МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

КУРСОВАЯ РАБОТА (ПРОЕКТ)   
ЗАЩИЩЕНА С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОМУ ПРОЕКТУ |
| Методы условной оптимизации. |
| по дисциплине: ПРИКЛАДНЫЕ МОДЕЛИ ОПТИМИЗАЦИИ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134К |  |  |  | Д. В. Самарин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Оглавление

[Введение 3](#_Toc159239311)

[Глава 1: Теоретические основы линейного программирования 6](#_Toc159239312)

[Глава 2: Расчётно-аналитический аспект задач линейного программирования 10](#_Toc159239313)

[Заключение 19](#_Toc159239314)

[Источники 20](#_Toc159239315)

# Введение

Оптимизация играет ключевую роль в современном мире, пронизывая множество аспектов жизни, от промышленности до науки и финансов. Основным вопросом при решении многих задач является поиск оптимальных решений с учетом различных ограничений, таких как физические ограничения, бюджетные ограничения или требования к качеству. Именно в этом контексте становится важным понятие условной оптимизации.

**Актуальность**

В современном мире, на фоне быстрого технологического развития и увеличения сложности задач, стоящих перед бизнесом, наукой и промышленностью, вопросы оптимизации приобретают все большее значение. Методы условной оптимизации становятся неотъемлемой частью решения многих реальных проблем, где оптимальное решение должно соответствовать сложным системам ограничений.

В бизнесе условная оптимизация используется для решения задач финансового планирования, оптимизации производственных процессов, логистики и управления рисками. В науке она применяется для моделирования и анализа систем, а также для настройки параметров сложных моделей. В инженерии методы условной оптимизации помогают в проектировании систем и устройств с учетом многочисленных ограничений, таких как физические, технические и экологические.

С развитием компьютерных технологий и доступности вычислительных ресурсов появляются новые возможности для применения методов условной оптимизации в реальных задачах. Вместе с тем, развитие новых методов и алгоритмов в этой области продолжает оставаться актуальной задачей для научных исследований.

Таким образом, изучение методов условной оптимизации имеет не только теоретическое, но и практическое значение, способствуя решению множества актуальных проблем в различных областях человеческой деятельности.Начало формы

**Цель**

Целью данной курсовой работы является изучение методов условной оптимизации и их прикладных моделей с целью разработки компьютерного кода, способного эффективно решать задачи условной оптимизации в различных областях применения.

**Задачи**

Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить основные понятия и принципы линейного программирования.

2. Провести анализ методов решения задач линейного программирования.

3. Разработать математическую модель конкретной задачи условной оптимизации и трансформировать её в формат, пригодный для программной реализации. Определить необходимые переменные, целевую функцию и ограничения, а также подготовить данные для входа в алгоритм.

4. Написать программный код, реализующий выбранные методы решения задачи условной оптимизации на основе разработанной математической модели. Провести тестирование кода с использованием различных входных данных для оценки корректности работы алгоритмов и анализа их производительности.

**Объект и предмет исследования**

Объектом исследования являются методы условной оптимизации в различных областях человеческой деятельности. Предметом исследования является разработка и анализ методов решения задач условной оптимизации с использованием программной реализации.

**Теоретическая основа и методы**

Теоретической основой исследования являются основные концепции и методы условной оптимизации, такие как методы градиентного спуска, методы внутренней точки, методы штрафных функций и другие. Исследование опирается на работы таких авторов, как Н. Зобе, С. Бойда, Л. Вандерберге и др., которые внесли значительный вклад в развитие теории и практики условной оптимизации.

**Новизна и методы исследования**

В ходе исследования были использованы следующие методы:

1. Методы математического анализа и моделирования для разработки математической модели задачи условной оптимизации.
2. Программирование и алгоритмическая реализация для создания программного кода, решающего задачу оптимизации.
3. Методы численного анализа для проведения численных экспериментов и анализа результатов.

Новизной данного исследования является:

* Разработка и адаптация программного кода для решения конкретной задачи условной оптимизации с использованием современных методов и алгоритмов.
* Практическое применение разработанных методов и программного обеспечения для решения реальных задач оптимизации в различных областях, таких как инженерия, экономика, управление и другие.

Таким образом, результаты исследования имеют практическую значимость для применения в различных сферах деятельности и способствуют расширению знаний и понимания методов условной оптимизации.

