ГУАП

КАФЕДРА № 44

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

доцент, к.т.н., доцент		А.А.Сенцов
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4

УСЛОВНАЯ ОПТИМИЗАЦИЯ

по курсу: МЕТОДЫ ОПТИМИЗАЦИИ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4242М		М.В.Климов
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2023

1. Цель работы и вариант

Ознакомиться с известными аналитическими и численными методами условной оптимизации, получить опыт решения условной оптимизационной задачи, приобрести навыки использования математических программных средств для решения задач на условный экстремум.

Вариант 7.

Задача Ж. Дан угол 45º и точка А внутри него. Как нужно провести прямую через точку А, чтобы отсечь от угла треугольник наименьшего периметра (рис. 1)? Ответ: вневписанная в треугольник окружность должна касаться MN в точке А.

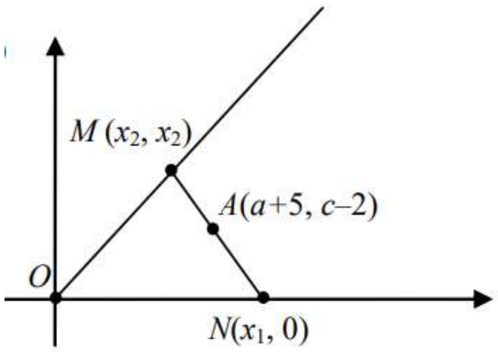


Рисунок 1. Поясняющий рисунок к задаче

Методом Штрафных функций.

Начальные данные: S = 7, e = 0.001.

2. Вычисления:

Целевая функция к данной задаче имеет вид:

$$f(x_1, x_2) = x_1 + (x_2 * \sqrt{2}) + \sqrt{{x_2}^2 + (x_1 - x_2)^2} \rightarrow min$$

Ограничивающая функция в данной задаче будет иметь вид:

$$q_1(x_1, x_2) = x_1 * x_2 \geq 0$$

$$g_1(x_1, x_2) = a + 5 + (c - 2) * \frac{x_1 - x_2}{x_2} - x_1 = 0$$

Метод штрафных функций так же, как и аналитические методы условной оптимизации, основан на переходе от задачи с ограничением к задаче без ограничений:

$$F(\mathbf{x}) = f(\mathbf{x}) + rP(\mathbf{x}) \to \min$$

Здесь r – некоторое число, называемое величиной штрафа.

P(x) – функция штрафа (penalty function), удовлетворяющая условиям:

$$P(\mathbf{x}) = \begin{cases} 0, \text{если } \mathbf{q}(\mathbf{x}) = 0, g_i(\mathbf{x}) \geq 0, i = \overline{1,s}; \\ > 0, \text{если } \mathbf{q}(\mathbf{x}) \neq 0 \text{ или } \exists g_i(\mathbf{x}) < 0, i = \overline{1,s}. \end{cases}$$

Функция штрафа равна нулю, если все ограничения выполняются и больше нуля, если нарушено хотя бы одно из ограничений.

Наиболее простые способы формирования такой функции:

$$P(\mathbf{x}) = \sum_{j=1}^{m} q_j^2(\mathbf{x}) - \sum_{i=1}^{s} \min(0, g_i(\mathbf{x})),$$

$$P(\mathbf{x}) = \sum_{j=1}^{m} |q_j(\mathbf{x})| - \sum_{i=1}^{s} \min(0, g_i(\mathbf{x})).$$

Алгоритм поиска экстремума методом штрафных функций можно сформулировать в виде трех шагов.

Шаг 1. Выбрать произвольное начальное значение r. Обычно выбирают r = 1.

Шаг 2. Найти решение x* задачи каким-либо численным методом.

Шаг 3. Если P(x *) > ε, увеличить r и перейти к шагу 2. Иначе: x* – искомая точка минимума.

В качестве начальной точки при численном решении задачи безусловной оптимизации на шаге 2 рекомендуется выбирать точку x*, полученную на предыдущей итерации метода штрафных функций.

