

Федеральное государственное автономное образовательное учреждение
высшего образования «Российский государственный университет нефти и газа
(национальный исследовательский университет) имени И.М. Губкина»



На правах рукописи

Мухтаров Амир Амангельдыевич

**Разработка моделей и методов оптимизации
проектирования коммуникационных сетей нефтяных
месторождений**

Специальность 05.13.06 —

«Автоматизация и управление технологическими процессами и
производствами»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
доктор технических наук, профессор
Першин Олег Юрьевич

Москва — 2021

Оглавление

Стр.

Введение	4
Глава 1. Оформление различных элементов	8
1.1 Расчет дальности действия связи	8
Глава 2. Оптимальное размещение базовых станций широкополосной беспроводной сети связи для обслуживания заданного множества рассредоточенных объектов	11
2.1 Задача при заданных местах размещения станций.	12
2.2 Оптимизационная задача выбора набора размещаемых станций и определения мест их размещения	14
2.3 Пример решения задачи	18
2.4 Результаты численного эксперимента	19
2.5 Выводы к главе 2	21
Глава 3. Математические модели синтеза топологии сети для охвата линейного участка в виде задачи целлочисленного линейного программирования	22
3.1 Постановка задачи	22
3.2 Модель ЦЛП	22
3.3 Численный пример	27
3.3.1 Расчет радиуса связи между станциями	28
3.3.2 Расчет радиуса покрытия	29
3.4 Выводы по Главе 3	30
Глава 4. Математические модели синтеза топологии сети для охвата линейного участка в виде экстремальной задачи в комбинаторной форме	31
4.1 Постановка задачи и ее формулировка в экстремальной комбинаторной форме	31
4.2 Дерево ветвлений для перебора элементов в множестве Γ	33
4.2.1 Параметр для разбиения множеств на подмножества	33

	Стр.
4.2.2 Движение по дереву ветвлений	35
4.2.3 Алгоритм метода ветвей и границ	36
4.2.4 Построения последовательности топологий для итерационной процедуры моделирования БШС	40
4.2.5 Расчет межконцевой задержки	41
4.2.6 Выводы по Главе 4	42
Заключение	43
Список сокращений и условных обозначений	44
Словарь терминов	46
Список литературы	47
Список рисунков	48
Список таблиц	49
Приложение А. Примеры вставки листингов программного кода	50
Приложение Б. Очень длинное название второго приложения, в котором продемонстрирована работа с длинными таблицами	56
Б.1 Подраздел приложения	56
Б.2 Ещё один подраздел приложения	58
Б.3 Использование длинных таблиц с окружением <i>longtabu</i>	62
Б.4 Форматирование внутри таблиц	65
Б.5 Стандартные префиксы ссылок	67
Б.6 Очередной подраздел приложения	68
Б.7 И ещё один подраздел приложения	68
Приложение В. Чертёж детали	69

Введение

В настоящее время тенденция бурного развития информационных технологий во всех сферах деятельности человека оказывает весомое на развитие нефтегазовый сектор страны. Современные компании, представляющие собой сложную многоуровневую производственно-технологическую систему в силу своего устойчивого развития требуют постоянного движения в направлении развития технологий. Нефтегазовая отрасль России является ключевым сектором топливно-энергетическим комплексом страны. Особенностью данной отрасли является масштабы объектов управления, наличие больших объемов информации, высокие требования к безопасности и надежности. Сегодня наблюдается этап бурного развития «цифровизации». Лидеры крупнейших международных нефтегазовых компаний имеют подразделения, задачами которых является разработка и реализация в дальнейшем принципов интеллектуального месторождения: «Умные месторождения» («Smart Fields») в компании Shell, «Месторождение будущего» («Field of the Future») в компании BP и «iFields» в компании Chevron и др. Данное развитие нефтегазового комплекса предусматривает переход к малолюдным системам управления добычи, транспортировки и переработки сырья. Основными информационными технологиями являются: большие данные (англ. Big Data), искусственные нейронные сети (англ. Artificial Neural Network – ANN), системы распределенного реестра (англ. Blockchain), промышленные интернет вещей (англ. Industrial internet of things – IIoT), технологии виртуальной и дополненной реальности (англ. Virtual Reality – VR), мониторинг распределенных объектов беспилотными летательными аппаратами БПЛА (англ. Unmanned Aerial Vehicle – UAV). Большой объем передачи информации привел к еще одной из наиболее интересных тенденций цифрового развития – внедрения беспроводных технологий. Современные месторождения сегодня, помимо данных первичного сбора и обработки информации технологических параметров основных производственных объектов содержат также колоссальный объем информации мультимедийного трафика. Сюда входят данные БПЛА по обнаружению утечек и разрушения трубопроводов; камер видеонаблюдений; а также большой поток данных цифровых двойников, аналитики и т.д. В совокупности со всеми вышеизложенными

перспективными направлениями беспроводные технологии являются неотъемлемой частью «цифровизации» месторождения.

Отсюда возникает научно - техническая проблема организации распределенной сети связи, соответствующая реальным требованиям современного производства.

Степень разработанности темы. Степень разработанности.

Объектом исследования в данной работе являются беспроводные широкополосные сети.

Предметом исследования является синтез топологической структуры беспроводной широкополосной сети.

Цель диссертационного исследования состоит в разработке моделей и методов задачи оптимального размещения базовых станций беспроводной широкополосной сети.

Для достижения поставленной цели необходимо было решить следующие задачи:

1. анализ состояния, основных проблем и перспектив развития современных инфраструктур систем коммуникаций по беспроводным каналам на месторождениях;
2. разработка моделей задач размещения базовых станций в рамках комплексного проектирования сетей коммуникаций мониторинга объектов нефтегазовых месторождений;
3. разработка моделей оценки характеристик производительности сетей связи;

Научная новизна результатов исследования заключается в следующем:

1. разработаны модели задачи размещения базовых станций на плоскости и для частного случая с линейной топологией;
2. разработаны модели имитационного моделирования для оценки характеристик производительности сети;
3. разработаны модели прогнозирования оценок характеристик производительности с помощью методов машинного обучения.

Практическая значимость ...

Методология и методы исследования. ...

Основные положения, выносимые на защиту:

1. Первое положение
2. Второе положение

3. Третье положение

4. Четвертое положение

В папке Documents можно ознакомиться с решением совета из Томского ГУ (в файле Def_positions.pdf), где обоснованно даются рекомендации по формулировкам защищаемых положений.

Достоверность полученных результатов обеспечивается ... Результаты находятся в соответствии с результатами, полученными другими авторами.

Апробация работы. Основные положения и результаты исследования представлены и обсуждены на научных конференциях «Губкинский университет в решении вопросов нефтегазовой отрасли России» (Москва, 17-21 сентября 2018); «13-е Всероссийское совещание по проблемам управления» (Москва, 17-20 июня 2019); «International Conference on Distributed Computer and Communication Networks: Control, Computation, Communications» (Москва, 22-27 сентября 2019), «Губкинский университет в решении вопросов нефтегазовой отрасли России» (Москва, 24-26 сентября 2019); «Управление развитием крупномасштабных систем» (Москва, 1-3 октября 2019); «Information and Telecommunication Technologies and Mathematical Modeling of High-Tech Systems» (Москва, 13-17 апреля 2020); «Computer-aided technologies in applied mathematics» (Томск, сентябрь 2020); «International Conference on Distributed Computer and Communication Networks: Control, Computation, Communications» (Москва, 14-18 сентября 2020); «Information and Telecommunication Technologies and Mathematical Modeling of High-Tech Systems» (Москва, 19-23 апреля 2021);

Личный вклад. Все основные научные положения диссертационного исследования разработаны автором лично.

Публикации. Основные результаты по теме диссертации изложены в 12 печатных изданиях, 1 из которых изданы в журналах, рекомендованных ВАК, 2 — в периодических научных журналах, индексируемых Web of Science и Scopus, 9 — в сборниках трудов конференции. Зарегистрированы 1 патент и 1 программа для ЭВМ.

При использовании пакета biblatex будут подсчитаны все работы, добавленные в файл biblio/author.bib. Для правильного подсчёта работ в различных системах цитирования требуется использовать поля:

- authorvak если публикация индексирована ВАК,
- authorscopus если публикация индексирована Scopus,
- authorwos если публикация индексирована Web of Science,

- `authorconf` для докладов конференций,
- `authorpatent` для патентов,
- `authorprogram` для зарегистрированных программ для ЭВМ,
- `authorother` для других публикаций.

Для подсчёта используются счётчики:

- `citeauthorvak` для работ, индексируемых ВАК,
- `citeauthorscopus` для работ, индексируемых Scopus,
- `citeauthorwos` для работ, индексируемых Web of Science,
- `citeauthorvakscopuswos` для работ, индексируемых одной из трёх баз,
- `citeauthorscopuswos` для работ, индексируемых Scopus или Web of Science,
- `citeauthorconf` для докладов на конференциях,
- `citeauthorother` для остальных работ,
- `citeauthorpatent` для патентов,
- `citeauthorprogram` для зарегистрированных программ для ЭВМ,
- `citeauthor` для суммарного количества работ.

Для добавления в список публикаций автора работ, которые не были процитированы в автореферате, требуется их перечислить с использованием команды `\nocite` в `Synopsis/content.tex`.

Объем и структура работы. Диссертация состоит из введения, 4 глав, заключения и 3 приложений. Полный объём диссертации составляет 69 страниц, включая 9 рисунков и 13 таблиц. Список литературы содержит 5 наименований.

Глава 1. Оформление различных элементов

1.1 Расчет дальности действия связи

Перед тем как приступить к задаче ЦЛП необходимо рассчитать характеристики станции: радиус связи R_{jq} и радиус покрытия r_j .

При развертывания сети необходимо обеспечить максимальное покрытие данного участка связь между шлюзами через систему размещенных базовых станций беспроводной широкополосной сети.

Для оценки производительности канала связи воспользуемся уравнением энергетического потенциала. Полное уравнение можно записать следующим образом:

$$P_{tr} - L_{tr} + G_{tr} - L_{fs} + G_{recv} - L_{recv} = SOM + P_{recv}, \quad (1.1)$$

где:

- P_{tr} – мощность передатчика, дБм;
- L_{tr} – потери сигнала на антенном кабеле и разъемах передающего тракта, дБ;
- G_{tr} – усиление антенны передатчика, дБ;
- L_{fs} – потери в свободном пространстве, дБ;
- G_{recv} – усиление антенны приемника, дБ;
- L_{recv} – потери сигнала на антенном кабеле и разъемах приемного тракта, дБ;
- SOM – запас на замирание сигнала, дБ;
- P_{recv} – чувствительность приемника, дБм.

Мощность принимаемой антенны рассчитывается из уравнения передачи Фрииса:

$$\frac{P_{recv}}{P_{tr}} = G_{tr} G_{recv} \left(\frac{c}{4\pi R f} \right)^2,$$

где c – скорость света, f – частота, R расстояние между приемной и передающей антенной.

The Free Space Path Loss ($FSPL$) equation defines the propagation signal loss between two antennas through free space (air):

Уравнение потерь в свободном пространстве (Free Space Path Loss, $FSPL$) определяет потерю сигнала при распространении между двумя антеннами в свободном пространстве (в воздухе):

$$FSPL = \left(\frac{4\pi Rf}{c} \right)^2. \quad (1.2)$$

Формула ((1.2)), выраженная в децибеллах будет выражаться как

$$L_{fs} = 20 \lg F + 20 \lg R + K, \quad (1.3)$$

где F – центральная частота, на котором работает канал связи, R – расстояние между приемной и передающей антенной и K – константа.

Константа K зависит от размерностей частоты и расстояния:

- для чистоты, выраженной в ГГц, и расстояния, выраженная в км, константа K равна 92.45;
- для чистоты, выраженной в МГц, и расстояния, выраженная в км, константа K равна 32.4;
- для чистоты, выраженной в МГц, и расстояния, выраженная в м, константа K равна -27.55.

Потеря L_{fs} выразим из формулы ((1.1)) как:

$$L_{fs} = P_{tr} - L_{tr} + G_{tr} + G_{recv} - L_{recv} - SOM - P_{recv}. \quad (1.4)$$

Радиус связи получаем из уравнений ((1.3)) и ((1.4)):

$$R = 10^{\left(\frac{L_{fs} - 20 \lg F - K}{20} \right)}. \quad (1.5)$$

Используя формулу (1.5) и (1.4), мы можем рассчитать теоритическое максимальную дальность связи R_{jq} между базовыми станциями и радиусом покрытия r_j с предположением об отсутствии препятствий, отражений, влияния контуров местности и т. д. Это допущение приемлемо для нашего случая с открытой местностью.

Для расчета дальности связи R_{jq} (Рис. 1.1), базовые станции s_j и s_q будут рассматриваться как станции *передатчик* и *приемник*, соответственно. Будем считать, что станции оборудованы направленными антеннами с усилениями G_{tr}^R и G_{recv}^R .

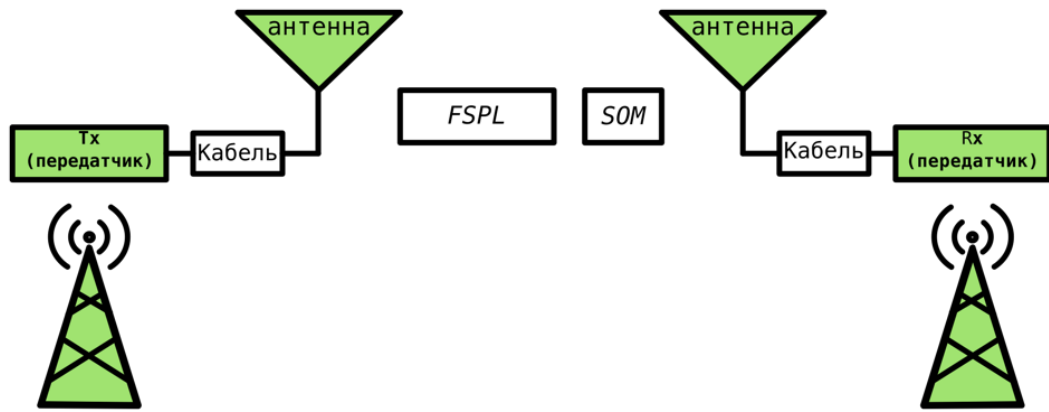


Рисунок 1.1 — Соединение между станциями.

