# Advanced Optimization

Boglárka G.-Tóth and Tamás Vinkó

Autumn of 2021

Dept of Computational Optimization
University of Szeged

these slides are adapted from:
B6.3 Integer Programming course by Raphael Hauser at University Of Oxford

# What is integer programming?

# What is integer programming?

*Integer Programming* concerns the mathematical analysis of and design of algorithms for optimisation problems of the following forms.

- (Linear) Integer Program:

$$
\begin{aligned}
\max_{\mathbf{x}\in\mathbb{R}^n} \ & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\
\text{s.t. } & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \quad \text{(componentwise)} \\
& \mathbf{x} \geq 0, \quad \text{(componentwise)} \\
& \mathbf{x} \in \mathbb{Z}^n,
\end{aligned}
$$

  where $\mathbf{A} \in \mathbb{R}^{m\times n}$ is a matrix and $\mathbf{b} \in \mathbb{R}^m, \mathbf{c} \in \mathbb{R}^n$ are vectors with rational coefficients.

- Binary (Linear) Integer Program:

$$\begin{aligned} \max_{\mathbf{x}} \ & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{s.t. } & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \in \mathbb{B}^n := \{0,1\}^n. \end{aligned}$$

This is a special case of a linear integer program, as it can be reformulated as

$$\begin{aligned} \max_{\mathbf{x}} \ & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{s.t. } & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \leq \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \\ & \mathbf{x} \geq 0, \\ & \mathbf{x} \in \mathbb{Z}^n. \end{aligned}$$

- (Linear) Mixed Integer Program:

$$\begin{aligned} \max_{\mathbf{x},\mathbf{y}} \ & \mathbf{c}^{\mathrm{T}}\mathbf{x} + \mathbf{h}^{\mathrm{T}}\mathbf{y} \\ \text{s.t.} \ & \mathbf{A}\mathbf{x} + \mathbf{G}\mathbf{y} \leq \mathbf{b} \\ & \mathbf{x}, \mathbf{y} \geq 0, \\ & \mathbf{y} \in \mathbb{Z}^p, \end{aligned}$$

where **G** is a matrix and **h** a vector with rational coefficients.

# Introductory Examples

# Introductory Examples

**Example (The Assignment Problem)**

- $n$ people are to carry out $n$ jobs,
- each person carries out exactly one job,
- assigning person $i$ to job $j$ incurs a cost $c_{ij}$,
- find assignment that minimises the total cost.

Decision variables:

For $(i, j \in [1, n] := \{1, \ldots, n\})$,

$$x_{ij} = \begin{cases} 1 & \text{if person } i \text{ assigned to carry out job } j, \\ 0 & \text{otherwise.} \end{cases}$$

Constraints:

- Each person does exactly one job:

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad (i \in [1, n])$$

- Each job is done by exactly one person:

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad (j \in [1, n])$$

- Variables are binary:

$$x_{ij} \in \mathbb{B} := \{0, 1\}, \qquad (i, j \in [1, n]).$$

Objective function: the total cost $\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$.

Model:

$$
\begin{aligned}
\min_{x \in \mathbb{R}^{n \times n}} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} = 1 \quad \text{for } i = 1, \ldots, n, \\
& \sum_{i=1}^{n} x_{ij} = 1 \quad \text{for } j = 1, \ldots, n, \\
& x_{ij} \in \mathbb{B} \quad \text{for } i, j = 1, \ldots, n.
\end{aligned}
$$

**Example (The 0-1 Knapsack Problem)**

- A knapsack of volume $b$ has to be packed with a selection of $n$ items,
- item $i$ has volume $a_i$ and value $c_i$,
- pack the knapsack with a set of items of maximal total value.

Knapsack model:

$$\begin{aligned} \max \ & \sum_{i=1}^{n} c_i x_i \\ \text{s.t.} \ & \sum_{i=1}^{n} a_i x_i \leq b, \\ & x \in \mathbb{B}^n, \end{aligned}$$

with decision variables are defined as follows,

$$x_i = \begin{cases} 1 & \text{if item } i \text{ is selected,} \\ 0 & \text{otherwise.} \end{cases}$$

**Example (The Integer Knapsack Problem)**

The same as Example 2, but multiple copies of each type of item are available.

