

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

На правах рукописи

Сунь Цюши

**Машинное обучение для оптимизации
распределения ресурсов в беспроводных системах
связи**

**Научная специальность 1.2.2. Математическое моделирование, численные
методы и комплексы программ**

ДИССЕРТАЦИЯ

на соискание ученой степени

кандидата технических наук

Перевод с английского языка

Научный руководитель:

Доктор физико-математических наук, профессор

Петросян Ованес Леонович

Санкт-Петербург

2024

Оглавление

Введение	5
Глава 1 Распределение ресурсов в однородной сотовой сети с помощью метаэвристических алгоритмов	16
1.1 Общие знания	16
1.1.1 Общие знания о сотовых сетях	16
1.1.2 Общие знания о распределении ресурсов	17
1.2 Системная модель сотовой сети	18
1.3 Метаэвристические алгоритмы	20
1.3.1 Алгоритмы роевого интеллекта	21
1.3.2 Алгоритмы дифференциальной эволюции	24
1.3.3 Алгоритмы случайного поиска	25
1.3.4 Алгоритмы эволюционной стратегии	27
1.4 Моделирование	28
1.4.1 Настройка среды	28
1.4.2 Численные результаты	29
1.5 Заключение к главе 1	34
Глава 2 Распределение ресурсов в однородной D2D сети с помощью глубокого обучения под наблюдением	35
2.1 Общие знания	35
2.1.1 Общие знания о сетях устройство устройство	35
2.1.2 Общие знания об основах машинного обучения	36
2.2 Системная модель сетей D2D	37
2.3 Архитектура алгоритма	38
2.3.1 Глубокие нейронные сети	39
2.3.2 Формирование массивов данных и обучение моделей	41
2.4 Моделирование	42
2.4.1 Настройка среды	42

2.4.2 Численные результаты	43
2.5 Заключение к главе 2	45
Глава 3 Распределение ресурсов в однородной D2D сети с помощью графового обучения под наблюдением	47
3.1 Общие сведения об изучении графиков	47
3.2 Графовое представление сетей D2D	50
3.3 Модели обучения графовых нейронных сетей	51
3.4 Модель 1: Нейронная сеть с передачей сообщений	52
3.5 Модель 2: Графовые сети с механизмом внимания	53
3.6 Моделирование	54
3.6.1 Настройка среды	55
3.6.2 Численные результаты	55
3.7 Заключение к главе 3	58
Глава 4 Распределение ресурсов в гетерогенной D2D сети с помощью графовой сети с механизмом внимания, улучшенной данными о рёбрах графа	60
4.1 Общие сведения о гетерогенных графах	60
4.2 Системная модель и графовое представление гетерогенных D2D сети	62
4.2.1 Модель системы	62
4.2.2 Графовое представление	63
4.3 Гетерогенные графовые сети с механизмом внимания	65
4.3.1 Процесс гетерогенной трансформации	66
4.3.2 Слой внимания на уровне узла	67
4.3.3 Слой внимания на уровне ребра	69
4.3.4 Функция потерь	70
4.4 Моделирование	72
4.4.1 Настройка среды	72
4.4.2 Сравнение производительности	73
4.4.3 Численные результаты	74
4.5 Заключение к главе 4	77
Глава 5 Обучение с подкреплением с использованием теории среднего поля для оптимального распределения ресурсов в гетерогенной сети D2D	79
5.1 Общие сведения о глубоком обучении с подкреплением	79

5.2 Системная модель гетерогенной сети D2D	82
5.3 Формулировка задачи распределения ресурсов	84
5.3.1 Частично наблюдаемые марковские процессы принятия решений	84
5.3.2 Состояние окружающей среды	84
5.3.3 Пространство наблюдений	85
5.3.4 Пространство действий	85
5.3.5 Функция вознаграждения	86
5.4 Многотипное среднеполевое многоагентное обучение с подкреплением	86
5.4.1 Многотипная формула среднего поля	87
5.4.2 Многотипная среднеполевая аппроксимация	87
5.4.3 Многотипное обновление среднего поля	88
5.4.4 Многотипное среднеполевое решение	88
5.5 Моделирование	90
5.5.1 Настройка среды	90
5.5.2 Сравнение производительности	91
5.6 Заключение к главе 5	95
Заключение и дальнейшие работы	97
5.7 Заключение	97
5.8 Будущие работы	99
5.8.1 Дальнейшее расширение подхода GNN	99
5.8.2 Дальнейшее расширение подхода RL	99
Литература	101

Введение

Актуальность темы диссертационного исследования

Беспроводная связь подразумевает передачу данных между минимум двумя устройствами без использования физических кабелей; вместо этого она основывается на установлении соединений с помощью радиоволн. Распространение компьютеров, мобильных телефонов и планшетов подчеркнуло жизненно важную роль систем мобильной связи, обеспечивающих связь между мобильными устройствами и передатчиками, такими как точки доступа (AP) и базовые станции (BS). За последние десятилетия системы мобильной связи прошли путь от первого поколения (1G) до четвертого поколения (4G) с продолжающимся развитием сетей связи пятого поколения (5G) и Beyond 5G (B5G). В отличие от сетей 4G, основной целью сетей 5G и B5G является увеличение пропускной способности в 1000 раз, что позволяет достичь скорости до 10 Гбит/с. Кроме того, эти передовые сети призваны минимизировать задержки до практически незаметных уровней и обеспечить повсеместное подключение, а также другие важные характеристики. По сравнению с существующими технологиями 4G, системы 5G отличаются повышенной спектральной эффективностью, что позволяет увеличить скорость передачи данных на определенной территории. Кроме того, системы 5G повышают надежность связи, поддерживая большее количество одновременных подключений устройств и потребляя при этом меньше энергии. Они также позволяют увеличить количество одновременных и мгновенных подключений устройств. Понятие "Интернет вещей"(IoT) относится к платформе, которая способствует совместному подключению широкого спектра устройств.

В многочисленных исследованиях признается, что в связи с экспоненциальным ростом количества сетевых устройств и генерируемых данных управление сетевыми ресурсами становится все более сложной задачей. Для решения этой проблемы был предложен ряд технологических мер. К ним относятся стратегии эффективного распределения ресурсов, совершенствование алгоритмов сжатия дан-

ных и систем кодирования каналов. В данной диссертации основное внимание уделяется оптимизации методов распределения ресурсов, при этом признается сложность и нецелесообразность решения всех подходов. В беспроводных системах связи ресурсы обычно включают в себя полосу пропускания, мощность, частоту и время. Пропускная способность системы беспроводной связи ограничивается как доступными ресурсами, так и методом распределения ресурсов, используемым передатчиком. Эти факторы в совокупности определяют объем информации, который может быть получен приемником. Различные методы распределения ресурсов обеспечивают различные характеристики системы. Основной задачей эффективного метода распределения ресурсов является разумное распределение ограниченных ресурсов между получателями, что позволяет максимально эффективно использовать эти ограниченные ресурсы для достижения оптимальной производительности системы. Следовательно, разработка грамотной стратегии распределения ресурсов имеет первостепенное значение.

Обзор предыдущих исследований

Оптимизация ресурсов в беспроводных сетях, таких как распределение мощности и формирование луча, часто формулируется в виде задач смешанного целочисленного нелинейного программирования (MINLP), которые трудно решить и обычно являются NP-трудными задачами. Для получения решений MINLP-задач были предложены различные методы оптимизации и обучения, некоторые из которых перечислены ниже:

Подходы, основанные на модели: Дробное программирование (FP) предлагает ценный подход, предлагая квадратичные преобразования, упрощающие оптимизацию на основе соотношений. Это достигается путем преобразования исходной невыпуклой задачи в последовательность выпуклых задач [2]. Другой подход, взвешенная минимальная среднеквадратичная ошибка (WMMSE), использует локальную информацию о канале для определения оптимальных точек, которые максимизируют как скорость, так и вес [3]. Тем не менее, большинство существующих алгоритмов разработаны в основном как частично оптимальные решения. Несмотря на то, что эти алгоритмы показывают отличные результаты в имитационных экспериментах, их практическая реализация в промышленных сценариях остается сложной. Их эффективность в значительной степени зависит от аналитических и эффективно решаемых математических моделей, которые может быть сложно построить в реальных условиях из-за специфического распреде-

ления пользователей и географического окружения.

Метаэвристические алгоритмы: Метаэвристические алгоритмы - это широко используемые методы стохастического поиска, предлагающие надежные решения для сложных задач. Они отлично работают в ситуациях, когда точные математические модели недоступны, поскольку подходят к проблемам оптимизации как к "черным ящикам"[7]. В области оптимального распределения мощности для систем связи были разработаны различные современные методы. Адаптивная оптимизация роя частиц (PSO) адаптирована для распределения ресурсов в сетевых беспроводных датчиках [8]. Кроме того, для решения проблем распределения ресурсов в сетях связи D2D был разработан усовершенствованный одноступенчатый алгоритм искусственной пчелиной колонии (ABC)[9]. В пользу этих алгоритмов говорит их относительно простая реализация, независимость от подробной информации о системе и адаптивность для получения практических решений в режиме реального времени. Однако стоит отметить, что их производительность может быть чувствительна к конкретным параметрам, что может потребовать корректировки для различных случаев использования.

Машинное обучение (ML): Машинное обучение получило значительное распространение благодаря экспоненциальному росту числа беспроводных устройств и генерируемых ими богатых массивов данных. Кроме того, доступность высокопроизводительных вычислительных инструментов, включая графические процессоры, ускорила обучение этих алгоритмов. В зависимости от процесса обучения машинное обучение можно разделить на глубокое обучение и обучение с подкреплением (RL). Глубокое обучение включает в себя автономные системы, которые используют данные для выявления закономерностей и прогнозирования [4]. Эту область можно разделить на обучение с учителем и без учителя. В обучении с учителем размеченные образцы данных используются для изучения отображений в пространствах входа-выхода классических алгоритмов оптимизации. Например, глубокие нейронные сети (DNN) используются для аппроксимации отображения вход-выход, присутствующего в обычных алгоритмах с минимальной взвешенной средней квадратичной ошибкой (WMMSE)[32]. С другой стороны, обучение без учителя использует нейронные сети для параметризации функции распределения ресурсов, выполняя оптимизацию, используя функцию потерь напрямую. Это избавляет от необходимости решать конкретные примеры задач и не полагается на предварительно помеченные образцы. Однако важно отметить, что методы

обучения без учителя могут потребовать больше времени на обучение для параметризации модели по сравнению с обучением с учителем [6]. Однако традиционные методы глубокого обучения не учитывают топологию сети, и для достижения удовлетворительной производительности модели требуется обучение на большом выборке данных.

Графовые нейронные сети (GNNs): GNN стали ценным инструментом для решения задач, связанных с неевклидовыми структурированными данными в сетях связи, предлагая эффективное использование знаний о предметной области и способность использовать пространственную информацию, скрытую в топологии сети. В [11], методы планирования каналов достигаются путем встраивания графов, а не полагаясь на совершенную информацию о состоянии канала (CSI), демонстрируя надежную производительность даже при ограниченных наборах данных. Для решения проблем распределения ресурсов была представлена нейронная сеть с передачей сообщений (Message Passing Graph Neural Network, MPGNN). В [12] алгоритм WMMSE улучшает сходимость за счет интеграции обучающегося модуля GNN. В ответ на присущую современным сетям гетерогенность исследователи разработали модель гетерогенной графовой нейронной сети (HetGNN) для решения задач распределения ресурсов в гетерогенных сетях. В [13] была разработана нейросеть Heterogeneous Interference Graph Neural Network (HIGNN) для обучения политикам управления мощностью и формирования луча. Каналы связи между передатчиками и приемниками рассматриваются как отдельные типы узлов, и HetGNN была использована для обучения политикам управления мощностью. Тем не менее, многие существующие глубокие модели не учитывают особенности узлов и ребер, что оставляет возможности для улучшения их производительности, особенно когда размер скрытого слоя сети превышает размер особенностей узлов и краев.

Обучение с подкреплением (RL): RL - это автономная вычислительная система, которая развивает способности к принятию решений через процесс итеративных экспериментов и последующей оценки результатов. Агент приобретает эти навыки принятия решений в процессе взаимодействия с окружающей средой [14]. И глубокое обучение, и обучение с подкреплением являются автономными системами обучения. Разница в том, что глубокое обучение изучает закономерности на основе обучающего набора и применяет полученные знания к новому набору. В отличие от этого, обучение с подкреплением учит принимать решения динамически,

основываясь на обратной связи от взаимодействия с окружающей средой. Следует отметить, что глубокое обучение и обучение с подкреплением могут работать вместе. В RL глубокие нейронные сети могут быть использованы для обучения отображениям Q-таблиц, что приводит к глубокому обучению с подкреплением (DRL)[53].

Резюмируя вышесказанное, можно сказать, что исследование влияния неоднородности, неполной информации и случайных факторов на формирование сети, в частности, устойчивых топологий сети, паттернов взаимодействия игроков в динамике, кооперативного поведения, а также равновесной структуры, является важным и значимым.

Цель исследования

Основной целью данной диссертации является создание набора алгоритмов оптимизации ресурсов, использующих искусственный интеллект и методы оптимизации для управления и контроля такими ресурсами, как мощность и формирование луча, с учетом множества ограничений в ситуациях принятия решений в реальном времени для наилучшего использования всех ресурсов в базовой сети. Основываясь на более реалистичных ограничениях, предложенный онлайн-алгоритм распределяет доступные ресурсы на пропускную способность и сквозную задержку на пути маршрутизации от узла-источника до узла назначения. Кроме того, преимущества нового предложенного алгоритма будут видны при создании требований реального времени для чувствительных к задержкам приложений 5G, а также при решении проблемы распределения ресурсов для крупномасштабных сетей, использовании меньшего количества ресурсов и снижении затрат. Кроме того, предложенный алгоритм может адаптироваться к различным требованиям QoS-сервисов, гарантируя высокий уровень QoS и обеспечивая высокий доступ для более приоритетных классов в сценариях перегрузки.

Основные задачи

Для достижения общих целей исследования были сформулированы следующие конкретные задачи, решение каждой из которых требует решения соответствующих задач:

1. Построение математической модели задачи распределения ресурсов в декартовой системе координат для различных сложных сценариев сетей связи (отовые сети, D2D-сети, крупномасштабные сверхплотные сети), которая соответствует принципам управления системой связи и используется для модели-

рования рабочих параметров и производительности системы в реалистичной среде.

2. Улучшить и оптимизировать существующие метаэвристические алгоритмы оптимизации на основе стохастического поиска для оптимизации проблемы распределения ресурсов в однородных сотовых сетях на основе среды "черного ящика".
3. Разработать традиционный алгоритм на основе глубокого обучения для оптимизации задачи распределения ресурсов в однородных D2D-сетях со структурированными данными.
4. Разработать алгоритмическую схему на основе глубокого обучения графов для оптимизации задачи распределения ресурсов для гетерогенных D2D-сетей с неструктурированными данными.
5. Разработать алгоритмический каркас на основе обучения с подкреплением для оптимизации задачи распределения ресурсов гетерогенных D2D-сетей с крупномасштабной сверхплотной природой.

Научная новизна

Научная новизна диссертации заключается в том, что проведенные исследования и проведённый анализ позволили найти новые решения проблем, связанных с распределением ресурсов и управлением помехами в беспроводных сетях. Научная новизна результатов диссертационного исследования может быть классифицирована следующим образом:

1. В результате сравнительных испытаний был установлен эвристический алгоритм, наиболее подходящий для решения задачи распределения ресурсов в сетях связи. Предложен новый метод распределения ресурсов, основанный на этом эвристическом алгоритме и методах глубокого обучения.

2. Беспроводная сеть связи формулируется как задача оптимизации графа. Предложен новый метод на основе графовой нейронной сети для обучения модели с учителем и без учителя.

3. Предложен новый GNN-алгоритм для граф-структурированных гетерогенных сетей связи. Алгоритм является новшеством в одновременном учете особенностей ребер в графах коммуникационных помех и неоднородных характеристик

элементов графа. Он уникальным образом использует как отличительные особенности рёбер, так и присущую им гетерогенность для расширения возможностей обучения графовых нейронных сетей.

4. Предложен многотипный мультиагентный алгоритм обучения с подкреплением, основанный на теории среднего поля, для крупномасштабных сверхплотных сетей. Этот алгоритм рассматривает пары приемопередатчиков как агентов, объединяя их в группы на основе типов их соединений. Для обучения агентов используется многотипное подкрепляющее обучение на основе теории среднего поля, направленное на выработку оптимальной политики. Примечательно, что это первое применение многотипового, многоинтеллектуального среднеполевого обучения с подкреплением для решения задач оптимизации в сетях связи.

Методология и техника исследования

Исследование проводилось в соответствии с традиционными методологиями, обычно используемыми в диссертациях. Эти методики включали в себя различные этапы, в том числе обзор литературы, постановку задач, выбор подходящих программных средств и методов решения проблем, разработку программного обеспечения, экспериментальное тестирование и оценку разработанных решений. Кроме того, исследование включало анализ результатов, полученных в ходе этих процессов.

В теоретическом разделе диссертации применялись различные методологии. Это анализ и создание архитектурных решений, использование методов проектирования информационных систем, программного обеспечения и взаимодействия, применение методов объектно-ориентированного и функционального программирования, использование инструментов и методик экспериментального тестирования программного обеспечения, реализация алгоритмического моделирования, применение методов программирования для беспроводных сетей. Практическая составляющая исследования была направлена на применение алгоритмических моделей и методов работы с коммуникационными системами.

Теоретическая и практическая значимость

В практическом применении изучение стратегий оптимизации и распределения ресурсов беспроводных сетей связи имеет большое значение для повышения качества и уровня обслуживания сети. В частности, это может помочь операторам связи лучше понять потребности пользователей и предоставить им более стабильные, безопасные и быстрые сетевые услуги. В то же время это может снизить

стоимость аппаратных ресурсов сети, повысить экономическую эффективность и конкурентоспособность оператора, чтобы лучше удовлетворить рыночный спрос.

Теоретически мы улучшаем существующие алгоритмы, а также разрабатываем новую структуру алгоритмов оптимизации для сценарно-специфических сложных систем связи. Одновременно мы исследуем природу новых разработанных алгоритмов. Сходимость алгоритмов демонстрируется с помощью математических доказательств и анализа сложности, а устойчивость алгоритмов проверяется с помощью теоретического анализа и экспериментов на различных наборах данных.

Апробация результатов

Основные результаты данной работы были представлены на международных конференциях “Устойчивость и процессы управления“ (Санкт-Петербург, 2021); “Международная конференция по роевому интеллекту“ (Шэньчжэнь, 2023).

Публикации

По результатам диссертации были опубликованы следующие работы: [7, 84, 85, 87]. Следующие работы [7, 87] опубликованы в рецензируемых журналах из списка ВАК. А [88] принят.

Благодарности

Автор выражает глубокую благодарность Ованесу Петросяну, доктор физико-математических наук, профессору Санкт-Петербургского государственного университета, за его поддержку и руководство в исследовательской работе над этой статьей. В процессе подготовки данной статьи авторы также признательны своим родителям за заботу, понимание и поощрение в различных ситуациях.

Основные научные результаты

Исходя из вышеизложенного, схема распределения ресурсов на основе ML является важнейшим инструментом для реализации услуг 5G и B5G. Хорошее обучение нейросетевой модели позволяет добиться приближенной к эвристическим алгоритмам производительности. В то же время она отвечает требованиям сетей связи к принятию решений в режиме реального времени и с малой задержкой. Таким образом, целью данной работы является оптимизация соответствующих схем распределения ресурсов в сотовых и D2D-сетях с помощью методов ML для увеличения пропускной способности системы и улучшения качества обслуживания. Основной вклад каждой главы кратко изложен ниже:

В главе 1 рассматриваются вопросы, связанные с управлением мощностью в

системах сотовых сетей. Кратко представлена теория теории распределения ресурсов и управления помехами. Сформулированная невыпуклая задача рассматривается как "черный ящик" а для поиска приближенных оптимальных решений используются эвристические алгоритмы. В этой главе впервые проводится всестороннее сравнение популярных на сегодняшний день эвристических алгоритмов, тестируется их производительность в задачах оптимизации сетевых ресурсов и определяется алгоритм с наилучшей производительностью. В алгоритмах машинного обучения, основанных на данных, обучающий набор определяет верхнюю границу модели, и эвристические алгоритмы, описанные в этой главе, могут обеспечить близкие к глобальному оптимальные решения для последующего обучения с учителем [7].

В главах 2 и 3 исследуются более реалистичные подходы для формирования луча и распределения мощности в сетях D2D. Мы комбинируем эвристические алгоритмы с глубоким обучением, предлагая алгоритмы распределения под наблюдением, основанные на DNN и GNN, соответственно. Состояния канала и близкие к оптимальным распределения, сгенерированные эвристическими алгоритмами, используются в качестве обучающего набора для глубокого обучения, и модели обучаются для изучения отображения признаков на переменные оптимизации. Также учитывается топология сети для изучения стратегии распределения ресурсов в рамках графового подхода к обучению [85,88].

В главе 4 исследуется сценарий гетерогенной сети связи D2D. Графовые нейросетевые модели обучаются с помощью обучения без учителя. Мы предлагаем алгоритмы распределения, основанные на GAT и EGAT соответственно. По сравнению с GNN, учитывающей данные о спектре, в предыдущей главе, исследование в этой главе сосредоточено на более обобщенном GAT, учитывающем пространственные данные. Кроме того, для повышения эффективности алгоритмов вводятся краевые особенности, чтобы усилить процесс обучения. Цель состоит в том, чтобы максимизировать суммарную скорость системы путем совместной оптимизации конструкции формирования луча и распределения мощности [87].

В главе 5 исследуется сценарий гипермасштабной плотной гетерогенной коммуникационной сети - типичной гетерогенной мультиагентной системы. В этой главе предлагается алгоритм распределения ресурсов, основанный на многоинтеллектуальном обучении с усилением (MARL) и игре типа среднего поля (MFTG). Учитывая взаимодействие между каждым агентом и различными типами средних

полей, преодолевается проблема масштабируемости при вычислении индивидуального вознаграждения, что приводит к эффективному распределению ресурсов.

В заключении главы подводятся итоги работы и предлагается несколько потенциальных тем для будущих исследований.

Положения, выносимые на защиту

1. Описаны и разработаны новые математические модели сложных сетей связи (гетерогенных краевых информационных D2D-сетей, крупномасштабных сверхплотных гетерогенных D2D-сетей). Задача распределения ресурсов в коммуникационных системах строится как модель графового представления сети и парной мультиагентной системы для обучения стратегиям оптимизации с использованием нейросетевых методов. С помощью компьютерного программирования построена новая динамическая интерактивная среда моделирования системы связи.
2. Усовершенствованы существующие эвристические алгоритмы, предназначенные для решения проблем распределения ресурсов. Усовершенствованные алгоритмы не зависят от математической модели системы и могут находить близкие к оптимальным решения путем случайного поиска в сценариях "черного ящика". Их производительность превосходит производительность методов оптимизации, основанных на математических моделях, особенно в крупномасштабных сетях. Для подтверждения высокой производительности и стабильности усовершенствованного алгоритма были проведены обширные имитационные эксперименты.
3. Разработан новый алгоритм решения задачи распределения ресурсов в однородных сетях связи, основанный на метаэвристических алгоритмах и методах глубокого обучения. Оптимизационная политика, получаемая предложенным алгоритмом путем обучения, может удовлетворить требования динамического распределения ресурсов и в то же время достичь производительности приближенного эвристического алгоритма. Превосходство и масштабируемость алгоритма проверены с помощью имитационных экспериментов.
4. Разработан новый алгоритм решения проблемы распределения ресурсов гетерогенной сети на основе свойств рёбер графа и методов обучения графов.

Предложенный алгоритм повышает обучаемость графовых нейронных сетей за счет данных о рёбрах графа, а его производительность превышает производительность традиционных графовых нейросетевых алгоритмов. Превосходство и масштабируемость предложенного алгоритма подтверждены имитационными экспериментами.

5. Разработаны новые многотипные алгоритмы обучения с подкреплением, предназначенные для получения стратегий оптимизации в обширных гетерогенных мультиагентных коммуникационных системах. Различные пары приемопередатчиков концептуализируются как отдельные наборы в среднем поле, а стратегии взаимодействия между различными наборами исследуются с помощью теории среднего поля. С помощью математических доказательств показано, что стратегии оптимизации на основе многотипных средних полей в гетерогенных системах агентов работают лучше, чем стандартные методы обучения с подкреплением на основе средних полей. Эффективность и устойчивость предложенного алгоритма дополнительно подтверждается с помощью имитационных экспериментов.

Глава 1

Распределение ресурсов в однородной сотовой сети с помощью метаэвристических алгоритмов

В этой главе рассматривается проблема распределения мощности в сотовых сетях. Вносятся улучшения в существующие метаэвристические алгоритмы, а также проводится сравнение ряда современных стохастических алгоритмов с эталонными алгоритмами. Результаты моделирования демонстрируют превосходство предложенных алгоритмов над традиционными эталонными алгоритмами.

1.1 Общие знания

1.1.1 Общие знания о сотовых сетях

Сотовая сеть - это сеть мобильной связи, получившая свое название от того, как устроена зона покрытия базовых станций, похожая на соты. В центре сети находится базовая станция, которая делит зону связи на несколько небольших шестиугольных или сотовых участков, каждый из которых обслуживается одной базовой станцией. Сотовые сети являются основой систем мобильной связи [16]. Сотовые сети состоят из следующих основных компонентов: Базовая станция (BS): BS - это основной компонент сотовой сети, который используется для передачи и приема беспроводных сигналов. Каждая BS охватывает небольшую территорию, называемую сотой. Сота: Сота - это зона покрытия сотовой сети, обычно имеющая форму шестиугольника или подобную. Каждая ячейка обслуживается одной BS, что позволяет эффективно распределять коммуникационные ресурсы. Мобильные устройства: Мобильные устройства, такие как сотовые телефоны,

планшеты и смарт-часы, подключаются к сотовой сети для связи. Они обычно оснащены встроенными антеннами для связи с базовой станцией [1].

1.1.2 Общие знания о распределении ресурсов

В стремительно развивающейся сфере беспроводных технологий мобильные приложения и услуги получили широкое распространение в различных сферах жизни. Они охватывают широкий спектр приложений, начиная от потоковой передачи видео в формате 4К и заканчивая работой беспилотников и сервисами определения местоположения внутри помещений. Важно понимать, что эти разнообразные приложения часто предъявляют особые требования к базовой сетевой инфраструктуре. Например, сфера автономного управления транспортными средствами требует сверхнизкой задержки связи для обеспечения безопасного и оперативного движения автомобилей. Следовательно, распределение ресурсов играет ключевую роль в удовлетворении специфических требований этих приложений.

В практических системах связи концепция распределения ресурсов включает в себя широкий и всеохватывающий набор стратегий. Эти стратегии включают в себя управление доступом к каналу, распределение энергетических ресурсов, распределение доступной полосы пропускания, объединение пользователей и устройств, политику управления энергопотреблением и разработку оптимальных конфигураций формирования луча. Такой многогранный подход к распределению ресурсов необходим для обеспечения эффективной работы систем связи в соответствии с разнообразными требованиями, выдвигаемыми современными приложениями.

Распределение ресурсов - сложный и критически важный вопрос в сотовых сетях, требующий высокой степени оптимизации и динамического управления. При правильном распределении ресурсов сотовые сети могут эффективно предоставлять услуги связи, увеличивать пропускную способность сети и удовлетворять растущие потребности пользователей. Исследования в этой области продолжают стимулировать развитие коммуникационных технологий и повышать производительность сетей.

1.2 Системная модель сотовой сети

Мы рассматриваем классический сценарий нисходящей многосотовой связи - массивную сеть MISO с M -антенными BS и одноантенными пользовательскими устройствами (UE). Проблема распределения мощности в сотовой сети заключается в задании интерферирующих каналов множественного доступа (IMAC)[17]. Все BS в зоне покрытия сети одновременно обслуживают все UE. Однако, поскольку разные соты используют одну и ту же частоту, UE все равно подвержены межсотовой и внутрисотовой интерференции. Обозначим BS как $\mathcal{N} = \{1, \dots, N\}$, а UE как $\mathcal{K} = \{1, \dots, K\}$. Обозначим D_{nk} как множество соседних UE k -го UE в n -й ячейке, обозначим C_n как множество соседних ячеек n -й ячейки. Предположим, что $n \in \mathcal{N}$, $k \in \mathcal{K}$, $k' \in D_{nk}$, $n' \in C_n$, тогда принимаемый сигнал k -го UE от n -й BS в n -й ячейке может быть сформулирован следующим образом

$$\begin{aligned}
 y_{nk} = & \underbrace{g_{n,nk}^H w_{nk} \sqrt{p_{nk}} s_{nk}}_{\text{desired signal}} + \underbrace{\sum_{k' \neq k} g_{n,nk}^H w_{nk'} \sqrt{p_{nk'}} s_{nk'}}_{\text{intra-cell interference}} \\
 & + \underbrace{\sum_{n' \neq n} \sum_{k'} g_{n',nk}^H w_{n'k'} \sqrt{p_{n'k'}} s_{n'k'}}_{\text{inter-cell interference}} + z_{nk},
 \end{aligned} \tag{1.1}$$

где $g_{n,nk}$ обозначает отклик канала от n -й BS к k -му UE в n -й ячейке, p_{nk} - соответствующую мощность передачи. $s_{nk} \sim \mathcal{U}(0, 1)$ - передаваемый сигнал. $z_{nk} \sim \mathcal{N}(0, \sigma^2)$ - аддитивный белый гауссовский шум (AWGN). Вектор согласованного формирования луча (CB) от n -й BS к k -му UE обозначается как w_{nk} .

В литературе, посвященной координированным многоточечным (CoMP) методам формирования луча, рассматриваются различные схемы. В данном исследовании для упрощения задачи мы решили использовать схему формирования луча с нулевым усилением. Тогда отношение сигнал/помеха плюс шум (SINR) для k -го UE может быть рассчитано как

$$\gamma_{nk} = \frac{g_{nk,nk} p_{nk}}{\sum_{k' \neq k} g_{nk,nk'} p_{nk'} + \sum_{n' \neq n} \sum_k g_{nk,n'k} p_{n'k} + \sigma^2}, \tag{1.2}$$

где $g_{nk,nk} = |g_{n,nk}^H w_{nk}|^2$ обозначают независимые канальные коэффициенты усиления желаемого сигнала. $g_{nk,nk'} = |g_{n,nk}^H w_{nk'}|^2$ обозначим канальный коэффициент усиления внутрисотовой помехи от соседних UE в n -й соте. $g_{nk,n'k} = |g_{n',nk}^H w_{n'k}|^2$

обозначают канальный коэффициент усиления межсотовой помехи от соседних UE в соседних ячейках n -й соты.

