таких как маршрутизация и управление потоками данных в сети.
- Iperf3 используется для генерации трафика и проверки пропускной способности между узлами сети в эмуляторе Mininet.

На рисунке 3 показана разработанная структура визуальной программной системы проектирования и эмуляции ПКС SDNLoadBalancer, а диаграммы классов основных компонентов приведены на рисунках 4 – 6.



Рисунок 3 – Структура визуальной программной системы интеллектуальной маршрутизации и балансировки потоков данных в ПКС SDNLoadBalancer

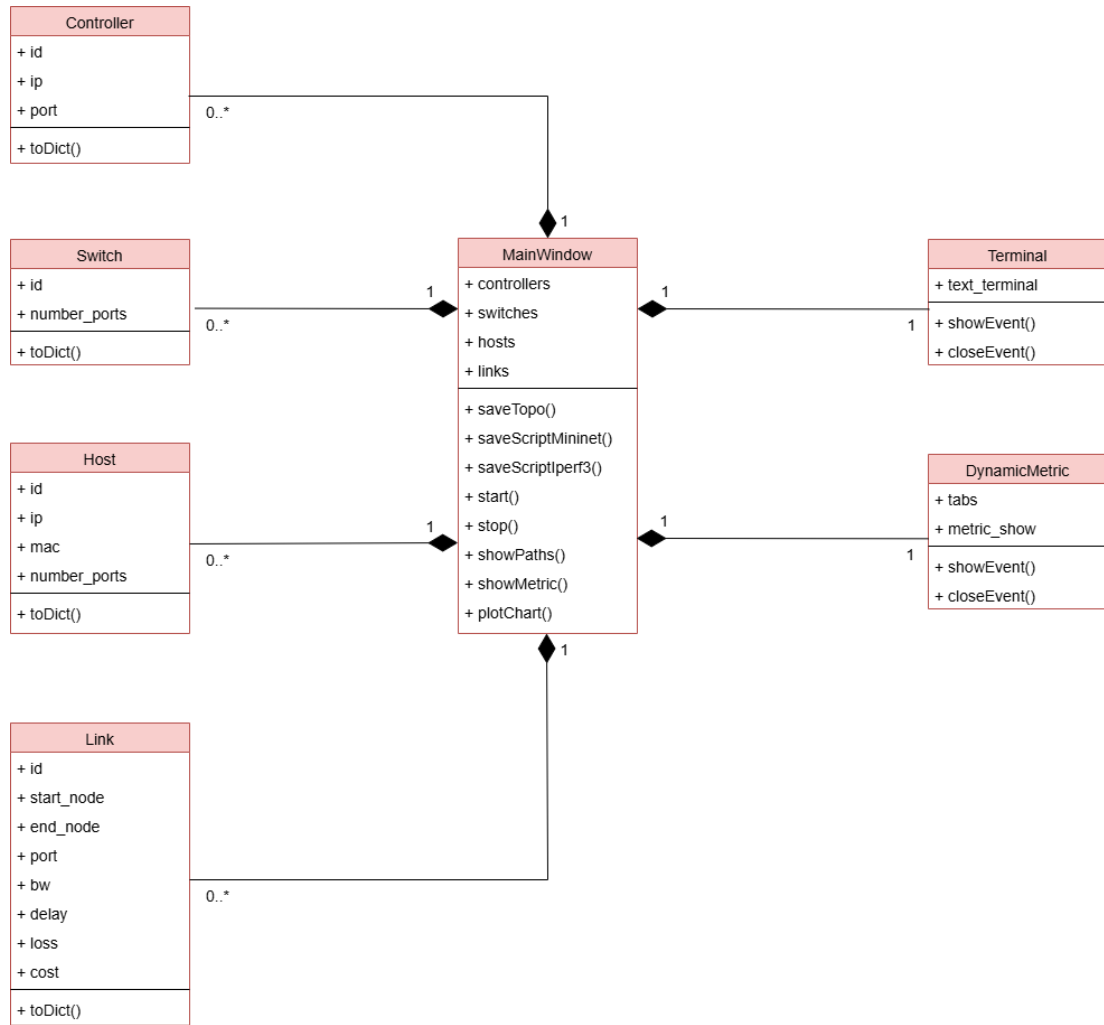


Рисунок 4 – Диаграмма основных классов графического редактора

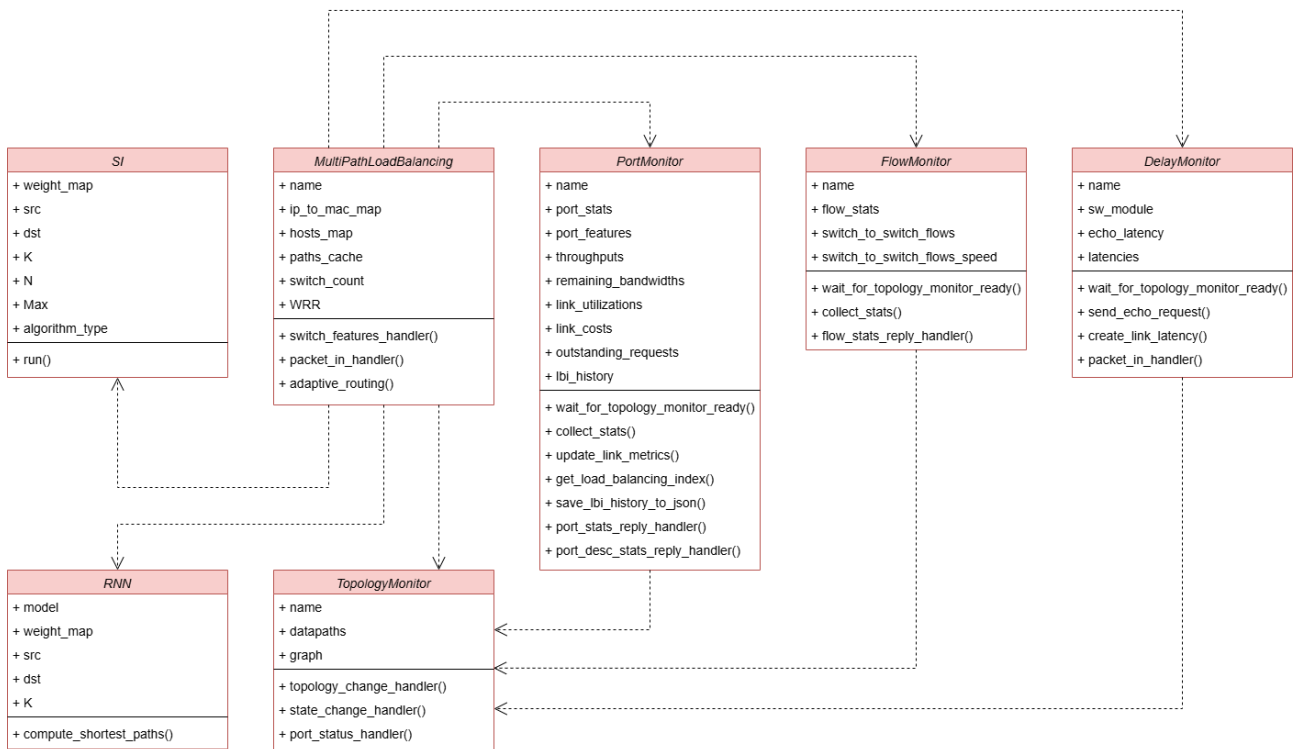
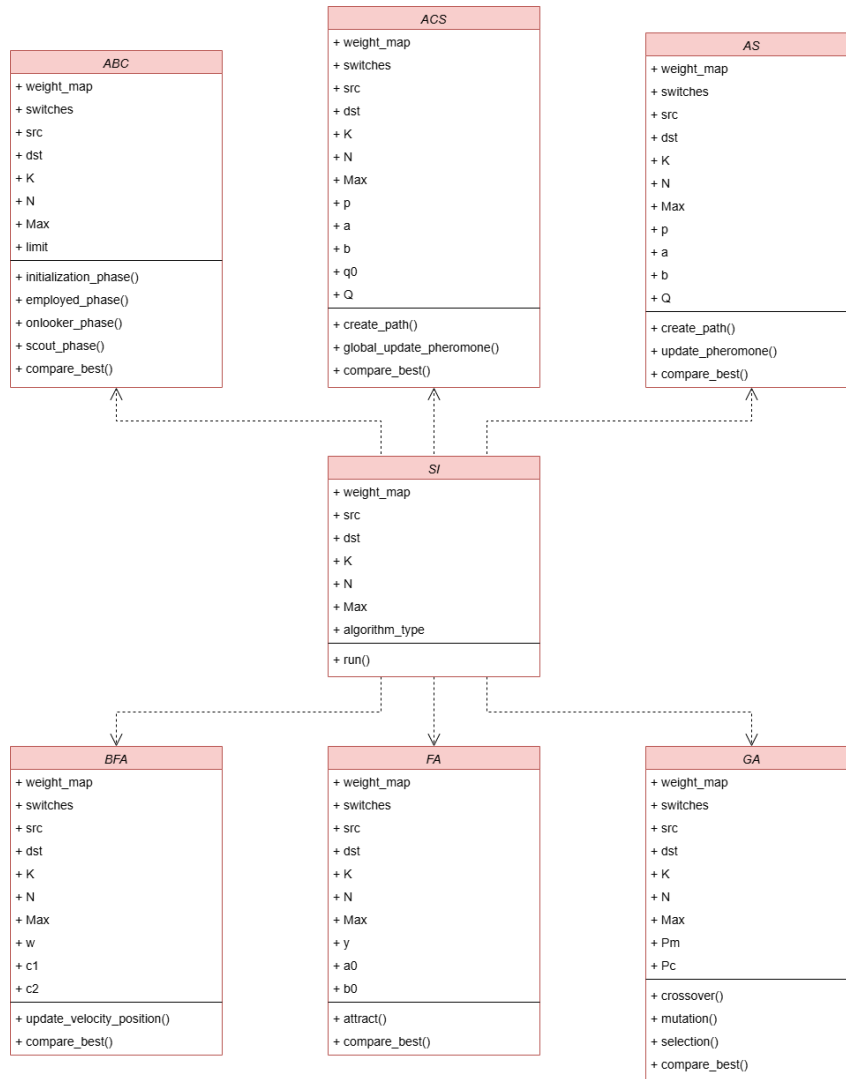


Рисунок 5 – Диаграмма основных классов комплексной программной системы мониторинга и интеллектуального управления ПКС