Integer knapsack model:

$$\begin{aligned} \max \ & \sum_{i=1}^{n} c_i x_i \\ \text{s.t.} \ & \sum_{i=1}^{n} a_i x_i \leq b, \\ & x \geq 0, \\ & x \in \mathbb{Z}^n, \end{aligned}$$

# Linear Programming

# Linear Programming

An important special case of an integer programming problem is one without integrality constraints, e.g.,

$$\begin{aligned} \max_{\mathbf{x}} \ & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \geq 0. \end{aligned}$$

Such problems are called *linear programming* problems (or LPs).

We will see that LPs play an important role in algorithms designed to solve general IPs through the concept of of *LP relaxation*:

Consider the IP problem

$$
\begin{aligned}
\text{(IP)} \quad z^* = \max_{\mathbf{x}} \; & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\
\text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b}, \\
& \mathbf{x} \in \mathbb{Z}_+^n.
\end{aligned}
$$

If we give up on the integrality constraints $x_i \in \mathbb{Z}$, we obtain an LP,

$$
\begin{aligned}
\text{(LP)} \quad \bar{z} = \max_{\mathbf{x}} \; & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\
\text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b}, \\
& \mathbf{x} \ge 0.
\end{aligned}
$$

Giving up on the integrality constraints has two effects on the feasible set $\mathscr{F}$ (the set of decision vectors $\mathbf{x}$ that satisfy the constraints of the problem)

- $\mathscr{F}$ becomes larger,
- $\mathscr{F}$ becomes convex.

**Proposition**

*The consequence of the first effect is that* $\bar{z} \geq z^*$.

**Proof.**

If the optimal objective value $z^*$ of (IP) is achieved at the point $\mathbf{x}^*$, then $\mathbf{x}^*$ is feasible for (IP), and hence it is also feasible for (LP). Therefore,

$$\bar{z} \geq \mathbf{c}^\mathrm{T}\mathbf{x} = z^*.$$

□

As we shall learn, the consequence of the second effect is that it is much easier to solve the problem (LP) than (IP).

A first idea for solving IPs is to solve the LP relaxation and round the optimal values of the decision variables to the nearest feasible integer valued feasible solution.

While this occasionally works, it is not always a good idea:

- Rounding may be non-trivial, e.g., when the LP relaxation of a binary program takes an optimal solution $\mathbf{x}^*$ with many values near 0.5.
- The rounded solution may be far from optimal.
- The rounded solution may be infeasible.

# The Simplex Algorithm

# The Simplex Algorithm in Dictionary Form

We will now discuss an algorithm for solving general linear programming problems.

**Example**

Consider the LP instance

$$\begin{aligned} z = \max_x\ & 5x_1 + 4x_2 + 3x_3 \\ \text{s.t.}\quad & 2x_1 + 3x_2 + x_3 \leq 5 \\ & 4x_1 + x_2 + 2x_3 \leq 11 \\ & 3x_1 + 4x_2 + 2x_3 \leq 8 \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

Preliminary step I: introduce *slack variables* $x_4, x_5, x_6 \geq 0$ to reformulate inequality constraints as a system of linear equations,

$$\begin{aligned} z = \max\ & 5x_1 + 4x_2 + 3x_3 + 0x_4 + 0x_5 + 0x_6 \\ \text{s.t.}\quad & 2x_1 + 3x_2 + x_3 + x_4 = 5 \\ & 4x_1 + x_2 + 2x_3 + x_5 = 11 \\ & 3x_1 + 4x_2 + 2x_3 + x_6 = 8 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0. \end{aligned}$$

Preliminary step II: express in *dictionary form*

$$\max z \quad \text{s.t.} \quad x_1, \ldots, x_6 \geq 0,$$

and where the variables are linked via the linear system

$$\begin{aligned} x_4 &= 5 - 2x_1 - 3x_2 - x_3 \\ x_5 &= 11 - 4x_1 - x_2 - 2x_3 \\ x_6 &= 8 - 3x_1 - 4x_2 - 2x_3 \\ z &= 0 + 5x_1 + 4x_2 + 3x_3. \end{aligned}$$

Step 0: $x_1, x_2, x_3 = 0$, $x_4 = 5$, $x_5 = 11$, $x_6 = 8$ is an initial feasible solution. $x_1, x_2, x_3$ are called the *nonbasic variables* and $x_4, x_5, x_6$ *basic variables*.