Скорость нисходящей линии связи nk может быть выражена в терминах нормированной полосы пропускания как

$$C_{nk} = \log_2 (1 + \gamma_{nk}). \quad (1.3)$$

Основной целью данного исследования является определение идеального уровня мощности, оптимизирующего общую суммарную скорость сети, при соблюдении ограничения на максимальную мощность каждого передатчика. Поставленная задача может быть выражена в виде

$$\begin{aligned} \max_{p_{nk}, w_{nk}} \quad & \sum_{n=1}^N \sum_{k=1}^K \log_2 (1 + \gamma_{nk}) \\ \text{s.t.} \quad & 0 \leq p_{nk} \leq p_{\max}, \forall n \in \mathcal{N}, k \in \mathcal{K}. \end{aligned} \quad (1.4)$$

Целевая функция представляет собой сложную, в виде невыпуклой нелинейной оптимизационной, задачу, которая усложняется наличием ограничений. В результате поиск глобального оптимального решения становится затруднительным. Эвристические алгоритмы способны приближенно находить глобально оптимальные решения, но при этом затрачиваются значительные вычислительные ресурсы. Для того чтобы удовлетворить спрос на приложения реального времени, требующие решений невысокой сложности, мы предлагаем использовать методы глубокого обучения для параметризации таких решений. Несмотря на то что достижение теоретического оптимума для обучаемых решений может представлять определенные трудности, факты практического применения постоянно показывают, что методы глубокого обучения часто дают удовлетворительные результаты.

Для модельных подходов, как правило, сложно оценить отставание производительности от оптимального решения, а практическая реализация ограничена из-за несовершенства математической модели. Кроме того, из-за несовершенства математической модели в реальных сценариях связи трудно адаптировать метод, основанный на модели, к гетерогенным сотовым сетям. Поэтому в следующем разделе рассматриваются метаэвристические алгоритмы, не требующие моделирования.

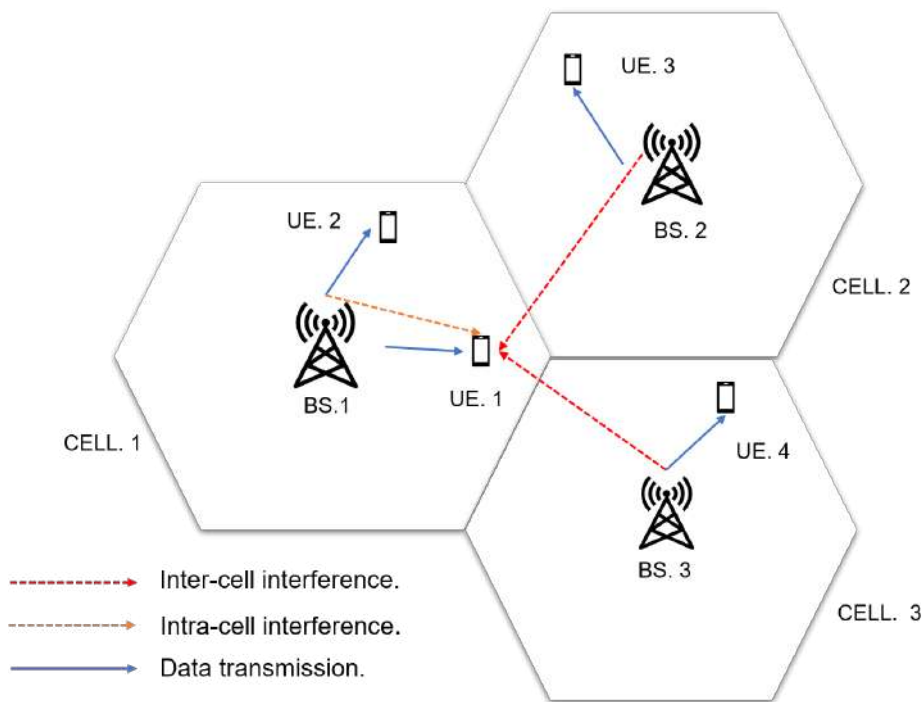


Рис. 1.1: Пример беспроводной сотовой сети

1.3 Метаэвристические алгоритмы

Для поиска эффективного подхода к решению этой оптимизационной задачи мы рассматриваем различные конкурсы по глобальной оптимизации, позволяющие выбрать лучшие алгоритмы из возможных методик. Кроме того, в процессе соревнования рождаются новые идеи, которые могут быть развиты в практические решения. Мы рассматриваем специальную сессию и конкурс по крупномасштабной глобальной оптимизации за последние десять лет и концентрируемся на различных математических методах оптимизации с одной целью. Мы выбираем девять метаэвристических алгоритмов из числа победивших, таких как искусственная пчелиная колония (ABC), самоадаптивная дифференциальная эволюция (jDE и iDE), генерационная оптимизация роя частиц (GPSO), расширенная оптимизация муравьиной колонии (EACO), дифференциальная эволюция (DE), оптимизация роя частиц (PSO), имитация отжига (SA), monotonic basin hopping (MBH), ковариационная матрица стратегии эволюции адаптации (CMA-ES). Мы рассматриваем девять метаэвристических алгоритмов из четырех основных типов. Краткое описание этих алгоритмов приведено ниже:

1.3.1 Алгоритмы роевого интеллекта

Искусственная колония пчел

Алгоритм искусственной пчелиной колонии (АВС) - это стохастический метод поиска, основанный на интеллектуальном кормовом поведении пчелиного роя. В этом алгоритме для каждого решения-кандидата указывается местоположение источника пищи в пространстве поиска, а качество источника пищи используется в качестве оценки пригодности.

Модель включает в себя три основных элемента: рабочие пчелы, наблюдатели и источники пищи. Количество пчел-работниц равно количеству источников пищи. Рабочие пчелы вылетают из улья для поиска источника пищи и сбора информации о качестве других источников пищи в окрестностях обнаруженного места. Вернувшись в улей, они передают информацию о найденном источнике пищи наблюдателям. Наблюдатели оценивают новое местоположение по информации, полученной от занятых пчел, в соответствии с вероятностью выбора качества и отдают предпочтение источнику пищи с высоким значением пригодности. Наблюдатель становится занятой пчелой, когда выбирает новый источник пищи для исследования. При отказе от исследованного источника пчела переключается на пчелу-разведчика и в случайном порядке ищет новые пищевые ресурсы в пространстве поиска. Этот процесс повторяется до тех пор, пока не будет найден оптимальный источник пищи.

Достоинства АВС: не требует большого количества параметров, устойчив, быстро сходится, обладает высокой гибкостью. Недостатки АПК: он может преждевременно сходиться на этапе поиска, а точность классификации полученного наилучшего значения может не удовлетворять требованиям.

Оптимизация с помощью роя частиц

Метод роя частиц (МРЧ) - это метод роевого интеллекта. Первоначальная идея МРЧ была навеяна поведением популяций птиц и рыб. МРЧ и эволюционные стратегические методы имеют много стандартных черт [18]. Этот алгоритм моделирует поведение информационного взаимодействия и сотрудничества членов группы. Отличие от генетического алгоритма состоит в том, что МРЧ не требует операторов эволюции, таких как кроссинговер и мутация. В модели существует популяция решений-кандидатов, называемых частицами. Эти частицы перемеща-

ются в пространстве поиска по своему положению и скорости. Движение каждой частицы направлено к наилучшим позициям в пространстве поиска, которые обновляются по мере того, как другие частицы находят лучшие позиции. Предполагается, что таким образом рой будет двигаться к наилучшим решениям. Процесс МРЧ можно сформулировать следующим образом:

$$v_{id}^{t+1} = v_{id}^t + c_1 * rand(0, 1) * (p_{id}^t - x_{id}^t) + c_2 * rand(0, 1) * (p_{gd}^t - x_{id}^t), \quad (1.5)$$

and

$$x_{id}^{t+1} = x_{id}^t + v_{id}^t, \quad (1.6)$$

где x_{id}^t и v_{id}^t - положение и скорость каждой частицы, параметр d - размер популяции, i - индекс каждой частицы, t - число итераций. c_1 и c_2 - коэффициенты обучения. p_i - значение, исследованное i -й частицей, p_g - значение, исследованное соседями i -й частицы. МРЧ можно реализовать в виде алгоритма 1:

Algorithm 1 МРЧ

Require: Сгенерировать начальную выборку

Ensure: Лучший вектор

while Условие останова не достигнуто **do**

for Каждой частицы x с положением p_i **do**

 Вычислить значение приспособленности

if значение приспособленности больше чем текущее лучшее значение p_{best} **then**

 Сохранить лучшее значение в p_{best}

end if

end for

 Выбрать частицу с наилучшей приспособленностью и сохранить в g_{best}

for Каждой частицы **do**

 Вычислить скорость частицы

 Обновить положение частицы

end for

end while

Достоинства МРЧ: он имеет простой расчет без перекрытия и мутации. Недостатки МРЧ: он может попасть в локальный оптимум в высокоразмерном пространстве и имеет низкую скорость сходимости на итерации.

Генеративная оптимизация роя частиц

Генеративный метод роя частиц (ГМРЧ) - это вариант стандартного алгоритма МРЧ [19]. В алгоритме МРЧ скорость является одним из наиболее значимых параметров; если скорость частиц в рое обновляется эффективно, то усилия по поиску не будут тратиться впустую, ища в неправильных направлениях. Процедура МРЧ заключается в перемещении частиц для поиска позиций оптимальных решений. Скорость, с которой частицы меняют положение, обычно регулируется путем умножения скорости на коэффициент. В отличие от стандартного алгоритма, сначала вычисляется скорость для всех частиц, а затем обновляется положение.

ГМРЧ может решать задачи стохастической оптимизации по итерационной схеме со случайными семенами. Однако он не подходит для решения многоцелевых задач.

Расширенная оптимизация с помощью муравьиного алгоритма

Муравьиный алгоритм (МА) - это классическая биоинспирированная технология, основанная на кормовом поведении муравьев. Алгоритм МА моделирует процесс поиска колонией муравьев кратчайшего пути от гнезда к источнику пищи. В модели группа имитационных агентов имитирует кормовое поведение естественных муравьев в поисках минимального значения функции. Каждый агент отправляется из гнезда в поисках источника пищи и в конце испытания прибывает в гнездо. На пути поиска источника пищи каждый агент оставляет маркер, который называется феромоном. Концентрация феромонов на каждом пути используется для оценки дальности пути и качества источника пищи. Информация, содержащаяся в феромоне на пути, играет важную роль при выборе агентом этого пути. Чем выше значение пригодности при оценке пути, тем выше вероятность того, что путь будет принят. Расширенный МА улучшает оригинальный алгоритм за счет использования многоядерного гауссова распределения, основанного на трех параметрах, которые вычисляются в зависимости от качества каждого предыдущего решения. Ранжирование значений целевой функции осуществляется с помощью метода оракульного штрафа (oracle penalty method).

Достоинства РМА: параллельный процесс может искать решения независимо и одновременно. Недостатки РМА: распределение вероятностей на итерациях меняется, а время сходимости нестабильно.

1.3.2 Алгоритмы дифференциальной эволюции

Дифференциальная эволюция

Алгоритм дифференциальной эволюции (ДЭ) является одним из наиболее популярных методов решения задач непрерывной оптимизации. Алгоритм ДЭ основан на стратегии эволюции, но не вдохновлен естественной парадигмой, как остальные алгоритмы [21]. Она предлагается для поиска минимального значения недифференцируемых и нелинейных непрерывных функций. Классический алгоритм ДЭ имеет две существенные особенности, подлежащие настройке: стратегию обучения и параметры управления. Стратегия обучения включает в себя основные типы операторов генетических алгоритмов, такие как мутация, кроссинговер и селекция. Базовый вариант алгоритма ДЭ работает на основе наличия популяции решений-кандидатов. Агенты перемещаются в пространстве поиска, совмещая позиции существующих агентов из популяции. Если значение новой позиции агента улучшается, то он принимается и становится частью популяции. При этом ожидается, но не гарантируется, что в конечном итоге будет найдено глобальное оптимальное решение.

При мутации мутантный вектор формируется следующим образом:

$$v_{i,G+1} = x_{r,G} + F(x_{r,G} - x_{r,G}) \quad (1.7)$$

где F - коэффициент пропорциональности, G - количество итераций. x_{r1} , x_{r2} и x_{r3} - случайные искомые векторы в текущей итерации. При кроссинговере пробный вектор образуется путем объединения родительского вектора с мутировавшим вектором.

$$u_{i,G+1} = \begin{cases} v_{i,G+1} & \text{if } rand_j \leq Cr \\ x_{r,G} & \text{if } rand_j > Cr \end{cases} \quad (1.8)$$

где Cr - коэффициент кроссинговера. j - случайное число в результирующем массиве. v_i - текущее наилучшее значение, x_i - наилучшее искомое значение. ДЭ может быть реализован в виде алгоритма 2:

Достоинства алгоритма ДЭ: он позволяет решать оптимизационные задачи с высокой вычислительной сложностью. Недостатки ДЭ: требуется настройка параметров, сходимость неустойчива.

Algorithm 2 Дифференциальная эволюция

Require: Сгенерировать случайную выборку**Ensure:** Лучший вектор

```

while Условие останова не достигнуто do
  for Каждого решения  $x_i$  в популяции do
    Сгенерировать новое решение  $s_i$ ;
    if приспособленность( $(s_i)$ )  $\geq$  приспособленность( $(x_i)$ ) then
      Оставить  $s_i$  в популяции;
    else
      Оставить  $x_i$  в популяции;
    end if
  end for
  Оценить приспособленность новой популяции
  Обновить значение лучшего решения
end while
Вернуть лучшее решение

```

Варианты дифференциальной эволюции

Это два различных варианта алгоритма ДЭ, основанных на механизме самоадаптации. Стратегии обучения и параметры управления, используемые в стандартном алгоритме ДЭ, в значительной степени зависят от конкретной задачи оптимизации. Выбор стратегии и настройка параметров могут потребовать значительных временных затрат. Было сделано много различных предложений по самоадаптации параметров CR и F оригинального алгоритма дифференциальной эволюции. Доказано, что существует множество различных предложений по адаптации параметров CR и F . Первый вариант (jDE) не использует операторы ДЭ для обновления параметров F и CR , процедура больше похожа на управление параметрами, чем на самоадаптацию [22]. Во втором варианте (iDE) для обновления параметров CR и F для каждой особи используется разновидность выбранного оператора ДЭ [23].

1.3.3 Алгоритмы случайного поиска**Имитация отжига**

Имитация отжига - это стохастический метод глобальной поисковой оптимизации для поиска. Этот алгоритм имитирует процедуру статистического отжига растущих кристаллов для достижения глобальной оптимальной конфигурации внут-

ренной энергии [24]. Процесс отжига происходит путем первоначального возбуждения атомов в материале при высокой температуре. Этот шаг нагревает атомы и ускоряет их движение. На следующем этапе происходит медленное охлаждение для снижения возбудимости, в результате чего атомы переходят в более стабильную конфигурацию. Важнейшим компонентом реализации процесса имитационного отжига является инициализация случайного решения в окрестности текущего оптимального решения и оценка объективных функций. Если значение функции стоимости меньше ее текущего оптимального значения, решение принимается, и принимается новое оптимальное значение. Если значение функции стоимости больше, чем текущее оптимальное значение, то точка принимается или отклоняется с некоторой вероятностью. Для расчета вероятности вводится параметр температуры. В процессе охлаждения температура снижается, при этом вероятность принятия сходится к нулю. И весь процесс отжига завершается после большого числа испытаний. Такая стратегия позволяет избежать попадания в ловушку локального оптимального решения.

Преимущества метода имитации отжига: он позволяет решать задачи с произвольными системами и функциями стоимости. Недостатки метода имитации отжига: требует настройки параметров и может попасть в локальные минимумы.

Monotonic Basin Hopping

Monotonic Basin Hopping (МВН) - стохастический метод глобальной оптимизации. Этот алгоритм представляет собой двухфазный подход, сочетающий глобальный шаговый алгоритм с процедурой локальной минимизации на каждой итерации [25]. Модель алгоритма использует случайные возмущения для перескока бассейнов и алгоритм локального поиска для оптимизации каждого бассейна. Итерации выполняются следующим образом: На первом этапе с помощью случайного возмущения осуществляется переход между бассейнами координат. На втором этапе используется процедура локальной оптимизации для оценки новых координат и принятия решения о принятии или отклонении координат на основе минимизированного значения функции. Первоначальная идея этого алгоритма заключается в том, чтобы отобразить целевую функцию на поиск локального минимума из начальной точки. Этот механизм позволяет существенно повысить эффективность решения задач. Основная идея МВН заключается в отображении целевой функции $f(x_0)$ на локальный минимум, найденный начиная с x_0 , МВН

может быть реализован в виде алгоритма 3:

Algorithm 3 Monotonic basin hopping

Require: $x_0 \leftarrow$ Выбрать начальное решение

Ensure: оптимальное значение $x, f(x)$

$x_0 \leftarrow$ выбрать начальное решение $g_0 = 0$ and $d_0 = 0$

$x_0 \leftarrow$ минимизировать (f, x_0)

repeat

$y \leftarrow$ случайное возмущение(x)

$y \leftarrow$ минимизировать (f, y)

$x \leftarrow$ принять (x, z)

until Достигнуто условие останова

В данной главе мы объединяем оригинальное обобщение этой концепции, в результате чего получаем метаалгоритм, оперирующий с любой популяцией с помощью подходящего алгоритма. Реальный метод восстанавливается при использовании популяции, содержащей одну особь, в сочетании с локальным оптимизатором.

1.3.4 Алгоритмы эволюционной стратегии

Ковариационная матрица стратегии эволюции адаптации (CMA-ES) - это стохастический метод решения инволютивных нелинейных невыпуклых непрерывных задач оптимизации "черного ящика"[26]. В его основе лежит идея самоадаптации в эволюционных стратегиях. Механизм этого алгоритма заключается в построении параметрического распределения на пространстве поиска, в котором заранее определены функции признаков. Из этого параметрического распределения выбирается популяция кандидатов на решение. Затем эти кандидаты оцениваются функцией "черного ящика". Учитывая значения функций в выборочных точках, обновление и хранение ковариационной матрицы является доминирующей временной и пространственной сложностью на каждой итерации алгоритма. Ковариационная матрица, в которой доминирует временная и пространственная сложность, обновляется и хранится на каждой итерации алгоритма.

Преимущества CMA-ES: подходит для решения небольших неразделимых оптимизационных задач. Недостатки CMA-ES: высокая сложность и преждевременная стагнация.

1.4 Моделирование

1.4.1 Настройка среды

К развертыванию алгоритма предъявляются высокие требования по низкой вычислительной сложности, что и рассматривается здесь. Конфигурация платформы моделирования выражается в следующем: ЦП Intel i7 10750H и ОЗУ 16 ГБ. Для сравнения производительности этих метаэвристических алгоритмов и поиска наилучшего алгоритма для решения задачи распределения мощности проводится серия имитационных экспериментов.

Таблица 1.1: Параметры моделирования сотовой сети.

Условные обозначения	Параметр моделирования	Значение
N	Количество BS	4, 9, 16
M	Среднее количество пользователей на соту	2, 4, 8
K	Общее количество пользователей	MN
f_d	Доплеровская частота	10 HZ
P_{\min}	Минимальная выделенная мощность	5 dBm
P_{\max}	Максимальная выделенная мощность	38 dBm
R_{\min}	Расстояние до внутреннего пространства	0.01 km
R_{\max}	Половина расстояния между ячейками	1 km
T	Период времени	20 ms
σ^2	Спектральная плотность мощности шума	- 114 dBm/Hz
z	Замирания сигнала	8 dB

Рассматриваются беспроводные сотовые сети различного масштаба с количеством ячеек 2×2 , 3×3 и 4×4 . В каждой ячейке пользователи распределены случайным и равномерным образом в диапазоне $r \in [R_{\min}, R_{\max}]$. Замирания на малых масштабах соответствуют распределению Рэлея, и принята модель Джейкса с f_d . Крупномасштабные замирания определяются по формуле $\beta = -120,9 - 37,6 \log_{10}(d) + 10 \log_{10}(z)$ дБ в соответствии со стандартом Long-Term Evolution (LTE), где z - величина замирания сигнала, а d - расстояние от передатчика до приемника (км). В таблице 2.1 собраны основные параметры сети. По результатам моделирования максимальное число итераций определено как 1000. В общем случае метаэвристические алгоритмы используют методы случайного поиска, при которых эффективность оптимизации сильно зависит от начального значения и тонкой настройки параметров. Следовательно, воспроизводимость результатов оптими-

зации, полученных в одинаковых условиях, не может быть гарантирована. Поэтому для сравнения эффективности предложенных алгоритмов было проведено 20 испытаний и выполнен статистический анализ результатов. Эффективность сравниваемых методов оценивается путем усреднения результатов 20 испытаний.

1.4.2 Численные результаты

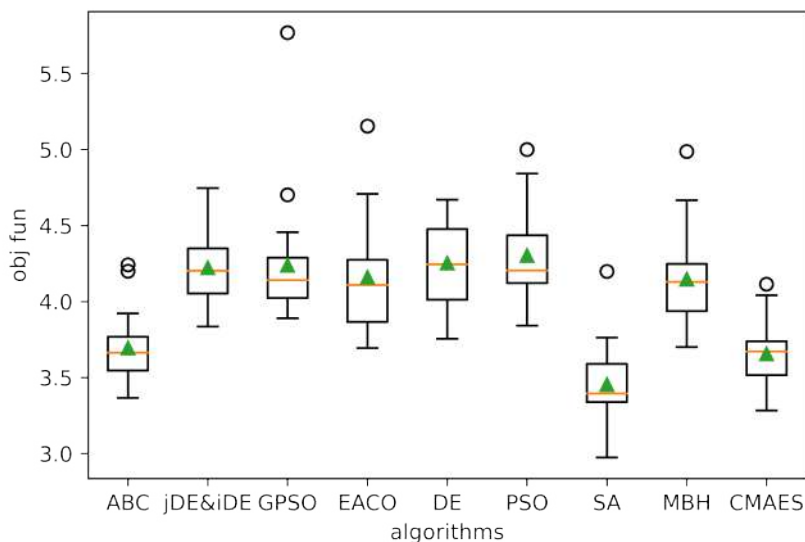


Рис. 1.2: Средние значения приспособленности (N=4).

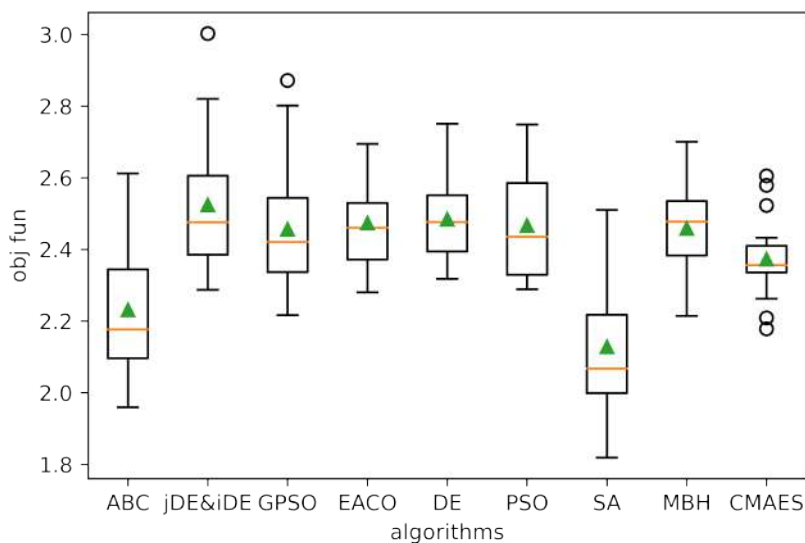


Рис. 1.3: Средние значения приспособленности (N=9).

В этом разделе мы приводим результаты моделирования, свидетельствующие о производительности метаэвристических алгоритмов. Для настройки параметров

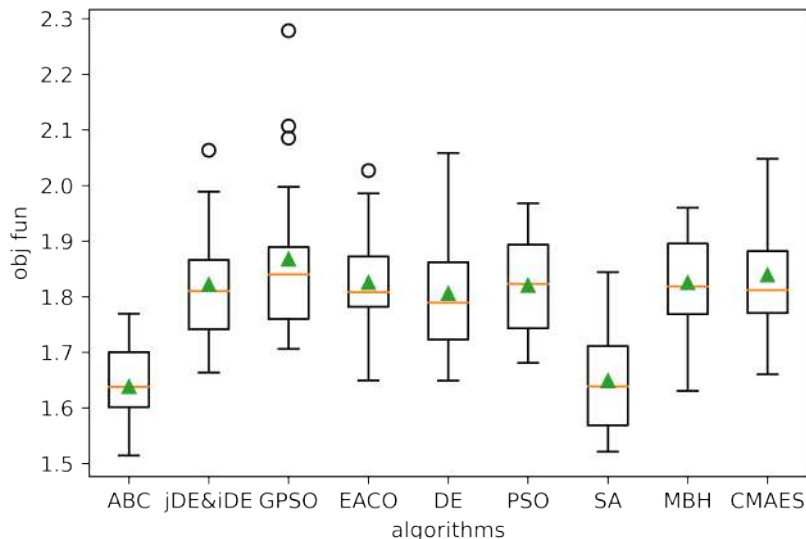


Рис. 1.4: Средние значения приспособленности (N=16).

метаэвристического алгоритма используется простой метод генерации-оценки. Генерируется набор априорных конфигураций-кандидатов. Затем каждая из этих конфигураций оценивается с целью поиска оптимальной конфигурации. Кроме того, исследуются вычислительные возможности указанных алгоритмов на различных масштабах сети. При этом длительность процесса поиска составляет 50000 оценок. При этом в качестве сравнения используются три эталонных алгоритма: FP, взвешенная минимальная среднеквадратичная ошибка и случайная стратегия (RAND), о которых говорилось ранее.

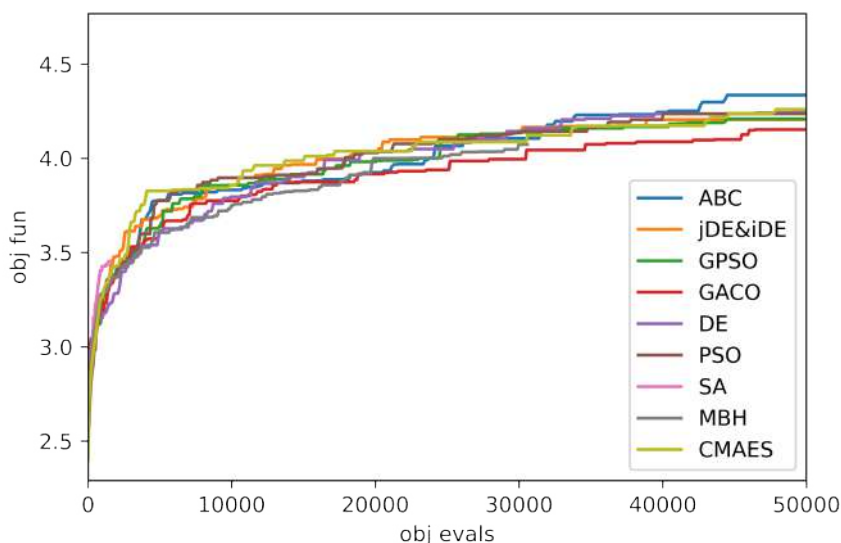
На рисунках 1.2, 1.3 и 1.4 показан процесс поиска для метаэвристических алгоритмов с различными значениями N . Усреднённая суммарная скорость сети используется как значение целевой функции наряду с оценками пригодности. В соответствии со скоростью роста значения пригодности было выделено из диапазона поиска два интервала - [8000, 15000] и [37000, 43000]. Можно заметить, что алгоритмы дифференциальной эволюции всегда имеют хорошую производительность при быстром росте, а алгоритмы роевого интеллекта - при медленном росте. Полученные результаты статистически свидетельствуют о том, что после фиксированного числа поколений производительность предложенных алгоритмов одинакова.

В таблице 1.2 приведено полученное решение численных экспериментов. Была рассмотрена средняя производительность указанных алгоритмов для 20 испытаний. Исходя из предыдущих результатов, алгоритмы роевого интеллекта: МРЧ и

Таблица 1.2: Полученное решение (бит/с/Гц) численных экспериментов.

Алгоритмы	$N=4$			$N=9$			$N=16$		
	max	mean	std	max	mean	std	max	mean	std
АПК	4.241	3.694	0.224	2.612	2.231	0.197	1.769	1.638	0.067
jDE&iDE	4.747	4.221	0.222	3.002	2.524	0.181	2.063	1.821	0.093
ГМРЧ	4.768	4.238	0.402	2.871	2.456	0.169	2.278	1.867	0.143
PMA	5.153	4.160	0.345	2.695	2.474	0.118	2.026	1.825	0.083
ДЭ	4.670	4.252	0.275	2.750	2.484	0.122	2.058	1.805	0.100
МРЧ	4.999	4.303	0.313	2.748	2.467	0.151	1.968	1.820	0.089
ИО	4.198	3.454	0.268	2.510	2.128	0.177	1.844	1.648	0.090
МВН	4.987	4.147	0.292	2.700	2.458	0.131	1.960	1.825	0.083
СМАES	4.114	3.655	0.216	2.605	2.373	0.103	2.048	1.839	0.086

ГМРЧ показывают наилучшие результаты при $N = 4$ и $N = 16$ соответственно. Алгоритмы дифференциальной эволюции: jDE&iDE показывают наилучшие результаты при $N = 9$.

Рис. 1.5: Распределение наилучшей приспособленности по 20 испытаниям ($N=4$).

На рисунках 1.5, 1.6 и 1.7 показаны соответствующие распределения наилучшей пригодности для метаэвристических алгоритмов. СМА-ES является наиболее надёжной методикой, на основании среднего значения наиболее незначительного стандартного отклонения лучших значений.