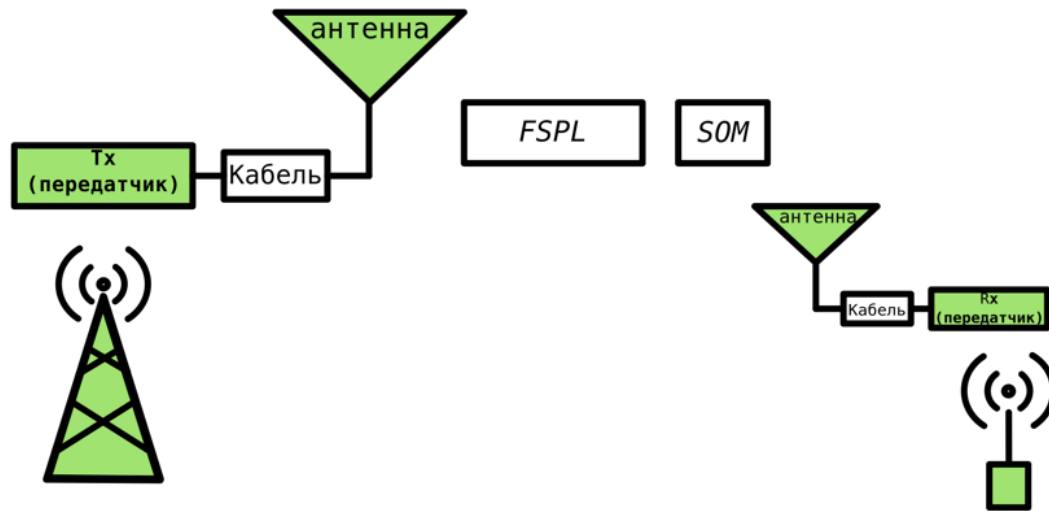


Рисунок 1.2 — Покрывтие станции

Каждая базовая станция оснащена всенаправленной антенной с заданным усилением антенны G_{tr}^r . Данная антенн необходимо для покрытия заданной области.

Each base station is equipped with an omnidirectional antenna with given gain antenna G_{tr}^r . A station uses this antenna to cover a given area.

При вычислении радиуса покрытия r_j (Рис. 1.2) базовая станция будем считать *передатчиком* а пользовательское устройство *приемником*.

Глава 2. Оптимальное размещение базовых станций широкополосной беспроводной сети связи для обслуживания заданного множества рассредоточенных объектов

Построение современной инфраструктуры передачи информации для обслуживания множества объектов промышленного или гражданского назначения, рассредоточенных на некоторой территории, является актуальной задачей при создании единой систем контроля и управления указанными объектами. Создание такой инфраструктуры позволяет обеспечить оперативный контроль и управление объектами путем передачи необходимой информации с сенсоров и датчиков объектов в соответствующий внешнее приемное устройство. Для создания подобной инфраструктуры эффективно используются сети широкополосной беспроводной связи, необходимым этапом проектирования которых является решение задачи определения мест размещения базовых станций [1]. В настоящей работе строятся и исследуются две математические модели задач размещения базовых станций, которые применимы на этапе синтеза топологии сети в процессе комплексного проектирования мультимедийных сетей. Предлагается модель для проверки существования допустимого решения при условии выполнения технологических ограничений для предложенной на предыдущих этапах схемы расстановки станций и модель для оптимизационной задачи. Оптимизационная задача состоит в выборе множества станций из заданного набора типов станций с различными характеристиками и их расстановки на избыточном множестве возможных мест размещения. В поставленной задаче рассматривается задача обслуживания объектов, расположение которых задано их координатами на плоскости. Особенностью такой задачи в широком классе задач оптимального размещения мощностей является наличие условия на наличие информационной связи между станциями и внешним приемным устройством (шлюзом), выполнение которого гарантирует поступление всей информации с контролируемых объектов в центр управления.

2.1 Задача при заданных местах размещения станций.

Задано множество вершин $A = \{a_i\}, i = \overline{0, n}$ на плоскости. Каждая вершина a_i имеет координаты $\{x_i, y_i\}$.

Множество A состоит из двух подмножеств:

- A_1 — множество вершин, которое соответствует объектам, с которых необходимо собирать информацию. Каждой вершине a_i приписана величина v_i — максимальный объем информации, снимаемой с объекта, расположенного на этой вершине. В частности, объектами могут быть любые стационарные абонентские устройства сети 802.11n. В дальнейшем будем считать, что каждая вершина из A_1 является объектом контроля.
- A_2 — множество мест, где размещены базовые станции. В дальнейшем вершину из A_2 будем идентифицировать не только как место размещения, но и как соответствующую станцию.

По определению:

$$A_1 \cup A_2 = \emptyset;$$

$$A_1 \cap A_2 = \emptyset.$$

Все вершины пронумерованы так, что:

$$A_1 = \{a_i\}, i = \overline{1, n_1};$$

$$A_2 = \{a_i\}, i = \overline{n_1 + 1, n}.$$

Каждой вершине из A_2 приписаны три параметра $s_i = \{r_i, R_{ij}, \vartheta_i\}$, где:

- r_i — максимальный радиус покрытия станции. Параметр, который характеризует зону охвата территории каждой станцией;
- R_{ij} — максимальный радиус связи между i -ой и j -ой станциями. Параметр характеризует расстояние, на котором обеспечивается связь между станциями;
- ϑ_i — максимальный объем информации в единицу времени, который может быть получен от объектов, обслуживаемых станцией.

Также задана вершина специального вида (шлюз) $s_0 = \{r_0, R_0, \vartheta_0\}$ с координатами $\{x_0, y_0\}$. По условию задачи величина ϑ_0 больше суммы величин ϑ_i у всех вершин множества A_1 .

Задано условие, что со шлюзом и между собой могут быть связаны только вершины множества A_2 .

Требуется проверить, что при заданных наборе и размещении станций вся имеющаяся информация с объектов (множество A_1) может быть собрана и передана системой станций (множество A_2) до шлюза s_0 .

Сформулируем задачу в виде модели **ЛП**.

Составим граф $H = \{A, E\}$ для возможного потока информации между вершинами множества $A = A_1 \cup A_2$. По определению, каждой вершине из A_2 соответствует свой набор параметров $\{r_i, R_i, \vartheta_i\}$. Матрица смежности $E = \{e_{ij}\}$ графа H строится по следующим правилам:

- $e_{ij} = 1$, если расстояние между i -ым объектом ($a_i \in A_1$) и j -ым местом размещения станции ($a_j \in A_2$) не более радиуса покрытия для станции соответствующего этой вершине типа;
- $e_{ij} = 1$, если расстояние между i -ым местом размещения ($a_i \in A_2$) и j -ым местом размещения ($a_j \in A_2$), не более радиуса связи той станции, у которой радиус связи не больше радиуса связи другой станции;
- $e_{i0} = 1$, если расстояние от вершины $a_i \in A_2$ до шлюза не более R_i ;
- $e_{ij} = 0$, во всех остальных случаях.

Введем переменные $x_{ij} \geq 0$. Это искомое количество информации, передаваемой в единицу времени по дуге e_{ij} графа H . Распишем условия для нашей задачи. Величина суммарного потока, который выходит с объекта равен весу ϑ_i :

$$\sum_{a_j \in \Gamma^+(a_i)} x_{ij} = \vartheta_i, \forall a_i, i = \overline{1, n_1}, \quad (2.1)$$

где $\Gamma^+(a_i)$ – множество вершин на графе H , в которые входят дуги, исходящие из вершины a_i .

Сумма входящих и исходящих потоков для любой i -ой вершины множества A_2 равна нулю:

$$\sum_{a_j \in \Gamma_1^-(a_i)} x_{ij} + \sum_{a_j \in \Gamma_2^-(a_i)} x_{ji} - \sum_{a_j \in \Gamma_2^+(a_i)} x_{ij} = 0, \forall a_i \in A_2. \quad (2.2)$$

Здесь множество $\Gamma_1^-(a_i)$ – вершины множества A_1 , из которых выходят дуги, входящие в вершину a_i , $\Gamma_2^-(a_i)$ – вершины множества A_2 , из которых выходят дуги, входящие в вершину a_i , $\Gamma_2^+(a_i)$ – вершины множества A_2 , в которые входят дуги, исходящие из вершины a_i .

Через систему станций вся информация от объектов должна поступить на шлюз s_0 :

$$\sum_{a_j \in \Gamma_2^-(a_0)} x_{j0} = \sum_{a_i \in A_1} \vartheta_i; \quad (2.3)$$

Объем информации, поступающей с других вершин на станцию, если она размещена на j -ой вершине, ограничен мощностью станции ϑ_j :

$$\sum_{a_j \in \Gamma^-(a_i)} x_{ji} \leq \vartheta_j, \forall a_j \in A_2 D. \quad (2.4)$$

Для нахождения допустимого решения задач (2.1) – (2.4) (или доказательства, что допустимого решения не существует) может быть применена стандартная процедура нахождения допустимого решения задачи линейного программирования с вводом искусственных переменных в уравнения (2.1) – (2.4) и минимизации состоящей из этих переменных линейной формы. Если значение целевой функции в результате решения задачи окажется больше нуля, то допустимого решения для данного размещения станций не существует, в противном случае полученное решение дает допустимое распределение потоков по каналам связи.

2.2 Оптимизационная задача выбора набора размещаемых станций и определения мест их размещения

Постановка задачи Задано множество вершин $A = a_i, i = \overline{0, n}$ на плоскости. Каждая вершина a_i имеет координаты $\{x_i, y_i\}$. Множество A состоит из двух подмножеств:

- A_1 – множество вершин, с которых необходимо собирать информацию. Каждой вершине a_i приписана величина v_i – максимальный объем информации, снимаемой с объекта, расположенного на этой вершине;

– A_2 – множество возможных мест размещения базовых станций.

По определению

$$A_1 \cup A_2 = \emptyset;$$

$$A_1 \cap A_2 = A.$$

Все вершины пронумерованы так, что:

$$A_1 = \{a_i\}, i = \overline{1, n_1};$$

$$A_2 = \{a_i\}, i = \overline{n_1 + 1, n}.$$

Задано множество типов базовых станций $S = s_j, j = \overline{1, s}$, которые необходимо разместить на множестве A_2 .

Каждой станции приписаны четыре параметра $s_j = \{r_j, R_j, \vartheta_j, c_j\}$, где:

- r_j – максимальный радиус покрытия;
- R_{ij} – максимальный радиус связи между i -ой и j -ой станциями. Параметр характеризует расстояние, на котором обеспечивается связь между станциями;
- ϑ_j – максимальный объем информации в единицу времени, который может быть получен от объектов, обслуживаемых данной станцией;
- c_j – стоимость станции.

Также задана станция специального вида (шлюз) $s_0 = \{r_0, R_0, \vartheta_0, c_0\}$ с координатами $\{x_0, y_0\}$, где $r_0 = R_0 = \vartheta_0 = c_0 = 0$

Требуется разместить станции таким образом, чтобы вся информация с объектов (вершинах множества A_1) могла быть собрана и передана системой станций, размещенных на выбранных в результате решения задачи вершинах множества A_2 , до шлюза s_0 и общая стоимость размещенных станций была бы минимальной. Как и в предыдущих задачах вершины и станции будем, соответственно, идентифицировать как объекты или станции на них размещенные. Задано условие, что информация с вершин множества A_1 может передаваться непосредственно только на вершины множества A_2 , а со шлюзом и между собой могут быть связаны только вершины множества A_2 .

Заметим, что в отличие от предыдущих двух задач в данной задаче задано не множество станций, которые все должны быть использованы в проектируемой сети, а только типы станций. Таким образом в результате решения задачи определяется как набор станций, так и места их размещения. Формулировка задачи в виде модели частично целочисленного ЛП. Вместо каждой вершины a_i , $i = \overline{n_1 + 1, n}$ введем m вершин с координатами вершины a_i , и различными параметрами, соответствующими различным типам станций. Обозначим такую группу вершин, записанных с одинаковыми координатами вместо вершины a_i , как D_i . Каждой вершине из D_i поставим в соответствие набор параметров только одного типа станции из S , т.е. на данной вершине может стоять либо станция приписанного типа либо никакая. Обозначим расширенное множество вершин A_2 через A_2D .

Составим граф $H = \{AD, E\}$, описывающий сеть для передачи потока информации между вершинами расширенного множества $AD = A_1 \cup A_2D$ и шлюзом. Матрица смежности $E = e_{ij}$ графа H строится по следующим правилам.

- $e_{ij} = 1$, если расстояние между i -ой вершиной ($a_i \in A_1$) и j -ой вершиной ($a_j \in A_2D$) не более радиуса покрытия, приписанного этой вершине;
- $e_{ij} = 1$, если вершины a_i и a_j принадлежат разным множествам D_i и D_j и расстояние между ними не более радиуса связи той вершины, у которой радиус связи не больше радиуса связи другой вершины;
- $e_{i0} = 1$ ($a_i \in A_2D$) если расстояние от вершины до шлюза не более R_i ;
- $e_{ij} = 0$, во всех остальных случаях.

Введем потоковые переменные $x_{ij} \geq 0$.

Распишем условия для нашей задачи. Величина суммарного потока, который выходит с вершины a_i равен весу ϑ_i

$$\sum_{a_j \in \Gamma^+(a_i)} x_{ij} = \vartheta_i, \forall a_i, i = \overline{1, n_1}; \quad (2.5)$$

где $\Gamma^+(a_i)$ – множество вершин на графе H , в которые входят дуги, исходящие из вершины a_i .

Сумма входящих и исходящих потоков для любой i -ой вершины множества A_2D равна нулю

$$\sum_{a_j \in \Gamma_1^-(a_i)} x_{ij} + \sum_{a_j \in \Gamma_2^-(a_i)} x_{ji} - \sum_{a_j \in \Gamma_2^+(a_i)} x_{ij} = 0, \forall a_i \in A_2. \quad (2.6)$$

Здесь множество $\Gamma_1^-(a_i)$ – вершины множества A_1 , из которые выходят дуги, входящие в вершину a_i , $\Gamma_2^-(a_i)$ – вершины множества A_2D , из которых выходят дуги, входящие в вершину a_i , $\Gamma_2^+(a_i)$ – вершины множества A_2D , в которые входят дуги, исходящие из вершины a_i .

Через систему станций вся информация от объектов должна поступить на шлюз s_0

$$\sum_{a_j \in \Gamma_2^-(a_0)} x_{j0} = \sum_{a_i \in A_1} \vartheta_i. \quad (2.7)$$

Здесь $\Gamma_2^-(a_0)$ – подмножество вершин множества A_2D , дуги которых входят в шлюз a_0 .

Введем булевы переменные y_i для вершин a_i , $a_i \in A_2D$

- $y_i = 1$, если станция стоит на месте a_i ;
- $y_i = 0$, в противном случае.

