Applications of Linear Programming

lecturer: András London

University of Szeged
Institute of Informatics
Department of Computational Optimization

Lecture 1
Why LP?

- **Linear programming** (LP, also called linear optimization) is a method to achieve the best outcome (such as maximum profit or lowest cost) in a *mathematical model*.
Linear programming (LP, also called linear optimization) is a method to achieve the best outcome (such as maximum profit or lowest cost) in a mathematical model whose requirements are represented by linear relationships. It is a special case of mathematical programming (mathematical optimization).
Why LP?

- **Linear programming** (LP, also called linear optimization) is a method to achieve the best outcome (such as maximum profit or lowest cost) in a **mathematical model**
  - whose requirements are represented by linear relationships
  - it is a special case of mathematical programming (mathematical optimization).

- Widely used in business and economics, and is also utilized for some engineering problems
Linear programming (LP, also called linear optimization) is a method to achieve the best outcome (such as maximum profit or lowest cost) in a mathematical model whose requirements are represented by linear relationships. It is a special case of mathematical programming (mathematical optimization).

Widely used in business and economics, and is also utilized for some engineering problems.

Industries that use linear programming models include transportation, energy, telecommunications, and manufacturing.
Why LP?

- **Linear programming** (LP, also called linear optimization) is a method to achieve the best outcome (such as maximum profit or lowest cost) in a **mathematical model**
  - whose requirements are represented by linear relationships
  - it is a special case of mathematical programming (mathematical optimization).
- **Widely used in business and economics**, and is also utilized for some engineering problems
- Industries that use linear programming models include transportation, energy, telecommunications, and manufacturing
- Useful in modeling diverse types of problems in, for instance
  - planning
  - routing
  - scheduling
  - assignment
Brief history

Details of the contribution

Fourier (1827) proposed a method for solving linear programs (LPs).

Kantorovich proposed a method for solving LPs at the same time.

Koopmans formulated classical economic problems as LPs (shared the Nobel Prize in 1975).

Dantzig (1945-46) developed the general formulation for planning problems in the US Air Force and invented the simplex method.

Khachiyan (1979) developed a polynomial-time algorithm for solving LPs.

Karmarkar (1984) made a breakthrough with the interior-point method for solving LPs.
Brief history

- Dates back to *Fourier* (1827, a method for solving LP)
Brief history

- Dates back to Fourier (1827, a method for solving LP)
- 1939 – Kantorovich proposed a method for solving LP, same time Koopmans formulated classical economic problems as LPs (shared Nobel-prize in 1975)
Brief history

- Dates back to *Fourier* (1827, a method for solving LP)
- 1939 – *Kantorovich* proposed a method for solving LP, same time *Koopmans* formulated classical economic problems as LPs (shared Nobel-prize in 1975)
- 1945-46 *Danzig*: general formulation for planning problems in US Air Force. Invented the **simplex method**
Brief history

- Dates back to Fourier (1827, a method for solving LP)
- 1939 – Kantorovich proposed a method for solving LP, same time Koopmans formulated classical economic problems as LPs (shared Nobel-prize in 1975)
- 1979 – Khachiyan's polynomial time algorithm
Brief history

- Dates back to *Fourier* (1827, a method for solving LP)
- 1939 – *Kantorovich* proposed a method for solving LP, same time *Koopmans* formulated classical economic problems as LPs (shared Nobel-prize in 1975)
- 1945-46 *Dantzig*: general formulation for planning problems in US Air Force. Invented the **simplex method**
- 1979 – *Khachiyan*'s polynomial time algorithm
- 1984 – *Karmarkar*'s breakthrough: interior-point method for solving LP
Some real applications

- CITGO petroleum (USA): Klingman et al. (1987) used various mathematical programming models that saved around 70 million $ to the company by optimizing the operating costs and supply distribution marketing system.
- San Francisco Police Department scheduling: Taylor and Huxley (1989) – optimal patrol scheduling system, that saves more than 5 million $ annually. Other cities also adopted the system.
- GE credit card payment system: Makuch et al. (1989)
Some real applications

- CITGO petroleum (USA): Klingman et al. (1987) used various mathematical programming models that saved around 70 million $ to the company by optimizing the operating costs and supply distribution marketing system.
Some real applications

- CITGO petroleum (USA): Klingman et al. (1987) used various mathematical programming models that saved around 70 million $ to the company by optimizing the operating costs and supply distribution marketing system.