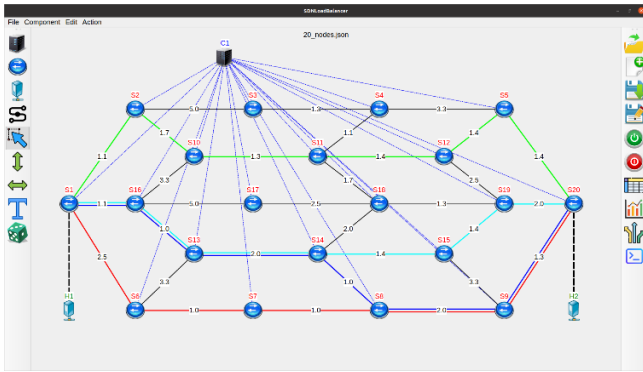


**Рисунок 6** – Диаграмма основных классов компонента интеллектуальной маршрутизации потоков данных на основе алгоритмов роевого интеллекта

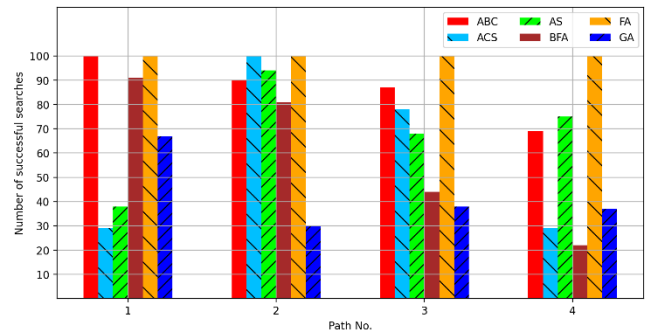
После проектирования и настройки структуры ПКС через графический интерфейс система автоматически генерирует соответствующие сценарии для Mininet, Ryu и Iperf3. Всего одним щелчком мыши можно запустить эксперимент без необходимости вручную вводить команды в терминале. В процессе эксперимента динамические сетевые метрики, такие как пропускная способность, задержка и стоимость, отображаются в виде таблицы в режиме реального времени. Результаты эксперимента сохраняются в формате .json, что позволяет пользователю легко строить графики для дальнейшего анализа и исследования производительности сети.

**В пятой главе** приведены результаты экспериментальных исследований разработанного математического и программного обеспечения интеллектуальной маршрутизации и балансировки потоков данных в ПКС на основе нейронных сетей и роевых алгоритмов.

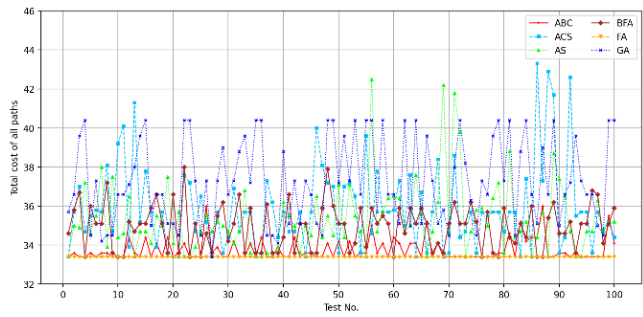
Для исследования методов и алгоритмов интеллектуальной маршрутизации в ПКС рассматривалась топология ПКС в соответствии с рисунком 7(а). Каждый алгоритм был запущен 100 раз для точной оценки его эффективности. Полученные результаты сравнивались по таким показателям, как количество успешных поисков для каждого маршрута, общая стоимость всех маршрутов и время выполнения алгоритма.



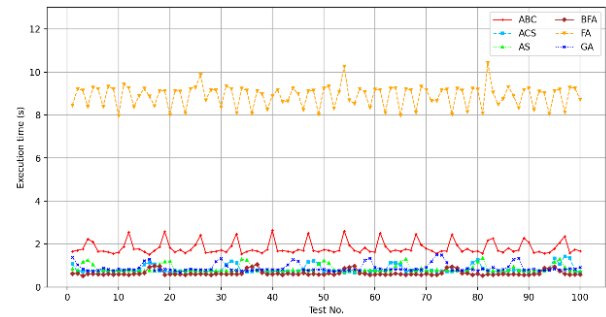
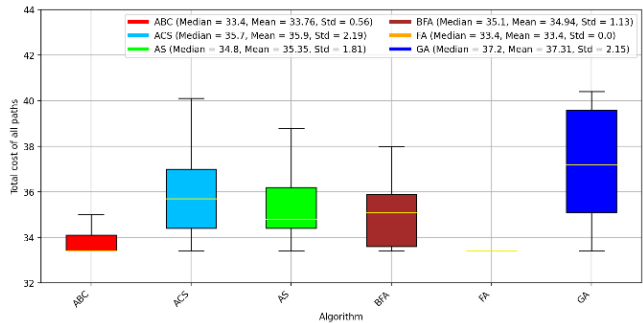
а)



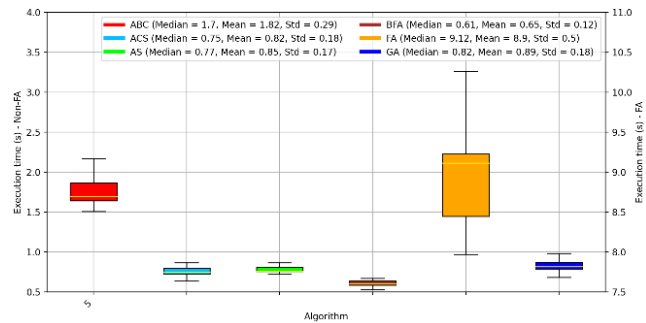
б)



в)