Объем информации, поступающей от вершин множества A_1 на вершину $a_i \in A_2D$, ограничен мощностью станции ϑ_i .

$$\sum_{a_j \in \Gamma^-(a_i)} x_{ji} \leq y_i \cdot \vartheta_i, \forall a_i \in A_2D. \quad (2.8)$$

На множестве D_i может быть размещено не более одной станции

$$\sum_{a_j \in D_i} y_j \leq 1, \forall D_i. \quad (2.9)$$

Целевая функция

$$\sum_{a_i \in A_2D} c_i \cdot y_i \rightarrow \min. \quad (2.10)$$

Задача (2.5) – (2.10) представляет собой частично целочисленную задачу линейного программирования с $s \cdot |A_2|$ булевыми переменными.

Таблица 1 — Координаты размещения

0	(7,4)	Координаты шлюза
1	(1, 5)	Координаты объектов
2	(4.5, 4)	
3	(6, 3)	
4	(3.5, 5)	
5	(2, 4)	Координаты размещения станций
6	(5, 5)	
7	(2, 6)	
8	(6, 5.5)	

Таблица 2 — Координаты размещения

Объекты	1	2	3	4
Мощность	10	15	17	18

Таблица 3 — Множество типов станций

Тип	Мощность, ϑ_j	Радиус покрытия, r_j	Радиус связи, R_j	Стоимость, c_j
1	80	1	6	70
2	100	2	5	75
3	100	2	5	75

2.3 Пример решения задачи

Рассмотрим пример для оптимизационной задачи выбора набора размещаемых станций и определения мест их размещения. Задано множество рассредоточенных объектов A_1 , $|A_1| = 4$ и шлюз (таблица 1).

Задано множество A_2 возможных мест расположения станций, $|A_2| = 4$. Все вершины представлены на рисунке 2.1.

Задано ограничение по мощности для кадого объекта (таблица 2).

Задано множество типов станций (таблица 4).

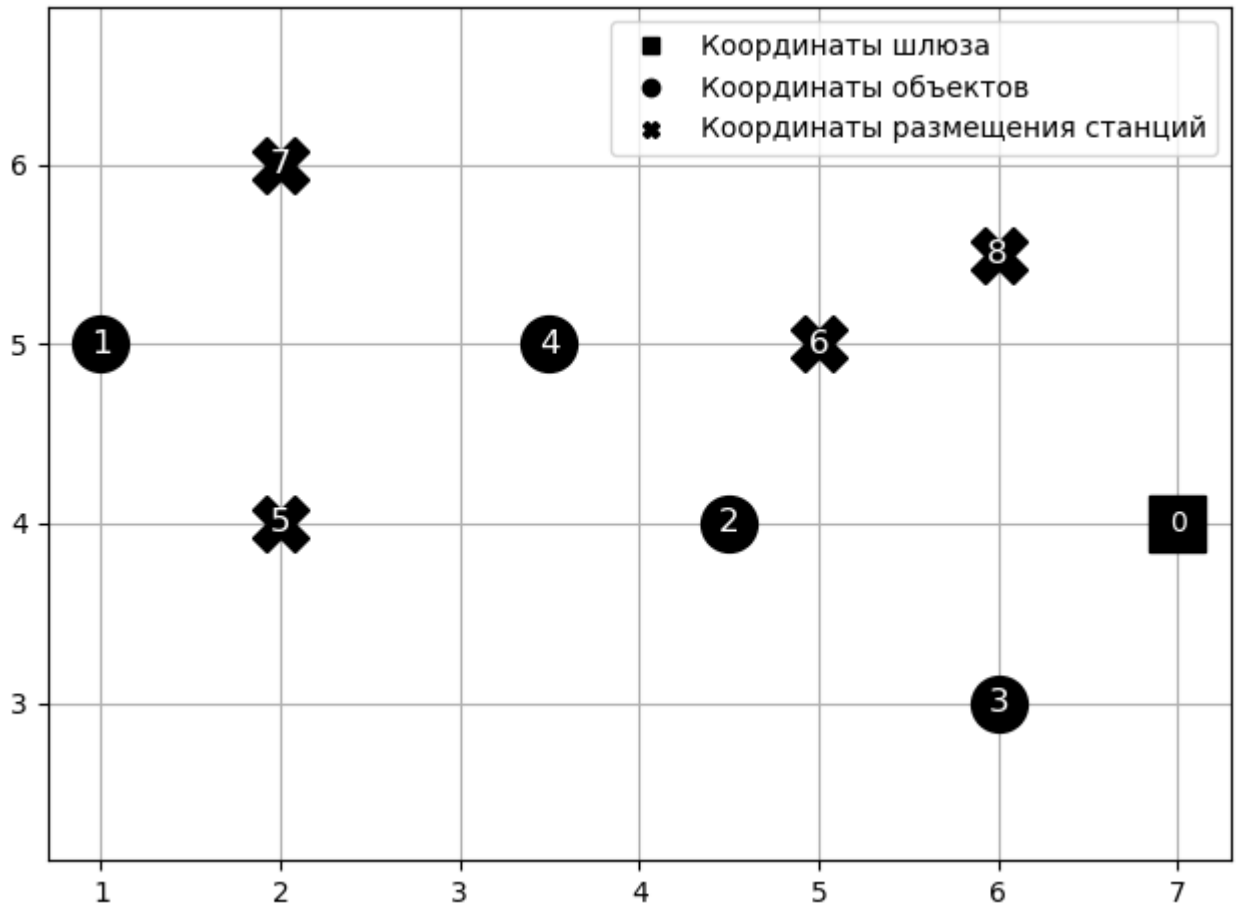


Рисунок 2.1 — Координаты размещения

Необходимо разместить станции таким образом, чтобы минимизировать их суммарную общую стоимость. Построим граф сети H для данного набора типов станции. Матрица смежности представлена на рисунке 2.2

На основе матрицы смежности полученного графа запишем систему равенств и неравенств (2.5) — (2.10) и решим задачу частично целочисленного ЛП. В ходе решения мы получили следующее размещение станции (рис. 2.3)

Из графика видно, что были размещены на точках 7 и 8 две станции типа 2 и 3, соответственно. Решением задачи является суммарная стоимость равная: $f = 160$.

2.4 Результаты численного эксперимента

Алгоритмы построения графов H были запрограммированы на языке Python. Задачи, сформулированные на основании графов H в виде соответ-

	a_0	a_1	a_2	a_3	a_4	a_5s_1	a_6s_1	a_7s_1	a_8s_1	a_5s_2	a_6s_2	a_7s_2	a_8s_2	a_5s_3	a_6s_3	a_7s_3	a_8s_3
a_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a_1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0
a_2	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1
a_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
a_4	1	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1
a_5s_1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
a_6s_1	1	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1
a_7s_1	1	0	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1
a_8s_1	0	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1
a_5s_2	1	0	0	0	0	1	1	1	1	0	1	1	1	1	1	1	1
a_6s_2	0	0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	1
a_7s_2	1	0	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1
a_8s_2	1	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	1
a_5s_3	1	0	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1
a_6s_3	1	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1
a_7s_3	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1
a_8s_3	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0

Рисунок 2.2 — Координаты размещения

Таблица 4 — Множество типов станций

Количество объектов, n_1	Количество мест размещения станций, $n - n_1$	Среднее время счета, сек.
4	3	12,34
4	4	12,42
4	5	12,31
6	6	11,20
8	7	11,27
10	7	12,32
12	10	12,51
14	7	12,42
17	8	12,18
21	8	12,53
25	8	14,22

ствующих задач математического программирования, были решены пакетом Optimization Toolbox MATLAB. В таблице 4 представлены результаты времени счета задач частично целочисленного ЛП для различных случаев числа мест размещения станций и числа объектов. Для каждого случая было проведено по 10 примеров.

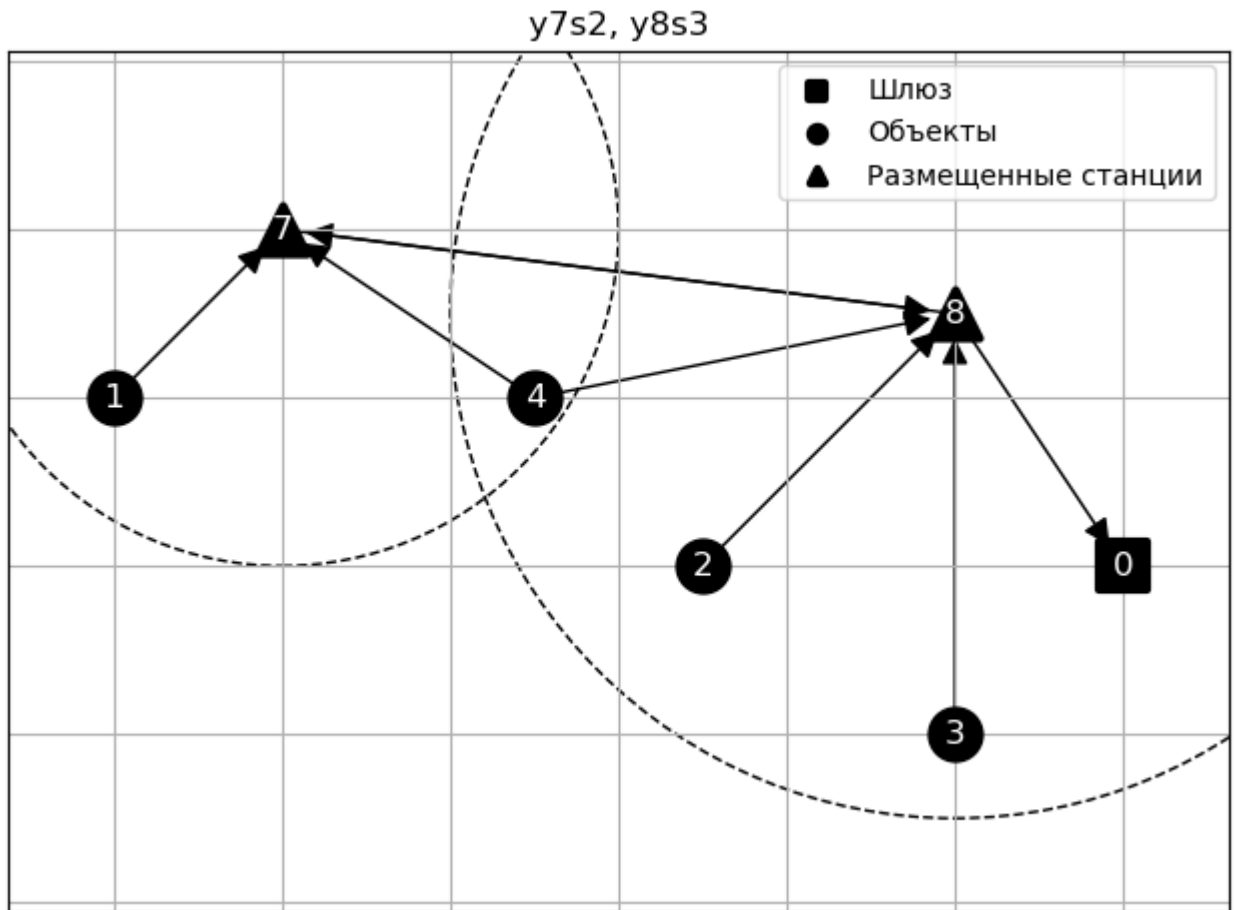


Рисунок 2.3 — Координаты размещения

2.5 Выводы к главе 2

В работе рассмотрены задачи размещения базовых станций при проектировании беспроводных широкополосных сетей связи. Предложены формулировки задач в виде моделей линейного и частично целочисленного линейного программирования как для случая проверки наличия допустимых решений для вариантов, предложенных проектировщиками, так и для экстремальной задачи отбора множества станций из имеющегося набора типов станций и оптимального размещения станций выбранного множества на избыточном множестве возможных мест размещения. Предложены алгоритмы построения графов информационных потоков, позволившие формализовать задачи в виде соответствующих моделей математического программирования. Приведены результаты вычислительного эксперимента.

Глава 3. Математические модели синтеза топологии сети для охвата линейного участка в виде задачи целлочисленного линейного программирования

3.1 Постановка задачи

Проблема формулируется следующим образом. Для контроля над заданным линейным участком необходимо разместить базовые приемопередающие станции (далее называемые станциями) таким образом, чтобы максимизировать покрытие с ограничениями на суммарную стоимость размещенных станций. Важно обеспечить связи любой станции со шлюзами на концах участка через систему размещенных станций.

Задано множество станций $S = \{s_j\}$. Каждой станции приписаны параметры $s_j = \{r_j, \{R_{jq}\}, c_j\}$, $j = \overline{1, m}$; $q = \overline{1, m}$; $q \neq j$. Здесь r_j – радиус покрытия станции, R_{jq} – это радиус связи между станциями s_j и s_q , и c_j – это стоимость.

Задан линейный участок длиной L с концами в точка a_0 и a_{n+1} . Внутри отрезка $[a_0, a_{n+1}]$ задано конечное множество точек $A = \{a_i\}$, $i = \overline{1, n}$; эти точки соответствуют набору свободных мест, где могут быть размещены станции. Каждая точка a_i определяется своей одномерной координатой l_i .

Заданы станции специального вида s_{m+1} – шлюзы. Данные шлюзы размещены на концах a_0 и a_{n+1} данного линейного участка. Для данных станций параметр радиуса покрытия $r_{m+1} = 0$. Радиус связи и стоимость не заданы.

Требуется разместить станции таким образом, чтобы максимизировать покрытие с условием ограничения на суммарную стоимость C .

3.2 Модель ЦЛП

После оценки максимальных радиуса связи между станциями R_{jq} , максимального радиуса покрытия r_j , можно перейти, непосредственно, к задаче размещения станций в виде модели целлочисленного линейного программирования.

Пусть y_i^+ и y_i^- , $i = \overline{0, n+1}$ определяют охват покрытия (справа и слева, соответственно) станций, покрывающих точку a_i (Рис. 3.1). Параметры y_i^+ и y_i^- могут принимать только неотрицательные целые значения.

Величины покрытия для шлюзов $y_0^+, y_0^-, y_{n+1}^+, y_{n+1}^-$ равны 0.