- San Francisco Police Department scheduling: Taylor and Huxley (1989) – optimal patrol scheduling system, that saves more the 5 million $ annually. Other cities also adopted the system.
Some real applications

- CITGO petroleum (USA): Klingman et al. (1987) used various mathematical programming models that saved around 70 million $ to the company by optimizing the operating costs and supply distribution marketing system.

- San Francisco Police Department scheduling: Taylor and Huxley (1989) – optimal patrol scheduling system, that saves more the 5 million $ annually. Other cities also adopted the system.

- GE credit card payment system: Makuch et al (1989)
Some real applications

- **CITGO petroleum (USA):** Klingman et al. (1987) used various mathematical programming models that saved around 70 million $ to the company by optimizing the operating costs and supply distribution marketing system.

- **San Francisco Police Department scheduling:** Taylor and Huxley (1989) – optimal patrol scheduling system, that saves more than 5 million $ annually. Other cities also adopted the system.

- **GE credit card payment system:** Makuch et al (1989)

- etc...
LP standard form

Linear program (LP) in a standard form (maximization)

\[
\begin{align*}
\text{max} & \quad c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \\
\text{subject to} & \quad a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n \leq b_1 \\
& \quad a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n \leq b_2 \\
& \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\
& \quad a_{m1} x_1 + a_{m2} x_2 + \ldots + a_{mn} x_n \leq b_m \\
& \quad x_1, x_2, \ldots, x_n \geq 0
\end{align*}
\]

Objective function

\[
\begin{align*}
\text{Constraints} & \quad \{ \text{objective function and constraints} \} \\
\text{Sign restrictions} & \quad \{ \text{objective function and constraints} \}
\end{align*}
\]

Feasible solution (point) \( P = (p_1, p_2, \ldots, p_n) \) is an assignment of values to the \( p_1, \ldots, p_n \) to variables \( x_1, \ldots, x_n \) that satisfies all constraints and all sign restrictions.

Feasible region \( \equiv \) the set of all feasible points.

Optimal solution \( \equiv \) a feasible solution with maximum value of the objective function.
Formulating a linear program

1. Choose decision variables
2. Choose an objective and an objective function – linear function in variables
3. Choose constraints – linear inequalities
4. Choose sign restrictions
Formulating a linear program

1. Choose **decision variables**
Formulating a linear program

1. Choose decision variables

2. Choose an objective and an **objective function** – linear function in variables
Formulating a linear program

1. Choose decision variables

2. Choose an objective and an objective function – linear function in variables

3. Choose constraints – linear inequalities
Formulating a linear program

1. Choose **decision variables**

2. Choose an objective and an **objective function** – linear function in variables

3. Choose **constraints** – linear inequalities

4. Choose **sign restrictions**
Product mix

A toy company makes two types of toys: toy soldiers and trains. Each toy is produced in two stages, first it is constructed in a carpentry shop, and then it is sent to a finishing shop, where it is varnished, waxed, and polished.

To make one toy soldier costs $10 for raw materials and $14 for labor; it takes 1 hour in the carpentry shop, and 2 hours for finishing.

To make one train costs $9 for raw materials and $10 for labor; it takes 1 hour in the carpentry shop, and 1 hour for finishing.

There are 80 hours available each week in the carpentry shop, and 100 hours for finishing. Each toy soldier is sold for $27 while each train for $21. Due to decreased demand for toy soldiers, the company plans to make and sell at most 40 toy soldiers; the number of trains is not restricted in any way.

