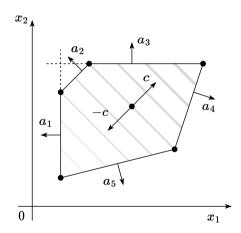






What is Linear Programming?



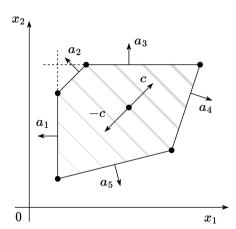
Generally speaking, all problems with linear objective and linear equalities/inequalities constraints could be considered as Linear Programming. However, there are some formulations.

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



What is Linear Programming?



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

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

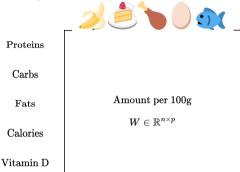
for some vectors $c\in\mathbb{R}^n$, $b\in\mathbb{R}^m$ and matrix $A\in\mathbb{R}^{m\times n}$. Where the inequalities are interpreted component-wise.

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 $A \in \mathbb{R}^{m \times n}$.

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

s.t. Ax = b (LP.Standard) $x_i > 0, i = 1, \dots, n$

Example: Diet problem



$$\min_{c \,\in\, \mathbb{R}^p,\, ext{price per 100g}} c^T x$$

$$x \in \mathbb{R}^n$$
, nutrient requirements $egin{aligned} Wx \succeq r \ x \in \mathbb{R}^p ext{, amount of products, 100g} \end{aligned}$

$$f \to \min_{x,y,z}$$

Example: Diet problem Proteins Carbs Amount per 100g Fats $W \in \mathbb{R}^{n imes p}$ Calories Vitamin D $\min c^T x$

 $c\in\mathbb{R}^p,$ price per 100g $x\in\mathbb{R}^p$ $r\in\mathbb{R}^n,$ nutrient requirements $x\in\mathbb{R}^p$, amount of products, 100g $x\succeq 0$

Imagine, that you have to construct a diet plan from some set of products: bananas, cakes, chicken, eggs, fish. Each of the products has its vector of nutrients. Thus, all the food information could be processed through the matrix W. Let us 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:

$$egin{aligned} \min_{x \in \mathbb{R}^p} c^ op x \ & ext{s.t.} \ Wx \succeq r \ & x_i \geq 0, \ i = 1, \dots, n \end{aligned}$$

♦Open In Colab

Minimization of convex function as LP

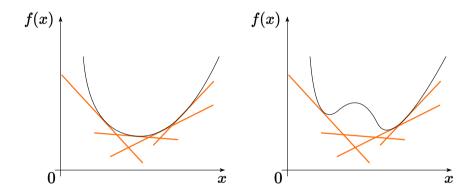


Figure 1: How LP can help with general convex problem

• The function is convex iff it can be represented as a pointwise maximum of linear functions.

⊕ ∩ **ø**

Minimization of convex function as LP

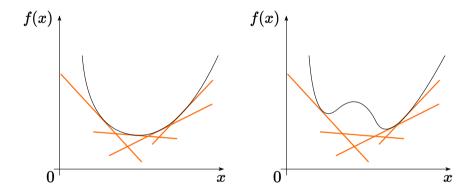


Figure 1: How LP can help with general convex problem

- The function is convex iff it can be represented as a pointwise maximum of linear functions.
- In high dimensions, the approximation may require too many functions.

Minimization of convex function as LP

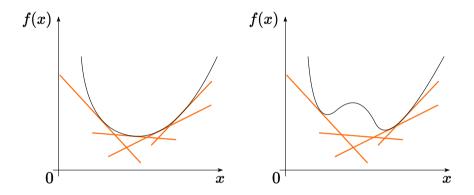


Figure 1: How LP can help with general convex problem

- The function is convex iff it can be represented as a pointwise maximum of linear functions.
- In high dimensions, the approximation may require too many functions.
- More efficient convex optimizers (not reducing to LP) exist.

The prototypical transportation problem deals with the distribution of a commodity from a set of sources to a set of destinations. The object is to minimize total transportation costs while satisfying constraints on the supplies available at each of the sources, and satisfying demand requirements at each of the destinations.



Figure 2: Western Europe Map. Popen In Colab



େ ଚ

Customer / Source	Arnhem [€ /ton]	Gouda [€ /ton]	Demand [tons]
London	n/a	2.5	125
Berlin	2.5	n/a	175
Maastricht	1.6	2.0	225
Amsterdam	1.4	1.0	250
Utrecht	0.8	1.0	225
The Hague	1.4	0.8	200
Supply [tons]	550 tons	700 tons	

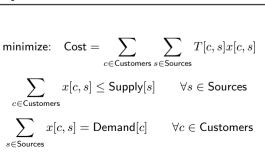
$$\label{eq:minimize:Cost} \begin{aligned} & \text{minimize:} \quad \text{Cost} = \sum_{c \in \text{Customers}} \sum_{s \in \text{Sources}} T[c, s] x[c, s] \end{aligned}$$

Customer / Source	Arnhem [€ /ton]	Gouda [€ /ton]	Demand [tons]
London	n/a	2.5	125
Berlin	2.5	n/a	175
Maastricht	1.6	2.0	225
Amsterdam	1.4	1.0	250
Utrecht	0.8	1.0	225
The Hague	1.4	0.8	200
Supply [tons]	550 tons	700 tons	

$$\label{eq:cost} \text{minimize:} \quad \mathsf{Cost} = \sum_{c \in \mathsf{Customers}} \sum_{s \in \mathsf{Sources}} T[c,s] x[c,s]$$

$$\sum_{c \in \mathsf{Customers}} x[c,s] \leq \mathsf{Supply}[s] \qquad \forall s \in \mathsf{Sources}$$

