Properties of linear programming

Dipartimento di Ingegneria Industriale e dell’Informazione
Università degli Studi di Pavia

Industrial Automation
Outline

1. Representations of LP problems
2. LP: properties of the feasible region
   - Basics of convex geometry
3. The graphical solution for two-variable LP problems
4. Properties of linear programming
5. Algorithms for solving LP problems
Outline

1. Representations of LP problems

2. LP: properties of the feasible region
   - Basics of convex geometry

3. The graphical solution for two-variable LP problems

4. Properties of linear programming

5. Algorithms for solving LP problems
## Representations of LP problems

### LP in canonical form (LP-C)

$$\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax & \leq b \\
& \quad x & \geq 0
\end{align*}$$

Inequality “≤” constraints. Positivity constraints on all variables.
Representations of LP problems

LP in canonical form (LP-C)

\[
\begin{align*}
\min & \quad c^T x \\
\text{subject to} & \quad Ax \leq b \\
& \quad x \geq 0
\end{align*}
\]

Inequality “\(\leq\)” constraints. Positivity constraints on all variables.

LP in standard form (PL-S)

\[
\begin{align*}
\min & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

Equality constraints. Positivity constraints on all variables.
<table>
<thead>
<tr>
<th>Representations of LP problems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LP in canonical form (LP-C)</strong></td>
</tr>
</tbody>
</table>
| $\min \ c^T x$  
  $Ax \leq b$  
  $x \geq 0$  
| Inequality “$\leq$” constraints. Positivity constraints on all variables. |
| **LP in standard form (PL-S)** |
| $\min \ c^T x$  
  $Ax = b$  
  $x \geq 0$  
| Equality constraints. Positivity constraints on all variables. |
| **LP in generic form** |
| Mixed constraints $\leq$, $\geq$, $=$ and/or some variable is not constrained to be positive. |
Representations of LP problems

LP in canonical form (LP-C)

\[
\begin{align*}
\text{min} & \quad c^T x \\
Ax & \leq b \\
x & \geq 0
\end{align*}
\]

Inequality “\(\leq\)” constraints. Positivity constraints on all variables.

LP in standard form (PL-S)

\[
\begin{align*}
\text{min} & \quad c^T x \\
Ax & = b \\
x & \geq 0
\end{align*}
\]

Equality constraints. Positivity constraints on all variables.

LP in generic form

Mixed constraints \(\leq, \geq, =\) and/or some variable is not constrained to be positive.

The three forms are equivalent even if the conversion from one form to another one is possible only changing the number of variables and/or constraints.
Conversion between constraints

From \( \leq \) to =

\[ a_i^T x \leq b_i \iff \exists s_i \in \mathbb{R} : \begin{cases} a_i^T x + s_i = b_i \\ s_i \geq 0 \end{cases} \]

The additional variable \( s_i \) is called \textit{slack variable}
Conversion between constraints

From \( \leq \) to =

\[
a_i^T x \leq b_i \iff \exists s_i \in \mathbb{R} : \begin{cases} a_i^T x + s_i = b_i \\ s_i \geq 0 \end{cases}
\]

The additional variable \( s_i \) is called *slack variable*

From \( \geq \) to =

\[
a_i^T x \geq b_i \iff \exists s_i \in \mathbb{R} : \begin{cases} a_i^T x - s_i = b_i \\ s_i \geq 0 \end{cases}
\]

The additional variable \( s_i \) is called *excess variable*
Conversion between constraints

From $\leq$ to $=$

$$a_i^T x \leq b_i \iff \exists s_i \in \mathbb{R} : \begin{cases} a_i^T x + s_i = b_i \\ s_i \geq 0 \end{cases}$$

The additional variable $s_i$ is called *slack variable*.

From $\geq$ to $=$

$$a_i^T x \geq b_i \iff \exists s_i \in \mathbb{R} : \begin{cases} a_i^T x - s_i = b_i \\ s_i \geq 0 \end{cases}$$

The additional variable $s_i$ is called *excess variable*.

In both cases, a single constraint is replaced by two constraints.
Positivity constraints

Variables without sign constraints

\[ x_i \in \mathbb{R} \iff \exists x_i^+, x_i^- \in \mathbb{R} : \begin{cases} x_i = x_i^+ - x_i^- \\ x_i^+ \geq 0 \\ x_i^- \geq 0 \end{cases} \]

\( x_i^+ \) and \( x_i^- \) are two new variables representing the positive and negative part of \( x_i \in \mathbb{R} \), respectively.