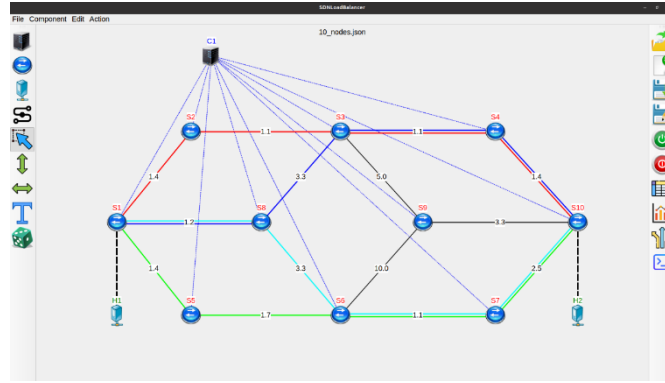
г)



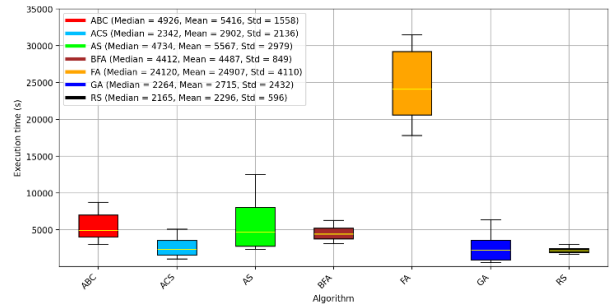
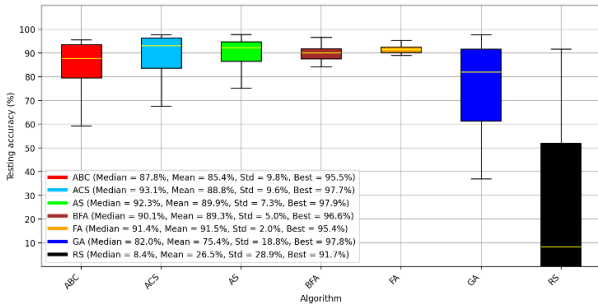
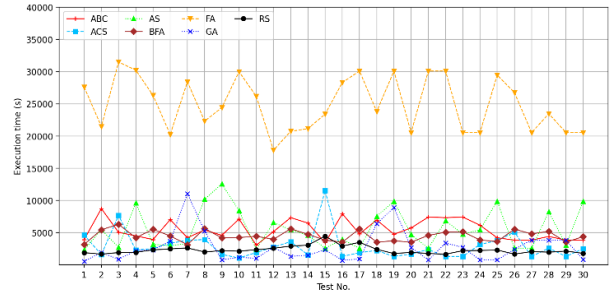
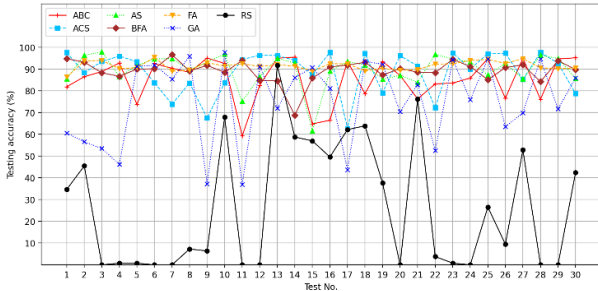
**Рисунок 7** – Исследование методов многопутевой маршрутизации на основе роевого интеллекта: (а) – экспериментальная топология ПКС из 20 узлов, (б) – количество успешных поисков для каждого маршрута, (в) – общая стоимость всех маршрутов для каждого алгоритма, (г) – время выполнения каждого алгоритма

В ходе проведенных исследований установлено, что алгоритм светлячков (FA) обеспечивает наивысшую точность, но его время выполнения долгое и нестабильное, что делает его подходящим для задач, требующих абсолютной точности без строгих требований к времени выполнения. Алгоритм искусственной пчелиной колонии (ABC) также достигает почти оптимальной точности при приемлемом времени выполнения. Алгоритм системы муравьиной колонии (ACS) и алгоритм муравьиной системы (AS) отличаются быстрой и стабильной работой, что делает их идеальными для задач, требующих высокой скорости обработки и среднего уровня точности. Алгоритм стаи птиц (BFA) имеет самое короткое и стабильное время выполнения, являясь оптимальным выбором при приоритете скорости и стабильности. Генетический алгоритм (GA) обладает широкой способностью к исследованию, но ему не хватает стабильности как по времени, так и по точности.

В качестве базовой линии для сравнения с алгоритмами роевого интеллекта при оптимизации гиперпараметров проектированной модели нейросети использовался метод случайного поиска (RS).