Customer / Source	Arnhem [€ /ton]	Gouda [€ /ton]	Demand [tons]
London	n/a	2.5	125
Berlin	2.5	n/a	175
Maastricht	1.6	2.0	225
Amsterdam	1.4	1.0	250
Utrecht	0.8	1.0	225
The Hague	1.4	0.8	200
Supply [tons]	550 tons	700 tons	
	<u> </u>	~	1



This can be represented in the following graph:

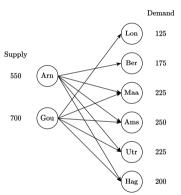


Figure 3: Graph associated with the problem

How to derive LP?





Max-min

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

How to derive LP?



Max-min

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

Equality to inequality

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



Max-min

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

Equality to inequality

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

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

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

Max-min

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

Equality to inequality

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

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

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

Unsigned variables to nonnegative variables.

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

Example: Chebyshev approximation problem

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

Could be equivalently written as an LP with the replacement of the maximum coordinate of a vector:



Example: Chebyshev approximation problem

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

Could be equivalently written as an LP with the replacement of the maximum coordinate of a vector:

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

ℓ_1 approximation problem

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

Could be equivalently written as an LP with the replacement of the sum of coordinates of a vector:

ℓ_1 approximation problem

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

Could be equivalently written as an LP with the replacement of the sum of coordinates of a vector:

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

A manufacturing facility receives an order for 100 liters of a solution with a specific composition (e.g., 4% sugar solution). The facility has on hand:

Component	Sugar (%)	Cost (\$/I)
Concentrate A (Dobry cola) Concentrate B (Sever cola)	10.6 4.5	1.25 1.02
Water (Psyzh) `	0.0	0.62

Goal: Find the minimum-cost blend to meet the order.

Objective Function



A manufacturing facility receives an order for 100 liters of a solution with a specific composition (e.g., 4% sugar solution). The facility has on hand:

Component	Sugar (%)	Cost (\$/I)
Concentrate A (Dobry cola)	10.6	1.25
Concentrate B (Sever cola)	4.5	1.02
Water (Psyzh)	0.0	0.62

Goal: Find the minimum-cost blend to meet the order.

Objective Function

Minimize cost:

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

where x_c is the volume of component c used, and P_c is its price.

A manufacturing facility receives an order for 100 liters of Volume Constraint a solution with a specific composition (e.g., 4% sugar solution). The facility has on hand:

Component	Sugar (%)	Cost (\$/I)
Concentrate A (Dobry cola)	10.6	1.25
Concentrate B (Sever cola)	4.5	1.02
Water (Psyzh)	0.0	0.62

Goal: Find the minimum-cost blend to meet the order.

Objective Function

Minimize cost:

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

where x_c is the volume of component c used, and P_c is its price.



A manufacturing facility receives an order for 100 liters of a solution with a specific composition (e.g., 4% sugar solution). The facility has on hand:

Volume Constraint Ensure total volume V:

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

Component	Sugar (%)	Cost (\$/I)
Concentrate A (Dobry cola)	10.6	1.25
Concentrate B (Sever cola)	4.5	1.02
Water (Psyzh)	0.0	0.62

Composition Constraint

Goal: Find the minimum-cost blend to meet the order.

Objective Function

Minimize cost:

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

where x_c is the volume of component c used, and P_c is its price.

⊕ ი

A manufacturing facility receives an order for 100 liters of a solution with a specific composition (e.g., 4% sugar solution). The facility has on hand:

Component	Sugar (%)	Cost (\$/I)
Concentrate A (Dobry cola)	10.6	1.25
Concentrate B (Sever cola)	4.5	1.02
Water (Psyzh)	0.0	0.62

Volume Constraint
Ensure total volume V:

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

Composition Constraint

Ensure 4% sugar content:

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

Objective Function

Minimize cost:

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

where x_c is the volume of component c used, and P_c is its price.

Sugar (%)

10.6

A manufacturing facility receives an order for 100 liters of a solution with a specific composition (e.g., 4% sugar solution). The facility has on hand:

Water (Psyzh) 0.0 0.62
Concentrate B (Sever cola) 4.5 1.02

Volume Constraint Ensure total volume V:

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

Composition Constraint Ensure 4% sugar content:

Goal: Find the minimum-cost blend to meet the order.

Cost (\$/I)

1.25

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

Linearized version:

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

This can be solved using linear programming. **Source** code

Component

Concentrate A (Dobry cola)

where x_c is the volume of component c used, and P_c is its

price.

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

How to derive LP?