Note that basic variables are expressed in terms of nonbasic ones!

$$
\begin{aligned}
x_4 &= 5 - 2x_1 - 3x_2 - x_3 \\
x_5 &= 11 - 4x_1 - x_2 - 2x_3 \\
x_6 &= 8 - 3x_1 - 4x_2 - 2x_3 \\
z &= 0 + 5x_1 + 4x_2 + 3x_3.
\end{aligned}
$$

Step 1: We note that as long as $x_1$ is increased by at most

$$\frac{5}{2} = \min\left(\frac{5}{2}, \frac{11}{4}, \frac{8}{3}\right),$$

all $x_i$ remain nonnegative, but $z$ increases.

Setting $x_1 = 5/2$ and substituting into the dictionary, we find $x_2, x_3, x_4 = 0$, $x_5 = 1$, $x_6 = 1/2$, $z = 25/2$ as an improved feasible solution.

We call $x_1$ the *pivot* of the iteration.

$$
\begin{aligned}
x_4 &= 5 - 2x_1 - 3x_2 - x_3 \\
x_5 &= 11 - 4x_1 - x_2 - 2x_3 \\
x_6 &= 8 - 3x_1 - 4x_2 - 2x_3 \\
z &= 0 + 5x_1 + 4x_2 + 3x_3.
\end{aligned}
$$

We can now express the variables $x_1, x_5, x_6, z$ in terms of the *new nonbasic variables* $x_2, x_3, x_4$ (those currently set to zero) to obtain a new dictionary.

To do this, use line 1 of the dictionary to express $x_1$ in terms of $x_2, x_3, x_4$,

$$
x_1 = \frac{1}{2}\left(5 - 3x_2 - x_3 - x_4\right)
$$

and substitute the right hand side for $x_1$ in the remaining equations.

The new dictionary then looks as follows,

$$x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \tag{1}$$

$$x_5 = 1 + 5x_2 + 2x_4 \tag{2}$$

$$x_6 = \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \tag{3}$$

$$z = \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4. \tag{4}$$

Of course, we are still solving

$$\max z \quad \text{s.t.} \quad x_1, \ldots, x_6 \geq 0,$$

subject to the relationships (1)–(4) holding between the variables, and the new LP instance is equivalent to the old one.

However, a better feasible solution can be read off the new dictionary by setting the nonbasic variables to zero!

$$
\begin{aligned}
x_1 &= \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\
x_5 &= 1 + 5x_2 + 2x_4 \\
x_6 &= \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\
z &= \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4.
\end{aligned}
$$

Step 2: We continue in the same vein: increasing the value of $x_2$ or $x_4$ is useless, as this would decrease the objective value $z$.

Thus, $x_3$ is our pivot, and we can increase its value up to

$$1 = \min(5, +\infty, 1),$$

leading to the improved solution $x_2, x_4, x_6 = 0$, $x_1 = 2$, $x_3 = 1$, $x_5 = 1$, $z = 13$ and the dictionary corresponding to $x_2, x_4, x_6$ as nonbasic variables:

$$
\begin{aligned}
x_3 &= 1 + x_2 + 3x_4 - 2x_6 \\
x_1 &= 2 - 2x_2 - 2x_4 + x_6 \\
x_5 &= 1 + 5x_2 + 2x_4 \\
z &= 13 - 3x_2 - x_4 - x_6.
\end{aligned}
$$

At this point we can stop the algorithm for the following reasons:

- from the last line of the dictionary we see that for any strictly positive value of $x_2, x_4$ or $x_6$ the objective value $z$ is necessarily strictly smaller than 13,
- and from the other lines of the dictionary we see that as soon as the values of $x_2, x_4$ and $x_6$ are fixed, the values of $x_3, x_1$ and $x_5$ are fixed too.
- Thus, the last dictionary yields a certificate of optimality for the identified solution.

