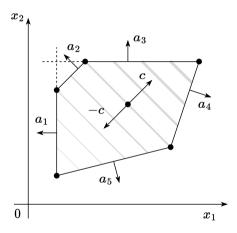


Примеры задач линейного программирования



Что такое линейное программирование?

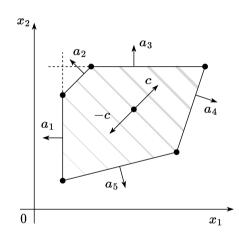


В общем случае все задачи с линейной целевой функцией и линейными функциональными ограничениями можно считать задачами линейного программирования. Однако существует несколько стандартных формулировок.

$$\min_{x \in \mathbb{R}^n} c^\top x$$
 s.t. $Ax \leq b$

для некоторых векторов $c\in\mathbb{R}^n$, $b\in\mathbb{R}^m$ и матрицы $A\in\mathbb{R}^{m\times n}$, где неравенства — покомпонентные. Мы будем часто использовать эту формулировку для построения интуиции.

Что такое линейное программирование?



В общем случае все задачи с линейной целевой функцией и линейными функциональными ограничениями можно считать задачами линейного программирования. Однако существует несколько стандартных формулировок.

$$\min_{x \in \mathbb{R}^n} c^\top x$$
 (LP.Basic) s.t. $Ax < b$

для некоторых векторов $c\in\mathbb{R}^n$, $b\in\mathbb{R}^m$ и матрицы $A\in\mathbb{R}^{m\times n}$, где неравенства — покомпонентные. Мы будем часто использовать эту формулировку для построения интуиции. Широко используется **стандартная форма** записи задачи линейного программирования. Пусть заданы векторы $c\in\mathbb{R}^n$, $b\in\mathbb{R}^m$ и матрица $A\in\mathbb{R}^{m\times n}$.

$$\min_{x \in \mathbb{R}^n} c^\top x$$

s.t. Ax = b (LP.Standard)

$$x_i \ge 0, \ i = 1, \dots, n$$

⊕ ი €

Пример: задача о диете



 $c\in\mathbb{R}^p$, цена за 100г

 $r\in\mathbb{R}^n$, ограничения

 $x \in \mathbb{R}^p$, количество продуктов

$$\min_{x \in \mathbb{R}^p} c^T x$$

$$egin{aligned} Wx \succeq r \ x \succeq 0 \end{aligned}$$

Пример: задача о диете Белки Жиры Количество на 100г Углеводы $W \in \mathbb{R}^{n imes p}$ Калории Витамин С $\min c^T x$ $c \in \mathbb{R}^p$, цена за 100г $Wx \succ r$ $r \in \mathbb{R}^n$, ограничения

 $x \succ 0$

Представьте, что вам нужно составить план диеты из некоторых продуктов: бананы, пироги, курица, яйца, рыба. Каждый из продуктов имеет свой вектор питательных веществ. Таким образом, все питательные вещества можно представить в виде матрицы W.



 $x \in \mathbb{R}^p$, количество продуктов



Представьте, что вам нужно составить план диеты из некоторых продуктов: бананы, пироги, курица, яйца, рыба. Каждый из продуктов имеет свой вектор питательных веществ. Таким образом, все питательные вещества можно представить в виде матрицы W. Предположим, что у нас есть вектор требований для каждого питательного вещества $r \in \mathbb{R}^n$. Нам нужно найти самую дешёвую диету, которая удовлетворяет всем требованиям:





Представьте, что вам нужно составить план диеты из некоторых продуктов: бананы, пироги, курица, яйца, рыба. Каждый из продуктов имеет свой вектор питательных веществ. Таким образом, все питательные вещества можно представить в виде матрицы W. Предположим, что у нас есть вектор требований для каждого питательного вещества $r \in \mathbb{R}^n$. Нам нужно найти самую дешёвую диету, которая удовлетворяет всем требованиям:

$$\begin{aligned} \min_{x \in \mathbb{R}^p} c^\top x \\ \text{s.t.} \ \ Wx \succeq r \\ x_i \geq 0, \ i = 1, \dots, p \end{aligned}$$

POpen In Colab

 $x \succ 0$

 $x \in \mathbb{R}^p$, количество продуктов

 $r \in \mathbb{R}^n$, ограничения

Минимизация выпуклой функции как задача линейного программирования

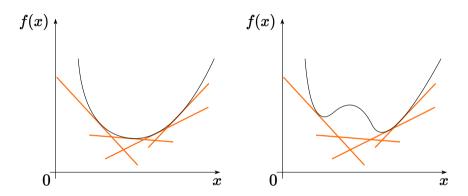


Рис. 1: Как задача линейного программирования может помочь с общей задачей выпуклой оптимизации

Функция выпукла, если она может быть представлена как поточечный максимум линейных функций.



Минимизация выпуклой функции как задача линейного программирования

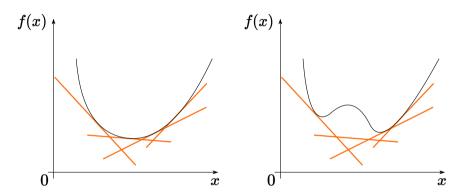


Рис. 1: Как задача линейного программирования может помочь с общей задачей выпуклой оптимизации

- Функция выпукла, если она может быть представлена как поточечный максимум линейных функций.
- В пространствах большой размерности аппроксимация может потребовать огромного количества функций.

Примеры задач линейного программирования

Минимизация выпуклой функции как задача линейного программирования

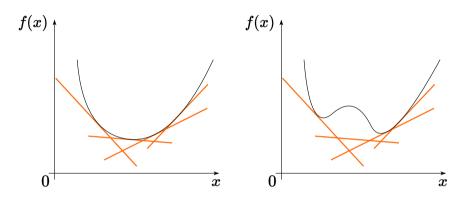


Рис. 1: Как задача линейного программирования может помочь с общей задачей выпуклой оптимизации

- Функция выпукла, если она может быть представлена как поточечный максимум линейных функций.
- В пространствах большой размерности аппроксимация может потребовать огромного количества функций. Существуют более эффективные солверы для выпуклой оптимизации (не сводящиеся к LP).

Типичная транспортная задача заключается в распределении товара от производителей к потребителям. Цель состоит в минимизации общих затрат на транспортировку при соблюдении ограничений на количество товара на каждом источнике и удовлетворении требований к спросу на каждом пункте назначения.



Рис. 2: Карта Западной Европы. �Open In Colab





Пункт назначения / Источник	Арнем [€ /тонна]	Гауда [€ /тонна]	Спрос [тонн]
Лондон	n/a	2.5	125
Берлин	2.5	n/a	175
Маастрихт	1.6	2.0	225
Амстердам	1.4	1.0	250
Утрехт	0.8	1.0	225
Гаага	1.4	8.0	200
Макс. производство [тонн]	550	700	

Минимизировать: Стоимость =
$$\sum_{c \in \Pi \text{VHKTЫ назначения}} \sum_{s \in \Pi \text{CTO-ЧНИКИ}} T[c,s]x[c,s]$$



Пункт назначения / Источник	Арнем [€ /тонна]	Гауда [€ /тонна]	Спрос [тонн]
Лондон	n/a	2.5	125
Берлин	2.5	n/a	175
Маастрихт	1.6	2.0	225
Амстердам	1.4	1.0	250
Утрехт	0.8	1.0	225
Гаага	1.4	8.0	200
Макс. производство [тонн]	550	700	

Минимизировать: Стоимость =
$$\sum_{c \in \Pi \text{ункты назначения } s \in \text{Источники}} T[c,s]x[c,s]$$

$$\sum_{c \in \mathsf{\Pi}\mathsf{УHKTЫ}\ \mathsf{Haзhaчehu}\mathsf{y}} x[c,s] \leq \mathsf{\Pi}\mathsf{octabka}[s] \qquad \forall s \in \mathsf{Исtoчникu}$$



Пункт назначения / Источник	Арнем [€ /тонна]	Гауда [€ /тонна]	Спрос [тонн]
Лондон	n/a	2.5	125
Берлин	2.5	n/a	175
Маастрихт	1.6	2.0	225
Амстердам	1.4	1.0	250
Утрехт	0.8	1.0	225
Гаага	1.4	8.0	200
Макс. производство [тонн]	550	700	

$$c$$
 \in Пункты назначения s \in Источники $\sum x[c,s] \leq \mathsf{Поставка}[s] \qquad orall s \in \mathsf{Источ}$ ники

Минимизировать: Стоимость = $\sum T[c,s]x[c,s]$

$$\sum \quad x[c,s] = \mathsf{Cnpoc}[c] \qquad orall c \in \mathsf{Пункты}$$
 назначения

Задачу можно представить в виде следующего графа:

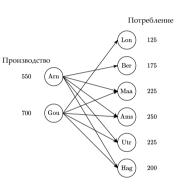


Рис. 3: Граф, связанный с задачей

 $c \in \Pi$ үнкты назначения

s∈Источники

Как получить задачу линейного программирования?