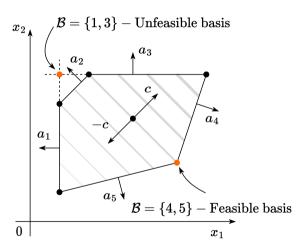
Objective Function

Minimize cost:

Simplex Algorithm





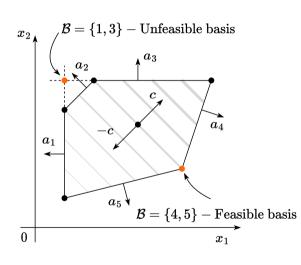


We will consider the following simple formulation of LP, which is, in fact, dual to the Standard form:

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

 s.t. $Ax \leq b$
 (LP.Inequality)

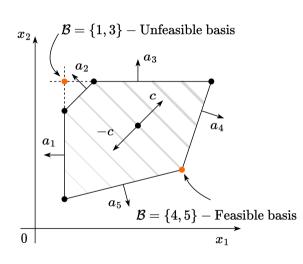
• Definition: a basis \mathcal{B} is a subset of n (integer) numbers between 1 and m, so that rank $A_{\mathcal{B}}=n$.



We will consider the following simple formulation of LP, which is, in fact, dual to the Standard form:

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

- Definition: a basis \mathcal{B} is a subset of n (integer) numbers between 1 and m, so that rank $A_{\mathcal{B}}=n$.
- Note, that we can associate submatrix $A_{\mathcal{B}}$ and corresponding right-hand side $b_{\mathcal{B}}$ with the basis \mathcal{B} .



We will consider the following simple formulation of LP, which is, in fact, dual to the Standard form:

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

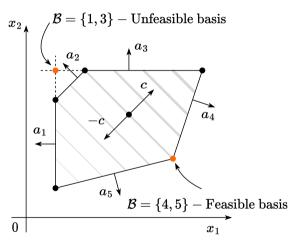
numbers between 1 and m, so that rank $A_B = n$.

• Note, that we can associate submatrix A_B and

• Definition: a **basis** \mathcal{B} is a subset of n (integer)

- Note, that we can associate submatrix $A_{\mathcal{B}}$ and corresponding right-hand side $b_{\mathcal{B}}$ with the basis \mathcal{B} .
- Also, we can derive a point of intersection of all these hyperplanes from the basis: $x_{\mathcal{B}} = A_{\mathcal{B}}^{-1} b_{\mathcal{B}}$.

⊕ n ø



We will consider the following simple formulation of LP, which is, in fact, dual to the Standard form:

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

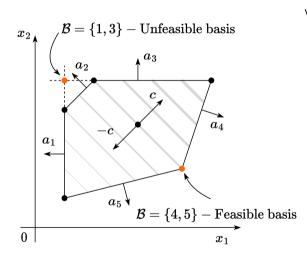
numbers between 1 and m, so that rank $A_{\mathcal{B}}=n$.

Note, that we can associate submatrix $A_{\mathcal{B}}$ and

• Definition: a basis \mathcal{B} is a subset of n (integer)

- Note, that we can associate submatrix $A_{\mathcal{B}}$ and corresponding right-hand side $b_{\mathcal{B}}$ with the basis \mathcal{B} .
- Also, we can derive a point of intersection of all these hyperplanes from the basis: $x_{\mathcal{B}} = A_{\mathbb{R}}^{-1} b_{\mathcal{B}}$.
- If $Ax_{\mathcal{B}} \leq b$, then basis \mathcal{B} is **feasible**.

Simplex Algorithm



We will consider the following simple formulation of LP, which is, in fact, dual to the Standard form:

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

- numbers between 1 and m, so that ${\rm rank}A_{\cal B}=n$.

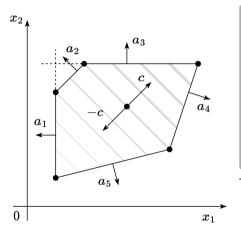
 Note, that we can associate submatrix $A_{\cal B}$ and
- corresponding right-hand side b_B with the basis B.

• Definition: a **basis** \mathcal{B} is a subset of n (integer)

- Also, we can derive a point of intersection of all these hyperplanes from the basis: $x_{\mathcal{B}} = A_{\mathcal{B}}^{-1} b_{\mathcal{B}}$.
- If $Ax_{\mathcal{B}} \leq b$, then basis \mathcal{B} is **feasible**.
- A basis ${\cal B}$ is optimal if $x_{\cal B}$ is an optimum of the LP.Inequality.

⊕ ∩ ∅

The solution of LP if exists lies in the corner

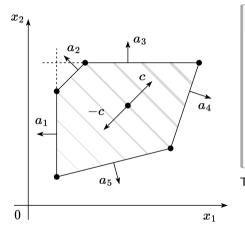


Theorem

1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point

The high-level idea of the simplex method is following:

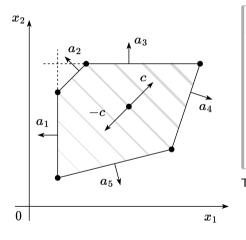
The solution of LP if exists lies in the corner



Theorem

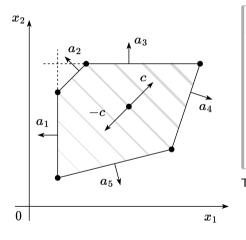
- 1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point
- 2. If Standard LP has solutions, then at least one such solution is a basic optimal point.

The high-level idea of the simplex method is following:



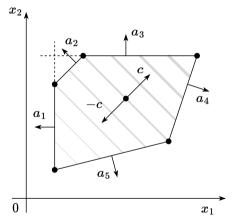
Theorem

- 1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point
- 2. If Standard LP has solutions, then at least one such solution is a basic optimal point.
- 3. If Standard LP is feasible and bounded, then it has an optimal solution.