The variable \( x_i \) is replaced with \( x_i^+ - x_i^- \) in the whole LP problem and constraints \( x_i^+ \geq 0, x_i^- \geq 0 \) are added.
Positivity constraints

Variables without sign constraints

\[ x_i \in \mathbb{R} \iff \exists x_i^+, x_i^- \in \mathbb{R} : \begin{cases} 
  x_i = x_i^+ - x_i^- \\
  x_i^+ \geq 0 \\
  x_i^- \geq 0 
\end{cases} \]

\( x_i^+ \) and \( x_i^- \) are two new variables representing the positive and negative part of \( x_i \in \mathbb{R} \), respectively.

The variable \( x_i \) is replaced with \( x_i^+ - x_i^- \) in the whole LP problem and constraints \( x_i^+ \geq 0, x_i^- \geq 0 \) are added.

Variables with sign constraints: from “\( \leq 0 \)” to “\( \geq 0 \)”:

\[ x_i \leq 0 \longrightarrow \xi_i \geq 0 \]

with \( \xi_i = -x_i \) that replaces \( x_i \) in the whole LP problem.
Example 1

Write the following problem in standard form

$$\max \{ c^T x : Ax = b \}$$
Example 1

Write the following problem in standard form

\[
\max_x \{ c^T x : Ax = b \}
\]

- There is no positivity constraint: we introduce two vectors \( x^+ \in \mathbb{R}^n, x^- \in \mathbb{R}^n \) and substitute \( x \) with \( x^+ - x^- \). We get

\[
\max_{x^+, x^-} \{ c^T (x^+ - x^-) : A(x^+ - x^-) = b, x^+ \geq 0, x^- \geq 0 \}.
\]
Example 1

Write the following problem in standard form

\[
\max_x \{ c^T x : Ax = b \}
\]

- There is no positivity constraint: we introduce two vectors \( x^+ \in \mathbb{R}^n, x^- \in \mathbb{R}^n \) and substitute \( x \) with \( x^+ - x^- \). We get

\[
\max_{x^+, x^-} \{ c^T (x^+ - x^-) : A(x^+ - x^-) = b, x^+ \geq 0, x^- \geq 0 \}.
\]

- Defining \( \xi = \begin{bmatrix} x^+ \\ x^- \end{bmatrix} \) the problem becomes

\[
\max_\xi \{ [c^T - c^T] \xi : [A - A] \xi = b, \xi \geq 0 \}
\]

In the conversion process the number of variables doubled
Example 2: conversion between canonical and standard forms

From canonical (PL-C) to standard (PL-S) form

\[
\max \ c^T x \quad \longrightarrow \quad \max \ \left\{ \begin{bmatrix} c & 0 \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} : \begin{bmatrix} A & I \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} = b, \begin{bmatrix} x \\ s \end{bmatrix} \geq 0 \right\}
\]

We introduced the vector of slack variables \( s \in \mathbb{R}^n \).
Example 2: conversion between canonical and standard forms

From canonical (PL-C) to standard (PL-S) form

\[
\begin{align*}
\max_{Ax \leq b, \ x \geq 0} c^T x & \quad \rightarrow \quad \max_{[x] \in \mathbb{R}^n} \left\{ \begin{bmatrix} c \\ 0 \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} : \begin{bmatrix} A & I \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} = b, \ \begin{bmatrix} x \\ s \end{bmatrix} \geq 0 \right\} \\
& \text{(1)}
\end{align*}
\]

We introduced the vector of slack variables \( s \in \mathbb{R}^n \).

From PL-S to PL-C

\[
\begin{align*}
\max_{Ax = b, \ x \geq 0} c^T x & \quad \rightarrow \quad \max_{x \in \mathbb{R}^n} \left\{ c^T x : \begin{bmatrix} A \\ -A \end{bmatrix} x \leq \begin{bmatrix} b \\ -b \end{bmatrix}, \ x \geq 0 \right\} \\
& \text{(2)}
\end{align*}
\]
Example 2: conversion between canonical and standard forms

From canonical (PL-C) to standard (PL-S) form

\[ \begin{align*} 
\max_{Ax \leq b \atop x \geq 0} c^T x & \quad \rightarrow \quad \max \begin{bmatrix} c & 0 \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} : \begin{bmatrix} A & I \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} = b, \begin{bmatrix} x \\ s \end{bmatrix} \geq 0 \end{align*} \] (1)

We introduced the vector of slack variables \( s \in \mathbb{R}^n \).