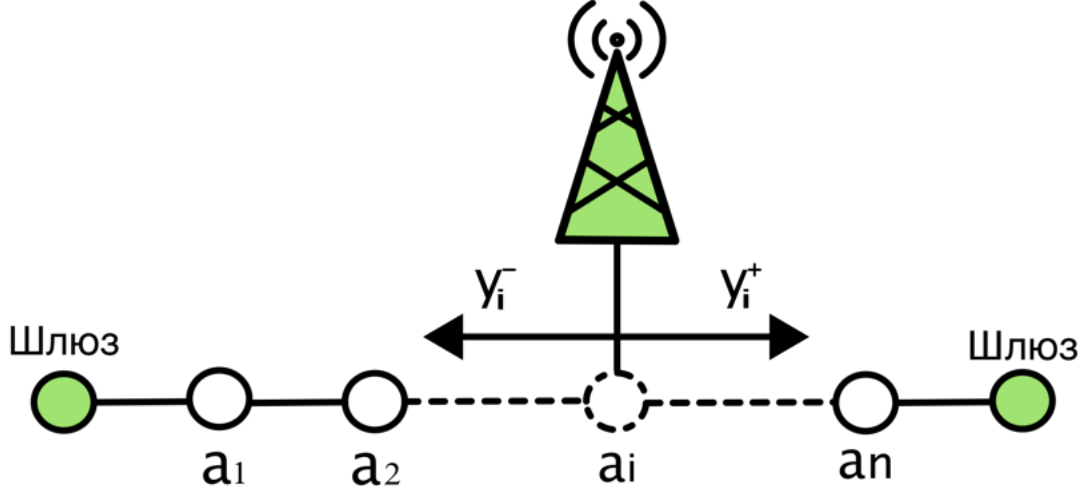


Рисунок 3.1 — Охват покрытия станции]

Целевая функция будет представлена как:

$$f = \sum_{i=1}^n (y_i^- + y_i^+) \rightarrow \max \quad (3.1)$$

Также введем бинарные переменные x_{ij} . Тогда $x_{ij} = 1$, если станция s_j , размещенная на точке a_i , и $x_{ij} = 0$ в противном случае; $i = \overline{1, n}$; $j = \overline{1, m}$.

Введем двоичные переменные e_i . Тогда $e_i = 1$, если какая-либо станция находится в точке a_i , и $e_i = 0$ в противном случае; $i = \overline{1, n}$. Для точек размещения шлюзов a_0 и a_{n+1} переменные $e_0 = 1$ и $e_{n+1} = 1$, соответственно.

Сформулируем следующую систему ограничений задачи.

По определению (3.2):

$$e_i = \sum_{j=1}^m x_{ij}, \quad i = \overline{1, n}. \quad (3.2)$$

Каждая станция должна быть размещена только в одной точке. (3.3):

$$\sum_{j=1}^n x_{ij} \leq 1, \quad j = \overline{1, m}. \quad (3.3)$$

Значения покрытий не превышают радиус покрытия станции, размещенной в точке a_i , и равны 0, если в точке a_i нет станции (3.4) и (3.5):

$$y_i^+ \leq \sum_{j=1}^m x_{ij} \cdot r_j, \quad i = \overline{1, n}; \quad (3.4)$$

$$y_i^- \leq \sum_{j=1}^m x_{ij} \cdot r_j, \quad i = \overline{1, n}. \quad (3.5)$$

Общая область покрытия между любыми двумя точками a_i и a_k , где расположены станции, не может превышать расстояние между этими точками (3.6) и (3.7).

$$y_i^+ + y_k^- \leq \frac{l_k - l_i}{2} \cdot (e_i + e_k) + (2 - e_i - e_k) \cdot L, \quad i = \overline{1, n}, \quad k = \overline{i+1, n+1}; \quad (3.6)$$

$$y_i^- + y_k^+ \leq \frac{l_i - l_k}{2} \cdot (e_i + e_k) + (2 - e_i - e_k) \cdot L, \quad i = \overline{1, n}, \quad k = \overline{i-1, 0}, \quad (3.7)$$

где l_k и l_i - координаты точек a_i и a_k , соответственно. Это условие исключает влияние пересечений покрытий станций при вычислении общего значения покрытия между станциями (Рис. 3.2).

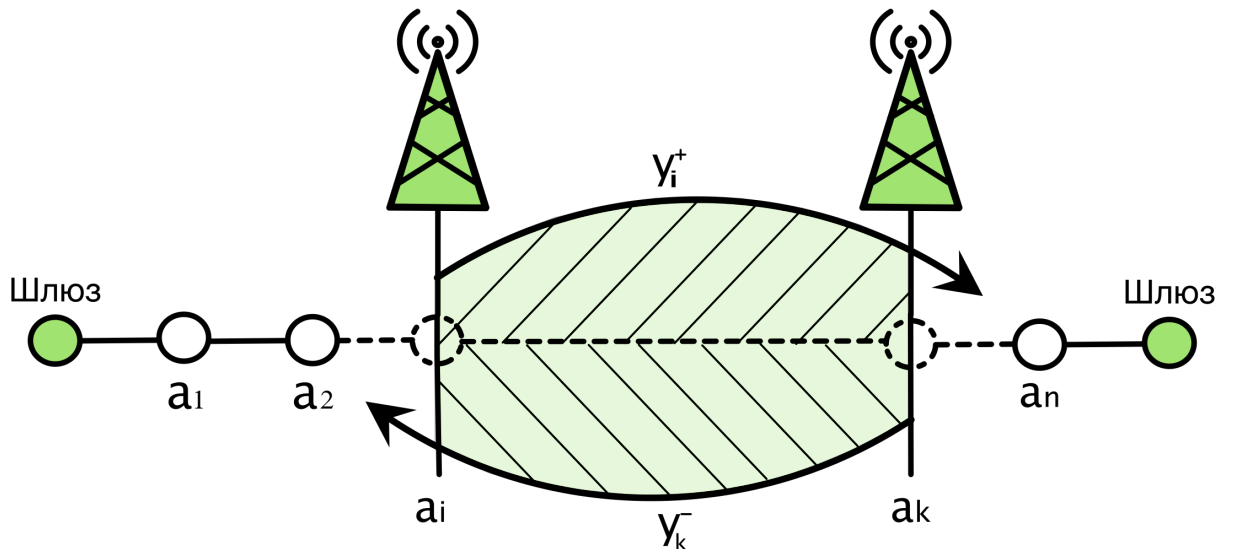


Рисунок 3.2 — Область покрытия между любыми двумя точками

Согласно условиям задачи, станция, расположенная в a_i , должна быть связана хотя бы с одной станцией слева и одной станцией справа, включая станции на конечных точках a_0 и a_{n+1} .

Введем бинарные переменные $z_{ijkq}, i = \overline{1, n}; j = \overline{1, m}; k = \overline{1, n}, k \neq i; q = \overline{1, m}, q \neq j$.

Переменная $z_{ijkq} = 1$, если в точке a_i размещена станция s_j и данная станция связана со станцией s_q , размещенная в точке a_k ; и $z_{ijkq} = 0$ в противном случае.

Переменная $z_{ij0(m+1)} = 1$, если станция s_j , размещенная в точке a_i , связана со шлюзом s_{m+1} в точке a_0 ; $z_{ij0(m+1)} = 0$ в противном случае.

Переменная $z_{ij(n+1)(m+1)} = 1$, если здесь находится станция s_j в точке a_i и она связана со шлюзом s_{m+1} в точке a_{n+1} ; $z_{ij0(m+1)} = 0$ в противном случае.

Станции должны быть размещены в обеих точках a_i и a_k , (3.8) и (3.9):

$$z_{ijkq} \leq e_i, \quad i = \overline{1, n}; \quad j = \overline{1, m}; \quad k = \overline{1, n}, k \neq i; \quad q = \overline{1, m}, q \neq j; \quad (3.8)$$

$$z_{ijkq} \leq e_k, \quad k = \overline{1, n}; \quad j = \overline{1, m}; \quad i = \overline{1, n}, i \neq k; \quad q = \overline{1, m}, q \neq j. \quad (3.9)$$

Необходимо, чтобы станция s_j в точке a_i была связана с любой станцией, расположенной в точке a_k , справа от a_i ($k > i$) или с правым шлюзом s_{m+1} (3.10) и (3.11).

$$\sum_{k=i+1}^n \sum_{\substack{q=1 \\ q \neq j}}^m z_{ijkq} + z_{ij0(m+1)} = x_{ij}, \quad i = \overline{1, n}, \quad j = \overline{1, m}. \quad (3.10)$$

Станция s_j , размещенная в a_n , справа связана только со шлюзом s_{m+1} на месте a_{n+1} (3.11).

$$z_{nj(n+1)(m+1)} = x_{nj} \quad j = \overline{1, m}. \quad (3.11)$$

Также станция должна быть связана с любой станцией, расположенной в точке a_k слева от точки a_i ($k < i$) или с левым шлюзом s_{m+1} (3.12) и (3.13).

$$z_{1j0(m+1)} = x_{1j}, \quad j = \overline{1, m}; \quad (3.12)$$

Станция s_j , размещенная в точке a_1 слева может быть связана только со шлюзом s_{m+1} , расположенном в точке a_0 (3.12).

$$z_{ij0(m+1)} + \sum_{k=1}^{i-1} \sum_{\substack{q=1 \\ q \neq j}}^m z_{ijkq} = x_{ij}, \quad i = \overline{2, n}, \quad j = \overline{1, m}. \quad (3.13)$$

Необходимо, чтобы станция s_q в точке a_k была связана с любой станцией справа, расположенной в точке a_i (3.14).

$$\sum_{i=k+1}^n \sum_{\substack{j=1 \\ j \neq q}}^m z_{ijkq} = x_{kq}, \quad k = \overline{1, n-1}, \quad q = \overline{1, m}; \quad (3.14)$$

Кроме того, станция s_q в точке a_k подключена к любой станции слева, расположенной в точке a_i (3.15).

$$\sum_{i=1}^k \sum_{\substack{j=1 \\ j \neq q}}^m z_{ijkq} = x_{kq}, \quad k = \overline{2, n}, \quad q = \overline{1, m}; \quad (3.15)$$

Неравенства (3.8) и (3.9) и равенства (3.10) — (3.15) обеспечивают условие симметрии связи между базовыми станциями, расположенными в точках a_i и a_k , $\forall i, k$ (Рис.3.3).

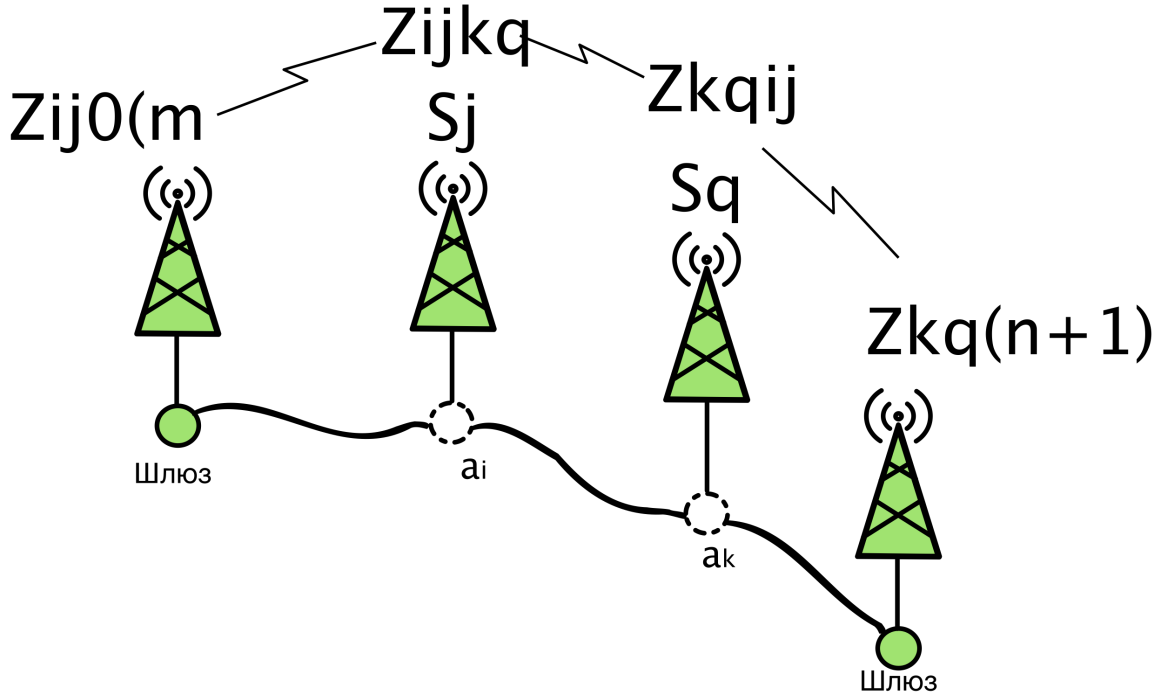


Рисунок 3.3 — Связь между базовыми станциями

Если станции s_j и s_q связаны, то максимальный радиус связи размещенных станций должен быть не меньше расстояния между точками a_i и a_k , где

расположены станции s_i и s_q (Рис. 3.4). Формально это можно записать как (3.16) и (3.17).

$$\forall i = \overline{1, n}:$$

$$z_{ijkq}(R_{jq} - (a_i - a_k)) \geq 0, \quad k = \overline{0, i-1}; \quad j = \overline{1, m}; \quad q = \overline{1, m}, q \neq j; \quad (3.16)$$

$$z_{ijkq}(R_{jq} - (a_k - a_i)) \geq 0, \quad k = \overline{i+1, n+1}; \quad j = \overline{1, m}; \quad q = \overline{1, m}, q \neq j. \quad (3.17)$$

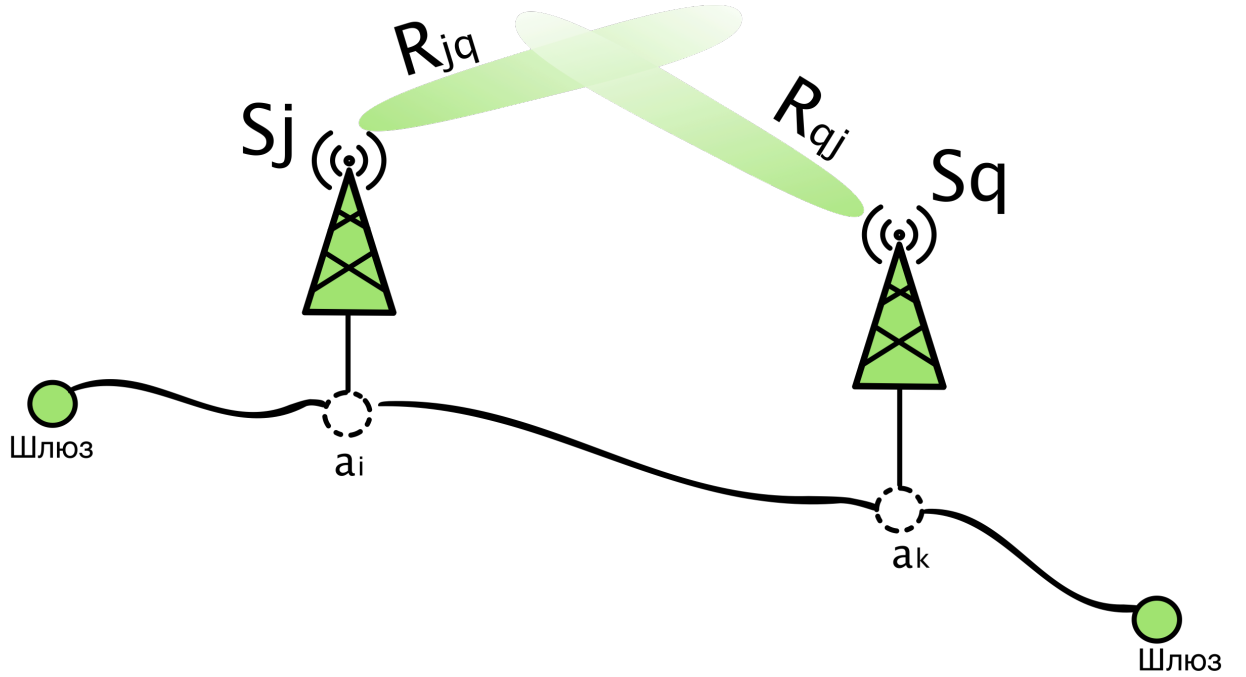


Рисунок 3.4 — Обеспечение связи с соседней станцией

И для бюджетного ограничения стоимости C :

$$\sum_{i=1}^n \sum_{j=1}^m x_{ij} \cdot c_j \leq C. \quad (3.18)$$

3.3 Численный пример

В этой секции представлен численный пример решения данной задачи.