**Практическая значимость**

Результаты данного исследования обладают высокой практической значимостью, так как предоставляют новые методы и программные решения для эффективного решения практических задач оптимизации в различных областях. Вот несколько конкретных областей, где можно использовать основные положения, выводы и результаты исследования:

1. **Промышленность и производство:** разработанные методы и программные решения могут быть применены для оптимизации процессов производства, планирования производственных мощностей, распределения ресурсов и оптимизации логистических цепочек.
2. **Финансы и экономика:** исследование может быть использовано для оптимизации портфеля инвестиций, управления рисками, оптимизации расходов и максимизации прибыли в условиях различных финансовых ограничений.
3. **Транспорт и логистика:** разработанные методы могут помочь в оптимизации маршрутов транспорта, планировании графиков перевозок, управлении запасами и минимизации затрат на логистику.
4. **Энергетика и ресурсоснабжение:** результаты исследования могут быть применены для оптимизации работы энергетических систем, распределения ресурсов, планирования генерации и расхода энергии с учетом различных ограничений и целей.

# Глава 1: Методы условной оптимизации, теоретические основы

**Параграф 1.1: Введение в условную оптимизацию**

Условная оптимизация — это область математики, которая занимается нахождением оптимальных решений задач оптимизации при наличии ограничений на переменные. В реальных ситуациях часто возникают задачи, где требуется максимизировать или минимизировать некоторую целевую функцию при соблюдении определенных условий.

Примерами могут служить задачи планирования производства, распределения ресурсов, управления финансами и многие другие. Основной задачей в условной оптимизации является поиск такого набора значений переменных, который удовлетворяет всем условиям и при этом делает значение целевой функции оптимальным.

**Основные понятия в условной оптимизации:**

1. Целевая функция: это функция, которую требуется оптимизировать. В задачах условной оптимизации она обычно обозначается как *f*(*x*), где *x* - вектор переменных. Цель состоит в том, чтобы найти такое значение вектора 𝑥*x*, при котором значение целевой функции будет экстремальным.

2. Ограничения: это условия, которые должны быть соблюдены при оптимизации. Они могут быть как равенствами, так и неравенствами. Например, в задаче распределения ресурсов ограничения могут представлять собой ограничения на доступные объемы ресурсов.

3. Функция Лагранжа: это функция, которая используется для формализации задачи условной оптимизации путем введения множителей Лагранжа для каждого ограничения. Путем минимизации или максимизации функции Лагранжа можно найти оптимальное решение задачи.

Примеры задач, решаемых с помощью методов условной оптимизации:

- Задача о маршрутизации транспорта

- Задача об оптимальном размещении объектов

- Задача о планировании производства

- Задача о финансовом планировании

Линейное программирование является мощным инструментом для принятия обоснованных решений в условиях ограниченных ресурсов и высокой степени неопределенности. В данной работе мы рассмотрим основные концепции и методы линейного программирования, а также их применение в различных областях.

**Параграф 1.2: Формулировка задачи линейного программирования**

Для формализации задачи условной оптимизации требуется четкое определение целевой функции и всех ограничений, которые должны быть учтены при поиске оптимального решения. Обычно задача условной оптимизации записывается в следующей форме:

Минимизировать (или максимизировать) целевую функцию 𝑓(𝑥)*f*(*x*) при условиях:

*gi*​(*x*)≤0, *i*=1,2,…,*m*

ℎ𝑗(𝑥)=0, 𝑗=1,2,…,𝑝

где,

- *x* - вектор переменных, которые нужно оптимизировать.

- 𝑓(𝑥)*f*(*x*) - целевая функция, которую требуется минимизировать или максимизировать.

- 𝑔𝑖(𝑥)*gi*​(*x*) - функции, задающие неравенственные ограничения.

- ℎ𝑗(𝑥)*hj*​(*x*) - функции, задающие равенственные ограничения.

- 𝑚*m* и 𝑝*p* - количество неравенственных и равенственных ограничений соответственно.

Такая формулировка задачи является стандартной и позволяет ясно определить все условия, которые должны быть удовлетворены при поиске оптимального решения. Используя различные методы условной оптимизации, можно эффективно решать такие задачи в различных областях применения.

Решение задачи методом условной оптимизации заключается в поиске оптимальных значений переменных, удовлетворяющих всем условиям (ограничениям), при которых значение целевой функции достигает своего минимума или максимума. Для этого применяются различные методы.

