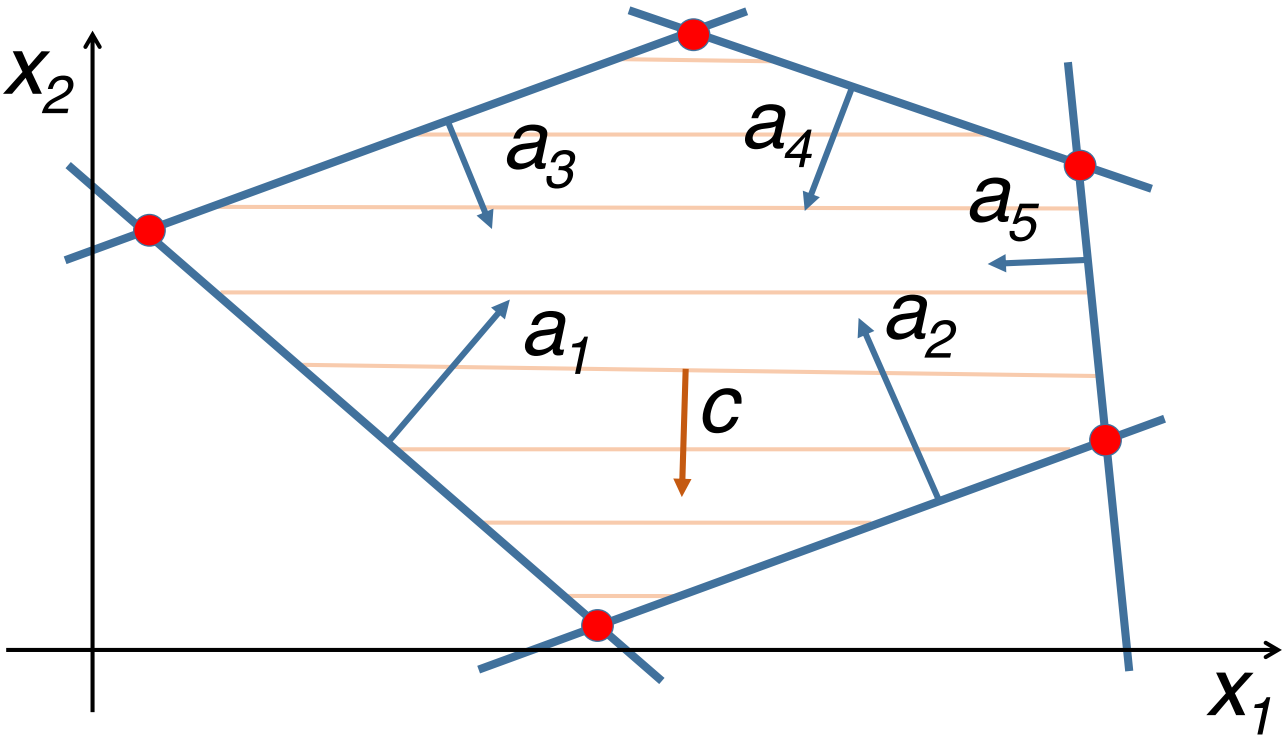
What is LP

Generally speaking, all problems with linear objective and linear equalities\inequalities constraints could be considered as Linear Programming. However, there are some widely accepted formulations.

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



for some vectors $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and matrix $A \in \mathbb{R}^{m \times n}$. Where the inequalities are interpreted component-wise.

Standard form

This form seems to be the most intuitive and geometric in terms of visualization. Let us have vectors $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and matrix $A \in \mathbb{R}^{m \times n}$.

