# Information on Optimization (Discrete Optimization - Nonlinear Optimization)

Edoardo Amaldi

DEIB – Politecnico di Milano
edoardo.amaldi@polimi.it

Course material on WeBeep "2023-24 - Optimization"

A.A. 2023-24

**Course's aim**: Present the main concepts and methods of discrete and nonlinear optimization, covering also modeling and application aspects.

Link to detailed program

"Discrete Optimization" and "Nonlinear Optimization" (5 credits) correspond to two overlapping parts of "Optimization" (8 credits).

- Discrete Optimization includes Chapters 1-3, the exercise sets n. 1-5, the computer labs n. 1-3, including a brief review of ~~AMPL~~/Python basics.

- Nonlinear Optimization includes Chapters 1, 2, 4 and 5, the exercise sets 1, 6-9, the computer labs 4-6, including a brief review ~~MATLAB~~/Python basics.

## Prerequisites

but there are recalls on the slides, if we need to catch up with these arguments

For Discrete Optimization part:

- linear programming (simplex algorithm, LP duality)
- graph optimization (minimum spanning tree, maximum flow)
- basics of integer linear programming (Branch and Bound, Gomory cuts)
- basics of Python/AMPL modeling language

For Nonlinear Optimization part: basics of Python.

## Schedule

- Monday 13.15 - 15.15 Room B.4.4
- Thursday 13.15 - 15.15 Room B.2.4
- Friday 13.15 - 16.15 (L + Ex/Lab) Room B.4.4

Lectures (L), exercises (E) and computer laboratory (Lab) sessions.

## Computer laboratory sessions

- Discrete Optimization part: one hour on AMPL/Python, 3 two-hour meetings using AMPL/Python
- Nonlinear Optimization part: one hour on MATLAB/Python (Optimization toolbox), 3 two-hour meetings using MATLAB/Python.

## Instructors

- Lectures:
  - Edoardo Amaldi `edoardo.amaldi@polimi.it`
- Exercises:
  - Marta Pascoal `marta.brazpascoal@polimi.it`
- Computer labs:
  - Maximiliano Cubillos `maximiliano.cubillos@polimi.it`

## Teaching material

- Material for the lectures, exercises and computer labs made available progressively on WeBeep.
- List of references in the course program.

## Evaluation

Written exam covering all the material presented in the lectures and the meetings devoted to the exercises and the computer labs.

For students enrolled in D.O. or N.O., the exam will cover only the corresponding part of the material. See course program for details.

Students enrolled in both D.O. and N. O. (5 credits each) take the exam of "Optimization" (8 credits) and conduct a project/individual study (2 credits) to be defined with the instructor.

# OPTIMIZATION

joint course with

"Discrete Optimization" and "Nonlinear Optimization"

Edoardo Amaldi

DEIB – Politecnico di Milano
edoardo.amaldi@polimi.it

Course material on WeBeep "2023-24 - Optimization"

Academic year 2032-24

# Chapter 1: Introduction

Optimization is an active and successful branch of applied mathematics with a very wide range of relevant applications.

Given $X \subseteq \mathbb{R}^n$ and $f : X \to \mathbb{R}$ to be minimized, find an optimal solution $\underline{x}^* \in X$, i.e., such that

$$f(\underline{x}^*) \leq f(\underline{x}) \qquad \forall \underline{x} \in X.$$

**Course's aim**: Present the main concepts and methods of discrete and nonlinear (continuous) optimization, covering also modeling aspects.

See course's information slides also for prerequisites and joint courses.

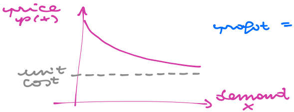
Many decision-making problems cannot be appropriately formulated/approximated in terms of linear models due to **intrinsic nonlinearity**.

## Examples

1) Production planning

Determine the production levels so as to maximize the total profit while respecting the resource availability constraints.

- "Price elasticity": unit profit decreases when amount produced increases.




profit $= p(x) \cdot x - \underbrace{c}_{\text{unit cost, here const but may become } c(x)} \cdot x = (p(x) - c) \cdot x \Rightarrow$ non-linear

- "Economy of scale": unit cost often decreases when amount produced increases.

2) Discrete decisions modeled with binary/integer variables.

Special type of nonlinearity: $\boxed{x \in \mathbb{Z}} \Leftrightarrow sin(\pi x) = 0$

this integrality constraint is
already a step towards
non linear constraints

ILP $\Rightarrow$ non linear opt

# 1.1 Examples of problems and models

## 1) Location and transportation

Given

- $m$ warehouses, indexed by $i = 1 \dots m$, with capacity $p_i$ and area $A_i \subseteq \mathbb{R}^2$
- $n$ clients with coordinates $(a_j, b_j)$ and demand $d_j$, with $j = 1 \dots n$,

decide where to locate warehouses and how to serve clients so as to minimize transportation costs while respecting capacities and demands.


DOMAINS

$w_{ij} \geq 0 \quad \forall i \, \forall j$

DECISION VARIABLES

- $\binom{x_i}{y_i}$ coords of warehouse $i$ $\quad \forall i$
- $w_{ij}$ the amount of product sold from warehouse $i$ to client $j$ $\quad \forall i \, \forall j$
- $t_{ij}$ the distance among $i$ and $j$ (an helper variable)

MODEL

$$\min \sum_i \sum_j (t_{ij} w_{ij})$$

s.t.

$$\sum_{j=1}^{n} w_{ij} \leq p_i \quad \forall i \quad \text{(capacity)}$$

$$\sum_{i=1}^{m} w_{ij} \geq d_j \quad \forall j \quad \text{(demand)}$$

$$t_{ij} = \sqrt{(x_i - a_j)^2 + (y_i - b_j)^2} \quad \forall i \, \forall j \quad \text{(helper variable definition)}$$

$$\binom{x_i}{y_i} \in A_i \subseteq \mathbb{R}^2 \quad \forall i \quad \text{(set belonging constraint)}$$

# 1.1 Examples of problems and models

## 1) Location and transportation

Given

- $m$ warehouses, indexed by $i = 1 \dots m$, with capacity $p_i$ and area $A_i \subseteq \mathbb{R}^2$
- $n$ clients with coordinates $(a_j, b_j)$ and demand $d_j$, with $j = 1 \dots n$,

decide where to locate warehouses and how to serve clients so as to minimize transportation costs while respecting capacities and demands.

Assumptions: single type of product and $\sum_{i=1}^{m} p_i \geq \sum_{j=1}^{n} d_j$

Decision variables:

Optimization model:

demand with $\geq$ and not $=$ as a natural choice, as we may satisfy the demand more than necessary even at the optimal set

MODEL

$$\min \sum_i \sum_j (t_{ij} w_{ij})$$

$$\text{s.t.} \quad \sum_{j=1}^{n} w_{ij} \leq p_i \quad \forall i \quad \text{(capacity)}$$

$$\sum_{i=1}^{m} w_{ij} \geq d_j \quad \forall j \quad \text{(demand)}$$

$$t_{ij} = \sqrt{(x_i - a_j)^2 + (y_i - b_j)^2} \quad \forall i \forall j \quad \text{(helper variable definition)}$$

$$\binom{x_i}{y_i} \in A_i \subseteq \mathbb{R}^2 \quad \forall i \quad \text{(set belonging constraint)}$$

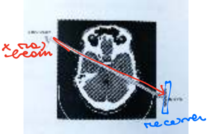
$$w_{ij} \geq 0 \quad \forall i \forall j \tag{1}$$

also there could have been a equality constraint (we have sure to have enough product)

$t_{ij}$ not really necessary, but just useful to write the model

## 2) Image reconstruction (Computerized Tomography)

Volume $V \subseteq \mathbb{R}^3$ subdivided into $n$ small cubes $V_j$ ("voxels").