Theorem

- 1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point
- 2. If Standard LP has solutions, then at least one such solution is a basic optimal point.
- 3. If Standard LP is feasible and bounded, then it has an optimal solution.



1 Theorem

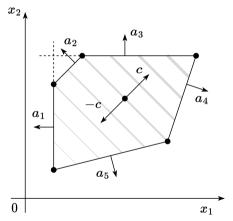
- 1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point
- 2. If Standard LP has solutions, then at least one such solution is a basic optimal point.
- 3. If Standard LP is feasible and bounded, then it has an optimal solution.

For proof see Numerical Optimization by Jorge Nocedal and Stephen J. Wright theorem 13.2

The high-level idea of the simplex method is following:

Ensure, that you are in the corner.

♥ ೧ Ø



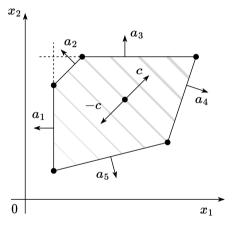
Theorem

- 1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point
- 2. If Standard LP has solutions, then at least one such solution is a basic optimal point.
- 3. If Standard LP is feasible and bounded, then it has an optimal solution.

For proof see Numerical Optimization by Jorge Nocedal and Stephen J. Wright theorem 13.2

- Ensure, that you are in the corner.
- Check optimality.





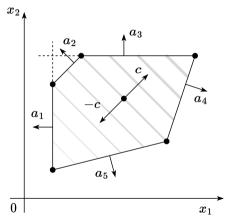
Theorem

- 1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point
- 2. If Standard LP has solutions, then at least one such solution is a basic optimal point.
- 3. If Standard LP is feasible and bounded, then it has an optimal solution.

For proof see Numerical Optimization by Jorge Nocedal and Stephen J. Wright theorem 13.2

- Ensure, that you are in the corner.
- Check optimality.
- If necessary, switch the corner (change the basis).





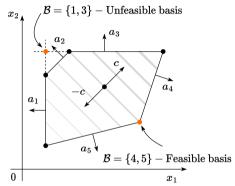
Theorem

- 1. If Standard LP has a nonempty feasible region, then there is at least one basic feasible point
- 2. If Standard LP has solutions, then at least one such solution is a basic optimal point.
- If Standard LP is feasible and bounded, then it has an optimal solution.

For proof see Numerical Optimization by Jorge Nocedal and Stephen J. Wright theorem 13.2

- Ensure, that you are in the corner.
- Check optimality.
- If necessary, switch the corner (change the basis).
- Repeat until converge.





Since we have a basis, we can decompose our objective vector c in this basis and find the scalar coefficients $\lambda_{\mathcal{B}}$:

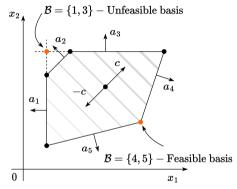
$$\lambda_{\mathcal{B}}^T A_{\mathcal{B}} = c^T \leftrightarrow \lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$$

i Theorem

If all components of $\lambda_{\mathcal{B}}$ are non-positive and \mathcal{B} is feasible, then \mathcal{B} is optimal.

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





Since we have a basis, we can decompose our objective vector c in this basis and find the scalar coefficients $\lambda_{\mathcal{B}}$:

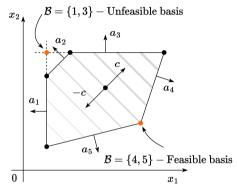
$$\lambda_{\mathcal{B}}^T A_{\mathcal{B}} = c^T \leftrightarrow \lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$$

1 Theorem

If all components of $\lambda_{\mathcal{B}}$ are non-positive and \mathcal{B} is feasible, then \mathcal{B} is optimal.

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





Since we have a basis, we can decompose our objective vector c in this basis and find the scalar coefficients $\lambda_{\mathcal{B}}$:

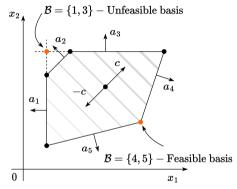
$$\lambda_{\mathcal{B}}^T A_{\mathcal{B}} = c^T \leftrightarrow \lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$$

i Theorem

If all components of $\lambda_{\mathcal{B}}$ are non-positive and \mathcal{B} is feasible, then \mathcal{B} is optimal.

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





Since we have a basis, we can decompose our objective vector c in this basis and find the scalar coefficients $\lambda_{\mathcal{B}}$:

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

i Theorem

If all components of $\lambda_{\mathcal{B}}$ are non-positive and \mathcal{B} is feasible, then \mathcal{B} is optimal.

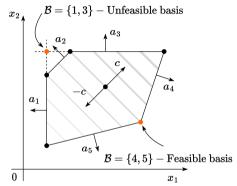
$$\exists x^* : Ax^* \leq b, c^T x^* < c^T x_{\mathcal{B}}$$

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

$$\lambda_{\mathcal{B}}^T A_{\mathcal{B}} x^* \geq \lambda_{\mathcal{B}}^T b_{\mathcal{B}}$$

$$c^T x^* \geq \lambda_{\mathcal{B}}^T A_{\mathcal{B}} x_{\mathcal{B}}$$





Since we have a basis, we can decompose our objective vector c in this basis and find the scalar coefficients $\lambda_{\mathcal{B}}$:

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

i Theorem

If all components of $\lambda_{\mathcal{B}}$ are non-positive and \mathcal{B} is feasible, then \mathcal{B} is optimal.