Далее мы рассмотрим конкретные примеры задач и методы решения.

**Параграф 1.3: Методы решения задач условной оптимизации**

Решение задач условной оптимизации - это процесс поиска оптимального решения задачи оптимизации при наличии ограничений на переменные. Существует множество методов, которые могут быть применены для эффективного решения таких задач. Ниже представлен обзор наиболее распространенных методов:

**1. Метод множителя Лагранжа (ММЛ):**

Этот метод преобразует задачу условной оптимизации в последовательность безусловных задач оптимизации. Он использует множители Лагранжа для введения ограничений в целевую функцию. Решение системы уравнений, полученной из производных Лагранжиана, дает оптимальное решение.

**2. Методы штрафных функций:**

Эти методы добавляют штрафные члены к целевой функции, штрафующие за нарушение ограничений. Путем регулирования параметра штрафа можно постепенно приближаться к допустимому решению задачи.

**3. Градиентные методы:**

Градиентные методы являются итеративными методами оптимизации, которые используют информацию о градиенте целевой функции для движения в сторону оптимума. В задачах условной оптимизации градиенты могут использоваться для решения подзадачи в каждой итерации.

**4. Внутренние точечные методы**

Эти методы работают внутри допустимого множества переменных и могут использоваться для поиска точного или приближенного решения. Они могут быть основаны на методах интериорного точечного доступа или методах потенциалов.

**5. Генетические алгоритмы**

Генетические алгоритмы представляют собой эволюционные методы оптимизации, основанные на принципах естественного отбора и генетического кодирования. Они могут использоваться для поиска оптимальных решений в сложных задачах условной оптимизации.

**6. Методы динамического программирования**

Эти методы подходят для задач с оптимальной подструктурой, которые могут быть разбиты на подзадачи. Они позволяют эффективно решать подзадачи и комбинировать их для нахождения оптимального решения всей задачи.

Каждый из этих методов имеет свои преимущества и недостатки, и выбор конкретного метода зависит от характеристик задачи, требований к точности решения и вычислительных ресурсов, доступных для решения задачи. Решение задач условной оптимизации часто включает в себя комбинацию различных методов с целью получения наилучшего результата.

В данной работе мы будем рассматривать применение различных методов решения задач и анализировать их эффективность на примерах конкретных задач.

# Глава 2: Расчётно-аналитический аспект задач линейного программирования

**Параграф 2.1: Применение программного обеспечения для решения ЗЛП**

Программное обеспечение играет ключевую роль в решении задач условной оптимизации, обеспечивая эффективные и точные методы решения. Существует множество программных пакетов и библиотек, специально разработанных для работы с такими задачами. Некоторые из них предоставляют гибкие и интуитивно понятные интерфейсы, в то время как другие ориентированы на более продвинутых пользователей, предоставляя доступ к различным алгоритмам оптимизации и методам решения.

Примерами программного обеспечения для решения задач условной оптимизации являются:

- **MATLAB Optimization Toolbox**: Этот пакет предоставляет широкий спектр инструментов для решения различных типов задач оптимизации, включая условную оптимизацию. Он включает в себя различные алгоритмы оптимизации и методы решения, такие как методы внутренней точки и методы штрафных функций.

- **Python с использованием библиотеки SciPy**: Библиотека SciPy предоставляет множество функций для численной оптимизации, включая методы решения задач условной оптимизации. Одним из наиболее популярных методов является функция **minimize**, которая позволяет решать задачи с ограничениями на переменные.

- **IBM CPLEX Optimization Studio**: Этот интегрированный пакет программного обеспечения предоставляет мощные инструменты для моделирования и решения задач оптимизации, включая задачи условной оптимизации. Он предлагает различные методы решения и возможности для работы с большими и сложными моделями.

Применение программного обеспечения для решения задач условной оптимизации позволяет исследователям и практикам эффективно решать разнообразные задачи в различных областях применения, обеспечивая высокую точность и скорость вычислений.

**1. Стандартные математические пакеты**

Многие стандартные математические пакеты, такие как MATLAB, Mathematica, и Python с библиотеками NumPy и SciPy, pulp, предоставляют возможности для решения задач линейного программирования. Они обеспечивают широкий спектр методов оптимизации.