• Максимум-минимум

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} c^\top x & & \max_{x \in \mathbb{R}^n} -c^\top x \\ \text{s.t. } & Ax \leq b & & \text{s.t. } & Ax \leq b \end{aligned}$$



• Максимум-минимум

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} c^\top x & & \max_{x \in \mathbb{R}^n} -c^\top x \\ \text{s.t. } & Ax \leq b & & \text{s.t. } & Ax \leq b \end{aligned}$$

• Равенство к неравенству

$$Ax = b \leftrightarrow \begin{cases} Ax \le b \\ Ax \ge b \end{cases}$$



Максимум-минимум

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} c^\top x & & \max_{x \in \mathbb{R}^n} -c^\top x \\ \text{s.t. } & Ax \leq b & & \text{s.t. } & Ax \leq b \end{aligned}$$

Равенство к неравенству

$$Ax = b \leftrightarrow \begin{cases} Ax \le b \\ Ax \ge b \end{cases}$$

Неравенство к равенству, увеличивая размерность задачи на m.

$$Ax \le b \leftrightarrow \begin{cases} Ax + z = b \\ z \ge 0 \end{cases}$$



Максимум-минимум

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} c^\top x & & \max_{x \in \mathbb{R}^n} -c^\top x \\ \text{s.t. } & Ax \leq b & & \text{s.t. } & Ax \leq b \end{aligned}$$

Равенство к неравенству

$$Ax = b \leftrightarrow \begin{cases} Ax \le b \\ Ax \ge b \end{cases}$$

Неравенство к равенству, увеличивая размерность задачи на m.

$$Ax \le b \leftrightarrow \begin{cases} Ax + z = b \\ z \ge 0 \end{cases}$$

Неотрицательные переменные

$$x \leftrightarrow \begin{cases} x = x_+ - x_- \\ x_+ \ge 0 \\ x_- \ge 0 \end{cases}$$

Пример: задача аппроксимации Чебышева

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_{\infty} \leftrightarrow \min_{x \in \mathbb{R}^n} \max_i |a_i^T x - b_i|$$

Можно записать эквивалентную задачу линейного программирования с заменой максимальной координаты вектора:



Пример: задача аппроксимации Чебышева

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_{\infty} \leftrightarrow \min_{x \in \mathbb{R}^n} \max_i |a_i^Tx - b_i|$$

Можно записать эквивалентную задачу линейного программирования с заменой максимальной координаты вектора:

$$\begin{split} \min_{t \in \mathbb{R}, x \in \mathbb{R}^n} t \\ \text{s.t. } a_i^T x - b_i \leq t, \ i = 1, \dots, m \\ - a_i^T x + b_i \leq t, \ i = 1, \dots, m \end{split}$$



Пример: задача ℓ_1 аппроксимации

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_1 \leftrightarrow \min_{x \in \mathbb{R}^n} \sum_{i=1}^m |a_i^Tx - b_i|$$

Можно записать эквивалентную задачу линейного программирования с заменой суммы координат вектора:



Пример: задача ℓ_1 аппроксимации

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_1 \leftrightarrow \min_{x \in \mathbb{R}^n} \sum_{i=1}^m |a_i^T x - b_i|$$

Можно записать эквивалентную задачу линейного программирования с заменой суммы координат вектора:

$$\begin{aligned} & \min_{t \in \mathbb{R}^m, x \in \mathbb{R}^n} \mathbf{1}^T t \\ \text{s.t. } & a_i^T x - b_i \leq t_i, \ i = 1, \dots, m \\ & - a_i^T x + b_i \leq t_i, \ i = 1, \dots, m \end{aligned}$$



Задача смешивания: от нелинейных ограничений к ЛП 1 Производственное предприятие получает заказ на 100 литров

раствора с определённой концентрацией (например, 4% сахарного раствора). На складе есть:

Компонент	Caxap (%)	Стоимость (\$/л)
Концентрат А (Добрый кола)	10.6	1.25
Концентрат В (Север кола)	4.5	1.02
Вода (Псыж)	0.0	0.62
вода (певіж)	0.0	0.02

Цель: Найти смесь с минимальной стоимостью, которая

удовлетворит заказ.

Целевая функция



Задача смешивания: от нелинейных ограничений к ЛП 1 Производственное предприятие получает заказ на 100 литров

раствора с определённой концентрацией (например, 4% сахарного раствора). На складе есть:

,	Стоимость (\$/л)
10.6	1.25
4.5	1.02
0.0	0.62
	4.5

Цель: Найти смесь с минимальной стоимостью, которая удовлетворит заказ.

Целевая функция

Минимизировать стоимость:

$$\mathsf{Cost} = \sum_{c \in C} x_c P_c$$

где x_c — объём используемого компонента c, и P_c — его цена.

Задача смешивания: от нелинейных ограничений к ЛП 1

Производственное предприятие получает заказ на 100 литров раствора с определённой концентрацией (например, 4% сахарного Ограничение на объём раствора). На складе есть:

Компонент	Caxap (%)	Стоимость (\$/л)
Концентрат А (Добрый кола)	10.6	1.25
Концентрат В (Север кола)	4.5	1.02
Вода (Псыж)	0.0	0.62
эода (позик)		0.02

Цель: Найти смесь с минимальной стоимостью, которая удовлетворит заказ.

Целевая функция

Минимизировать стоимость:

$$\mathsf{Cost} = \sum_{c \in C} x_c P_c$$

где x_c — объём используемого компонента c, и P_c — его цена.

Задача смешивания: от нелинейных ограничений к ЛП 1 Производственное предприятие получает заказ на 100 литров

раствора с определённой концентрацией (например, 4% сахарного раствора). На складе есть:

Компонент	Caxap (%)	Стоимость (\$/л)
Концентрат А (Добрый кола)	10.6	1.25
Концентрат В (Север кола)	4.5	1.02
Вода (Псыж)	0.0	0.62

Ограничение на объём Убедитесь, что общий объём V:

 $V = \sum_{c \in C} x_c$

Ограничение на состав

Цель: Найти смесь с минимальной стоимостью, которая удовлетворит заказ.

Целевая функция

Минимизировать стоимость:

$$\mathsf{Cost} = \sum_{c \in C} x_c P_c$$

где x_c — объём используемого компонента c, и P_c — его цена.

Задача смешивания: от нелинейных ограничений к ЛП 1 Производственное предприятие получает заказ на 100 литров

раствора с определённой концентрацией (например, 4% сахарного раствора). На складе есть:

Цель: Найти смесь с минимальной стоимостью, которая

Компонент	Caxap (%)	Стоимость (\$/л)	
Концентрат А (Добрый кола)	10.6	1.25	
Концентрат В (Север кола)	4.5	1.02	(
Вода (Псыж)	0.0	0.62	`

Убедитесь, что общий объём V:

$$V = \sum_{c \in C} x_c$$

Ограничение на состав Убедитесь, что содержание сахара — 4%:

Ограничение на объём

$$\bar{A} = \frac{\sum_{c \in C} x_c A_c}{\sum_{c \in C} x_c}$$

 $\mathsf{Cost} = \sum_{c \in C} x_c P_c$

где
$$x_c$$
 — объём используемого компонента c , и P_c — его цена.

1 Source

удовлетворит заказ. Целевая функция

Минимизировать стоимость:

 $f \to \min_{x,y,z}$ Как получить задачу линейного программирования?

Задача смешивания: от нелинейных ограничений к ЛП 1 Производственное предприятие получает заказ на 100 литров

раствора с определённой концентрацией (например, 4% сахарного раствора). На складе есть:

Компонент

удовлетворит заказ.

Целевая функция

Минимизировать стоимость:

Caxap (%) Стоимость (\$/л) Концентрат А (Добрый кола) 10.6 1 25 Концентрат В (Север кола) 4.5 1.02 Вода (Псыж) 0.0 0.62

Ограничение на состав

Ограничение на объём

Убедитесь, что общий объём V:

Убедитесь, что содержание сахара — 4%:

 $\bar{A} = \frac{\sum_{c \in C} x_c A_c}{\sum_{c} x_c}$

Линеаризованная версия:

 $0 = \sum_{c \in C} x_c (A_c - \bar{A})$

 $V = \sum_{c \in C} x_c$

Это можно решить с помощью линейного программирования. **%**Код

где x_c — объём используемого компонента c, и P_c — его цена.

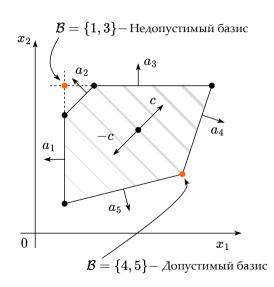
 $\mathsf{Cost} = \sum_{c \in C} x_c P_c$

Цель: Найти смесь с минимальной стоимостью, которая

Симплекс-метод



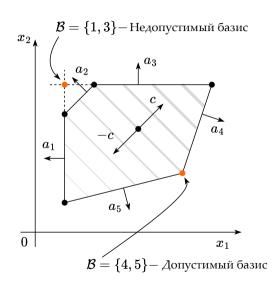




Рассмотрим следующую простую формулировку задачи линейного программирования:

$$\min_{x \in \mathbb{R}^n} c^\top x$$
 s.t. $Ax \leq b$

• Определение: **базис** \mathcal{B} — это подмножество n (целых) чисел между 1 и m, такое что $\mathrm{rank} A_{\mathcal{B}} = n$.