From PL-S to PL-C

\[ \begin{align*} 
\max_{Ax = b \atop x \geq 0} c^T x & \quad \rightarrow \quad \max_x \left\{ c^T x : \begin{bmatrix} A & -A \end{bmatrix} x \leq \begin{bmatrix} b \\ -b \end{bmatrix}, x \geq 0 \right\} \end{align*} \] (2)

Meaning of equivalence between the two forms:

- In (1): \( x^* \) is optimal for PL-C \( \iff \exists s^* : \begin{bmatrix} x^* \\ s^* \end{bmatrix} \) is optimal for PL-S
- In (2): \( x^* \) is optimal for PL-S \( \iff \begin{bmatrix} x^* \end{bmatrix} \) is optimal for PL-C
Example 3

Write the following problem in canonical form

\[ \min_{x_1, x_2, x_3} \ c_1 x_1 + c_2 x_2 + c_3 x_3 \]  \hspace{1cm} (3)

\[ a_{11} x_1 + a_{12} x_2 \leq b_1 \]  \hspace{1cm} (4)
\[ a_{22} x_2 + a_{23} x_3 \geq b_2 \]  \hspace{1cm} (5)
\[ a_{31} x_1 + a_{32} x_3 = b_3 \]  \hspace{1cm} (6)
\[ x_1 \geq 0 \]  \hspace{1cm} (7)
\[ x_2 \leq 0 \]  \hspace{1cm} (8)
Example 3

Write the following problem in canonical form

\[
\begin{aligned}
\min_{x_1, x_2, x_3} & \quad c_1 x_1 + c_2 x_2 + c_3 x_3 \\
\text{s.t.} & \quad a_{11} x_1 + a_{12} x_2 \leq b_1 \\
& \quad a_{22} x_2 + a_{23} x_3 \geq b_2 \\
& \quad a_{31} x_1 + a_{32} x_3 = b_3 \\
& \quad x_1 \geq 0 \\
& \quad x_2 \leq 0
\end{aligned}
\]

1. Positivity constraints on all variables:
   - replace \( x_2 \) with \( \xi_2 = -x_2 \)
   - \( x_3 \) is not sign constrained: we set \( x_3 = x_3^+ - x_3^- \) and add the constraints \( x_3^+ \geq 0 \) and \( x_3^- \geq 0 \)
Example 3

The original problem is now

\[
\min_{x_1, \xi_2, x_3^+, x_3^-} \quad c_1 x_1 - c_2 \xi_2 + c_3 x_3^+ - c_3 x_3^- \\
\]

\[
a_{11} x_1 - a_{12} \xi_2 \leq b_1 \\
-a_{22} \xi_2 + a_{23} x_3^+ - a_{23} x_3^- \geq b_2 \\
a_{31} x_1 + a_{32} x_3^+ - a_{32} x_3^- = b_3 \\
x_1 \geq 0 \\
\xi_2 \geq 0 \\
x_3^+ \geq 0 \\
x_3^- \geq 0
\]
Example 3
The original problem is now

$$\min_{x_1, \xi_2, x_3^+, x_3^-} c_1 x_1 - c_2 \xi_2 + c_3 x_3^+ - c_3 x_3^-$$  \hfill (9)$$

$$a_{11} x_1 - a_{12} \xi_2 \leq b_1$$  \hfill (10)$$

$$-a_{22} \xi_2 + a_{23} x_3^+ - a_{23} x_3^- \geq b_2$$  \hfill (11)$$

$$a_{31} x_1 + a_{32} x_3^+ - a_{32} x_3^- = b_3$$  \hfill (12)$$

$$x_1 \geq 0$$  \hfill (13)$$

$$\xi_2 \geq 0$$  \hfill (14)$$

$$x_3^+ \geq 0$$  \hfill (15)$$

$$x_3^- \geq 0$$  \hfill (16)$$

2. Constraints ”≤”:

- we replace (11) with $$a_{22} \xi_2 - a_{23} x_3^+ + a_{23} x_3^- \leq -b_2$$
- we replace (12) with $$a_{31} x_1 + a_{32} x_3^+ - a_{32} x_3^- \leq b_3$$ and
  $$-a_{31} x_1 - a_{32} x_3^+ + a_{32} x_3^- \leq -b_3$$
Example 3

The LP problem is now in canonical form

\[
\min_{x_1, \xi_2, x_3^+, x_3^-} c_1 x_1 - c_2 \xi_2 + c_3 x_3^+ - c_3 x_3^-
\]

\[
a_{11} x_1 - a_{12} \xi_2 \leq b_1
\]

\[
+ a_{22} \xi_2 - a_{23} x_3^+ + a_{23} x_3^- \leq -b_2
\]

\[
a_{31} x_1 + a_{32} x_3^+ - a_{32} x_3^- \leq b_3
\]

\[
- a_{31} x_1 - a_{32} x_3^+ + a_{32} x_3^- \leq -b_3
\]

\[
x_1 \geq 0
\]

\[
\xi_2 \geq 0
\]

\[
x_3^+ \geq 0
\]

\[
x_3^- \geq 0
\]
Example 3 - matrix notation

We define $x = [x_1 \quad \xi_2 \quad x_3^+ \quad x_3^-]^T$ and obtain

$$\min_{Ax \leq b, \quad x \geq 0} \begin{bmatrix} c_1 & -c_2 & c_3 & -c_3 \end{bmatrix} x$$

\[ (26) \]

$$A = \begin{bmatrix} a_{11} & -a_{12} & 0 & 0 \\ 0 & a_{22} & -a_{23} & +a_{23} \\ a_{31} & 0 & a_{32} & -a_{32} \\ -a_{31} & 0 & -a_{32} & +a_{32} \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ -b_2 \\ b_3 \\ -b_3 \end{bmatrix}$$
Example 3 - matrix notation