Assumption: matter density is constant within each voxel.

Problem: Given measurments of $m$ beams, reconstruct 2-D image of $V$ ("slice"), i.e., determine the density $x_j$ for each $V_j$.

density map, knowing the exiting intensity measured

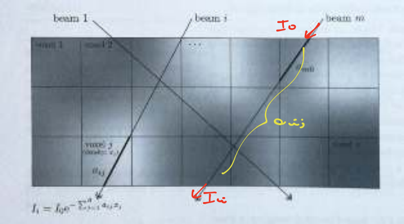
## 2) Image reconstruction (Computerized Tomography)

Volume $V \subseteq \mathbb{R}^3$ subdivided into $n$ small cubes $V_j$ ("voxels").

Assumption: matter density is constant within each voxel.

Problem: Given measurments of $m$ beams, reconstruct 2-D image of $V$ ("slice"), i.e., determine the density $x_j$ for each $V_j$.

2-D illustration:




For $i$-th beam: $a_{ij}$ is the path length within $V_j$,

$I_0$ is the X-ray intensity at source and $I_i$ at the exit.

The $i$-th beam total log-attenuation $\boxed{\log \frac{I_0}{I_i}}$ is linear in the density: $\boxed{\sum_{j=1}^{n} a_{ij} x_j}$

Given $m$ beams with prescribed directions,

$$\sum_{j=1}^{n} a_{ij} x_j = b_i = \log \frac{I_0}{I_i} \qquad i = 1, \ldots, m$$

$$x_j \geq 0 \qquad j = 1, \ldots, n$$

is usually infeasible due to measurement errors, non uniformity of $V_j$s,...

→ we can move to solve the LS (least squares) version

Given $m$ beams with prescribed directions,

$$\sum_{j=1}^{n} a_{ij} x_j = b_i = \log \frac{I_0}{I_i} \qquad i = 1, \ldots, m$$

$$x_j \geq 0 \qquad j = 1, \ldots, n$$

is usually infeasible due to measurement errors, non uniformity of $V_j$s,...

Possibile formulation:

$$\begin{array}{ll} \min & \sum_{i=1}^{m} (b_i - \sum_{j=1}^{n} a_{ij} x_j)^2 \\ s.t. & x_j \geq 0 \quad j = 1, \ldots, n. \end{array}$$

issue: now we have a lot of $x_j$'s, so lots of possible optimal sets
$\Rightarrow$ we add a penalization term

Given $m$ beams with prescribed directions,

$$\sum_{j=1}^{n} a_{ij} x_j = b_i = \log \frac{I_0}{I_i} \qquad i = 1, \ldots, m$$

$$x_j \geq 0 \qquad j = 1, \ldots, n$$

is usually infeasible due to measurement errors, non uniformity of $V_j$s,...

Possibile formulation:

$$\begin{array}{ll} \min & \sum_{i=1}^{m} (b_i - \sum_{j=1}^{n} a_{ij} x_j)^2 \\ s.t. & x_j \geq 0 \quad j = 1, \ldots, n. \end{array}$$

Since $n \gg m$, to avoid alternative optimal solutions we may minimize:

$$f(\underline{x}) = \sum_{i=1}^{m} (b_i - \sum_{j=1}^{n} a_{ij} x_j)^2 + \delta \sum_{j=1}^{n} x_j \qquad \text{with } \delta > 0$$

tries to drive to 0 as many $x_j$ as possible

$\Rightarrow$ regularization

$f(\underline{x})$ may also involve

- nonlinear terms accounting for the properties of matter/image
- stochastic model of attenuation and maximum likelihood estimator.

Also optimize the number/directions of beams.

4-D optimization to account for respiratory motion.

## 3) Combinatorial auctions

Participants (bidders) can place bids on combinations of discrete items.

Examples: airport time slots, wireless bandwidth, delivery routes, railroad segments, rare stamps or coins,...

Consider

- set $N$ of $n$ bidders,
- set $M$ of $m$ distinct items,
- for every $S \subseteq M$, $b_j(S)$ is the bid that $j \in N$ is willing to pay for $S$.

## 3) Combinatorial auctions

Participants (bidders) can place bids on combinations of discrete items.

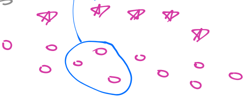
Examples: airport time slots, wireless bandwidth, delivery routes, railroad segments, rare stamps or coins,...

$b_j(S)$: how much bidder $j$ values the items of subset $S$

Consider

- set $N$ of $n$ bidders,
- set $M$ of $m$ distinct items,
- for every $S \subseteq M$, $b_j(S)$ is the bid that $j \in N$ is willing to pay for $S$.

$\triangledown$: bidders
o: items



Assumption: if $S \cap T = \emptyset$ then $b_j(S) + b_j(T) \leq b_j(S \cup T)$

bidders would pay more for $S \cup T$ rather than $S$ and $T$ individually (since the complete collection is more valuable)

$\Rightarrow$ we will give out larger groups of items, so it's more complex to optimize

Key problem: Determine the winner of each item so as to maximize total revenue.

For every $S \subseteq M$

let

- $b(S) = \max_{j \in N} b_j(S)$ — max amount that one bidder would give for that subset

- DECISION VARIABLE — $x_S = \begin{cases} 1 & \text{if the highest bid on } S \text{ was accepted} \\ 0 & \text{otherwise} \end{cases} \quad \forall S \subseteq M$

MODEL $\max \sum_{S \subseteq M} x_S \cdot b_S$ (maximize the revenue)

Formulation:

s.t. $\sum_{S: i \in S} x_S \leq 1$ (each item must be assigned to at most one subset $S$)

$x_S \in \{0,1\}$ — issue: model is exact theoretically, but requires too much variables, $2^M$

# General optimization problem

$$\boxed{\begin{array}{lll} \min & f(\underline{x}) & \\ s.t. & g_i(\underline{x}) \leq 0 & 1 \leq i \leq m \\ & \underline{x} \in S \subseteq \mathbb{R}^n & \end{array}}$$

(algebraic) constraint

set constraint

- the algebraic and set constraints define the **feasible region**

$$X = S \cap \{\underline{x} \in \mathbb{R}^n \ : \ g_i(\underline{x}) \leq 0, 1 \leq i \leq m\},$$

where $g_i : S \to \mathbb{R}$ for $i = 1, \ldots, m$.

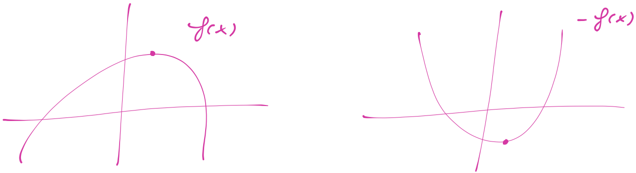
- **objective function** $f(\underline{x})$ with $f : X \to \mathbb{R}$.

Assume w.l.o.g. that

- minimization problem since

$$\max\{f(\underline{x}) \;:\; \underline{x} \in X\} = -\min\{-f(\underline{x}) \;:\; \underline{x} \in X\}.$$

Illustration:




- all algebraic constraints are inequality constraints since

$$g(\underline{x}) = 0 \qquad \equiv \qquad \begin{cases} g(\underline{x}) \leq 0 \\ g(\underline{x}) \geq 0. \end{cases}$$

**Definition**

i) A feasible solution $\underline{x}^* \in X$ is a **global optimum** if

$$f(\underline{x}^*) \leq f(\underline{x}) \qquad \forall \underline{x} \in X.$$

ii) A feasible solution $\overline{\underline{x}} \in X$ is a **local optimum** if $\exists\ \epsilon > 0$ such that

$$f(\overline{\underline{x}}) \leq f(\underline{x}) \qquad \forall \underline{x} \in X \cap \mathcal{N}_\epsilon(\overline{\underline{x}})$$

an optimum over a suitable interval

where $\mathcal{N}_\epsilon(\overline{\underline{x}}) = \{\underline{x} \in X : \|\underline{x} - \overline{\underline{x}}\| \leq \epsilon\}$.