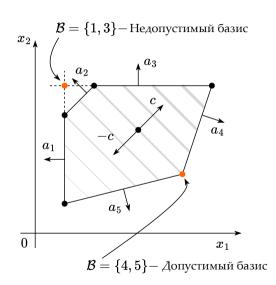


Рассмотрим следующую простую формулировку задачи линейного программирования:

$$\min_{x \in \mathbb{R}^n} c^\top x$$
 s.t. $Ax \leq b$

- Определение: **базис** \mathcal{B} это подмножество n (целых) чисел между 1 и m, такое что $\mathrm{rank} A_{\mathcal{B}} = n$.
- Обратите внимание, что мы можем связать подматрицу $A_{\mathcal{B}}$ и соответствующую правую часть $b_{\mathcal{B}}$ с базисом $\mathcal{B}.$



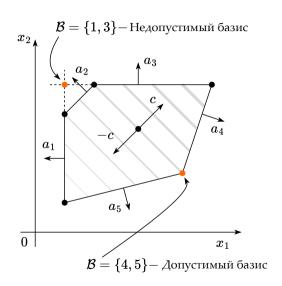


Рассмотрим следующую простую формулировку задачи линейного программирования:

$$\min_{x \in \mathbb{R}^n} c^\top x$$
 s.t. $Ax \leq b$

- ullet Определение: **базис** \mathcal{B} это подмножество n (целых) чисел между 1 и m, такое что $\mathrm{rank} A_{\mathcal{R}} = n$.
- Обратите внимание, что мы можем связать подматрицу $A_{\mathcal{B}}$ и соответствующую правую часть $b_{\mathcal{B}}$ с базисом $\mathcal{B}.$
- Также мы можем получить точку пересечения всех этих гиперплоскостей из базиса: $x_{\mathcal{B}} = A_{\mathcal{B}}^{-1}b_{\mathcal{B}}.$



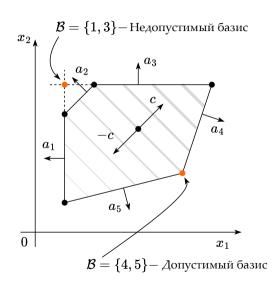


Рассмотрим следующую простую формулировку задачи линейного программирования:

$$\min_{x \in \mathbb{R}^n} c^\top x$$
 s.t. $Ax \leq b$

- ullet Определение: **базис** \mathcal{B} это подмножество n (целых) чисел между 1 и m, такое что $\mathrm{rank} A_{\mathcal{R}} = n$.
- Обратите внимание, что мы можем связать подматрицу $A_{\mathcal{B}}$ и соответствующую правую часть $b_{\mathcal{B}}$ с базисом $\mathcal{B}.$
- Также мы можем получить точку пересечения всех этих гиперплоскостей из базиса: $x_{\mathcal{B}} = A_{\mathcal{B}}^{-1}b_{\mathcal{B}}.$
- ullet Если $Ax_{\mathcal{B}} \leq b$, то базис \mathcal{B} является **допустимым**.

Симплекс-метод



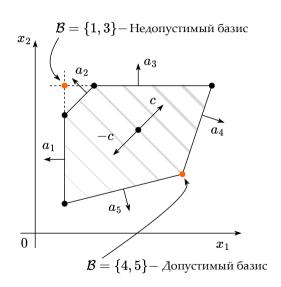
Рассмотрим следующую простую формулировку задачи линейного программирования:

$$\min_{x \in \mathbb{R}^n} c^\top x$$
 s.t. $Ax \leq b$

- Определение: **базис** \mathcal{B} это подмножество n (целых) чисел между 1 и m, такое что $\mathrm{rank} A_{\mathcal{R}} = n$.
- Обратите внимание, что мы можем связать подматрицу $A_{\mathcal{B}}$ и соответствующую правую часть $b_{\mathcal{B}}$ с базисом $\mathcal{B}.$
- Также мы можем получить точку пересечения всех этих гиперплоскостей из базиса: $x_{\mathcal{B}} = A_{\mathcal{B}}^{-1}b_{\mathcal{B}}.$
- ullet Если $Ax_{\mathcal{B}} \leq b$, то базис ${\mathcal{B}}$ является **допустимым**.
- \mathcal{B} Базис \mathcal{B} оптимален, если $x_{\mathcal{B}}$ является решением задачи LP.Inequality.



Геометрия симплекс-метода

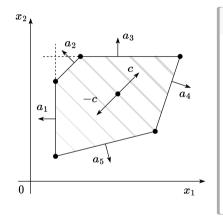


Рассмотрим следующую простую формулировку задачи линейного программирования:

$$\min_{x \in \mathbb{R}^n} c^\top x$$
 s.t. $Ax < b$

- Определение: **базис** \mathcal{B} это подмножество n (целых) чисел между 1 и m, такое что $\operatorname{rank} A_{\mathcal{R}} = n$.
- Обратите внимание, что мы можем связать подматрицу $A_{\mathcal{B}}$ и соответствующую правую часть $b_{\mathcal{B}}$ с базисом $\mathcal{B}.$
- Также мы можем получить точку пересечения всех этих гиперплоскостей из базиса: $x_{\mathcal{B}} = A_{\mathcal{B}}^{-1} b_{\mathcal{B}}$.
- Если $Ax_{\mathcal{B}} \leq b$, то базис \mathcal{B} является **допустимым**.
- Р Базис $\mathcal B$ оптимален, если $x_{\mathcal B}$ является решением задачи LP.Inequality.
- $x_{\mathcal{B}}$ называют **базисной точкой** или базисным решением (иногда её тоже называют **базисом**).

2 Симплекс-метод



Theorem
 Если задача линейного программирования в стандартной форме имеет непустое бюджетное множество, то существует

по крайней мере одна допустимая базисная точка.



i Theorem

- 1. Если задача линейного программирования в стандартной форме имеет непустое бюджетное множество, то существует по крайней мере одна допустимая базисная точка.
- 2. Если задача линейного программирования в стандартной форме имеет решения, то по крайней мере одно из таких решений является оптимальной базисной точкой.



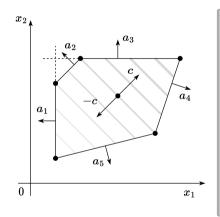
i Theorem

- 1. Если задача линейного программирования в стандартной форме имеет непустое бюджетное множество, то существует по крайней мере одна допустимая базисная точка.
- 2. Если задача линейного программирования в стандартной форме имеет решения, то по крайней мере одно из таких решений является оптимальной базисной точкой.
- Если задача линейного программирования в стандартной форме допустима и ограничена, то она имеет оптимальное решение.



i Theorem

- 1. Если задача линейного программирования в стандартной форме имеет непустое бюджетное множество, то существует по крайней мере одна допустимая базисная точка.
- 2. Если задача линейного программирования в стандартной форме имеет решения, то по крайней мере одно из таких решений является оптимальной базисной точкой.
- Если задача линейного программирования в стандартной форме допустима и ограничена, то она имеет оптимальное решение.



i Theorem

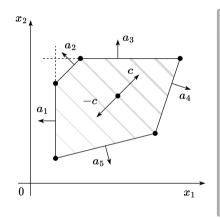
- 1. Если задача линейного программирования в стандартной форме имеет непустое бюджетное множество, то существует по крайней мере одна допустимая базисная точка.
- 2. Если задача линейного программирования в стандартной форме имеет решения, то по крайней мере одно из таких решений является оптимальной базисной точкой.
- Если задача линейного программирования в стандартной форме допустима и ограничена, то она имеет оптимальное решение.

Для доказательства см. теорему 13.2 в Numerical Optimization by Jorge Nocedal and Stephen J. Wright

Верхнеуровневая идея симплекс-метода:

• Убедитесь, что вы находитесь в вершине.





1 Theorem

- 1. Если задача линейного программирования в стандартной форме имеет непустое бюджетное множество, то существует по крайней мере одна допустимая базисная точка.
- 2. Если задача линейного программирования в стандартной форме имеет решения, то по крайней мере одно из таких решений является оптимальной базисной точкой.
- 3. Если задача линейного программирования в стандартной форме допустима и ограничена, то она имеет оптимальное решение.

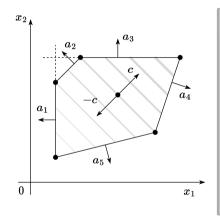
Для доказательства см. теорему 13.2 в Numerical Optimization by Jorge Nocedal and Stephen J. Wright

Верхнеуровневая идея симплекс-метода:

- Убедитесь, что вы находитесь в вершине.
- Проверьте оптимальность.