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

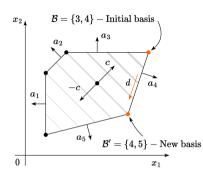
$$A_{\mathcal{B}}x^* \leq b_{\mathcal{B}}$$

$$\lambda_{\mathcal{B}}^T A_{\mathcal{B}}x^* \geq \lambda_{\mathcal{B}}^T b_{\mathcal{B}}$$

$$c^T x^* \geq \lambda_{\mathcal{B}}^T A_{\mathcal{B}}x_{\mathcal{B}}$$

$$c^T x^* > c^T x_{\mathcal{B}}$$

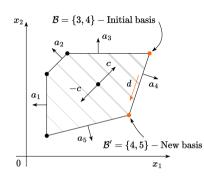




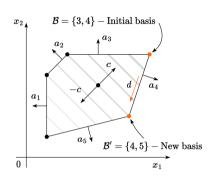
Suppose, some of the coefficients of λ_B are positive. Then we need to go through the edge of the polytope to the new vertex (i.e., switch the basis)

• Suppose, we have a basis \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$



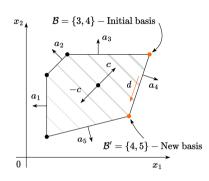


- Suppose, we have a basis \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Let's assume, that $\lambda^k_{\mathcal{B}}>0$. We'd like to drop k from the basis and form a new one:



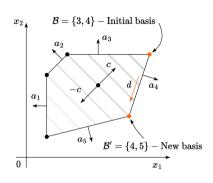
- Suppose, we have a basis \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Let's assume, that $\lambda_{\mathcal{B}}^k > 0$. We'd like to drop k from the basis and form a new one:

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



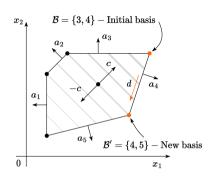
- Suppose, we have a basis \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Let's assume, that $\lambda_{\mathcal{B}}^k>0$. We'd like to drop k from the basis and form a new one:

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



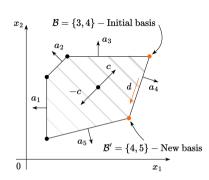
- Suppose, we have a basis \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Let's assume, that $\lambda_{\mathcal{B}}^k>0$. We'd like to drop k from the basis and form a new one:

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



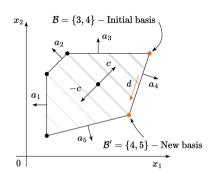
- Suppose, we have a basis \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Let's assume, that $\lambda_{\mathcal{B}}^k > 0$. We'd like to drop k from the basis and form a new one:

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



- Suppose, we have a basis \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Let's assume, that $\lambda_{\mathcal{B}}^k > 0$. We'd like to drop k from the basis and form a new one:

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



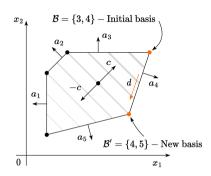
Suppose, some of the coefficients of λ_B are positive. Then we need to go through the edge of the polytope to the new vertex (i.e., switch the basis)

- Suppose, we have a basis \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Let's assume, that $\lambda_{\mathcal{B}}^k > 0$. We'd like to drop k from the basis and form a new one:

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

• For all $j \notin \mathcal{B}$ calculate the projection stepsize:

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



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)

- Suppose, we have a basis \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Let's assume, that $\lambda_{\mathcal{B}}^k > 0$. We'd like to drop k from the basis and form a new one:

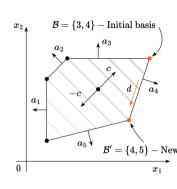
$$\begin{cases} A_{\mathcal{B}\backslash\{k\}}d = 0 \\ a_k^Td = -1 \end{cases} \qquad c^Td = \lambda_{\mathcal{B}}^TA_{\mathcal{B}}d = \sum_{i=1}^n \lambda_{\mathcal{B}}^i(A_{\mathcal{B}}d)^i = -\lambda_{\mathcal{B}}^k < 0$$

• For all $j \notin \mathcal{B}$ calculate the projection stepsize:

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

• Define the new vertex, that you will add to the new basis:

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



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)

- Suppose, we have a basis \mathcal{B} : $\lambda_{\mathcal{B}}^T = c^T A_{\mathcal{B}}^{-1}$
- Let's assume, that $\lambda_B^k > 0$. We'd like to drop k from the basis and form a new one:
 - $\begin{cases} A_{\mathcal{B}\setminus\{k\}}d = 0 \\ a_{\iota}^{T}d = -1 \end{cases} \qquad c^{T}d = \lambda_{\mathcal{B}}^{T}A_{\mathcal{B}}d = \sum_{i=1}^{n} \lambda_{\mathcal{B}}^{i}(A_{\mathcal{B}}d)^{i} = -\lambda_{\mathcal{B}}^{k} < 0$
- For all $i \notin \mathcal{B}$ calculate the projection stepsize:

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

Define the new vertex, that you will add to the new basis:

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

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

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

 $c^T x_{\mathcal{B}'} = c^T (x_{\mathcal{B}} + \mu_t d) = c^T x_{\mathcal{B}} + \mu_t c^T d$

Note, that changing basis implies objective function decreasing

$$f \to \min_{x,y,z}$$
 Simplex Algorithm

We aim to solve the following problem:

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

The proposed algorithm requires an initial basic feasible solution and corresponding basis.