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

Canonical form

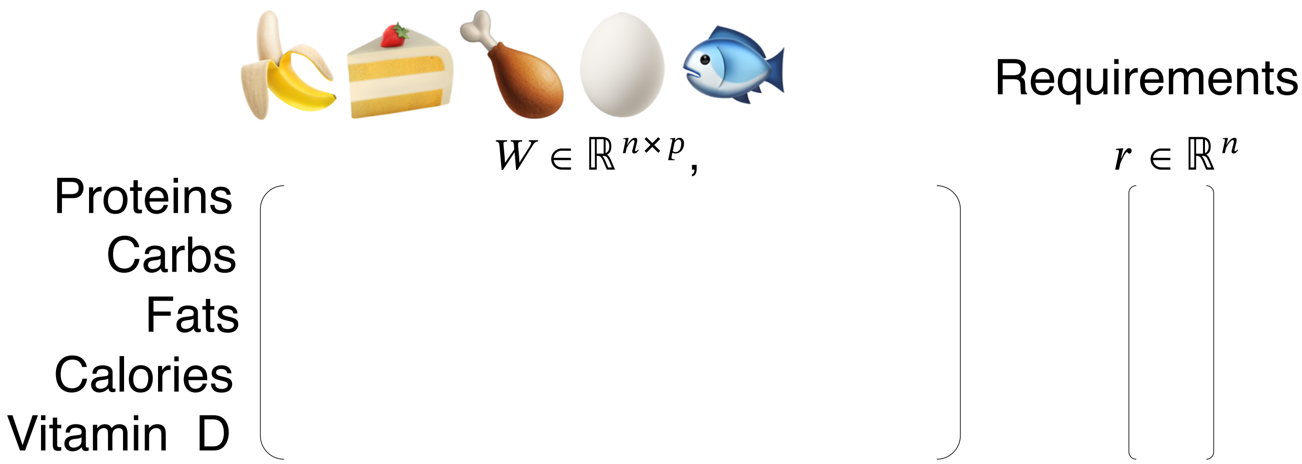
$$
\begin{aligned}
&\min_{x \in \mathbb{R}^n} c^\top x \\
\text{s.t. } & Ax \leq b \\
& x_i \geq 0, \; i = 1, \dots, n
\end{aligned}
\tag{LP.Canonical}
$$

Real world problems

Diet problem

Imagine, that you have to construct a diet plan from some set of products: 🍌🍰🍗🥚🐟. Each of the products has its own vector of nutrients. Thus, all the food information could be processed through the matrix W. Let also assume, that we have the vector of requirements for each of nutrients $r \in \mathbb{R}^n$. We need to find the cheapest configuration of the diet, which meets all the requirements:

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



$c \in \mathbb{R}^p$ - cost per 100 g

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

$$
Wx \geq r
$$

How to retrieve LP

Basic transformations

Inequality to equality by increasing the dimension of the problem by m.

$$
Ax \leq b \leftrightarrow \begin{cases} Ax + z = b \\ z \geq 0 \end{cases}
$$

unsigned variables to nonnegative variables.

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

Chebyshev approximation problem

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

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

l_1 approximation problem

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

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

Idea of simplex algorithm

- The Simplex Algorithm walks along the edges of the polytope, at every corner choosing the edge that decreases $c^\top x$ most
- This either terminates at a corner, or leads to an unconstrained edge ($-\infty$ optimum)

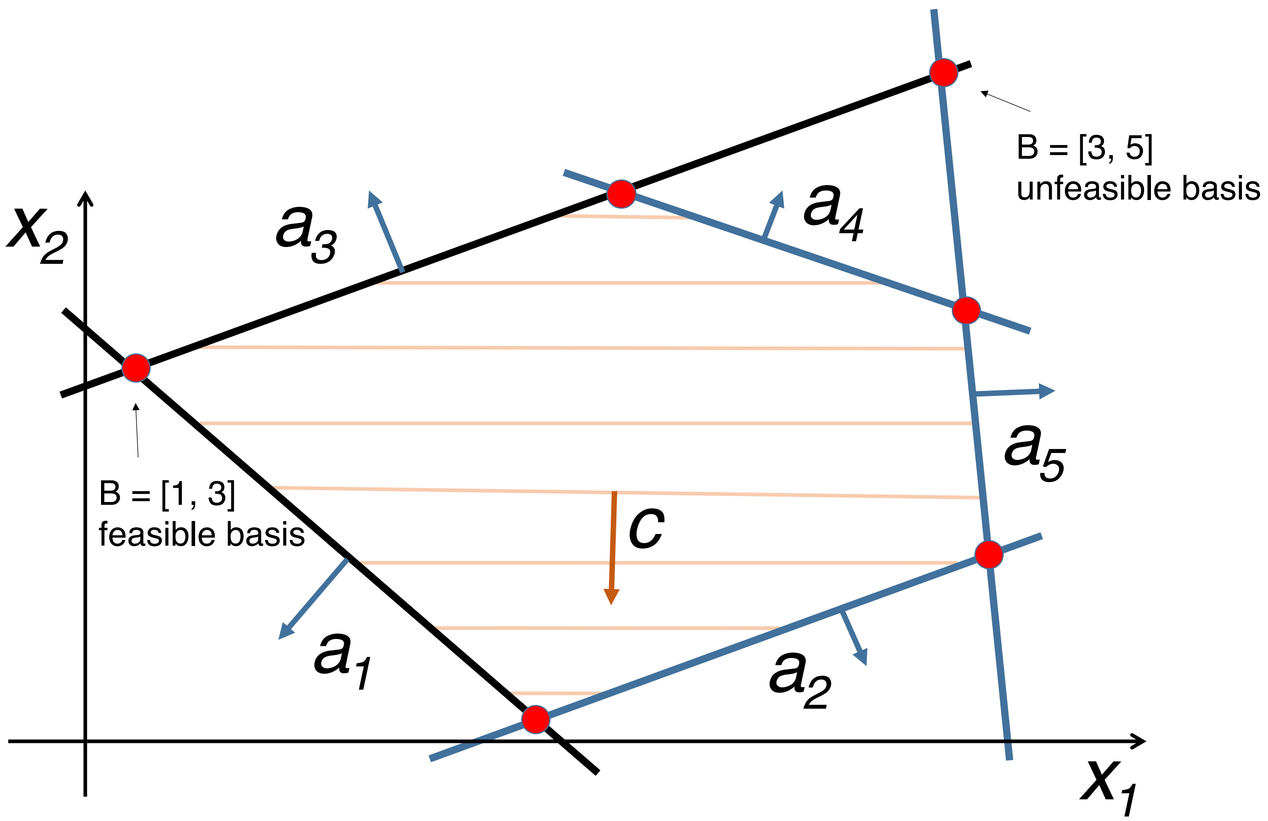
We will illustrate simplex algorithm for the simple inequality form of LP:

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

Definition: a **basis** B is a subset of n (integer) numbers between 1 and m, so that $\text{rank} A_B = n$. Note, that we can associate submatrix A_B and corresponding right-hand side b_B with the basis B. Also, we can derive a point of intersection of all these hyperplanes from basis: $x_B = A_B^{-1} b_B$.

If $A x_B \leq b$, then basis B is **feasible**.

A basis B is optimal if x_B is an optimum of the LP.Inequality.



Since we have a basis, we can decompose our objective vector c in this basis and find the scalar coefficients λ_B:

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

Main lemma

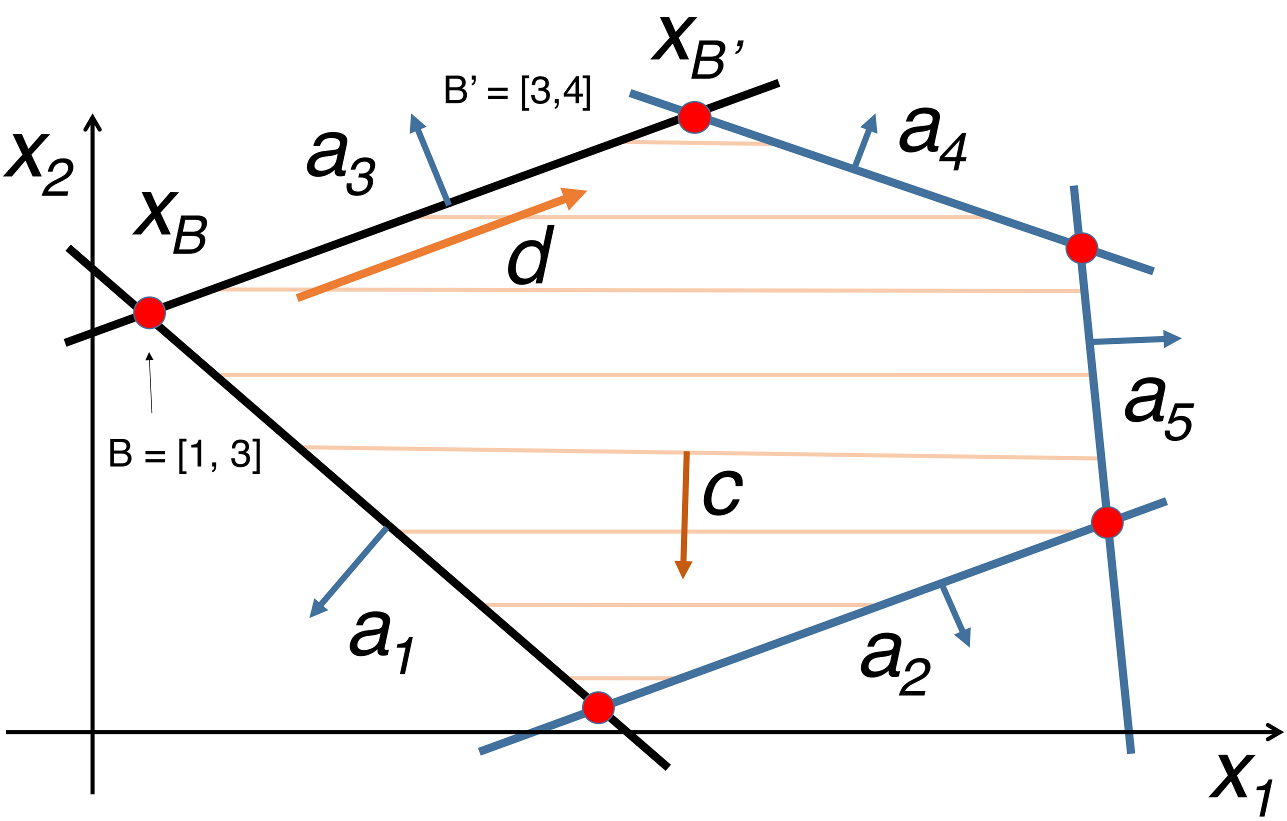
If all components of λ_B are non-positive and B is feasible, then B is optimal.