Симплекс-метол





i Theorem

- 1. Если задача линейного программирования в стандартной форме имеет непустое бюджетное множество, то существует по крайней мере одна допустимая базисная точка.
- 2. Если задача линейного программирования в стандартной форме имеет решения, то по крайней мере одно из таких решений является оптимальной базисной точкой.
- Если задача линейного программирования в стандартной форме допустима и ограничена, то она имеет оптимальное решение.

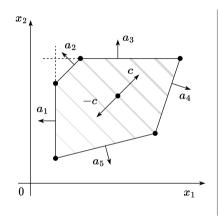
Для доказательства см. теорему $13.2\ \mathrm{B}\ \mathrm{Numerical}\ \mathrm{Optimization}$ by Jorge Nocedal and Stephen J. Wright

Верхнеуровневая идея симплекс-метода:

- Убедитесь, что вы находитесь в вершине.
- Проверьте оптимальность.

 Если необходимо, перейдите к другой вершине (измените базис).





i Theorem

- 1. Если задача линейного программирования в стандартной форме имеет непустое бюджетное множество, то существует по крайней мере одна допустимая базисная точка.
- 2. Если задача линейного программирования в стандартной форме имеет решения, то по крайней мере одно из таких решений является оптимальной базисной точкой.
- Если задача линейного программирования в стандартной форме допустима и ограничена, то она имеет оптимальное решение.

Для доказательства см. теорему 13.2 в Numerical Optimization by Jorge Nocedal and Stephen J. Wright

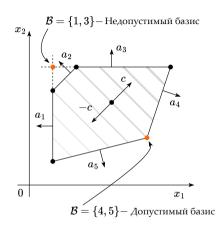
- Убедитесь, что вы находитесь в вершине.
- Проверьте оптимальность.

- Если необходимо, перейдите к другой вершине (измените базис).
- Повторяйте, пока не сойдётесь.

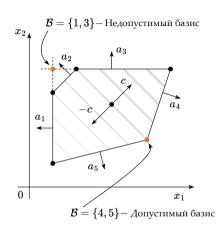








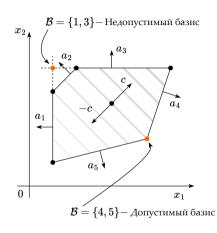




Поскольку у нас есть базис, мы можем разложить наш целевой вектор c в этом базисе и найти скалярные коэффициенты $\lambda_{\mathcal{B}}$:

$$\lambda_{\mathcal{B}}^T A_{\mathcal{B}} = c^T \leftrightarrow \lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$$





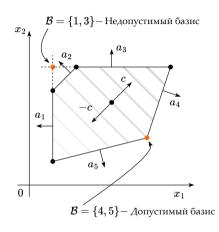
Поскольку у нас есть базис, мы можем разложить наш целевой вектор c в этом базисе и найти скалярные коэффициенты $\lambda_{\mathcal{B}}$:

$$\lambda_{\mathcal{B}}^T A_{\mathcal{B}} = c^T \leftrightarrow \lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$$

i Theorem

Если все компоненты $\lambda_{\mathcal{B}}$ неположительны и \mathcal{B} допустим, то \mathcal{B} оптимален.





Поскольку у нас есть базис, мы можем разложить наш целевой вектор c в этом базисе и найти скалярные коэффициенты $\lambda_{\mathcal{B}}$:

$$\lambda_{\mathcal{B}}^T A_{\mathcal{B}} = c^T \leftrightarrow \lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$$

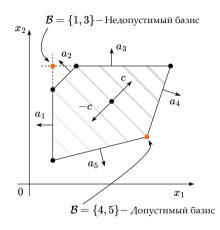
1 Theorem

Если все компоненты $\lambda_{\mathcal{B}}$ неположительны и \mathcal{B} допустим, то \mathcal{B} оптимален.

Доказательство Предположим противное, то есть $\lambda_{\mathcal{B}} \leq 0$ и \mathcal{B} допустим, но не оптимален.

$$\exists x^* : Ax^* \le b, c^T x^* < c^T x_{\mathcal{B}}$$

Симплекс-метод



Поскольку у нас есть базис, мы можем разложить наш целевой вектор c в этом базисе и найти скалярные коэффициенты $\lambda_{\mathcal{B}}$:

$$\lambda_{\mathcal{B}}^T A_{\mathcal{B}} = c^T \leftrightarrow \lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$$

1 Theorem

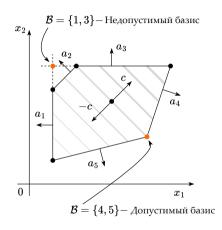
Если все компоненты $\lambda_{\mathcal{B}}$ неположительны и \mathcal{B} допустим, то \mathcal{B} оптимален.

Доказательство Предположим противное, то есть $\lambda_{\mathcal{B}} \leq 0$ и \mathcal{B} допустим, но не оптимален.

$$\exists x^* : Ax^* \leq b, c^T x^* < c^T x_{\mathcal{B}}$$

$$A_{\mathcal{B}} x^* \leq b_{\mathcal{B}}$$





Поскольку у нас есть базис, мы можем разложить наш целевой вектор c в этом базисе и найти скалярные коэффициенты $\lambda_{\mathcal{B}}$:

$$\lambda_{\mathcal{B}}^T A_{\mathcal{B}} = c^T \leftrightarrow \lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$$

1 Theorem

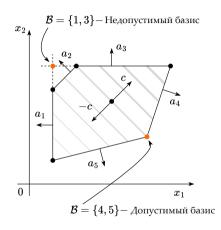
Если все компоненты $\lambda_{\mathcal{B}}$ неположительны и \mathcal{B} допустим, то \mathcal{B} оптимален.

Доказательство Предположим противное, то есть $\lambda_{\mathcal{B}} \leq 0$ и \mathcal{B} допустим, но не оптимален.

$$\exists x^* : Ax^* \leq b, c^T x^* < c^T x_{\mathcal{B}}$$

$$A_{\mathcal{B}} x^* \leq b_{\mathcal{B}} \mid \lambda_{\mathcal{B}}^T \cdot \leq 0$$





Поскольку у нас есть базис, мы можем разложить наш целевой вектор c в этом базисе и найти скалярные коэффициенты $\lambda_{\mathcal{B}}$:

$$\lambda_{\mathcal{B}}^T A_{\mathcal{B}} = c^T \leftrightarrow \lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$$

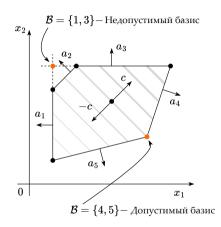
1 Theorem

Если все компоненты $\lambda_{\mathcal{B}}$ неположительны и \mathcal{B} допустим, то \mathcal{B} оптимален.

Доказательство Предположим противное, то есть $\lambda_{\mathcal{B}} \leq 0$ и \mathcal{B} допустим, но не оптимален.

$$\begin{split} \exists x^*: Ax^* \leq b, c^Tx^* < c^Tx_{\mathcal{B}} \\ A_{\mathcal{B}}x^* \leq b_{\mathcal{B}} \mid \lambda_{\mathcal{B}}^T \cdot \leq 0 \\ \lambda_{\mathcal{B}}^TA_{\mathcal{B}}x^* \geq \lambda_{\mathcal{B}}^Tb_{\mathcal{B}} \end{split}$$

Симплекс-метод



Поскольку у нас есть базис, мы можем разложить наш целевой вектор c в этом базисе и найти скалярные коэффициенты $\lambda_{\mathcal{B}}$:

$$\lambda_{\mathcal{B}}^T A_{\mathcal{B}} = c^T \leftrightarrow \lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$$

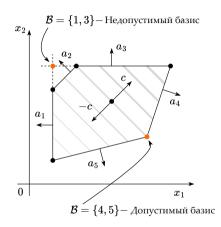
1 Theorem

Если все компоненты $\lambda_{\mathcal{B}}$ неположительны и \mathcal{B} допустим, то \mathcal{B} оптимален.

Доказательство Предположим противное, то есть $\lambda_{\mathcal{B}} \leq 0$ и \mathcal{B} допустим, но не оптимален.

$$\begin{split} \exists x^* : Ax^* \leq b, c^Tx^* < c^Tx_{\mathcal{B}} \\ A_{\mathcal{B}}x^* \leq b_{\mathcal{B}} \mid \lambda_{\mathcal{B}}^T \cdot \leq 0 \\ \lambda_{\mathcal{B}}^TA_{\mathcal{B}}x^* \geq \lambda_{\mathcal{B}}^Tb_{\mathcal{B}} \\ c^Tx^* \geq \lambda_{\mathcal{B}}^TA_{\mathcal{B}}x_{\mathcal{B}} \end{split}$$





Поскольку у нас есть базис, мы можем разложить наш целевой вектор c в этом базисе и найти скалярные коэффициенты $\lambda_{\mathcal{B}}$:

$$\lambda_{\mathcal{B}}^T A_{\mathcal{B}} = c^T \leftrightarrow \lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$$

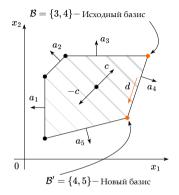
Theorem

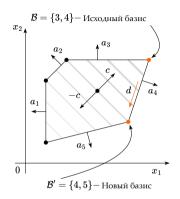
Если все компоненты $\lambda_{\mathcal{B}}$ неположительны и \mathcal{B} допустим, то $\mathcal B$ оптимален.