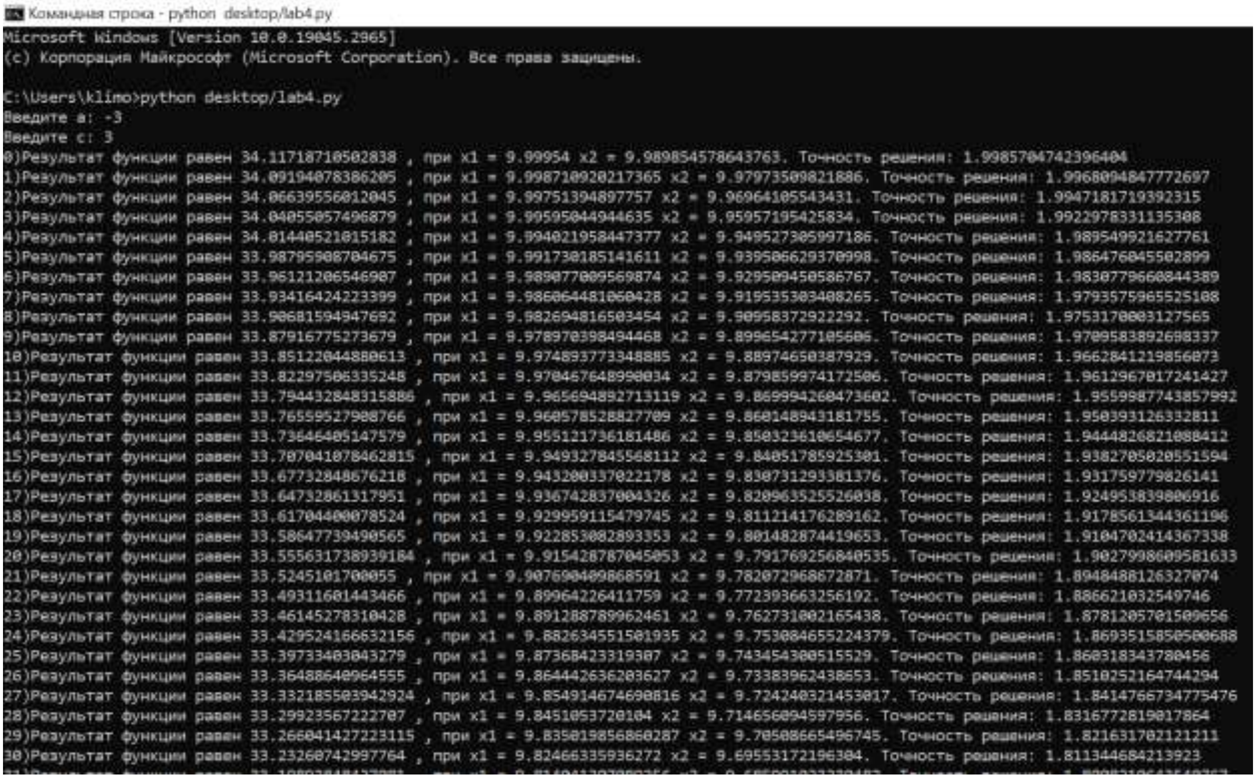
Программа по итогу должна проводить два алгоритма, один из которых это нахождение локального минимума при заданной начальной точке, а другой

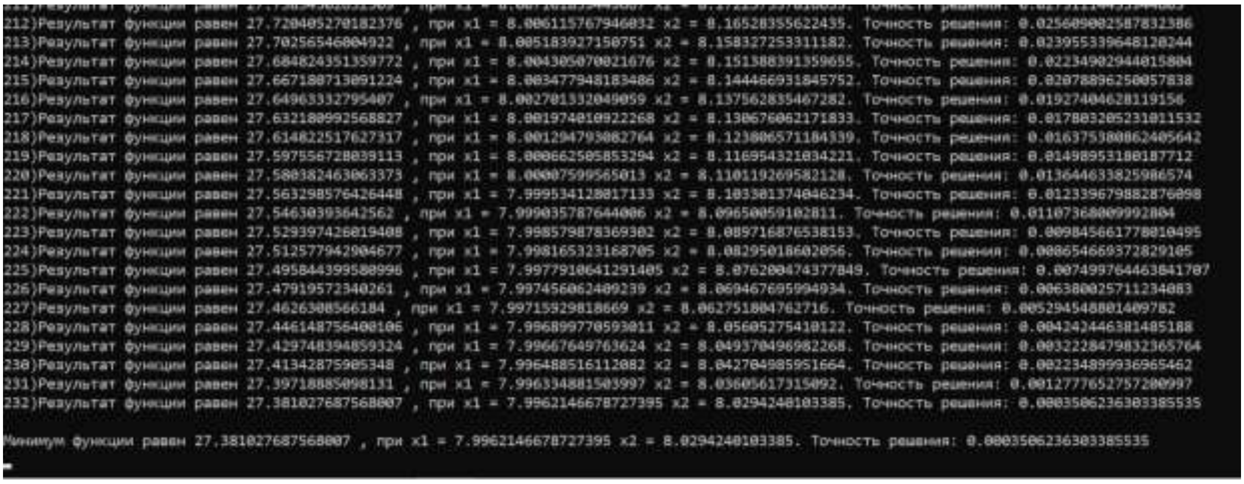
– изменение начальной функции и перезапуск первого алгоритма при недостаточно точном выполнении заданных условий.