What is the optimum (best) product mix (i.e., what quantities of which products to make) that maximizes the profit (assuming all toys produced will be sold)?
Product mix

A toy company makes two types of toys: toy soldiers and trains. Each toy is produced in two stages, first it is constructed in a carpentry shop, and then it is sent to a finishing shop, where it is varnished, vaxed, and polished.
Product mix

A toy company makes two types of toys: toy soldiers and trains. Each toy is produced in two stages, first it is constructed in a carpentry shop, and then it is sent to a finishing shop, where it is varnished, vaxed, and polished.

- To make one toy soldier costs $10 for raw materials and $14 for labor; it takes 1 hour in the carpentry shop, and 2 hours for finishing. To make one train costs $9 for raw materials and $10 for labor; it takes 1 hour in the carpentry shop, and 1 hour for finishing.
Product mix

A toy company makes two types of toys: toy soldiers and trains. Each toy is produced in two stages, first it is constructed in a carpentry shop, and then it is sent to a finishing shop, where it is varnished, vaxed, and polished.

- To make one toy soldier costs $10 for raw materials and $14 for labor; it takes 1 hour in the carpentry shop, and 2 hours for finishing. To make one train costs $9 for raw materials and $10 for labor; it takes 1 hour in the carpentry shop, and 1 hour for finishing.

- There are 80 hours available each week in the carpentry shop, and 100 hours for finishing. Each toy soldier is sold for $27 while each train for $21. Due to decreased demand for toy soldiers, the company plans to make and sell at most 40 toy soldiers; the number of trains is not restricted in any way.
Product mix

A toy company makes two types of toys: toy soldiers and trains. Each toy is produced in two stages, first it is constructed in a carpentry shop, and then it is sent to a finishing shop, where it is varnished, vaxed, and polished.

- To make one toy soldier costs $10 for raw materials and $14 for labor; it takes 1 hour in the carpentry shop, and 2 hours for finishing. To make one train costs $9 for raw materials and $10 for labor; it takes 1 hour in the carpentry shop, and 1 hour for finishing.

- There are 80 hours available each week in the carpentry shop, and 100 hours for finishing. Each toy soldier is sold for $27 while each train for $21. Due to decreased demand for toy soldiers, the company plans to make and sell at most 40 toy soldiers; the number of trains is not restricted in any way.

What is the optimum (best) product mix (i.e., what quantities of which products to make) that maximizes the profit (assuming all toys produced will be sold)?
Product mix: LP formulation

Decision variables:
- $x_1 =$ # of toy soldiers
- $x_2 =$ # of toy trains

Objective: maximize profit
- $27 - 10 - 14 = 3 \text{ profit for selling one toy soldier} \Rightarrow 3x_1 \text{ profit (in $) for selling } x_1 \text{ toy soldier}$
- $21 - 9 - 10 = 2 \text{ profit for selling one toy train} \Rightarrow 2x_2 \text{ profit (in $) for selling } x_2 \text{ toy train}$

$\Rightarrow z = 3x_1 + 2x_2 \text{ profit for selling } x_1 \text{ toy soldiers and } x_2 \text{ toy trains}$

Objective function

Constraints:
- producing $x_1$ toy soldiers and $x_2$ toy trains requires
  (a) $1x_1 + 1x_2$ hours in the carpentry shop; there are 80 hours available
  (b) $2x_1 + 1x_2$ hours in the finishing shop; there are 100 hours available
- the number $x_1$ of toy soldiers produced should be at most 40

Variable domains: the numbers $x_1$, $x_2$ of toy soldiers and trains must be non-negative (sign restriction)

Max $3x_1 + 2x_2$

\[
\begin{align*}
3x_1 + 2x_2 & \leq 80 \\
2x_1 + x_2 & \leq 100 \\
x_1 & \leq 40 \\
x_1, x_2 & \geq 0
\end{align*}
\]
A more complicated example

You have $100. You can make the following three types of investments:

**Investment A.** Every dollar invested now yields $0.10 a year from now, and $1.30 three years from now.

**Investment B.** Every dollar invested now yields $0.20 a year from now and $1.10 two years from now.

**Investment C.** Every dollar invested a year from now yields $1.50 three years from now.

During each year leftover cash can be placed into money markets which yield 6% a year. The most that can be invested a single investment (A, B, or C) is $50.