Получен также численный пример результатов эксперимента с различными плотностями пользователей и масштабами сети. По сравнению со значениями усредненной суммарной скорости на рисунке 1.8 и рисунке 1.9, производительность

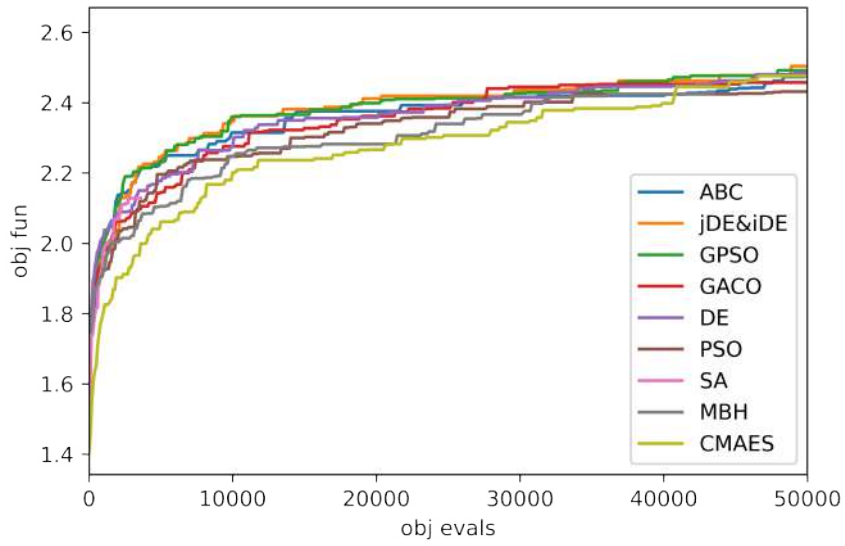


Рис. 1.6: Распределение наилучшей приспособленности по 20 испытаниям (N=9).

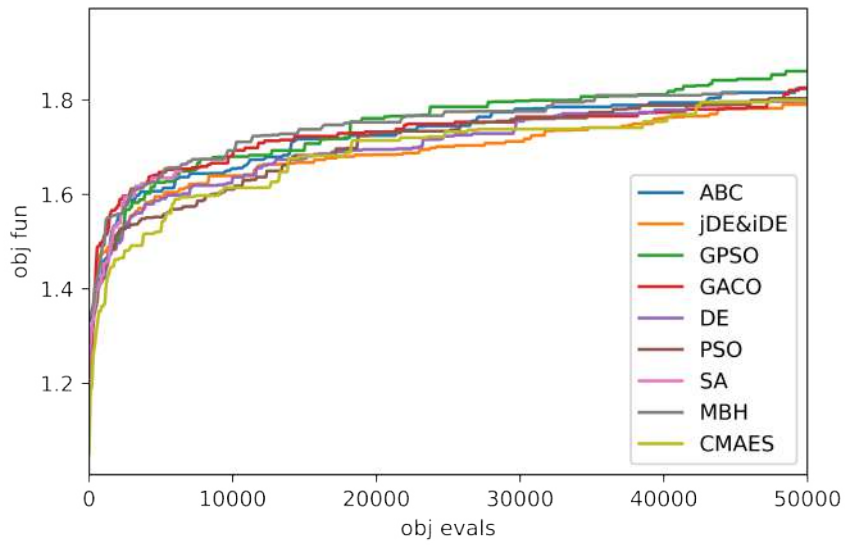


Рис. 1.7: Средние значения приспособленности (N=16).

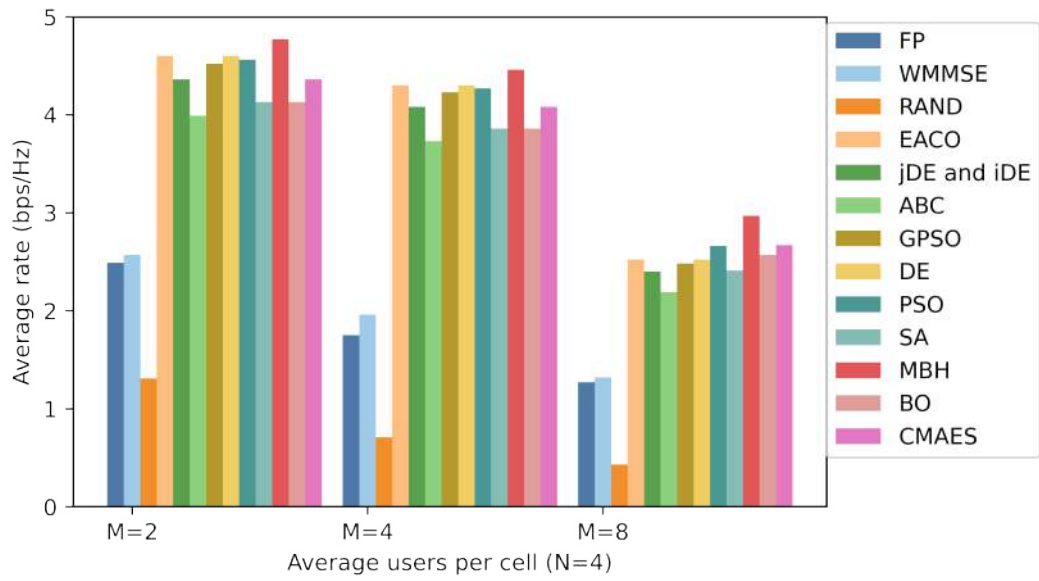


Рис. 1.8: Зависимость средней скорости сети от количества пользователей на ячейку.

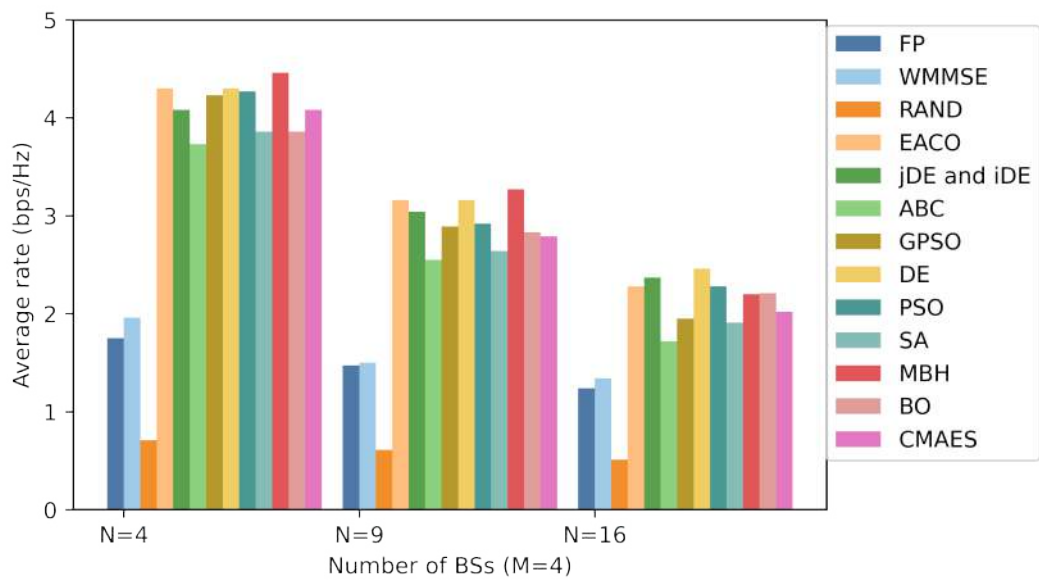


Рис. 1.9: Зависимость средней скорости сети от количества ячеек.

метаэвристических алгоритмов не является стабильной, особенно в зависимости от конкретных эффектов масштаба решения. Кроме того, с увеличением масштаба вычислений решения наилучшее значение функции приспособленности метаэвристических алгоритмов существенно снижается по сравнению с результатом работы обычного алгоритма. В крупномасштабных сценариях такой подход со временем обходится значительно дороже, чем другие алгоритмы, а это значит, что потенциал для повышения эффективности метаэвристических алгоритмов еще есть.

1.5 Заключение к главе 1

Исследована задача оптимального распределения мощности в сотовой сети с IMAC и реализованы метаэвристические подходы, не требующие моделирования, для решения этой задачи. Для соответствия целям оптимизации задачи распределения мощности в качестве целевой функции используется суммарный показатель SINR сети. Затем предлагается ряд метаэвристических алгоритмов, которые работают как "черный ящик" для поиска оптимального распределения мощности при ограничениях, основанных на конкретной информации о состоянии канала.

Результаты моделирования показывают, что предложенные метаэвристические алгоритмы превосходят обычные эталонные алгоритмы в различных сценариях. Можно заметить, что метаэвристические алгоритмы обладают хорошими обобщающими способностями при моделировании сетей связи. Результаты экспериментов статистически показывают, что трудно определить алгоритм-победителя. В целом метаэвристические методы показывают хорошие результаты, а фактический разрыв в производительности связан с масштабами решений. Наиболее устойчивой является ковариационная матрица стратегии эволюции адаптации (СМА-ES). Алгоритмы дифференциальной эволюции (ДЭ, jDE&iDE) и алгоритмы роевого интеллекта (ГМРЧ, МРЧ) демонстрируют превосходство в общих сценариях.

Глава 2

Распределение ресурсов в однородной D2D сети с помощью глубокого обучения под наблюдением

В этой главе мы объединяем эвристические алгоритмы с глубокими нейронными сетями (ГНС) и предлагаем схему распределения ресурсов под названием PSO-DNN. Основное внимание уделяется сквозному обучению, когда отображение входных и выходных значений изучается напрямую, рассматривая алгоритм оптимизации как "черный ящик". Результаты моделирования показывают, что обученная на основе глубокой нейронной сети модель может предоставлять решения, которые приближаются к производительности эвристических алгоритмов и отвечают требованиям распределения ресурсов в реальном времени.

2.1 Общие знания

2.1.1 Общие знания о сетях устройство устройство

Сети Device-to-Device (D2D) - это системы беспроводной связи, в которых конечные устройства устанавливают прямые соединения, исключая необходимость ретрансляции через обычные базовые станции или инфраструктуру. В этом контексте конечные устройства устанавливают прямые соединения друг с другом, сокращая задержки связи и повышая эффективность работы [27].

Для оптимизации эффективности связи ресурсы, такие как спектр и полоса пропускания, распределяются между пользователями. D2D-связь способна увеличить пропускную способность сети, снизить нагрузку на базовые станции и обеспечить работу большего количества устройств и пользователей. Такой подход

к прямой связи обеспечивает повышенную гибкость и эффективность беспроводных систем связи.

Формирование луча служит основополагающей техникой в беспроводной связи для эффективного распределения ресурсов и управления помехами, особенно в сценариях с использованием нескольких антенн. В контексте D2D-связи главной целью проектирования формирования луча является максимизация общей скорости передачи данных путем оптимизации стратегии формирования луча для каждой D2D-пары при соблюдении ограничений на максимальную мощность передачи. Решение этой задачи предполагает решение проблемы непрерывной оптимизации.

2.1.2 Общие знания об основах машинного обучения

Хотя эвристические алгоритмы предлагают приближенные оптимальные решения, время, затрачиваемое на вычисления, все еще не удовлетворяет все более жестким требованиям к времени работы сетей 5G и будущих сетей. Для решения этой проблемы в решении задач оптимального управления широкое распространение получили методы машинного обучения, эффективно ускоряющие сходимость традиционных алгоритмов к точным решениям. В этой главе эвристические алгоритмы рассматриваются как "черные ящики" что позволяет напрямую изучать отображения "вход-выход" с помощью сквозного подхода к обучению [56]. Сквозное обучение упрощает процесс за счет использования единой модели для получения прямых отображений, устраняя необходимость в экспертных знаниях и делая реализацию относительно простой.

Методы, основанные на машинном обучении, доказали свою эффективность в ускорении распределения ресурсов в беспроводных сетях. Эти работы демонстрируют, что машинное обучение не только значительно снижает временную сложность, но и позволяет добиться высокой производительности, близкой к традиционным алгоритмам. Такое повышение производительности является важнейшей характеристикой, особенно актуальной для требований 5G и других доменов беспроводной связи.

Машинное обучение включает в себя две основные методологии обучения: обучение с учителем и обучение без учителя. Основное различие между этими двумя методологиями заключается в необходимости использования размеченных наборов данных для обучения модели.

Обучение с учителем требует значительного количества размеченных наборов данных для облегчения процедуры обучения, что требует опыта людей, которые могут предоставить соответствующие примеры [29]. Применение контролируемого обучения доказало свою эффективность в решении проблем с распределением каналов. Нейронная сеть может быть использована для обучения поиску связи между характеристиками пары D2D (например, геометрическим расстоянием между ними или информацией о состоянии канала) и бинарным решением в контексте задачи планирования каналов D2D. Параметры нейронной сети обычно обновляются с помощью размеченных выходных значений и целей. Для расчета расхождений между измерениями используется функция потерь кросс-энтропии.

Обучение без учителя относится к парадигме обучения, которая не требует использования помеченных данных в процессе обучения [30]. Обучение без контроля может решать сложные задачи непрерывной оптимизации, включая распределение мощности и проектирование формирования луча. Метод обучения без контроля может быть использован для обнаружения оптимального отображения информации о состоянии канала на формирование луча в контексте проектирования формирования луча D2D. Обычно в качестве функции потерь используется показатель отрицательной суммы для определения подходящего времени обновления нейронной сети. В этом сценарии использование размеченных обучающих образцов необязательно.

Поскольку эта глава посвящена в первую очередь распределению ресурсов с помощью машинного обучения, она начинается с краткого обзора основ машинного обучения. В этом обзоре рассматриваются различия между обучением с учителем и без учителя, классические методы машинного обучения и ключевые метрики для оценки эффективности обучения.

2.2 Системная модель сетей D2D

Мы исследовали одноячеичную D2D-сеть, подобную изображенной на рисунке 2.1, в которой несколько пар приемопередатчиков конкурируют за фиксированный объем полосы пропускания B . Мы предполагаем, что по каждому соединению может одновременно передаваться и приниматься один поток данных. Задача оптимизации пропускной способности заключается в разработке передатчика с формированием луча для каждого потока данных на каждом соединении [31].

Пусть имеется $K = \{1, \dots, k\}$ каналов связи; другими словами, k передающих антенн обслуживают k одноантенных пользовательских устройств (ПУ). Отклик канала от передатчика канала связи j до приемника i равен h_{ij} . i и j - индексы приемника и передатчика. Пусть x_i - вектор формирования луча для связи i . Соответственно, сигнал, поступающий на приемник i , является суперпозицией сигналов от множества передатчиков, что описывается следующим образом

$$y_i = h_{ii}^H x_i + \sum_{j \neq i} h_{ij}^H x_j + n_i, \quad (2.1)$$

где $n_i \sim \mathcal{N}(0, \sigma_i^2)$ обозначает аддитивный белый гауссовский шум. Достижимая скорость передачи суммы по каналу i может быть выражена в виде функции:

$$R_i(X) = W \log \left(1 + \frac{\|h_{ii}^H x_i\|_2^2}{\sum_{j \neq i} \|h_{ij}^H x_j\|_2^2 + \sigma_i^2} \right), \quad (2.2)$$

где X - множество всех векторов формирования луча, причем $X = (x_1, \dots, x_k)$. При использовании одной антенны на передатчике задача формирования луча сводится к задаче распределения мощности.

Как правило, суммарная производительность системы связи определяется функцией полезности от возможных скоростей соединения. В качестве функции полезности здесь используется взвешенная сумма скоростей. Учитывая ограничение мощности каждого передатчика, задача оптимизации формулируется как

$$\begin{aligned} \max_X \sum_i w_i R_i(X) \\ s.t. \|x_i\|_2^2 \leq P_{\max}, \forall i, \end{aligned} \quad (2.3)$$

где w_i обозначает вес соединения i , а P_{\max} - ограничение на мощность передатчи по каждому соединению. Когда все веса соединений равны 1, задачу можно рассматривать как задачу максимизации суммарной скорости.

2.3 Архитектура алгоритма

В этом разделе мы построили эффективную схему на основе нейронной сети глубокого обучения для решения задач распределения ресурсов в сетях ММО. Для

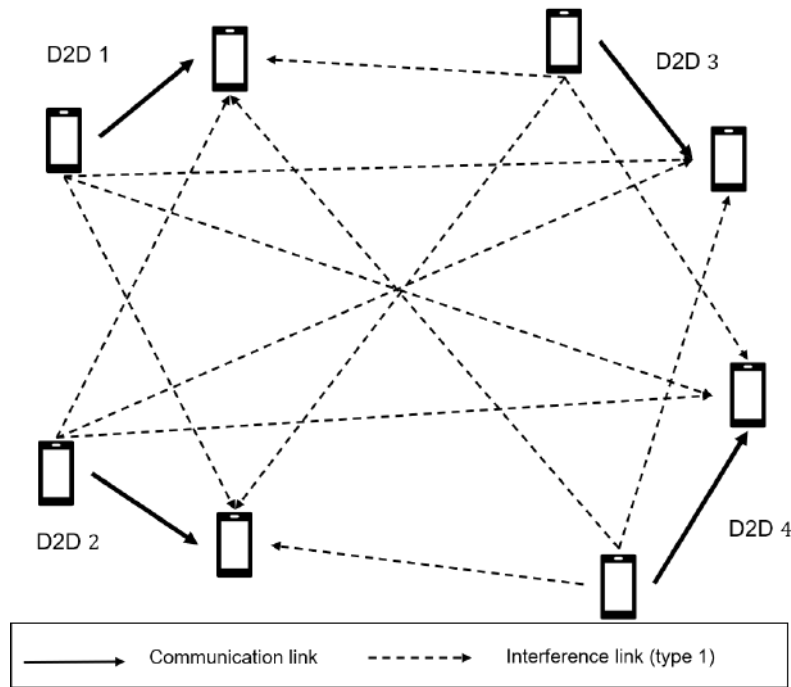


Рис. 2.1: Пример D2D сети с 2 парами приёмопередатчиков.

достижения близкой к глобальной оптимальной скорости передачи в режиме реального времени мы представили двухэтапную методику распределения мощности на основе ГНС. На первом этапе мы используем эвристический метод случайного поиска для определения оптимального распределения мощности, которое оптимизирует суммарную скорость системы для каждого статического состояния канала. На втором этапе мы прогнозируем распределение мощности в онлайн-обучении в реальном времени с помощью хорошо обученной модели глубокой нейронной сети. Далее описывается введение алгоритма PSO и разработка структуры ГНС для оптимального формирования луча:

2.3.1 Глубокие нейронные сети

Мы разрабатываем архитектуру полносвязной ГНС, которая прогнозирует оптимальное распределение ресурсов для K ПУ, работающих в нисходящем канале связи [13]. Задача состоит в том, чтобы найти закон $P(\cdot)$, который моделирует отображение PSO, обозначаемое F , для оценки оптимального распределения ресурсов $\hat{P} \triangleq \{\hat{p}_k\}$. Выбирается ГНС-параметризация $P(\cdot)$ с обучаемыми параметрами, а векторы формирования луча оцениваются как $\hat{P} = P_\theta(F)$.

На первом этапе, в режиме пакетного обучения с учителем, алгоритм PSO, требующий больших вычислительных затрат, определяет наилучшую стратегию распределения и использует ее в качестве выходной метки. Помеховые отношения

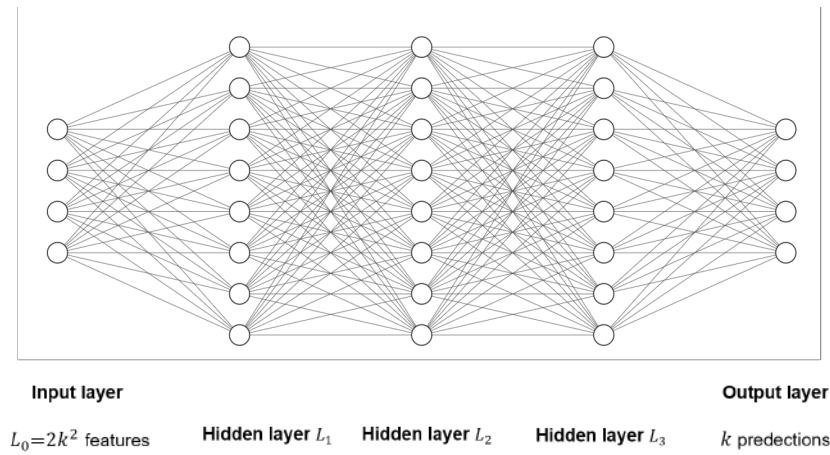


Рис. 2.2: Производительность в зависимости от расстояния между парами.

характеризуются как набор канальных коэффициентов $\{h_{ij}\}$, отношение связи характеризуется как набор канальных коэффициентов $\{h_{ii}\}$. Заметим, что h - это комплексная функция, состоящая из вещественной и комплексной частей. Эти две части мы принимаем за признаки и вводим их в нейронную сеть. Z_0 представляет собой вектор признаков входного слоя. R - множество вещественных чисел. Входной признак ГНС можно сформулировать как:

$$Z_0 = \left[h_{ii}^{complex}, \dots, h_{ii}^{real}, \dots, h_{ij}^{complex}, \dots, h_{ij}^{real}, \dots \right], \quad (2.4)$$

где $L_0 = 2k^2$ - размерность входного признака. Масштабирование и векторизация входных признаков применяются на входах предлагаемой ГНС. Мы используем стратегию оптимального распределения по наибольшей абсолютной пропорции, что аналогично входным признакам. Аналогично входным признакам, мы используем метод максимального абсолютного масштабирования для оптимальных векторов лучевой коррекции следующим образом:

$$\bar{x}_k = \frac{x_k^{optimal}}{\max \left(x_1^{optimal}, \dots, x_k^{optimal} \right)} \in [0, 1]. \quad (2.5)$$

Для обработки нелинейных вычислений в качестве функции активации в скрытых слоях используется выпрямленный линейный блок (ReLU). На i -м скрытом слое имеется L_i нейронов, где $i = 1, 2, 3$. Z_0 - входной коэффициент, выход i -го скрытого слоя определяется как $Z_i = f_r(w_{i-1}Z_{i-1} + b_{i-1}) \in R^{L_i}$, где $f_r(Z) = \max(0, Z)$ - весовая матрица и вектор смещения соответственно. Для согласования предсказаний выходного слоя между 0 и 1 в соответствии с выходными метками

на выходном слое используется сигмоидальная функция $f_{\sigma}(Z) = \frac{1}{1+e^{-Z}}$. Размерность выходного фактора составляет k . Таким образом, прогнозируемое распределение ресурсов для K ПУ, работающих в нисходящем канале, с помощью ГНС выражается следующим образом:

$$[\hat{x}_1, \hat{x}_2, \dots, \hat{x}_k] = f_{\sigma}(W_3 Z_3 + b_3), \quad (2.6)$$

где W_i - весовые матрицы, а b_i - векторы смещения, которые корректируются для уменьшения потерь и более точного прогнозирования оптимальных значений распределения мощности. Мы оцениваем функции потерь на основе прогнозируемых и идеальных значений мощности: средняя квадратичная ошибка (MSE). Формула для функции потерь MSE имеет вид:

$$L_{MSE} = \frac{1}{K} \sum_{k=1}^K (\bar{x}_k - \hat{x}_k)^2. \quad (2.7)$$

Обратное распространение - это процесс, в котором градиент функции потерь передается от выходного слоя к входному. В результате весовая матрица W_i и вектор смещения b_i обновляются с целью уменьшения потерь, что позволяет более эффективно обучать выборки и более точно предсказывать оптимальную стратегию распределения ресурсов.

2.3.2 Формирование массивов данных и обучение моделей

Мы создаем набор данных для процедуры автономного контролируемого обучения с количеством выборок $S = 5 \times 10^5$. Усиление канала и местоположение пользовательских устройств относительно базовых станций распределяются случайным образом, чтобы получить вектор канала для каждого ПУ. Алгоритм PSO используется для определения соответствующих оптимально назначенных мощностей, которые вычисляются и сохраняются в наборе данных.

В процессе пакетного обучения весь имеющийся набор данных разбивается на 80% обучающих и 20% проверочных наборов. После обучения с учителем на новом тестовом наборе оценивается распределение мощности в онлайн режиме в условиях изменяющихся во времени сцен. Предложенная алгоритмическая методика реализована с использованием открытого фреймворка глубокого обучения PyTorch.

2.4 Моделирование

2.4.1 Настройка среды

Для оценки эффективности алгоритма на основе ГНС мы сравниваем его в первую очередь с подходом на основе WMMSE (weighted minimum mean-square error), который является общим эталоном в литературе для задач максимизации общей скорости сети. Также для сравнения мы рассматриваем следующие эталоны. Обратите внимание, что все результаты, относящиеся к производительности теста WMMSE, усреднены по 100 независимым испытаниям. Настройки системы и гиперпараметры глубокой сети приведены в таблице 2.1. Бенчмарки представлены следующим образом:

- WMMSE [3]: Подход, основанный на оптимизации, позволяющий преобразовать задачу уменьшения средневзвешенной квадратичной ошибки из трудности максимизации суммарной скорости в интерферирующих широкополосных каналах.
- WMMSE-NN [32]: Трехслойная управляемая ГНС, обучающаяся отображению классического WMMSE.
- PSO [33]: Метод итеративной стохастической оптимизации на основе роевого интеллекта.

Таблица 2.1: Настройка системы и гиперпараметры DNN

Параметр	Значение
Радиус квадратного участка	500 m
Мощность передачи БС	20 dBm
D2D Расстояние между парами	2-50 m
Экспонента потерь пути	$\eta = 3.76$
Спектральная плотность мощности шума	-174 dBm/Hz
Полоса пропускания канала	5 MHz
Размер скрытого слоя	(512-512-512-256)
Размер эпохи	30
Размер выборки	32
Скорость обучения	1×10^{-3}
Коэффициент отсева	0.5
Оптимизатор	ADAM

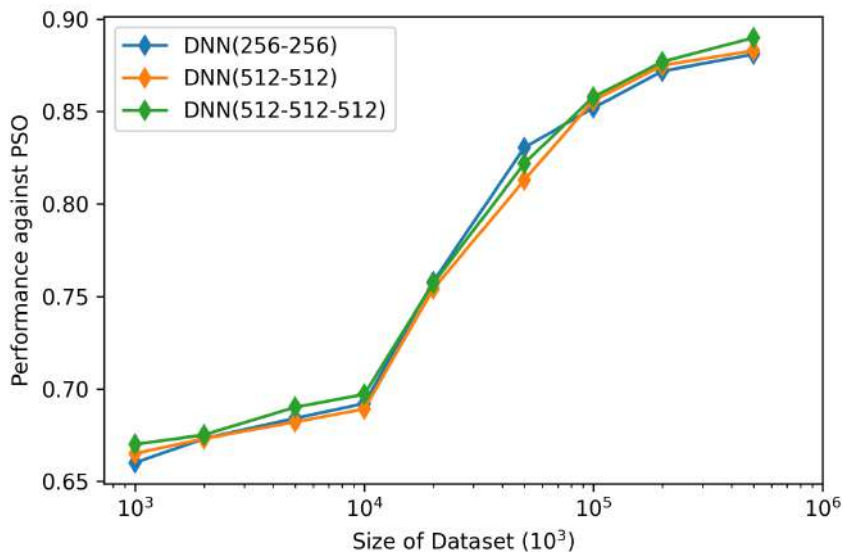


Рис. 2.3: Процентная производительность PSO для моделей с различным количеством свёрточных слоев и размерами скрытых слоев на наборах данных разного размера.

2.4.2 Численные результаты

На рисунке 2.3 показана сравнительная производительность ГНС с различными структурами в зависимости от размера обучающего множества. Повысить производительность нейронной сети позволяет либо добавление большего числа свёрточных слоёв, либо параметризация с помощью более крупных слоев многослойного персептрона (МП). Чтобы продемонстрировать преимущества расширения МП, мы показываем результаты работы ГНС с числом слоев (2 и 3) и размером скрытых слоев МП (256 и 512). При подаче на вход образцов размером 5×10^5 относительная производительность трехслойной ГНС с размером скрытого слоя 512 составляет 91.5%. Добавление дополнительных скрытых слоев в ГНС не дает существенного улучшения, а наоборот, может легко привести к переобучению. Для того чтобы сбалансировать производительность и сложность, мы используем эту структуру ГНС в последующих испытаниях.

На рисунках 2.4 и 2.5 показаны результаты суммарной скорости и времени выполнения в зависимости от количества ПУ. Как видно из рисунка 2.4, подход на основе ГНС превосходит свой аналог WMMSE при увеличении числа пользовательских устройств. Относительная суммарная производительность ГНС по сравнению с оптимальным PSO-алгоритмом составляет 91.5%.

Кроме того, на рисунке 2.5 показано сравнение времени работы PSO, WMMSE-NN и WMMSE для 1000 состояний системы. Модели ИИ выполняются графиче-

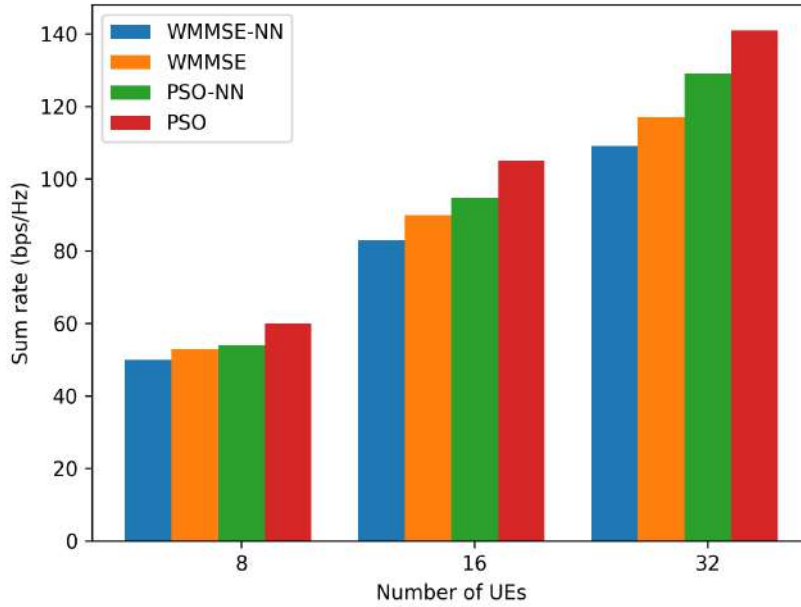


Рис. 2.4: Общая скорость суммирования для ГНС с различной плотностью ПУ.