# Direct Computation of Dictionaries

Let us now try to understand how the dictionary

$$
\begin{aligned}
x_3 &= 1 + x_2 + 3x_4 - 2x_6 \\
x_1 &= 2 - 2x_2 - 2x_4 + x_6 \\
x_5 &= 1 + 5x_2 + 2x_4 \\
z &= 13 - 3x_2 - x_4 - x_6,
\end{aligned} \tag{5}
$$

(which was obtained after two pivoting steps) could have been obtained directly from the input data of the original LP instance

$$
\begin{aligned}
\text{(LPI)} \quad \max\, & 5x_1 + 4x_2 + 3x_3 + 0x_4 + 0x_5 + 0x_6 \\
\text{s.t. } & 2x_1 + 3x_2 + x_3 + x_4 = 5 \\
& 4x_1 + x_2 + 2x_3 + x_5 = 11 \\
& 3x_1 + 4x_2 + 2x_3 + x_6 = 8 \\
& x_1, x_2, x_3, x_4, x_5, x_6 \geq 0
\end{aligned}
$$

The constraints of (LPI) imply a functional dependence between the nonnegative decision variables $x_i$, expressed by the linear system

$$Ax = b, \tag{6}$$

where

$$A = \begin{bmatrix} 2 & 3 & 1 & 1 & 0 & 0 \\ 4 & 1 & 2 & 0 & 1 & 0 \\ 3 & 4 & 2 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 5 \\ 11 \\ 8 \end{bmatrix}.$$

The basic variables of dictionary (5) are $x_3, x_1, x_5$. Writing

$$x_B := \begin{bmatrix} x_3 & x_1 & x_5 \end{bmatrix}^{\mathrm{T}}, \qquad x_N := \begin{bmatrix} x_2 & x_4 & x_6 \end{bmatrix}^{\mathrm{T}}$$

$$A_B := \begin{bmatrix} 1 & 2 & 0 \\ 2 & 4 & 1 \\ 2 & 3 & 0 \end{bmatrix}, \qquad A_N := \begin{bmatrix} 3 & 1 & 0 \\ 1 & 0 & 0 \\ 4 & 0 & 1 \end{bmatrix}$$

(6) can be written as $A_B\, x_B + A_N\, x_N = b$.

$$A_B\, x_B + A_N\, x_N = b.$$

Solving for the basic variables $x_B$, we obtain

$$x_B = A_B^{-1}\left(b - A_N x_N\right). \tag{7}$$

Likewise, the objective function can be written as

$$z = c_B^{\mathrm{T}}\, x_B + c_N^{\mathrm{T}}\, x_N,$$

where

$$c_B = \begin{bmatrix} 3 & 5 & 0 \end{bmatrix}^{\mathrm{T}}, \quad c_N = \begin{bmatrix} 4 & 0 & 0 \end{bmatrix}^{\mathrm{T}},$$

and substituting from (7), we find

$$z = c_B^{\mathrm{T}} A_B^{-1} b + \left(c_N^{\mathrm{T}} - c_B^{\mathrm{T}} A_B^{-1} A_N\right) x_N.$$

Dictionary (5) is now just the system of equations

$$x_B = A_B^{-1} b - A_B^{-1} A_N x_N, \qquad z = c_B^{\mathrm{T}} A_B^{-1} b + \left(c_N^{\mathrm{T}} - c_B^{\mathrm{T}} A_B^{-1} A_N\right) x_N.$$

## Definition

A *dictionary* of the LP problem (P) $\max_x\{c^{\mathrm{T}}x : \ Ax = b,\ x \geq 0\}$ is a system of equations

$$\begin{aligned} x_B &= A_B^{-1}b - A_B^{-1}A_N x_N, \\ z &= c_B^{\mathrm{T}} A_B^{-1} b + \left(c_N^{\mathrm{T}} - c_B^{\mathrm{T}} A_B^{-1} A_N\right) x_N, \end{aligned}$$

equivalent to

$$\begin{aligned} Ax &= b, \\ z &= c^{\mathrm{T}} x, \end{aligned}$$

where up to column perturbation $A = \left[\, A_B \ A_N \,\right]$ and $x = \left[\, x_B^{\mathrm{T}} \ x_N^{\mathrm{T}} \,\right]^{\mathrm{T}}$ is a block decomposition such that $A_B$ is nonsingular.

A dictionary is called *feasible* if $A_B^{-1}b \geq 0$, so that $x = (x_B, x_N) = (A_B^{-1}b, 0)$ is a feasible (but generally suboptimal) solution. $(x_B, x_N)$ is then called a *basic feasible solution*.