Доказательство Предположим противное, то есть $\lambda_{\mathcal{B}} \leq 0$ и \mathcal{B} допустим, но не оптимален.

$$\begin{split} \exists x^* : Ax^* \leq b, c^Tx^* < c^Tx_{\mathcal{B}} \\ A_{\mathcal{B}}x^* \leq b_{\mathcal{B}} \mid \lambda_{\mathcal{B}}^T \cdot \leq 0 \\ \lambda_{\mathcal{B}}^T A_{\mathcal{B}}x^* \geq \lambda_{\mathcal{B}}^T b_{\mathcal{B}} \\ c^Tx^* \geq \lambda_{\mathcal{B}}^T A_{\mathcal{B}}x_{\mathcal{B}} \\ c^Tx^* \geq c^Tx_{\mathcal{B}} \end{split}$$

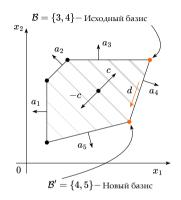
Симплекс-метол



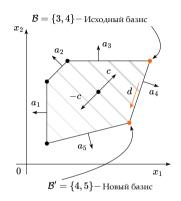


Предположим, что некоторые из коэффициентов $\lambda_{\mathcal{B}}$ положительны. В этом случае необходимо осуществить переход по ребру многогранника к новой вершине, то есть произвести замену базиса.

• Предположим, что у нас есть базис $\mathcal{B} \colon \lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$

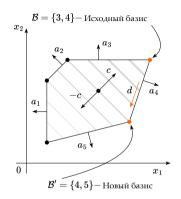


- Предположим, что у нас есть базис \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Предположим, что $\lambda_{\mathcal{B}}^k > 0$. Мы хотим удалить k из базиса и сформировать новый:



- Предположим, что у нас есть базис \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Предположим, что $\lambda_{\mathcal{B}}^k > 0$. Мы хотим удалить k из базиса и сформировать новый:

$$\begin{cases} A_{\mathcal{B}\backslash\{k\}}d = 0\\ a_k^Td = -1 \end{cases}$$

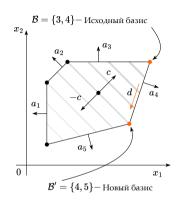


Предположим, что некоторые из коэффициентов $\lambda_{\mathcal{B}}$ положительны. В этом случае необходимо осуществить переход по ребру многогранника к новой вершине, то есть произвести замену базиса.

Симплекс-метод

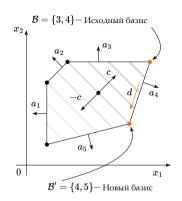
- Предположим, что у нас есть базис \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Предположим, что $\lambda_{\mathcal{B}}^k > 0$. Мы хотим удалить k из базиса и сформировать новый:

$$\begin{cases} A_{\mathcal{B}\backslash\{k\}}d = 0 \\ a_k^T d = -1 \end{cases} c^T$$



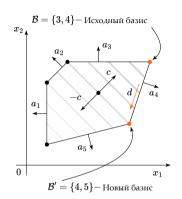
- Предположим, что у нас есть базис \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Предположим, что $\lambda_B^k > 0$. Мы хотим удалить k из базиса и сформировать новый:

$$\begin{cases} A_{\mathcal{B}\backslash\{k\}}d = 0 \\ a_k^T d = -1 \end{cases} \qquad c^T d = \lambda_{\mathcal{B}}^T A_{\mathcal{B}} d$$



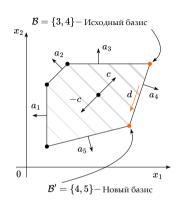
- Предположим, что у нас есть базис \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Предположим, что $\lambda_{\mathcal{B}}^k > 0$. Мы хотим удалить k из базиса и сформировать новый:

$$\begin{cases} A_{\mathcal{B}\backslash\{k\}}d = 0 \\ a_k^Td = -1 \end{cases} \qquad c^Td = \lambda_{\mathcal{B}}^TA_{\mathcal{B}}d = \sum_{i=1}^n \lambda_{\mathcal{B}}^i(A_{\mathcal{B}}d)^i$$



- Предположим, что у нас есть базис \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Предположим, что $\lambda_{\mathcal{B}}^k > 0$. Мы хотим удалить k из базиса и сформировать новый:

$$\begin{cases} A_{\mathcal{B}\backslash \{k\}}d = 0 \\ a_k^Td = -1 \end{cases} \qquad c^Td = \lambda_{\mathcal{B}}^TA_{\mathcal{B}}d = \sum_{i=1}^n \lambda_{\mathcal{B}}^i(A_{\mathcal{B}}d)^i = -\lambda_{\mathcal{B}}^k < 0$$



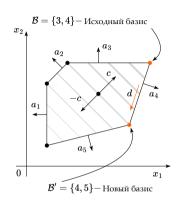
Предположим, что некоторые из коэффициентов $\lambda_{\mathcal{B}}$ положительны. В этом случае необходимо осуществить переход по ребру многогранника к новой вершине, то есть произвести замену базиса.

- Предположим, что у нас есть базис $\mathcal{B}\colon \lambda^T_{\mathcal{B}} = c^T A_{\mathcal{B}}^{-1}$
- Предположим, что $\lambda_{\mathcal{B}}^k>0.$ Мы хотим удалить k из базиса и сформировать новый:

$$\begin{cases} A_{\mathcal{B}\backslash\{k\}}d = 0 \\ a_k^Td = -1 \end{cases} \qquad c^Td = \lambda_{\mathcal{B}}^TA_{\mathcal{B}}d = \sum_{i=1}^n \lambda_{\mathcal{B}}^i(A_{\mathcal{B}}d)^i = -\lambda_{\mathcal{B}}^k < 0$$

• Для всех $j \notin \mathcal{B}$ рассчитаем размер шага проекции:

$$\mu_j = \frac{b_j - a_j^T x_{\mathcal{B}}}{a_j^T d}$$



Предположим, что некоторые из коэффициентов $\lambda_{\mathcal{B}}$ положительны. В этом случае необходимо осуществить переход по ребру многогранника к новой вершине, то есть произвести замену базиса.

- Предположим, что у нас есть базис \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Предположим, что $\lambda_{\mathcal{B}}^k>0.$ Мы хотим удалить k из базиса и сформировать новый:

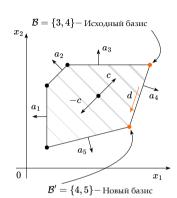
$$\begin{cases} A_{\mathcal{B}\backslash\{k\}}d = 0 \\ a_k^Td = -1 \end{cases} \qquad c^Td = \lambda_{\mathcal{B}}^TA_{\mathcal{B}}d = \sum_{i=1}^n \lambda_{\mathcal{B}}^i(A_{\mathcal{B}}d)^i = -\lambda_{\mathcal{B}}^k < 0$$

• Для всех $j \notin \mathcal{B}$ рассчитаем размер шага проекции:

$$\mu_j = \frac{b_j - a_j^T x_{\mathcal{B}}}{a_j^T d}$$

• Определим новую вершину, которую мы добавим в новый базис:

$$\begin{split} t &= \arg\min_{j} \{\mu_{j} \mid \mu_{j} > 0\} \\ \mathcal{B}' &= \mathcal{B} \backslash \{k\} \cup \{t\} \\ x_{\mathcal{B}'} &= x_{\mathcal{B}} + \mu_{t} d = A_{\mathcal{B}'}^{-1} b_{\mathcal{B}'} \end{split}$$



Предположим, что некоторые из коэффициентов $\lambda_{\mathcal{B}}$ положительны. В этом случае необходимо осуществить переход по ребру многогранника к новой вершине, то есть произвести замену базиса.

- Предположим, что у нас есть базис \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$ • Предположим, что $\lambda_{\mathcal{B}}^k > 0$. Мы хотим удалить k из базиса и сформировать новый:

$$\begin{cases} A_{\mathcal{B}\backslash\{k\}}d = 0\\ a_k^Td = -1 \end{cases}$$

$$\begin{cases} A_{\mathcal{B}\backslash\{k\}}d = 0 \\ a_k^T d = -1 \end{cases} \qquad c^T d = \lambda_{\mathcal{B}}^T A_{\mathcal{B}}d = \sum_{i=1}^n \lambda_{\mathcal{B}}^i (A_{\mathcal{B}}d)^i = -\lambda_{\mathcal{B}}^k < 0$$

• Для всех $j \notin \mathcal{B}$ рассчитаем размер шага проекции:

$$\mu_j = \frac{b_j - a_j^T x_{\mathcal{B}}}{a_j^T d}$$

Определим новую вершину, которую мы добавим в новый базис:

$$t = \arg\min_{j}\{\mu_{j} \mid \mu_{j} > 0\}$$

$$\mathcal{B}' = \mathcal{B} \backslash \{k\} \cup \{t\}$$

$$x_{\mathcal{B}'} = x_{\mathcal{B}} + \mu_{t}d = A_{\mathcal{B}'}^{-1}b_{\mathcal{B}'}$$

• Обратите внимание, что изменение базиса приводит к уменьшению целевой функции: $c^T x_{\mathcal{B}'} = c^T (x_{\mathcal{B}} + \mu_t d) = c^T x_{\mathcal{B}} + \mu_t c^T d$

Нам нужно решить следующую задачу:

$$\min_{x \in \mathbb{R}^n} c^\top x$$
 s.t. $Ax \le b$

Предложенный алгоритм требует начального допустимого базиса.

