**Федеральное государственное образовательное**

**бюджетное учреждение**

**высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ**

**ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ**

**ФЕДЕРАЦИИ»**

**(Финансовый университет)**

**Факультет**

**информационных технологий и анализа больших данных**

**Направление «Прикладная математика и информатика»**

**Домашнее задание № 5**

«**Метод внутренней точки**»

Студенты группы ПМ19-3:

Захаров Д. В.

Исмоилова М. В.

Константинов К. Л.

Мосолова К. Д.

Самофалова Т. А.

Руководитель:

Аксенов Дмитрий Андреевич

**Москва 2022**

Оглавление

[**1.** **Постановка задачи (физическая модель)** 3](#_Toc101913818)

[**2.** **Математическая модель и алгоритмы** 3](#_Toc101913819)

[**Метод первой фазы для нахождения внутренней точки** 4](#_Toc101913820)

[**Метод Ньютона (способ решения двойственной задачи)** 4](#_Toc101913821)

[**Метод логарифмических барьеров.** 5](#_Toc101913822)

[**Прямо-двойственный метод внутренней точки** 7](#_Toc101913823)

[**3. Варианты использования системы** 9](#_Toc101913824)

[**4. Архитектура решения.** 10](#_Toc101913825)

[**5. Тестирование** 14](#_Toc101913826)

[**6. Выводы и заключение** 15](#_Toc101913827)

# **Постановка задачи (физическая модель)**

HR-агентству «Рога и копыта» необходимо определять оптимальную зарплату людей, которых оно нанимает для других компаний. Сотрудники агентства провели различные исследования и определили основные факторы, влияющие на зарплаты. Для каждой отдельной компании им удалось составить функциональные зависимости зарплат от данных факторов. Также, сотрудникам удалось выразить ограничения, накладываемые на зарплату через определенные ими факторы.

Для решения задачи поиска минимума функций было решено выбрать следующие математические алгоритмы:

• Решение задачи оптимизации для функции с ограничениями типа равенства методом Ньютона (способ решения двойственной задачи);

• Решение задачи оптимизации для функции с ограничениями типа НЕравенства методом логарифмических барьеров (прямой метод внутренней точки);

• Решение задачи оптимизации для функции с ограничениями типа НЕравенства прямо-двойственным методом внутренней точки;

Под каждый обозначенный выше алгоритм была написана отдельная функция. (их описание см. в разделе 5). Также была сделана визуализация прямого метода внутренней точки, все алгоритмы были протестированы на переданных заказчиком функциях и ограничениях, была замерена скорость работы каждого алгоритма и проведено их сравнение между собой. В следующих разделах отчета каждое действие рассмотрено более подробно. Также предложены варианты развития данного проекта.

# **Математическая модель и алгоритмы**

Рассмотрим задачу выпуклой оптимизации:

*(1)*

Заметим, что в здесь, как и в прочих выпуклых задачах условной оптимизации ограничения вида равенства могут быть только линейными, иначе допустимое множество не будет выпуклым. Ограничения типа неравенств и целевая функция могут быть линейными или нелинейными, но обязательно выпуклыми.

Важной частью решения задачи оптимизации с ограничениями типа неравенства является выбор допустимой внутренней точки. Если она не была задана изначально или не подходит под ограничение, для поиска внутренней точки применяется метод первой фазы.

## **Метод первой фазы для нахождения внутренней точки**

Пусть мы имеем ограничения типа неравенства вида: . Чтобы точка была допустимой для исходных ограничений, достаточно, чтобы она была допустимой для .

Алгоритм:

1. Выбирается любая точка .

2. Рассчитывается

3. Если < 0, то точка является внутренней, и ее можно использовать для основного метода оптимизации.

Если > 0 или = 0, продолжается перебор точек. Если подходящая точка так и не была найдена, значит, либо множества, заданные ограничениями, не пересекаются ( > 0), либо внутренняя точка находится на границе ( = 0), и тогда метод барьеров схлопнется.

## **Метод Ньютона (способ решения двойственной задачи)**

Рассмотрим случай, когда заданы только ограничения типа равенства.

Функция Лагранжа будет иметь вид:

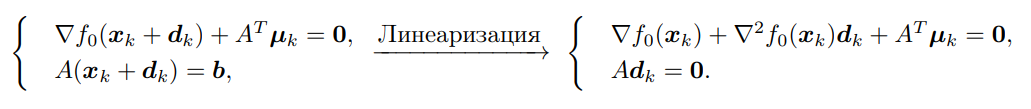
Если точка является глобальным решением задачи (1), то найдутся такие , что ( удовлетворяют системе Куна-Таккера:

*(2)*

Пусть имеется некоторая точка , лежащая на прямой Ax=b. Тогда можно найти очередное направление оптимизации , решив СЛАУ:

*(3)*

Так как строго выпукло, то . Следовательно, матрица СЛАУ (3) является невырожденной, и пара () определена однозначно.

Можно заметить, что система (3) эквивалента линеаризации системы Куна-Таккера (2):

*(4)*

решается система (4), откуда находится пара ()

3. Рассчитывается новая точка и проверяется выполнение условия (5). Если оно не выполняется, повторяется пункт 2 данного алгоритма.