# Linear Programming Duality

# LP Duality

Let us again consider the LP instance we studied previously,

$$
\begin{aligned}
\text{(P)} \quad \max\ & 5x_1 + 4x_2 + 3x_3 \\
\text{s.t. } & 2x_1 + 3x_2 + x_3 \leq 5 \\
& 4x_1 + x_2 + 2x_3 \leq 11 \\
& 3x_1 + 4x_2 + 2x_3 \leq 8 \\
& x_1, x_2, x_3 \geq 0.
\end{aligned}
$$

We saw that the optimal value is 13.

In integer programming, instead of solving an LP relaxation to optimality one is often interested in finding merely upper and lower bounds on the optimal value.

A lower bound is provided by any feasible solution. For example, $x_1, x_2 = 1$, $x_3 = 0$ is feasible with objective value 9.

How can we obtain upper bounds?

Multiplying the first constraint by 3 we obtain

$$6x_1 + 9x_2 + 3x_3 \leq 15,$$

and since $x_1, x_2, x_3 \geq 0$, this yields an upper bound on the objective function:

$$z = 5x_1 + 4x_2 + 3x_3 \leq 6x_1 + 9x_2 + 3x_3 \leq 15,$$

Likewise, taking the sum of the first two constraints yields the valid upper bound

$$z = 5x_1 + 4x_2 + 3x_3 \leq 6x_1 + 4x_2 + 3x_3 \leq 16.$$

More generally, such bounds can be obtained from any sum of positive multiples of the constraints for which the resulting coefficients are no smaller than the corresponding coefficients of the objective function:

$$\begin{bmatrix}5 & 4 & 3\end{bmatrix} \leq \begin{bmatrix}y_1 & y_2 & y_3\end{bmatrix} \begin{bmatrix}2 & 3 & 1 \\ 4 & 1 & 2 \\ 3 & 4 & 2\end{bmatrix}, \quad y_1, y_2, y_3 \geq 0$$

$$\Rightarrow z \leq \begin{bmatrix}y_1 & y_2 & y_3\end{bmatrix} \begin{bmatrix}5 \\ 11 \\ 8\end{bmatrix}.$$

The best such upper bound is obtained by solving the LP instance

$$
\begin{aligned}
\text{(D)} \quad \min_{y}\ & 5y_1 + 11y_2 + 8y_3 \\
\text{s.t. } & 2y_1 + 4y_2 + 3y_3 \geq 5, \\
& 3y_1 + y_2 + 4y_3 \geq 4, \\
& y_1 + 2y_2 + 2y_3 \geq 3, \\
& y_1, y_2, y_3 \geq 0.
\end{aligned}
$$

This is called the *dual* of the LP instance (P), the latter being called the *primal*.

More generally, an LP of the form

$$
\begin{aligned}
\text{(P)} \quad z^* = \max_{(x,s)} \; & c^{\mathrm{T}}x + d^{\mathrm{T}}s \\
\text{s.t. } & Ax + Cs \leq a, \\
& Bx + Ds = b, \\
& x \geq 0, \\
& s \text{ arbitrary}
\end{aligned}
$$

is associated with a dual

$$
\begin{aligned}
\text{(D)} \quad w^* = \min_{(y,t)} \; & a^{\mathrm{T}}y + b^{\mathrm{T}}t \\
\text{s.t. } & A^{\mathrm{T}}y + B^{\mathrm{T}}t \geq c \\
& C^{\mathrm{T}}y + D^{\mathrm{T}}t = d \\
& y \geq 0, \\
& t \text{ arbitrary.}
\end{aligned}
$$

(D) has itself a dual: casting (D) in primal form,

$$
\begin{aligned}
\text{(D')} \quad \max_{(y,t)} \; & -a^{\mathrm{T}}y - b^{\mathrm{T}}t \\
\text{s.t.} \; & -A^{\mathrm{T}}y - B^{\mathrm{T}}t \leq -c \\
& -C^{\mathrm{T}}y - D^{\mathrm{T}}t = -d \\
& y \geq 0,
\end{aligned}
$$

the bi-dual is found to be

$$
\begin{aligned}
\text{(P')} \quad \min_{(x,s)} \; & -c^{\mathrm{T}}x - d^{\mathrm{T}}s \\
\text{s.t.} \; & -Ax - Cs \geq -a, \\
& -Bx - Ds = -b, \\
& x \geq 0,
\end{aligned}
$$

which is just the primal cast in dual form.