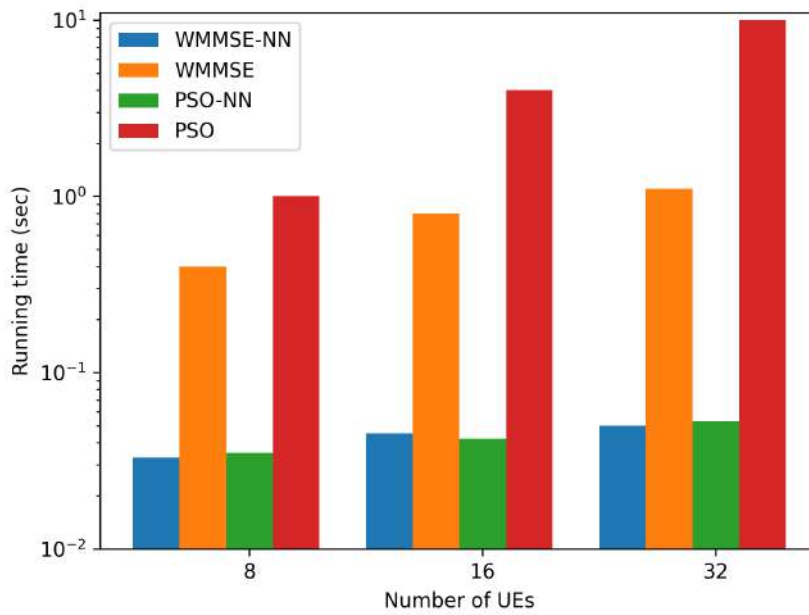


Рис. 2.5: Среднее время работы для 100 испытаний.

ском процессоре Nvidia 2070s с использованием архитектуры DNN, обученной в пакетном режиме. Мы видим, что предложенный DNN-подход превосходит вычислительно трудоемкую методику WMMSE за счет резкого сокращения времени выполнения. Например, в случае $K = 12$ ПУ для WMMSE требуется 1 056 секунд, а для ГНС - всего 0,08 секунды. Для итерационного WMMSE время работы резко возрастает с увеличением размерности задачи. В отличие от этого, распределение мощности на основе ГНС требует меньшего времени обработки и гораздо менее вариативно. Для обученной модели ГНС количество вычислений постоянно. Колебания времени обработки происходят из-за неопределенности вычислений различных чисел с плавающей запятой и системного времени считывания. Для эвристического алгоритма колебания времени происходят в основном за счет различных инициализаций, т.е. различные начальные точки поиска могут приводить к значительным различиям во времени нахождения оптимального решения.

2.5 Заключение к главе 2

В этой главе мы представили схему PSO-DNN, основанную на контролируемом обучении и предназначенную для решения задач распределения ресурсов в сетях D2D, особенно в сложных реальных сценариях беспроводной связи. Мы продемонстрировали эффективность этого решения в конкретных сетевых конфигурациях и сценариях использования, подчеркнув его способность к аппроксимации эвристики. Стоит отметить, что эти возможности, такие как аппроксимация эвристики, в целом применимы к широкому кругу сетей и сценариев. Было показано, что ГНС способны аппроксимировать практически любую функцию при адекватном обучении, что делает данный фреймворк универсальным.

Наши результаты показывают, что обученная модель PSO-DNN превосходит алгоритм на основе WMMSE, особенно по времени вычислений, при этом ее производительность лишь немного хуже, чем у PSO-алгоритма (на 92%). Это значительное улучшение, учитывая, что экономия вычислительного времени составляет 99%.

Однако есть ряд проблем, которые мы не до конца рассмотрели в данной работе, и необходимы дальнейшие исследования. Важным моментом является определение оптимальной структуры ГНС, которая в значительной степени зависит от особенностей настройки крупномасштабной системы связи D2D и выбора ги-

перпараметров. Кроме того, ГНС неэффективно используют топологию коммуникационной сети, что может сказаться на эффективности обучения. В следующей главе мы исследуем структуры оптимальных решений для более сложных задач распределения мощности и интегрируем их в фреймворк на основе глубокого обучения.

Глава 3

Распределение ресурсов в однородной D2D сети с помощью графового обучения под наблюдением

Как уже говорилось в главе 2, модели глубокого обучения нашли широкое применение в различных сетях связи, доказав свою высокую эффективность при решении целого ряда вопросов, таких как проектирование сети, прогнозирование трафика и распределение ресурсов. Тем не менее, во многих из этих исследований топология сети использовалась лишь частично. Это ограничение возникает потому, что большинство глубоких нейронных сетей приспособлены для работы с данными, структурированными евклидовым образом. В последние годы для устранения этого недостатка появились методы глубокого обучения на основе графов, примером которых являются графовые нейронные сети, предлагающие решения для неевклидовых структурированных данных. Графовые нейронные сети продемонстрировали способность решать проблемы в коммуникационных сетях. Их сила заключается в способности эффективно улавливать скрытую пространственную информацию в топологии сети и эффективно обобщать ее даже в тех случаях, когда сеть динамична или обладает невидимой топологией. В этой главе мы представляем две системы нейронного обучения графов с учителем на основе PSO.

3.1 Общие сведения об изучении графиков

Графы - это основополагающая математическая концепция, используемая для изображения универсальных отношений, состоящих из множества узлов и связей, которые их соединяют. В последнее время графы нашли обширное применение в

представлении множества реальных сценариев, что привело к их признанию в качестве фундаментальных структур данных. В области машинного обучения подходы, основанные на графовых структурах и известные как машинное обучение на графах, включают такие методы, как встраивание графов и нейронные сети [34].

Встраивание графа означает преобразование узлов или ребер графа в векторное пространство с уменьшенной размерностью. Этот процесс называют встраиванием сети или обучением представлению графа [35]. Для того чтобы сохранить целостность структуры и характеристик графов, объемные, многомерные, разнообразные, сложные и постоянно меняющиеся данные преобразуются в стандартизированное, уменьшенное по размерности и компактное векторное представление. Соответствующие исследования направлены на решение таких задач, как классификация и кластеризация узлов, предсказание связей, реконструкция и визуализация графов. Предложенная методика отличается низкой вычислительной сложностью. Поэтому в реальных сценариях встраивание графов рассматривается как подготовительный шаг.

Графовая нейронная сеть (ГНС) - это нейронная сеть, предназначенная для анализа и обработки данных на основе графов [38]. Модели ГНС отлично справляются с входными данными, характеризующимися разнообразием масштабов, неоднородностью и сложными топологическими особенностями. Их способность эффективно извлекать сложную топологическую информацию, выявлять критические и сложные аспекты данных и выполнять быструю обработку данных является убедительной и надежной. Система ГНС включает в себя ряд типов моделей, в том числе графовые свёрточные сети (GCN), графовые сети с механизмом внимания (GAT), графовые автокодировщики (GAE) и графовые пространственно-временные сети (GSTN).

В теории графов простой граф определяется как $G = (V, E)$, где V - множество узлов, а E - ребра, соединяющие эти узлы. Исходя из отношения связности, $\mathcal{N}(v_i)$ обозначает множество соседних узлов v_i , а каждый элемент матрицы степеней D равен $D_{ii} = \mathcal{N}(v_i)$. Вводится матрица Лапласиана неориентированного графа, которая определяется как $L = D - A$. Кроме того, нормированная матрица Лапласиана определяется как $L = I_N - D^{-\frac{1}{2}}AD^{\frac{1}{2}}$, где N - количество узлов, а I_N - матрица тождеств размерности N . Матрица признаков узлов графа выражается как $X \in R^{N \times d}$, где d - размерность вектора признаков узлов.

Вначале представлена парадигма встраивания графов (GE). В математике под вложением понимается функция отображения, обозначаемая как $f : X \rightarrow Y$, которая облегчает перенос точек из одного пространства X в другое пространство Y . Как правило, встраивания выполняются путем отображения данных из высокоразмерного абстрактного пространства в более низкоразмерное. Такая практика обусловлена тем, что нейронные сети лучше справляются с обработкой данных с меньшей размерностью, что приводит к повышению эффективности.

Графовые свёрточные сети (GCN) вдохновлены принципами моделей свёрточных нейронных сетей (CNN)[36]. Основной задачей при создании окончательной модели нейронной сети является введение переменных параметров, а затем использование градиентного спуска для оптимизации их работы. Метод свертки графов на основе спектрального анализа использует концепции обработки сигналов для представления атрибутов узлов в графе. Вышеупомянутый метод можно классифицировать как свёрточное преобразование в рамках преобразования Фурье. Процедура обработки включает в себя интегрирование сигнала во временной области с собственной функцией Лапласа.

Операция свертки графа, обозначаемая как X_* , в GCN определяется следующим образом:

$$X_* = W \left(I_N + D^{-\frac{1}{2}} A D^{\frac{1}{2}} \right) X, \quad (3.1)$$

где W обозначает обучаемую весовую матрицу. Для решения проблемы потенциального градиентного взрыва к процессу свертки графа применяется дополнительное преобразование в виде

$$X_* = W \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{\frac{1}{2}} \right) X, \quad (3.2)$$

где $\tilde{A} = I_N + A$ и $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$.

Напротив, пространственные GCN устанавливают операции свертки в соответствии с топологией графа. Например, в нейронные сети для передачи сообщений (MPNN) предлагается использовать функцию передачи сообщений, которая включает в себя фазу передачи сообщений и фазу считывания [39]. Фаза передачи сообщений определяется следующим образом:

$$m_{v_i}^t = \sum_{v_j \in \mathcal{N}(v_i)} \mathcal{M}^t (X_i^{t-1}, X_j^{t-1}, e_{ij}), \quad (3.3)$$

где $m_{v_i}^t$ обозначает сообщение, агрегированное из соседнего узла v_i , \mathcal{M}^t - функция агрегирования, а t - число итераций. В дальнейшем процесс считывания сообщений определяется следующим образом:

$$X_i^t = u^t (X_i^{t-1}, m_{v_i}^t), \quad (3.4)$$

где u^t обозначает функцию считывания сообщений.

Графовая модель сети внимания представляет собой новый подход к обработке данных, структурированных графами, с использованием механизма внимания [37]. Механизм внимания может избирательно концентрироваться на наиболее важной информации среди огромного количества данных. Основной задачей GAT является вычисление скрытого представления каждого узла внутри графа путем приоритизации окружающих узлов в его окрестности с помощью механизма внимания. Предлагаемый подход представляет собой метод свертки графа, который работает на основе пространства, определяя веса соседних узлов на основе их соответствующих степеней. Поэтому он подходит для решения задач индуктивного обучения и может быть применен к задачам, связанным со скрытыми графами. GAT интегрирует механизм внимания в шаг распространения и использует многоголовый механизм внимания для повышения устойчивости модели. Формула GAT выглядит следующим образом:

$$X_i^t = \sigma \left(\frac{1}{K} \sum_{k=1}^K \sum_{j \in \mathcal{N}(v_i)} \alpha^k (X_i^{t-1}, X_j^{t-1}) W^{t-1} X_j^{t-1} \right) \quad (3.5)$$

где σ обозначает функцию активации, а α^k обозначает k -ый механизм внимания.

3.2 Графовое представление сетей D2D

Мы формулируем оптимизацию скорости суммирования как обучение на прямом однородном графе. Наборы $G = (V, E)$ представляют собой формальное описание сети, где V и E - множества вершин и ребер. Индекс каждого канала связи и интерференционного канала обозначается как вершина $i \in V$ и ребро $(i, j) \in E$. Вершина i является пересечением сущностей, а ребро (i, j) задает направленную связь между вершинами i и j . Соседнее множество вершины i обозначается через $\mathcal{N}_i = \{j \in V \mid (j, i) \in E\}$. v_i и e_{ij} определяют атрибуты вершины i и ребра ij ,

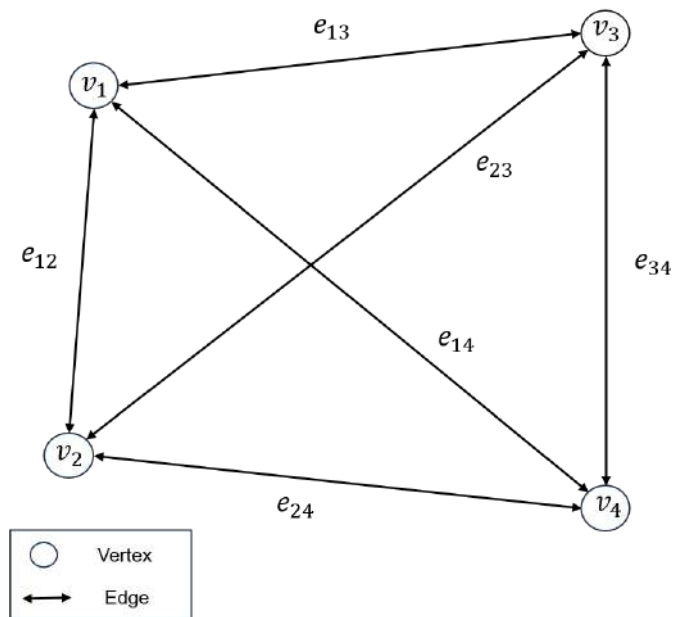


Рис. 3.1: Пример представления графа сети D2D.

соответственно. Простая сеть и ее графовое представление показаны на рисунке 3.1.

Рассмотрим нескольких типов вершин или ребер в неоднородном графе. Обозначим множество типов вершин как S , множество типов ребер как T . Граф может быть представлен двумя функциями отображения, которые отображают каждую вершину на соответствующий тип вершины $\varphi : V \rightarrow S$ и каждое ребро на соответствующий тип ребра $\psi : E \rightarrow T$. Обозначим i -ю вершину как v_i , $\mathcal{N}_i = \{j \in V \mid (j, i) \in E\}$ - множество смежных вершин i . Характеристики вершин представлены $V \triangleq \{v_i\}_i$. Характеристики ребер представлены как $E \triangleq \{e_{ij}\}_{i,j}$, если существует ребро (j, i) и 0 в противном случае.

Атрибуты каждой вершины содержат ее вес w_i , отклик прямого канала h_{ii} и дисперсию шума σ_i^2 . Атрибуты каждой вершины содержат отклик канала $e_{ij} = [h_{ij}, h_{ji}]$ от передатчиков-помех к приемникам-помехам. Кроме того, поскольку существуют различные виды соединений с уникальным количеством передающих и принимающих антенн, размерность признаков графа может меняться.

3.3 Модели обучения графовых нейронных сетей

Обучение сети на основе графовых нейронных сетей (GNN) с учителем состоит из двух этапов: (i) На первом этапе с помощью эвристического алгоритма осуществляется поиск помеченных значений оптимальных назначений и соответствующих

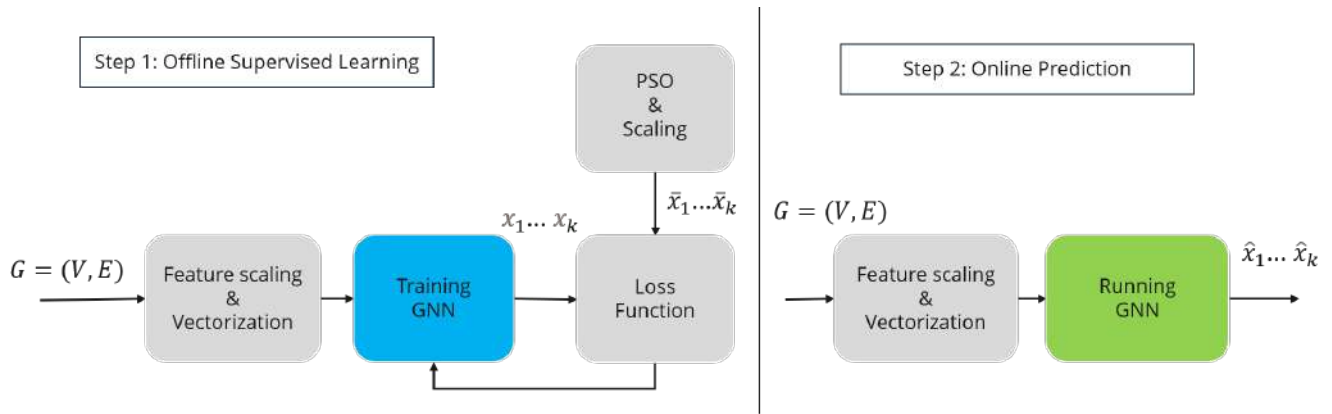


Рис. 3.2: Архитектура фреймворка PA-GNN.

состояний сети в качестве набора данных и обучение сети. (ii) На втором этапе обученная модель запускается в онлайн режиме в реальном времени для прогнозирования оптимального распределения ресурсов [45].

3.4 Модель 1: Нейронная сеть с передачей сообщений

Графовые нейронные сети имеют иерархическое построение нейронной сети, как и обычный многослойный перцептрон (МП). Для обновления представления каждого узла в каждом слое МП объединяют признаки ребер и соседних узлов. Схема обновления i -го узла на слое l в графовой сети выражается следующим образом:

$$Aggregation : \alpha_v^{(l)} = \phi^\alpha \left(\left\{ \beta_u^{(l-1)} : u \in \mathcal{N}(v) \right\} \right), \quad (3.6)$$

$$Combination : \beta_v^{(l)} = \phi^\beta \left(\beta_v^{(l-1)}, \alpha_v^{(l)} \right), \quad (3.7)$$

где $\alpha_v^{(l)}$ представляет собой признак, агрегированный узлом v из соседнего узла на слое l . $\beta_v^{(l)}$ обозначает признак узла v на слое l . ϕ^α - параметризованная функция, которая кодирует атрибуты вершин и ребер для каждого ребра (j, i) . Тогда каждая вершина i агрегирует обновления ребер. ϕ^β представляет собой комбинационную функцию, используемую для получения обновления вершины путем объединения агрегированного обновления ребер $\alpha_v^{(l)}$ с текущими атрибутами вершины $\beta_v^{(l-1)}$. Передача сообщений завершается, когда знания каждой вершины встраиваются в обновления ребер и, соответственно, усваиваются соседними вершинами.

Агрегирование окрестностей с помощью операций, инвариантных относительно

но перестановки элементов (например, сумма, среднее и максимум), позволяет учесть инвариантные признаки канала помех. Поскольку размер признаков рёбер зависит от количества антенн, признаки из разных отношений не могут быть напрямую обработаны обычной графовой нейронной сетью, и эти признаки должны рассматриваться отдельно. Поэтому мы назначаем отдельные функции обновления для каждого отношения, используя многослойный персептрон для параметризации. Сначала в каждом отношении происходит передача сообщения. Затем целевая вершина производит выборку и агрегирует частичные обновления из нескольких отношений для получения окончательного обновления. Определим обновление в вершине i как:

$$e_{ij}^{(l)} = \phi^e \left(v_j^{(l-1)}, e_{ij}^{(0)} \right) \quad (3.8)$$

$$v_i^{(l)} = \phi^v \left(v_i^{(l-1)}, \max_{j \in \mathcal{N}_i} e_{ij}^{(l)} \right) \quad (3.9)$$

Начальные атрибуты ребер $e_{ji}^{(0)}$ сохраняются на всех этапах обновления ребер. Это позволяет сохранить стабильность работы. Атрибуты каждой вершины используются в качестве исходных данных в процессе прямого вычисления, после чего выполняется несколько итераций для получения вектора формирования луча.

3.5 Модель 2: Графовые сети с механизмом внимания

Необходимо отобразить эти векторы в высокоразмерное пространство с помощью слоя графовой сети внимания (GAT), чтобы расширить информацию о топологии сети, содержащуюся в исходных низкоразмерных признаках \mathbf{n}_k , полученных от сети. На вход сети GAT подается набор признаков узлов, обозначаемых как $\mathbf{n} = (\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_K)$, $\mathbf{n}_i \in \mathbb{R}^{d_v}$, где каждое \mathbf{n}_i является характеристикой существующей пары БС-ПУ v_i .

Теоретически GAT может использовать все узлы, кроме центрального, для расчета сходства с ним. Размеры пространств признаков узлов и ребер могут быть различными. Для повышения выразительности узловых признаков добавляется слой нейронных сетей с общим параметром для линейного преобразования признаков (как для узлов, так и для ребер), который обозначается Att . Определяется $\mathbf{W} \in \mathbb{R}^{d_v \times d_v}$ как обучаемая весовая матрица, линейно преобразующая входные

признаки в признаки высокого уровня, размерность вектора признаков узла изменяется с d_V на d'_V . Механизм самовнушения вычисляет сходство между центральным узлом и соседними узлами, где сходство вычисляет слой нейронной сети. Параметры обозначаются β^T , и два преобразованных вектора признаков после сшивки поступают в эту сеть. Коэффициент важности узла v_i по отношению к узлу v_j выражается как

$$c_{i,j} = \text{Attention}(\mathbf{Wn}_i, \mathbf{Wn}_j). \quad (3.10)$$

Мы используем метод ”масочного внимания позволяющий фокусироваться исключительно на соседних узлах первого порядка, включая сам узел. При этом вычисляется коэффициент корреляции между узлами и соседними с ними узлами, а затем применяется метод регуляризации softmax. Коэффициент может быть сформулирован в виде

$$\text{Attention}(\mathbf{Wn}_i, \mathbf{Wn}_j) = \beta^T \text{LeakyReLU}(\beta^T [\mathbf{Wn}_i \parallel \mathbf{Wn}_j]). \quad (3.11)$$

На протяжении всего этого процесса происходит совместное объединение признаков, параметры которых определяются вектором внимания β^T и активационной функцией *LeakyReLU*. Результирующий вес, нормированный с помощью функции softmax для узла v_j , выражается как

$$\alpha_{i,j} = \text{softmax}_j(c_{i,j}) = \left(\frac{\exp(\sigma(\beta^T [\mathbf{Wn}_i \parallel \mathbf{Wn}_j]))}{\sum_{k \in \mathcal{N}_i} \exp(\sigma(\beta^T [\mathbf{Wn}_{nk} \parallel \mathbf{Wn}_k]))} \right), \quad (3.12)$$

где $\beta_n^T \in \mathbb{R}^{2d'_V}$ обозначает вектор внимания для соседей по узлу. Механизм внимания реализуется в виде однослойной нейронной сети с прямолинейным движением, характеризуемой параметрами \mathbf{W} и β . В этой нейронной сети используется нелинейность *LeakyReLU* с отрицательным наклоном 0,2. Переменная $\alpha_{i,j}$ обозначает оценку внимания, которая используется для количественной оценки значимости соседнего узла v_j относительно узла v_i .

3.6 Моделирование

В этом разделе оценивается эффективность PSO-MPNN и PSO-GAT при различных параметрах системы. Для подтверждения эффективности предложенного ме-

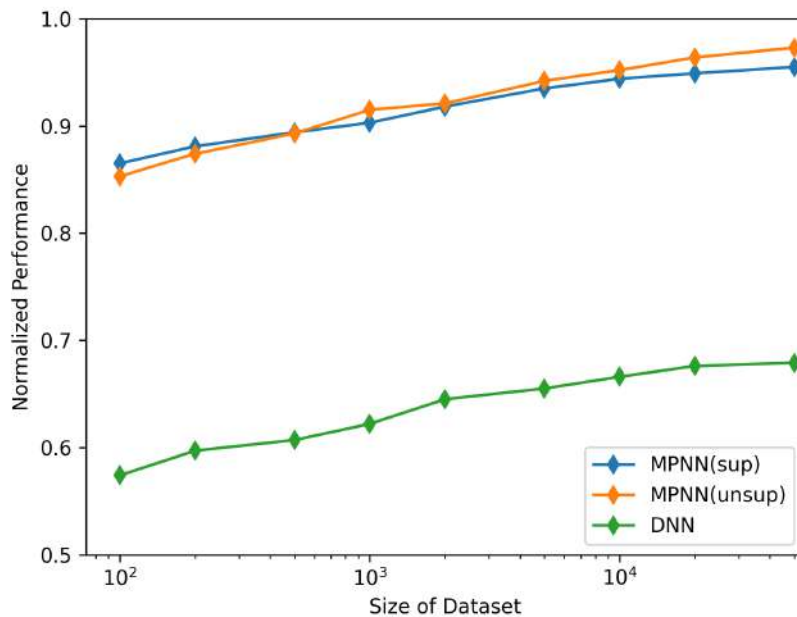


Рис. 3.3: Зависимость производительности от размера набора данных.

тогда мы сравниваем его с современными базовыми методами, включая методы встраивания гетерогенных сетей и методы на основе графовых нейронных сетей.

3.6.1 Настройка среды

Для оценки эффективности предложенной управляемой графовой нейронной сети (GNN) мы используем ту же однородную сетевую среду D2D, которая была описана в главе 2. Вычислительное устройство, использованное для проведения симуляций в данном исследовании, оснащено центральным процессором Intel(R) Core CPU i7-12700H, графическим процессором Nvidia RTX 2070s и оперативной памятью объёмом 32 Гбайт. В качестве основы экспериментальной методики использовалась библиотека Deep Graph Library (DGL) и фреймворк PyTorch.

3.6.2 Численные результаты

Вначале рассматривается эффективность обучения на основе подхода, основанного на обучении. Как показано на рисунке 3.3, производительность ГНС постепенно улучшается по мере увеличения размера обучающей выборки. В целом существенной разницы в производительности между обучаемыми ГНС с контролем и без контроля нет, а ГНС с контролем при меньшем количестве выборок демонстрируют более высокую производительность, возможно, потому, что вход-

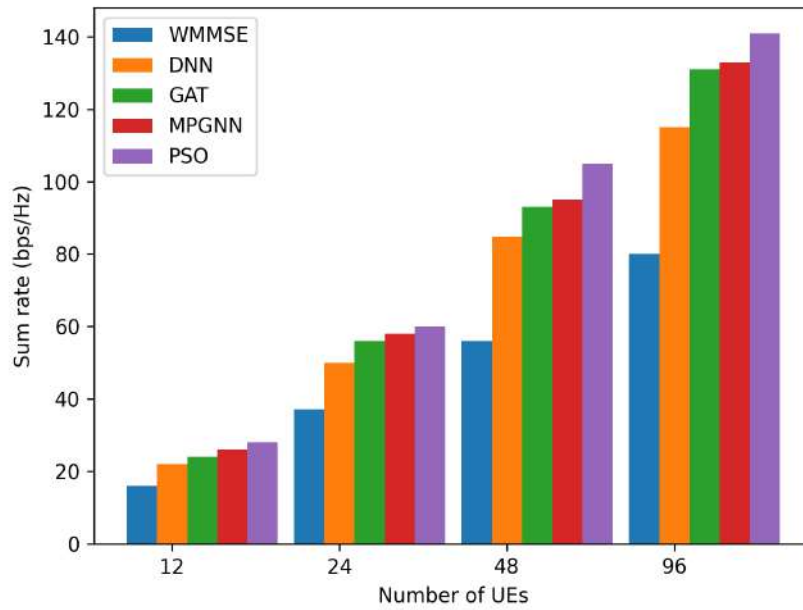


Рис. 3.4: Зависимость производительности от количества пар D2D.

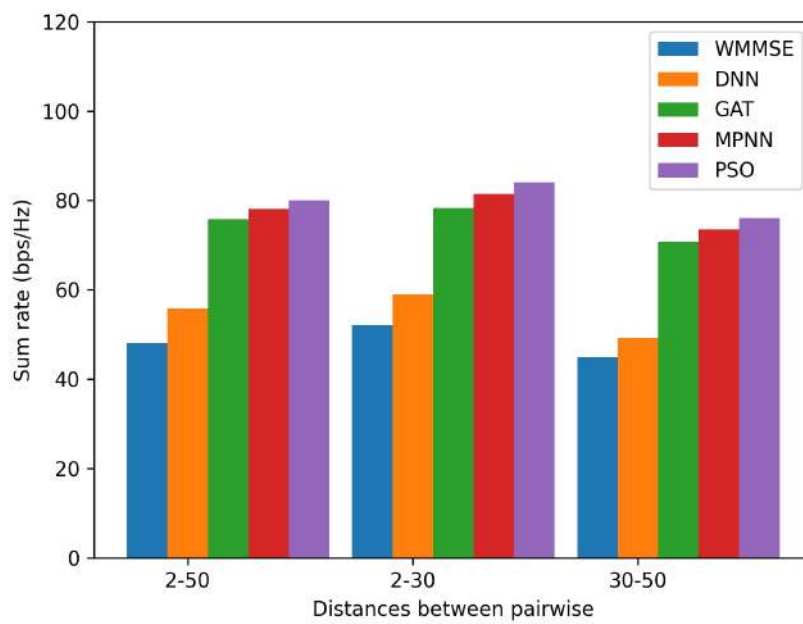


Рис. 3.5: Производительность в зависимости от расстояния между парами.

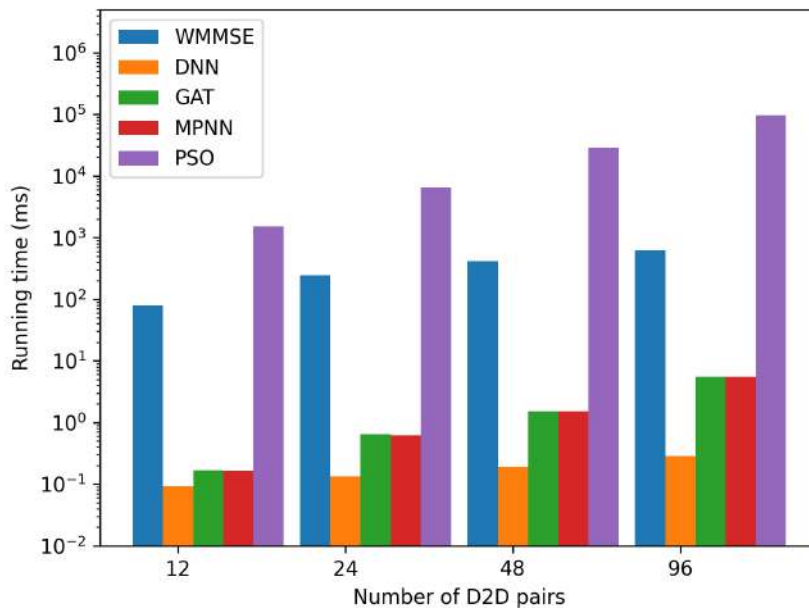


Рис. 3.6: Время выполнения различных бенчмарков.

ные признаки предлагаемого нами метода включают предварительные знания об оптимальном распределении, близком к глобальному. Кроме того, классические глубокие сети, обученные с учителем, демонстрируют наихудшие результаты среди всех методов. Ввиду того, что такие глубокие сети основаны на данных, они обычно требуют больших наборов обучающих данных. Благодаря высокой эффективности выборки, графовые нейронные сети часто рекомендуются для решения реальных задач в беспроводных сетях, особенно когда сбор большого количества обучающих данных может быть дорогостоящим или неудобным.