Illustration:

For difficult problems, we settle for good local optima within a reasonable computing time.

# Main classes of optimization problems

Terminology: programming $\equiv$ optimization

| $f$ | $g_i$ | $S$ | problem type |
|---|---|---|---|
| linear | linear | $S = \mathbb{R}^n$ | Linear Programming (LP) |
| linear | linear | $S \subseteq \mathbb{Z}^n$ | Integer LP (ILP) |
| linear | linear | $S \subseteq \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2}$ with $n = n_1 + n_2$ | Mixed Integer LP (MILP) |
| at least one nonlinear | | $S \subseteq \mathbb{R}^n$ | Nonlinear Programming (NLP) |
| at least one nonlinear | | $S \subseteq \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2}$ with $n = n_1 + n_2$ | Mixed Integer NLP (MINLP) |

Some important special cases:

*Quadratic programming*: $f(\underline{x}) = \underline{x}^T Q \underline{x} + \underline{c}^T \underline{x}$ with linear constraints

*Convex programming*: functions $f$ and $g_i$s and set $S$ are convex.

# Some fields of application

- health care planning and management (treatment planning, workforce scheduling, operating theater scheduling,...)
- logistics (location of plants and services, transportation, routing) and supply chain design and management
- data mining and machine learning: classification, clustering, approximation,..
- optimal control (determine the trajectory of a robot arm, airplane, shuttle)
- computational biology (determine the 3-D structure of proteins,...)
- economics (risk management, portfolio optimization, combinatorial auctions, equilibria of games,. . . )
- network planning and management (wired and wireless telecommunications, electric networks,...)
- production planning and inventory management (manufacturing, chemical processes, energy generation,...)

# Some fields of application

- management of environmental and territorial resources (water, forest,...)
- design of experiments (for chemical and pharmaceutical companies)
- signal and image processing (2-D and 3-D reconstruction)
- statistics (e.g., nonlinear regression, estimation of distribution parameters)
- agriculture and agri-food industry
- dimensioning and optimization of structures (bridge, aircraft profile,...)
- ...

# Chapter 2: Fundamentals of convex analysis

Edoardo Amaldi

DEIB – Politecnico di Milano
edoardo.amaldi@polimi.it

Course material on WeBeep "2023-24 - Optimization"

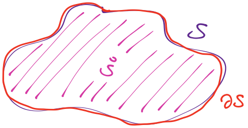
Academic year 2023-24

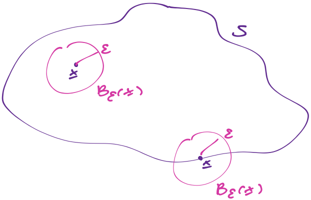
# 2.1 Basic concepts

In $\mathbb{R}^n$ with Euclidean norm

- $\underline{x} \in S \subseteq \mathbb{R}^n$ is an **interior point** of $S$ if $\exists\, \varepsilon > 0$ such that $B_\varepsilon(\underline{x}) = \{\underline{y} \in \mathbb{R}^n \ : \ \|\underline{y} - \underline{x}\| < \varepsilon\} \subseteq S$.
- $\underline{x} \in \mathbb{R}^n$ is a **boundary point** of $S$ if, for every $\varepsilon > 0$, $B_\varepsilon(\underline{x})$ contains at least one point of $S$ and one point of $\mathbb{R}^n \setminus S$.
- Set of all interior points of $S \subseteq \mathbb{R}^n$ is the **interior** of $S$, denoted by *int(S)*.
- Set of all boundary points of $S$ is the **boundary** of $S$, denoted by $\partial(S)$.

Illustrations:



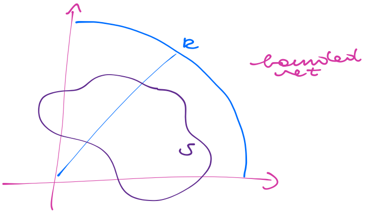
In $\mathbb{R}^n$ with Euclidean norm

- $S \subseteq \mathbb{R}^n$ is **open** if $S = int(S)$; $S$ is **closed** if its complement is open.
  Intuitively, a closed set contains all the points in $\partial(S)$.

- $S \subseteq \mathbb{R}^n$ is **bounded** if $\exists\ M > 0$ such that $\|\underline{x}\| \leq M$ for every $\underline{x} \in S$.

- $S \subseteq \mathbb{R}^n$ closed and bounded is **compact**.

Illustrations:



Properties:

$S \subseteq \mathbb{R}^n$ is closed if and only if every sequence $\{\underline{x}_i\}_{i \in \mathbb{N}} \subseteq S$ that converges, converges to $\underline{x} \in S$.

$S \subseteq \mathbb{R}^n$ is compact if and only if every sequence $\{\underline{x}_i\}_{i \in \mathbb{N}} \subseteq S$ admits a subsequence that converges to a point $\underline{x} \in S$.

For convex analysis see:

Bazaraa, Sherali, Shetty, Nonlinear Programming – Theory and Algorithms, third edition, Wiley Interscience, 2006 (Chapters 2 and 3)

# Existence of an optimal solution

without any assumption on $S$ and $f$

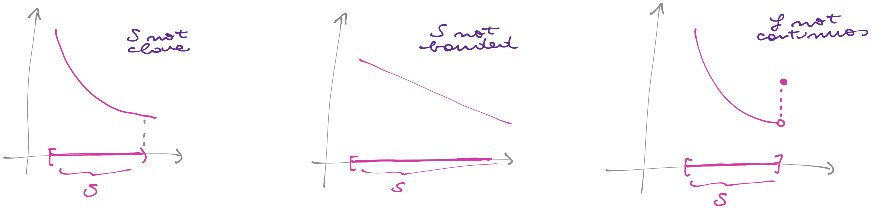
In general, when minimizing $f : S \subseteq \mathbb{R}^n \to \mathbb{R}$, we only know that a largest lower bound (infimum) exists, that is

$$\inf_{\underline{x} \in S} f(\underline{x}).$$

**Theorem (Weierstrass)**:

Let $S \subseteq \mathbb{R}^n$ be nonempty and compact, and $f : S \to \mathbb{R}$ be continuous. Then $\exists\ \underline{x}^* \in S$ such that $\boxed{f(\underline{x}^*) \leq f(\underline{x})}$ for every $\underline{x} \in S$. so there exists a global optimum, and it was reached

Examples where the result does not hold:




When $\underline{x}^* \in S$ exists, we can write $\min_{\underline{x} \in S} f(\underline{x})$.

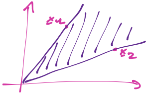
# Cones and affine subspaces

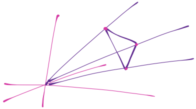
Consider any $S \subset \mathbb{R}^n$

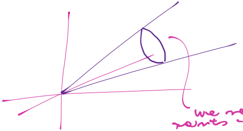
**Definition**: $\boxed{cone(S)}$ denotes the set of all **conic combinations** of points of $S$, i.e., all $\underline{x} = \sum_{i=1}^{m} \alpha_i \underline{x}_i$ with $\underline{x}_1, \ldots, \underline{x}_m \in S$ and $\alpha_i \geq 0$ for every $i$, $1 \leq i \leq m$.