Proof:

$$
\begin{aligned}
\exists x^* : Ax^* \leq b, c^\top x^* < c^\top x_B \\
A_B x^* \leq b_B \\
\lambda_B^\top A_B x^* \geq \lambda_B^\top b_B \\
c^\top x^* \geq \lambda_B^\top A_B x_B \\
c^\top x^* \geq c^\top x_B
\end{aligned}
$$

Changing basis

Suppose, some of the coefficients of λ_B are positive. Then we need to go through the edge of the polytope to the new vertex (i.e., switch the basis)



$$
x_{B'} = x_B + \mu d = A_{B'}^{-1} b_{B'}
$$

Finding an initial basic feasible solution

Let us consider $\mathrm{LP.Canonical}$.

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

The proposed algorithm requires an initial basic feasible solution and corresponding basis. To compute this solution and basis, we start by multiplying by -1 any row i of $Ax = b$ such that $b_i < 0$. This ensures that $b \geq 0$. We then introduce artificial variables $z \in \mathbb{R}^m$ and consider the following LP:

$$
\begin{aligned}
&\min_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} \quad 1^\top z \\
\text{s.t. } & Ax + Iz = b \\
& x_i, z_j \geq 0, \;\; i = 1, \dots, n \; j = 1, \dots, m
\end{aligned}
\tag{LP.Phase 1}
$$

which can be written in canonical form $\min\{\tilde{c}^\top \tilde{x} \mid \tilde{A}\tilde{x} = \tilde{b}, \tilde{x} \geq 0\}$ by setting

$$
\tilde{x} = \begin{bmatrix} x \\ z \end{bmatrix}, \quad \tilde{A} = [A \; I], \quad \tilde{b} = b, \quad \tilde{c} = \begin{bmatrix} 0_n \\ 1_m \end{bmatrix}
$$

An initial basis for LP.Phase 1 is $\tilde{A}_B = I, \tilde{A}_N = A$ with corresponding basic feasible solution $\tilde{x}_N = 0, \tilde{x}_B = \tilde{A}_B^{-1}\tilde{b} = \tilde{b} \geq 0$. We can therefore run the simplex method on LP.Phase 1, which will converge to an optimum $\tilde{x}^*$. $\tilde{x} = (\tilde{x}_N \; \tilde{x}_B)$. There are several possible outcomes:

- $$\tilde{c}^\top \tilde{x} > 0$$

  . Original primal is infeasible.