Как под алгоритм для метода Штрафных функций был выбран метод Градиентного спуска.

Защита от ввода некорректных данных не проводится, поскольку при некорректных данных пользователь получит лишь очень странный результат, тем не менее подходящий под введённые условия.

Код программы приведён в приложении 1, а результат выполнения представлен на рисунках 1-3.





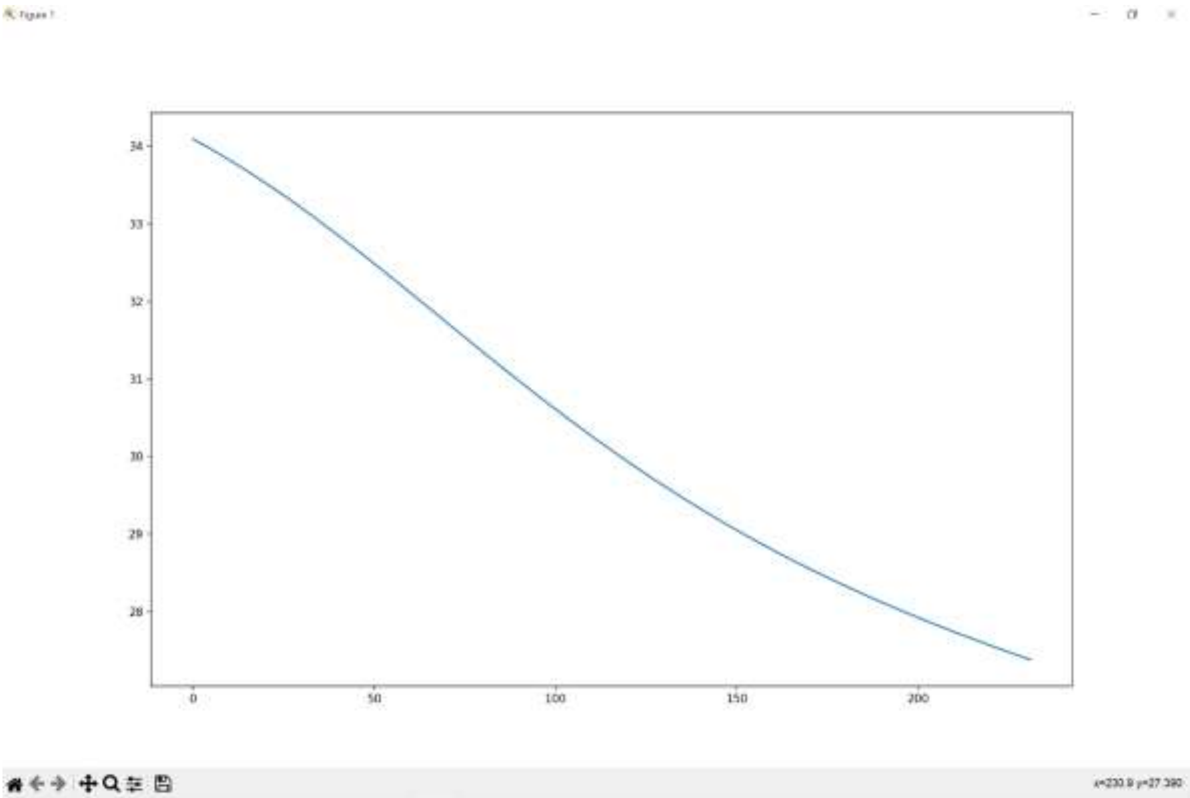


Рисунок 1-3. Результат выполнения программы в консоли.