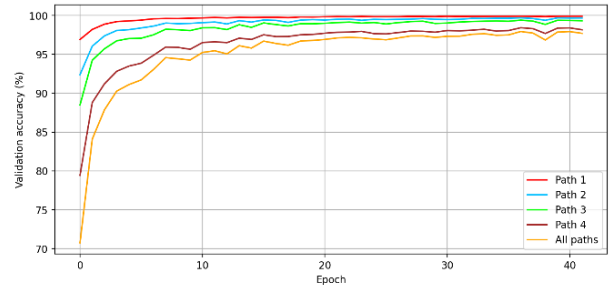
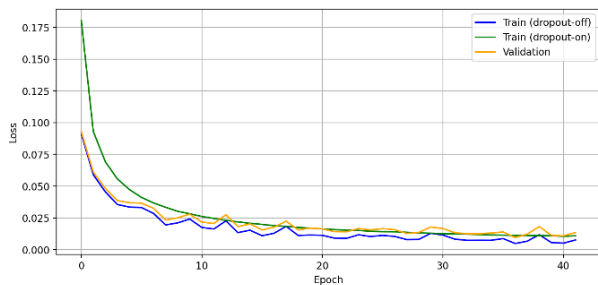
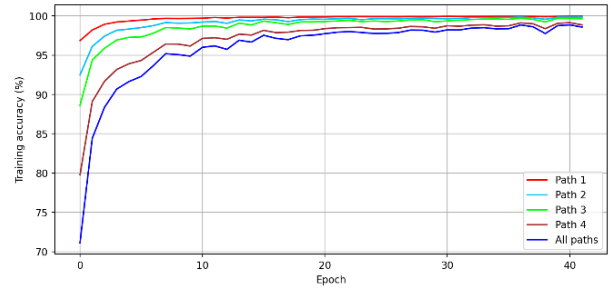
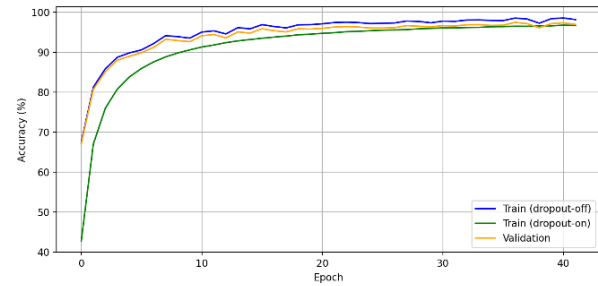


а)



б)

в)



г)

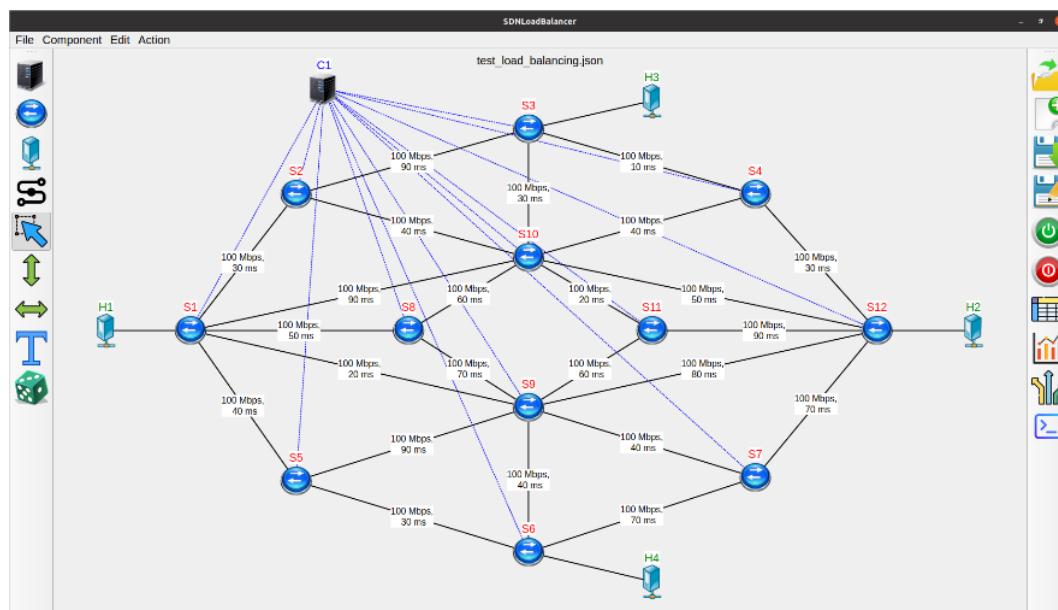
д)

**Рисунок 8** – Исследование методов многопутевой маршрутизации на основе нейронных сетей: (а) – экспериментальная топология ПКС из 10 узлов, (б) – точность тестирования каждого алгоритма, (в) – время выполнения каждого алгоритма, (г) – точность и значение функции потерь модели по эпохам, (д) – точность модели для каждого маршрута