We aim to solve the following problem:

We start by reformulating the problem:

y > 0, z > 0

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

The proposed algorithm requires an initial basic feasible solution and corresponding basis.

∌ o €

(2)

We aim to solve the following problem:

We start by reformulating the problem:

y > 0, z > 0

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

$$\sup_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} c^\top (y - z)$$
s.t. $Ax \le b$
s.t. $Ay - Az \le b$
(2)

The proposed algorithm requires an initial basic feasible solution and corresponding basis.

Given the solution of Problem 2 the solution of Problem 1 can be recovered and vice versa

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

Now we will try to formulate new LP problem, which solution will be basic feasible point for Problem 2. Which means, that we firstly run Simplex algorithm for Phase-1 problem and run Phase-2 problem with known starting point. Note, that basic feasible solution for Phase-1 should be somehow easily established.



$$\begin{aligned} & \min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} c^\top (y-z) \\ \text{s.t. } & Ay - Az \leq b \\ & y \geq 0, z \geq 0 \end{aligned} \qquad \text{(Phase-2 (Main LP))}$$



$$\begin{aligned} & \min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} c^\top (y-z) \\ \text{s.t. } & Ay - Az \leq b \\ & y \geq 0, z \geq 0 \end{aligned} \qquad \text{(Phase-2 (Main LP))}$$

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

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



$$\min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} c^\top (y-z)$$
 s.t. $Ay - Az \le b$ (Phase-2 (Main LP))
$$y > 0, z > 0$$

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

If Phase-2 (Main LP) problem has a feasible solution, then Phase-1 optimum is zero (i.e. all slacks ξ_i are zero).

Proof: trivial check.

$$\min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} c^\top (y-z)$$
 s.t. $Ay - Az \le b$ (Phase-2 (Main LP))
$$y > 0, z > 0$$

(Phase-1)

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

s.t.
$$Ay - Az \le b + \xi$$

$$y\geq 0, z\geq 0, \xi\geq 0$$

- If Phase-2 (Main LP) problem has a feasible solution, then Phase-1 optimum is zero (i.e. all slacks ξ_i are zero).
- Proof: trivial check. • If Phase-1 optimum is zero (i.e. all slacks ξ_i are zero), then we get a feasible basis for Phase-2. **Proof:** trivial check.

$$\min_{y \in \mathbb{R}^n, z \in \mathbb{R}^n} c^\top (y-z)$$
 s.t. $Ay - Az \le b$ (Phase-2 (Main LP))
$$y > 0, z > 0$$

(Phase-1)

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

s.t.
$$Ay - Az \le b + \xi$$

$$y\geq 0, z\geq 0, \xi\geq 0$$

- If Phase-2 (Main LP) problem has a feasible solution, then Phase-1 optimum is zero (i.e. all slacks ξ_i are zero).
- Proof: trivial check. • If Phase-1 optimum is zero (i.e. all slacks ξ_i are zero), then we get a feasible basis for Phase-2. **Proof:** trivial check.

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

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

- If Phase-2 (Main LP) problem has a feasible solution, then Phase-1 optimum is zero (i.e. all slacks ξ_i are zero).
- **Proof:** trivial check.

 If Phase-1 optimum is zero (i.e. all slacks ξ_i are zero), then we get a feasible basis for Phase-2. **Proof:** trivial check.

 Now we know, that if we can solve a Phase-1 problem then we will either find a starting point for the simplex method in the original method (if slacks are zero) or verify that the original problem was infeasible (if slacks are non-zero).

(Phase-1)

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

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

- If Phase-2 (Main LP) problem has a feasible solution, then Phase-1 optimum is zero (i.e. all slacks ξ_i are zero). Proof: trivial check.
- If Phase-1 optimum is zero (i.e. all slacks ξ_i are zero), then we get a feasible basis for Phase-2. Proof: trivial check.

 Now we know, that if we can solve a Phase-1 problem then we will either find a starting point for the simplex method in the original method (if slacks are zero) or verify that the original problem was infeasible (if slacks are non-zero).

(Phase-1)

But how to solve Phase-1? It has basic feasible solution (the problem has 2n + m variables and the point below ensures 2n + m inequalities are satisfied as equalities (active).)

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

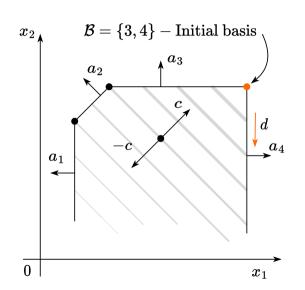
Convergence of the Simplex Algorithm



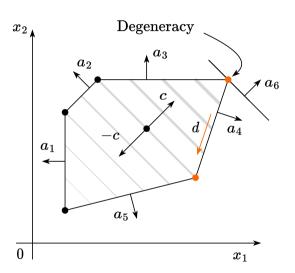


Unbounded budget set

In this case, all μ_j will be negative.



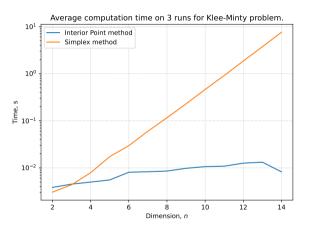
Degeneracy



One needs to handle degenerate corners carefully. If no degeneracy exists, one can guarantee a monotonic decrease of the objective function on each iteration.