We define \( x = [x_1 \quad \xi_2 \quad x_3^+ \quad x_3^-]^T \) and obtain

\[
\min_{A x \leq b, \quad x \geq 0} \left[ \begin{array}{cccc}
 c_1 & -c_2 & c_3 & -c_3
\end{array} \right] x
\]

(26)

\[
A = \begin{bmatrix}
 a_{11} & -a_{12} & 0 & 0 \\
 0 & a_{22} & -a_{23} & +a_{23} \\
 a_{31} & 0 & a_{32} & -a_{32} \\
 -a_{31} & 0 & -a_{32} & +a_{32}
\end{bmatrix}
\]

\[
b = \begin{bmatrix}
 b_1 \\
 -b_2 \\
 b_3 \\
 -b_3
\end{bmatrix}
\]

Meaning of equivalence between different forms

If \( x^* = [x_1^* \quad \xi_2^* \quad (x_3^+)^* \quad (x_3^-)^*] \) is the optimal solution to (26) and \( [\tilde{x}_1 \quad \tilde{x}_2 \quad \tilde{x}_3] \) is the optimal solution to the original problem, it holds

\[
\tilde{x}_1 = x_1^*
\]

\[
\tilde{x}_2 = -\xi_2^*
\]

\[
\tilde{x}_3 = (x_3^+)^* - (x_3^-)^*
\]
Outline

1. Representations of LP problems
2. LP: properties of the feasible region
   - Basics of convex geometry
3. The graphical solution for two-variable LP problems
4. Properties of linear programming
5. Algorithms for solving LP problems
Convex geometry

Hyperplane

The set $H = \{ x \in \mathbb{R}^n : a^T x = b \}$ with $a \in \mathbb{R}^n$, $a \neq 0$, $b \in \mathbb{R}$ is called \textit{hyperplane} in $\mathbb{R}^n$. The boundary of the closed half-spaces

$$H^- = \{ x \in \mathbb{R}^n : a^T x \leq b \}$$

$$H^+ = \{ x \in \mathbb{R}^n : a^T x \geq b \}$$

is the \textit{supporting hyperplane} $H$.
Polyhedra and polytopes

A polyhedron in $\mathbb{R}^n$ is the intersections of a finite and strictly positive number of half-spaces in $\mathbb{R}^n$.

- If $K$ is a polyhedron, $\exists A, b$ of suitable dimensions such that $K = \{x \in \mathbb{R}^n : Ax \leq b\}$.

- If $K$ is bounded, it is called polytope.
Convex geometry

Polyhedra and polytopes

A polyhedron in $\mathbb{R}^n$ is the intersections of a finite and strictly positive number of half-spaces in $\mathbb{R}^n$.

- If $K$ is a polyhedron, $\exists A, b$ of suitable dimensions such that $K = \{x \in \mathbb{R}^n : Ax \leq b\}$.

- If $K$ is bounded, it is called polytope.

A polytope is a closed and convex set
Convex geometry

Polyhedra and polytopes

A polyhedron in $\mathbb{R}^n$ is the intersections of a finite and strictly positive number of half-spaces in $\mathbb{R}^n$.

- If $K$ is a polyhedron, $\exists A, b$ of suitable dimensions such that $K = \{x \in \mathbb{R}^n : Ax \leq b\}$.
- If $K$ is bounded, it is called polytope.

- A polytope is a closed and convex set
- The feasible region of an LP problem is a polyhedron
Convex geometry

Remarks

The pair \((A, b)\) defining the polyhedron \(K = \{x \in \mathbb{R}^n : Ax \leq b\}\) is not unique.
Convex geometry

Remarks

The pair \((A, b)\) defining the polyhedron \(K = \{x \in \mathbb{R}^n : Ax \leq b\}\) is not unique.

- If \(\tilde{A}\) and \(\tilde{b}\) coincide with \(A\) and \(b\) up to a row permutation, then \((\tilde{A}, \tilde{b})\) defines \(K\)
Remarks

The pair \((A, b)\) defining the polyhedron \(K = \{x \in \mathbb{R}^n : Ax \leq b\}\) is not unique.

- If \(\tilde{A}\) and \(\tilde{b}\) coincide with \(A\) and \(b\) up to a row permutation, then \((\tilde{A}, \tilde{b})\) defines \(K\)
- \((\alpha A, \alpha b), \alpha > 0\) defines \(K\)
Convex geometry

Remarks

The pair \((A, b)\) defining the polyhedron \(K = \{x \in \mathbb{R}^n : Ax \leq b\}\) is not unique.