3. Выводы.

В процессе выполнения лабораторной работы была решена поставленная задача по нахождению минимального периметра методом Штрафных функций. При тестировании возникали ошибки при нахождении исходной точки и начальной точки поиска в различных четвертях координатной плоскости. В остальном, решение было получено в пределах менее 0.001 от предполагаемой точки при начальных данных а=-3, с=3 за 232 шага.

Также был проведён тест для а=7, с=7 (Рисунок 4), в котором достигнуть нужного результата удалось только более чем за 70 000 итераций основного цикла.

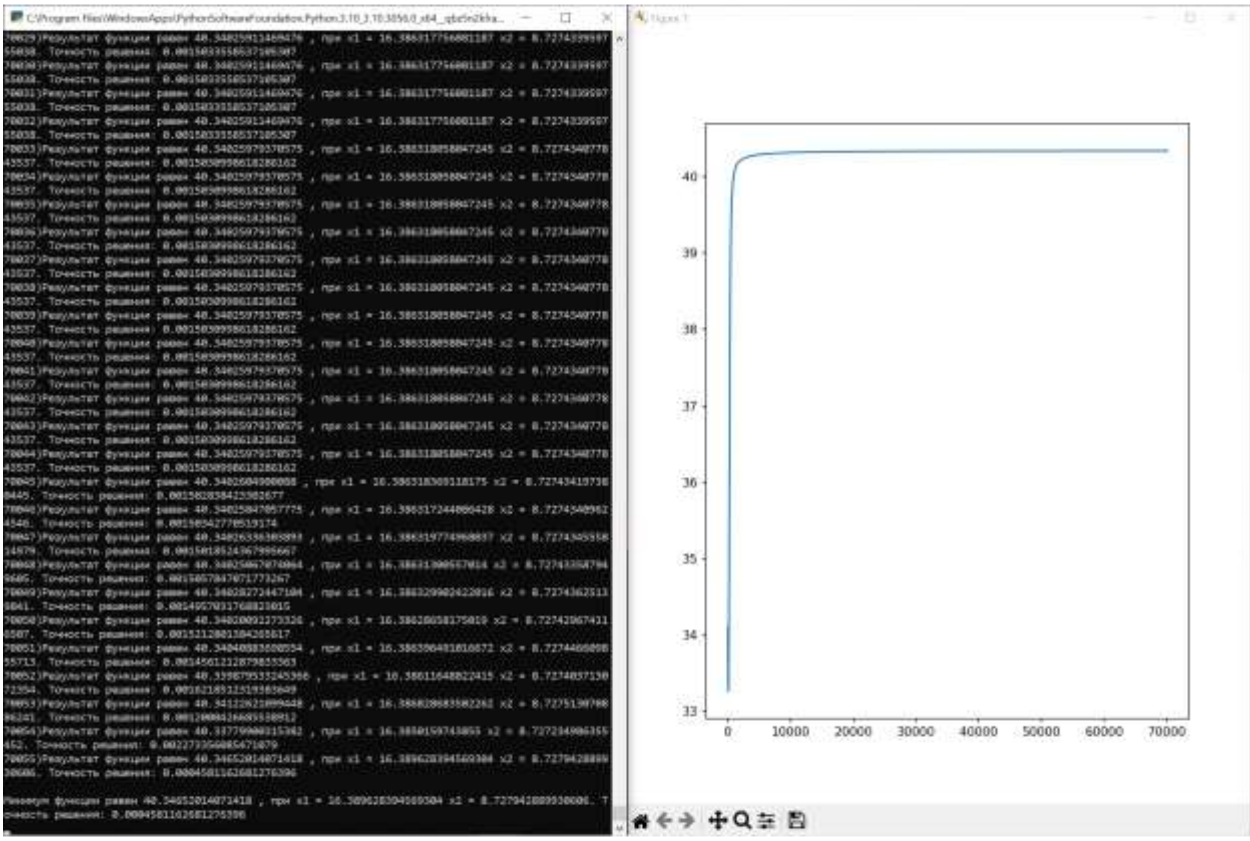


Рисунок 4 – Программное решение задачи при а=7, с=7

ПРИЛОЖЕНИЕ 1

Содержимое программы

```
import numpy as np
import matplotlib.pyplot as plt
import math;

def s(res, x):
    return (res[x])

def fun(x):
    return (x[0,0] + ((2 ** (1/2)) * x[1,0]) + (((x[1,0] ** 2) + ((x[0,0]
- x[1,0]) ** 2)  ) ** (1/2)))

def g1(x):
    a = x[0,0]*x[1,0]
    if (a<0):
        return a
    else:
        return (0)

def g2(a, c, x):
    return ((a+5)+((c-2)*(x[0,0] - x[1,0])/x[1,0]) - x[0,0])

def Pfun(g1, g2,s,a,c,x):
    return(abs(g2(a, c, x)))

def gfun(s,a,c,r, x):
    result = []
    result.append(1 + ((x[0,0]-x[1,0])/(((x[1,0] ** 2) + ((x[0,0] -
x[1,0]) ** 2) ) ** (-1/2))) + r*( 2*((a+5)+((c-2)*(x[0,0]-
x[1,0])/x[1,0])- x[0,0])*(((c-2)/x[1,0]) - 1) - abs(g1(x)/x[0,0])))
    result.append((2 ** (1/2)) + ((x[1,0]+x[1,0]-x[0,0])/(((x[1,0] ** 2)
+ ((x[0,0] - x[1,0]) ** 2)  ) ** (-1/2))) + r*(2*((a+5)+((c-2)*(x[0,0]-
```

```