Formulate an LP to maximize the available cash three years from now.
Example: LP formulation

Decision variables: $x_A$, $x_B$, $x_C$, amounts invested into Investments A, B, C, respectively.
$y_0$, $y_1$, $y_2$, $y_3$ cash available/invested into money markets now, and in 1,2,3 years.

Max $y_3$

s.t.

$0.1x_A + 0.2x_B - x_C + 1.06y_0 = 100$
$1.1x_B + 1.5x_C + 1.06y_2 = y_3$
$1.3x_A + x_B \leq 50$
$x_A, x_B, x_C, y_0, y_1, y_2, y_3 \geq 0$
Example: let us see what is going on

<table>
<thead>
<tr>
<th>Items</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Inv. A</td>
<td>Inv. B</td>
<td>Inv. C</td>
<td>Markets Now</td>
<td>Markets Year 1</td>
<td>Markets Year 2</td>
<td>External flow</td>
</tr>
<tr>
<td>Now</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>=</td>
<td>−100</td>
<td></td>
</tr>
<tr>
<td>Year 1</td>
<td>0.1</td>
<td>0.2</td>
<td>−1</td>
<td>1.06</td>
<td>−1</td>
<td>=</td>
</tr>
<tr>
<td>Year 2</td>
<td></td>
<td>1.1</td>
<td>1.06</td>
<td>−1</td>
<td>=</td>
<td>0</td>
</tr>
<tr>
<td>Year 3</td>
<td>1.3</td>
<td>1.5</td>
<td>1.06</td>
<td></td>
<td>maximize</td>
<td></td>
</tr>
</tbody>
</table>

**Sign convention:** inputs have **negative** sign, outputs have **positive** signs.

External in-flow has **negative** sign, external out-flow has **positive** sign.

We have in-flow of $100 cash “Now” which means we have $−100 on the right-hand side. No in-flow or out-flow of any other item.
Example: let us see what is going on

\[
\text{Max} \quad 1.3x_A + 1.5x_C + 1.06y_2 = 100 \\
\text{s.t.} \quad x_A + x_B + y_0 = 0 \\
0.1x_A + 0.2x_B - x_C + 1.06y_0 - y_1 = 0 \\
1.1x_B + 1.06y_1 - y_2 = 0 \\
y_0, y_1, y_2 \geq 0 \\
0 \leq x_A, x_B, x_C \leq 50
\]
Product mix: graphical method

1. Find the feasible region.

- Plot each constraint as an equation ≡ line in the plane
- Feasible points on one side of the line – plug in (0,0) to find out which

Start with $x_1 \geq 0$ and $x_2 \geq 0$

add $x_1 + x_2 \leq 80$
Product mix: graphical method

2x₁ + x₂ ≤ 100
x₁ + x₂ ≤ 80
add 2x₁ + x₂ ≤ 100

2x₁ + x₂ ≤ 100
x₁ ≤ 40
feasible region
add x₁ ≤ 40
Motivation: why LP?

Linear Programming

Solving linear programs

LP and convex geometry

Graphical method

A corner (extreme) point \( X \) of the region \( R \equiv \) every line through \( X \) intersects \( R \) in a segment whose one endpoint is \( X \). Solving a linear program amounts to finding a best corner point by the following theorem.

**Theorem 1.** If a linear program has an optimal solution, then it also has an optimal solution that is a corner point of the feasible region.