- If \(\tilde{A}\) and \(\tilde{b}\) coincide with \(A\) and \(b\) up to a row permutation, then \((\tilde{A}, \tilde{b})\) defines \(K\).
- \((\alpha A, \alpha b), \alpha > 0\) defines \(K\).
- A constraint in \(Ax \leq b\) is redundant if \(K\) does not change when removed. If redundant constraints are added to or removed from those defining \(K\), one gets a new pair \((\tilde{A}, \tilde{b})\) that still defines \(K\).
Convex geometry

Remarks
- The empty set is a polyhedron ...
Convex geometry

Remarks

- The empty set is a polyhedron ...

![Graph showing lines in a coordinate system with x1 and x2 axes.](image)
Convex geometry

Remarks

- The empty set is a polyhedron ...

... it is also a polytope
Convex geometry

Remarks

- The empty set is a polyhedron ...
- It is also a polytope
- $\mathbb{R}^n$ is not a polyhedron
Convex geometry

Extreme points

Let $S \subseteq \mathbb{R}^n$ be a convex set. A point $z \in S$ is called *extreme point* if there are not two points $x, y \in S$ different from $z$, such that $z$ belongs to the segment $\overline{xy}$.
**Convex geometry**

**Definition**

Let $K \subset \mathbb{R}^n$ be a polyhedron. Then

- its extreme points are called *vertices*;
- the intersection of $K$ with one or more supporting hyperplanes is called *face*;
- faces of dimension 1 are called *edges*. Faces of dimension $n - 1$ are called *facets* or maximal faces.
Convex geometry

**Theorem**

A polyhedron has a finite number\(^a\) of vertices.

\(^a\)It can be zero.
Convex geometry

Theorem

A polyhedron has a finite number\(^a\) of vertices.

\(^a\)It can be zero.

Representations of a polytope

Definition

The point \(z \in \mathbb{R}^n\) is a convex combination of \(k\) points \(x_1, x_2, \ldots, x_k\) if \(\exists \lambda_1, \lambda_2, \ldots, \lambda_k \geq 0\) verifying \(\sum_{i=1}^{k} \lambda_i = 1\) and such that

\[
z = \sum_{i=1}^{k} \lambda_i x_k
\]

A segment \(\overline{xy}\) is the set of the convex combinations of \(x\) and \(y\).
Convex geometry

Minkowski-Weyl theorem

Let $P$ be a polytope. Then, a point $x \in P$ is a convex combination of the vertices of $P$. 

Example

All points $x$ of the triangle can be written as $x = \sum_{i=1}^{3} \lambda_i p_i$ for suitable $\lambda_i \geq 0$ such that $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

Remark
The theorem does not hold for generic polyhedra (think about a cone ...)

(DIS) Properties of linear programming
Convex geometry

Minkowski-Weyl theorem

Let $P$ be a polytope. Then, a point $x \in P$ is a convex combination of the vertices of $P$.

Example

All points $x$ of the triangle can be written as $x = \sum_{i=1}^{3} \lambda_i p_i$ for suitable $\lambda_i \geq 0$ such that $\lambda_1 + \lambda_2 + \lambda_3 = 1$. 
**Convex geometry**

**Minkowski-Weyl theorem**

Let $P$ be a polytope. Then, a point $x \in P$ is a convex combination of the vertices of $P$.

**Example**

All points $x$ of the triangle can be written as $x = \sum_{i=1}^{3} \lambda_i p_i$ for suitable $\lambda_i \geq 0$ such that $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

**Remark**

The theorem does not hold for generic polyhedra (think about a cone ...).
Outline

1. Representations of LP problems
2. LP: properties of the feasible region
   - Basics of convex geometry
3. The graphical solution for two-variable LP problems
4. Properties of linear programming
5. Algorithms for solving LP problems
The graphical solution for two-variable LP problems

The feasible region and optimal solution of LP problems with only two variables \( x = [x_1, x_2]^T \) can be represented graphically.