Examples: *polyedral cones* and "ice cream" cones

generated by a finite # of points



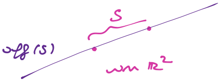


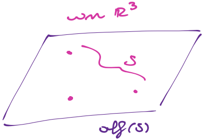
we need all points here

**Definition**: $\boxed{aff(S)}$ denotes the smallest **affine subspace** that contains $S$.

$aff(S)$ coincides with the set of all **affine combinations** of points in $S$, i.e., all $\underline{x} = \sum_{i=1}^{m} \alpha_i \underline{x}_i$ with $\underline{x}_1, \ldots, \underline{x}_m \in S$, $\sum_{i=1}^{m} \alpha_i = 1$, and $\alpha_i \in \mathbb{R}$ for every $i$, $1 \leq i \leq m$.

Examples:



# 2.2 Elements of convex analysis

**Definitions**:

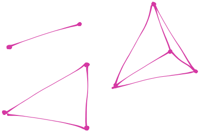

- $C \subset \mathbb{R}^n$ is **convex** if

$$\alpha \underline{x}_1 + (1-\alpha)\underline{x}_2 \in C \quad \forall \underline{x}_1, \underline{x}_2 \in C \quad \text{and} \quad \forall \alpha \in [0,1].$$

α let us move over the points of the segment from x1 to x2

- $\underline{x} \in \mathbb{R}^n$ is a **convex combination** of $\underline{x}_1, \ldots, \underline{x}_m \in \mathbb{R}^n$ if

$$\underline{x} = \sum_{i=1}^{m} \alpha_i \, \underline{x}_i$$

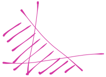


with $\sum_{i=1}^{m} \alpha_i = 1$ and $\alpha_i \geq 0$ for every $i$, $1 \leq i \leq m$.

~ affine ~ conic

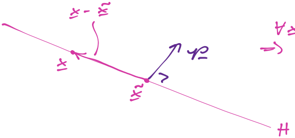
relevant examples are the hyperplanes, or the regions of set to inequalities

**Property**: If $C_i$ with $i = 1, \ldots, k$ are convex, then $\cap_{i=1}^{k} C_i$ is convex.



# Examples of convex sets

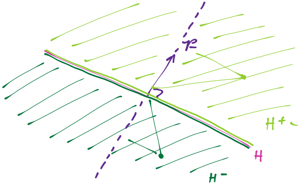
**1) Hyperplane** $H = \{\underline{x} \in \mathbb{R}^n \ : \ \underline{p}^t \underline{x} = \beta\}$ with $\underline{p} \neq \underline{0}$.




$\forall \underline{x} \ \underline{p}^T \underline{x} = \beta$
$\Rightarrow H = \{\underline{x} \in \mathbb{R}^n : \underline{p}^T(\underline{x} - \tilde{\underline{x}}) = 0\}$
which means $\underline{p}$ is $\perp$ to all $\underline{x} \in H$

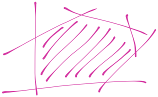
N.B.: $H$ is closed since $H = \partial(H)$

**2)** Closed **half-spaces** $H^+ = \{\underline{x} \in \mathbb{R}^n \ : \ \underline{p}^t \underline{x} \geq \beta\}$ and $H^- = \{\underline{x} \in \mathbb{R}^n \ : \ \underline{p}^t \underline{x} \leq \beta\}$ with $\underline{p} \neq \underline{0}$.




contains the vector $\underline{p}$

**3) Feasible region** $X = \{\underline{x} \in \mathbb{R}^n \; : \; A\underline{x} \geq \underline{b}, \; \underline{x} \geq \underline{0}\}$ of a Linear Program (LP)



$$\begin{array}{rl} \min & \underline{c}^t\underline{x} \\ s.t. & A\underline{x} \geq \underline{b} \\ & \underline{x} \geq \underline{0} \end{array}$$

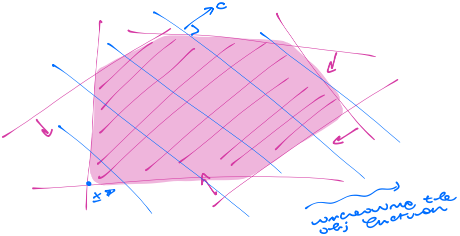
n variables

$X$ is a convex and closed subset (intersection of $m + n$ closed half-spaces if $A \in \mathbb{R}^{m \times n}$).

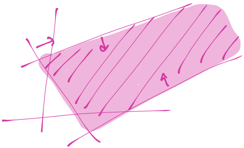
m constraint

**Definition**: The intersection of a finite number of closed half-spaces is a **polyedron**.

Illustration:




can also be unbounded

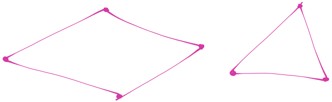


N.B.: The set of optimal solutions of a LP is a polyhedron (add $\underline{c}^t\underline{x} = z^*$ with optimal $z^*$)

# Convex hulls and extreme points

**Definition**: The **convex hull** of $S \subseteq \mathbb{R}^n$, denoted by $\boxed{conv(S)}$, is the intersection of all convex sets containing $S$.

Illustration:



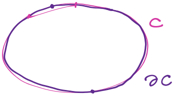
Equivalent characterizations (external/internal descriptions): $conv(S)$ and set of all convex combinations of points in $S$.

**Definition**: Given $C \subseteq \mathbb{R}^n$ convex, $\underline{x} \in C$ is an **extreme point** of $C$ if it cannot be expressed as convex combination of two different points of $C$, that is

$$\underline{x} = \alpha \underline{x}_1 + (1-\alpha)\underline{x}_2 \qquad \text{with } \underline{x}_1, \underline{x}_2 \in C \text{ and } \alpha \in (0,1)$$

implies that $\underline{x}_1 = \underline{x}_2$.

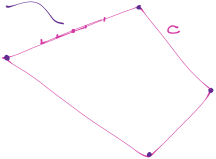
Examples:




$\partial C$

all the boundary points (in this example) are all extreme points

now extreme points are just the vertices, as on the edges there are more choices



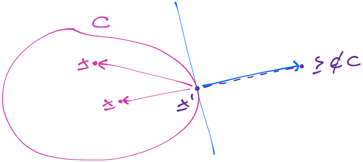
# Projection on a convex set

**Lemma (Projection)**:

Let $C \subseteq \mathbb{R}^n$ be nonempty, closed and convex, then for every $\boxed{\underline{y} \notin C}$ there exists a unique $\boxed{\underline{x}' \in C}$ at minimum distance from $\underline{y}$.
Moreover, $\underline{x}' \in C$ is the closest point to $\underline{y}$ if and only if characterization of the point $\underline{x}'$

$$(\underline{y} - \underline{x}')^t(\underline{x} - \underline{x}') \leq \underline{0} \qquad \forall \underline{x} \in C.$$

Geometric Ilustration:




**Definition**: $\underline{x}'$ is the **projection** of $\underline{y}$ on $C$.

# Separation theorem

Geometrically intuitive but fundamental result.

same assumptions of before

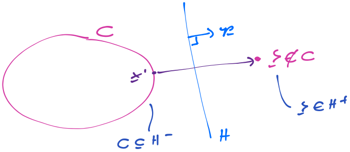
**Theorem (Separating hyperplane)**

Let $C \subset \mathbb{R}^n$ be nonempty, closed and convex and $\boxed{\underline{y} \notin C}$, then $\boxed{\exists\ \underline{p} \in \mathbb{R}^n}$ such that $\boxed{\underline{p}^t \underline{x} < \underline{p}^t \underline{y}}$ for every $\underline{x} \in C$.