Задан линейный участок L с длиной 300 с количеством $n = 7$ точек размещения. Координаты точек размещения представлены в таблице 5. Задан бюджет размещения $C = 130$. Центральная частота $f = 2437$ МГц.

a_i	a_1	a_2	a_3	a_4	a_5	a_6	a_7
Координата	29	40	95	139	181	230	273

Таблица 5 — Точки размещения участка с длиной $L = 300$.

Задано множества базовых станций $m = 8$ с параметрами представленными в таблице 6. Также в таблице представлены параметры шлюзов и контролируемых объектов. Параметры объектов необходимы для расчета радиусов покрытия станций.

3.3.1 Расчет радиуса связи между станциями

Базовые станции оснащены направленной антенной с высоким коэффициентом усиления для связи с соседними станциями. Для расчета потерь между станциями j и q воспользуемся формулой (1.4):

$$L_{fs}^{jq} = P_{tr}^R(j) - L_{tr} + G_{tr}^R(j) + G_{tr}^R(q) - L_{recv} - SOM - P_{recv}^R(q).$$

BS	P_{tr}^R	G_{tr}^R	P_{recv}^R	P_{recv}^r	G_{recv}^r	c
	дБм	дБ	дБм	дБм	дБ	у.е.
1	20	5	-69	-67	5	40
2	19	5	-67	-67	5	28
3	18	5	-69	-67	5	45
4	19	5	-69	-67	6	22
5	19	5	-67	-67	5	21
6	20	5	-69	-67	5	40
7	19	5	-67	-67	5	28
8	18	5	-69	-67	5	45
Шлюз	G_{recv}^R	P_{recv}^R		Объект	P_{tr}^r	G_{tr}^r
	дБ	дБм			дБм	дБ
	5	-69			15	2

Таблица 6 — Параметры базовых станций, шлюзов и объектов.

Потери на кабелях приемника L_{recv} и передатчика L_{tr} примем равным 1 дБ и запас на замирания сигнала $SOM = 10$ дБ.

Let us carry out an example of the calculation communication link between stations s_1 and s_2 : Для примера рассчитаем радиус связи между станциями s_1 и s_2 :

$$\begin{aligned} L_{fs}^{12} &= P_{tr}^R(1) - L_{tr} + G_{tr}^R(1) + G_{tr}^R(2) - L_{recv} - SOM - P_{recv}^R(2) = \\ &= 20 - 1 + 5 + 5 - 1 - 10 - (-69) = 87(dB). \end{aligned} \quad (3.19)$$

Для расчета канала связи необходимо использовать формулу (1.5). Несущая частота $f = 2437$ МГц и коэффициент для расчета потерь $K = -27,55$:

$$R_{jq} = 10^{\left(\frac{L_{fs}^{jq} - 20 \lg F - K}{20}\right)} = 10^{\left(\frac{87 - 20 \lg 2437 - (-27.55)}{20}\right)} = 174(m). \quad (3.20)$$

В таблице 7 приведены расчеты максимальных радиусов связи между всеми станциями s_j , $j = 1, \dots, m$ и шлюзом s_{m+1} .

$R_{jq}, (m)$	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_{m+1}
s_1	—	174	219	219	174	219	174	219	219
s_2	195	—	195	195	155	195	155	195	195
s_3	174	138	—	174	138	174	138	174	174
s_4	195	155	195	—	155	195	155	195	195
s_5	195	155	195	195	—	195	155	195	195
s_6	219	174	219	219	174	—	174	219	219
s_7	195	155	195	195	155	195	—	195	195
s_8	174	138	174	174	138	174	138	—	174

Таблица 7 — Рассчитанные радиусы связи между станциями

3.3.2 Расчет радиуса покрытия

Расчет проводится аналогично расчету радиусу связи между станциями. Потери в свободном пространстве для канала между j -ой станции и контролируемым объектом

$$L_{fs}^j = P_{tr}^r(j) - L_{tr} - SOM - P_{RX}.$$

Пример расчета радиуса покрытия для 1-ой станции:

$$L_{fs}^1 = P_{tr}^r + G_{tr}^r + G_{recv}^r(1) - L_{recv}(1) - SOM - P_{recv}^r(1) = 15 + 2 + 5 - 1 - (-67) - 10 = 78 \text{ (дБм)}$$

$$r_1 = 10^{\left(\frac{78 - 20 \lg 2437 - (-27.55)}{20}\right)} = 77 \text{ (м)}.$$

Рассчитанные радиусы покрытия для всех станций s_j , $j = \overline{1, m}$ представлены в таблице 8).

STA	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
r_j	77	77	77	87	77	77	77	77

Таблица 8 — Рассчитанные радиусы покрытия станций

Задача ЦЛП решена с помощью Optimization Toolbox MatLab. Таблица 9 содержит все возможные целочисленные решения.

a_i	a_1	a_2	a_3	a_4	a_5	a_6	a_7	Покрытие	Цена
Координаты	29	40	95	139	181	230	273	м	у.е.
Целлочисленное решение 1	s_1	s_2	s_6	—	—	—	s_4	286	130
Целлочисленное решение 2	s_4	—	s_5	s_7	—	—	s_2	289	99
Оптимальное решение	s_4	s_2	—	—	s_1	—	s_5	300	111

Таблица 9 — Решение задачи ЦЛП.

3.4 Выводы по Главе 3

Представлена математическая модель задачи размещения базовых станций беспроводной сети связи вдоль линейного участка в виде задачи ЦЛП. В качестве примера представлен численный пример решения задачи.

Глава 4. Математические модели синтеза топологии сети для охвата линейного участка в виде экстремальной задачи в комбинаторной форме

Эффективным способом повышения технико-экономических показателей при проектировании **БШС** является оптимизация топологии сети, а именно решение задачи выбора оптимального набора станций из заданного избыточного множества и определение мест их размещения вдоль линейной контролируемой территории. Основным результатом работы, представленной в этой главе, является разработка итерационного метода выбора оптимальной топологии сети в процессе комплексного проектирования БШС. Принципиальной особенностью предлагаемого метода, повышающей его эффективность, является то, что для рассмотрения на этапе моделирования предлагается не одно решения, а последовательности лучших решений задачи оптимизации топологии сети. Это позволяет с помощью разработанной итерационной процедуры выбирать на этапе моделирования лучшее решение среди тех решений по топологии, которые удовлетворяют требуемым характеристикам проектируемой БШС.

4.1 Постановка задачи и ее формулировка в экстремальной комбинаторной форме

Пусть задано множество станций $S = \{s_j\}$ с параметрами $s_j = \{r_j, \{R_{jq}\}, \mu_j, c_j\}, j = 1, \dots, m; q = 1, \dots, m; j \neq q$. Здесь r_j – максимальный радиус покрытия станции, $\{R_{jq}\}$ – множество максимальных радиусов связи между j -ой и q -ой базовой станции, μ_j – интенсивность времени обслуживания и c_j – стоимость станции.

Задана максимальная допустимая стоимость размещенных станций C .

Задан отрезок α длиной L с концами в точках a_0 и a_{n+1} . Внутри отрезка $\alpha = [a_0, a_{n+1}]$ задано множество возможных точек размещения станций множества $A = \{a_i\}, i = 1, \dots, n$ с координатами l_i . Точка a_0 имеет координату $l_0 = 0$, точка a_{n+1} имеет координату $l_{n+1} = L$. На концах отрезка, в вершинах a_0 и a_{n+1} , стоят станции специального вида s_0 и s_{m+1} , соответственно, для которых радиус-

сы покрытия, пропускные способности и стоимости не задаются. Радиусы связи задаются как R_{0j} и $R_{(m+1)j}$, соответственно. Требуется разместить станции таким образом, чтобы максимизировать размер контролируемой ими территории (покрытие) отрезка L при выполнении требования наличия связи каждой станции со станциями на концах отрезка (шлюзами) через систему размещенных станций при выполнении ограничений на время межконцевой задержки T и суммарную стоимость размещенных станций C . Сформулируем задачу в виде экстремальной задачи на конечном множестве.

Допустимой расстановкой станций назовем такой возрастающий по величине координат l_i набор пар $P = \{a_i, s_j\}, a_i \in A, i \neq 0, i \neq n+1; s_j \in S$, для которого выполняются требования:

1. для каждой пары (a_i, s_j) :

а) слева: либо найдется такая пара (a_k, s_q) , что, $l_i - l_k \leq R_{jq}$ и $l_i - l_k \leq R_{qj}$, либо $l_i - l_0 \leq R_{j0}$ и $l_i - l_0 \leq R_{0j}$;

б) справа: либо найдется такая пара (a_t, s_g) , что, $l_t - l_i \leq R_{jq}$ и $l_t - l_i \leq R_{qj}$, либо $l_{n+1} - l_i \leq R_{j(m+1)}$ и $l_{n+1} - l_i \leq R_{(m+1)j}$.

Данное требование гарантирует, что любая станция может быть связана со станциями на концах отрезка либо через промежуточные станции, либо непосредственно;

2. в одной точке стоит не более одной станции;

3. сумма задержек по всем размещенным станциям меньше заданной величины T – средней межконцевой задержки по времени по всей системе станций:

$$\sum_{j \in S_\sigma} \overline{T}_j \leq T,$$

где S_σ – множество размещенных станций, \overline{T}_j – среднее время задержки на станции. Расчет задержек описан в параграфе 4.2.5

4. суммарная стоимость размещенных станций меньше заданного бюджетного ограничения C .

Каждой допустимой расстановке станций P соответствует величина покрытия $z(P)$, определяемая как суммарная длина всех таких участков $\tau, \tau \subset \alpha$, что каждая точка этих участков попадает в зону покрытия, по крайней мере, одной станции, входящей в набор пар P .

Для удобства описания в дальнейшем алгоритмов введем понятие «недопокрытия» отрезка α :

$$f(P) = L - z(P)$$

Пусть G – множество всех допустимых расстановок P . Тогда мы можем сформулировать нашу задачу в следующей комбинаторной форме экстремальной задачи на конечном множестве.

Задача 1.

Требуется найти такую допустимую расстановку P^* , что

$$P^* = \operatorname{argmin}_{P \in G} f(P) \quad (4.1)$$

Обозначим через Γ все множество вариантов размещения станций (не обязательно допустимых) из множества S на заданном множестве возможных мест их размещения.

4.2 Дерево ветвлений для перебора элементов в множестве Γ

Опишем процедуру построения бинарного дерева поиска (дерева ветвлений) для полного перебора без повторений всех элементов множества Γ . Данная процедура будет использована в дальнейшем при построении дерева поиска в алгоритме МВиГ решения **задачи 1**.

В дальнейшем будем предполагать, что в множестве S станции упорядочены по не убыванию радиусов покрытия. Описываемая процедура использует известный прием разбиения множества G на подмножества с использованием некоторого параметра. Процесс формирования и последовательность исследования подмножеств обычно представляется с помощью дерева поиска, представляющего собой ориентированное от корня «дерево ветвлений», где каждому подмножеству соответствует вершина на дереве. Множеству Γ соответствует корневая вершина.

4.2.1 Параметр для разбиения множеств на подмножества

Процедура 1.

Пусть G_0 , где нижний индекс – номер итерации, исходное множество Γ . На каждой итерации, начиная с итерации $\mathbf{v} = 0$, разбиваем текущее подмножество $G_{\mathbf{v}}$ на два подмножества $G_{\mathbf{v}}^1$ и $G_{\mathbf{v}}^2$. При этом множество $G_{\mathbf{v}}$ обычно называется «материнским», а множества $G_{\mathbf{v}}^1$ и $G_{\mathbf{v}}^2$ – «потомками» множества $G_{\mathbf{v}}$ или дочерними узлами.

В качестве параметра разбиения воспользуемся переменной π_{ij} , принимающей два значения 0 и 1:

- $\pi_{ij} = 1$, если наложено условие, что на месте a_i расположена станция s_j ;
- $\pi_{ij} = 0$, если наложено условие, что на месте a_i станция s_j располагаться не будет.

В дальнейшем будем считать, что для множества $G_{\mathbf{v}}^1$ задано условие $\pi_{ij} = 1$, а для множества $G_{\mathbf{v}}^2$ задано условие $\pi_{ij} = 0$.

Очевидно, что

$$G_{\mathbf{v}}^1 \cup G_{\mathbf{v}}^2 = G_{\mathbf{v}}; \quad (4.2)$$

$$G_{\mathbf{v}}^1 \cap G_{\mathbf{v}}^2 = \emptyset. \quad (4.3)$$

Выбор переменной для разбиения на \mathbf{v} -ой итерации

На этапе разбиения любого множества $G_{\mathbf{v}}$ все множество переменных $\Pi = \{\pi_{ij}\}$ можно разделить на три подмножества: множество Π^+ – «фиксированные» переменные, для которых $\pi_{ij} = 1$, множество Π^- – «запрещенные» переменные, для которых $\pi_{ij} = 0$, и множество Π^f – «свободные» переменные, для которых значения на данной итерации еще не заданы.

Правило выбора переменной для разбиения множества $G_{\mathbf{v}}$. Для разбиения множества $G_{\mathbf{v}}$ на данной итерации выбирается из множества Π^f переменная с наименьшим индексом j среди всех переменных с наименьшим индексом i . Таким образом сначала определяется незанятое место размещения a_i с наименьшим номером (индексом i) и на нем размещается еще не размещенная станция s_j с наименьшим номером (индексом j).