x[1,0])/x[1,0])-       x[0,0])*((c-2)/(-1*(x[1,0]       **2)))        -
abs(g1(x)/x[1,0])))
    return result

def grad(s,a,c,gfun, r,  x0, e):
    result = []
    la = 0.0001
    gradi = gfun(s,a,c,r,x0)
    a1 = abs(gradi[0])
    a2 = abs(gradi[1])
    while ((a1 > (e)) and (a2 > (e))):
        gradi = gfun(s,a,c,r,x0)
        x0[0,0] = x0[0,0] - (la * gradi[0])
        x0[1,0] = x0[1,0] - (la * gradi[1])
        if((a1 == abs(gradi[0])) and (a2 == abs(gradi[1]))):
            a1 = 0
            a2 = 0
            break
        else:
            a1 = abs(gradi[0])
            a2 = abs(gradi[1])

    return (x0)



a = abs(float(input('Введите a: ')))
c = abs(float(input('Введите c: ')))
e = 0.001
n = 0
S = 7
x0 =  np.mat([[10.0], [10.0]])
r = 1
x = grad(S,a,c,gfun, r,  x0, e)
toc = abs(g1(x)) + abs(g2(a, c, x))
```

```
print(f'{n})Результат функции равен {fun(x)} , при x1 = {x[0,0]} x2 = {x[1,0]}. Точность решения: {toc}')
resultfun = []

while (toc>e):
    r = r + 1
    n = n + 1
    x = grad(S,a,c,gfun, r,  x0, e)
    toc =  abs(g1(x)) + abs(g2(a, c, x))
    resultfun.append(fun(x))
    print(f'{n})Результат функции равен {fun(x)} , при x1 = {x[0,0]} x2 = {x[1,0]}. Точность решения: {toc}')

print(f'')
print(f'Минимум функции равен {fun(x)} , при x1 = {x[0,0]} x2 = {x[1,0]}. Точность решения: {toc}')

xx=np.arange(0,n,1)
plt.plot(xx, resultfun)
plt.show()
```

ПРИЛОЖЕНИЕ 2

Вопросы:

1. **Метод Лагранжа. Процедура сближения параметрических множителей.**

Метод, предложенный Лагранжем, может быть сформулирован в виде следующей теоремы.

Теорема о множителях Лагранжа.