Поскольку в библиотеке scipy.optimize была найдена функция, которая уже реализует выполнение шагов 2-3 при правильной передаче в нее параметров, было решено использовать ее, внеся доработку в преобразование введенных пользователем целевой функции и ограничений к нужному виду.

## **Метод логарифмических барьеров.**

Рассмотрим задачу оптимизации (1) с ограничениями типа неравенств:

Идея метода заключается в сведении задачи на условный минимум к решению последовательности задач поиска минимума вспомогательной функции:

*(6)*

Очевидно, что раз , то , а значит, можно без труда найти значение логарифма. Новая точка находится методом Ньютона:

*(7)*

Если условие (5) выполнилось, алгоритм прекращает работу. В противном случае *(8)*

Данный метод содержит две проблемы: начальная точка должна быть внутренней, а новые посчитанные точки не должны оказаться вне определённого ограничениями множества. Для решения первой проблемы используют метод первой фазы, описанный выше, а вторая подразумевает проверку на принадлежность на каждой итерации алгоритма.

Алгоритм:

1. Пользователь вводит:

1) функцию в явном виде

2) ограничения типа неравенств в явном виде ( )

3) координаты начальной точки, удовлетворяющей ограничениям типа неравенств (если этот параметр опущен, точка находится методом первой фазы)

4) точность e (по умолчанию 10^(-6))

2. Задается

3. По формуле (6) рассчитывается функция

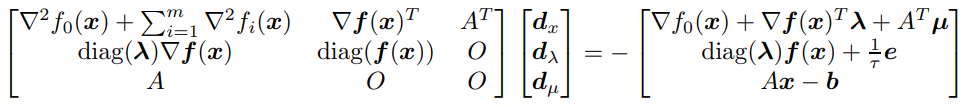
4. Рассчитывается новая точка по формуле (7).

5.Проверяется, принадлежит ли эта точка допустимому множеству. Если нет, то наилучшая точка была получена на предыдущем шаге, и алгоритм заканчивает свою работу.

6. В противном случае проверяется выполнение условия (5). Если условие выполняется, алгоритм прекращает свою работу. Иначе находится новое по формуле (8) и повторяются шаги 3-6.

## **Прямо-двойственный метод внутренней точки**

Прямо-двойственный метод Ньютона оптимизирует одновременно прямые переменные x и двойственные переменные µ путем решения линеаризованной системы Куна-Таккера (2).

Направления оптимизации находятся путем решения следующей СЛАУ:

*(9)*

Здесь  **–** это вектор-столбец ограничений типа неравенство; – это матрица, на диагонали которой расположен вектор (размера m\*1); e – единичный вектор; А – матрица коэффициентов в ограничении типа равенств, где на позиции (i,j ) расположен коэффициент в j ограничении при i-ой переменной.

Если обозначить левую матрицу как M1, а правую как M2, то = *(10)*

Вычислив направления оптимизации, можно вычислить и длину шага:

*(11)*

находим с помощью метода поиска backtracking: выбираем как элемент последовательности {1, ½, ¼ ,…}, пока не найдем то, при котором будет выполняться ).

После чего выполняется обновление значений:

*(12)*

*(13)*

*(14)*

Алгоритм.

1. Пользователь вводит:

1) функцию в явном виде

2) ограничения типа равенств (Ax=b)

3) ограничения типа неравенств в явном виде (g<=0)

4) координаты начальной точки, удовлетворяющей заданным ограничениям

5) точность e (по умолчанию 10^(-6))

2. Задаются:

3. Находятся направления оптимизации по формуле (10).

4. Вычисляется длина шага по формуле (11) описанному выше методу backtracking.

5. Вычисляются новые значения по формулам (12-14).

6. Проверяется выполнение условие (5). Если оно выполняется, алгоритм прекращает свою работу, в противном случае повторяются шаги 3-6.

# **3. Варианты использования системы**

ВИ1: Пользователь хочет найти решение задачи оптимизации для функции с ограничениями типа равенства методом Ньютона (способом решения двойственной задачи)

1. Пользователь импортирует функцию eq\_dual\_newton()
2. Пользователь передает в написанную функцию математическую функцию в формате строки (все знаки аналогичны стилю питона (\*\* - степень, \* - умножение и т.д), список ограничений (справа от равенства должен быть только 0) и начальную точку (параметр x0) в виде кортежа

2.1 Пользователь может задать точность предоставленного ему решения (параметр tol) По умолчанию функция возвращает пять знаков после запятой.

1. Функция возвращает пользователю кортеж из двух элементов: numpy.array – точка экстремума и значение функции в данной точке.