$\exists$ hyperplane $H = \{\underline{x} \in \mathbb{R}^n\ :\ \underline{p}^t \underline{x} = \beta\}$ with $\underline{p} \neq \underline{0}$ separating $\underline{y}$ from $C$, i.e., such that

$$C \subseteq H^- = \{\underline{x} \in \mathbb{R}^n\ :\ \underline{p}^t \underline{x} \leq \beta\} \quad \text{and} \quad \underline{y} \notin H^- \ \ (\underline{p}^t \underline{y} > \beta)$$

Illustration:




Proof:

actually there exist an number of separating hyperplanes

$\Rightarrow$ we can take $\boxed{\underline{p} = \underline{y} - \underline{x}'}$ where $\underline{x}'$ is the projection using the previous lemma

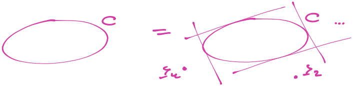
as from the lemma we had $(\underline{y} - \underline{x}')^T(\underline{x} - \underline{x}') := \underline{p}^T(\underline{x} - \underline{x}') \leq 0 \quad \forall \underline{x} \in H$

$\underline{p}^T \underline{x} \leq \underline{p}^T \underline{x}' := \beta$

- $\forall \underline{x} \in C \quad \underline{p}^T \underline{x} \leq \beta$ as a consequence
- about $\underline{y}$ we have $\underline{p}^T \underline{y} - \beta = \underline{p}^T \underline{y} - \underline{p}^T \underline{x}' = \underline{p}^T(\underline{y} - \underline{x}') = (\underline{y} - \underline{x}')^T(\underline{y} - \underline{x}') = \|\underline{y} - \underline{x}'\|^2 > 0$

# Consequences of separation theorem

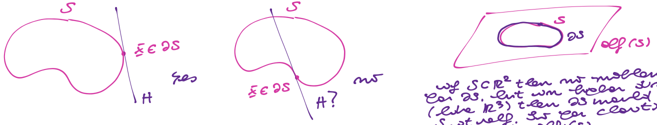
**1)** Any nonempty, closed and convex set $C \subseteq \mathbb{R}^n$ is the intersection of all closed half-spaces containing it.




(i.e. among all the separating hyperplanes taking all $z \notin C$)

**Definition**: Let $S \subset \mathbb{R}^n$ with $S \neq \emptyset$ and $\underline{\overline{x}} \in \partial(S)$ ( boundary w.r.t. $\mathit{aff}(S)$ ), $H = \{\underline{x} \in \mathbb{R}^n \ : \ \underline{p}^t(\underline{x} - \underline{\overline{x}}) = 0\}$ is a **supporting hyperplane** of $S$ at $\underline{\overline{x}}$ if $S \subseteq H^-$ or $S \subseteq H^+$.

Illustration:

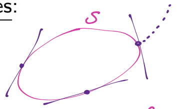



if $S \subset \mathbb{R}^2$ then no problem for $\partial S$, but in higher dim (like $\mathbb{R}^3$) then $\partial S$ would be $S$ itself so we refer to aff(S)

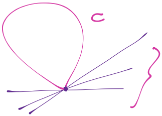
**2) Supporting hyperplane**:

If $C \neq \emptyset$ is convex then for every $\boxed{\underline{\overline{x}} \in \partial(C)}$ there exists (at least) a supporting hyperplane $H$ at $\underline{\overline{x}}$, i.e., $\exists \ \underline{p} \neq \underline{0}$ such that $\underline{p}^t(\underline{x} - \underline{\overline{x}}) \leq 0$, for each $\underline{x} \in C$.

Examples:




qualunque $\underline{x} \in \partial S$ ha un supporting hyperplane (dim: limit from the separating hyperplanes)




cone of $\exists \infty$ supporting hyperplanes

Central result of Optimization (Game theory) from which we will derive the optimality conditions for Nonlinear Optimization.

n variables
m constraints

**3) Farkas Lemma**:

Let $A \in \mathbb{R}^{m \times n}$ and $\underline{b} \in \mathbb{R}^m$. Then

$$\exists\ \underline{x} \in \mathbb{R}^n \text{ such that } \boxed{A\underline{x} = \underline{b}} \text{ and } \boxed{\underline{x} \geq \underline{0}} \;\Leftrightarrow\; \boxed{\nexists\ \underline{y} \in \mathbb{R}^m} \text{ such that } \boxed{\underline{y}^t A \leq \underline{0}^t} \text{ and } \boxed{\underline{y}^t \underline{b} > 0.}$$

iff sys admits a non negative sol

then second sys is infeasible

Provides an infeasibility certificate, also known as theorem of the alternative.

Alternative: exactly one of $A\underline{x} = \underline{b}, \underline{x} \geq \underline{0}$ and $\underline{y}^t A \leq \underline{0}^t, \underline{y}^t \underline{b} > 0$ is feasible.

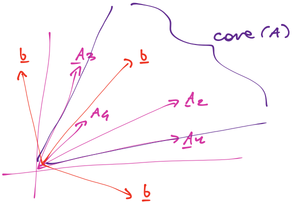
$A\underline{x} = \underline{b}$

neg wt xs:

$\underbrace{\sum_j \underline{A}_j x_j \ni \underline{b}}$

non neg coeffs → cone generated by the cols of A

Geometric interpretation:




$\underline{b}$ belongs to (convex) cone generated by the columns of $A$, i.e.
$cone(A) = \{\underline{z} \in \mathbb{R}^m \ :\ \underline{z} = \sum_{j=1}^{n} x_j A_j, x_1 \geq 0, \ldots, x_n \geq 0\}$
if and only if no hyperplane separating $\underline{b}$ from $cone(A)$ exists.

Alternative: $\boxed{\underline{b} \in cone(A)}$ or $\boxed{\underline{b} \notin cone(A)}$

Alternative:

$\underline{b} \in cone(A) or \underline{b} \notin cone(A)$

Proof (Farkas Lemma):

($\Rightarrow$) Consider $\underline{\tilde{x}} \geq \underline{0}$ st $A\underline{\tilde{x}} = \underline{b}$ (we assume sys 1 feasible)
now $\forall \underline{y}$ st $\underline{y}^T A \leq 0$ we have that

$$\underline{y}^T \underline{b} = \underline{y}^T (A \underline{\tilde{x}}) = \underbrace{(\underline{y}^T A)}_{\leq 0} \underbrace{\underline{\tilde{x}}}_{\geq 0} \leq 0 \Rightarrow$$

so, $\nexists \underline{y}$ st $\underline{y}^T \underline{b} > 0$ while also $\underline{y}^T A \leq 0$

(⇐) We prove this by showing that (!1 ⇒ !2).
So now we assume that sys1 is unfeasible, that is

$\nexists\, \underline{x} \geq 0 : A\underline{x} = \underline{b}$ i.e. $\underline{b} \notin \text{cone}(A)$

Consider the $\text{cone}(A) = \{ \underline{z} \in \mathbb{R}^m : \underline{z} = \sum_{j=1}^{n} \underline{A}_j x_j \mid x_j \geq 0 \ \forall j \}$
which
- is non empty ($\underline{0} \in \text{cone}(A)$)
- is closed and convex
- and $\underline{b} \notin \text{cone}(A)$