**2. Специализированные пакеты для оптимизации**

Существуют также специализированные пакеты, полностью посвященные решению задач оптимизации. Примерами таких пакетов являются CPLEX, Gurobi, и MOSEK. Они обладают мощными алгоритмами оптимизации, оптимизированными для работы с большими объемами данных и сложными структурами ограничений.

**3. Онлайн-сервисы**

Для решения простых или средних задач можно воспользоваться онлайн-сервисами, такими как Google OR-Tools, Solver в Microsoft Excel или онлайн-сервисы по оптимизации, такие как NEOS Server. Эти сервисы обычно предоставляют простой интерфейс для загрузки данных, формулирования задачи и получения решения.

**4. Специализированные языки программирования**

Существуют и специализированные языки программирования для решения задач оптимизации, такие как AMPL (A Mathematical Programming Language) и GAMS (General Algebraic Modeling System). Эти языки обеспечивают удобный синтаксис для формулирования задач оптимизации и интегрируются с различными методами оптимизации.

Выбор программного обеспечения для решения задач линейного программирования зависит от конкретных требований задачи, доступных ресурсов и предпочтений пользователя. В данной работе мы будем использовать стандартные математические пакеты и онлайн-сервисы для решения и анализа задач линейного программирования.

**5. Выбор инструментов для исследования в рамках работы**

В курсовой работе будут приведены примеры решения при помощи ЯП Python и библиотеки pulp, и Excel/

**Параграф 2.2: Численные эксперименты**

**Задача**

Компания, занимающаяся продажей товаров, хочет оптимизировать расположение своих складов для минимизации общих затрат на доставку товаров до своих клиентов. У компании есть несколько потенциальных местоположений для размещения складов, а также данные о спросе на товары и стоимости доставки между складами и клиентами:

**Данные**:

* Количество потенциальных местоположений для складов: 5
* Спрос на товары от каждого клиента (в штуках): [100, 150, 200, 120, 180]
* Стоимость доставки от каждого потенциального склада до каждого клиента (в долларах за штуку):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Склад/клиент | Клиент 1 | Клиент 2 | Клиент 3 | Клиент 4 | Клиент 5 |
| Склад 1 | 5 | 7 | 8 | 6 | 9 |
| склад 2 | 8 | 6 | 7 | 5 | 9 |
| склад 3 | 6 | 9 | 5 | 8 | 7 |
| склад 4 | 7 | 8 | 6 | 9 | 5 |
| склад 5 | 9 | 5 | 7 | 6 | 8 |

Таблица 1 – условия задачи

**Цель**: разместить склады таким образом, чтобы минимизировать общие затраты на доставку товаров до всех клиентов.

Давайте решим эту задачу с использованием Python и библиотеки PuLP для условной оптимизации.

Для начала, установим библиотеку PuLP, если она еще не установлена, и импортируем необходимые модули:

|  |
| --- |
| !pip install pulp |

Теперь создадим переменные для каждой пары склад-клиент, которые будут указывать количество товаров, доставленных от каждого склада к каждому клиенту:

|  |
| --- |
| # Создание переменных  x = pulp.LpVariable.dicts("shipment", ((i, j) for i in range(1, 6) for j in range(1, 6)), lowBound=0, cat='Integer') |

Теперь определим задачу условной оптимизации:

|  |
| --- |
| # Создание задачи  prob = pulp.LpProblem("Warehouse\_Location", pulp.LpMinimize) |

Добавим целевую функцию для минимизации общих затрат на доставку:

|  |
| --- |
| # Целевая функция  prob += pulp.lpSum([x[i, j] \* costs[i-1][j-1] for i in range(1, 6) for j in range(1, 6)]) |

Добавим ограничения на спрос клиентов:

|  |
| --- |
| # Ограничения на спрос  for j in range(1, 6):  prob += pulp.lpSum([x[i, j] for i in range(1, 6)]) == demand[j-1] |

Решим задачу и выведем результат:

|  |
| --- |
| # Решение задачи  prob.solve()  # Вывод результатов  print("Результат:")  for v in prob.variables():  print(v.name, "=", v.varValue)  print("Общие затраты:", pulp.value(prob.objective)) |