Симплекс-метод



Нам нужно решить следующую задачу:

$$\min_{x \in \mathbb{R}^n} c^\top x$$
 s.t. $Ax \leq b$

Предложенный алгоритм требует начального допустимого базиса.

Начнём с переформулировки задачи:

(1)
$$\min_{y\in\mathbb{R}^n,z\in\mathbb{R}^n}c^\top(y-z)$$
 s.t. $Ay-Az\leq b$

 $y \ge 0, z \ge 0$



(2)

Нам нужно решить следующую задачу:

Начнём с переформулировки задачи:

$$\min_{x \in \mathbb{R}^n} c^\top x$$

$$\text{s.t. } Ax \le b$$

$$\min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} c^\top (y - z)$$

$$\text{s.t. } Ay - Az \le b$$

$$y \ge 0, z \ge 0$$

Предложенный алгоритм требует начального допустимого базиса.

Зная решение задачи (2), можно восстановить решение задачи (1), и наоборот.

$$x = y - z$$
 \Leftrightarrow $y_i = \max(x_i, 0), \quad z_i = \max(-x_i, 0)$

Теперь мы попытаемся сформулировать новую задачу линейного программирования, решение которой будет допустимой базисной точкой для Задачи 2. Это означает, что мы сначала запускаем симплекс-метод для задачи Phase-1, а затем запускаем задачу Phase-2 с известным начальным решением. Обратите внимание, что допустимое базисное решение для Phase-1 должно быть легко вычислимо.



$$\min_{y\in\mathbb{R}^n,z\in\mathbb{R}^n}c^{\top}(y-z)$$
 s.t. $Ay-Az\leq b$ (Фаза-2 (главная задача ЛП)) $y\geq 0,z\geq 0$





$$\min_{y\in\mathbb{R}^n,z\in\mathbb{R}^n}c^{\top}(y-z)$$
 s.t. $Ay-Az\leq b$ (Фаза-2 (главная задача ЛП)) $y\geq 0,z\geq 0$

$$\min_{\xi\in\mathbb{R}^m,y\in\mathbb{R}^n,z\in\mathbb{R}^n}\sum_{i=1}^m\xi_i$$
 s.t. $Ay-Az\leq b+\xi$
$$y\geq 0,z\geq 0,\xi\geq 0$$
 (Фаза-1)



$$\min_{y\in\mathbb{R}^n,z\in\mathbb{R}^n}c^\top(y-z)$$
 s.t. $Ay-Az\leq b$ (Фаза-2 (главная задача ЛП))
$$y>0,z>0$$

$$\min_{\xi\in\mathbb{R}^m,y\in\mathbb{R}^n,z\in\mathbb{R}^n}\sum_{i=1}^m\xi_i$$
 s.t. $Ay-Az\leq b+\xi$ $y\geq 0,z\geq 0,\xi\geq 0$

• Если Фаза-2 (главная задача ЛП) имеет допустимое решение, то оптимум Фаза-1 равен нулю (т.е. все переменные ξ_i равны нулю). Доказательство: тривиальная проверка.



$$\min_{y\in\mathbb{R}^n,z\in\mathbb{R}^n}c^{\top}(y-z)$$
 s.t. $Ay-Az\leq b$ (Фаза-2 (главная задача ЛП)) $y>0,z>0$

$$\min_{\substack{\xi\in\mathbb{R}^m,y\in\mathbb{R}^n,z\in\mathbb{R}^n\\ \text{s.t. }}}\sum_{i=1}^m\xi_i$$
 s.t. $Ay-Az\leq b+\xi$
$$y\geq 0,z\geq 0,\xi\geq 0$$

- Если Фаза-2 (главная задача ЛП) имеет допустимое решение, то оптимум Фаза-1 равен нулю (т.е. все переменные ξ_i равны нулю). Доказательство: тривиальная проверка.
- Если оптимум Фаза-1 равен нулю (т.е. все переменные ξ_i равны нулю), то мы получаем допустимый базис для Фаза-2.

Доказательство: тривиальная проверка.

$$\min_{y\in\mathbb{R}^n,z\in\mathbb{R}^n}c^{\top}(y-z)$$
 s.t. $Ay-Az\leq b$ (Фаза-2 (главная задача ЛП)) $y>0,z>0$

$$\min_{\xi\in\mathbb{R}^m,y\in\mathbb{R}^n,z\in\mathbb{R}^n}\sum_{i=1}^m\xi_i$$
 s.t. $Ay-Az\leq b+\xi$
$$y\geq 0,z\geq 0,\xi\geq 0$$
 (Фаза-1)

• Если Фаза-2 (главная задача ЛП) имеет допустимое решение, то оптимум Фаза-1 равен нулю (т.е. все переменные ξ_i равны нулю).

Доказательство: тривиальная проверка.

• Если оптимум Фаза-1 равен нулю (т.е. все переменные ξ_i равны нулю), то мы получаем допустимый базис для Фаза-2.

Доказательство: тривиальная проверка.

$$\min_{y\in\mathbb{R}^n,z\in\mathbb{R}^n}c^{ op}(y-z)$$
 s.t. $Ay-Az\leq b$ (Фаза-2 (главная задача ЛП)) $y\geq 0,z\geq 0$

$$\min_{\xi\in\mathbb{R}^m,y\in\mathbb{R}^n,z\in\mathbb{R}^n}\sum_{i=1}^m\xi_i$$
 s.t. $Ay-Az\leq b+\xi$ $y\geq 0,z\geq 0,\xi\geq 0$

• Если Фаза-2 (главная задача ЛП) имеет допустимое решение, то оптимум Фаза-1 равен нулю (т.е. все переменные ξ_i равны нулю). Доказательство: тривиальная проверка.

• Если оптимум Фаза-1 равен нулю (т.е. все переменные ξ_i равны нулю), то мы получаем допустимый базис для Фаза-2.

Доказательство: тривиальная проверка.

• Теперь мы знаем, что если мы можем решить задачу Фаза-1, то мы либо найдём начальную точку для симплекс-метода в исходном методе (если переменные ξ_i равны нулю), либо проверим, что исходная задача не имеет допустимого решения (если переменные ξ_i не равны нулю).



$$\min_{y\in\mathbb{R}^n,z\in\mathbb{R}^n}c^{\top}(y-z)$$
 s.t. $Ay-Az\leq b$ (Фаза-2 (главная задача ЛП))

$$\min_{\xi\in\mathbb{R}^m,y\in\mathbb{R}^n,z\in\mathbb{R}^n}\sum_{i=1}^m\xi_i$$
 s.t. $Ay-Az\leq b+\xi$
$$y>0,z>0,\xi>0$$

- Если Фаза-2 (главная задача ЛП) имеет допустимое решение, то оптимум Фаза-1 равен нулю (т.е. все переменные ξ_i равны нулю). Доказательство: тривиальная проверка.
- Если оптимум Фаза-1 равен нулю (т.е. все переменные ξ_i равны нулю), то мы получаем допустимый базис для Фаза-2.

Доказательство: тривиальная проверка.

- Теперь мы знаем, что если мы можем решить задачу Фаза-1, то мы либо найдём начальную точку для симплекс-метода в исходном методе (если переменные ξ_i равны нулю), либо проверим, что исходная задача не имеет допустимого решения (если переменные ξ_i не равны нулю).
- Но как решить задачу Фаза-1? Она имеет допустимое базисное решение (задача имеет 2n+mпеременных, и точка ниже гарантирует, что 2n+m неравенств удовлетворяются как равенства (активны).)



y > 0, z > 0

$$\min_{y\in\mathbb{R}^n,z\in\mathbb{R}^n}c^{\top}(y-z)$$
 s.t. $Ay-Az\leq b$ (Фаза-2 (главная задача ЛП))

$$\min_{\xi\in\mathbb{R}^m,y\in\mathbb{R}^n,z\in\mathbb{R}^n}\sum_{i=1}^m\xi_i$$
 s.t. $Ay-Az\leq b+\xi$
$$y>0,z>0,\xi>0$$

- Если Фаза-2 (главная задача ЛП) имеет допустимое решение, то оптимум Фаза-1 равен нулю (т.е. все переменные ξ_i равны нулю). Доказательство: тривиальная проверка.
- Если оптимум Фаза-1 равен нулю (т.е. все переменные ξ_i равны нулю), то мы получаем допустимый базис для Фаза-2.

