

# Разработка программного обеспечения для решения ЗЛП

Федоров Даниил Михайлович МетОпт 1.1

Вариант 9

Максимизировать  $Z = 2x_1 + x_2 + 3x_3 + 2x_4$   
при условиях:

$$x_1 + 2x_2 + x_3 \leq 11$$

$$x_1 + x_3 + x_4 = 8$$

$$x_2 + x_4 \geq 3$$

Описание работы программы по решению задачи линейного программирования (ЗЛП) с использованием симплекс-метода:

## 1. Загрузка задачи ЗЛП:

Программа принимает на вход txt-файл с текстовым представлением задачи линейного программирования (ЗЛП). Формат файла включает целевую функцию и систему ограничений. Программа извлекает коэффициенты целевой функции, типы ограничений (например,  $\leq$ ,  $\geq$ ,  $=$ ) и правые части ограничений, преобразуя их в удобные для вычислений структуры данных (списки и матрицы).

## 2. Приведение задачи к каноническому виду:

Программа преобразует ЗЛП в каноническую форму, где все ограничения приведены к виду равенств ( $=$ ). Для этого:

- Ограничения с типом  $\leq$  преобразуются путём добавления дополнительных переменных с коэффициентом 1.
- Ограничения с типом  $\geq$  преобразуются путём умножения на -1 и добавления новых переменных. Добавленные переменные становятся частью базиса задачи.

### 3. Составление симплекс-таблицы:

Программа строит начальную симплекс-таблицу, где представлены коэффициенты целевой функции, коэффициенты ограничений и свободные члены (правая часть ограничений). Таблица используется для итераций симплекс-метода.

### 4. Формирование базиса:

Если начальный базис не полный (т.е. некоторые переменные не входят в базис), программа дополнительно формирует базис, приводя соответствующие столбцы таблицы к единичному виду.

### 5. Преобразование отрицательных свободных коэффициентов:

Программа избавляется от строк с отрицательными свободными членами в правой части, выполняя нужные преобразования, чтобы улучшить решение.

### 6. Цикл оптимизации:

В цикле симплекс-метод выполняет следующие шаги:

- Рассчитываются дельты для всех столбцов симплекс-таблицы, которые показывают, насколько улучшится целевая функция при изменении значений переменных.
  - Проверяется оптимальность решения: для задачи максимизации все дельты должны быть отрицательными, для минимизации — положительными.
  - Если решение не оптимально, выбирается разрешающий столбец (столбец с минимальной дельтой для максимизации или максимальной для минимизации), а затем рассчитываются симплекс-отношения для определения разрешающей строки.
  - Строка с минимальным отношением  $Q$  становится базисной, и симплекс-таблица обновляется.
- Цикл продолжается до тех пор, пока не будет найдено оптимальное решение или не будет доказано, что решение невозможно.

### 7. Получение решения:

Когда оптимальное решение найдено, программа извлекает

значения переменных, вычисляет значение целевой функции и возвращает результат в виде списка с оптимальными значениями переменных и значением целевой функции.

## Как использовать программу

Для работы программы текстовый файл с задачей и сама программа должны находиться в одной папке.

Для того, чтобы программа смогла корректно обработать данные они должны быть записаны в таком виде:

1	$2x_1 + x_2 + 3x_3 + 2x_4 \rightarrow \max$
2	$x_1 + 2x_2 + x_3 \leq 11$
3	$x_1 + x_3 + x_4 = 8$
4	$x_2 + x_4 \geq 3$

Изначально:

Максимизировать  $Z = 2x_1 + x_2 + 3x_3 + 2x_4$   
при условиях:

$$\begin{aligned}x_1 + 2x_2 + x_3 &\leq 11 \\x_1 + x_3 + x_4 &= 8 \\x_2 + x_4 &\geq 3\end{aligned}$$

Для максимизации  $\rightarrow \max$ , а для минимизации  $\rightarrow \min$ .

После того, как мы убедились, что условие находится в той же директории и в текстовом файле, а данные внутри записаны корректно, мы можем импортировать наш класс *SimplexMethod* и создать его экземпляр.

```
from Lab_1_code import SimplexMethod  
sm = SimplexMethod()
```

Для того, чтобы загрузить наше условие в программу нужно воспользоваться методом класса

```
sm.load_problem('test_1')
```

где в кавычках указать название нашего файла без формата.

После этого воспользуемся следующим методом, чтобы запустить решение.

```
answer = sm.get_solution()  
print(answer)
```

Готово, программа выведет ответ.

## Пример работы на моём варианте

Создаем файл с задачей

Максимизировать  $Z = 2x_1 + x_2 + 3x_3 + 2x_4$   
при условиях:

$$x_1 + 2x_2 + x_3 \leq 11$$

$$x_1 + x_3 + x_4 = 8$$

$$x_2 + x_4 \geq 3$$

test\_1.txt:

```
2x1 +x2 +3x3 +2x4 -> max  
x1 +2x2 +x3 <= 11  
x1 +x3 +x4 = 8  
x2 +x4 >= 3
```