- $$\tilde{c}^\top \tilde{x} = 0 \to 1^\top z^* = 0$$

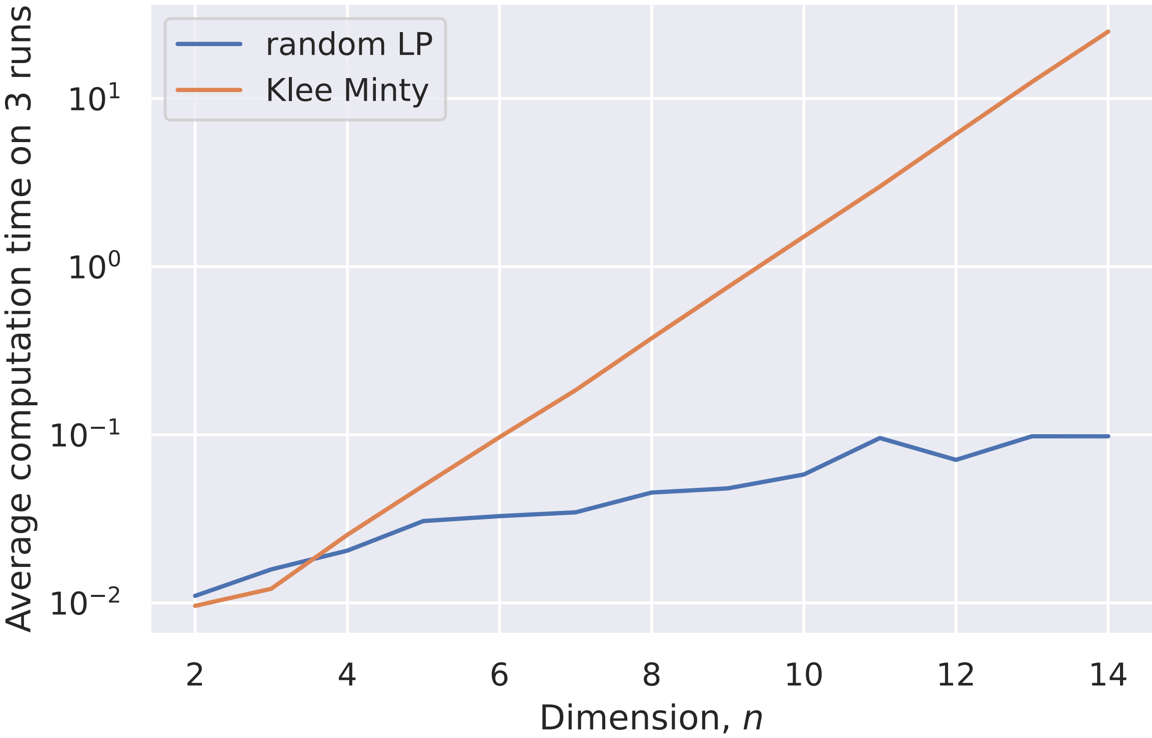
  . The obtained solution is a start point for the original problem (probably with slight modification).

Convergence

Klee Minty example

In the following problem simplex algorithm needs to check $2^n - 1$ vertexes with $x_0 = 0$.

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



Strong duality

There are four possibilities:

- Both the primal and the dual are infeasible.
- The primal is infeasible and the dual is unbounded.
- The primal is unbounded and the dual is infeasible.
- Both the primal and the dual are feasible and their optimal values are equal.

Summary

- A wide variety of applications could be formulated as the linear programming.
- Simplex algorithm is simple, but could work exponentially long.

- Khachiyan's ellipsoid method is the first to be proved running at polynomial complexity for LPs. However, it is usually slower than simplex in real problems.
- Interior point methods are the last word in this area. However, good implementations of simplex-based methods and interior point methods are similar for routine applications of linear programming.

Code

Open in Colab

Materials

- Linear Programming. in V. Lempitsky optimization course.
- Simplex method. in V. Lempitsky optimization course.
- Overview of different LP solvers
- TED talks watching optimization
- Overview of ellipsoid method
- Comprehensive overview of linear programming
- Converting LP to a standard form

Железо vs Софт

В 2007 году Биксби провел впечатляющий эксперимент . Он взял все версии пакета CPLEX, начиная с его первого по- явления в 1991 году, и опробовал их на большом количестве известных практических задач целочисленного линейного программирования . Ученые собрали внушительные коллекции таких задач . Биксби выбрал из них 1892, а затем сравнил скорость их решения, от версии к версии, на одном и том же компьютере .

Оказалось, что за 15 лет скорость решения увеличилась в 29 000 раз! Интересно, что самое большое ускорение, почти десятикратное, произошло в 1998 году, причем не случайно . До этого математики в течение 30 лет разрабатывали новые теории и методы, из которых очень мало было внедрено в практику . В 1998 году в версии CPLEX6 .5 была поставлена задача реализовать по максимуму все эти идеи . В результате наши возможности в линейном программировании вышли на качественно новый уровень .

Процесс продолжается . Gurobi появился в 2009 году и к 2012-му ускорился в 16,2 раза . А общий эффект в 1991– 2012 годах — в 29000×16,2 = 469800 раз! Повторим, что это произошло независимо от скорости компьютера, иными словами, исключительно благодаря развитию математических идей .

Если верить закону Мура, то за 1992–2012 годы компьютеры ускорились примерно в 8000 раз . Сравните с почти полу- миллионным ускорением алгоритмов! Получается, что если вам нужно решить задачу линейного программирования, то лучше использовать старый компьютер и современные методы, чем наоборот, новейший компьютер и методы начала 1990-х .

Литвак Н., Райгородский А. - Кому нужна математика. Понятная книга о том, как устроен цифровой мир - 2017.

Mixed Integer Programming

Mixed Integer Programming