⇒ we can apply the separating hyperplane thm, which tells us

$\exists\, \underline{y} \in \mathbb{R}^m, \beta \in \mathbb{R} : \underbrace{\underline{y}^T \underline{b}}_{\text{point outside the cone}(A)} > \beta$ and $\underbrace{\underline{y}^T \underline{z}}_{\text{points inside the cone}(A)} \leq \beta \quad \forall \underline{z} \in \text{cone}(A)$

Since $\underline{0} \in \text{cone}(A)$ then $\underline{y}^T \underline{0} = 0 \leq \beta$ so $\beta \geq 0$, and $\underline{y}^T \underline{b} > 0$.
Moreover,

$$\left.\begin{array}{l} \underline{y}^T \underline{z} \leq \beta \ \ \forall \underline{z} \in \text{cone}(A) \\ \text{but } \underline{z} = \sum \underline{A}_j x_j = A\underline{x} \end{array}\right\} \Rightarrow \underbrace{(\underline{y}^T A)}\ \underbrace{\underline{x}}_{\geq 0} \leq \beta$$

if $(\underline{y}^T A)_i > 0$ then we could take $x_i$ large enough to have $(\underline{y}^T A)\underline{x} > \beta$

since $\underline{x} \geq 0$, this is true only if $\underline{y}^T A \leq \underline{0}$
Thus $\underline{y} \neq \underline{0}$, $\underline{y}^T A \leq 0$ and $\underline{y}^T \underline{b} > 0$

⇐ which is equivalent to

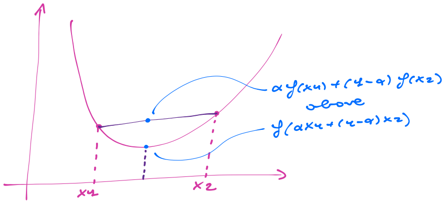
having sys2 unfeasible
(with $\underline{y}$ in the role of $\underline{y}$)

# 2.2.2 Convex functions

**Definitions**:

- A function $f : C \to \mathbb{R}$ defined on a convex set $C \subseteq \mathbb{R}^n$ is **convex** if

$$\underbrace{f(\alpha \underline{x}_1 + (1-\alpha)\underline{x}_2)}_{f(\text{convex comb})} \;\leq\; \underbrace{\alpha f(\underline{x}_1) + (1-\alpha) f(\underline{x}_2)}_{\text{convex comb of the images}} \quad \forall \underline{x}_1, \underline{x}_2 \in C \quad \text{and} \quad \forall \alpha \in [0,1],$$




- $f$ is **strictly convex** if the inequality holds with $<$ for all $\underline{x}_1, \underline{x}_2 \in C$ with $\underline{x}_1 \neq \underline{x}_2$ and $\alpha \in (0,1)$.

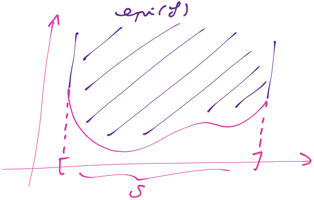
- $f$ is **concave** if $-f$ is convex; $f$ is **linear** if it is both convex and concave.

connection between convexity of sets and of functions

**Definitions**:

- The **epigraph** of $f : S \subseteq \mathbb{R}^n \to \mathbb{R}$, denoted by $epi(f)$, is the subset of $\mathbb{R}^{n+1}$

$$epi(f) = \{(\underline{x}, y) \in S \times \mathbb{R} \ : \ y \geq f(x)\}.$$




- Let $f : C \to \mathbb{R}$ be convex, the **domain** of $f$ is the subset of $\mathbb{R}^n$

$$dom(f) = \{\underline{x} \in C \ : \ f(\underline{x}) < +\infty\}.$$

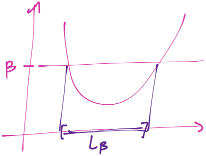
**Properties**:

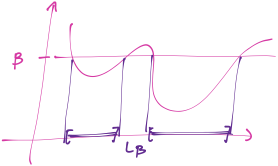
Let $C \subseteq \mathbb{R}^n$ with $C \neq \emptyset$ and $f : C \to \mathbb{R}$ be convex.

- For each $\beta \in \mathbb{R}$ (also $\beta \in +\infty$), the *level sets*

$$L_\beta = \{\underline{x} \in C : f(\underline{x}) \leq \beta\} \quad \text{and} \quad \{\underline{x} \in C : f(\underline{x}) < \beta\}$$

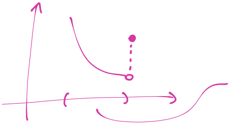
  are convex subsets of $\mathbb{R}^n$.




- $f$ is continuous in the relative interior (with respect to *aff*$(C)$) of its domain.

still for the smallest of 2 points



here the function (convex) non cont

- $\boxed{f \text{ is convex}}$ if and only if $\boxed{epi(f) \text{ is a convex subset}}$ of $\mathbb{R}^{n+1}$ (exercise 1.5).

# Optimal solution of convex problems

convexity is useful to avoid local minima

Consider $\min_{x \in C \subseteq \mathbb{R}^n} \; f(\underline{x})$ where $C \subseteq \mathbb{R}^n$ and $f : C \to \mathbb{R}$ are convex.

**Proposition**:

i) If $C$ and $f$ are convex, each local minimum of $f$ on $C$ is a global minimum.

ii) If $f$ is strictly convex on $C$, $\exists$ at most one global minimum (if not unbounded).

C convex

Proof:

(1) Suppose that $\underline{x}'$ is a loc min and $\exists \underline{x}^* \in C$ a glob min s.t. $f(\underline{x}^*) < f(\underline{x}')$. We can study the convex comb

$$f(\alpha \underline{x}' + (1-\alpha)\underline{x}^*) \leq \alpha f(\underline{x}') + (1-\alpha)\underbrace{f(\underline{x}^*)}_{< f(\underline{x}')} < f(\underline{x}') \quad \forall \alpha \in [0,1]$$

This contradicts the fact that $\underline{x}'$ was a local min. So actually all loc min are also global

(2) If $f$ is strictly convex and $\underline{x}_1^*$ and $\underline{x}_2^*$ are glob min, then the convexity of $C$ implies that

$$\tfrac{1}{2}\underline{x}_1^* + \tfrac{1}{2}\underline{x}_2^* \in C \text{ still}$$

and the strict convexity of $f$ implies that

$$f\left(\tfrac{1}{2}\underline{x}_1^* + \tfrac{1}{2}\underline{x}_2^*\right) < \tfrac{1}{2} f(\underline{x}_1^*) + \tfrac{1}{2} f(\underline{x}_2^*) \Rightarrow$$ $\underline{x}_1^*$ and $\underline{x}_2^*$ cannot be two glob min

$\Rightarrow$ at most one glob min could exist

**Special case**: Linear programming (LP) problems

$$\begin{array}{rl} \min & \underline{c}^t \underline{x} \\ s.t. & A\underline{x} \geq \underline{b} \\ & \underline{x} \geq \underline{0} \end{array}$$

**Proposition**:

Given any LP with $P = \{\underline{x} \in \mathbb{R}^n \ : \ A\underline{x} \geq \underline{b}, \underline{x} \geq \underline{0}\} \neq \emptyset$, then either $\exists$ (at least) one optimal extreme point or the objective function value is unbounded below over $P$.

Geometric illustration:

# Characterizations of convex functions

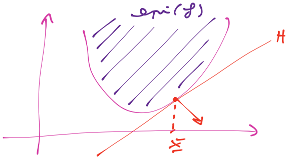
**Proposition 1**: $f : C \to \mathbb{R}$ of class $\mathcal{C}^1$ with nonempty convex and open $C \subseteq \mathbb{R}^n$ is convex if and only if

$$\boxed{f(\underline{x}) \geq f(\overline{x}) + \nabla^t f(\overline{x})(\underline{x} - \overline{x}) \qquad \forall \underline{x}, \overline{x} \in C.}$$

$f$ is strictly convex if and only if inequality holds with $>$ for all $\underline{x}, \overline{x} \in C$ with $\underline{x} \neq \overline{x}$.