Пусть x0 – решение задачи условной оптимизации

$f(x1, ..., xn) \to \min (\max)$ (1)

$q1(x1, ..., xn) = 0, \dots, qm(x1, ..., xn) = 0.$

Тогда, если все функции в задаче (1) дифференцируемы в точке x0, то их градиенты в этой точке линейно зависимы, то есть существуют такие не все равные нулю множители λ0, λ1, …, λm, для которых:

$\lambda 0\nabla f(x0) + \lambda 1\nabla q1(x0) + \dots + \lambda m\nabla qm(x0) = 0.$ (2)

Множители λ0, λ1, …, λm называют множителями Лагранжа.

Рассмотрим геометрическое обоснование теоремы о множителях Лагранжа на примере двумерной задачи с одним ограничением:

$f(x, y) \to \min, q(x, y) = 0.$ (3)

Функция f(x, y) представляет собой некоторую поверхность в трехмерном пространстве. Для изображения рельефа функции f(x, y) на плоскости xy будем использовать линии уровня – множества точек, с одинаковым значением функции. Таким образом, график задачи будет содержать семейство линий уровня целевой функции и кривую-ограничение (Рисунок 1). Для решения задачи требуется среди линий уровня целевой функции f(x, y) найти линию с наименьшим значением, имеющую хотя бы одну общую точку с кривой q(x, y) = 0. Из рис. 1 видно, что для любой линии уровня, пересекающей кривую ограничения, существует линия уровня с меньшим значением целевой функции, имеющая общую точку с кривой-ограничением.

Следовательно, искомая точка будет являться точкой касания линии уровня и кривой ограничения.

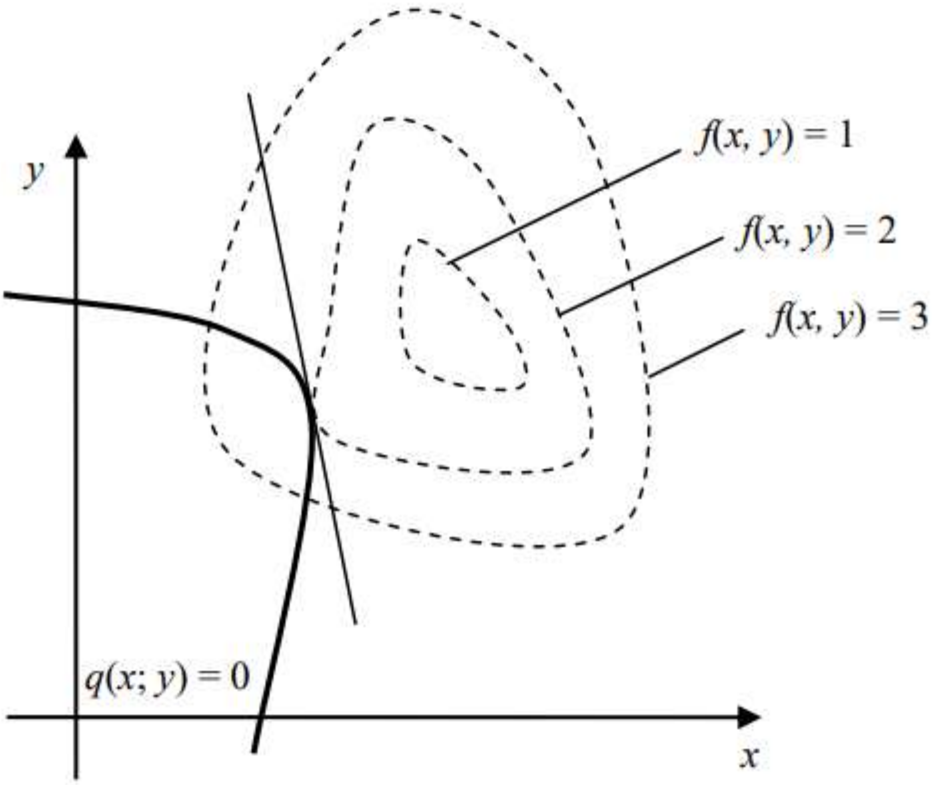


Рисунок 1. Геометрическое обоснование метода Лагранжа

Поиск точки касания опирается на одно из геометрических свойств градиента функции [4] – вектор градиента, построенный в любой точке функции, перпендикулярен касательной к функции в этой точке. Известно [4], что в точке касания двух кривых касательные к этим кривым совпадают, следовательно, векторы градиентов функций в точке касания коллинеарны. Таким образом, искомая точка минимума задачи (3.10) должна удовлетворять уравнению:

λ0∇f(x0, y0) = λ1∇q(x0, y0).