Доказательство: тривиальная проверка.

- Теперь мы знаем, что если мы можем решить задачу Фаза-1, то мы либо найдём начальную точку для симплекс-метода в исходном методе (если переменные ξ_i равны нулю), либо проверим, что исходная задача не имеет допустимого решения (если переменные ξ_i не равны нулю).
- Но как решить задачу Фаза-1? Она имеет допустимое базисное решение (задача имеет 2n+mпеременных, и точка ниже гарантирует, что 2n+m неравенств удовлетворяются как равенства (активны).)



y > 0, z > 0

$$\min_{y\in\mathbb{R}^n,z\in\mathbb{R}^n}c^{\top}(y-z)$$
 s.t. $Ay-Az\leq b$ (Фаза-2 (главная задача ЛП))

$$\min_{\xi\in\mathbb{R}^m,y\in\mathbb{R}^n,z\in\mathbb{R}^n}\sum_{i=1}^m\xi_i$$
 s.t. $Ay-Az\leq b+\xi$
$$y>0,z>0,\xi>0$$

- Если Фаза-2 (главная задача ЛП) имеет допустимое решение, то оптимум Фаза-1 равен нулю (т.е. все переменные ξ_i равны нулю). Доказательство: тривиальная проверка.
- Если оптимум Фаза-1 равен нулю (т.е. все переменные ξ_i равны нулю), то мы получаем допустимый базис для Фаза-2.

Доказательство: тривиальная проверка.

- Теперь мы знаем, что если мы можем решить задачу Фаза-1, то мы либо найдём начальную точку для симплекс-метода в исходном методе (если переменные ξ_i равны нулю), либо проверим, что исходная задача не имеет допустимого решения (если переменные ξ_i не равны нулю).
- Но как решить задачу Фаза-1? Она имеет допустимое базисное решение (задача имеет 2n+mпеременных, и точка ниже гарантирует, что 2n+m неравенств удовлетворяются как равенства (активны).)

$$z=0 \quad y=0 \quad \xi_i=\max(0,-b_i)$$



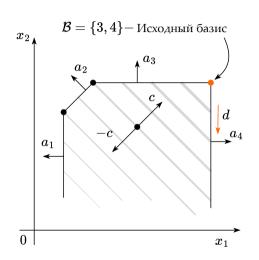
y > 0, z > 0

Сходимость симплекс-метода



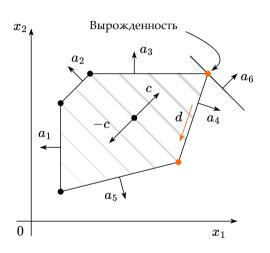


Неограниченное бюджетное множество



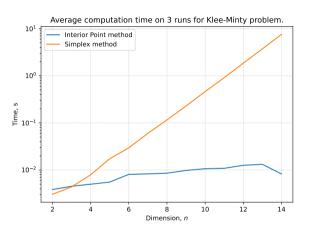
В этом случае не найдётся ни одного положительного $\mu_j.$

Вырожденность вершин



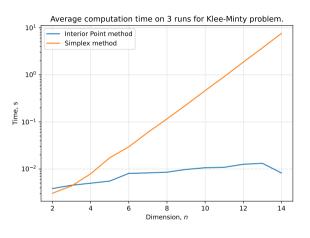
Случаи вырожденности требуют особого рассмотрения. В отсутствие вырожденности на каждой итерации гарантируется монотонное убывание значения целевой функции.





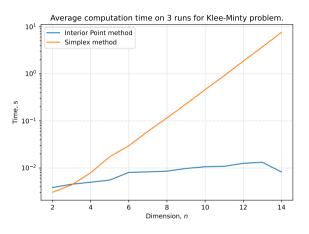
 Много прикладных задач может быть сформулировано в виде задач линейного программирования.





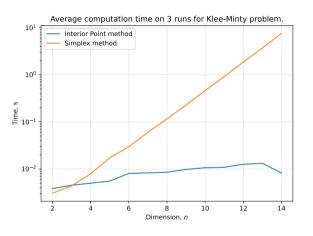
- Много прикладных задач может быть сформулировано в виде задач линейного программирования.
- Симплекс-метод прост в своей основе, но в худшем случае может работать экспоненциально долго.





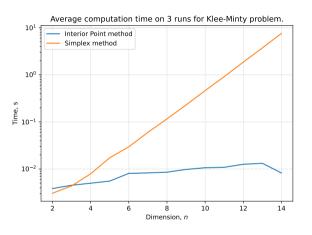
- Много прикладных задач может быть сформулировано в виде задач линейного программирования.
- Симплекс-метод прост в своей основе, но в худшем случае может работать экспоненциально долго.
- Метод эллипсоидов Хачияна (1979) стал первым алгоритмом с доказанной полиномиальной сложностью для задач ЛП. Однако он обычно работает медленнее, чем симплекс-метод в реальных небольших задачах.





- Много прикладных задач может быть сформулировано в виде задач линейного программирования.
- Симплекс-метод прост в своей основе, но в худшем случае может работать экспоненциально долго.
- Метод эллипсоидов Хачияна (1979) стал первым алгоритмом с доказанной полиномиальной сложностью для задач ЛП. Однако он обычно работает медленнее, чем симплекс-метод в реальных небольших задачах.
- Основной прорыв метод Кармаркара (1984) для решения задач ЛП с использованием метода внутренней точки.





- Много прикладных задач может быть сформулировано в виде задач линейного программирования.
- Симплекс-метод прост в своей основе, но в худшем случае может работать экспоненциально долго.
- Метод эллипсоидов Хачияна (1979) стал первым алгоритмом с доказанной полиномиальной сложностью для задач ЛП. Однако он обычно работает медленнее, чем симплекс-метод в реальных небольших задачах.
- Основной прорыв метод Кармаркара (1984) для решения задач ЛП с использованием метода внутренней точки.
- Методы внутренней точки являются последним словом в этой области. Тем не менее, для типовых задач ЛП качественные реализации симплекс-метода и методов внутренней точки показывают схожую производительность.

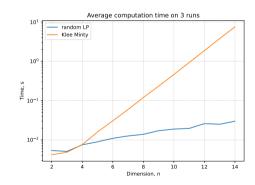


Пример Klee Minty

Так как число вершин конечно, сходимость алгоритма гарантирована (за исключением вырожденных случаев, которые здесь не рассматриваются). Тем не менее, сходимость может быть экспоненциально медленной из-за потенциально большого числа вершин. Существует пример, в котором симплекс-метод вынужден пройти через все вершины многогранника.

В следующей задаче симплекс-метод должен проверить 2^n-1 вершин с $x_0=0$.

$$\begin{aligned} \max_{x \in \mathbb{R}^n} 2^{n-1}x_1 + 2^{n-2}x_2 + \dots + 2x_{n-1} + x_n \\ \text{s.t.} \ \ x_1 &\leq 5 \\ 4x_1 + x_2 &\leq 25 \\ 8x_1 + 4x_2 + x_3 &\leq 125 \\ \dots \\ 2^nx_1 + 2^{n-1}x_2 + 2^{n-2}x_3 + \dots + x_n &\leq 5^n \\ x &> 0 \end{aligned}$$





Смешанное целочисленное программирование (МІР)





Рассмотрим следующую задачу смешанного целочисленного программирования (MIP):

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \rightarrow \max_{x_1, x_2, x_3, x_4}$$
 s.t. $5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$ (3)
$$x_i \in \{0, 1\} \quad \forall i$$



Рассмотрим следующую задачу смешанного целочисленного программирования (MIP):

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$
 s.t. $5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$
$$x_i \in \{0, 1\} \quad \forall i$$

Упростим её до:

(3)

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$
 s.t. $5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$ (4)

$$x_i \in [0,1] \quad \forall i$$

Рассмотрим следующую задачу смешанного целочисленного программирования (МІР):

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$

s.t.
$$5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$$

 $x_i \in \{0,1\} \quad \forall i$ Оптимальное решение

 $x_1 = 0, x_2 = x_3 = x_4 = 1, \text{ if } z = 21.$

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$

s.t.
$$5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$$

 $x_i \in [0, 1] \quad \forall i$





(4)

Рассмотрим следующую задачу смешанного целочисленного программирования (МІР):

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$

s.t.
$$5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$$

$$x_i \in \{0, 1\} \quad \forall i$$

 $x_i \in \{0,1\} \quad \forall i$ Оптимальное решение

$$x_1=0, x_2=x_3=x_4=1, \ \text{if} \ z=21.$$

Упростим её до:

(3)

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$

s.t.
$$5x_1+7x_2+4x_3+3x_4\leq 14$$

$$x_i\in [0,1] \quad \forall i$$

Оптимальное решение

$$x_1=x_2=1, x_3=0.5, x_4=0, \ \text{if} \ z=22.$$

(4)

Рассмотрим следующую задачу смешанного целочисленного программирования (MIP):

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$

s.t.
$$5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$$

 $x_i \in \{0,1\} \quad \forall i$ Оптимальное решение

$$x_1=0, x_2=x_3=x_4=1, \ {\rm if} \ z=21.$$

Упростим её до:

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$
 s.t. $5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$ (4)

(3)

Оптимальное решение

$$x_1=x_2=1, x_3=0.5, x_4=0, \ {\rm id} \ z=22.$$

 $x_i \in [0,1] \quad \forall i$

• Округление $x_3 = 0$: даёт z = 19.