Далее можем оценить масштабируемость предложенного метода в сетях с различными парными расстояниями и плотностью связей. Как показано на рисунке 3.4, мы рассматриваем производительность предложенного метода в сравнении с эталонными образцами для сетей различного размера и плотности связей. При наличии 12 пар в сети производительность MPGNN и GAT достаточно близка и отличается от близкого к глобальному оптимального решения PSO всего на 0,05. При последовательном увеличении числа связей в 2, 4 и 8 раз и аналогичном увеличении площади области коэффициент нормализации MPGNN остается стабильно выше 0,97, а GAT - только на уровне 0,95. Это говорит о том, что в данном конкретном случае супервизорный GNN не является лучшим решением. Возможные причины, по которым MPGNN несколько превосходит GAT, заключаются в том, что по сравнению с GAT, который при обучении ориентируется в основном

на признаки узлов, MPGNN параллельно агрегирует веса ребер как дополнительную информацию внутри узлов. Это дополнение к выборочным признакам способствует несколько более высокой производительности по сравнению с GAT.

На рисунке 3.5 показана эффективность работы алгоритма на основе графовой нейронной сети при различных расстояниях сопряжения D2D. Примечательно, что алгоритм на основе ГНС постоянно поддерживает средний коэффициент нормализации выше 0,96 по сравнению с алгоритмом PSO, даже при изменении распределения расстояний сопряжения. Такая устойчивая производительность может быть объяснена способностью ГНС учитывать зависимость усиления канала от расстояния путем встраивания геометрической информации о беспроводной сети в характеристики узла. Таким образом, система оптимизации на основе ГНС превосходит другие эталонные схемы по всем трем параметрам.

Как показано на рисунке 3.6, PSO и WMMSE занимают много времени при увеличении числа пар D2D, и задача усложняется по мере расширения сети. Традиционные алгоритмы PA, как правило, занимают много времени и не подходят для приложений реального времени. В отличие от них, решение задачи оптимизации в сетях D2D значительно облегчает подход на основе ГНС. Он более чем в 100 раз быстрее стандартных алгоритмов и более чем в 1000 раз быстрее эвристического алгоритма PSO. Столь значительное ускорение предлагаемого метода является хорошим предзнаменованием для его реализации в беспроводных сетях в режиме реального времени. Поскольку предложенный метод MPGNN имеет ту же структуру сети и те же входные признаки, что и GAT, его производительность во время работы сопоставима. За счет игнорирования особенностей графа глубокая сеть достигает лучших временных характеристик, чем графовая, однако ее терминальная производительность и эффективность выборки низки.

3.7 Заключение к главе 3

В этой главе мы разработали два масштабируемых нейросетевых фреймворка на основе графовых нейронных сетей, а именно MPGNN и GAT, для решения проблемы распределения ресурсов в однородных беспроводных сетях D2D. Мы сосредоточились на создании нейронных структур, отвечающих таким важным критериям эффективности, как минимальная стоимость обучения, высокая вычислительная эффективность и сильная обобщающая способность, которые имеют пер-

востепенное значение для современных подходов, основанных на обучении.

Кроме того, в наш подход были органично интегрированы эвристические и обучающие методы оптимизации. В данной работе мы предлагаем модель графовой нейронной сети, которая использует обучение с учителем для обновления параметров. Этот подход отличается от большинства других методов ГНС, поскольку эффективно использует особенности узлов и ребер для иллюстрации взаимодействия глубинных компонентов, что повышает выразительность модели.

Проведенное нами моделирование показало, что предложенный метод PSO-GNN превосходит другие методы на основе глубоких сетей по максимизации коэффициента суммирования. Это объясняется, прежде всего, эффективным использованием априорных знаний об оптимизированных решениях, которые ищут эвристические алгоритмы. Кроме того, наблюдая разницу в производительности MPGNN и GAT, мы получили представление о влиянии особенностей рёбер на производительность модели, что открывает путь для дальнейших исследований.

Глава 4

Распределение ресурсов в гетерогенной D2D сети с помощью графовой сети с механизмом внимания, улучшенной данными о рёбрах графа

Методы глубокого обучения на основе графов стали перспективным подходом для решения проблем распределения сетевых ресурсов. Учитывая присущую мобильным сетям топологическую природу, использование GNN оказывается выгодным при работе с данными, структурированными в виде графов. Однако основное ограничение этих методов заключается в том, что в процессе обучения они уделяют особое внимание характеристикам узлов, зачастую игнорируя или чрезмерно упрощая роль характеристик рёбер, которые не менее важны. В этой главе мы представляем новый дизайн, известный как Heterogeneous Edge Feature Enhanced Graph Attention Network (HEGAT). Этот дизайн напрямую связывает развивающуюся топологию сети с оптимальными стратегиями распределения ресурсов на этапе обучения. С помощью обширного моделирования мы установили исключительную производительность и надежные обобщающие возможности HEGAT в сотовых сетях, характеризующихся различными параметрами.

4.1 Общие сведения о гетерогенных графах

В предыдущих главах подчеркивалась необходимость того, чтобы методы распределения ресурсов более эффективно соответствовали базовой структуре коммуникационных сетей. Обычные глубокие нейронные сети явно предназначены для обработки данных, организованных в евклидову структуру, таких как изоб-

ражения и видео. В ответ на это ограничение в данной области появились GNN - класс моделей глубокого обучения на основе графов, способных обрабатывать данные, не имеющие евклидовой структуры, и призванные устранить этот пробел. Хотя GNN добились значительного прогресса в решении проблем распределения ресурсов в системах связи, они сталкиваются с двумя серьезными проблемами. Одна из них заключается в том, что в большинстве предыдущих исследований основное внимание уделялось характеристикам узлов, при этом упускалась из виду потенциальная ценность, заложенная в характеристиках ребер. Также существует острая необходимость в более полной оценке значимости структуры соседних ячеек и оценке связей между ребрами и узлами. Существуют также некоторые исследования, связанные с ребрами графов:

Нейронные сети с передачей сообщений (MPNN) содержат несколько фаз передачи и считывания сообщений. Эта парадигма полностью учитывает особенности узлов. В то же время, для характеристики сети также используются признаки ребер, и этот механизм обновления признаков был предложен для предсказания признаков узлов [39]. Хотя MPNN добавляет информацию о ребрах на этапе передачи сообщений, ее механизм передачи не может изучить топологические отношения между узлами и ребрами. Реляционные графовые сети (Relational Graph Convolution Networks, RGCN) используют правила прямого прохождения для добавления дополнительных весов ребер в матрицу весов правил [40]. Однако эксперименты показали, что такое простое агрегирование сложно вычислить и оно не дает существенного улучшения производительности. В нейронной сети Edge Feature Graph Neural Network (EGNN) используется функция агрегирования для объединения признаков узлов при обучении отдельных весов внимания для каждого признака размерности. Полученные результаты измерений затем объединяются [44]. Однако это явление также приводит к потере информации о ребрах.

С другой стороны, графы в окружающем мире часто состоят из различных типов узлов и ребер и в совокупности известны как гетерогенные информационные сети (HIN)[40]. В этой статье мы будем называть их гетерогенными графами для простоты. Гетерогенность - фундаментальная характеристика этих графов, включающая различные типы узлов и ребер, каждый из которых имеет свои особенности, расположенные в различных пространствах признаков. Эти гетерогенные графы, обогащенные обширной информацией, находят применение в многочисленных задачах интеллектуального анализа данных. Из-за их сложности обыч-

ные графовые нейронные сети не могут быть прямолинейно использованы для решения уникальных задач, которые ставят перед собой гетерогенные графы. В свете этих соображений разработка архитектуры графовой нейронной сети с механизмом внимания для гетерогенных графов должна отвечать нескольким новым требованиям. Например: Графовая нейронная сеть с гетерогенными помехами (HIGNN) предназначена для работы с учётом гетерогенности сети [13]. В качестве узлов выступают коммуникационные соединения между D2D, классифицирующиеся в зависимости от количества антенн на передатчиках в их соединениях. В системах нисходящей связи D2D этот метод используется для распределения мощности. Для решения проблемы распределения ресурсов в сценариях связи с сочетанием сетей D2D и сотовых сетей [45] представлена гетерогенная сверхплотная сеть (HUDN). Каналы связи рассматриваются как узлы, а узлы классифицируются в соответствии с типами устройств, к которым они подключаются. Гетерогенная GNN (HetGNN) была представлена для определения политики распределения мощности для многосотовых многопользовательских систем. Однако в этих работах не учитывались характеристики ребер графа.

4.2 Системная модель и графовое представление гетерогенных D2D сети

4.2.1 Модель системы

В данном разделе рассматривается гетерогенная беспроводная сеть и моделируется сценарий передачи данных по нисходящей линии связи, состоящий из множества одноходовых D2D-пар, как показано на рис. 4.1. Основной целью данной работы является оптимизация политики управления ресурсами каждой D2D-пары в реальном времени в соответствии с требованиями пользователей. В отличие от предыдущих исследований, в настоящей работе основное внимание уделяется практическому сценарию, включающему гетерогенные сети, в которых для установления коммуникационных соединений используется несколько типов каналов. Важно отметить, что данная схема учитывает наличие в сети передатчиков с различным количеством антенн и приемников с одной антенной.

Рассматривается гетерогенная сеть D2D, в которой пары приемопередатчиков используют один и тот же спектр полосы пропускания B . Обозначим $\mathcal{M} \triangleq \{1, \dots, M\}$

как множество типов линий связи и $m, n \in \mathcal{M}$. Величина, обозначающая количество передающих антенн для канала связи типа n , обозначается как A_n . Пусть j_n обозначает j -е соединение типа n , $\mathbf{h}_{j_n j_n} \in \mathbb{C}^{A_n \times 1}$ обозначают канал связи между передатчиком и приемником соединения i_m , $\mathbf{h}_{j_n i_m} \in \mathbb{C}^{A_m \times 1}$ обозначим канал помеховой связи от передатчика соединения i_m к приемнику соединения j_n . $\mathbf{x}_{j_n} \in \mathbb{C}^{A_n}$ представляют собой вектор формирования луча соединения j_n . Тогда принимаемый сигнал на приемнике соединения j_n определяется следующим образом:

$$y_{j_n} = \mathbf{h}_{j_n j_n}^H \mathbf{x}_{j_n} s_{j_n} + \sum_{i_m \neq j_n} \mathbf{h}_{j_n i_m}^H \mathbf{x}_{i_m} s_{i_m} + z_{j_n}, \quad (4.1)$$

где $z_{j_n} \sim \mathcal{N}(0, \sigma_{j_n}^2)$ обозначают аддитивный белый гауссовский шум (AWGN), $s_{j_n} \sim \mathcal{CN}(0, 1)$ обозначает искомый знак соединения j_n . Пусть $\mathbf{X} \triangleq \{\mathbf{x}_{j_n}\}_{j_n}$ обозначает матрицу вектора формирования луча для всех линий связи. Скорость передачи данных приемнику j_n формулируется как:

$$\xi_{j_n}(\mathbf{X}) = B \log \left(1 + \frac{\left\| \mathbf{h}_{j_n j_n}^H \mathbf{x}_{j_n} \right\|_2^2}{\sum_{i_m \neq j_n} \left\| \mathbf{h}_{j_n i_m}^H \mathbf{x}_{i_m} \right\|_2^2 + \sigma_{j_n}^2} \right). \quad (4.2)$$

Для оценки эффективности системы используется взвешенная сумма скоростей. Целевая функция для проектирования формирования луча с целью максимизации общей скорости сети может быть выражена следующим образом:

$$\begin{aligned} \max_{\mathbf{X}} \sum_{j_n} w_{j_n} \xi_{j_n}(\mathbf{X}) \\ s.t. \left\| \mathbf{x}_{j_n} \right\|_2^2 \leq p_{\max}, \forall j, n, \end{aligned} \quad (4.3)$$

где p_{\max} - ограничение на максимальную передаваемую мощность канала связи, а w_{i_m} - вес канала i_m .

4.2.2 Графовое представление

Используя полностью связный граф, мы создаем представление канала помех D2D. Пары D2D рассматриваются как узлы, а интерференционные связи - как ребра. Как показано на рис 4.1, существует два типа узлов, в зависимости от типа связи, и три типа ребер, в зависимости от типа узла, с которым они соединяются.

Пара j_n -передатчик-приемник является j -ым узлом типа n , обозначаемым как v_{j_n} . Атрибуты узла содержат в себе состояние прямого канала, вес и тип связи.

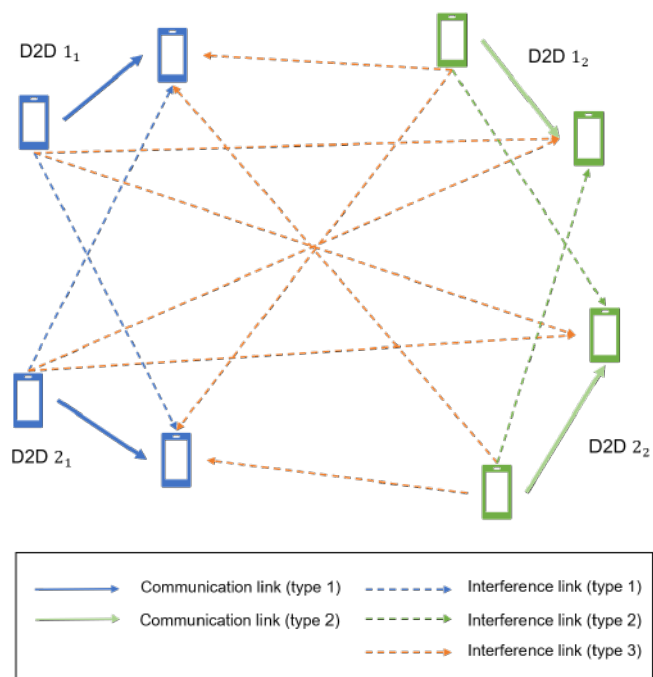


Рис. 4.1: Графовое представление сети D2D с двумя типами связей.

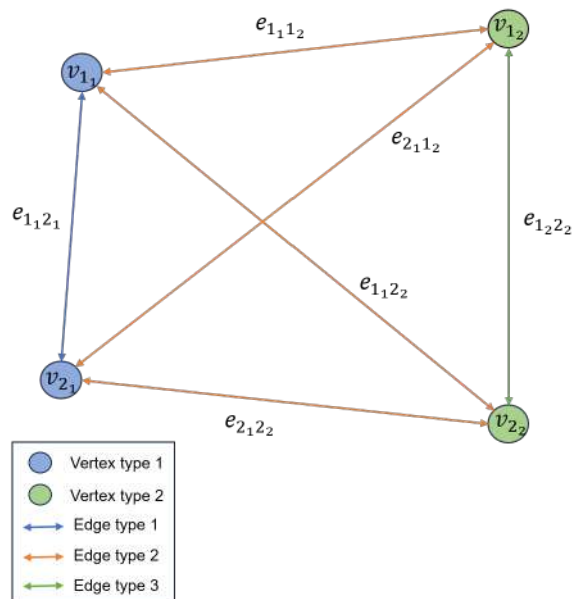


Рис. 4.2: Графовое представление.

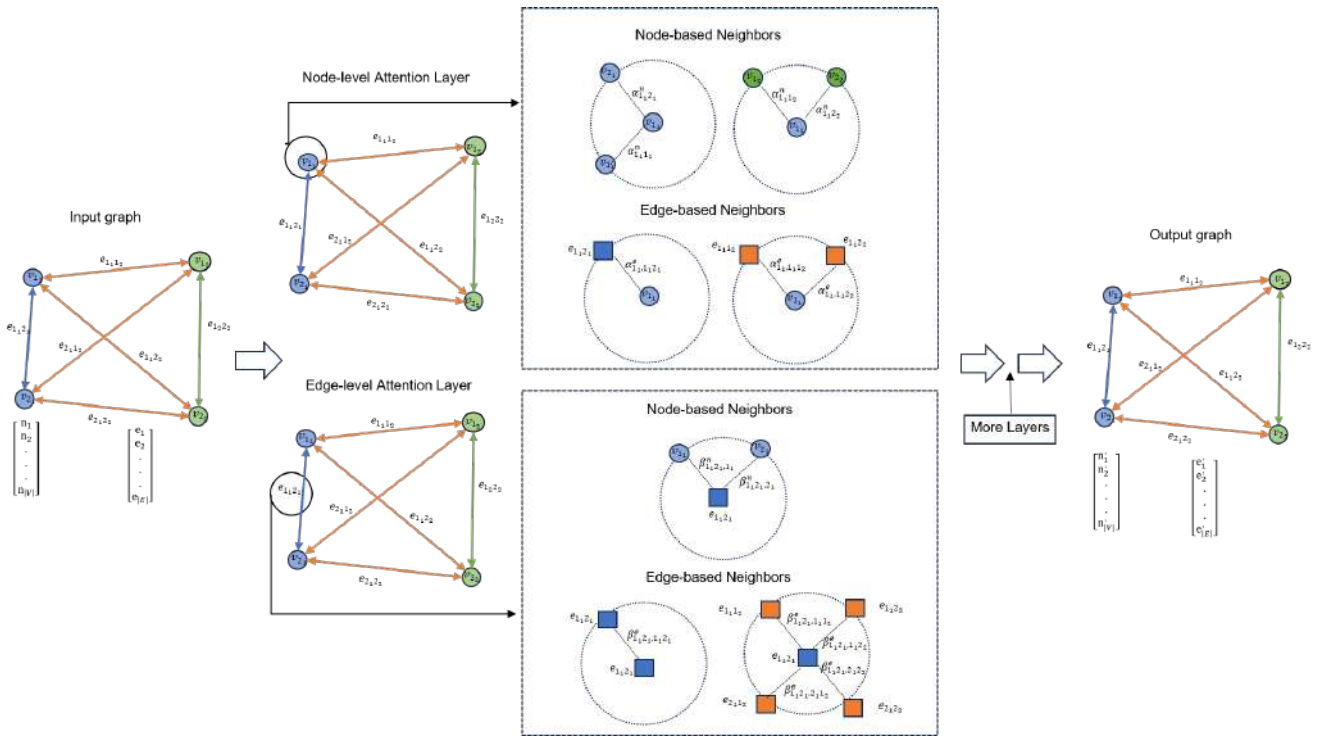


Рис. 4.3: Архитектура предлагаемой системы HNENN для вложения графов. Механизм многоуровневого внимания HNENN, слой внимания на уровне узла и слой внимания на уровне ребра, альтернативный узел обучения и вложения ребёр.

Ребро $e_{j_n i_m}$, соединяющее два узла, v_{j_n} и v_{i_m} , представляет собой интерференционную связь, где атрибуты содержат состояния интерференционных каналов и типов. В беспроводных сетях атрибуты как узлов, так и ребер не менее важны, чем топология сети. Для учета этих атрибутов мы определяем граф неоднородной сети как $G = (V, E)$, где V представляет собой множество узлов, а $E \subset \{\{j_n, i_m\} | j_n, i_m \in V\}$ представляет собой множество ребер. Отображение узлов и ребер на соответствующие им признаки обозначается через $R : \phi \rightarrow \mathbb{C}^{d_V}$ и $S : \psi \rightarrow \mathbb{C}^{d_E}$, где d_V и d_E обозначают размерность пространства признаков для узла и ребра, соответственно.

4.3 Гетерогенные графовые сети с механизмом внимания

В этом разделе мы представляем структуру предлагаемой системы HNENN. В ней используется иерархическое объединение разнородного внимания на уровне узлов и ребер, что позволяет обучать соответствующие вложения узлов и ребер. Характеристики узлов и ребер обновляются путем агрегирования информации из окрестностей.

В гетерогенном графе узловых соседей \mathcal{N}_i данного узла i можно определить

как совокупность узлов, непосредственно связанных с узлом i . Следует отметить, что в число соседей данного узла входит и сам узел. Аналогично, узловых соседей \mathcal{N}_{ij} данного ребра ij можно определить как совокупность узлов, непосредственно связанных с ребром ij .

Как показано на рисунке 4.2, кружки представляют узлы, а квадраты - ребра. В слое внимания на уровне узлов соседями узла v_{1_1} являются узлы v_{1_1} и v_{2_1} типа 1, а также v_{1_2} и v_{2_2} типа 2. В слое внимания на краевом уровне соседями узлов на основе краевых $e_{1_1 2_1}$ являются v_{1_1} и v_{2_1} типа 1.

В гетерогенном графе ребровой сосед \mathcal{E}_i некоторой вершины i может быть определен как совокупность ребер, непосредственно соединенных с вершиной i . Аналогично, реберный сосед \mathcal{E}_{ij} данного ребра ij может быть определен как совокупность ребер линейного графа, непосредственно связанных с ребром узла ij . Следует отметить, что в число соседей данного ребра входит и само ребро.

Как показано на рисунке 4.2, в слое внимания на уровне узлов соседями узла v_{1_1} по ребрам являются $e_{1_1 2_1}$ типа 1, $e_{1_1 1_2}$ и $e_{1_1 2_2}$ типа 2. В слое внимания на уровне края соседями по краю $e_{1_1 2_1}$ являются $e_{1_1 2_1}$ типа 1, $e_{1_1 1_2}$, $e_{1_1 2_2}$, $e_{2_1 1_2}$ и $e_{2_1 2_2}$ типа 2.

4.3.1 Процесс гетерогенной трансформации

Перед тем как агрегировать информацию о соседних узлах или ребрах в коммуникационной сети, необходимо понять, что соседи каждого узла или ребра могут играть различные роли и проявлять различную значимость при изучении вложенных узлов или ребер для распределения ресурсов. Предлагается использовать механизм внимания на уровне узлов и ребер, который позволит эффективно изучать значимость соседей каждого узла или ребра в гетерогенном графе. Используя предложенный механизм внимания, мы можем объединить представления о влиятельных соседях для построения вложений узлов или ребер, которые показывают, какой вклад они вносят в распределение ресурсов.

Вследствие гетерогенности узлов и ребер различные типы узлов и ребер имеют различную размерность пространства признаков. Признак состоит из атрибута и типа, причем первые являются непрерывными комплексными переменными, а вторые - дискретными. Для каждого типа узла или ребра мы используем матрицу преобразований M , предложенную в [89]. Эта матрица позволяет преобразовывать признаки узлов разных типов в единое пространство признаков. Обозначим \mathbf{p}_i^{attr} как вектор атрибутов узла i , а \mathbf{p}_i^{type} как вектор типов узлов i . Математически

процесс преобразования узла можно представить в виде $\mathbf{n}_i^{attr} = M_\kappa \mathbf{n}_i^{attr}$. Заметим, что мы повторно использовали \mathbf{n}_i^{attr} для упрощения формулировки. Признак узла i можно представить в виде $\mathbf{n}_i = [\mathbf{n}_i^{attr} \parallel \mathbf{n}_i^{type}]$, который получается путем конкатенации его преобразованного атрибута узла и информации о типе. Аналогично, процесс преобразования ребра можно представить как $\mathbf{e}_{ij}^{attr} = M_\chi \mathbf{e}_{ij}^{attr}$. Характеристика ребра ij может быть представлена в виде $\mathbf{e}_{ij} = [\mathbf{e}_{ij}^{attr} \parallel \mathbf{e}_{ij}^{type}]$. Механизм внимания на уровне узлов и ребер может работать с произвольными типами узлов и ребер, используя разнородные операции преобразования.

4.3.2 Слой внимания на уровне узла

Слой внимания на уровне узлов в HNENN специально разработан для обучения вложений узлов путем использования ценной информации из разнородных краевых признаков. Очевидно, что различные соседи каждого узла играют разные роли и имеют разную степень значимости при формировании вложений узлов. Для решения этой проблемы мы предлагаем ввести механизм внимания на уровне узла. Этот механизм определяет коэффициенты важности соседей на основе узлов и ребер для узла i .

В l -ном слое HNENN входной признак представляет собой набор признаков узлов, обозначаемый как $\mathbf{n} = \{\mathbf{n}_i \mid i \in [1, \dots, |V|]\}$, где \mathbf{n}_i - вектор признаков узла i . Кроме того, существует набор признаков ребер, обозначаемый как $\mathbf{e} = \{\mathbf{e}_{ij} \mid i, j \in [1, \dots, |V|]\}$, где \mathbf{e}_{ij} соответствует признаку ребра, направленного из узла j в узел i . Важность узла j или ребра ij по отношению к узлу i может быть выражена следующим образом:

$$\mathbf{c}_{ij}^n = Att_{node}^n (W_n \mathbf{n}_i, W_n \mathbf{n}_j), \quad (4.4)$$

$$\mathbf{c}_{i,j}^e = Att_{node}^e (W_n \mathbf{n}_i, W_e \mathbf{e}_{ij}), \quad (4.5)$$

где Att_{node}^n и Att_{node}^e представляют собой внимание на уровне узла, осуществляемое глубокой нейронной сетью. Для любого заданного узла i связи, сформированные со всеми его соседями по узлу, имеют общий вес Att_{node} . Вес соседних узлов или ребер относительно конкретного узла зависит от их особенностей. Примечательно, что значимость асимметрична, то есть значимость узла i или ребра ij для узла j не обязательно равна значимости узла j для узла i или узла i для ребра ij .

Это наблюдение показывает, что внимание на уровне узлов может сохранять свой асимметричный характер, что является важным свойством в контексте гетерогенных графов. Аналогичным образом это свойство может быть распространено и на внимание на уровне ребер. W_n и W_e - обучаемые весовые матрицы, которые преобразуют признаки вершин из низкой размерности в высокую с помощью линейного отображения - распространенного метода расширения признаков.

Учет структурной информации осуществляется с помощью маскировки внимания, благодаря чему на встраивание конкретной вершины i влияют только соседние вершины j или ребра ij . Соответственно, коэффициент важности узла j по отношению к узлу i нормируется с помощью функции softmax как

$$\alpha_{ij}^n = \text{softmax}_j(\mathbf{c}_{ij}^n) = \left(\frac{\exp(\sigma(a_n^T [W_n \mathbf{n}_i \| W_n \mathbf{n}_j]))}{\sum_{k \in \mathcal{N}_j} \exp(\sigma(a_n^T [W_n \mathbf{n}_i \| W_n \mathbf{n}_k]))} \right), \quad (4.6)$$

где $a_n \in \mathbb{R}^{2d_v^{l+1}}$ обозначает вектор внимания соседей на уровне узла. Он отображает конкатенированные высокоразмерные признаки в вещественное число. Конкретно в данной работе это отображение реализуется с помощью однослойной нейронной сети с прямой связью. А σ обозначает активацию LeakyReLU. Коэффициент важности ребра ij для узла i нормируется с помощью функции softmax как

$$\alpha_{i,j}^e = \text{softmax}_{ij}(\mathbf{c}_{i,j}^e) = \left(\frac{\exp(\sigma(a_e^T [W_n \mathbf{n}_i \| W_e \mathbf{e}_{ij}]))}{\sum_{st \in \mathcal{E}_i} \exp(\sigma(a_e^T [W_n \mathbf{n}_i \| W_e \mathbf{e}_{st}]))} \right), \quad (4.7)$$

где $a_e \in \mathbb{R}^{d_v^{l+1} + d_e^{l+1}}$ обозначает вектор внимания соседей по ребрам на уровне узла.

Собрав информацию о коэффициентах значимости α_{ij}^n и $\alpha_{i,j}^e$, можно выполнить процесс агрегирования узла встраивания i с учетом соответствующих коэффициентов значимости. Пусть \mathbf{n}_i^{l+1} обозначает выход признаков узла i для узловых соседей. Он может быть выражен как линейная комбинация признаков,

$$\mathbf{n}_i^l[\mathcal{N}_i] = \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^n W_n \mathbf{n}_j^l \right). \quad (4.8)$$

Гетерогенный характер графовых данных приводит к высокой дисперсии, обусловленной в основном их безмасштабными характеристиками. Для повышения устойчивости процесса обучения самовнимания (self-attention) [90] расширяется

внимание на уровне узлов до многоповторного внимания [91]. Это означает, что внимание на уровне узлов повторяется K раз и соединяет изученные вложения для получения следующего представления признаков на выходе:

$$\mathbf{n}_i^l [\mathcal{N}_i] = \sigma \left(\frac{1}{K} \sum_{k=1}^K \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{n,k} W_n^k \mathbf{n}_j^l \right). \quad (4.9)$$

Аналогично, выход признаков узла i для соседей по ребрам может быть выражен как:

$$\mathbf{n}_i^l [\mathcal{E}_i] = \sigma \left(\frac{1}{K} \sum_{k=1}^K \sum_{ij \in \mathcal{E}_i} \alpha_{i,ij}^{e,k} W_e^k \mathbf{e}_{ij}^l \right). \quad (4.10)$$

Следовательно, вложение узла i в слой $(l+1)$ получается путем комбинирования вложения соседей по ребрам $\mathbf{n}_i^{l+1} [\mathcal{E}_i]$ и вложения соседей по узлам $\mathbf{n}_i^{l+1} [\mathcal{N}_i]$ как:

$$\mathbf{n}_i^{l+1} = \text{concat} (\mathbf{n}_i^l [\mathcal{N}_i], \mathbf{n}_i^l [\mathcal{E}_i]) \quad (4.11)$$

где \mathbf{n}_i^{l+1} - обновленное вложение для узла i с механизмом внимания на уровне узла на l -м уровне.

4.3.3 Слой внимания на уровне ребра

Признаки рёбер улучшают вложения узлов в слое внимания на уровне узлов. Аналогично, в слое внимания на уровне ребер используется тот же подход для получения вложений ребер путем слияния характеристик узлов. Для обновления вложений ребер на начальном этапе необходимо получить информацию о значимости соседних узлов и рёбер для каждого ребра.