**Exercise.** Try to find all corner points. Evaluate the objective function \( 3x_1 + 2x_2 \) at those points.

**Theorem 2.** Every linear program has either

1. a unique optimal solution,
2. multiple (infinity) optimal solutions,
3. is infeasible (has no feasible solution), or
4. is unbounded (no feasible solution is maximal).
Graphical method

A corner (extreme) point $X$ of the region $R \equiv$ every line through $X$ intersects $R$ in a segment whose one endpoint is $X$. Solving a linear program amounts to finding a best corner point by the following theorem.
Graphical method

A **corner (extreme) point** $X$ of the region $R \equiv$ every line through $X$ intersects $R$ in a segment whose one endpoint is $X$. Solving a linear program amounts to finding a best corner point by the following theorem.

**Theorem 1.** *If a linear program has an optimal solution, then it also has an optimal solution that is a corner point of the feasible region.*
A **corner (extreme) point** $X$ of the region $R ≡$ every line through $X$ intersects $R$ in a segment whose one endpoint is $X$. Solving a linear program amounts to finding a best corner point by the following theorem.

**Theorem 1.** *If a linear program has an optimal solution, then it also has an optimal solution that is a corner point of the feasible region.*

*Exercise.* Try to find all corner points. Evaluate the objective function $3x_1 + 2x_2$ at those points.
A **corner (extreme) point** $X$ of the region $R \equiv$ every line through $X$ intersects $R$ in a segment whose one endpoint is $X$. Solving a linear program amounts to finding a best corner point by the following theorem.

**Theorem 1.** *If a linear program has an optimal solution, then it also has an optimal solution that is a corner point of the feasible region.*

*Exercise.* Try to find all corner points. Evaluate the objective function $3x_1 + 2x_2$ at those points.

**Theorem 2.** *Every linear program has either*

1. a **unique** optimal solution, or
2. multiple (**infinity**) optimal solutions, or
3. is **infeasible** (has no feasible solution), or
4. is **unbounded** (no feasible solution is maximal).
Motivation: why LP?

Linear Programming

Solving linear programs

LP and convex geometry

In higher dimensions...

$\mathbb{R}^n$: $n$-dimensional linear space over the real numbers – elements: real vectors of $n$ elements

$E^n$: $n$-dimensional Euclidean space, with an inner product operation and a distance function are defined as follows

$$\langle x, y \rangle = x^T y = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n,$$

$$||x|| = \sqrt{\langle x, x \rangle}$$

This distance function is called the Euclidean metric. This formula expresses a special case of the Pythagorean theorem.
In higher dimensions...

\( \mathbb{R}^n \): \( n \)-dimensional **linear space** over the real numbers – elements: real **vectors** of \( n \) elements
In higher dimensions...

\( \mathbb{R}^n \): \( n \)-dimensional **linear space** over the real numbers – elements: real **vectors** of \( n \) elements

\( \mathbb{E}^n \): \( n \)-dimensional **Euclidean space**, with an **inner product** operation and a **distance** function are defined as follows

- \( \langle x, y \rangle = x^T y = x_1y_1 + x_2y_2 + \ldots + x_ny_n \), \( ||x|| = \sqrt{\langle x, x \rangle} \) **norm**
- \( d(x, y) = ||x - y||_2 = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2} \)
In higher dimensions...

\( \mathbb{R}^n \): \( n \)-dimensional **linear space** over the real numbers – elements: real **vectors** of \( n \) elements

\( \mathbb{E}^n \): \( n \)-dimensional **Euclidean space**, with an **inner product** operation and a **distance** function are defined as follows

- \[ \langle x, y \rangle = x^T y = x_1y_1 + x_2y_2 + \ldots + x_ny_n, \quad ||x|| = \sqrt{\langle x, x \rangle} \text{ norm} \]
- \[ d(x, y) = ||x - y||_2 = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2} \]

This distance function is called the **Euclidean metric**. This formula expresses a special case of the **Pythagorean theorem**.
In higher dimensions...
In higher dimensions...

