

# Линейное программирование

МЕТОДЫ ВЫПУКЛОЙ ОПТИМИЗАЦИИ

НЕДЕЛЯ 6

Даня Меркулов  
Пётр Остроухов

# **Линейное программирование и симплекс-метод**

## **Семинар**

**Оптимизация для всех! ЦУ**

# Линейное программирование

# Линейное программирование. Общие формы

Для некоторых векторов  $c \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^m$  и матрицы  $A \in \mathbb{R}^{m \times n}$

- Базовая форма задачи линейного программирования:

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} c^\top x \\ & \text{s.t. } Ax \leq b \end{aligned} \quad (\text{LP.Basic})$$

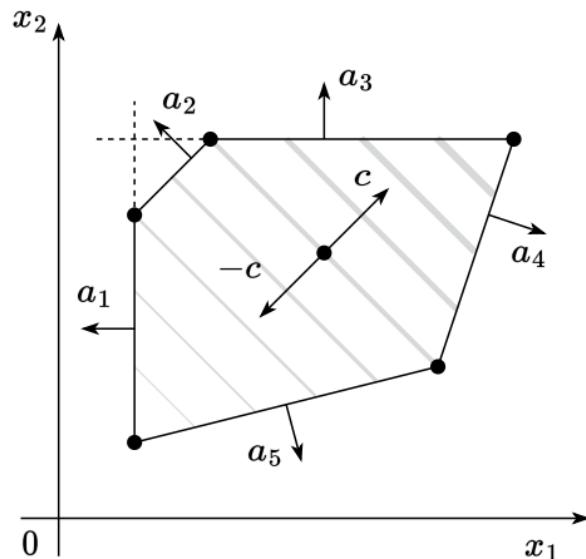


Рисунок 1. Иллюстрация задачи линейного программирования.

# Линейное программирование. Общие формы

Для некоторых векторов  $c \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^m$  и матрицы  $A \in \mathbb{R}^{m \times n}$

- Базовая форма задачи линейного программирования:

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} c^\top x \\ \text{s.t. } & Ax \leq b \end{aligned} \quad (\text{LP.Basic})$$

- Стандартная форма задачи линейного программирования:

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} c^\top x \\ \text{s.t. } & Ax = b \\ & x_i \geq 0, \quad i = 1, \dots, n \end{aligned} \quad (\text{LP.Standard})$$



Рисунок 1. Иллюстрация задачи линейного программирования.



# **Симплекс-метод**

# Основы симплекс-метода



Рисунок 2. Основные понятия симплекс-метода.



Рисунок 3. Изменение базиса симплекс-метода.

## Основные понятия симплекс-метода

- Базис  $B$  является подмножеством  $n$  (целых) чисел между 1 и  $m$ , таких что  $\text{rank } A_B = n$ . Обратите внимание, что мы можем связать подматрицу  $A_B$  и соответствующую правую часть  $b_B$  с базисом  $B$ . Также мы можем получить точку пересечения всех этих гиперплоскостей из базиса:  $x_B = A_B^{-1} b_B$ .

# Основы симплекс-метода



Рисунок 2. Основные понятия симплекс-метода.



Рисунок 3. Изменение базиса симплекс-метода.

## Основные понятия симплекс-метода

- **Базис  $B$**  является подмножеством  $n$  (целых) чисел между 1 и  $m$ , таких что  $\text{rank } A_B = n$ . Обратите внимание, что мы можем связать подматрицу  $A_B$  и соответствующую правую часть  $b_B$  с базисом  $B$ . Также мы можем получить точку пересечения всех этих гиперплоскостей из базиса:  $x_B = A_B^{-1} b_B$ .
- Если  $Ax_B \leq b$ , то базис  $B$  является **допустимым**.

# Основы симплекс-метода

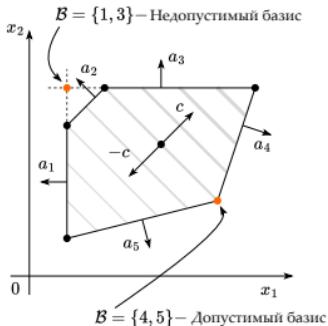


Рисунок 2. Основные понятия симплекс-метода.

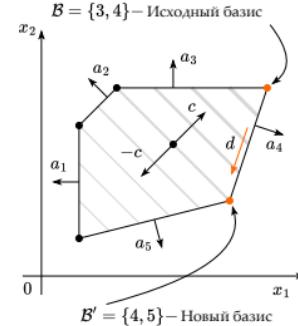


Рисунок 3. Изменение базиса симплекс-метода.