**Isocost lines**

Given a level \( \alpha \in \mathbb{R} \) the level surface of the cost is

\[
C_\alpha \left[ c^T x \right] = \left\{ x \in \mathbb{R}^2 : c^T x = \alpha \right\}.
\]

For different values of \( \alpha \) one gets parallel lines called *isocost lines*.
Example 1

Product mix

\[
\max_{M_1, M_2} 30M_1 + 20M_2
\]

8\(M_1 + 4M_2 \leq 640\) \hspace{1cm} (28)
4\(M_1 + 6M_2 \leq 540\) \hspace{1cm} (29)
\(M_1 + M_2 \leq 100\) \hspace{1cm} (30)

\(M_1, M_2 \geq 0\). \hspace{1cm} (31)

Feasible region = hatched area
Example 1

Product mix

\[
\max_{M_1, M_2} 30M_1 + 20M_2
\]

\[8M_1 + 4M_2 \leq 640 \quad (28)\]

\[4M_1 + 6M_2 \leq 540 \quad (29)\]

\[M_1 + M_2 \leq 100 \quad (30)\]

\[M_1, M_2 \geq 0. \quad (31)\]

Isocost lines: \( C_\alpha [30M_1 + 20M_2] : M_2 = \frac{\alpha}{20} - \frac{30}{20} M_1 \)

E.g. \( \alpha = 1800 \rightarrow \) line passing through \((0, 90)\) and \((60, 0)\)
Example 1

Product mix

\[
\begin{align*}
\max_{M_1, M_2} & \quad 30M_1 + 20M_2 \\
8M_1 + 4M_2 & \leq 640 \quad (28) \\
4M_1 + 6M_2 & \leq 540 \quad (29) \\
M_1 + M_2 & \leq 100 \quad (30) \\
M_1, M_2 & \geq 0. \quad (31)
\end{align*}
\]

As \(\alpha\) increases, isocost lines move in the arrow direction.
Example 1

**Product mix**

\[
\begin{align*}
\text{max } & 30M_1 + 20M_2 \\
\text{subject to } & 8M_1 + 4M_2 \leq 640 \quad (28) \\
& 4M_1 + 6M_2 \leq 540 \quad (29) \\
& M_1 + M_2 \leq 100 \quad (30) \\
& M_1, M_2 \geq 0. \quad (31)
\end{align*}
\]

The optimal solution is \((60, 40)\) and it is given by \(C_{2600}\): for greater values of \(\alpha\), the isocost line does not intersect the feasible region.

The optimal solution is a vertex of the feasible region.
Example 2

**Diet problem**

\[
\begin{align*}
\text{min } & 20A_1 + 30A_2 \\
\text{s.t. } & A_1 \geq 2 \quad (32) \\
& 2A_1 + A_2 \geq 12 \quad (33) \\
& 2A_1 + 5A_2 \geq 36 \quad (34) \\
& A_2 \geq 4 \quad (35) \\
& A_1, A_2 \geq 0. \quad (36)
\end{align*}
\]

Feasible region = hatched area
Example 2

Diet problem

\[
\begin{align*}
\min & \quad 20A_1 + 30A_2 \\
\text{subject to} & \quad A_1 \geq 2 \quad (32) \\
& \quad 2A_1 + A_2 \geq 12 \quad (33) \\
& \quad 2A_1 + 5A_2 \geq 36 \quad (34) \\
& \quad A_2 \geq 4 \quad (35) \\
& \quad A_1, A_2 \geq 0. \quad (36)
\end{align*}
\]

Isocost lines: \( C_\alpha [20A_1 + 30A_2] : A_2 = -\frac{20}{30}A_1 + \frac{\alpha}{30} \)
Example 2

Diet problem

\[
\text{min } 20A_1 + 30A_2 \\
A_1, A_2 \geq 0
\]

\[
A_1 \geq 2 \quad (32)
\]

\[
2A_1 + A_2 \geq 12 \quad (33)
\]

\[
2A_1 + 5A_2 \geq 36 \quad (34)
\]

\[
A_2 \geq 4 \quad (35)
\]

\[
A_1, A_2 \geq 0. \quad (36)
\]

As $\alpha$ decreases, isocost lines move in the arrow direction.
Example 2

Diet problem

\[
\begin{align*}
\text{min} & \quad 20A_1 + 30A_2 \\
A_1, A_2 & \geq 0
\end{align*}
\]

\[
\begin{align*}
A_1 & \geq 2 \quad (32) \\
2A_1 + A_2 & \geq 12 \quad (33) \\
2A_1 + 5A_2 & \geq 36 \quad (34) \\
A_2 & \geq 4 \quad (35) \\
A_1, A_2 & \geq 0 \quad (36)
\end{align*}
\]

The optimal solution is \((3, 6)\) and it is given by \(C_{240}\). The optimal solution is a vertex of the feasible region.
Example: multiple solutions

LP problem

\[
\begin{align*}
\max & \quad 30x_1 + 30x_2 \\
\text{s.t.} & \quad x_2 \leq 80 \quad (37) \\
& \quad x_1 \leq 60 \quad (38) \\
& \quad x_1 + x_2 \leq 100 \quad (39) \\
& \quad x_1, x_2 \geq 0. \quad (40)
\end{align*}
\]