Инициализируем программу:

```
from Lab_1_code import SimplexMethod

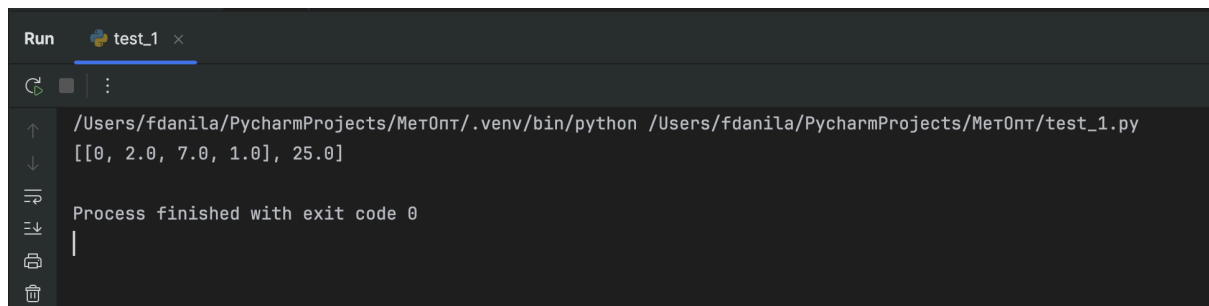
sm = SimplexMethod()

sm.load_problem('test_1')

answer = sm.get_solution()

print(answer)
```

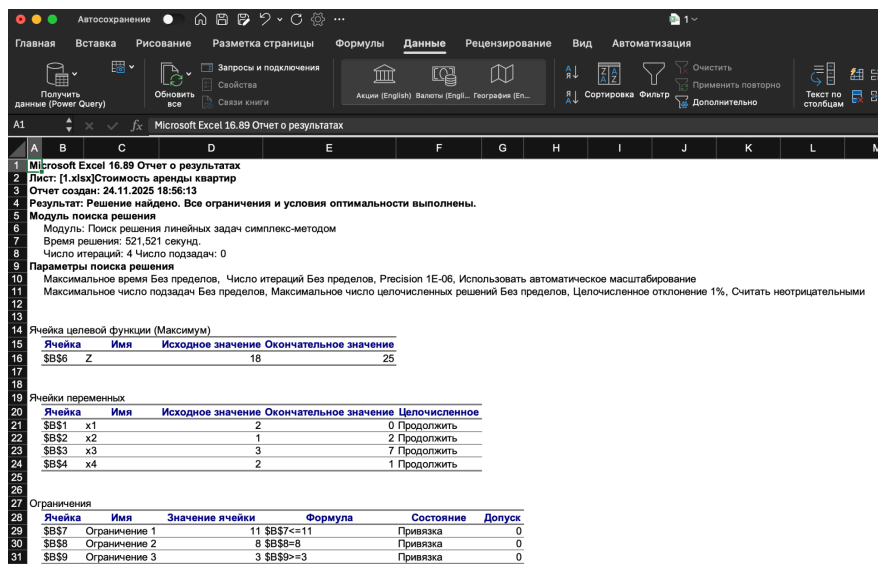
Вывод программы:



```
Run test_1 x
/Users/fdanila/PycharmProjects/Мет0нт/.venv/bin/python /Users/fdanila/PycharmProjects/Мет0нт/test_1.py
[[0, 2.0, 7.0, 1.0], 25.0]
Process finished with exit code 0
```

[[0, 2.0, 7.0, 1.0], 25.0]

Решение через Excel поиском решения



Microsoft Excel 16.89 Отчет о результатах

Лист: [1.xlsx] Стоимость аренды квартир

Отчет создан: 24.11.2025 18:56:13

Результат: Решение найдено. Все ограничения и условия оптимальности выполнены.

Модуль поиска решения

Модуль: Поиск решения линейных задач симплекс-методом

Время решения: 521,521 секунд.

Число итераций: 4 Число подзадач: 0

Параметры поиска решения

Максимальное время Без пределов, Число итераций Без пределов, Precision 1E-06, Использовать автоматическое масштабирование

Максимальное число подзадач Без пределов, Максимальное число целочисленных решений Без пределов, Целочисленное отклонение 1%, Считать неотрицательными

Ячейка	Имя	Исходное значение	Окончательное значение
\$B\$6	Z	18	25

Ячейки переменных

Ячейка	Имя	Исходное значение	Окончательное значение	Целочисленное
\$B\$1	x1	2	0	Продолжить
\$B\$2	x2	1	2	Продолжить
\$B\$3	x3	3	7	Продолжить
\$B\$4	x4	2	1	Продолжить

Ограничения

Ячейка	Имя	Значение ячейки	Формула	Состояние	Допуск
\$B\$7	Ограничение 1	11	\$B\$7<=11	Привязка	0
\$B\$8	Ограничение 2	0	\$B\$8=0	Привязка	0
\$B\$9	Ограничение 3	3	\$B\$9>=3	Привязка	0

Как мы видим, ответы совпадают.

Вывод: Данная работа мне понравилась, хоть и само создание алгоритма было очень муторным, так как в процессе всплывало моментов, которые требовали доработки, изначально хотел сделать программу немного по другому принципу решения, но решил не отклоняться от ТЗ. Во многих моментах, когда я уже не мог понять как доработать ту или иную функцию, я обращался либо к сайтам, которые по похожим алгоритмам решают задачу, либо к БЯМ. Работа помогла мне прокачать навыки работы алгоритмами и дала понять, что человечество придумало такую удобную вещь, как 'Поиск решения' в Excel.