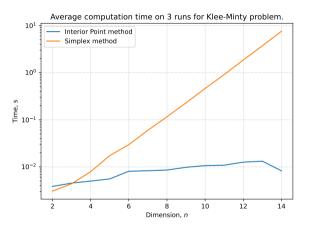
Exponential convergence



• A wide variety of applications could be formulated as linear programming.



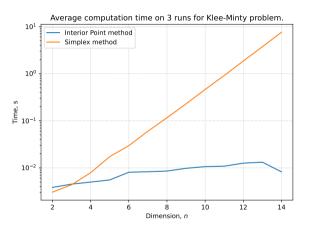
Exponential convergence



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



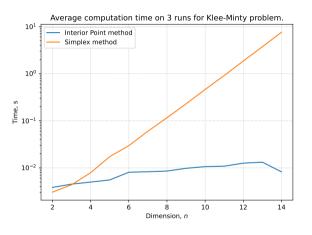
Exponential convergence



- A wide variety of applications could be formulated as linear programming.
- Simplex algorithm is simple but could work exponentially long.
- Khachiyan's ellipsoid method (1979) is the first to be proven to run at polynomial complexity for LPs.
 However, it is usually slower than simplex in real problems.



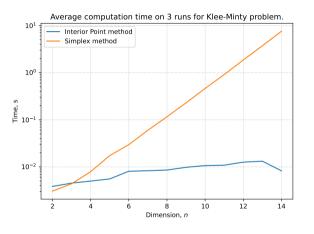
Exponential convergence



- A wide variety of applications could be formulated as linear programming.
- Simplex algorithm is simple but could work exponentially long.
- Khachiyan's ellipsoid method (1979) is the first to be proven to run at polynomial complexity for LPs. However, it is usually slower than simplex in real problems.
- Major breakthrough Narendra Karmarkar's method for solving LP (1984) using interior point method.



Exponential convergence



- A wide variety of applications could be formulated as linear programming.
- Simplex algorithm is simple but could work exponentially long.
- Khachiyan's ellipsoid method (1979) is the first to be proven to run at polynomial complexity for LPs.
 However, it is usually slower than simplex in real problems.
- Major breakthrough Narendra Karmarkar's method for solving LP (1984) using interior point method.
- 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.



Klee Minty example

Since the number of edge points is finite, the algorithm should converge (except for some degenerate cases, which are not covered here). However, the convergence could be exponentially slow, due to the high number of edges. There is the following iconic example when the simplex algorithm should perform exactly all vertexes.

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

$$\max_{x \in \mathbb{R}^n} 2^{n-1}x_1 + 2^{n-2}x_2 + \dots + 2x_{n-1} + x_n$$
 s.t. $x_1 \le 5$

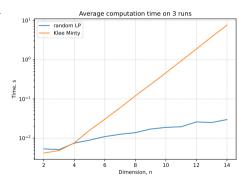
$$4x_1 + x_2 \le 25$$

$$8x_1 + 4x_2 + x_3 \le 125$$

$$\dots$$

$$2^n x_1 + 2^{n-1}x_2 + 2^{n-2}x_3 + \dots + x_n \le 5^n$$

 $x > 0$





Mixed Integer Programming



Mixed Integer Programming



Consider the following Mixed Integer Programming (MIP):

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

$$x_i \in \{0, 1\} \quad \forall i$$
(3)



Consider the following Mixed Integer Programming (MIP): Relax it to:

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

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

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

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

$$\begin{array}{c} x_1, x_2, x_3, x_4 \\ \text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \end{array}$$

s.t.
$$5x_1 + ix_2 + 4x_3 + 3x$$

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

$$4x_3 + 3x_4 \le 14$$

Consider the following Mixed Integer Programming (MIP): Relax it to:

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

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

S.t.
$$3x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$$
Optimal solution $x_i \in \{0,1\} \quad \forall i$

$$4 \le 14$$

$$\vee i$$

$$x_1 - x_2 - x_4 - 1$$
 and $x_1 - 21$

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

$$x_1 \equiv 0, x_2 \equiv x_3 \equiv x_4 \equiv 1, \text{ and } z \equiv 21$$

$$x = x_4 = 1$$
, and $z = 21$

$$,\,$$
 and $z=21.$

, and
$$z=z_1$$

and
$$z=z_1$$

and
$$z=z_1$$
.

and
$$z=21$$
.

and
$$z=21$$
.

and
$$z=21$$
.

(3)

$$3 + 3a$$

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

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

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







Mixed Integer Programming

Optimal solution

Consider the following Mixed Integer Programming (MIP): Relax it to:

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

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

$$c_4 \le 14$$

$$x_4 \le 14$$

$$_4 \le 14$$

$$\forall i$$

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

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

(3)

Optimal solution

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

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

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

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

Consider the following Mixed Integer Programming (MIP): Relax it to:

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

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

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

Optimal solution
$$x_i \in \{0, 1\}$$

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

$$x_4 = 1$$
 and $x = 21$

$$x_4 = 1$$
, and $z = 21$.

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

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

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

Optimal solution

- $x_1 = x_2 = 1, x_3 = 0.5, x_4 = 0, \text{ and } z = 22.$
- Rounding $x_3 = 0$: gives z = 19.

Consider the following Mixed Integer Programming (MIP): Relax it to:

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

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

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

Optimal solution

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

$$x_3 = x_4 = 1$$
, and $z = 21$.

and
$$z=21$$
.