## Основные понятия симплекс-метода

- **Базис  $B$**  является подмножеством  $n$  (целых) чисел между 1 и  $m$ , таких что  $\text{rank } A_B = n$ . Обратите внимание, что мы можем связать подматрицу  $A_B$  и соответствующую правую часть  $b_B$  с базисом  $B$ . Также мы можем получить точку пересечения всех этих гиперплоскостей из базиса:  $x_B = A_B^{-1} b_B$ .
- Если  $Ax_B \leq b$ , то базис  $B$  является **допустимым**.
- Базис  $B$  является **оптимальным**, если  $x_B$  является оптимумом LP .Basic.

# Основы симплекс-метода



Рисунок 4. Основные понятия симплекс-метода.



Рисунок 5. Изменение базиса симплекс-метода.

## Интуиция симплекс-метода

- Алгоритм симплекс-метода последовательно перемещается по рёбрам многогранника, в каждой вершине выбирая ребро, которое обеспечивает наибольшее уменьшение величины  $c^T x$

# Основы симплекс-метода



Рисунок 4. Основные понятия симплекс-метода.



Рисунок 5. Изменение базиса симплекс-метода.

## Интуиция симплекс-метода

- Алгоритм симплекс-метода последовательно перемещается по рёбрам многогранника, в каждой вершине выбирая ребро, которое обеспечивает наибольшее уменьшение величины  $c^T x$
- Процесс либо завершается в некоторой вершине, либо уходит по неограниченному ребру, что означает неограниченность задачи снизу ( $-\infty$  оптимум)

# Основы симплекс-метода



Рисунок 6. Основные понятия симплекс-метода.



Рисунок 7. Изменение базиса симплекс-метода.

Существование решения стандартной задачи линейного программирования

- Если стандартная задача линейного программирования имеет непустое допустимое множество, то существует по крайней мере одна допустимая точка базиса

# Основы симплекс-метода



Рисунок 6. Основные понятия симплекс-метода.

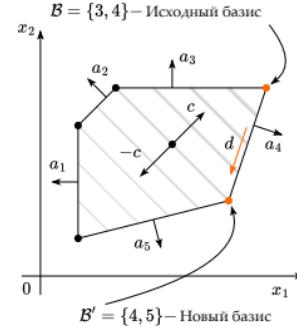


Рисунок 7. Изменение базиса симплекс-метода.



Существование решения стандартной задачи линейного программирования

- Если стандартная задача линейного программирования имеет непустое допустимое множество, то существует по крайней мере одна допустимая точка базиса
- Если стандартная задача линейного программирования имеет решения, то по крайней мере одно из таких решений является оптимальной точкой базиса.

# Основы симплекс-метода

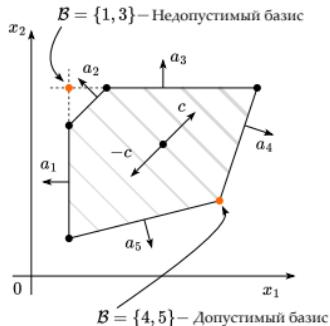


Рисунок 6. Основные понятия симплекс-метода.



Рисунок 7. Изменение базиса симплекс-метода.



Существование решения стандартной задачи линейного программирования

- Если стандартная задача линейного программирования имеет непустое допустимое множество, то существует по крайней мере одна допустимая точка базиса
- Если стандартная задача линейного программирования имеет решения, то по крайней мере одно из таких решений является оптимальной точкой базиса.
- Если стандартная задача линейного программирования является допустимой и ограниченной, то она имеет оптимальное решение.

# Основы симплекс-метода



Рисунок 8. Основные понятия симплекс-метода.

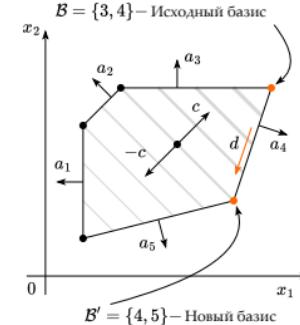


Рисунок 9. Изменение базиса симплекс-метода.

## 💡 Теорема об оптимуме в вершине

Пусть  $\lambda_B$  будут координатами нашего вектора  $c$  в базисе  $B$ :

$$\lambda_B^\top A_B = c^\top \leftrightarrow \lambda_B^\top = c^\top A_B^{-1}$$

Если все компоненты  $\lambda_B$  неотрицательны и  $B$  является допустимым, то  $B$  является оптимальным.