# Duality Theorems

# Duality Theorems

To analyse the relationship between the primal-dual pair (P), (D), we will henceforth consider LPs in the following *standard form* into which any LP may be cast under an appropriate reformulation,

$$
\begin{aligned}
\text{(P)} \quad \max_{x} \; & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \; & \sum_{j=1}^{n} a_{ij} x_j \le b_i, \; (i = 1, \ldots, m),
\end{aligned}
$$

$$
\begin{aligned}
\text{(D)} \quad \min_{y} \; & \sum_{i=1}^{m} y_i b_i \\
\text{s.t.} \; & \sum_{i=1}^{m} y_i a_{ij} = c_j, \; (j = 1, \ldots, n) \\
& y_i \ge 0, \; (i = 1, \ldots, m).
\end{aligned}
$$

**Theorem (Weak Duality Theorem)**

i) *If $x$ is primal feasible and $y$ is dual feasible (feasible for (P), (D) respectively), then*

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} y_i b_i. \tag{8}$$

ii) *If equality holds in* (8)*, then $x$ is primal optimal and $y$ is dual optimal.*

iii) *If either (P) or (D) is unbounded, then the other programme is infeasible (has no feasible solutions).*

**Theorem (Strong Duality Theorem)**

i) *If (P) and (D) both have feasible solutions, then they have optimal solutions x and y such that* $\sum_{j=1}^{n} c_j x_j = \sum_{i=1}^{m} y_i b_i$.

ii) *If either (P) or (D) is infeasible, then the other programme either unbounded or infeasible.*

# Polyhedra and Polytopes

# Polyhedra and Polytopes

**Definition**

A *polyhedron* is a set $\mathcal{P} \subset \mathbb{R}^n$ described as an intersection of finitely many affine half spaces

$$\mathcal{P} = \left\{ x \in \mathbb{R}^n : \sum_{j=1}^{n} a_{ij} x_j \leq b_i,\ (i = 1, \dots, m) \right\},$$

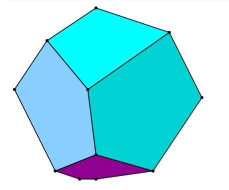
for some $A = (a_{ij}) \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.

## Definition

A *polytope* is a set $\mathcal{P}' \subset \mathbb{R}^n$ described as the convex hull of finitely many points

$$\mathcal{P}' = \operatorname{conv}\left\{x^k : k \in [1,p]\right\} := \left\{\sum_{k=1}^{p} \lambda_k x^k : \sum_{k=1}^{p} \lambda_k = 1,\ \lambda_k \geq 0,\ \forall k\right\}$$

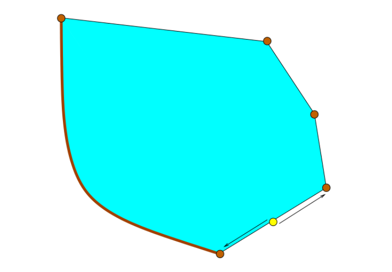
for some $x^1, \ldots, x^p \in \mathbb{R}^n$.



## Definition

Let $C \subset \mathbb{R}^n$ be a convex set. A point $x \in C$ is an *extreme point* of $C$ if $x$ is not a convex combination of two points in $C$ distinct from $x$

$$\nexists\, x_1, x_2 \in C \setminus \{x\}, \lambda \in (0,1) \text{ s.t. } x = \lambda x_1 + (1-\lambda) x_2.$$



**Theorem (Minkowski's Theorem)**

*If $\mathcal{P} \subset \mathbb{R}^n$ is a polyhedron and bounded, then $\mathcal{P}$ is a polytope, that is, $\mathcal{P}$ has a finite set $X$ of extreme points, and $\mathcal{P} = \operatorname{conv}(X)$.*

# Example of Branch & Bound in Action

# Example of Branch & Bound in Action

from page 5 of

`http://www.inf.u-szeged.hu/~london/Linprog/linprog4handout.pdf`