Важность узла j или ребра ij по отношению к узлу i может быть выражена следующим образом:

$$\mathbf{c}_{ij,i}^n = \text{Att}_{edge}^n (W_e \mathbf{e}_{ij}, W_n \mathbf{n}_i), \quad (4.12)$$

$$\mathbf{c}_{ij,ik}^e = \text{Att}_{edge}^e (W_e \mathbf{e}_{ij}, W_e \mathbf{e}_{ik}), \quad (4.13)$$

Коэффициент важности ребра ij к узлу i нормируется с помощью функции softmax как

$$\beta_{ij,i}^n = \text{softmax}_i(\mathbf{c}_{ij,i}^n) = \left(\frac{\exp(\sigma(q_n^T [W_e \mathbf{e}_{ij} \| W_n \mathbf{n}_i]))}{\sum_{k \in \mathcal{N}_{ij}} \exp(\sigma(q_n^T [W_e \mathbf{e}_{ij} \| W_n \mathbf{n}_k]))} \right), \quad (4.14)$$

где $q_n \in \mathbb{R}^{d_v^{l+1} + d_e^{l+1}}$ обозначает вектор внимания соседей по узлу на уровне ребра. Коэффициент важности ребра ij по отношению к ребру ik нормируется с помощью функции softmax как

$$\beta_{ij,ik}^e = \text{softmax}_{ij}(\mathbf{c}_{ij,ik}^e) = \left(\frac{\exp(\sigma(q_e^T [W_e \mathbf{e}_{ij} \| W_e \mathbf{e}_{ik}]))}{\sum_{st \in \mathcal{E}_{ij}} \exp(\sigma(q_e^T [W_e \mathbf{e}_{ij} \| W_e \mathbf{e}_{st}]))} \right), \quad (4.15)$$

где $q_e \in \mathbb{R}^{2d_e^{l+1}}$ обозначает вектор внимания соседей на уровне ребра. Тогда вывод признака ребра ij для узловых соседей может быть выражен как

$$\mathbf{e}_{ij}^l[\mathcal{N}_{ij}] = \sigma \left(\frac{1}{K} \sum_{k=1}^K \sum_{i \in \mathcal{N}_{ij}} \beta_{ij,i}^{n,k} W_n^k \mathbf{n}_i^l \right). \quad (4.16)$$

Выходной признак ребра ij для соседей по узлу может быть выражен как

$$\mathbf{e}_{ij}^l[\mathcal{E}_{ij}] = \sigma \left(\frac{1}{K} \sum_{k=1}^K \sum_{ik \in \mathcal{E}_{ij}} \beta_{ij,ik}^{e,k} W_e^k \mathbf{e}_{ik}^l \right). \quad (4.17)$$

Следовательно, вложение ребра ij в слой $(l+1)$ получается путем комбинирования вложения соседей по ребрам $\mathbf{e}_{ij}^l[\mathcal{E}_{ij}]$ и вложения соседей по узлам $\mathbf{e}_{ij}^l[\mathcal{N}_{ij}]$ как

$$\mathbf{e}_{ij}^{l+1} = \text{concat}(\mathbf{e}_{ij}^l[\mathcal{N}_{ij}], \mathbf{e}_{ij}^l[\mathcal{E}_{ij}]) \quad (4.18)$$

Для лучшего понимания процесса агрегирования, в котором участвуют слой внимания на уровне краев и слой внимания на уровне узлов, на рисунке 4.3 представлена краткая иллюстрация. Конечное вложение создаётся путем объединения узловых и краевых вложений. В дальнейшем это конечное вложение может быть использовано в различных задачах, и для этого могут быть разработаны различные функции потерь. Псевдокод предлагаемой схемы приведен в алгоритме 4.

4.3.4 Функция потерь

В процессе прямого вычисления каждый узел i принимает на вход свои начальные признаки $\mathbf{n}_i[0]$ и генерирует вложение \mathbf{n}_i^1 . Если слой $l > 1$, то слой внимания на

Algorithm 4 Heterogeneous Node and Edge Graph Neural Network (HNENN)

Input: Гетерогенный граф $G = (V, E)$

Признаки узлов и рёбер.

Output: Вес внимания на уровне узла α ,

Вес внимания на уровне ребра β ,

Конечное узловое вложение \mathbf{n}_i^l .

while $l \leq L$ **do**

for Каждого узла $i \in V$ **do**

 Выполнить поиск соседних узлов \mathcal{N}_i и соседних рёбер \mathcal{E}_i .

 Выполнить гетерогенное преобразование $\mathbf{n}_i^{attr} = \mathbf{M}_\kappa \mathbf{n}_i^{attr}$.

 Выполнить конкатенацию атрибутов и типов $\mathbf{n}_i = [\mathbf{n}_i^{attr} \parallel \mathbf{n}_i^{type}]$

 Вычислить коэффициент важности α_{ij}^n и α_{ij}^e , найти вложение соседних узлов $\mathbf{n}_i^l[\mathcal{N}_i]$ и соседних рёбер $\mathbf{n}_i^l[\mathcal{E}_i]$.

$\mathbf{n}_i^{l+1} = \text{concat}(\mathbf{n}_i^l[\mathcal{N}_i], \mathbf{n}_i^l[\mathcal{E}_i])$

end for

for Каждого ребра $ij \in E$ **do**

 Выполнить поиск соседних узлов \mathcal{N}_{ij} и соседних рёбер \mathcal{E}_{ij} .

 Выполнить гетерогенное преобразование $\mathbf{e}_{ij}^{attr} = \mathbf{M}_\chi \mathbf{e}_{ij}^{attr}$.

 Выполнить конкатенацию атрибутов и типов $\mathbf{e}_{ij} = [\mathbf{e}_{ij}^{attr} \parallel \mathbf{e}_{ij}^{type}]$.

 Вычислить коэффициент важности $\beta_{ij,i}^n$ и $\beta_{ij,ik}^e$, найти вложение соседних узлов $\mathbf{e}_{ij}^l[\mathcal{N}_{ij}]$ и соседних рёбер $\mathbf{e}_{ij}^l[\mathcal{E}_{ij}]$.

$\mathbf{e}_{ij}^{l+1} = \text{concat}(\mathbf{e}_{ij}^l[\mathcal{N}_{ij}], \mathbf{e}_{ij}^l[\mathcal{E}_{ij}])$

end for

end while

Минимизировать функцию потерь \mathcal{L} .

Выполнить метод обратного распространения ошибки и обновить θ in HNENN

уровне узлов рекурсивно генерирует обновленные признаки \mathbf{n}_i^l путем агрегирования \mathbf{n}_i^{l-1} . Затем \mathbf{n}_i^{l-1} передается на выходной слой для получения оценки вектора формирования луча $\hat{\mathbf{x}}_{i_m} = \mathbf{n}_i^l$. Функция потерь \mathcal{L} определяется как отрицательное ожидание функции полезности при различных реализациях канала:

$$\mathcal{L}(\theta) = -E_H \left[\sum_{i,m} B \log \left(1 + \frac{\|h_{i_m i_m}^H \hat{\mathbf{x}}_{i_m}\|_2^2}{\sum_{j_n \neq i_m} \|h_{i_m j_n}^H \hat{\mathbf{x}}_{j_n}\|_2^2 + \sigma_{i_m}^2} \right) \right]. \quad (4.19)$$

Для обновления параметров модели θ графовой нейронной сети без контроля используется метод обратного распространения, основанный на приведенной выше формулировке. Как следствие, в результате обучения получается графовая нейросетевая модель.

4.4 Моделирование

В этом разделе оценивается производительность HNENN при различных параметрах системы. Для подтверждения эффективности предложенного метода мы сравниваем его с современными базовыми методами, включая методы встраивания гетерогенных сетей и методы на основе графовых нейронных сетей.

4.4.1 Настройка среды

Было проведено моделирование гетерогенной сети D2D. Сеть состоит из двух типов передатчиков: с одной передающей антенной и с двумя передающими антеннами. Кроме того, имеется два типа каналов связи - SISO (Single-Input Single-Output) и MISO (Multiple-Input Single-Output) с соотношением 2:1.

При моделировании передатчики каждой пары D2D генерировались случайным образом с помощью равномерного распределения в пределах квадратной области с длиной стороны D . Соответственно, приемники равномерно распределялись на заданном попарном расстоянии от передатчиков. Вычисление отклика канала формулируется следующим образом: $h_{j_n i_m} = \sqrt{\beta_{j_n i_m}} g_{j_n i_m}$, где $\beta_{j_n i_m}$ - вещественное число, а $g_{j_n i_m}$ - комплексное число, представляющее собой вклад крупномасштабной и мелкомасштабной компонент замираний соответственно. Для отражения эффекта крупномасштабных замираний была использована модель потерь пути в зависимости от расстояния, а для отражения мелкомасштабных замираний - модель замираний Рэля с нулевым средним и единичной дисперсией [32]. В частности, мелкомасштабные замирания описываются независимыми и идентично распределенными комплексными гауссовскими переменными с нулевым средним значением. При крупномасштабном эффекте замираний потери пути и затенение измеряются с помощью модели масштабной корреляции расстояний. Отклик канала представляет собой комплексный вектор. Поэтому его необходимо нормировать, чтобы разделить вещественную и мнимую составляющие. После этого каждый из этих элементов по отдельности добавляется в модель GNN.

Имитационные эксперименты проводились с использованием фреймворка на графическом процессоре NVIDIA RTX 2080 Ti. Была реализована 4-слойная архитектура HNENN. Для обеспечения справедливости были выделены одинаковые обучающие, валидационные и тестовые наборы данных для всех сравниваемых методов. Наборы данных для всех молекулярных сетей были разделены на обуча-

ющих, валидационный и тестовый наборы в соотношении 8:1:1. Моделирование выполнялось три раза, и в каждом эксперименте фиксировалась средняя производительность. Настройки системы и гиперпараметры GNN сведены в таблицу 4.1.

Таблица 4.1: Настройка системы и гиперпараметры GNN

Параметр	Значение
Длина территории D	500
Расстояние между парами D2D $[d_{\min}, d_{\max}]$	[2 m, 5 m]
Мощность передатчика $[p_{\min}, p_{\max}]$	[10 dBm, 30 dBm]
Модель затухания сигнала	$148 + 40 \log_{10}(d [\text{km}])$
Коэффициент обучения	0.0001
Количество слоёв сети GNN	4
Оптимизатор	ADAM
Размер выборки	64
Количество эпох	300

4.4.2 Сравнение производительности

Был сравнен предложенный метод с несколькими методами обучения представлению графов: MPGNN, NENN, нейронной сетью со случайными графами (REGNN) и случайной стратегией (RAND).

- MPGNN (2021) [86]: Представленный подход представляет собой современный метод, основанный на обучении, который опирается на механизм передачи сообщений и специально разработан для гетерогенных сетей.
- NENN (2020) [73]: Предлагаемый подход использует иерархический механизм, включающий в себя внимание как на уровне узлов, так и на уровне ребер, для улучшения встраивания узлов и ребер в нескольких слоях нейронной сети. Он специально разработан для однородных сетей.
- REGNN (2021) [42]: В подходе используются специальные архитектуры REGNN и нелинейная свертка графа с прямым проходом для объединения пространственных весов с коэффициентами каналов. Это позволяет адаптировать его к сетям с различными типами связей.
- RAND: Выбирает переменные в пределах ограничений в соответствии с равномерным распределением.

4.4.3 Численные результаты

Сравнение с различными масштабами сети

Для сравнения производительности различных алгоритмов в более широком диапазоне были заданы параметры в пределах [2m, 50m] с 1600 обучающими значениями. При сохранении постоянной плотности связей каждое удвоение числа связей приводит к увеличению сторон квадратной области в $\sqrt{2}$ раз от предыдущей длины. Как видно из рисунка 4.4, по мере расширения области скорость сети, достигнутая HNENN, неизменно превосходит другие методы. В частности, при увеличении числа связей до 12, 24, 48 и 96 HNENN демонстрирует превосходство в скорости сети, превосходя современную MPGNN на 8,9%, 8,4%, 7,3% и 6,8% соответственно.

Далее была оценена работа различных алгоритмов при различной плотности связей, причем количество связей экспоненциально увеличивается при сохранении фиксированной длины стороны области. Как показано на рисунке 4.5, все алгоритмы демонстрируют снижение производительности по соединений по сравнению с результатами на рисунке 4.4. Это объясняется тем, что в перегруженной среде соединения расположены более близко друг к другу, что приводит к более сильным помехам в канале. При увеличении числа соединений до 12, 24, 48 и 96 улучшение производительности HNENN по сравнению с MPGNN составляет 4,2%, 3,8%, 4,5% и 5,7% соответственно, причем разрыв в производительности увеличивается с ростом плотности соединений.

Сравнение с различными параметрами сети

На рисунке 4.6 показаны результаты сравнения производительности для 48 пар D2D с различным количеством обучающих значений. При числе обучающих значений 200, 400, 800 и 1600 производительность HNENN повышается на 1,8%, 2,4%, 3,2% и 4,5% по сравнению с MPGNN. Поскольку HNENN предназначен для гетерогенных графов, при агрегировании признаков он дополнительно учитывает признаки соседних вершин и ребер как вложенные признаки. Такие признаки с высокой эффективностью выборки очень актуальны для решения практических задач в беспроводных сетях, поскольку получение достаточного количества обучающих значений требует больших вычислительных затрат. Кроме того, HNENN может получить высокую производительность при меньшем количестве обучаю-

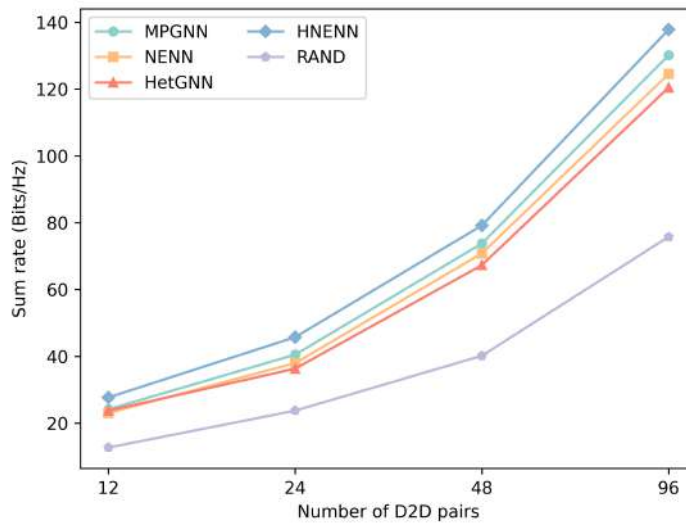


Рис. 4.4: Сравнение производительности при расширении пространственных областей.

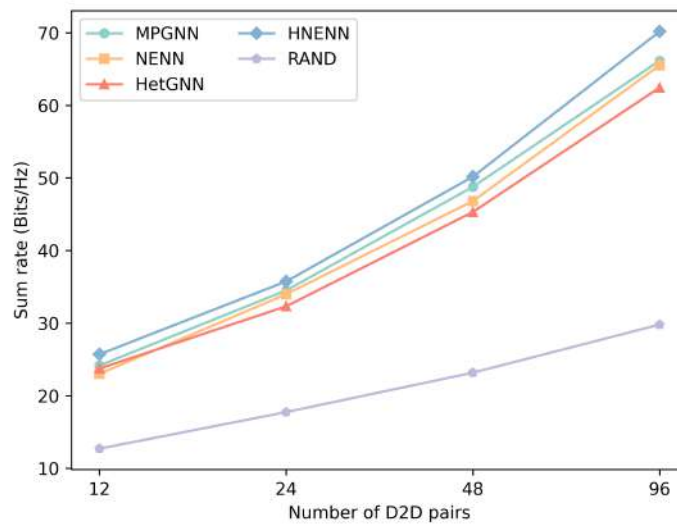


Рис. 4.5: Сравнение производительности при увеличении плотности каналов связи.

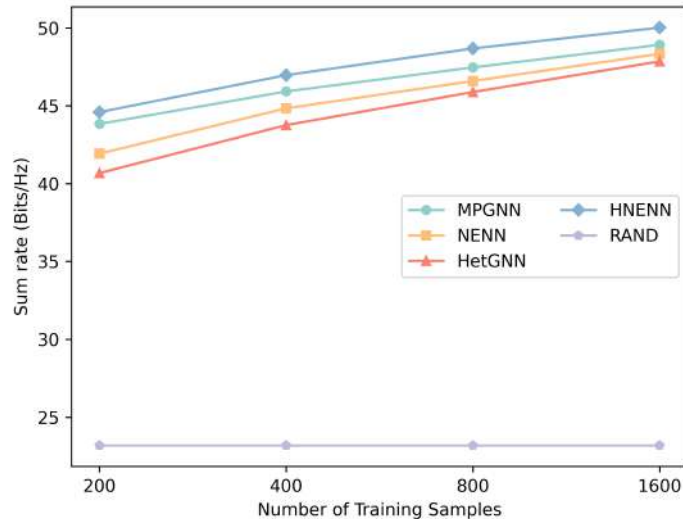


Рис. 4.6: Сравнение производительности при различном количестве обучающих значений.

щих значений, что имеет потенциал для практического применения.

На рисунке 4.7 сравнивается производительность HNENN и эталонных методов при различных расстояниях сопряжения D2D. Примечательно, что даже при различном распределении расстояний D2D-связи производительность HNENN неизменно превосходит эталонные схемы. Это объясняется сильной зависимостью коэффициента усиления канала от расстояния, когда при уменьшении расстояния происходит ослабление коэффициента усиления канала. Следовательно, преимущество HNENN в производительности сохраняется при различных расстояниях между D2D-парами.

Сравнение времени работы

Как видно из рисунка 4.8, предложенный HNENN имеет время работы того же порядка величины, что и другие эталонные алгоритмы, поскольку они имеют схожую структуру сети и входные характеристики. Однако HNENN требует больше времени, чем другие эталонные методы, поскольку он требует дополнительного времени на получение и агрегирование характеристик вложения узлов и ребер, когда каждый узел итеративно обновляет характеристики вложения себя и всех соседних узлов и ребер. При числе связей 96 время работы HNENN составляет 0,002 с, что значительно меньше требуемого времени принятия решений в беспроводных системах (0,02 с) и удовлетворяет потребности в принятии решений в реальном времени. Таким образом, предложенный метод эффективен для реали-

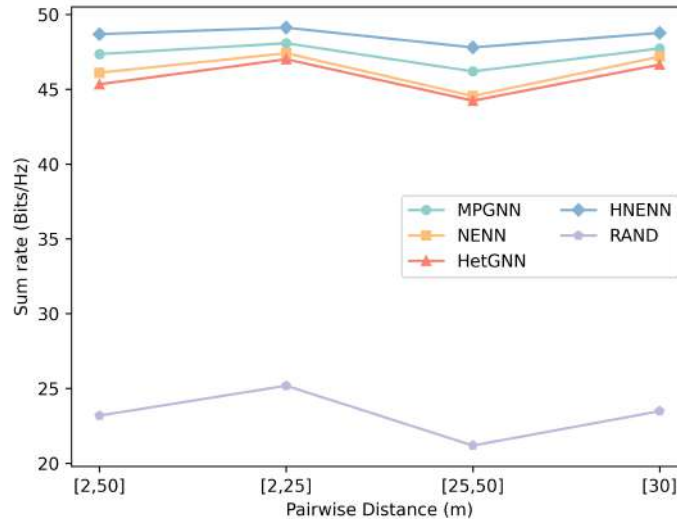


Рис. 4.7: Сравнение производительности при использовании различных парных расстояний.

зации в реальном времени в беспроводных сетях.

4.5 Заключение к главе 4

В данной главе представлена формулировка задачи управления ресурсами беспроводной сети на гетерогенном графе. Граф представляет собой функцию отображения, связывающую характеристики узлов и ребер с переменными, связанными с задачей управления ресурсами. Эти переменные, значения которых еще предстоит определить, задаются на узлах и ребрах графа. Для решения проблемы гетерогенности была предложена модель HNENN. Модель HNENN эффективно объединяет особенности узлов и ребер, что значительно повышает эффективность вложения как узлов, так и ребер в несколько слоев нейронной сети. В ней используется последовательное расположение слоев внимания на уровне узлов и ребер для учета важности соседних узлов и краев. Результаты моделирования показывают, что предложенная схема обеспечивает превосходные показатели суммарной скорости сети. Кроме того, эффективность и устойчивость этой системы были подтверждены в различных сценариях.

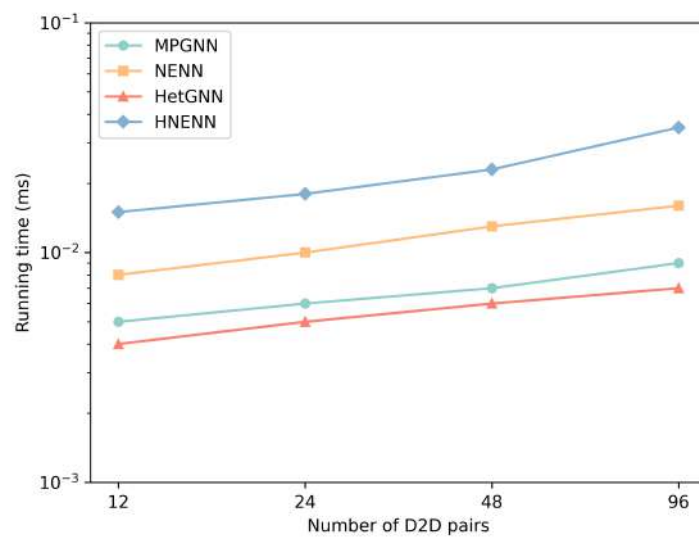


Рис. 4.8: Сравнение времени работы при различном количестве D2D-соединений.

Глава 5

Обучение с подкреплением с использованием теории среднего поля для оптимального распределения ресурсов в гетерогенной сети D2D

Гетерогенные сети и технологии связи D2D стали перспективными решениями для обработки растущего мобильного трафика [46]. Однако экспоненциальный рост пространства состояний и действий в гетерогенных сетях D2D делает традиционные подходы к обучению непрактичными [49]. В этой главе мы предлагаем усовершенствованный подход, сочетающий мультиагентное обучение с подкреплением (MARL) и теорию игр типа среднего поля (MFTG) для приближительного моделирования взаимодействий между различными классами пользователей в гетерогенных D2D-сетях [82]. Проведя обширные эксперименты, было продемонстрировано, что предложенный подход на основе многотипной среднеполевой двойной глубокой Q-сети (MTMF-Q) превосходит по производительности эталонные методы, что подчеркивает его потенциал для практической реализации в сценариях сверхплотных сетей связи.

5.1 Общие сведения о глубоком обучении с подкреплением

Методы глубокого обучения с подкреплением (DRL), в частности MARL, привлекают внимание как эффективные подходы для разработки решений задач распределения ресурсов беспроводных сетей с помощью искусственного интеллекта (ИИ) [47]. Взаимодействие между сетевым субъектом, выступающим в роли агента, и динамическим и непредсказуемым сетевым окружением часто моделируется

как частично наблюдаемый марковский процесс принятия решений (POMDP)[59] для обучения оптимальным стратегиям выбора. Хуанг и др. [58] предложили схему совместного выполнения задач с учетом затрат на основе мультиагентных глубоких детерминированных градиентов политики (MADDPG) для энергосберегающих D2D-сетей с ограниченной емкостью вычислительной техники, сети и аккумуляторов. Эта схема обеспечивает взаимодействие между несколькими мобильными устройствами, потребляющими энергию, для минимизации выполнения задач. В работе Го и др. [62] проблема распределения кредитов решается с помощью многоагентной проксимальной оптимизации политики (MAPPO), которая маргинализирует действия текущего пользовательского оборудования (ПУ), сохраняя действия других ПУ фиксированными. Для обеспечения надежного и стабильного обучения Мседди и др. [61] предложили совместный подход на основе QMIX [63], в котором используются гибридные сети, подчиняющиеся ограничениям монотонности. Эти исследования показывают использование методов DRL и MARL для решения задач распределения ресурсов в беспроводных сетях, демонстрируя их потенциал для достижения эффективных и совместных решений.

Управление сценариями с большим числом агентов сопряжено с определенными трудностями, поскольку классические методы MARL не позволяют работать с большим числом агентов, чем несколько десятков [76]. Масштабируемость многоагентных систем наталкивается на два существенных препятствия: Количество взаимодействий между агентами растет пропорционально квадрату их числа. Это приводит к увеличению вычислительной сложности и усложняет процесс обучения политикам. Сила взаимодействия между агентами различна и меняется со временем, что создает трудности в достижении сходимости функции потерь [80]. Эти препятствия свидетельствуют о сложностях, связанных с увеличением числа агентов в системах MARL, и подчеркивают необходимость разработки новых подходов и методов для эффективного решения этих проблем [79].

Теория игр среднего поля (MFG) является практическим подходом к решению задач с большим числом агентов. В таких сценариях каждый агент взаимодействует с окружающей средой, учитывая свое взаимодействие с коллективным поведением других агентов [65]. Такой подход позволяет существенно снизить вычислительную сложность, связанную с анализом многоагентных систем. Интеграция MFG и MARL демонстрирует большой потенциал для применения в об-

ласти коммуникаций. Ye и др.[48] использовали методы DRL для обучения импульсных нейронных сетей (SNN). Они объединили мультиагентное обучение с подкреплением в среднем поле (MFRL) и импульсную оптимизацию политики близости (S-PPO) для облегчения выбора канала и управления мощностью. Ян и др.[66] предложили согласование кластеров пользователей (UCM), использующее вариации алгоритма усиления пользовательского канала. Они решили задачу распределения ресурсов в сценариях с плотным размещением пользователей с помощью метода градиентной глубокой детерминированной политики среднего поля (MFDDPG). Чен и др. [67] представили метод оптимизации политики доверительного домена среднего поля (MFTRPO), который оптимизирует политику доверительного домена и методы встраивания признаков нейронной сети для распределения энергии с целью максимизации эффективности связи. Важно отметить, что при использовании теории MFG предполагается однородность агентов, что означает, что их свойства должны быть одинаковыми. Однако в реальных сетевых средах часто встречается множество типов агентов с различными атрибутами [81].

Для устранения недостатков MFG и учета неоднородности агентов была предложена более адаптируемая структура игры, получившая название MFTG. В MFTG агенты не обязаны быть однородными, а взаимодействие между агентами и окружающей средой формулируется с учетом взаимодействия между коллективным поведением в рамках одной категории и разных категорий. В области коммуникации MFTG изучалась в различных исследованиях. Например, в работе Zhang et al. [68] предложена гетерогенная структура MFMARL для распределения ресурсов маршрутизации с целью оптимизации энергоэффективности связи в интегрированной сети "пространство-воздух-земля"(SAGI-Net). Различные типы устройств, такие как наземные устройства и беспилотные летательные аппараты (БПЛА), рассматриваются в этой системе как гетерогенные агенты. Ли и др. рассматривают проблему управления мощностью и проектирования траектории, моделируя ее как дискретную MFTG. Они используют алгоритм кластеризации для классификации устройств по различным категориям и предлагают алгоритм среднеполевого Q-обучения (MF-Q) для решения совместной задачи оптимизации. В целом модель MFTG лучше подходит для решения задач оптимизации в гетерогенных сетях, где агенты имеют различные характеристики и поведение.

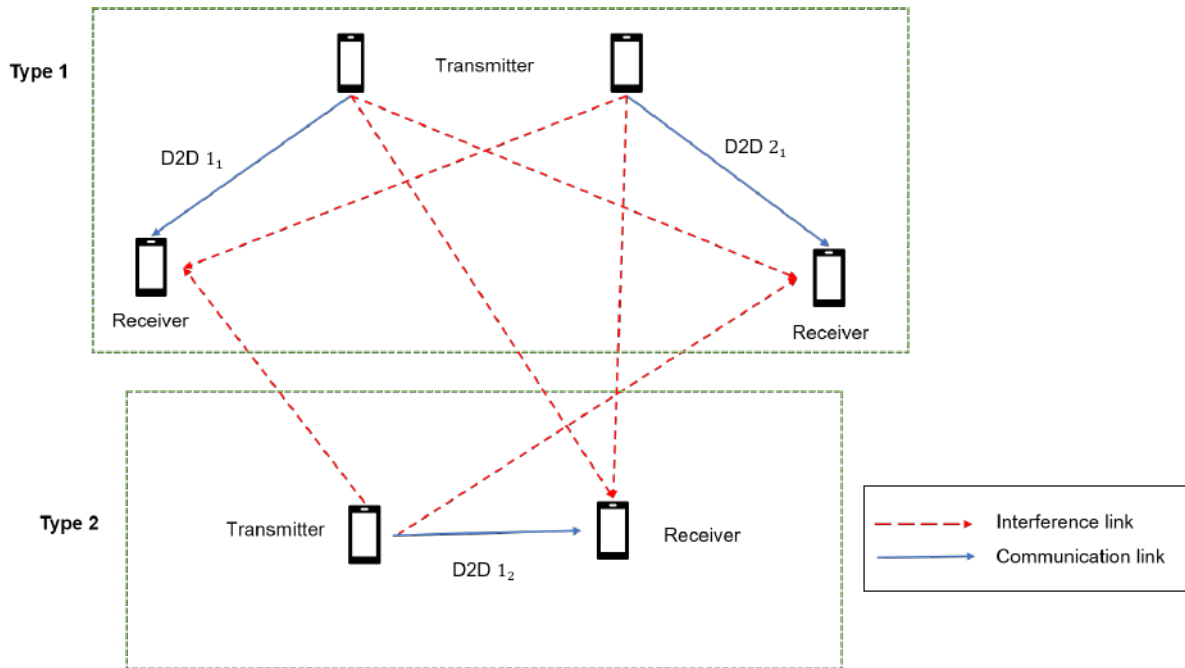


Рис. 5.1: Пример гетерогенной D2D-сети с двумя различными типами связей.

5.2 Системная модель гетерогенной сети D2D

Мы исследуем беспроводную сеть D2D с несколькими типами каналов связи, каждый из которых может обладать различными характеристиками в неоднородной среде. В данном сценарии мы предполагаем, что каждый приемник оснащен одной антенной. Однако важно отметить, что количество передающих антенн может варьироваться в различных каналах связи. Все соединения в системе работают в конфигурации MISO. Следует отметить, что точное количество антенн не стандартизировано и может варьироваться в разных каналах связи.