# Примеры задач линейного программирования

# Примеры задач линейного программирования.

## Производственные планы

Предположим, вы думаете о том, чтобы начать бизнес по производству *Продукта X*.

Давайте найдем максимальную недельную прибыль для вашего бизнеса в  Production Plan Problem.

# **Максимальный поток и минимальный разрез**

# Пример задачи о максимальном потоке

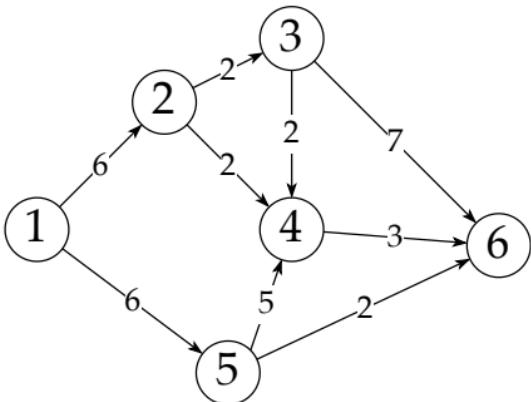


Узлы представляют собой маршрутизаторы, рёбра представляют собой каналы связи; каждому узлу соответствует пропускная способность — узел 1 может общаться с узлом 2 до 6 Мбит/с, узел 2 может общаться с узлом 4 до 2 Мбит/с и т.д.

# Пример задачи о максимальном потоке

Вопрос:

- Сеть узлов и рёбер представляет собой каналы связи, каждый с указанной пропускной способностью.



Узлы представляют собой маршрутизаторы, рёбра представляют собой каналы связи; каждому узлу соответствует пропускная способность — узел 1 может общаться с узлом 2 до 6 Мбит/с, узел 2 может общаться с узлом 4 до 2 Мбит/с и т.д.

# Пример задачи о максимальном потоке

Вопрос:

- Сеть узлов и рёбер представляет собой каналы связи, каждый с указанной пропускной способностью.
- Пример: Может ли узел 1 (источник) общаться с узлом 6 (сток) на 6 Мбит/с, 12 Мбит/с? Какова максимальная скорость?



Узлы представляют собой маршрутизаторы, рёбра представляют собой каналы связи; каждому узлу соответствует пропускная способность — узел 1 может общаться с узлом 2 до 6 Мбит/с, узел 2 может общаться с узлом 4 до 2 Мбит/с и т.д.

# Пример задачи о максимальном потоке

Вопрос:

- Сеть узлов и рёбер представляет собой каналы связи, каждый с указанной пропускной способностью.
- Пример: Может ли узел 1 (источник) общаться с узлом 6 (сток) на 6 Мбит/с, 12 Мбит/с? Какова максимальная скорость?



Узлы представляют собой маршрутизаторы, рёбра представляют собой каналы связи; каждому узлу соответствует пропускная способность — узел 1 может общаться с узлом 2 до 6 Мбит/с, узел 2 может общаться с узлом 4 до 2 Мбит/с и т.д.

# Пример задачи о максимальном потоке



## Вопрос:

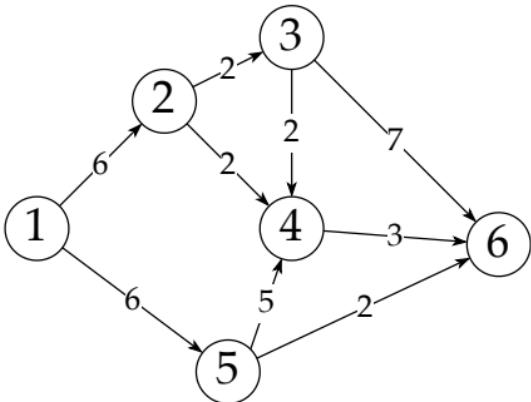
- Сеть узлов и рёбер представляет собой каналы связи, каждый с указанной пропускной способностью.
- Пример: Может ли узел 1 (источник) общаться с узлом 6 (сток) на 6 Мбит/с, 12 Мбит/с? Какова максимальная скорость?

## Матрица пропускных способностей:

$$C = \begin{bmatrix} 0 & 6 & 0 & 0 & 6 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 5 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Узлы представляют собой маршрутизаторы, рёбра представляют собой каналы связи; каждому узлу соответствует пропускная способность — узел 1 может общаться с узлом 2 до 6 Мбит/с, узел 2 может общаться с узлом 4 до 2 Мбит/с и т.д.

# Пример задачи о максимальном потоке



Узлы представляют собой маршрутизаторы, рёбра представляют собой каналы связи; каждому узлу соответствует пропускная способность — узел 1 может общаться с узлом 2 до 6 Мбит/с, узел 2 может общаться с узлом 4 до 2 Мбит/с и т.д.