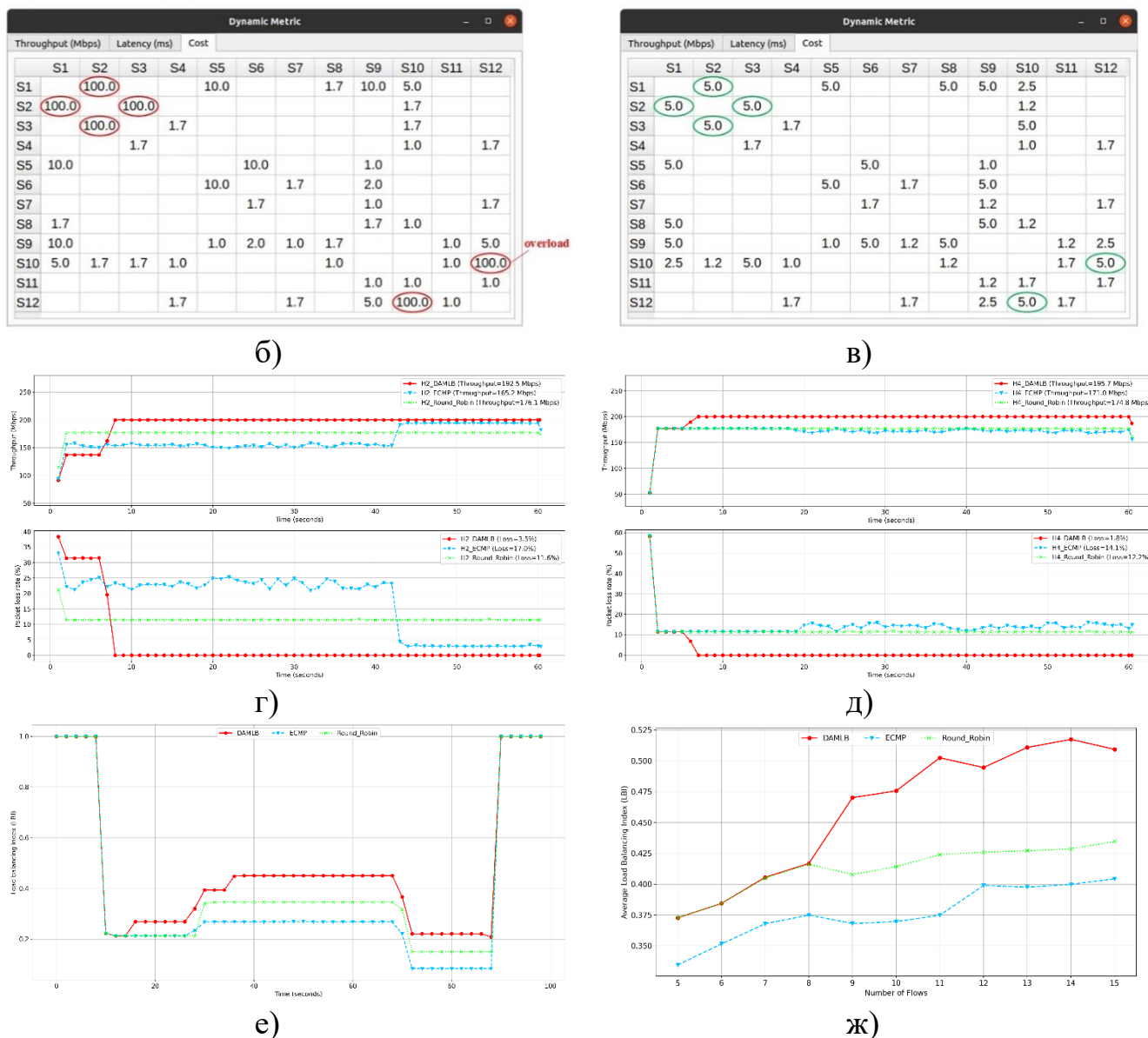
В ходе проведенных исследований установлено, что алгоритмы роевого интеллекта значительно превосходят метод случайного поиска по точности при оптимизации гиперпараметров, хотя метод случайного поиска (RS) имеет самое короткое и стабильное время выполнения, что делает его подходящим для быстрых тестов, но неэффективным для поиска оптимальных решений. Генетический алгоритм (GA) и алгоритм системы муравьиной колонии (ACS) являются быстрыми вариантами, при этом генетический алгоритм (GA) отличается высокой скоростью, но большой изменчивостью, в то время как алгоритм системы муравьиной колонии (ACS) сбалансирован между точностью и временем выполнения благодаря эффективному механизму обновления феромонов. Алгоритм стаи птиц (BFA) обеспечивает высокую стабильность и достигает хороших результатов при среднем времени выполнения, тогда как алгоритм искусственной пчелиной колонии (ABC) и алгоритм муравьиной системы (AS) требуют большего времени выполнения, но тщательно исследуют пространство решений, что приводит к высокой и стабильной точности. Алгоритм светлячков (FA) достигает наивысшей точности и стабильности между запусками, однако требует продолжительного времени выполнения и больших вычислительных ресурсов, что делает его подходящим для задач оптимизации, где требуется абсолютная точность.

Использование механизма Dropout в обучении модели улучшает ее способность к обобщению, предотвращает переобучение и способствует стабильному снижению функции потерь на обоих наборах данных, позволяя достигать точности 98% при вычислении маршрутов. При прогнозировании для разных маршрутов точность постепенно снижается от маршрута 1 к маршруту 4. Это может быть связано с накоплением ошибок в архитектуре сети RNN, а также с тем, что более поздние маршруты обычно длиннее и содержат больше узлов и ребер, что усложняет предсказание.

Для исследования алгоритма динамической адаптивной балансировки потоков данных в ПКС рассматривалась топология ПКС в соответствии с рисунком 9(а).



a)



**Рисунок 9** – Результаты исследования алгоритма динамической адаптивной балансировки потоков данных в ПКС: (а) – экспериментальная топология ПКС из 12 узлов, (б) – стоимость каналов до применения метода DAMLB, (в) – стоимость каналов после применения метода DAMLB, (г) – пропускная способность и процент потерь пакетов на H2, (д) – пропускная способность и процент потерь пакетов на H4, (е) – индекс балансировки нагрузки по времени, (ж) – среднее значение индекса балансировки нагрузки по количеству потоков данных. До применения алгоритма DAMLB наблюдалась перегрузка на некоторых каналах сети, что указывало на неравномерное распределение трафика. После включения механизма балансировки потоков данных, трафик распределяется более равномерно между различными маршрутами. Стоимость каналов значительно улучшилась, и перегрузка была устранена.

Алгоритм DAMLB превосходит методы ECMP и Round Robin, обеспечивая более высокую и стабильную пропускную способность, особенно при высокой нагрузке, а также снижая процент потерь пакетов почти до нуля после стабилизации. Более высокий индекс LBI подтверждает его эффективность в равномерном распределении трафика, что минимизирует перегрузки и улучшает общую производительность сети.