$f(\cdot)$ stays always [illegible]
[illegible]

first order Taylor approx, the tangent hyperplane




Geometric interpretation:

The linear approximation of $f$ at $\overline{x}$ (1st order Taylor's expansion) bounds below $f(\underline{x})$ and

$$\boxed{H} = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^{n+1} \ : \ (\nabla^t f(\overline{x}) \quad -1) \begin{pmatrix} x \\ y \end{pmatrix} = -f(\overline{x}) + \nabla^t f(\overline{x}) \, \overline{x} \right\}$$

is a supporting hyperplane of $\boxed{epi(f)}$ at $(\overline{x}, f(\overline{x}))$, with $epi(f) \subseteq H^-$.

**Proposition 2**: $f : C \to \mathbb{R}$ of class $\mathcal{C}^2$ with nonempty convex and open $C \subseteq \mathbb{R}^n$ is convex if and only if the Hessian matrix $\nabla^2 f(\underline{x}) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right)$ is positive semidefinite at every $\underline{x} \in C$.

For $f \in \mathcal{C}^2$, if $\nabla^2 f(\underline{x})$ is positive definite $\forall \underline{x} \in C$ then $f(\underline{x})$ is strictly convex.

N.B.: Sufficient condition not necessary:

eg think of $f(x) = x^4$
- is strictly convex
- but $f''(0) = 0$ so it's not true that is pos definite

**Definition**:

A symmetric matrix $A$ $n \times n$ is *positive definite* if $\underline{y}^t A \underline{y} > 0 \quad \forall \underline{y} \in \mathbb{R}^n$ with $\underline{y} \neq \underline{0}$,

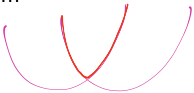
A symmetric matrix $A$ $n \times n$ is *positive semidefinite* if $\underline{y}^t A \underline{y} \geq 0 \quad \forall \underline{y} \in \mathbb{R}^n$.

Equivalent definitions: based on the sign of the eigenvalues/principal minors of $A$ or of the diagonal coefficients of specific factorizations of $A$ (e.g., Cholesky factorization).

# Convexity-preserving operations

Certain operations preserve the convexity of functions:

- weigthed sum with non-negative weights
- pointwise maximum
- ...



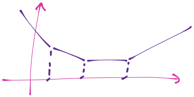
See exercise 1.4

# Subgradients of convex/concave functions

Convex/concave not everywhere differentiable (continuous) functions, e.g. $f(x) = |x|$.

but

comes up often, especially when solving with the dual problem



Generalization of the concept of gradient for $\mathcal{C}^1$ functions to piecewise $\mathcal{C}^1$ functions.

**Definitions**: Let $C \subseteq \mathbb{R}^n$ and $f : C \to \mathbb{R}$ be convex.

- $\underline{\gamma} \in \mathbb{R}^n$ is a **subgradient** of $f$ at $\overline{x} \in C$ if

we try the $\infty$ choices of true vectors $\gamma$ to make that inequality at the $\bar{x}$ of non derivability

$$f(\underline{x}) \geq f(\overline{x}) + \underline{\gamma}^t(\underline{x} - \overline{x}) \qquad \forall \underline{x} \in C,$$

- The **subdifferential**, denoted by $\partial f(\underline{x})$, is the set of all the subgradients of $f$ at $\underline{x}$.

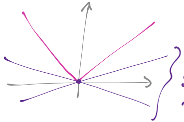
Example: $f(x) = x^2$, the only subgradient at $\overline{x} = 3$ is $\gamma = 6$.

$0 \leq (x-3)^2 = x^2 - 6x + 9$

$\Rightarrow f(x) = x^2 \geq 6x - 9 = 9 + 6(x-3) =$

$= f(3) + 6(x-3)$

$\Rightarrow$ $\bar{x} = 3$ and the only $\gamma$ is $\gamma = 6$



all possible slopes of $\gamma$

$\gamma \in [-1, 1]$

Other examples:

1) For $f(x) = |x|$, $\quad \partial \begin{cases} = 1 & \text{for } x > 0 \\ \in [-1, 1] & \text{for } x = 0 \\ = -1 & \text{for } x < 0 \end{cases}$

2) Consider $f(x) = \min\{f_1(x), f_2(x)\}$ with $f_1(x) = 4 - |x|$ and $f_2(x) = 4 - (x-2)^2$.

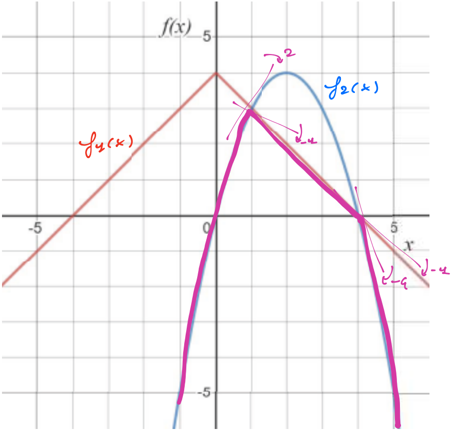
$$f(x) = \begin{cases} 4 - x & 1 \leq x \leq 4 \\ 4 - (x-2)^2 & \text{otherwise} \end{cases}$$

$\frac{\partial}{\partial x} f_1$

- for $x \in (1,4)$
  $\gamma(x) = -1$
- for $x < 1$ or $x > 4$
  $\gamma(x) = -2(x-2)$

$\frac{\partial}{\partial x} f_2$

- for $x = 1$
  $\gamma \in [-1, 2]$
  $f_1'(\bar{x})$ $f_2'(\bar{x})$
- for $x = 4$
  $\gamma \in [-4, -1]$
  $f_2'(\bar{x})$ $f_1'(\bar{x})$



**Properties**:

Let $C \subseteq \mathbb{R}^n$ and $f : C \to \mathbb{R}$ be convex.

1) $f$ admits at least a subgradient at every interior point $\overline{\underline{x}}$ of $C$.

In particular, if $\overline{\underline{x}} \in int(C)$ then $\exists\ \underline{\gamma} \in \mathbb{R}^n$ such that

$$H = \{(\underline{x}, y) \in \mathbb{R}^{n+1}\ :\ y = f(\overline{\underline{x}}) + \underline{\gamma}^t(\underline{x} - \overline{\underline{x}})\}$$

is a supporting hyperplane of $epi(f)$ at $(\overline{\underline{x}}, f(\overline{\underline{x}}))$.

$\exists$ of at least one subgradient at any point of $\mathring{C}$ (C convex) $\Leftarrow$ f convex on $\mathring{C}$

the set of all subgradients ($\gamma$) at that point x

2) If $\underline{x} \in C$, $\partial f(\underline{x})$ is a nonempty, convex, closed and bounded set.

3) $\boxed{\underline{x}^*}$ is a (global) minimum of $f$ on $C$ if and only if $\boxed{\underline{0} \in \partial f(\underline{x}^*)}$.

global since for f convex loc = global minimum

$f(\underline{x}) \geq f(\underline{x}^*) + \cancel{\underline{0}^T(\underline{x} - \underline{x}^*)}$

$\Rightarrow$ $\underline{x}^*$ is a minimum (if $\exists\ \gamma = 0 \in \partial f(\underline{x}^*)$)

# Chapter 3: Discrete Optimization – Integer Linear Programming

followup of FRO

Edoardo Amaldi

DEIB – Politecnico di Milano
edoardo.amaldi@polimi.it

Course material on WeBeep "2023-24 - Optimization"

Academic year 2023-24

# 3.1 Integer Programming models

A wide variety of decision-making problems in science, engineering and management can be formulated as discrete optimization problems:

$$\min_{\underline{x} \in X} c(\underline{x}) \iff \begin{array}{l} \min c(\underline{x}) \\ \text{st } \underline{x} \in X \end{array}$$

where $X$ discrete set and $c : X \to \mathbb{R}$.