В квадратной области с длиной ребра d , где каждая пара D2D распределена случайным образом, все пары имеют один и тот же спектр, обозначаемый как B . Описываются типы связей с помощью $\mathcal{M} \triangleq \{1, \dots, M\}$, где каждый тип m имеет различное количество передающих антенн, обозначаемых A_m . Каждый канал в системе индексируется как i_m для i -го канала типа m , причем $i_m \in \mathcal{I}$, а общее число каналов обозначается как I . Отклики канала, $h_{i_m i_m}$, отражают характеристики канала связи между передатчиком и приемником соединения i_m , а $h_{i_m j_n}$ - отклик канала помех от передатчика соединения j_n к приемнику соединения i_m . Эти отклики канала имеют первостепенное значение для определения качества связи и уровня помех в сети. В момент времени t коэффициент независимого канала приемника для соединения i_m выражается как:

$$y_{i_m}(t) = h_{i_m i_m}^H(t) w_{i_m}(t) \sqrt{p_{i_m}(t)} u_{i_m} + \sum_{j_n \neq i_m} h_{i_m j_n}^H(t) w_{j_n}(t) \sqrt{p_{j_n}(t)} u_{j_n} + z_{i_m}, \quad (5.1)$$

где $z_{i_m} \sim \mathcal{N}(0, \sigma_{i_m}^2)$ представляет собой аддитивный белый гауссовский шум (AWGN) u_{i_m} - обозначение сигнала с нулевым средним и единичной дисперсией. Вектор формирования луча, используемый передатчиком канала i_m для передачи сигнала $u_{i_m} \in \mathbb{C}$ на приемник, обозначается как $v_{i_m} \in \mathbb{C}^{A_m}$. p_{i_m} представляет собой мощность передачи канала i_m , и обозначим набор мощностей передачи как $P = \{p_{i_m}\}_{i_m}$. w_{i_m} . Нормированный вектор формирования луча равен w_{i_m} , и обозначим этих нормированных векторов как $W = \{w_{i_m}\}_{i_m}$. Вектор формирования луча может быть выражен как $v_{i_m} = w_{i_m} \sqrt{p_{i_m}}$. На временном интервале t скорость передачи данных к приемнику i типа m формулируется как:

$$\xi_{i_m}(W(t), P(t)) = B \log \left(1 + \frac{\left\| h_{i_m i_m}^H(t) w_{i_m}(t) \sqrt{p_{i_m}(t)} u_{i_m} \right\|_2^2}{\sum_{j_n \neq i_m} \left\| h_{i_m j_n}^H(t) w_{j_n}(t) \sqrt{p_{j_n}(t)} u_{j_n} \right\|_2^2 + \sigma_{i_m}^2} \right), \quad (5.2)$$

Особенно в тех случаях, когда тип соединения - "один вход - один выход" (SISO), проектирование формирования луча упрощается до задачи управления мощностью. Здесь p_{max} представляет собой ограничение на мощность. За период времени T целевая функция задачи оптимизации может быть рассчитана как:

$$\begin{aligned} \max_{w_{i_m}(t), p_{i_m}(t)} \quad & \frac{1}{T} \sum_{t=1}^T \sum_{i_m} \xi_{i_m}(t) (w_{i_m}(t), p_{i_m}(t)) \\ \text{s.t.} \quad & \left\| w_{i_m} \sqrt{p_{i_m}} \right\|_2^2 \leq p_{max}, \forall i \in \mathcal{I}, m \in \mathcal{M}. \end{aligned} \quad (5.3)$$

Основной целью исследования является оптимизация политики формирования луча и распределения мощности для достижения оптимальной суммарной скорости в гетерогенной сети D2D. Суммарная скорость означает общую достижимую скорость передачи данных в сети. Для обеспечения корректного сравнения и анализа была выполнена нормализация скорости передачи данных, деля ее на пропускную способность канала, в результате чего получаем единицу измерения бит в секунду на герц. Решение этой оптимизационной задачи имеет решающее значение для повышения общей производительности и эффективности сети.

5.3 Формулировка задачи распределения ресурсов

Задача оптимизации, сформулированная в предыдущем разделе, представляет собой задачу целочисленного нелинейного программирования, которая является NP задачей, трудно оптимизируемой с точки зрения подходов полиномиальной временной сложности.

5.3.1 Частично наблюдаемые марковские процессы принятия решений

В децентрализованной архитектуре несколько одноагентных систем объединяются в многоагентную систему. При увеличении масштаба системы обмен информацией может привести к значительной перегрузке сети, что делает её нецелесообразной. Была использована многоагентная система обучения с подкреплением (MARL) с частично наблюдаемыми марковскими процессами принятия решений (POMDP) для моделирования проблем распределения спектра и мощности, направленных на максимизацию суммарной скорости для сети связи D2D. POMDP с I агентами может быть описана кортежем:

$(\mathcal{S}, \mathcal{A}^1, \dots, \mathcal{A}^I, r^1, \dots, r^I, \mathcal{P}, \gamma)$, где \mathcal{S} обозначает пространство состояний, которое является конечным множеством состояний, $\mathcal{A}^i, i \in \{1, \dots, I\}$ - пространство действий агента i , r^i - функция вознаграждения агента i , \mathcal{P} - распределение переходов.

В этой системе каждая пара D2D функционирует как независимый обучающийся агент, каждый из которых имеет свою стратегию принятия решений. Эти агенты постоянно взаимодействуют с динамической средой, которая обеспечивает обратную связь относительно эффективности политики агентов на основе их действий в прошлом. Состояние эволюции среды и функция вознаграждения, получаемая каждым агентом, зависят от коллективных действий всех агентов. Следовательно, при адаптации и оптимизации своей стратегии каждый агент должен учитывать не только состояние среды, но и взаимодействие с другими обучающимися агентами.

5.3.2 Состояние окружающей среды

Понятие состояния среды является исчерпывающим описанием состояния системы и полностью отражает состояние системы на любом временном шаге. Состояние среды в POMDP - это целое, неограниченное состояние, содержащее все воз-

возможные состояния. Однако в POMDP система D2D должна оценивать или выводить текущее состояние среды на основе наблюдений, а не непосредственно наблюдать его.

Пусть \mathcal{S} представляет собой полное, не ограниченное пространство состояний состояния среды. Вероятность перехода из одного состояния в другое целесообразно выразить в виде $\mathcal{P}(s_{t+1}, \mathbf{a}, s_t) = Pr(s' | s, \mathbf{a})$, где s - текущее состояние шага, s' - состояние следующего шага для совместного действия \mathbf{a} .

5.3.3 Пространство наблюдений

Как было сказано выше, каждому агенту доступна только локальная информация о глобальной среде. На временном шаге t наблюдение агента i обозначается через:

$$O^i(t) = (p_i(t), w_i(t), \xi_i(t), \eta_i(t), \varpi_i(t)), \quad (5.4)$$

где $\eta_i(t)$ обозначает коэффициент усиления канала, $\varpi_i(t)$ обозначает сумму мощностей помех и шумов. Соответственно, оба обозначения выражаются как:

$$\eta_i(t) = \left| h_{ii}^H(t) w_i(t) \sqrt{p_i(t)} \right|^2, \quad (5.5)$$

$$\varpi_i(t) = \sum_{j \neq i} \left| h_{ij}^H(t) w_j(t) \sqrt{p_j(t)} \right|^2 + z_i. \quad (5.6)$$

Каждая пара D2D не имеет априорных знаний о распределении мощности, формировании луча и достижимых скоростях других пар D2D. Поэтому D2D не могут предпринимать действия, направленные на максимизацию суммарной скорости сети. В условиях динамичности состояния канала и неопределенности помех в канале эта задача POMDP может быть сформулирована как стохастическая игра, в которой каждый D2D выбирает взаимно независимые векторы формирования луча и стратегии распределения мощности передачи как функцию своих собственных локальных наблюдений.

5.3.4 Пространство действий

Пространство действий определяет пределы или границы набора действий, которые может выбрать агент при взаимодействии с окружающей средой. В качестве пространства действий каждого агента рассматриваются конкретные параметры формирования луча и мощности передачи. В отличие от традиционных

схем управления мощностью на основе алгоритмов, которые принимают непрерывные значения для удовлетворения практических ограничений схемы и облегчения обучения, в данной работе используются дискретные уровни мощности в диапазоне от 0 до p_{max} . Мощность передачи количественно выражается как уровень L . Набор доступных дискретных мощностей задаётся множеством:

$$\left\{ \frac{P_{max}}{L}, \frac{2p_{max}}{L}, \dots, P_{max} \right\}. \quad (5.7)$$

Как уже отмечалось выше, размерность вектора формирования луча определяется числом передающих антенн. Таким образом, размер пространства действий агентов с типом m составляет $|\mathcal{A}| = A_m \times L$, и D2D выбирает соответствующее направление луча и уровень мощности и немедленно получает вознаграждение за оценку своих действий.

5.3.5 Функция вознаграждения

Функция вознаграждения в RL строится на основе целевой функции задачи оптимизации. Каждый агент принимает совместное решение о направлении луча и уровне мощности для максимизации коэффициента полезного действия системы. Целью обучения всей мультиинтеллектуальной системы является поиск оптимальной стратегии, максимизирующей кумулятивно-дисконтную функцию вознаграждения (cumulative discount reward function), и мы определяем долгосрочное вознаграждение r_i для временного шага t как кумулятивно-дисконтное вознаграждение:

5.4 Многотипное среднеполевое многоагентное обучение с подкреплением

В многоагентных средах широко используются игры со средним полем (Mean Field Game), основная концепция которых строится на агрегированной информации других агентов. Вместо MFG, которая требует, чтобы все агенты выполняли однородные требования, теория MFTG больше подходит для реальных сценариев инженерных приложений. Агенты в сценарии MFTG могут быть разнородными, и учитывается как влияние на состояние среднего поля, так и общий эффект одного агента. В этой части предложенная задача представлена в виде многотипной среднеполевой формулировки MARL (MTMFRL).

5.4.1 Многотипная формула среднего поля

Агенты, соседствующие с агентом i , могут быть разделены на M типов. Каждый агент относится к определенному типу, и каждый тип состоит из X_m агентов. Мы предполагаем, что функция Q (функция "действие-значение") может быть дополнительно разложена на подмножества, основанные на разделении агентов, причем каждое подмножество содержит агентов одного типа. Пусть $a_m^{k_i}$ представляет собой действие агента k_i , принадлежащего к типу m и взаимодействующего с агентом j . Предположим, что

$$Q^j(s, \mathbf{a}) = Q^j\left(s, a^j, a_1^{k_1}, \dots, a_1^{k_{X_1}}, \dots, a_M^{k_1}, \dots, a_M^{k_{X_M}}\right), \quad (5.8)$$

может быть разложено в среднее значение взаимодействий в рамках одного типа агентов, которое разлагается далее по парам. Такое разложение можно рассматривать как обобщение попарного разложения на множество типов в MFRL, поскольку каждая компонента зависит от представления каждого типа. Таким образом, стандартная функция Q формулируется как:

$$Q^j(s, \mathbf{a}) = \sum_{m=1}^M Q^j\left(s, a^j, a_m^{k_1}, \dots, a_m^{k_{X_m}}\right) = \sum_{m=1}^M \frac{1}{X_m} \sum_{i=1}^{X_m} Q^j\left(s, a^j, a_m^{k_i}\right). \quad (5.9)$$

5.4.2 Многотипная среднеполевая аппроксимация

Были использованы одноточечные представления, предложенные в Yang et al. Для действий действие агента k , принадлежащего к типу m , можно представить как $a_m^k = \bar{a}_m^j + \delta^{j,k}$, где \bar{a}_m^j обозначает среднее действие смежных агентов j , а $\delta^{j,k}$ - разность между действием агента и средним действием его подмножеств. Пусть $\bar{a}_m^j = \frac{1}{X_m} \sum_{i=1}^{X_m} a_m^{k_i}$, теорема о среднем поле позволяет аппроксимировать функцию Q аппроксимацией Тейлора в виде:

$$Q^j(s, \mathbf{a}) \approx Q^j\left(s, \bar{a}_1^j, \bar{a}_2^j, \dots, \bar{a}_M^j\right). \quad (5.10)$$

Это иерархическое среднее поле подчеркивает взаимодействие между типами, поскольку агенты одного типа имеют одинаковые свойства, и их взаимодействие может быть оценено с помощью стандартного среднего поля. Мы модифицировали функцию среднего поля Q , включив в нее конечное число типов, каждому из которых соответствует свое среднее поле, с учетом того, что агенты разных

типов демонстрируют большую вариативность. Такая аппроксимация позволяет уменьшить сложность взаимодействия агентов и дисперсию функции Q , неявно сохраняя глобальные взаимодействия между любыми двумя агентами.

5.4.3 Многотипное обновление среднего поля

На временном интервале t функция среднего поля Q для агента j может быть обновлена итеративно:

$$Q_{t+1}^j \left(s, \bar{a}_1^j, \dots, \bar{a}_M^j \right) = (1 - \alpha) Q_t^j \left(s, \bar{a}_1^j, \dots, \bar{a}_M^j \right) + \alpha \left[r^j + \gamma V_t^j (s') \right], \quad (5.11)$$

где α обозначает скорость обучения. Функция среднего поля стоимости агента j на временном шаге t оценивается как:

$$V_t^j (s') = \sum_{a^j} \pi_t^j \left(a^j \mid s, \bar{a}_1^j, \dots, \bar{a}_M^j \right) \mathbb{E}_{a^{-j} \sim \pi_t^{-j}} \left[Q_t^j \left(s', a^j, \bar{a}_1^j, \dots, \bar{a}_M^j \right) \right], \quad (5.12)$$

где π_t^j представляет собой политику действий агента j . Политика Больцмана выбирает стратегию для каждого агента j :

$$\pi_t^j \left(a^j \mid s, \bar{a}_1^j, \dots, \bar{a}_M^j \right) = \frac{\exp \left(\varphi Q_t^j \left(s, a^j, \bar{a}_1^j, \dots, \bar{a}_M^j \right) \right)}{\sum_{a^{j'} \in A^j} \exp \left(\varphi Q_t^j \left(s, a^{j'}, \bar{a}_1^j, \dots, \bar{a}_M^j \right) \right)}, \quad (5.13)$$

где φ - гиперпараметр Больцмана softmax. С помощью итерационных уравнений происходит постоянное обновление средних действий всех агентов и соответствующих им стратегий. В конечном итоге она сходится к фиксированной точке, близкой к равновесию Нэша. Сходимость была доказана в.

5.4.4 Многотипное среднеполевое решение

Для оценки функции Q с многотипным средним полем (MTMF) предлагается алгоритм на основе DDQN. Схема алгоритма представлена на рисунке 5.2, DDQN использует две нейронные сети для обеспечения стабильности работы сети. Целевая сеть Q с параметром θ^- имеет ту же структуру сети, что и обучающая сеть Q с параметром θ . θ обновляется на каждом шаге, но θ^- обновляется до значения θ на каждом из нескольких шагов, после чего значения параметров фиксируются на оставшихся шагах.

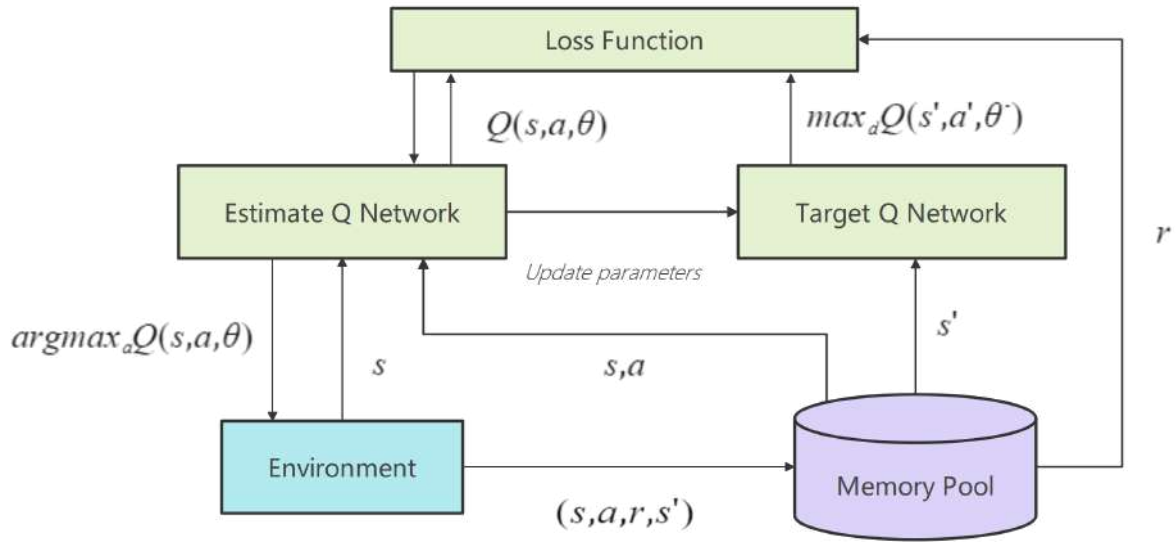


Рис. 5.2: Иллюстрация многоагентной структуры DDQN.

Воспроизведение опыта повышает эффективность обучения за счет повторного использования обучающих данных и минимизации корреляции последовательных данных. На каждом временном шаге t , после того как данные об опыте перехода $e_t = (s_t, \mathbf{a}_t, \mathbf{r}_{t+1}, s_{t+1})$ собраны агентом i в память воспроизведения, каждый агент DDQN равномерно выбирает из памяти мини-выборку опыта e и обновляет веса своей сети Q , используя стохастический градиентный спуск. Функция потерь формулируется следующим образом:

$$\mathcal{L}_{\theta^j} = (y^j - Q^j(s, \mathbf{a}, \theta))^2, \quad (5.14)$$

где $\mathbf{a} = [a^j, \bar{a}_1^j, \dots, \bar{a}_M^j]$ обозначает среднее совместное действие поля, и

$$y_t^j = r_{t+1}^j + \gamma Q^j \left(s^{t+1}, \arg \max_{\mathbf{a}'} Q^j(s^{t+1}, \mathbf{a}', \theta), \theta^- \right), \quad (5.15)$$

обозначает среднее значение поля, вычисленное при обновлении целевой сети. Оно используется для расчета ошибки временной разницы (T.D.). Таким образом, градиент функции потерь рассчитывается как:

$$\nabla_{\theta^j} \mathcal{L}_{\theta^j} = 2 (y^j - Q^j(s, \mathbf{a}, \theta)) \times \nabla_{\theta^j} Q^j(s, \mathbf{a}, \theta). \quad (5.16)$$

Алгоритм 5 описывает шаг алгоритма MTMF-Q.

Algorithm 5 Многотипная среднеполевая Q сеть с двойной глубиной.

Инициализировать память воспроизведения D .

Initialize target Q and estimate Q network with parameters θ^- and θ .

for Эпизод = $\{1, \dots, E\}$ **do do**

for Шаг = $\{1, \dots, T\}$ **do do**

 Выбрать действие a_t^j согласно Q -значению жадным методом ε для каждого агента.

 Вычислить новое среднее действие $\bar{a}_1^j, \dots, \bar{a}_M^j$ для каждого типа.

 Выполнить объединённое действие и получить награду r_t^j и следующее состояние s_{t+1}^j .

 Сохранить кортеж $\langle s_t, \mathbf{a}, r, s_{t+1} \rangle$ в память воспроизведения.

 Выбрать случайную мини-выборку из D .

 Выполнить градиентный спуск по функции потерь θ в оценке Q .

 Установить y_t^i

 Выполнить обновление $\theta^- = \theta$ каждые C шагов

end for

end for

5.5 Моделирование

Для анализа и оценки эффективности предложенного алгоритма было выполнено сравнение его с MF-Q, DDQN и случайной стратегией в одинаковых условиях.

5.5.1 Настройка среды

В данном разделе мы моделируем сценарий работы одной ячейки гетерогенной беспроводной сети D2D, рассматривая два типа D2D-соединений: 1) SISO-связи и 2) 2×1 MISO-связи, случайно распределенные в круговой области радиусом 500 м. Максимальная мощность передачи P_{max} и мощность шума установлены на уровне 30 дБм и -174 дБм соответственно. Расстояние связи каждого канала ограничено в пределах от $d_{min} = 2$ м до $d_{max} = 50$ м. Крупномасштабные замирания моделируются как $\beta = -120,9 - 37,6 \log_{10}(d) + 10 \log_{10}(z)$ с помощью телекоммуникационного стандарта Long-Term Evolution (LTE)[77]. Мелкомасштабные замирания представлены комплексной гауссовской переменной с нулевым средним и единичной дисперсией. Используя модель масштабирования в зависимости от расстояния в [73], рассчитываются потери пути и затенение, вызванные масштабными замираниями. В экспериментах, оценивающих эффективность предложенного метода при различных параметрах системы, соотношение каналов SISO и MISO устанавливается равным 1:1, а топология сети для каждого образца D2D определяется независимо. Гиперпараметры DDQN сведены в таблицу 5.1

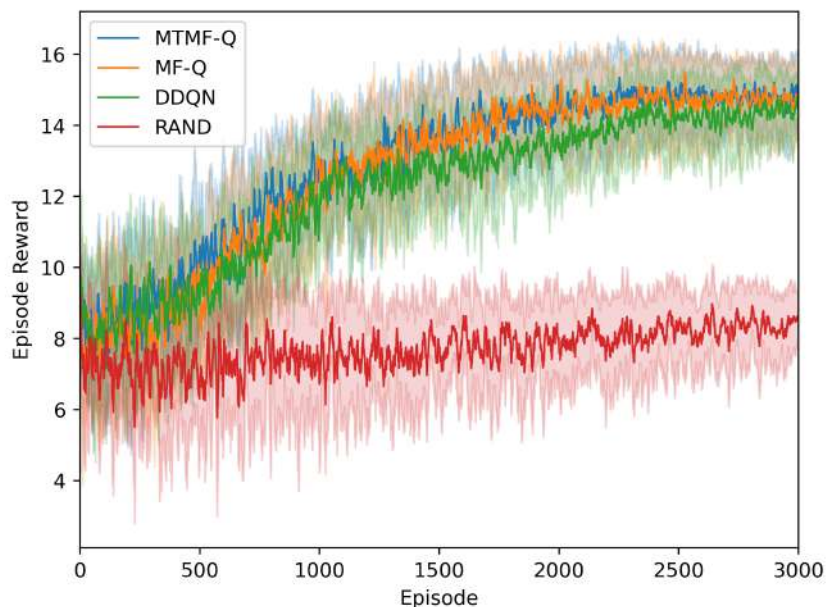


Рис. 5.3: Производительность обучения с 8 агентами.

Таблица 5.1: Гиперпараметры DDQN

Параметр	Значение
Количество эпизодов	3000
Шаг по времени на один эпизод	50
Размер мини-выборки	32
Размер шага памяти	500
Скорость обучения	0.001
Коэффициент затухания скорости обучения	0.0001
Коэффициент дисконтирования	0.9
Скрытые слои	5
Активационная функция	ReLU

5.5.2 Сравнение производительности

Для оценки эффективности предложенного алгоритма MTMF-Q в гетерогенных сетях было проведено исследование с использованием различного количества D2D-каналов. Рисунок 5.3, 5.4, 5.5, 5.6 иллюстрирует вознаграждение за эпизод, полученное на этапе обучения, при этом случайная стратегия рассматривается как нижняя граница для целей сравнения. Важно отметить, что в проведённых экспериментах ни один из алгоритмов не учитывал распределение кредитов или меж-агентное взаимодействие. Каждый агент при обновлении своих действий опира-

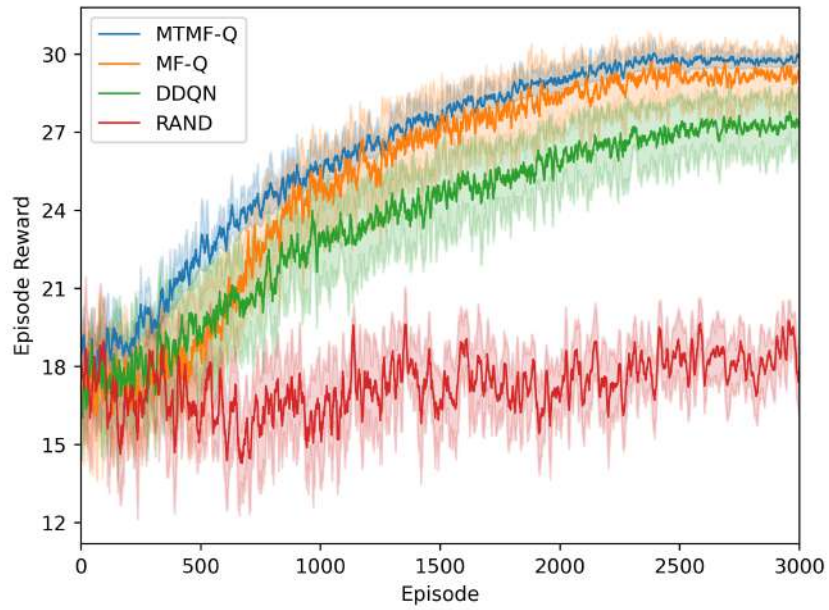


Рис. 5.4: Производительность обучения с 16 агентами.

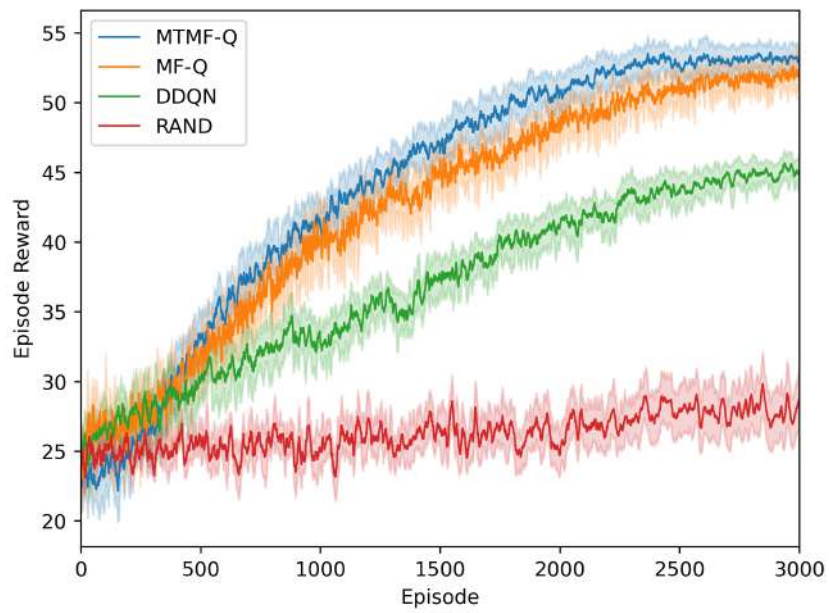


Рис. 5.5: Производительность обучения с 32 агентами.

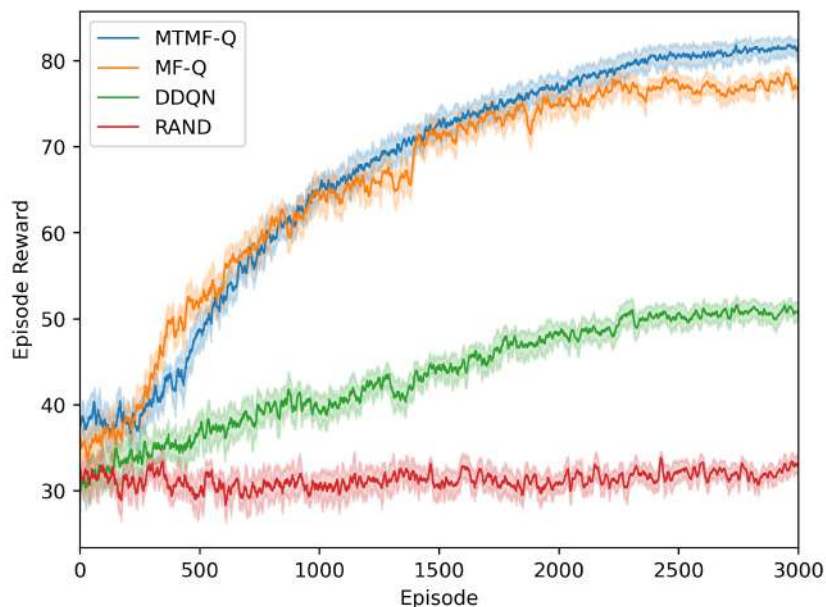


Рис. 5.6: Производительность обучения с 64 агентами.

ся исключительно на локальные наблюдения. Каждый алгоритм был запущен с различными начальными случайными семенами (seeds) и сглаживание кривых на каждом графике было произведено для улучшения читаемости.

При малых размерах сети производительность различных классов алгоритмов RL оказывается сопоставимой. Однако при увеличении размера сети алгоритм DDQN оказывается хуже подхода, основанного на MF, и сталкивается с проблемами сходимости. Это может быть связано с ограниченностью информации о наблюдениях, доступной каждому агенту, что снижает эффективность обновления информации отдельными агентами. С другой стороны, алгоритм на основе MF использует среднее поле двух классов агентов (MISO, SISO) в качестве фактора принятия решений. Значения Q , используемые для выбора действий, зависят от внутренних изменений параметров действий. Примечательно, что алгоритм MTMF-Q превосходит стандартный алгоритм MFQ, причем разрыв в производительности между ними увеличивается по мере роста числа D2D. Подход MTMF-Q превосходно справляется с многотипным средним полем разнородных агентов с различными пространствами действий, что обусловлено его основой в теории MFTG. Эффективность предложенного алгоритма подтверждается стабильным и быстрым ростом вознаграждения, наблюдаемым в крупномасштабных сетях агентов.

Рисунок 5.7 дает представление о производительности сети D2D в зависимости от различных уровней мощности. Варьировался уровень мощности сети D2D, со-

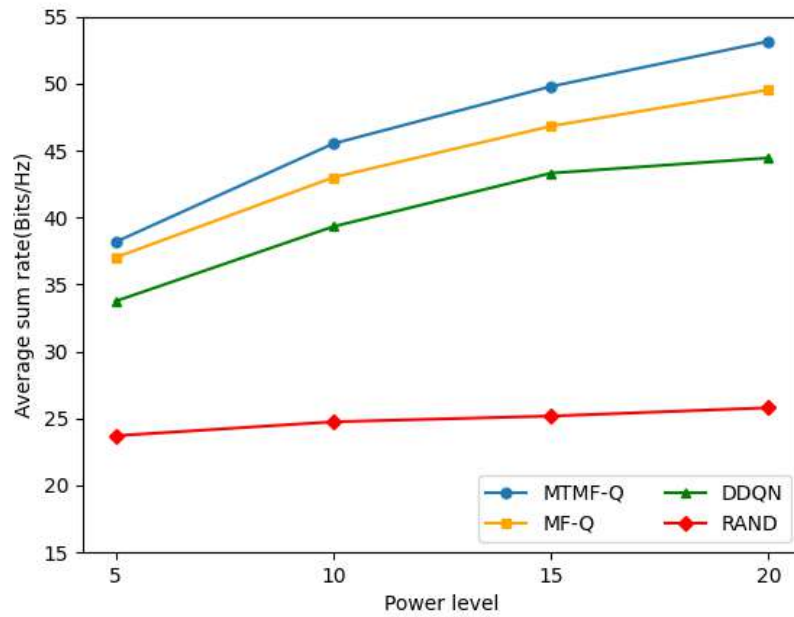


Рис. 5.7: Общая скорость сети в зависимости от уровня мощности.