На элементы разработанных программных средств получены свидетельства о государственной регистрации в реестре Федеральной службы по интеллектуальной собственности.

## **ЗАКЛЮЧЕНИЕ**

В процессе проведенных научных исследований и разработок получены следующие результаты:

1 Разработаны математическая модель и метод интеллектуальной маршрутизации в ПКС на основе алгоритмов роевого интеллекта, таких как генетический алгоритм, оптимизация муравьиной колонии, алгоритм стаи птиц, алгоритм искусственной пчелиной колонии и алгоритм светлячков. Результаты экспериментов показали, что предложенные алгоритмы способны эффективно находить оптимальные маршруты, повышая адаптивность к изменениям в сети.

2 Предложена нейросетевая модель многопутевой маршрутизации в ПКС, основанная на рекуррентной нейронной сети. Применение алгоритмов роевого интеллекта для оптимизации гиперпараметров позволило модели достигать высокой точности до 98% при выборе оптимальных маршрутов в режиме реального времени.

3 Предложены алгоритмы роевого интеллекта для оптимизации гиперпараметров разработанной модели нейросетевой многопутевой маршрутизации в ПКС, которые позволяют автоматизировать подбор архитектуры и гиперпараметров модели. Результаты экспериментов показали, что роевые алгоритмы превосходят такие методы, как поиск по сетке, байесовская оптимизация и случайный поиск.

4 Разработаны модель и алгоритм динамической балансировки потоков данных в ПКС, обеспечивающие равномерное распределение нагрузки между каналами. Экспериментальные результаты показывают, что предложенный подход превосходит такие методы, как ECMP и Round Robin.

5 Разработана архитектура библиотеки программных компонентов интеллектуальной маршрутизации и балансировки потоков данных в ПКС, обеспечивающая эффективное управление потоками данных в сети на основе нейронных сетей и роевых алгоритмов.

6 Разработана структура визуальной программной системы SDNLoadBalancer для организации распределенной обработки потоков данных в ПКС. Система включает графический редактор для проектирования сетевых топологий, инструменты мониторинга и управления, а также средства тестирования производительности сети. Эта система позволяет эффективно управлять распределением нагрузки и оперативно подстраивать маршрутизацию под изменяющиеся условия сети.

## **ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ**

### **Статьи в изданиях из перечня ВАК**

1. Нгуен В. Т. Исследование процессов балансировки нагрузки в программно-конфигурируемых сетях на основе генетического алгоритма / Д. А. Перепелкин, В. Т. Нгуен // Вестник Рязанского государственного радиотехнического университета. – 2021. – № 77. – С. 43-57.



2. Нгуен В. Т. Исследование и анализ процессов многопутевой маршрутизации и балансировки потоков данных в программно-конфигурируемых сетях на основе генетического алгоритма / Д. А. Перепелкин, В. Т. Нгуен // Вестник Рязанского государственного радиотехнического университета. – 2022. – № 79. – С. 31-48.

3. Нгуен В. Т. Интеллектуальная многопутевая маршрутизация в программно-конфигурируемых сетях на основе алгоритма искусственной пчелиной колонии / Д. А. Перепелкин, В. Т. Нгуен // Информационные технологии. – 2022. – Т. 28. – № 8. – С. 395-404.

4. Нгуен В. Т. Интеллектуальная многопутевая маршрутизация в программно-конфигурируемых сетях на основе алгоритмов оптимизации муравьиной колонии / Д. А. Перепелкин, М. А. Иванчикова, В. Т. Нгуен // Информационные технологии. – 2022. – Т. 28. – № 10. – С. 520-528.

5. Нгуен В. Т. Интеллектуальная многопутевая маршрутизация в программно-конфигурируемых сетях на основе алгоритма миграции стаи птиц / Д. А. Перепелкин, М. А. Иванчикова, В. Т. Нгуен // Вестник Рязанского государственного радиотехнического университета. – 2022. – № 82. – С. 44-59.

6. Нгуен В. Т. Нейросетевая многопутевая маршрутизация в программно-конфигурируемых сетях на основе генетического алгоритма / Д. А. Перепелкин, М. А. Иванчикова, В. Т. Нгуен // Информационные технологии. – 2023. – Т. 29. – № 12. – С. 622-629.

7. Нгуен В. Т. Нейросетевая многопутевая маршрутизация в программно-конфигурируемых сетях на основе алгоритмов оптимизации муравьиной колонии / Д. А. Перепелкин, В. Т. Нгуен // Вестник Рязанского государственного радиотехнического университета. – 2024. – № 89. – С. 39-55.

#### **Статьи в изданиях, включенных в международные базы Web of Science и Scopus**

8. Tin Nguyen. “Research of Multipath Routing Processes in Software Defined Networks Based on Ant Colony Optimization” / D. A. Perepelkin, V. T. Nguyen // 11th Mediterranean Conference on Embedded Computing (MECO), 2022, DOI: 10.1109/MECO55406.2022.9797090.

9. Tin Nguyen. “Research of Multipath Routing and Load Balancing Processes in Software Defined Networks Based on Artificial Bee Colony Algorithm” / D. A. Perepelkin, V. T. Nguyen // ELEKTRO, 2022, DOI: 10.1109/ELEKTRO53996.2022.9803416.

10. Tin Nguyen. “Research of Multipath Routing Processes in Software Defined Networks Based on Firefly Algorithm” / D. A. Perepelkin, V. T. Nguyen // International Russian Automation Conference (RusAutoCon), 2022, DOI: 10.1109/RusAutoCon54946.2022.9896320.