## Вопрос:

- Сеть узлов и рёбер представляет собой каналы связи, каждый с указанной пропускной способностью.
- Пример: Может ли узел 1 (источник) общаться с узлом 6 (сток) на 6 Мбит/с? 12 Мбит/с? Какова максимальная скорость?

## Матрица пропускных способностей:

$$C = \begin{bmatrix} 0 & 6 & 0 & 0 & 6 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 5 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

**Матрица потоков:**  $X[i, j]$  представляет собой поток от узла  $i$  к узлу  $j$ .

# Пример задачи о максимальном потоке



Узлы представляют собой маршрутизаторы, рёбра представляют собой каналы связи; каждому узлу соответствует пропускная способность — узел 1 может общаться с узлом 2 до 6 Мбит/с, узел 2 может общаться с узлом 4 до 2 Мбит/с и т.д.

## Вопрос:

- Сеть узлов и рёбер представляет собой каналы связи, каждый с указанной пропускной способностью.
- Пример: Может ли узел 1 (источник) общаться с узлом 6 (сток) на 6 Мбит/с? 12 Мбит/с? Какова максимальная скорость?

## Матрица пропускных способностей:

$$C = \begin{bmatrix} 0 & 6 & 0 & 0 & 6 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 5 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

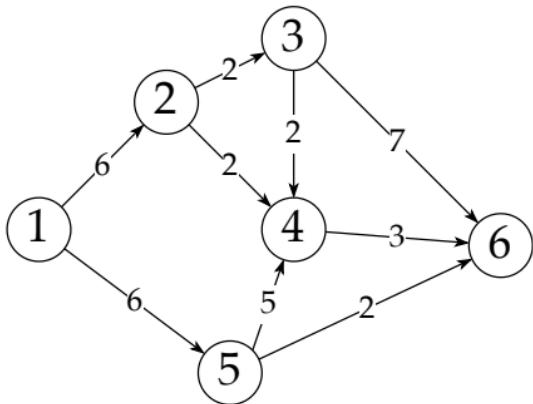
**Матрица потоков:**  $X[i, j]$  представляет собой поток от узла  $i$  к узлу  $j$ .

## Ограничения:

$$0 \leq X \quad X \leq C$$

Сохранение потока:  $\sum_{j=2}^N X(i, j) = \sum_{k=1}^{N-1} X(k, i), \quad i = 2, \dots, N - 1$

# Пример задачи о максимальном потоке



Данная настройка, когда все, что производится источником, будет идти в сток. Поток сети просто сумма всего, что выходит из источника:

$$\sum_{i=2}^N X(1, i) \quad (\text{Поток})$$

# Пример задачи о максимальном потоке



Данная настройка, когда все, что производится источником, будет идти в сток. Поток сети просто сумма всего, что выходит из источника:

$$\sum_{i=2}^N X(1, i) \quad (\text{Поток})$$

максимизировать  $\langle X, S \rangle$

при ограничениях  $-X \leq 0$

$$X \leq C$$

$$\langle X, L_n \rangle = 0, \quad n = 2, \dots, N - 1,$$

(Задача о максимальном потоке)

$L_n$  состоит из одного столбца ( $n$ ) единиц (кроме последней строки) минус одна строка (также  $n$ ) единиц (кроме первого столбца).

$$S = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad L_2 = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & -1 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

# Пример задачи о минимальном разрезе

Разрез сети разделяет вершины на два множества: одно содержит источник (мы называем это множество  $\mathcal{S}$ ), и одно содержит сток. Пропускная способность разреза — это общая величина рёбер, выходящих из  $\mathcal{S}$  — мы разделяем множества, «отрезая поток» по этим рёбрам.



Рёбра в разрезе:  $1 \rightarrow 2, 4 \rightarrow 6$ , и  $5 \rightarrow 6$ . Пропускная способность этого разреза:  $6 + 3 + 2 = 11$ .



Рёбра в разрезе:  $2 \rightarrow 3, 4 \rightarrow 6$ , и  $5 \rightarrow 6$ . Пропускная способность этого разреза:  $2 + 3 + 2 = 7$ .

Теорема о максимальном потоке и минимальном разрезе.

Максимальное значение потока  $s-t$  равно минимальной пропускной способности всех  $s-t$  разрезов.



# Примеры задач линейного программирования.

## Различные приложения

Посмотрите на различные практические приложения задач линейного программирования и симплекс-метода в [Related Collab Notebook](#).