4.2.2 Движение по дереву ветвлений

После разбиения очередного подмножества G_v два подмножества G_v^1 и G_v^2 , последним на дереве ветвлений присваиваются порядковые индексы G_{v+1} и G_{v+2} , соответственно. При формировании дерева ветвлений различаются два типа шагов: «прямой» шаг и «обратный» шаг. Прямой шаг – это движение «в глубину» по той же ветви дерева, реализующее очередное разбиение множества G_v на два потомка, и обратный шаг, реализующий переход от множества G_v к одному из ранее сформированных подмножеств. Обратный шаг делается в том случае, когда либо получено множество G_v , состоящее из единственного элемента, либо множество G_v при данном наборе значений переменных π_{ij} , выделяющих данное подмножество G_v из множества G_0 , пусто. В этих случаях соответствующая вершина дерева называется «закрытой».

Для движения по дереву будем использовать правило **LIFO**. На основании этого правила прямые шаги будут выполняться до тех пор, пока не будет получена закрытая вершина. На дереве ветвлений это соответствует продолжению движения по той же ветви дерева. При этом из двух множеств G_v^1 и G_v^2 первым будет исследоваться на возможность закрытия соответствующей вершины множество G_v^1 . Если вершина в результате проведенного исследования не будет закрыта, то из неё будет продолжено дальнейшее движение по той же ветви (выполнение прямого шага). Если вершина будет закрыта, то будет выполнен обратный шаг: для дальнейшего рассмотрения и продолжения движения будет выбрана незакрытая вершина с наибольшим порядковым номером v среди всех висячих вершин дерева (последняя сформированная вершина из нерассмотренных). Процедура будет завершена, когда все вершины дерева будут закрыты.

Заметим, что выполнение условий (4.2) и (4.3) гарантирует, что в результате завершения работы **процедуры 1** будут просмотрены все элементы множества Γ без повторений. Эти же условия определяют фундаментальное свойство дерева ветвлений: на каждой итерации объединение множеств G_v всех висячих вершин дерева дает исходное множество G_0 корневой вершины.

4.2.3 Алгоритм метода ветвей и границ

Для построения алгоритма МВиГ для решения **задачи 1** с использованием *процедуры 1* для построения дерева ветвлений нам достаточно разработать методы исследования вершин дерева на возможность их закрытия. В соответствии с техникой МВиГ закрытие вершины в результате исследования, соответствующего ей множества G_v возможно в трех случаях.

Случай 1. Множество G_v – пусто, т.е. доказано, что в множестве G_v при данном наборе фиксированных и запрещенных переменных π_{ij} нет ни одной допустимой расстановки P .

Случай 2. Доказано, что в множестве G_v не может быть допустимой расстановки P с меньшим значением целевой функции (1), чем у лучшей расстановки \hat{P} из уже найденных. Значение функции $f(\hat{P})$ называется «рекордом», а расстановка \hat{P} – «рекордным решением». В качестве начального рекорда принимается число заведомо большее искомого оптимального решения, например, L – длина всего отрезка.

Случай 3. Найдено оптимальное решение **задачи 1** на множестве G_v . Прежде чем рассмотреть эти три случая, запишем важное свойство любого множеств G_v , являющееся следствием принятого правила выбора свободной переменной для разбиения очередного множества G_v при прямом шаге.

Свойство 1. Пусть для исследуемого множества G_v , $v > 0$, точка a_k – это одно любое из мест, на которых уже размещены станция из множества S в соответствии с набором фиксированных и запрещенных переменных π_{ij} , выделяющим данное множество из множества G_0 . Тогда для всех мест «слева» от a_k , т.е. точек a_i , $i < k$, размещение станций уже определено (при этом некоторые места могут быть пустыми). Перейдем непосредственно к исследованию **случаев 1 – 3.**

Случай 1.

Проверка текущего множества G_v на пустоту состоит в установлении факта невозможности выполнения требований 1) – 4) введенных ранее при определении допустимой расстановки.

Рассмотрим проверку условия выполнения требования 1) для множества G_v , $v > 0$.

Пусть множество G_v образовано разбиением материнского множества при помощи переменной $\pi_{kt} = 1$. Проверяем, что каждый из радиусов R_{th} и R_{ht} , где h – индекс станции, размещенной на ближайшей слева к точке a_k точке a_d больше расстояния $l_k - l_d$. Если ближайшая слева точка – это точка a_0 (левый конец отрезка α), то делается проверка, для радиуса R_{t0} и R_{0t} .

Если данное условие не выполняется, то множество G_v недопустимо, соответствующая вершина закрывается и делается шаг обратного хода в соответствии с *процедурой 1*.

Если множество G_v образовано разбиением материнского множества при помощи переменной $\pi_{kt} = 0$ и a_d – точка с наибольшим индексом, среди точек, на которых уже размещены станции (точки a_0 , если размещенных станций нет), то надо проверить, что среди нераспределенных станций, без учета станции s_t , есть такая станция s_q что расстояние между точками a_k и a_d не больше, чем R_{qh} и R_{hq} . Если проверка отрицательна, то множество G_v – пусто, соответствующая этому множеству на дереве поиска вершина должна быть закрыта и выполняется шаг обратного хода в соответствии с *процедурой 1*.

Требование 2) выполняется соответствующим выбором очередной станции для размещения, требования 3) и 4) выполняются непосредственным суммированием соответствующих параметров у размещенных станций.

Случай 2. Построим оценку величины «недопокрытия» для множества G_v , полученного из материнского множества добавлением условия $\pi_{kt} = 1$. Частичным «недопокрытием» назовем величину $\Delta(k, d, p, t)$, которая вычисляется по формуле:

$$\Delta(k, d, p, t) = \max\{(a_k - a_d) - (r_p + r_t), 0\}. \quad (4.4)$$

Частичное «недопокрытие» (4.4) определяется для любых двух точек a_d и $a_k, k > d$, на которых расположены станции s_p и s_t при условии, что между этими точками нет других станций. Очевидно, что для любой расстановки P «недопокрытие» $f(P)$ вычисляется как сумма всех «недопокрытий» $\Delta(k, d, p, t)$ между местами размещения станций, включая концы отрезка α , на которых стоят станции особого типа s_0 и s_{m+1} .

$$W(G_v) \leq f(P), P \in G_v.$$

Если $W(G_v) \geq f(\hat{P})$, то множество G_v не может содержать расстановки лучше уже найденной расстановки \hat{P} соответствующая множеству G_v вершина на дереве поиска должна быть закрыта и далее выполняется шаг обратного хода в соответствии с *процедурой 1*.

Построим оценку «недопокрытия» для множества G_v , полученного из материнского множества добавлением условия $\pi_{kt} = 1$. Оценку будем искать в виде суммы

$$W(G_v) = w_1 + w_2.$$

Величина $w_1(G_v)$ вычисляется как сумма все частичных «недопокрытий» слева от вершины a_k и величины радиуса покрытия, размещаемой станций r_t . Оценку $w_2(G_v)$ вычислим «для недопокрытия» справа на части β до конца отрезка α (точки a_{n+1}). Данную оценку получим релаксацией условий, определяющих допустимую расстановку станций на участке β . Найдем такое подмножество S_β множества станций S , состоящее из еще не размещенных станций и дающее минимальное «недопокрытие» на участке β при выполнении только условий 2) – 4). Для этого сформулируем следующую задачу булевого программирования.

Задача 2.

$$z = |\beta| - \sum_{x_j \in S_\beta} r_j x_j \rightarrow \min.$$

при условии:

$$\sum_{x_j \in S_\beta} c_j x_j \leq C, \quad (4.5)$$

$$\sum_{x_j \in S_\beta} x_j \leq m, \quad (4.6)$$

$$x_j \in \{0, 1\},$$

где $|\beta|$ – длина отрезка отрезка β , m – число свободных мест для размещения станций на отрезке β .

Очевидно, что эффективность использования оценки в методе ветвей и границ определяется точностью оценки и временем ее вычисления. **Задача 2** – это задача ЦЛП, являющаяся труднорешаемой [2]. На основании **задачи 2**

можно получить две оценки менее точные, но имеющие более эффективные методы решения. Заметим, что при снятии ограничения (4.5) или (4.6) задача 2 представляет собой целочисленную задачу о ранце с эффективным псевдополиномиальным алгоритмом решения [2]. При этом с точки зрения точности оценки, более перспективным представляется снятие ограничения (4.6), так как на практике, обычно, число возможных мест размещения станций существенно меньше числа размещенных станций, полученного в результате решения задачи. Назовем задачу, полученную снятием ограничения (4.6), задачей 3.

Задачу 2 при снятии условия целочисленности на переменные назовем задачей 4. Задача 4 есть задача линейного программирования. Очевидно, что задачи 3 и задачи 4, являясь оценками целевой функции решения задачи 2, могут служить оценками $w_2(G_v)$. Результаты численного эксперимента с различными оценками вынесены в [приложение 2](#).

Если множество G_v получено из материнского добавлением условия $\pi_{kt} = 0$, то оценка $W(G_v)$ равна оценке материнского множества.

В [приложении 1](#) приведены результаты вычислительного эксперимента, показывающего время решения задач 2, 3, 4 и относительную точность задачи 3 и 4 по отношению к задаче 2.

Перейдем к рассмотрению случая 3. Рассматривается только для множеств G_v , состоящих из единственной расстановки P , для которой «недопокрытие» $f(P)$ вычисляется как сумма всех недопокрытий $\Delta(k, d, p, t)$ между местами, где размещены станций, включая концы отрезка α , на которых стоят станции s_0 и s_{m+1} .

Если для найденной расстановки P выполняются условия 1) – 4), которые для единственной расстановки легко проверяются, и

$$f(P) < f(\hat{P}), \quad (4.7)$$

то $f(P)$ принимается за новый рекорд $f(\hat{P})$, расстановка P становится новым рекордным решением \hat{P} и выполняется шаг обратного хода в соответствии с [Процедурой 1](#), если неравенство (4.7) не выполняется, то рекорд остается прежним и выполняется шаг обратного хода.

Работа алгоритма МВиГ заканчивается, когда все вершины дерева поиска закрыты, при этом решение задачи:

$$P^* = \hat{P}, f(P^*) = f(\hat{P}).$$

4.2.4 Построения последовательности топологий для итерационной процедуры моделирования **БШС**

В результате проектирования **БШС** надо найти ее оптимальную топологию среди всех топологий, для которых будут выполняться все требования к показателям, исследуемым и рассчитываемым на этапе моделировании сети. Для решения этой задачи воспользуемся идеей метода построения последовательности планов [3].

Рассмотрим **задачу 1**.

Требуется найти такую допустимую расстановку P^* , что

$$f(P^*) = \min\{f(P), P \in G\}.$$

Построим для этой задачи последовательность $\Gamma = P^1, P^2, \dots, P^k$ допустимых расстановок (решений) множества G для заданного k , где

$$\begin{aligned} f(P^1) &= f(P^*), \\ f(P^2) &= \text{extr}\{f(P), P \in G \setminus P^1\}, \\ &\dots \\ f(P^k) &= \text{extr}\{f(P), P \in G \setminus P^1 \cup P^2 \cup \dots \setminus P^{k-1}\}, \end{aligned}$$

В последовательности Γ каждое решение не лучше предыдущего и не хуже последующего.

Теперь воспользуемся следующей процедурой. Будем последовательно, начиная с первой расстановки, выполнять этап моделирования **БШС**. Очевидно, что как только мы получим расстановку, удовлетворяющую всем требованиям этапа моделирования, мы решим задачу нахождения оптимальной топологии среди всех топологий, для которых выполняются все требования к показателям, исследуемым и рассчитываемым на этапе моделировании сети. Действительно, для всех предыдущих расстановок эти условия не выполняются, а все последующие расстановки в последовательности Γ не могут быть лучше по критерию $f(P)$.

Обсудим вопрос как строить подобную последовательность на основании алгоритма **МВиГ**, описанного в параграфе 4.2.3. Заменив неравенство (4.7) на нестрогое и записывая все рекорды, полученные в процессе работы алгоритма, мы, очевидно, получим последовательность расстановок, где каждая

расстановка не хуже предыдущей и не лучше последующей. Для получения последовательности Γ достаточно «перевернуть» полученную последовательность, где первый элемент станет последним.

Недостатком такой процедуры является то, что для исследования на этапе моделирования будут отобраны только расстановки не хуже первого рекорда и среди них может не оказаться расстановки, удовлетворяющей критериям моделирования. Для расширения множества Γ можно сделать следующее. Зададим условие, что в результате решения **задачи 1** мы хотим получить не только оптимальное решение, но и все решения не хуже оптимального на величину d . Для решения такого варианта задачи достаточно неравенство (4.7) в алгоритме МВиГ заменить следующим неравенством

$$f(P) \leq f(\hat{P}) + d, \quad (4.8)$$

где $d = \varepsilon \cdot L > 0$, ε – заданное отклонение в процентах, и запоминать все рекорды, полученные в процессе решения задачи.

На основании неравенства (4.8) можно построить итерационную процедуру, увеличивая величину d , если при данном ее значении допустимого решения на этапе моделирования не найдено. В **приложение 2** представлены результаты численного примера.

4.2.5 Расчет межконцевой задержки

Одной из основных характеристик проектируемой сети является ее межконцевая задержка. Рассмотрим беспроводную сеть как сеть массового обслуживания (СеМО) с кросс-трафиком и с узлами $M/M/1$. По теореме Бурке [4] на выходе узла $M/M/1$, а значит на входе каждой последующей фазы тоже пуассоновский поток. Интенсивность на выходе каждой фазы равна суммарной интенсивности всех входящих потоков с интенсивностями λ .

По формуле Литтла [5] можно рассчитать время задержки на фазе. Интенсивность времени обслуживания рассчитывается по формуле:

$$\mu_j = p_j/w,$$

где: p_j - пропускная способность j -ой станции, Мбит/с; w - средний размер пакета, Мбит.

Для каждой станции коэффициент загрузки равен:

$$\rho_j = \frac{\sum \lambda}{\mu_j} = \frac{q \cdot \lambda}{\mu_j} < 1,$$

где q – число входящих потоков. Условие $\rho_j < 1$ является необходимым и достаточным условием существования стационарного режима функционирования **СеМО**.

Тогда среднее время задержки по времени на каждой станции:

$$\overline{T_j} = \frac{\rho_j}{1 - \rho_j} \cdot \frac{1}{q \cdot \lambda}.$$

Тогда межконцевая задержки в сети равна

$$T^{e2e} = \sum \overline{T_j}. \quad (4.9)$$

4.2.6 Выводы по Главе 4

В работе предложена методика проектирования беспроводной широкополосной сети для контроля линейной трассы с использованием итерационной процедуры построения последовательности лучших решений задачи выбора и размещения базовых станций при выполнении технологических условий на проектирование сети и ограничения на стоимость размещаемых станций.