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

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

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

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

- Rounding $x_3 = 0$: gives z = 19.
- Rounding $x_3 = 1$: Infeasible.

Consider the following Mixed Integer Programming (MIP): Relax it to:

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

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

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

Optimal solution

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

$$x_3 = x_4 = 1$$
, and $z = 21$.

and
$$z=21$$
.

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

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

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

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

- Rounding $x_3 = 0$: gives z = 19.
- Rounding $x_3 = 1$: Infeasible.

Consider the following Mixed Integer Programming (MIP): Relax it to:

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

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

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

Optimal solution
$$v_i \in \{0,1\}$$

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

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

 $z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \to \max_{x_1, x_2, x_3, x_4}$ st $5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$

s.t. $5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14$ $x_i \in [0, 1] \quad \forall i$

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

Optimal solution $x_i \in [0, 1]$

- Rounding $x_3 = 0$: gives z = 19.
- Rounding $x_3 = 1$: Infeasible.

- MIP is much harder, than LP
 - Naive rounding of LP relaxation of the initial MIP problem might lead to infeasible or suboptimal solution.

(3)

Consider the following Mixed Integer Programming (MIP): Relax it to:

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

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

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

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

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

Optimal solution

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

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

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

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

- Rounding $x_3 = 0$: gives z = 19.
- Rounding $x_3 = 1$: Infeasible.

- MIP is much harder, than LP
 - Naive rounding of LP relaxation of the initial MIP problem might lead to infeasible or suboptimal solution.

(3)

General MIP is NP-hard.

Optimal solution

Consider the following Mixed Integer Programming (MIP): Relax it to:

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

$$7x_2 + 4x_3 + 3x_4 \le 14$$

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

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

Optimal solution

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

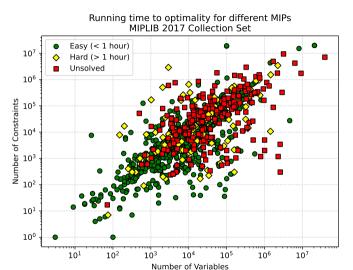
 $x_1 = x_2 = 1, x_3 = 0.5, x_4 = 0, \text{ and } z = 22.$

- Rounding $x_3 = 0$: gives z = 19.
- Rounding $x_3 = 1$: Infeasible.

- MIP is much harder, than LP
 - Naive rounding of LP relaxation of the initial MIP problem might lead to infeasible or suboptimal solution.
 - General MIP is NP-hard.
 - However, if the coefficient matrix of an MIP is a totally unimodular matrix, then it can be solved in polynomial time.

Unpredictable complexity of MIP

• It is hard to predict what will be solved quickly and what will take a long time

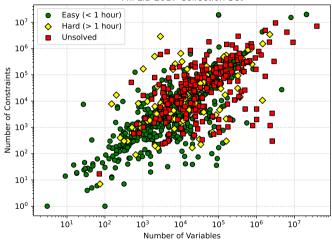




Unpredictable complexity of MIP

- It is hard to predict what will be solved quickly and what will take a long time
- ØDataset

Running time to optimality for different MIPs MIPLIB 2017 Collection Set

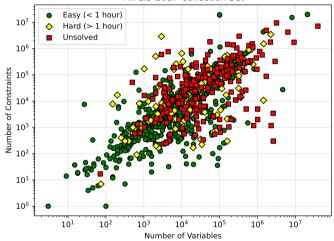




Unpredictable complexity of MIP

- It is hard to predict what will be solved quickly and what will take a long time
- ØDataset
- Source code

Running time to optimality for different MIPs MIPLIB 2017 Collection Set





Hardware progress vs Software progress

What would you choose, assuming, that the question posed correctly (you can compile software for any hardware and the problem is the same for both options)? We will consider the time period from 1992 to 2023.



Solving MIP with an old software on the modern hardware



Software

Solving MIP with a modern software on the old hardware

Hardware progress vs Software progress

What would you choose, assuming, that the question posed correctly (you can compile software for any hardware and the problem is the same for both options)? We will consider the time period from 1992 to 2023.



Solving MIP with an old software on the modern hardware



Software

speedup on MILP.

Solving MIP with a modern software on the old hardware

$$pprox 1.664.510 imes ext{speedup}$$

Moore's law states, that computational power doubles every 18 monthes.



R. Bixby conducted an intensive experiment with benchmarking all CPLEX software version starting from

1992 to 2007 and measured overall software progress (29000 times), later (in 2009) he was a cofounder of Gurobi optimization software, which gives additional ≈ 81

2 R. Bixby report



Hardware progress vs Software progress

What would you choose, assuming, that the question posed correctly (you can compile software for any hardware and the problem is the same for both options)? We will consider the time period from 1992 to 2023.



every 18 monthes.

Solving MIP with an old software on the modern hardware



Software

speedup on MILP.

Solving MIP with a modern software on the old hardware

$$pprox 1.664.510 imes ext{speedup}$$

Moore's law states, that computational power doubles



R. Bixby conducted an intensive experiment with benchmarking all CPLEX software version starting from 1992 to 2007 and measured overall software progress

(29000 times), later (in 2009) he was a cofounder of Gurobi optimization software, which gives additional ≈ 81

It turns out that if you need to solve a MILP, it is better to use an old computer and modern methods than vice versa, the newest computer and methods of the early 1990s!²

2 R. Bixby report Recent study Mixed Integer Programming