Feasible region = hatched area
Example: multiple solutions

LP problem

\[
\begin{align*}
\text{max } & 30x_1 + 30x_2 \\
\text{s.t. } & x_2 \leq 80 \quad (37) \\
& x_1 \leq 60 \quad (38) \\
& x_1 + x_2 \leq 100 \quad (39) \\
& x_1, x_2 \geq 0. \quad (40)
\end{align*}
\]

Isocost lines: \( C_\alpha [30x_1 + 30x_2] : x_2 = -x_1 + \frac{\alpha}{30} \)
Example: multiple solutions

LP problem

\[
\max_{x_1, x_2} 30x_1 + 30x_2
\]

\[
x_2 \leq 80 \quad (37)
\]

\[
x_1 \leq 60 \quad (38)
\]

\[
x_1 + x_2 \leq 100 \quad (39)
\]

\[
x_1, x_2 \geq 0. \quad (40)
\]

As \(\alpha\) decreases, isocost lines move in the arrow direction
Example: multiple solutions

LP problem

\[
\text{max } 30x_1 + 30x_2
\]

\[
x_2 \leq 80 \quad (37)
\]

\[
x_1 \leq 60 \quad (38)
\]

\[
x_1 + x_2 \leq 100 \quad (39)
\]

\[
x_1, x_2 \geq 0. \quad (40)
\]

The optimal isocost line is \( C_{6000} \) and intersects the face \( S \) of the feasible region: \( \forall x \in S \) is an optimal solution.
There exists at least an optimal solution that is a vertex of the feasible region.
Example: unbounded problem

LP problem

\[ \max_{x_1, x_2} x_1 + 2x_2 \]

\[ x_2 \leq 1 \quad (41) \]

\[ -x_1 - x_2 \leq -2 \quad (42) \]

\[ x_1, x_2 \geq 0. \quad (43) \]

Feasible region = hatched area
Example: unbounded problem

**LP problem**

\[
\begin{align*}
\text{max } & \quad x_1 + 2x_2 \\
\text{s.t. } & \quad x_2 \leq 1 \quad \text{(41)} \\
& \quad -x_1 - x_2 \leq -2 \quad \text{(42)} \\
& \quad x_1, x_2 \geq 0. \quad \text{(43)}
\end{align*}
\]

**Isocost lines:** \( C_\alpha [x_1 + 2x_2] : \quad x_2 = -\frac{1}{2} x_1 + \frac{\alpha}{2} \)

As \( \alpha \) increases, isocost lines move in the arrow direction.
Example: unbounded problem

\[ \begin{align*}
\text{LP problem} \\
\max_{x_1, x_2} & \quad x_1 + 2x_2 \\
x_2 & \leq 1 \quad (41) \\
-x_1 - x_2 & \leq -2 \quad (42) \\
x_1, x_2 & \geq 0 \quad (43)
\end{align*} \]

- The cost can grow unbounded: \( \forall \alpha > 0 \) the isocost line \( C_\alpha [x_1 + 2x_2] \) intersects the feasible region.
- The LP problem is unbounded
Example: unbounded problem

**LP problem**

\[
\begin{align*}
\max_{x_1, x_2} & \quad x_1 + 2x_2 \\
x_2 & \leq 1 \quad (41) \\
-x_1 - x_2 & \leq -2 \quad (42) \\
x_1, x_2 & \geq 0. \quad (43)
\end{align*}
\]

Unboundedness is often due to modeling errors. One would *automatically* detect it, especially when the number of variables is high.
Example: infeasible problem

**LP problem**

\[
\begin{align*}
\text{max } & \quad x_1 + x_2 \\
\text{s.t. } & \quad -x_1 + 2x_2 \leq -1 \quad (44) \\
& \quad x_1 - x_2 \leq -1 \quad (45) \\
& \quad x_1, x_2 \geq 0. \quad (46)
\end{align*}
\]

The feasibility region is empty → infeasible problem
Example: infeasible problem

LP problem

\[
\begin{align*}
\text{max } & x_1 + x_2 \\
\text{subject to } & -x_1 + 2x_2 \leq -1 \quad (44) \\
& x_1 - x_2 \leq -1 \quad (45) \\
& x_1, x_2 \geq 0. \quad (46)
\end{align*}
\]

Infeasibility is often due to modelling errors. One would automatically detect it, especially when the number of variables is high.
Outline

1. Representations of LP problems
2. LP: properties of the feasible region
   - Basics of convex geometry
3. The graphical solution for two-variable LP problems
4. Properties of linear programming
5. Algorithms for solving LP problems
Properties of linear programming

Fundamental theorem of linear programming

Let \( \{ \max c^T x : x \in X \} \) be an LP problem where \( X \) is a polyhedron and \( x \in \mathbb{R}^n \). If the problem is feasible, then only one of the following is true:

1. the problem is unbounded;
2. there is at least a vertex of \( X \) that is an optimal solution.