11. Tin Nguyen. “Research of Multipath Routing and Load Balancing Processes in Software Defined Networks Based on Bird Migration Algorithm” / D. A. Perepelkin, M. A. Ivanchikova, V. T. Nguyen // International Russian Smart Industry Conference (SmartIndustryCon), 2023, DOI: 10.1109/SmartIndustryCon57312.2023.10110788.

12. Tin Nguyen. “Neural Network Multipath Routing in Software Defined Networks Based on Artificial Bee Colony Algorithm” / D. A. Perepelkin, M. A. Ivanchikova, V. T. Nguyen // XVIII International Symposium Problems of Redundancy in Information and Control Systems (REDUNDANCY), 2023, DOI: 10.1109/Redundancy59964.2023.10330174.

#### **Статьи в других изданиях и материалах конференций**

13. Нгуен В. Т. Интеллектуальная многопутевая маршрутизация в программно-

конфигурируемых сетях на основе модели поведения роя светлячков / Д. А. Перепелкин, В. Т. Нгуен // Цифровая обработка сигналов. – 2023. – № 4. – С. 32-40.

14. Нгуен В. Т. Задача многопутевой маршрутизации в программно-конфигурируемых сетях на основе алгоритма искусственной пчелиной колонии / Д. А. Перепелкин, В. Т. Нгуен // Современные технологии в науке и образовании СТНО. – 2022. – Т. 10. – С. 132-135.

15. Нгуен В. Т. Оптимизация гиперпараметров рекуррентной нейронной сети на основе генетического алгоритма для задачи многопутевой маршрутизации в программно-конфигурируемых сетях / Д. А. Перепелкин, В. Т. Нгуен // Современные технологии в науке и образовании СТНО. – 2023. – Т. 10. – С. 125-127.

### **Свидетельства о государственной регистрации программ для ЭВМ**

16. Нгуен В. Т. Программный компонент многопутевой маршрутизации в программно-конфигурируемых сетях на основе генетического алгоритма / Д. А. Перепелкин, В. Т. Нгуен / Свидетельство о государственной регистрации программы для ЭВМ № 2024690323. Зарегистрировано в РОСПАТЕНТ 13.12.2024, заявка № 2024689850.

17. Нгуен В. Т. Программный компонент многопутевой маршрутизации в программно-конфигурируемых сетях на основе алгоритмов оптимизации муравьиной колонии / Д. А. Перепелкин, В. Т. Нгуен / Свидетельство о государственной регистрации программы для ЭВМ № 2024691003. Зарегистрировано в РОСПАТЕНТ 18.12.2024, заявка № 2024690314.

18. Нгуен В. Т. Программный компонент многопутевой маршрутизации в программно-конфигурируемых сетях на основе алгоритмов роевого интеллекта / Д. А. Перепелкин, В. Т. Нгуен / Свидетельство о государственной регистрации программы для ЭВМ № 2024690740. Зарегистрировано в РОСПАТЕНТ 17.12.2024, заявка № 2024689860.

19. Нгуен В. Т. Программный компонент нейросетевой маршрутизации потоков данных в программно-конфигурируемых сетях на основе генетического алгоритма / Д. А. Перепелкин, В. Т. Нгуен / Свидетельство о государственной регистрации программы для ЭВМ № 2024690797. Зарегистрировано в РОСПАТЕНТ 17.12.2024, заявка № 2024689856.

20. Нгуен В. Т. Программный компонент нейросетевой маршрутизации потоков данных в программно-конфигурируемых сетях на основе алгоритмов оптимизации муравьиной колонии / Д. А. Перепелкин, В. Т. Нгуен / Свидетельство о государственной регистрации программы для ЭВМ № 2024691119. Зарегистрировано в РОСПАТЕНТ 19.12.2024, заявка № 2024689853.

21. Нгуен В. Т. Программный компонент нейросетевой маршрутизации потоков данных в программно-конфигурируемых сетях на основе алгоритмов роевого интеллекта / Д. А. Перепелкин, В. Т. Нгуен / Свидетельство о государственной регистрации программы для ЭВМ № 2024691678. Зарегистрировано в РОСПАТЕНТ 23.12.2024, заявка № 2024689848.

22. Нгуен В. Т. Программный компонент динамической балансировки потоков данных в программно-конфигурируемых сетях с обеспечением качества сетевого сервиса / Д. А. Перепелкин, В. Т. Нгуен / Свидетельство о государственной регистрации программы для ЭВМ № 2025610943. Зарегистрировано в РОСПАТЕНТ 16.01.2025, заявка № 2024689864.

**НГУЕН Ван Тин**

**МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПРОЦЕССОВ  
ИНТЕЛЛЕКТУАЛЬНОЙ МАРШРУТИЗАЦИИ И БАЛАНСИРОВКИ  
ПОТОКОВ ДАННЫХ В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ  
НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ И РОЕВЫХ АЛГОРИТМОВ**

**АВТОРЕФЕРАТ**

диссертации на соискание ученой степени  
кандидата технических наук

Подписано в печать 26.03.25. Формат бумаги 60×84 1/16

Бумага писчая. Печать трафаретная. Усл. печ. л. 1,5

Тираж 100 экз. Заказ 568.

Рязанский радиотехнический государственный радиоуниверситет им. В.Ф. Уткина  
390005, Рязань, ул. Гагарина, 59/1.  
Редакционно-издательский центр РГРТУ.