Уравнение (2) и условия задачи (1), согласно лемме Ферма , являются необходимым условием локального экстремума составной функции, называемой функцией Лагранжа:

$$L\left(x_1,\ldots,x_n,\lambda_0,\lambda_1,\ldots,\lambda_m\right)=$$

$$=\lambda_0 f\left(x_1,\ldots,x_n\right)+\sum_{i=1}^{m}\lambda_i q_i\left(x_1,\ldots,x_n\right)\rightarrow \min\left(\max\right).$$

Приравнивая, согласно лемме Ферма, частные производные функции Лагранжа нулю, получим уравнение (2) и условия задачи (1) – систему из n + m уравнений для n + m + 1 неизвестных: x1, ..., xn, λ0, λ1, …, λm. Несоответствие числа уравнений числу неизвестных связано с тем, что коэффициенты λ0, λ1, …, λm определены с точностью до общего множителя, так как обе части уравнения (2) могут быть умножены на любую ненулевую константу. Это означает, в частности, что если λ0 ≠ 0, можно разделить обе части (2) на λ0, тем самым получить λ0 = 1 и уравнять число уравнений и число неизвестных. Таким образом, идея Лагранжа позволяет свести задачу поиска экстремума целевой функции n-го порядка с m ограничениями типа равенство (1) к задаче поиска стационарных точек составной целевой функции порядка n + m без ограничений.

2. Аппроксимирующие методы оптимизации. Описание и примеры.

Использование аппроксимаций (приближений) целевой функции –

принципиально иной подход к решению задачи оптимизации по сравнению с

методами исключения интервалов.

Идея аппроксимации применима не только к одномерной, но и к многомерной

оптимизации и заключается в следующем итерационном алгоритме:

1. Задать целевую функцию f(x) и начальное приближение x0 .
2. Построить аппроксимирующую функцию φ(x), такую, что она примерно равна f(x) в окрестности x0.
3. Найти экстремум x1 функции φ(x) аналитическим методом.

4. Если желаемая точность не достигнута, то вернуться на шаг 2, используя x1 в качестве нового начального приближения. Иначе – завершить алгоритм.

В основе этого подхода лежит идея о том, что близкие функции имеют сходные свойства и их экстремумы расположены рядом. В рамках аппроксимационного подхода могут быть описаны многие численные методы, его применимость не ограничена методами оптимизации. Эффективность аппроксимационного метода определяется тем, насколько близки свойства целевой функции и её приближения.

Программное решение задачи, полученное в результате выполнения данной работы является примером использования аппроксимирующего метода.

3. Можно ли применить другой алгоритм, который даст решение менее, чем за за 232 шага?

В данном случае метод барьерных функций плохо подходит, поскольку есть ограничения по типу $g_n(x) = 0$. Однако в данном методе за каждый шаг увеличивается множитель r:

```
while (toc>e):
    r = r + 1
    n = n + 1
    x = grad(S,a,c,gfun, r,  x0, e)
```

Данный множитель отвечает за перевес условий над основной функцией. В данном случае он просто инкрементируется, однако если увеличивать или уменьшать прирост можно добиться меньшего или большего количество запусков метода градиентного спуска соответственно. Однако стоит уточнить, что при больших значениях r начинает дольше выполняться функция grad(), отвечающая за метод градиентного спуска, следовательно изменение шага увеличения данной переменной в некоторых примерах может быть нецелесообразным и потому, чтобы в программе каждый раз не вычислять оптимальный прирост, было решено использовать тот, что дан на лекции(то есть инкрементирование).

При острой необходимости уменьшения количества шагов можно сильно увеличить приращение данной переменной, но уменьшать его (с откатом переменных до значений на предыдущей итерации цикла) если функция grad() начинает считать слишком долго (например, делает более 1000 проходов цикла. И при этом должно происходить прерывание выполнения данной функции).