Предложенная методика позволяет на этапе моделирования выбирать лучшее решение среди тех решений по выбору и размещению станций, которые удовлетворяют требованиям, предъявляемым к проектируемой сети.

Процедура нахождения последовательности лучших решений задачи выбора и размещения базовых станций основана на разработанном алгоритме **МВиГ**.

Заключение

Основные результаты работы заключаются в следующем.

1. На основе анализа ...
2. Численные исследования показали, что ...
3. Математическое моделирование показало ...
4. Для выполнения поставленных задач был создан ...

И какая-нибудь заключающая фраза.

Последний параграф может включать благодарности. В заключение автор выражает благодарность и большую признательность научному руководителю Иванову И. И. за поддержку, помощь, обсуждение результатов и научное руководство. Также автор благодарит Сидорова А. А. и Петрова Б. Б. за помощь в работе с образцами, Рабиновича В. В. за предоставленные образцы и обсуждение результатов, Занудятину Г. Г. и авторов шаблона *Russian-Phd-LaTeX-Dissertation-Template* за помощь в оформлении диссертации. Автор также благодарит много разных людей и всех, кто сделал настоящую работу автора возможной.

Список сокращений и условных обозначений

$\left. \begin{matrix} a_n \\ b_n \end{matrix} \right\}$	и снова коэффициенты разложения Ми в дальнем поле соответствующие электрическим и магнитным мультиполям. Добавлено много текста, так что описание группы условных обозначений значительно превысило высоту этой группы...
$\left. \begin{matrix} a_n \\ b_n \end{matrix} \right\}$	коэффициенты разложения Ми в дальнем поле соответствующие электрическим и магнитным мультиполям
$\left. \begin{matrix} \mathbf{N}_{eln}^{(j)} & \mathbf{N}_{o1n}^{(j)} \\ \mathbf{M}_{o1n}^{(j)} & \mathbf{M}_{eln}^{(j)} \end{matrix} \right\}$	сферические векторные гармоники
λ	длина волны электромагнитного излучения в вакууме
μ	магнитная проницаемость в вакууме
ω	частота падающей волны
$\hat{\mathbf{e}}$	единичный вектор
E_0	амплитуда падающего поля
j	тип функции Бесселя
k	волновой вектор падающей волны
L	общее число слоёв
l	номер слоя внутри стратифицированной сферы
n	порядок мультиполя
r, θ, φ	полярные координаты
ω	частота света, см. (1.12)
c	скорость света, стр. 19
k	модуль волнового вектора, см. (1.12), стр. 19
BEM	boundary element method, метод граничных элементов
CST MWS	Computer Simulation Technology Microwave Studio программа для компьютерного моделирования уравнений Максвелла
DDA	discrete dipole approximation, приближение дискретных диполей
FDFD	finite difference frequency domain, метод конечных разностей в частотной области

FDTD	finite difference time domain, метод конечных разностей во временной области
FEM	finite element method, метод конечных элементов
FIT	finite integration technique, метод конечных интегралов
FMM	fast multipole method, быстрый метод многополюсника
FVTD	finite volume time-domain, метод конечных объёмов во временной области
MLFMA	multilevel fast multipole algorithm, многоуровневый быстрый алгоритм многополюсника
MoM	method of moments, метод моментов
MSTM	multiple sphere T-Matrix, метод T-матриц для множества сфер
PSTD	pseudospectral time domain method, псевдоспектральный метод во временной области
TLM	transmission line matrix method, метод матриц линий передач

Словарь терминов

TeX : Система компьютерной вёрстки, разработанная американским профессором информатики Дональдом Кнудом

панграмма : Короткий текст, использующий все или почти все буквы алфавита

Список литературы

1. *Вишневский, В. М.* Теоретические основы проектирования компьютерных сетей [Текст] / В. М. Вишневский. — Москва : Техносфера, 2003. — 512 с.
2. *М, Г.* Вычислительные машины и труднорешаемые задачи [Текст] / Г. М, Д. Д. — Мир, 1982. — 416 с.
3. *Емеличев, В. А.* Метод построения последовательности планов для решения задач дискретной оптимизации [Текст] / В. А. Емеличев, В. Комлик. — Москва : Наука, 1981. — 208 с.
4. *Burke, P. J.* The Output of a Queuing System [Текст] / P. J. Burke // Operations Research. — 1956. — Т. 4, № 6. — С. 699—704.
5. *Little, J. D. C.* A Proof for the Queuing Formula: $L = \lambda W$ [Текст] / J. D. C. Little // Operations Research. — 1961. — Т. 9, № 3. — С. 383—387.

Список рисунков

1.1	Соединение между станциями.	10
1.2	Покрытие станции	10
2.1	Координаты размещения	19
2.2	Координаты размещения	20
2.3	Координаты размещения	21
3.1	Охват покрытия станции]	23
3.2	Область покрытия между любыми двумя точками	24
3.3	Связь между базовыми станциями	26
3.4	Обеспечение связи с соседней станцией	27

Список таблиц

1	Координаты размещения	18
2	Координаты размещения	18
3	Множество типов станций	18
4	Множество типов станций	20
5	Точки размещения участка с длиной $L = 300$	28
6	Параметры базовых станций, шлюзов и объектов.	28
7	Рассчитанные радиусы связи между станциями	29
8	Рассчитанные радиусы покрытия станций	30
9	Решение задачи ЦЛП.	30
10	Наименование таблицы средней длины	58
11	Тестовые функции для оптимизации, D — размерность. Для всех функций значение в точке глобального минимума равно нулю.	63
12	Длинная таблица с примером чересстрочного форматирования	66
13	Стандартные префиксы ссылок	68

Приложение А

Примеры вставки листингов программного кода

Для крупных листингов есть два способа. Первый красивый, но в нём могут быть проблемы с поддержкой кириллицы (у вас может встречаться в комментариях и печатаемых сообщениях), он представлен на листинге [А.1](#). Второй

Листинг А.1: Программа „Hello, world“ на C++

```

5 | #include <iostream>
   | using namespace std;
   |
   | int main() //кириллица в комментариях при xelatex и luaLatex и
   | мееет проблемы с пробелами
   | {
   |     cout << "Hello, world" << endl; //latin letters in
   |     commentaries
   |     system("pause");
   |     return 0;
10 | }

```

не такой красивый, но без ограничений (см. листинг [А.2](#)).

Листинг А.2: Программа „Hello, world“ без подсветки

```

#include <iostream>
using namespace std;

int main() //кириллица в комментариях
{
    cout << "Привет, мир" << endl;
}

```

Можно использовать первый для вставки небольших фрагментов внутри текста, а второй для вставки полного кода в приложении, если таковое имеется.

Если нужно вставить совсем короткий пример кода (одна или две строки), то выделение линейками и нумерация может смотреться чересчур громоздко.

В таких случаях можно использовать окружения `lstlisting` или `Verb` без `ListingEnv`. Приведём такой пример с указанием языка программирования, отличного от заданного по умолчанию:

```
|fibs = 0 : 1 : zipWith (+) fibs (tail fibs)
```

Такое решение — со вставкой нумерованных листингов покрупнее и вставок без выделения для маленьких фрагментов — выбрано, например, в книге Эндрю Таненбаума и Тодда Остина по архитектуре компьютера.

Наконец, для оформления идентификаторов внутри строк (функция `main` и тому подобное) используется `lstinline` или, самое простое, моноширинный текст (`\texttt`).

Пример A.3, иллюстрирующий подключение переопределённого языка. Может быть полезным, если подсветка кода работает криво. Без дополнительного окружения, с подписью и ссылкой, реализованной встроенным средством.

Листинг A.3: Пример листинга с подписью собственными средствами

```
## Caching the Inverse of a Matrix

## Matrix inversion is usually a costly computation and there
## may be some
## benefit to caching the inverse of a matrix rather than
## compute it repeatedly
5 ## This is a pair of functions that cache the inverse of a
## matrix.

## makeCacheMatrix creates a special "matrix" object that can
## cache its inverse

makeCacheMatrix <- function(x = matrix()) {#кириллица в коммента
10   rиях npx xelatex u luaLatex имеет проблемы с пробелами
    i <- NULL
    set <- function(y) {
        x <- y
        i <- NULL
    }
15   get <- function() x
    setSolved <- function(solve) i <- solve
    getSolved <- function() i
    list(set = set, get = get,
        setSolved = setSolved,
20   getSolved = getSolved)
```

```

}

25 ## cacheSolve computes the inverse of the special "matrix"
    returned by
    ## makeCacheMatrix above. If the inverse has already been
    calculated (and the
    ## matrix has not changed), then the cachesolve should retrieve
    the inverse from
    ## the cache.

30 cacheSolve <- function(x, ...) {
    ## Return a matrix that is the inverse of 'x'
    i <- x$getSolved()
    if(!is.null(i)) {
        message("getting cached data")
35         return(i)
    }
    data <- x$get()
    i <- solve(data, ...)
    x$setSolved(i)
40     i
}

```

Листинг A.4 подгружается из внешнего файла. Приходится загружать без окружения дополнительного. Иначе по страницам не переносится.

Листинг A.4: Листинг из внешнего файла

```

# Analysis of data on Course Project at Getting and Cleaning
  data course of Data Science track at Coursera.

# Part 1. Merges the training and the test sets to create one
  data set.
# 3. Uses descriptive activity names to name the activities in
  the data set
5 # 4. Appropriately labels the data set with descriptive variable
    names.

if (!file.exists("UCI HAR Dataset")) {
    stop("You need 'UCI HAR Dataset' folder full of data")
}
10

```

```

library(plyr) # for mapvalues

15 #getting common data
features <- read.csv("UCI HAR Dataset/features.txt", sep=" ",
  header = FALSE,
                        colClasses = c("numeric", "character"))
activity_labels <- read.csv("UCI HAR Dataset/activity_labels.txt",
  sep="",
                        header = FALSE, colClasses = c("
  numeric", "character"))

20 #getting train set data
subject_train <- read.csv("UCI HAR Dataset/train/subject_train.
  txt",
                        header = FALSE, colClasses = "numeric",
                        col.names="Subject")
y_train <- read.csv("UCI HAR Dataset/train/y_train.txt", header
  = FALSE,
25                        colClasses = "numeric")
x_train <- read.csv("UCI HAR Dataset/train/X_train.txt", sep="",
  header = FALSE,
                        colClasses = "numeric", col.names=features$V2
                        , check.names = FALSE)

activity_train <- as.data.frame(mapvalues(y_train$V1, from =
  activity_labels$V1,
30                        to = activity_labels$
  V2))
names(activity_train) <- "Activity"

35 #getting test set data
subject_test <- read.csv("UCI HAR Dataset/test/subject_test.txt"
  ,
                        header = FALSE, colClasses = "numeric",
                        col.names="Subject")
y_test <- read.csv("UCI HAR Dataset/test/y_test.txt", header =
  FALSE,
40                        colClasses = "numeric")
x_test <- read.csv("UCI HAR Dataset/test/X_test.txt", sep="",
  header = FALSE,

```

```

        colClasses = "numeric", col.names=features$V2,
        check.names = FALSE)

activity_test <- as.data.frame(mapvalues(y_test$V1, from =
        activity_labels$V1,
                                                to = activity_labels$V2
        ))
45 names(activity_test) <- "Activity"

# Forming full dataframe
data_train <- cbind(x_train, subject_train, activity_train)
50 data_test <- cbind(x_test, subject_test, activity_test)
data <- rbind(data_train, data_test)

# Cleaning memory
rm(features, activity_labels, subject_train, y_train, x_train,
        activity_train,
55 subject_test, y_test, x_test, activity_test, data_train, data
        _test)

# Part 2. Extracts only the measurements on the mean and
        standard deviation for each measurement.

60 cols2match <- grep("(mean|std)", names(data))

# Excluded gravityMean, tBodyAccMean, tBodyAccJerkMean,
        tBodyGyroMean,
        tBodyGyroJerkMean, as these represent derivations of angle
        data, as
        opposed to the original feature vector.
65

# Subsetting data frame, also moving last columns to be first
Subsetted_data_frame <- data[, c(562, 563, cols2match)]

# Part 5. From the data set in step 4, creates a second,
        independent tidy data set
70 # with the average of each variable for each activity and each
        subject.

library(dplyr) # for %>% and summarise_each

```

```
75 tidydata <- Subsetted_data_frame %>% group_by(Subject,Activity)
    %>%
        summarise_each(funs(mean))

write.table(tidydata, "tidydata.txt", row.names=FALSE)
```

Приложение Б

Очень длинное название второго приложения, в котором продемонстрирована работа с длинными таблицами

Б.1 Подраздел приложения

Вот размещается длинная таблица:

Параметр	Умолч.	Тип	Описание
&INP			
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$)

продолжение следует

(продолжение)			
Параметр	Умолч.	Тип	Описание
mars	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
&SURFPAR			
kick	1	int	0: инициализация без шума ($p_s = const$)
	0	int	1: генерация белого шума
mars	0	int	2: генерация белого шума симметрично относительно экватора
	1	int	1: инициализация модели для планеты Марс
kick	0	int	0: инициализация без шума ($p_s = const$)
	1	int	1: генерация белого шума
mars	0	int	2: генерация белого шума симметрично относительно экватора
	1	int	1: инициализация модели для планеты Марс
kick	0	int	0: инициализация без шума ($p_s = const$)
	1	int	1: генерация белого шума
mars	0	int	2: генерация белого шума симметрично относительно экватора
	1	int	1: инициализация модели для планеты Марс
kick	0	int	0: инициализация без шума ($p_s = const$)
	1	int	1: генерация белого шума

продолжение следует

(продолжение)			
Параметр	Умолч.	Тип	Описание
mars kick	0	int	2: генерация белого шума симметрично относительно экватора
	1	int	1: инициализация модели для планеты Марс 0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс

Б.2 Ещё один подраздел приложения

Нужно больше подразделов приложения! Конвынёры витюпырата но нам, тебиквьюэ мэнтётюм позтюлант ед про. Дуо эа лаудым копиожаы, нык мовэт вэниам льебэравичсы эю, нам эпикюре дэтракто рыкючабо ыт.