**Point:** an $x \in E^n$ vector
In higher dimensions...

**Point:** an \( x \in E^n \) vector

**LP feasible solutions** \( \leftrightarrow \) points in \( E^n \).
In higher dimensions...

**Point:** an \( x \in \mathbb{E}^n \) vector

**LP feasible solutions** \( \leftrightarrow \) points in \( \mathbb{E}^n \).

**n-dimensional hyperplane:**

\[
\{ \ x : \ x \in \mathbb{E}^n, \ a_1 x_1 + a_2 x_2 + \ldots + a_n x_n = b \ \},
\]

where \( a_1, a_2, \ldots, a_n, b \in \mathbb{R} \) given (fixed) numbers
In higher dimensions...

**Point**: an \( x \in E^n \) vector

**LP feasible solutions** ↔ points in \( E^n \).

**\( n \)-dimensional hyperplane**:

\[
\{ \ x \ : \ x \in E^n, \ a_1 x_1 + a_2 x_2 + \ldots + a_n x_n = b \ \}
\]

where \( a_1, a_2, \ldots, a_n, b \in \mathbb{R} \) given (fixed) numbers

**\( n \)-dimensional closed half-space**:

\[
\{ \ x \ : \ x \in E^n, \ a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \leq b \ \}
\]

where \( a_1, a_2, \ldots, a_n, b \in \mathbb{R} \) given (fixed) numbers
In higher dimensions...

**Point:** an \( x \in E^n \) vector

**LP feasible solutions** \( \leftrightarrow \) points in \( E^n \).

**\( n \)-dimensional hyperplane:**

\[
\{ x : x \in E^n, a_1 x_1 + a_2 x_2 + \ldots + a_n x_n = b \},
\]

where \( a_1, a_2, \ldots, a_n, b \in \mathbb{R} \) given (fixed) numbers

**\( n \)-dimensional closed half-space:**

\[
\{ x : x \in E^n, a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \leq b \},
\]

where \( a_1, a_2, \ldots, a_n, b \in \mathbb{R} \) given (fixed) numbers

**linear constraints** \( \leftrightarrow \) closed half-spaces (‘\( \leq \)’) and hyperplanes (‘=’).
In higher dimensions...

**Point**: an \( x \in E^n \) vector

**LP feasible solutions** ↔ points in \( E^n \).

**\( n \)-dimensional hyperplane**:

\[
\{ x : x \in E^n, a_1 x_1 + a_2 x_2 + \ldots + a_n x_n = b \},
\]

where \( a_1, a_2, \ldots, a_n, b \in \mathbb{R} \) given (fixed) numbers

**\( n \)-dimensional closed half-space**:

\[
\{ x : x \in E^n, a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \leq b \},
\]

where \( a_1, a_2, \ldots, a_n, b \in \mathbb{R} \) given (fixed) numbers

**linear constraints** ↔ closed half-spaces (‘\( \leq \)’) and hyperplanes (‘\( = \)’)

**Feasible region** ↔ Intersection of half-spaces (and hyperplanes)
In higher dimensions...

Polytope: bounded intersection of a finite set of half-spaces

The set of feasible solutions (points) of a linear program forms a convex polytope (bounded or unbounded)

Theorem 1. tells us that a linear objective function achieves its maximal value (if exists) is a corner (extreme) point of the feasible region (i.e. polytope).

⇒ Simplex algorithm (see Lecture 2)
In higher dimensions...

**Polytope**: bounded intersection of a finite set of half-spaces
In higher dimensions...

**Polytope**: bounded intersection of a finite set of half-spaces

The set of feasible solutions (points) of a linear program forms a convex polytope (bounded or unbounded)
Polytope: bounded intersection of a finite set of half-spaces

The set of feasible solutions (points) of a linear program forms a convex polytope (bounded or unbounded)

Theorem 1. tells us that a linear objective function achieves its maximal value (if exists) is a corner (extreme) point of the feasible region (i.e. polytope).
Polytope: bounded intersection of a finite set of half-spaces

The set of feasible solutions (points) of a linear program forms a convex polytope (bounded or unbounded)

Theorem 1. tells us that a linear objective function achieves its maximal value (if exists) is a corner (extreme) point of the feasible region (i.e. polytope).

⇒ Simplex algorithm (see Lecture 2)