Corollary

If \( X \) is a nonempty polytope, then there is a vertex of \( X \) that is an optimal solution.
Properties of linear programming

Proof of the corollary

- Let $x_1, x_2, \ldots, x_k$ be vertices of $P$ (their number is finite) and $z^* = \max \{ c^T x_i, i = 1, 2, \ldots, k \}$ (maximum of vertex costs).
- We want to show that $\forall y \in P$ one has $c^T y \leq z^*$. 
Properties of linear programming

Proof of the corollary

- Let \( x_1, x_2, \ldots, x_k \) be vertices of \( P \) (their number is finite) and \( z^* = \max \{ c^T x_i, \ i = 1, 2, \ldots, k \} \) (maximum of vertex costs).
- We want to show that \( \forall y \in P \) one has \( c^T y \leq z^* \).
- From Minkowski-Weyl theorem:
  \[ y \in P \Rightarrow \exists \lambda_1, \lambda_2, \ldots, \lambda_k \geq 0 : \sum_{i=1}^{k} \lambda_i = 1 \text{ and } y = \sum_{i=1}^{k} \lambda_i x_i. \]
Proof of the corollary

- Let $x_1, x_2, \ldots, x_k$ be vertices of $P$ (their number is finite) and $z^* = \max \left\{ c^T x_i, i = 1, 2, \ldots, k \right\}$ (maximum of vertex costs).

- We want to show that $\forall y \in P$ one has $c^T y \leq z^*$.

- From Minkowski-Weyl theorem:
  
  $y \in P \Rightarrow \exists \lambda_1, \lambda_2, \ldots, \lambda_k \geq 0 : \sum_{i=1}^{k} \lambda_i = 1$ and $y = \sum_{i=1}^{k} \lambda_i x_i$.

- Then

\[
c^T y = c^T \sum_{i=1}^{k} \lambda_i x_i = \sum_{i=1}^{k} \lambda_i (c^T x_i) \leq \sum_{i=1}^{k} \lambda_i z^* = z^*.
\]
Outline

1. Representations of LP problems
2. LP: properties of the feasible region
   • Basics of convex geometry
3. The graphical solution for two-variable LP problems
4. Properties of linear programming
5. Algorithms for solving LP problems
Vertex enumeration

If an LP problem is feasible and bounded one can

- compute all vertices $x_1, x_2, \ldots, x_k$ of $X$
- compute $z_i = c^T x_i$, $i = 1, 2, \ldots, k$ (cost of vertices)

and obtain an optimal solution as $x_k : c^T x_k = \max \{z_1, z_2, \ldots, z_k\}$
Algorithms for solving LP problems

Vertex enumeration

If an LP problem is feasible and bounded one can

- compute all vertices \(x_1, x_2, \ldots, x_k\) of \(X\)
- compute \(z_i = c^T x_i, \ i = 1, 2, \ldots, k\) (cost of vertices)

and obtain an optimal solution as \(x_k : c^T x_k = \max \{z_1, z_2, \ldots, z_k\}\)

The number of vertices of the feasible region can grow exponentially with \(n \rightarrow \text{computationally prohibitive}\)

Example: let \(X\) be an hypercube

<table>
<thead>
<tr>
<th>(n)</th>
<th>(X)</th>
<th>N. of vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>square</td>
<td>(2^2 = 4)</td>
</tr>
<tr>
<td>3</td>
<td>cube</td>
<td>(2^3 = 8)</td>
</tr>
<tr>
<td>1000</td>
<td>ipercube</td>
<td>(2^{1000} \approx 10^{300})</td>
</tr>
</tbody>
</table>

If the computation of a vertex requires \(10^{-9} s\), when \(n = 1000\)
the computation time is greater than \(10^{300} 10^{-9} = 10^{291} s > 10^{281}\) centuries
Efficient algorithms for linear programming

**Simplex algorithm**

Developed by G. Dantzig in 1947

- iterative procedure for generating vertices of $X$ with decreasing cost (for minimization problems) and for assessing their optimality.
  - $m$ constraints and $n$ variables: $\to$ maximal number of vertices
    \[
    \binom{n}{m} = \frac{n!}{m!(n-m)!}
    \]
  - in the worst case the complexity of the method is exponential in the dimension of the LP problem
  - ”on average” the method is numerically robust and much more efficient than vertex enumeration.

- infeasibility and unboundedness of the LP problem are automatically detected
Efficient algorithms for linear programming

Interior point method

Developed by N. Karmarkar in 1984

- iterative procedure that generates a sequence of points lying in the interior of $X$ and converging to an optimal vertex
  - Convergence to an optimal solution requires a computational time that grows polynomially with the number of variables and constraints of the LP problem
  - for large-scale LP problems, it can be much more efficient than the simplex algorithm

- infeasibility and unboundedness of the LP problem are automatically detected