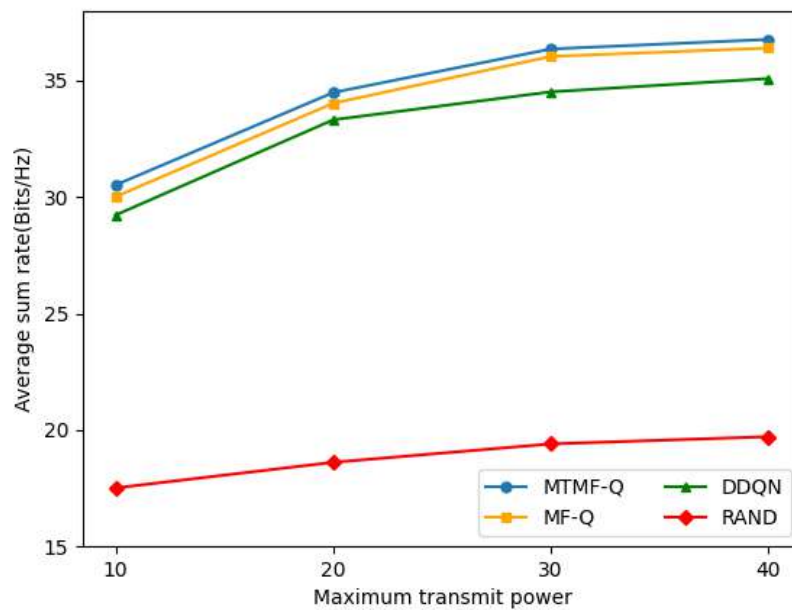


Рис. 5.8: Суммарная скорость сети в зависимости от максимальной мощности передачи.

храня остальные параметры фиксированными ($D2D=32$), чтобы оценить эффективность предложенного метода в большом пространстве действий. Когда пространство действий невелико, существенной разницы в производительности между подходом на основе MF и DDQN не наблюдается. Однако при увеличении уровня мощности, что приводит к увеличению пространства действий, метод на основе MF достигает более высокой средней награды. Примечательно, что алгоритм MTMF-Q по-прежнему превосходит другие алгоритмы по улучшению суммарного коэффициента сети.

На рисунке 5.8 показана суммарная скорость сети D2D в зависимости от различных максимальных мощностей передачи. Кривая демонстрирует медленный рост в диапазоне от 10 до 30 дБм и стабилизируется в диапазоне от 30 до 40 дБм. В диапазоне от 10 до 30 дБм постепенный рост кривой можно объяснить ограниченной общей достижимой скоростью, обусловленной низкой доступной мощностью передачи. При увеличении мощности передачи в диапазоне от 30 до 40 дБм, хотя для D2D-каналов становится доступной более высокая мощность передачи, помехи от других устройств, заложенные в функцию вознаграждения суммарной скорости, также увеличиваются. Следовательно, предельный выигрыш от увеличения мощности при улучшении суммарной скорости значительно снижается. В целом алгоритм MTMF демонстрирует эффективность и устойчивость при различных параметрах сети.

Для решения указанных выше проблем была предложена новая графовая нейронная сеть, получившая название Heterogeneous Edge Feature Enhanced Graph Attention Networks (HEGAT). HEGAT учитывает как гетерогенность графа, так и особенности его краев. Он объединяет особенности узлов и ребер на основе двухслойного механизма внимания, включающего внимание на уровне узлов и на уровне ребер. В частности, необходимо знать, насколько важны соседи по узлам и ребрам и как каждый узел вписывается в слой внимания на уровне узлов. Аналогично, встраивание каждого ребра формируется в слое внимания на уровне ребра.

5.6 Заключение к главе 5

Данная глава посвящена проблемам формирования луча и управления мощностью в гетерогенных D2D-сетях с несколькими типами каналов связи с целью максими-

зации суммарной скорости сети. При увеличении числа и типов D2D-устройств вычисление вознаграждения для каждого устройства в отдельности приводит к значительному снижению производительности, что нежелательно при применении существующих методов MFRL. Для решения этой проблемы мы предлагаем усовершенствованный метод MFRL под названием MTMF-Q, который объединяет теорию MFTG с MARL в гетерогенных сетях для достижения оптимального совместного формирования луча и распределения мощности. Алгоритм MTMF-Q разработан для устранения ограничений традиционных методов MARL в сценариях с большим числом агентов. Учитывая взаимодействие между каждым агентом и различными типами средних полей, алгоритм позволяет эффективно распределять ресурсы и преодолевать проблемы масштабируемости, связанные с вычислением индивидуального вознаграждения. Экспериментальные оценки показали, что MTMF-Q превосходит современный метод MF-Q в крупномасштабных сценариях. Более того, предложенный алгоритм демонстрирует стабильную работу в сетях с различными параметрами, что повышает его эффективность при практическом развертывании.

В заключение следует отметить, что предложенный подход способен преодолеть ограничения традиционных методов MARL, позволяя развертывать сверхплотные сети связи D2D. В дальнейших исследованиях планируется изучить сети связи с большим разнообразием типов устройств, приближаясь к реальным сценариям. Кроме того, планируется исследовать применение MF-MARL для распределения других коммуникационных ресурсов, что позволит еще больше повысить практическую ценность и применимость алгоритма.

Заключение и дальнейшие работы

Эта глава служит заключительным разделом диссертации, в ней обобщаются результаты исследований и вклад, сделанный в ходе работы над диссертацией, а также намечаются возможные направления будущих исследований.

5.7 Заключение

Исследования в основном сосредоточены на двух ключевых методах в беспроводных сетях: 1) эвристические алгоритмы и 2) машинное обучение. Работа организована вокруг темы распределения ресурсов, а выводы по каждой главе представлены ниже. Основные результаты работы заключаются в следующем:

1. Глава 1 знакомит с основами и методами распределения ресурсов в сотовых сетях. Проблема, рассматриваемая в данной работе, не имеет замкнутых выражений для переменных оптимизации, что делает эвристические алгоритмы подходящим методом. Имитационные эксперименты демонстрируют преимущества метаэвристических алгоритмов при распределении ресурсов и управлении помехами. Однако для достижения удовлетворительных решений они часто требуют многочисленных стохастических переборов, что может не удовлетворять требованиям сетей связи к принятию решений в реальном времени.
2. В главе 2 исследуется проблема проектирования формирования луча в более сложных сетях D2D. Разработана система контролируемого обучения на основе PSO для решения проблем непрерывной оптимизации в беспроводных сетях. Этот фреймворк имеет надежные возможности обучения остаточных сетей и быструю скорость обработки данных глубокого обучения. Кроме того, представлен рабочий процесс для решения общих проблем распределения ресурсов, включающий генерацию размеченных наборов данных с помощью методов оптимизации и их автономное обучение с использованием

методов машинного обучения. Глубокие нейронные сети обладают значительным потенциалом для переноса трудоемких вычислений на менее вычислительно сложное обучение и решения сложных, трудно моделируемых задач. Тем не менее, они требуют значительных массивов данных для лучшего обучения отображения политики распределения ресурсов.

3. В главе 3 рассматривается система обучения с учителем обучения на основе графовых нейронных сетей, основанная на схеме, представленной в главе 2. Модели DNN, обучаемые в автономном режиме, заменяются на MPGNN и GAT, эффективно использующие информацию, заложенную в топологии коммуникационной сети. Экспериментальные результаты показывают, что субоптимальная производительность модели может быть достигнута при использовании всего нескольких обучающих образцов.
4. В главе 4 исследуется проблема распределения ресурсов в более сложных, гетерогенных беспроводных сетях D2D. Разработан алгоритм обучения без контроля, названный HEGAT, для совместной оптимизации стратегий формирования луча и распределения мощности. Эта модель учитывает различные типы коммуникационных сетей и информацию о рёбрах, а также использует механизм внимания для определения наиболее критичных рёбер и узлов. Представленный в этой главе подход глубокого обучения является универсальным и превосходит по производительности лучшие методы оптимизации ресурсов, основанные на обучении графов.
5. Глава 5 посвящена проблеме распределения ресурсов в беспроводных сетях D2D с масштабной гетерогенностью. Методы, основанные на машинном обучении, предложенные в предыдущих главах, концентрируются на принятии решений, что зачастую невыполнимо для реальных приложений в крупномасштабных сетях. Комбинация игр среднего поля и многоагентного обучения с подкреплением используется для децентрализованного принятия решений при распределении ресурсов D2D с ограниченной информацией о состоянии канала (CSI). Этот подход демонстрирует потенциал для работы в больших сетях и обработки частично отсутствующих входных признаков.

Эти подходы, основанные на машинном обучении, позволяют значительно улучшить распределение радиоресурсов, включая выполнение в реальном времени,

масштабируемость, обобщение и децентрализованную реализацию.

5.8 Будущие работы

На основе работы, представленной в данной статье, можно выделить несколько перспективных направлений будущих исследований, которые кратко описаны ниже:

5.8.1 Дальнейшее расширение подхода GNN

Предложенный метод GNN в настоящее время реализуется централизованно для решения проблемы распределения беспроводных ресурсов. Однако централизованные подходы могут вызывать проблемы с конфиденциальностью. Результаты моделирования продемонстрировали превосходные возможности GNN, включая масштабируемость для крупномасштабных сетей, обобщение для различных настроек системы без переобучения и устойчивость к поврежденным входным характеристикам. Чтобы сохранить эти желаемые характеристики и решить проблемы конфиденциальности, необходимо изучить возможность распределенного развертывания GNN. Такой подход больше подходит для приложений беспроводной связи, чем централизованное развертывание. Одним из возможных направлений является объединение GNN с MARL для достижения этой цели.

5.8.2 Дальнейшее расширение подхода RL

Предложенный в данной статье подход RL служит предварительным шагом к реализации распределения ресурсов для крупномасштабных сетей D2D. Некоторые аспекты могут быть исследованы далее, как описано ниже:

1. Предложенный метод RL в настоящее время работает под наблюдением, опираясь на размеченные наборы данных для обучения. Воспроизведение опыта хранит данные в виде кортежей в массиве опыта, но не полностью использует топологическую информацию коммуникационной сети, что приводит к неоптимальной эффективности обучения. Изучение того, как эффективно использовать топологическую информацию коммуникационных сетей, является перспективным направлением исследований.

2. Масштабируемость метода RL может быть улучшена. Модель RL в данной работе рассматривает только относительно небольшое пространство состояний действий, что требует дискретизации для обучения модели и совместной оптимизации стратегий формирования луча и распределения мощности. Полезно изучить децентрализованный подход RL, способный работать с более широким спектром сетевых ресурсов, таких как память, планирование каналов и задержки. Кроме того, еще одним перспективным направлением исследований является изучение крупномасштабных пространств состояний действий.

Литература

- [1] Fooladivanda D., Rosenberg C. Joint resource allocation and user association for heterogeneous wireless cellular networks // *IEEE Transactions on Wireless Communications*. 2012. Vol. 12. N. 1. P. 248–257.
- [2] Shen K., Yu W. Fractional programming for communication systems—Part I: Power control and beamforming // *IEEE Transactions on Signal Processing*. 2018. Vol. 66. N. 10. P. 2616–2630.
- [3] Shi Q., Razaviyayn M., Luo Z., He C. An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel // *IEEE Transactions on Signal Processing*. 2011. Vol. 59. N. 9. P. 4331-4340.
- [4] Zhang X., Nakhai M. R., Zheng, G., Lambotharan S., Ottersten B. Calibrated learning for online distributed power allocation in small-cell networks // *IEEE Transactions on Signal Processing*. 2019. Vol. 67. N. 11. P. 8124–8136.
- [5] Sun H., Chen X., Shi Q., Hong M., Fu X., Sidiropoulos N. D. Learning to optimize: Training deep neural networks for interference management // *IEEE Transactions on Signal Processing*. 2018. Vol. 66. N. 20. P. 5438-5453.
- [6] Eisen M., Zhang C., Chamon L. F., Lee D. D., Ribeiro A. Learning optimal resource allocations in wireless systems // *IEEE Transactions on Signal Processing*. 2019. Vol. 67. N. 10. P. 2775-2790.
- [7] Sun Q., Wu H., Petrosian, O. Optimal Power Allocation Based on Metaheuristic Algorithms in Wireless Network // *Mathematics*. 2022. Vol. 10. N. 18. P. 3336-.
- [8] Kumar M., Sharma, S. C. PSO-based novel resource scheduling technique to improve QoS parameters in cloud computing // *Neural Computing and Applications*. 2020. Vol. 32. P. 12103–12126.

- [9] Khanolkar S., Sharma N., Anpalagan A. Energy-Efficient Resource Allocation in Underlay D2D Communication using ABC Algorithm // *Wireless Personal Communications*. 2022. Vol. 12. N. 5. P. 1443–1468.
- [10] Sun H., Chen X., Shi Q., Hong M., Fu X., Sidiropoulos N. D. Learning to optimize: Training deep neural networks for interference management // *IEEE Transactions on Signal Processing*. 2018. Vol. 66. N. 20. P. 5438–5453.
- [11] Blough D. M., Resta G., Santi P. Approximation algorithms for wireless link scheduling with SINR-based interference // *IEEE/ACM Transactions on networking*. 2020. Vol. 18. N. 20. P. 5438–5453.
- [12] Chowdhury A., Verma G., Rao C., Swami A., Segarra S. Unfolding WMMSE using graph neural networks for efficient power allocation // *IEEE Transactions on Wireless Communications*. 2021. Vol. 20. N. 9. P. 6004–6017.
- [13] Zhang X., Zhao H., Xiong J., Liu X., Zhou L., Wei J. Scalable power control/beamforming in heterogeneous wireless networks with graph neural networks // *IEEE Global Communications Conference (GLOBECOM)*. 2021. P. 01–06.
- [14] Zhang X., Zhao H., Xiong J., Liu X., Zhou L., Wei J. Scalable power control/beamforming in heterogeneous wireless networks with graph neural networks // *Innovations in multi-agent systems and applications-1*. 2010. P. 183–221.
- [15] Li T., Zhu K., Luong N. C., Niyato D., Wu Q., Zhang Y., Chen B. Applications of multi-agent reinforcement learning in future internet: A comprehensive survey // *IEEE Communications Surveys and Tutorials*. 2021. Vol. 24. N. 2. P. 1240–1279.
- [16] Rouwet W. *Open Radio Access Network (O-RAN) Systems Architecture and Design* // Elsevier. 2022
- [17] Li Y., Han S., Yang C. Multicell power control under rate constraints with deep learning // *IEEE Transactions on Wireless Communications*. 2021. Vol. 20. N. 12. P. 7813–7825.
- [18] Marini F., Walczak B. Particle swarm optimization (PSO). A tutorial. // *Chemometrics and Intelligent Laboratory Systems*. 2015. Vol. 149. P. 153–165.

- [19] Li J. Y., Zhan Z. H., Liu R. D., Wang C., Kwong S., Zhang J. Generation-level parallelism for evolutionary computation: A pipeline-based parallel particle swarm optimization // *IEEE Transactions on Cybernetics*. 2020. Vol. 51. N. 10. P. 4848–4859.
- [20] Schluter M., Egea J. A., Banga J. R. Extended ant colony optimization for non-convex mixed integer nonlinear programming // *Computers and Operations Research*. 2008. Vol. 36. N. 7. P. 2217–2229.
- [21] Price K. V. Differential evolution. In *Handbook of optimization: From classical to modern approach* // *Handbook of optimization: From classical to modern approach*. 2013. P. 187–214.
- [22] Elsayed S. M., Sarker R. A., Essam D. L. Differential evolution with multiple strategies for solving CEC2011 real-world numerical optimization problems // *IEEE Congress of Evolutionary Computation (CEC)*. 2011. P. 1041–1048.
- [23] Brest J., Greiner S., Boskovic B., Mernik M., Zumer V. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems // *IEEE transactions on evolutionary computation*. 2006. Vol. 10. N. 6. P. 646–657.
- [24] Corana A., Marchesi M., Martini C., Ridella S. Minimizing multimodal functions of continuous variables with the “simulated annealing” algorithm—Corrigenda for this article is available here // *ACM Transactions on Mathematical Software*. 1987. Vol. 13. N. 3. P. 262–280.
- [25] Wales D. J., Doye J. P. Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms // *The Journal of Physical Chemistry A*. 1997. Vol. 101. N. 28. P. 5111–5116.
- [26] Hansen N. The CMA evolution strategy: a comparing review // *Towards a new evolutionary computation: Advances in the estimation of distribution algorithms*. 2006. P. 75–102.
- [27] Ansari R. I., Chrysostomou C., Hassan S. A., Guizani M., Mumtaz S., Rodriguez J., Rodrigues J. J. 5G D2D networks: Techniques, challenges, and future prospects // *IEEE Systems Journal*. 2017. Vol. 12. N. 4. P. 3970–3984.

- [28] Li L., Xu Y., Yin J., Liang W., Li X., Chen W., Han Z. Deep reinforcement learning approaches for content caching in cache-enabled D2D networks // *IEEE Internet of Things Journal*. 2019. Vol. 7. N. 1. P. 544–557.
- [29] Ruan L., Dias M. P. I., Wong E. Deep neural network supervised bandwidth allocation decisions for low-latency heterogeneous e-health networks // *Journal of Lightwave Technology*. 2019. Vol. 37. N. 16. P. 4147–4154.
- [30] Huang H., Liu M., Gui G., Gacanin H., Sari H., Adachi F. Unsupervised learning-inspired power control methods for energy-efficient wireless networks over fading channels // *IEEE Transactions on Wireless Communications*. 2022. Vol. 21. N. 11. P. 9892–9905.
- [31] Jiang Y., Liu Q., Zheng F., Gao X., You X. Energy-efficient joint resource allocation and power control for D2D communications // *IEEE Transactions on Vehicular Technology*. 2015. Vol. 65. N. 8. P. 6119–6127.
- [32] Sun H., Chen X., Shi Q., Hong M., Fu X., Sidiropoulos N. D. Learning to optimize: Training deep neural networks for interference management // *IEEE Transactions on Signal Processing*. 2015. Vol. 66. N. 20. P. 5438–5453.
- [33] Su L., Ji Y., Wang P., Liu F. Learning to optimize: Training deep neural networks for interference management // *IEEE wireless communications and networking conference (WCNC)*. 2013. P. 129–133.
- [34] Lu L., He D., Li G. Y., Yu X. Graph-based robust resource allocation for cognitive radio networks // *IEEE Transactions on Signal Processing*. 2015. Vol. 63. N. 14. P. 3825–3836.
- [35] Xu M. Understanding graph embedding methods and their applications // *SIAM Review*. 2021. Vol. 63. N. 4. P. 825–853.
- [36] Zhang S., Tong H., Xu J. Graph convolutional networks: a comprehensive review // *Computational Social Networks*. 2019. Vol. 6. N. 1. P. 1–23.
- [37] Velickovic P., Cucurull G., Casanova A., Romero A., Lio P., Bengio Y. Graph attention networks // *stat*. 2017. Vol. 1050. N. 20. P. 10–48550.

- [38] Wu Z., Pan S., Chen F., Long G., Zhang C., Philip S. Y. A comprehensive survey on graph neural networks // *IEEE transactions on neural networks and learning systems*. 2020. Vol. 32. N. 1. P. 4–24.
- [39] Gilmer J., Schoenholz S. S., Riley P. F., Vinyals O., Dahl G. E. Neural message passing for quantum chemistry // *International conference on machine learning*. 2017. P. 1263–1272.
- [40] Shen Y., Shi Y., Zhang J., Letaief K. B. Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis // *IEEE Journal on Selected Areas in Communications*. 2020. Vol. 39. N. 1. P. 101–115.
- [41] Eisen M., Ribeiro A. Optimal wireless resource allocation with random edge graph neural networks // *IEEE Transactions on signal Processing*. 2020. Vol. 68. P. 2977–2991.
- [42] Chowdhury A., Verma G., Rao C., Swami A., Segarra S. Efficient power allocation using graph neural networks and deep algorithm unfolding // *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021 P. 4725–4729.
- [43] Schlichtkrull M., Kipf T. N., Bloem P., Van Den Berg R., Modeling relational data with graph convolutional networks // *Semantic Web: 15th International Conference*. 2018. P. 593–607.
- [44] Gong L., Cheng Q. Exploiting edge features for graph neural networks // *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019. P. 9211–9219.
- [45] Guo J., Yang C. Learning power control for cellular systems with heterogeneous graph neural network // *IEEE Wireless Communications and Networking Conference (WCNC)*. 2021. P. 1–6.
- [46] Feng D., Lu L., Yuan-Wu Y., Li G. Y., Li S., Feng G. Device-to-device communications in cellular networks // *IEEE Communications Magazine*. 2014. Vol. 52. N. 4. P. 49–55.

- [47] Zhang L., Xiao M., Wu G., Alam M., Liang Y. C., Li S. A survey of advanced techniques for spectrum sharing in 5G networks // *IEEE Wireless Communications*. 2017. Vol. 24. N. 5. P. 44–51.
- [48] Ye P. G., Liang W., Lu Q., Xiao R. F., Guo Z. Y., Sun K. X. Spiking mean field multi-agent reinforcement learning for dynamic resources allocation in D2D networks // *International Conference on Ubiquitous Computing and Communications*. 2021. P. 60–67.
- [49] Xu Y., Gui G., Gacanin H., Adachi F. A Survey on Resource Allocation for 5G Heterogeneous Networks: Current Research, Future Trends and Challenges // *IEEE Communications Surveys and Tutorials*. 2021. Vol. 23. N. 2. P. 668–695.
- [50] Jiang Y., Zou Y., Guo H., Tsiftsis T. A., Bhatnagar M. R., de Lamare R. C., Yao Y. D. Joint power and bandwidth allocation for energy-efficient heterogeneous cellular networks // *IEEE Transactions on Communications*. 2021. Vol. 67. N. 9. P. 6168–6178.
- [51] Abrardo A., Moretti M. Distributed power allocation for D2D communications underlying/overlying OFDMA cellular networks // *IEEE Transactions on Wireless Communications*. 2016. Vol. 16. N. 3. P. 1466–1475.
- [52] Yu C. H., Doppler K., Ribeiro C. B., Tirkkonen O. Resource sharing scheme for device-to-device communication underlying cellular networks // *IEEE transactions on communications*. 2015. Vol. 63. N. 12. P. 4838–4848.
- [53] Li T., Zhu K., Luong N. C., Niyato D., Wu Q., Zhang Y., Chen B. Applications of multi-agent reinforcement learning in future internet: A comprehensive survey // *IEEE Communications Surveys and Tutorials*. 2022. Vol. 24. N. 2. P. 1240–1279.
- [54] Barik P. K., Shukla A., Datta R., Singhal C. A resource sharing scheme for intercell D2D communication in cellular networks: A repeated game theoretic approach // *IEEE Transactions on Vehicular Technology*. 2020. Vol. 69. N. 7. P. 7806–7820.
- [55] Lee W., Lee K. Resource allocation scheme for guarantee of QoS in D2D communications using deep neural network // *IEEE Communications Letters*. 2020. Vol. 25. N. 3. P. 887–891.

- [56] Li L., Xu Y., Yin J., Liang W., Li X., Chen W., Han Z. Deep reinforcement learning approaches for content caching in cache-enabled D2D networks // IEEE Internet of Things Journal. 2019. Vol. 7. N. 1. P. 554–557.
- [57] Yan Y., Zhang B., Li C., Su C. Cooperative caching and fetching in d2d communications-a fully decentralized multi-agent reinforcement learning approach // IEEE Transactions on Vehicular Technology. 2019. Vol. 69. N. 12. P. 16095–16109.
- [58] Yan Y., Zhang B., Li C., Su C. Cooperative caching and fetching in d2d communications-a fully decentralized multi-agent reinforcement learning approach // IEEE Transactions on Vehicular Technology. 2019. Vol. 69. N. 12. P. 16095–16109.
- [59] Tang Q., Xie R., Yu F. R., Huang T., Liu Y. Decentralized computation offloading in IoT fog computing system with energy harvesting: A dec-POMDP approach // IEEE Internet of Things Journal. 2020. Vol. 7. N. 6. P. 4898–4911.
- [60] Guo D., Tang L., Zhang X., Liang Y. C. Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning // IEEE Transactions on Vehicular Technology. 2020. Vol. 69. N. 11. P. 13124–13138.
- [61] Mseddi A., Jaafar W., Moussaid A., Elbiaze H., Ajib W. Collaborative D2D pairing in cache-enabled underlay cellular networks // IEEE Global Communications Conference (GLOBECOM). 2021. P. 1–6.
- [62] Mseddi A., Jaafar W., Moussaid A., Elbiaze H., Ajib W. Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning // IEEE Transactions on Vehicular Technology. 2020. Vol. 69. N. 11. P. 13124–13138.
- [63] Rashid T., Samvelyan M., De Witt C. S., Farquhar G., Foerster J., Whiteson S. Monotonic value function factorisation for deep multi-agent reinforcement learning // The Journal of Machine Learning Research. 2020. Vol. 21. N. 1. P. 7234–7284.
- [64] Li L., Cheng Q., Tang X., Bai T., Chen W., Ding Z., Han Z. Resource allocation for NOMA-MEC systems in ultra-dense networks: A learning aided mean-field

- game approach // IEEE Transactions on Wireless Communications. 2020. Vol. 20. N. 3. P. 1487–1500.
- [65] Yang Y., Luo R., Li M., Zhou M., Zhang W., Wang J. Resource allocation for NOMA-MEC systems in ultra-dense networks: A learning aided mean-field game approach // International conference on machine learning. 2018. P. 5571–5580.
- [66] Yang C., Li J., Semasinghe P., Hossain E., Perlaza S. M., Han Z. Distributed interference and energy-aware power control for ultra-dense D2D networks: A mean field game // IEEE Transactions on Wireless Communications. 2016. Vol. 16. N. 2. P. 1205–1217.
- [67] Chen D., Qi Q., Zhuang Z., Wang J., Liao J., Han Z. Mean field deep reinforcement learning for fair and efficient UAV control // IEEE Internet of Things Journal. 2016. Vol. 8. N. 2. P. 813–828.
- [68] Zhang H., Tang H., Hu Y., Wei X., Wu C., Ding W., Zhang X. P. Heterogeneous Mean-Field Multi-Agent Reinforcement Learning for Communication Routing Selection in SAGI-Net // IEEE 96th Vehicular Technology Conference. P. 1–5.
- [69] Lixin L., Yan S., Cheng Q., Dawei W, Wensheng L, Wei C. Optimal trajectory and downlink power control for multi-type UAV aerial base stations // Chinese Journal of Aeronautics. 2021. Vol. 34. N. 9. P. 11–23.
- [70] Shen Y., Shi Y., Zhang J., Letaief K. B. Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis // IEEE Journal on Selected Areas in Communications. 2020. Vol. 39. N. 1. P. 101–115.
- [71] Subramanian S. G., Poupart P., Taylor M. E., Hegde N. Multi type mean field reinforcement learning // arXiv preprint arXiv:2002.02513. 2020.
- [72] Guo X., Hu A., Xu R., Zhang J. Learning mean-field games // Advances in Neural Information Processing Systems. 2019. Vol. 32.
- [73] Shen Y., Shi Y., Zhang J., Letaief K. B. Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis // IEEE Journal on Selected Areas in Communications. 2020. Vol. 39. N. 1. P. 101–115.

- [74] Djehiche B., Tcheukam A., Tembine H. Mean-field-type games in engineering // arXiv preprint arXiv:1605.03281. 2016.
- [75] Chen T., Zhang X., You M., Zheng G., Lambotharan S. Chen T, Zhang X, You M, et al. A GNN-based supervised learning framework for resource allocation in wireless IoT networks // IEEE Internet of Things Journal. 2021. Vol. 9. N. 3. P. 1712–1724.
- [76] Alsharif M. H., Kelechi A. H., Albreem M. A., Chaudhry S. A., Zia M. S., Kim S. Sixth generation (6G) wireless networks: Vision, research activities, challenges and potential solutions // Symmetry. 2020. Vol. 12. N. 4. P. 676.
- [77] Frias Z., Perez J. Techno-economic analysis of femtocell deployment in long-term evolution networks // EURASIP journal on wireless communications and networking. 2012. P. 1–15.
- [78] Zhang K., Yang Z., Başar T. Multi-agent reinforcement learning: A selective overview of theories and algorithms // Handbook of reinforcement learning and control. 2021. P. 321–384.
- [79] Van Hasselt H., Guez A., Silver D. Deep reinforcement learning with double q-learning // Proceedings of the AAAI conference on artificial intelligence. 2021. Vol. 30. N. 1.
- [80] Chen T., Zhang X., You M., Zheng G., Lambotharan S. Chen T, Zhang X, You M, et al. A deeper look at experience replay // arXiv preprint arXiv:1712.01275. 2017.
- [81] Guo H., Liang Y. C., Chen J., Larsson E. G. Weighted sum-rate maximization for intelligent reflecting surface enhanced wireless networks // 2019 IEEE Global Communications Conference (GLOBECOM). 2019. P. 1–6.
- [82] Zhao J., Liu Y., Chai K. K., Chen Y., Elkashlan M., Alonso-Zarate J. A NOMA-based D2D communications: Towards 5G // IEEE global communications conference (GLOBECOM). 2021. Vol. 9. N. 3. P. 1712–1724.
- [83] LeCun Y., Bengio Y., Hinton G. Deep learning // Nature. 2015. Vol. 521. N. 7553. P. 436–444.

- [84] Sun Q and Zhao C and Petrosian O and Li Y. Power allocation in wireless cellular networks: stochastic algorithm-based approach // Management processes and sustainability. 2022. Vol. 9. N. 1. P. 357–362.
- [85] Sun Q., Zhang Y., Wu H., Petrosian O. Resource Allocation in Heterogeneous Network with Supervised GNNs // International Conference on Swarm Intelligence. 2023. P. 350–361.
- [86] Chowdhury A., Verma G., Rao C., Swami A., Segarra S. Unfolding WMMSE using graph neural networks for efficient power allocation // IEEE Transactions on Wireless Communications. 2021. Vol. 20. N. 9. P. 6004–6017.
- [87] Qiushi S., Yang H., Petrosian O. Graph attention network enhanced power allocation for wireless cellular system // Informatics and Automation. 2024. Vol. 23. N. 1. P. 259–283.
- [88] Sun Q., Zhang Y., Wu H., Petrosian O. L. Deep Neural Network based Resource Allocation in D2D Wireless Networks // Vestnik of Saint Petersburg University. Applied Mathematics. Computer Science. Control Processes. 2023. Vol. 19. N. 4. P. 411–421.
- [89] Wang X., Ji H., Shi C., Wang B., Ye Y., Cui P., Yu P. S. Heterogeneous graph attention network // The world wide web conference. 2019. P. 2022–2032.
- [90] Velickovic P., Cucurull G., Casanova A., Romero A., Lio P., Bengio Y. Graph attention networks. // stat. 2017. Vol. 1050. N. 20.
- [91] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Polosukhin I. Attention is all you need. // Advances in neural information processing systems. 2017. Vol. 30.