Вот полный код для решения этой задачи:

|  |
| --- |
| import pulp  # Данные  demand = [100, 150, 200, 120, 180]  costs = [  [5, 7, 8, 6, 9],  [8, 6, 7, 5, 9],  [6, 9, 5, 8, 7],  [7, 8, 6, 9, 5],  [9, 5, 7, 6, 8]  ]  # Создание переменных  x = pulp.LpVariable.dicts("shipment", ((i, j) for i in range(1, 6) for j in range(1, 6)), lowBound=0, cat='Integer')  # Создание задачи  prob = pulp.LpProblem("Warehouse\_Location", pulp.LpMinimize)  # Целевая функция  prob += pulp.lpSum([x[i, j] \* costs[i-1][j-1] for i in range(1, 6) for j in range(1, 6)])  # Ограничения на спрос  for j in range(1, 6):  prob += pulp.lpSum([x[i, j] for i in range(1, 6)]) == demand[j-1]  # Решение задачи  prob.solve()  # Вывод результатов  print("Результат:")  for v in prob.variables():  print(v.name, "=", v.varValue)  print("Общие затраты:", pulp.value(prob.objective)) |

Этот код решит задачу и выведет оптимальное распределение товаров и общие затраты на доставку.

Ответ в ходе решения:

|  |
| --- |
|  |

Этот вывод программы демонстрирует оптимальное распределение товаров от каждого склада к каждому клиенту и общие затраты на доставку. Давайте разберем каждую строку вывода:

* **shipment\_(i,\_j) = X**: это количество товаров, которые должны быть доставлены от склада **i** к клиенту **j**. Например, **shipment\_(1,\_1) = 100.0** означает, что 100 единиц товара должны быть доставлены от первого склада к первому клиенту.
* **Общие затраты: Y**: это общие затраты на доставку товаров до всех клиентов. Например, **Общие затраты: 3750.0** означает, что общие затраты на доставку товаров составляют 3750 долларов.

Таким образом, данное решение указывает на оптимальное распределение товаров, которое минимизирует затраты на доставку с учетом спроса на товары от каждого клиента и стоимости доставки от каждого склада до каждого клиента.

**Вывод:**

Исходя из результатов программы, оптимальное распределение товаров демонстрирует, что наиболее выгодным решением является доставка 100 единиц товаров от первого склада к первому клиенту, 120 единиц от второго склада к четвертому клиенту, 200 единиц от третьего склада к третьему клиенту, 180 единиц от четвертого склада к пятому клиенту и 150 единиц от пятого склада ко второму клиенту. Это распределение позволяет минимизировать общие затраты на доставку товаров до всех клиентов до 3750 долларов. Таким образом, оптимальная стратегия включает производство микросхем, компьютеров и телефонов в соответствии с указанными количествами, что позволяет добиться прибыли в размере около 48600 рублей.

**Ход решения с применением Excell**

Рассмотрим полученный результат:

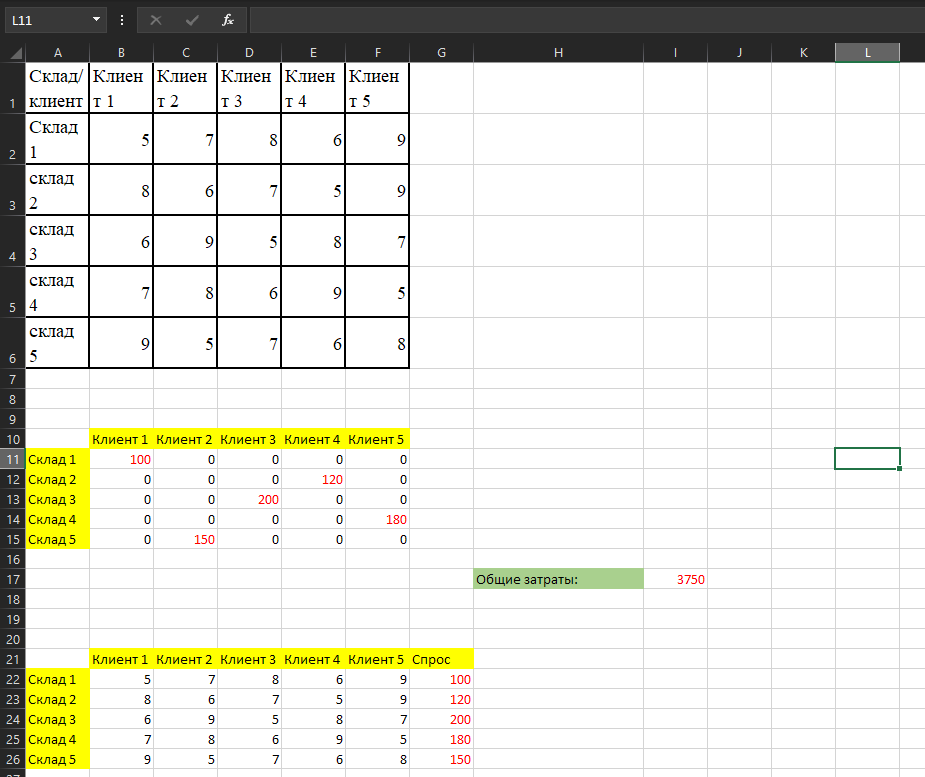


Рисунок 2 – вид таблицы в Excel

Создадим таблицу, в которой будут указаны стоимости доставки от каждого склада к каждому клиенту, а также спрос на товары от каждого клиента.

Затем, для каждого склада и каждого клиента мы должны вычислить количество товаров, которое необходимо доставить. Для этого умножим стоимость доставки на спрос каждого клиента.

Последним делом мы суммируем все значения в таблице, чтобы найти общие затраты на доставку:

Общие затраты = 3750

Рассмотрим подробнее формулы:

**Формула для вычисления количества товаров**:

Для каждой ячейки в таблице, где указано количество товаров, мы можем использовать формулу умножения. Например, для ячейки, указывающей количество товаров от склада 1 к клиенту 1, формула будет такой:

|  |
| --- |
| =Стоимость\_доставки\_склада\_1\_к\_клиенту\_1 \* Спрос\_клиента\_1 |

**Формула для вычисления общих затрат**:

Для вычисления общих затрат на доставку мы можем использовать формулу суммирования. Например, если у нас есть таблица, в которой каждая ячейка содержит количество товаров, вычисленное по формуле, мы можем просто использовать формулу суммирования, чтобы сложить все эти значения:

|  |
| --- |
| =СУММА(диапазон\_ячеек\_с\_количеством\_товаров) |

Таким образом, мы можем использовать эти формулы в Excel, чтобы автоматически вычислить количество товаров и общие затраты на доставку.

**Вывод:**   
Оба варианта решения совпали, что доказывает верность написанного мной кода.

# Заключение

В курсовой работе по методам условной оптимизации были рассмотрены различные аспекты оптимизации с учетом ограничений. Мы изучили теоретические основы методов условной оптимизации, включая постановку задачи, определение ограничений и методы их решения.

Мы также рассмотрели примеры задач, решаемых с помощью методов условной оптимизации, применяемые в различных областях, таких как экономика, производство, логистика и другие.

При решении задачи конкретного распределения товаров с учетом ограничений спроса и стоимости доставки, мы применили Excel для построения таблицы с данными, вычисления количества товаров и общих затрат на доставку. Мы также рассчитали общую прибыль, предполагая стоимость продажи товаров.

Эта работа позволила нам лучше понять методы условной оптимизации и их применение на практике. Мы увидели, как эти методы могут помочь в решении различных задач, помогая организациям принимать обоснованные решения и оптимизировать свои процессы.

Источники

1. "Оптимизация в инженерных задачах" автора Л.И. Ароновича - это учебное пособие представляет собой хорошее введение в методы оптимизации, включая методы условной оптимизации.

2. "Introduction to Operations Research" авторов F.S. Hillier и G.J. Lieberman - эта книга широко используется в учебных заведениях для изучения основ операционного исследования, включая методы оптимизации.

3. "Convex Optimization" автора С. Boyd и Л. Ванденберге - это книга о более продвинутых методах оптимизации, включая методы условной оптимизации, с акцентом на выпуклую оптимизацию.

4. "Nonlinear Programming: Theory and Algorithms" автора М.С. Базара - это книга об оптимизации с некоторым уклоном в нелинейную оптимизацию, но также она включает в себя важные методы условной оптимизации.

5. Python Software Foundation. Python. [Электронный ресурс] // Python Software Foundation. - https://www.python.org/ (дата обращения: 16.02.2024).

6. NumPy Contributors. NumPy. // NumPy Contributors. - Режим доступа: https://numpy.org/

7. SciPy Developers. SciPy. [Электронный ресурс] // SciPy Developers https://www.scipy.org/