Рассмотрим следующую задачу смешанного целочисленного программирования (МІР):

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$
 s.t. $5x_1 + 7x_2 + 4x_3 + 3x_4 < 14$

$$x_i \in \{0,1\} \quad \forall i$$
льное решение

 $x_i \in \{0,1\} \quad \forall i$ Оптимальное решение

$$x_1=0, x_2=x_3=x_4=1, \ {\rm if} \ z=21.$$

Упростим её до:

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$
 s.t. $5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$ (4)
$$x_i \in [0, 1] \quad \forall i$$

(3)

Оптимальное решение

$$x_1=x_2=1, x_3=0.5, x_4=0, \ \text{if} \ z=22.$$

- Округление $x_3 = 0$: даёт z = 19.
- Округление $x_3 = 1$: недопустимо.

Рассмотрим следующую задачу смешанного целочисленного программирования (МІР):

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$
 s.t. $5x_1 + 7x_2 + 4x_3 + 3x_4 < 14$

$$x_i \in \{0,1\} \quad \forall i$$
льное решение

 $x_i \in \{0,1\} \quad \forall i$ Оптимальное решение

$$x_1=0, x_2=x_3=x_4=1, \ {\rm if} \ z=21.$$

Упростим её до:

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$
 s.t. $5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$ (4)
$$x_i \in [0, 1] \quad \forall i$$

(3)

Оптимальное решение

$$x_1=x_2=1, x_3=0.5, x_4=0, \ \text{if} \ z=22.$$

- Округление $x_3 = 0$: даёт z = 19.
- Округление $x_3 = 1$: недопустимо.

Рассмотрим следующую задачу смешанного целочисленного программирования (MIP):

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$

s.t.
$$5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$$

 $x_i \in \{0,1\} \quad \forall i$ Оптимальное решение

$$x_1=0, x_2=x_3=x_4=1, \ {\rm if} \ z=21.$$

Упростим её до:

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$
 s.t. $5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$ (4)

 $x_i \in [0,1] \quad \forall i$

 $x_1 = x_2 = 1, x_2 = 0.5, x_4 = 0, \text{ if } z = 22.$

Оптимальное решение

альное решение

- Округление $x_3 = 0$: даёт z = 19.
- Округление $x_3 = 1$: недопустимо.

- MIP намного сложнее, чем ЛП
 - Наивное округление решения, полученного для ЛП-релаксации исходной задачи МІР, может привести к недопустимому или неоптимальному решению.

(3)



Рассмотрим следующую задачу смешанного целочисленного программирования (MIP):

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$

s.t.
$$5x_1+7x_2+4x_3+3x_4\leq 14$$
 Оптимальное решение
$$x_i\in\{0,1\}\quad\forall i$$

·

$$x_1=0, x_2=x_3=x_4=1, \ {\rm in} \ z=21.$$

Упростим её до:

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$
 s.t. $5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$ (4)

 $x_i \in [0,1] \quad \forall i$

Оптимальное решение

$$x_1=x_2=1, x_3=0.5, x_4=0, \ \text{if} \ z=22.$$

- ullet Округление $x_3=0$: даёт z=19.
- Округление $x_3 = 1$: недопустимо.

- MIP намного сложнее, чем ЛП
 - Наивное округление решения, полученного для ЛП-релаксации исходной задачи MIP, может привести к недопустимому или неоптимальному решению.

(3)

Общая задача МІР является NP-трудной задачей.



Рассмотрим следующую задачу смешанного целочисленного программирования (MIP):

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$

s.t. $5x_1+7x_2+4x_3+3x_4\leq 14$ Оптимальное решение $x_i\in\{0,1\}\quad\forall i$

$$x_1 = 0, x_2 = x_3 = x_4 = 1, \text{ if } z = 21.$$

Общая задача МІР является NP-трудной задачей.

Упростим её до:

$$z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$$

s.t.
$$5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$$

 $x_i \in [0, 1] \quad \forall i$

Оптимальное решение

$$x_1 = x_2 = 1, x_3 = 0.5, x_4 = 0, \text{ if } z = 22.$$

- Округление $x_3 = 0$: даёт z = 19.
- Округление $x_3 = 1$: недопустимо.

- MIP намного сложнее, чем ЛП
 - Наивное округление решения, полученного для ЛП-релаксации исходной задачи МІР, может привести к недопустимому или неоптимальному решению.

(3)

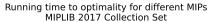
• Однако, если матрица коэффициентов МІР является полностью унимодулярной матрицей, то она может быть решена за полиномиальное время.

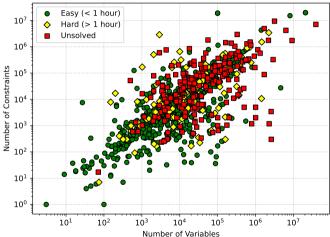


(4)

Непредсказуемая сложность МІР

 Трудно предсказать, что будет решено быстро, а что потребует много времени



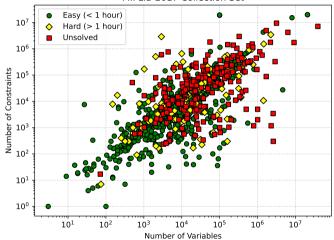




Непредсказуемая сложность МІР

- Трудно предсказать, что будет решено быстро, а что потребует много времени
- *•* Датасет

Running time to optimality for different MIPs MIPLIB 2017 Collection Set

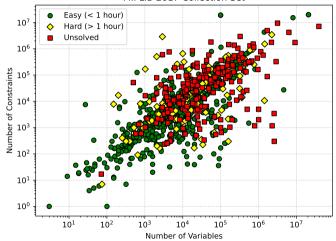




Непредсказуемая сложность МІР

- Трудно предсказать, что будет решено быстро, а что потребует много времени
- *•*Датасет
- 🗣Код

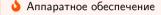
Running time to optimality for different MIPs MIPLIB 2017 Collection Set





Прогресс аппаратного vs программного обеспечения

Что бы вы выбрали, если предположить, что вопрос поставлен корректно (вы можете скомпилировать ПО для любого оборудования, и задача в обоих случаях одна и та же)? Мы рассмотрим период с 1992 по 2023 год.



Решение MIP с использованием старого ПО на современном оборудовании



Программное обеспечение

Решение MIP с использованием современного ПО на старом оборудовании



Прогресс аппаратного vs программного обеспечения

Что бы вы выбрали, если предположить, что вопрос поставлен корректно (вы можете скомпилировать ПО для любого оборудования, и задача в обоих случаях одна и та же)? Мы рассмотрим период с 1992 по 2023 год.

Аппаратное обеспечение

Решение MIP с использованием старого ПО на современном оборудовании

 $\approx 1.664.510$ х ускорение

Закон Мура утверждает, что вычислительная мощность удваивается каждые 18 месяцев.

Программное обеспечение

Решение МІР с использованием современного ПО на старом оборудовании

 $\approx 2.349.000 \times$ ускорение

Р. Бикси провёл масштабный эксперимент по сравнению производительности всех версий CPLEX с 1992 по 2007 год и измерил общий прогресс ΠO (29000) раз), позже (в 2009 году) он стал одним из основателей Gurobi Optimization, которое дало дополнительное ≈ 81 ускорение на МІР.

Прогресс аппаратного vs программного обеспечения

Что бы вы выбрали, если предположить, что вопрос поставлен корректно (вы можете скомпилировать ПО для любого оборудования, и задача в обоих случаях одна и та же)? Мы рассмотрим период с 1992 по 2023 год.



Решение MIP с использованием старого ПО на современном оборудовании

$$pprox 1.664.510$$
 х ускорение

Закон Мура утверждает, что вычислительная мощность удваивается каждые 18 месяцев.



на старом оборудовании

$$pprox 2.349.000$$
 х ускорение

Решение МІР с использованием современного ПО

Р. Бикси провёл масштабный эксперимент по сравнению производительности всех версий CPLEX с 1992 по 2007 год и измерил общий прогресс ΠO (29000) раз), позже (в 2009 году) он стал одним из основателей Gurobi Optimization, которое дало дополнительное ≈ 81 ускорение на МІР.

Оказывается, что если вам нужно решить МІР, лучше использовать старый компьютер и современные методы, чем наоборот, самый новый компьютер и методы начала 1990-х годов! 2



Источники

• Теория оптимизации (МАТН4230) курс @ СИНК, профессор Тейюн Цень