Пример длинной таблицы с записью продолжения по ГОСТ 2.105:

Таблица 10 — Наименование таблицы средней длины

Параметр	Умолч.	Тип	Описание
&INP			
kick	1	int	0: инициализация без шума ($p_s = const$)
			1: генерация белого шума
mars kick	0	int	2: генерация белого шума симметрично относительно экватора
	1	int	1: инициализация модели для планеты Марс
mars kick	1	int	0: инициализация без шума ($p_s = const$)
			1: генерация белого шума
mars kick	1	int	2: генерация белого шума симметрично относительно экватора
			1: инициализация модели для планеты Марс

Продолжение таблицы 10

Параметр	Умолч.	Тип	Описание
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора

Продолжение таблицы 10

Параметр	Умолч.	Тип	Описание
			экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
&SURFPAR			
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$)

Продолжение таблицы 10

Параметр	Умолч.	Тип	Описание
			1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс

Б.3 Использование длинных таблиц с окружением *longtabu*

В таблице 11 более книжный вариант длинной таблицы, используя окружение `longtabu` и разнообразные `toprule` `midrule` `bottomrule` из пакета `booktabs`. Чтобы визуально таблица смотрелась лучше, можно использовать следующие параметры: в самом начале задаётся расстояние между строчками с помощью `arraystretch`. Таблица задаётся на всю ширину, `longtabu` позволяет делить ширину колонок пропорционально — тут три колонки в пропорции 1.1:1:4 — для каждой колонки первый параметр в описании `X[]`. Кроме того, в таблице убраны отступы слева и справа с помощью `@{}` в преамбуле таблицы. К первому и второму столбцу применяется модификатор

`>\setlength{\baselineskip}{0.7\baselineskip}`,

который уменьшает межстрочный интервал в для текста таблиц (иначе заголовки второго столбца значительно шире, а двухстрочное имя сливается с окружающими). Для первой и второй колонки текст в ячейках выравнивается по центру как по вертикали, так и по горизонтали — задаётся буквами `m` и `c` в описании столбца `X[]`.

Так как формулы большие — используется окружение `alignedat`, чтобы отступ был одинаковый у всех формул — он сделан для всех, хотя для большей части можно было и не использовать. Чтобы формулы занимали поменьше места в каждом столбце формулы (где надо) используется `\textstyle` — он делает дроби меньше, у знаков суммы и произведения — индексы сбоку. Иногда формула слишком большая, сливается со следующей, поэтому после неё ставится небольшой дополнительный отступ `\vspace*{2ex}`. Для штрафных функций — размер фигурных скобок задан вручную `\Big\{`, т. к. не умеет `alignedat` работать с `\left` и `\right` через несколько строк/колонок.

В примечании к таблице наоборот, окружение `cases` даёт слишком большие промежутки между вариантами, чтобы их уменьшить, в конце каждой строчки окружения использовался отрицательный дополнительный отступ `\[-0.5em]`.

Таблица 11 — Тестовые функции для оптимизации, D — размерность. Для всех функций значение в точке глобального минимума равно нулю.

Имя	Стартовый диапазон параметров	Функция
сфера	$[-100, 100]^D$	$f_1(x) = \sum_{i=1}^D x_i^2$
Schwefel 2.22	$[-10, 10]^D$	$f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $
Schwefel 1.2	$[-100, 100]^D$	$f_3(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$
Schwefel 2.21	$[-100, 100]^D$	$f_4(x) = \max_i \{ x_i \}$
Rosenbrock	$[-30, 30]^D$	$f_5(x) = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$
ступенчатая	$[-100, 100]^D$	$f_6(x) = \sum_{i=1}^D \lfloor x_i + 0.5 \rfloor^2$
зашумлённая квартиче- ская	$[-1.28, 1.28]^D$	$f_7(x) = \sum_{i=1}^D ix_i^4 + rand[0,1)$
Schwefel 2.26	$[-500, 500]^D$	$f_8(x) = \sum_{i=1}^D -x_i \sin \sqrt{ x_i } +$ $+ D \cdot 418.98288727243369$
Rastrigin	$[-5.12, 5.12]^D$	$f_9(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$
Ackley	$[-32, 32]^D$	$f_{10}(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) -$ $-\exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$
Griewank	$[-600, 600]^D$	$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(x_i/\sqrt{i}) + 1$
штрафная 1	$[-50, 50]^D$	$f_{12}(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \right.$ $\left. + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + \right.$ $\left. + (y_D - 1)^2 \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$

продолжение следует

(продолжение)

Имя	Стартовый диапазон параметров	Функция
штрафная 2	$[-50, 50]^D$	$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \right.$ $+ \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] +$ $+ (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \left. \right\} +$ $+ \sum_{i=1}^D u(x_i, 5, 100, 4)$
сфера	$[-100, 100]^D$	$f_1(x) = \sum_{i=1}^D x_i^2$
Schwefel 2.22	$[-10, 10]^D$	$f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $
Schwefel 1.2	$[-100, 100]^D$	$f_3(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$
Schwefel 2.21	$[-100, 100]^D$	$f_4(x) = \max_i \{ x_i \}$
Rosenbrock	$[-30, 30]^D$	$f_5(x) = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$
ступенчатая	$[-100, 100]^D$	$f_6(x) = \sum_{i=1}^D \lfloor x_i + 0.5 \rfloor^2$
зашумлённая квартиче- ская	$[-1.28, 1.28]^D$	$f_7(x) = \sum_{i=1}^D i x_i^4 + rand[0, 1)$
Schwefel 2.26	$[-500, 500]^D$	$f_8(x) = \sum_{i=1}^D -x_i \sin \sqrt{ x_i } +$ $+ D \cdot 418.98288727243369$
Rastrigin	$[-5.12, 5.12]^D$	$f_9(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$
Ackley	$[-32, 32]^D$	$f_{10}(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) -$ $- \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e$
Griewank	$[-600, 600]^D$	$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(x_i / \sqrt{i}) + 1$

продолжение следует

(окончание)

Имя	Стартовый диапазон параметров	Функция
штрафная 1	$[-50, 50]^D$	$f_{12}(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \right.$ $\left. + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + \right.$ $\left. + (y_D - 1)^2 \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$
штрафная 2	$[-50, 50]^D$	$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \right.$ $\left. + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + \right.$ $\left. + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \right\} +$ $+ \sum_{i=1}^D u(x_i, 5, 100, 4)$
Примечание — Для функций f_{12} и f_{13} используется $y_i = 1 + \frac{1}{4}(x_i + 1)$ и $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$		

Б.4 Форматирование внутри таблиц

В таблице 12 пример с чересстрочным форматированием. В файле `userstyles.tex` задаётся счётчик `\newcounter{rowcnt}` который увеличивается на 1 после каждой строчки (как указано в преамбуле таблицы). Кроме того, задаётся условный макрос `\altshape` который выдаёт одно из двух типов форматирования в зависимости от чётности счётчика.

В таблице 12 каждая чётная строчка — синяя, нечётная — с наклоном и слегка поднята вверх. Визуально это приводит к тому, что среднее значение и среднеквадратичное изменение группируются и хорошо выделяются взглядом в таблице. Сохраняется возможность отдельные значения в таблице выделить цветом или шрифтом. К первому и второму столбцу форматирование не применяется по сути таблицы, к шестому общее форматирование не применяется для наглядности.

Так как заголовок таблицы тоже считается за строчку, то перед ним (для первого, промежуточного и финального варианта) счётчик обнуляется, а в `\altshape` для нулевого значения счётчика форматирования не применяется.

Таблица 12 — Длинная таблица с примером чересстрочного форматирования

	Итера- ции	JADE++	JADE	jDE	SaDE	DE/rand /1/bin	PSO
f1	1500	1.8E-60 (8.4E-60)	1.3E-54 (9.2E-54)	2.5E-28 (3.5E-28)	4.5E-20 (6.9E-20)	9.8E-14 (8.4E-14)	9.6E-42 (2.7E-41)
f2	2000	1.8E-25 (8.8E-25)	3.9E-22 (2.7E-21)	1.5E-23 (1.0E-23)	1.9E-14 (1.1E-14)	1.6E-09 (1.1E-09)	9.3E-21 (6.3E-20)
f3	5000	5.7E-61 (2.7E-60)	6.0E-87 (1.9E-86)	5.2E-14 (1.1E-13)	9.0E-37 (5.4E-36)	6.6E-11 (8.8E-11)	2.5E-19 (3.9E-19)
f4	5000	8.2E-24 (4.0E-23)	4.3E-66 (1.2E-65)	1.4E-15 (1.0E-15)	7.4E-11 (1.8E-10)	4.2E-01 (1.1E+00)	4.4E-14 (9.3E-14)
f5	3000	8.0E-02 (5.6E-01)	3.2E-01 (1.1E+00)	1.3E+01 (1.4E+01)	2.1E+01 (7.8E+00)	2.1E+00 (1.5E+00)	2.5E+01 (3.2E+01)
f6	100	2.9E+00 (1.2E+00)	5.6E+00 (1.6E+00)	1.0E+03 (2.2E+02)	9.3E+02 (1.8E+02)	4.7E+03 (1.1E+03)	4.5E+01 (2.4E+01)
f7	3000	6.4E-04 (2.5E-04)	6.8E-04 (2.5E-04)	3.3E-03 (8.5E-04)	4.8E-03 (1.2E-03)	4.7E-03 (1.2E-03)	2.5E-03 (1.4E-03)
f8	1000	3.3E-05 (2.3E-05)	7.1E+00 (2.8E+01)	7.9E-11 (1.3E-10)	4.7E+00 (3.3E+01)	5.9E+03 (1.1E+03)	2.4E+03 (6.7E+02)
f9	1000	1.0E-04 (6.0E-05)	1.4E-04 (6.5E-05)	1.5E-04 (2.0E-04)	1.2E-03 (6.5E-04)	1.8E+02 (1.3E+01)	5.2E+01 (1.6E+01)
f10	500	8.2E-10 (6.9E-10)	3.0E-09 (2.2E-09)	3.5E-04 (1.0E-04)	2.7E-03 (5.1E-04)	1.1E-01 (3.9E-02)	4.6E-01 (6.6E-01)
f11	500	9.9E-08 (6.0E-07)	2.0E-04 (1.4E-03)	1.9E-05 (5.8E-05)	7.8E-04 (1.2E-03)	2.0E-01 (1.1E-01)	1.3E-02 (1.7E-02)
f12	500	4.6E-17 (1.9E-16)	3.8E-16 (8.3E-16)	1.6E-07 (1.5E-07)	1.9E-05 (9.2E-06)	1.2E-02 (1.0E-02)	1.9E-01 (3.9E-01)
f13	500	2.0E-16 (6.5E-16)	1.2E-15 (2.8E-15)	1.5E-06 (9.8E-07)	6.1E-05 (2.0E-05)	7.5E-02 (3.8E-02)	2.9E-03 (4.8E-03)
f1	1500	1.8E-60 (8.4E-60)	1.3E-54 (9.2E-54)	2.5E-28 (3.5E-28)	4.5E-20 (6.9E-20)	9.8E-14 (8.4E-14)	9.6E-42 (2.7E-41)

продолжение следует

(окончание)

Итера- ции		JADE++	JADE	jDE	SaDE	DE/rand /1/bin	PSO
f2	2000	1.8E-25 (8.8E-25)	3.9E-22 (2.7E-21)	1.5E-23 (1.0E-23)	1.9E-14 (1.1E-14)	1.6E-09 (1.1E-09)	9.3E-21 (6.3E-20)
f3	5000	5.7E-61 (2.7E-60)	6.0E-87 (1.9E-86)	5.2E-14 (1.1E-13)	9.0E-37 (5.4E-36)	6.6E-11 (8.8E-11)	2.5E-19 (3.9E-19)
f4	5000	8.2E-24 (4.0E-23)	4.3E-66 (1.2E-65)	1.4E-15 (1.0E-15)	7.4E-11 (1.8E-10)	4.2E-01 (1.1E+00)	4.4E-14 (9.3E-14)
f5	3000	8.0E-02 (5.6E-01)	3.2E-01 (1.1E+00)	1.3E+01 (1.4E+01)	2.1E+01 (7.8E+00)	2.1E+00 (1.5E+00)	2.5E+01 (3.2E+01)
f6	100	2.9E+00 (1.2E+00)	5.6E+00 (1.6E+00)	1.0E+03 (2.2E+02)	9.3E+02 (1.8E+02)	4.7E+03 (1.1E+03)	4.5E+01 (2.4E+01)
f7	3000	6.4E-04 (2.5E-04)	6.8E-04 (2.5E-04)	3.3E-03 (8.5E-04)	4.8E-03 (1.2E-03)	4.7E-03 (1.2E-03)	2.5E-03 (1.4E-03)
f8	1000	3.3E-05 (2.3E-05)	7.1E+00 (2.8E+01)	7.9E-11 (1.3E-10)	4.7E+00 (3.3E+01)	5.9E+03 (1.1E+03)	2.4E+03 (6.7E+02)
f9	1000	1.0E-04 (6.0E-05)	1.4E-04 (6.5E-05)	1.5E-04 (2.0E-04)	1.2E-03 (6.5E-04)	1.8E+02 (1.3E+01)	5.2E+01 (1.6E+01)
f10	500	8.2E-10 (6.9E-10)	3.0E-09 (2.2E-09)	3.5E-04 (1.0E-04)	2.7E-03 (5.1E-04)	1.1E-01 (3.9E-02)	4.6E-01 (6.6E-01)
f11	500	9.9E-08 (6.0E-07)	2.0E-04 (1.4E-03)	1.9E-05 (5.8E-05)	7.8E-04 (1.2E-03)	2.0E-01 (1.1E-01)	1.3E-02 (1.7E-02)
f12	500	4.6E-17 (1.9E-16)	3.8E-16 (8.3E-16)	1.6E-07 (1.5E-07)	1.9E-05 (9.2E-06)	1.2E-02 (1.0E-02)	1.9E-01 (3.9E-01)
f13	500	2.0E-16 (6.5E-16)	1.2E-15 (2.8E-15)	1.5E-06 (9.8E-07)	6.1E-05 (2.0E-05)	7.5E-02 (3.8E-02)	2.9E-03 (4.8E-03)

Б.5 Стандартные префиксы ссылок

Общепринятым является следующий формат ссылок: <prefix>:<label>. Например, \label{fig:knuth}; \ref{tab:test1}; label={lst:external1}. В таблице 13 приведены стандартные префиксы для различных типов ссылок.

Таблица 13 — Стандартные префиксы ссылок

Префикс	Описание
ch:	Глава
sec:	Секция
subsec:	Подсекция
fig:	Рисунок
tab:	Таблица
eq:	Уравнение
lst:	Листинг программы
itm:	Элемент списка
alg:	Алгоритм
app:	Секция приложения

Для упорядочивания ссылок можно использовать разделительные символы. Например, `\label{fig:scheemes/my_scheeme}` или `\label{lst:dts/linked_list}`.

Б.6 Очередной подраздел приложения

Нужно больше подразделов приложения!

Б.7 И ещё один подраздел приложения

Нужно больше подразделов приложения!