ВИ2: Пользователь хочет найти решение задачи оптимизации для функции с ограничениями типа НЕравенства методом логарифмических барьеров (прямым методом внутренней точки)

1. Пользователь импортирует функцию log\_barriers()
2. Пользователь передает в написанную функцию математическую функцию в формате строки (все знаки аналогичны стилю питона (\*\* - степень, \* - умножение и т.д), список ограничений в виде ‘выражение [знак неравенства] 0’
   1. В случае, если пользователь хочет передать начальную точку, он ее передает в виде кортежа.
   2. В случае, если пользователь не знает начальной точки, точка рассчитывается методом первой фазы. Параметр start\_point по умолчанию пустой кортеж, передавать туда ничего не надо.
   3. Пользователь может задать точность предоставленного ему решения (параметр accuracy) По умолчанию функция возвращает шесть знаков после запятой.
3. Функция возвращает пользователю кортеж из двух элементов: numpy.array – точка экстремума и значение функции в данной точке.

ВИ3: пользователь хочет найти решение задачи оптимизации для функции с ограничениями типа НЕравенства прямо-двойственным методом внутренней точки

1. Пользователь импортирует функцию terrifying()
2. Пользователь передает в написанную функцию математическую функцию в формате строки (все знаки аналогичны стилю питона (\*\* - степень, \* - умножение и т.д), координаты начальной точки в виде списка (параметр x0), список ограничений в виде ‘выражение [знак неравенства (только <= <)] 0’, список коэффициентов при переменных для ограничений типа равенств (параметр A), список значений свободных членов для коэффициентов типа равенств (параметр b)
   1. Пользователь может задать точность предоставленного ему решения (параметр tol) По умолчанию функция возвращает шесть знаков после запятой.
3. Функция возвращает пользователю кортеж из двух элементов: numpy.array – точка экстремума и значение функции в данной точке.

# **4. Архитектура решения.**

1. eq\_dual\_newton.
2. log\_barriers.
3. terrifying.

А также функции, которые используются основными функциями:

1. get\_pain – представляет матрицы в правильном формате для решения СЛАУ

2. get\_first\_line – вычисляет первую строку матрицы СЛАУ

3. get\_second\_line – вычисляет вторую строку матрицы СЛАУ

4. get\_right\_pain – вычисляет правую матрицу в СЛАУ

5. search\_point – находит оптимальную точку

6. first\_phase – получение случайной начальной точки

Основные функции:

1. eq\_dual\_newton – решение задачи оптимизации для функции с ограничениями типа равенства методом Ньютона

*Аргументы функции:*

*Обязательные:*

- func: string

Для записи используется синтаксис питона

- equality: list

Ограничения типа равенство

- x0 : tuple

Стартовая точка.

Необязательные:

- tol: int

Количество знаков после запятой для выходных значений

*Выход:*

Кортеж из элементов:

Оптимальная точка, найденная данным методом

Значение функции в найденной точке

1. log\_barriers - Решение задачи оптимизации для функции с ограничениями типа НЕравенства методом логарифмических барьеров (прямой метод внутренней точки).

*Аргументы функции:  
  
Обязательные:*- func: string

Для записи используется синтаксис питона

- restrictions: list

Ограничения типа НЕравенство

Необязательные:

- start\_point : tuple

Стартовая точка.

- accuracy: int

Точность

- max\_steps: int

Максимальное количество итераций

*Выход:*

Кортеж из элементов:

Оптимальная точка, найденная данным методом

Значение функции в найденной точке

3. terrifying – решение задачи оптимизации для функции с ограничениями типа неравенства прямо-двойственным методом внутренней точки.

*Аргументы функции:*

Обязательные:

- func - функция в аналитическом виде для оптимизации. Обязательно задается в приведенной форме

- x0 – стартовая точка

- us – список ограничений, заданных строками. Обящательно задается в форме, где справа от знака < <= = стоит ноль. Другие знаки не поддерживаются

- a – список коэффициентов перед переменными в ограничении типа равенство. Если ограничений типа равенство нет, то передается пустой список

- b - список свободных членов в ограничении типа равенство. Если ограничений типа равенство нет, то передается пустой список

Необязательные:

- tol: int

Критерий остановки, точность

*Выход:*

Кортеж из элементов:

Оптимальная точка, найденная данным методом

Значение функции в найденной точке

# **5. Тестирование**





Исходя из полученных результатов, можем сделать выводы, что наиболее точными являются алгоритмы Ньютона и логарифмических барьеров. Учитывая скорость выполнения алгоритма, предпочтительнее метод логарифмических барьеров. Несмотря на то, что метод Ньютона выдаёт довольно точные результаты, его минус в том, что ограничения задаются исключительно равенствами, но и скорость вычислений часто меньше.

# **6. Выводы и заключение**

Таким образом, по результатам тестирования, получили, что решение поставленной задачи методом Ньютона будет наиболее оптимальным с точки зрения технической части. Данный алгоритм использует библиотечную функцию, что в данном случае будет плюсом. С точки зрения бизнеса, чаще встречаются ограничения типа неравенств. В таком случае решать задачу предлагается методом логарифмических барьеров. По результатам тестирования, данный метод и прямо-двойственной метод внутренней точки показали себя примерно одинаково. При этом, метод логарифмических барьеров куда проще реализуется. Как следствие, это поможет избежать технических ошибок или свести их к минимуму. Также, данный метод может использоваться без указания начальной точки, что может облегчить работу пользователя.

В будущем планируется усовершенствовать функционал алгоритма, улучшить защиту от ошибок со стороны пользователя и исправлять возникающие технические неполадки.