A natural and systematic way to tackle them is as Integer Optimization problems.

**Definitions**: A generic **Mixed Integer Linear Programming** (MILP) problem is

$$\begin{array}{lr} \min & \underline{c}^t \underline{x} \\ s.t. & A\underline{x} \geq \underline{b} \\ & \underline{x} \in \mathbb{Z}^{n_1} \times \mathbb{R}^{n_2} \end{array}$$

"linear" obj function

$n_1$ variables $\in \mathbb{Z}$
$n_2$ variables $\in \mathbb{R}$

with $A \in \mathbb{Z}^{m \times (n_1+n_2)}$, $\underline{c} \in \mathbb{Z}^{n_1+n_2}$ and $\underline{b} \in \mathbb{Z}^m$.

If $x_j \in \mathbb{Z}$ for all $j$, it is an **Integer Linear Programming** (ILP) problem.

If $x_j \in \{0, 1\}$ for all $j$, it is a **Binary Linear Programming** (0-1-ILP) problem.

W.l.o.g. only inequalities and all coefficients are integer. if are fractions we can scale everything by the common denominator

Recall: $\boxed{x_i \in \mathbb{Z}}$ is nonlinear constraint

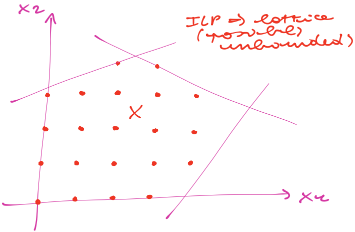
**Proposition**: 0-1-ILP is $\mathrm{NP}$-hard, (M)ILP are at least as difficult.

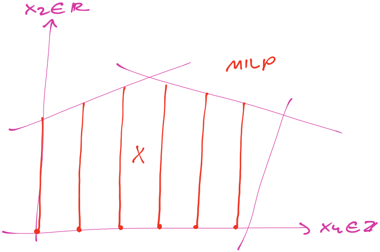
Theory: No algorithm can find, for every instance of 0-1-ILP (ILP/MILP), an optimal solution in polynomial time in the instance size, unless $\mathrm{P}{=}\mathrm{NP}$.

Practice: Many medium-size (M)ILPs are extremely challenging!

we probably we will not guarantee to get a global opt

Feasible regions of ILP/MILP:




(M)ILP is a powerful and versatile modeling/solution framework.

neither of those are convex

# 3.1.1 Modeling techniques and examples

- binary choice
- association between entities

linear models

- forcing constraints
- piecewise linear cost functions

mix of models

- modeling with exponentially many constraints
- disjunctive constraints
- linearizations

# 1) Binary choice

A binary variable allows to model a choice between two alternatives.

**Example 1: Knapsack problem**

Given

- $n$ objects
- profit $p_i$ and weight $a_i$ for each object $i$, with $1 \leq i \leq n$
- knapsack capacity $b$

decide which objects to select so as to maximize total profit while respecting the capacity constraint.

ILP formulation

variables

$x_i = \begin{cases} 1 & \text{if object } i \text{ was selected} \\ 0 & \text{otherwise} \end{cases}$

$\forall i = 1 \to n$

model

$\max \sum_{i=1}^{n} p_i x_i$

s.t. $\sum_{i=1}^{n} a_i x_i \leq b$ (capacity constraint)

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

Binary knapsack is $\mathrm{NP}$-hard.

## Example 2: Set Covering/Packing/Partitioning problems

Given

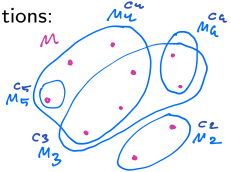
- groundset $M = \{1, 2, \ldots, m\}$ with $1 \leq i \leq m$,
- collection $\{M_1, \ldots, M_n\}$ of subsets indexed by $N = \{1, \ldots, n\}$ ($M_j \subseteq M$ for $j \in N$),
- a cost/weight $c_j$ for each $M_j$ with $j \in N$,

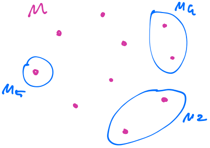
a subset of indices $F \subseteq N$ defines a

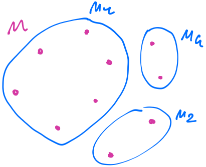
- **cover** of $M$ if $\cup_{j \in F} M_j = M$
- **packing** of $M$ if $M_{j_1} \cap M_{j_2} = \emptyset \ \forall j_1, j_2 \in F$, $j_1 \neq j_2$
- **partition** of $M$ if both a cover and a packing of $M$

Total cost/weight of a subset indexed by $F \subseteq N$ is $\sum_{j \in F} c_j$.

Illustrations:




$F_{cover} = \{1, 2, 3\}$ $F_{packing} = \{2, 4, 5\}$ $F_{part} = \{1, 2, 4\}$

**Set Covering problem**:

Given $M = \{1, 2, \ldots, m\}$, $\{M_1, \ldots, M_n\}$ indexed by $N = \{1, \ldots, n\}$, and a cost $c_j$ of $M_j$ for each $j \in N$, find a cover of $M$ with minimum total cost.

$A_j$ = vector of elements $\in M_j$

$A =$

ILP formulation

Parameters: incidence matrix $A = [a_{ij}]$ with $a_{ij} = 1$ if $i \in M_j$ and $a_{ij} = 0$ otherwise

Variables:

$x_j = \begin{cases} 1 & \text{if } M_j \text{ was selected} \\ 0 & \text{otherwise} \end{cases} \quad \forall j$

Model

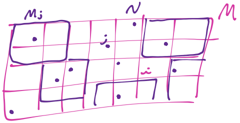
$$\min \sum_{j=1}^{n} c_j x_j$$

s.t. $\sum_j a_{ij} x_j \geq 1 \quad \forall i$ (covering constraint: each $i$ must $\in$ to at least one subset $M_j$)

$x_j \in \{0, 1\} \quad \forall j$

Set Covering is $\mathrm{NP}$-hard.

Application: Emergency service location (ambulances or fire stations)




$M = \{$ areas to be covered $\}$ and $N = \{$ candidate sites $\}$

$M_j = \{$ areas reachable in at most $\tau$ minutes from candidate site $j$ $\}$

Decide where to locate ambulances so as to minimize the total cost, while guaranteeing that the next call is served within $\tau$ minutes.

we want to select the most subsets (we max) just ensuring that they do not intersect

**Set Packing problem**:

$$\max\left\{\sum_{j=1}^{n} c_j x_j : A\underline{x} \leq \underline{1},\ \underline{x} \in \{0,1\}^n\right\}$$

where the $c_j$ represent "profits"

$\max \sum_j c_j x_j$

st $\sum_j a_{ij} x_j \leq 1 \quad \forall i$ (packing: each unit must be chosen at most once)

$x_j \in \{0,1\}$

Application: Combinatorial auctions (see introduction)

Determine the winner of each item so as to maximize total revenue:

$$\begin{array}{lll} \max & \sum_{S \subseteq M} b(S) x_S & \\ s.t. & \sum_{S \subseteq M \,:\, i \in S} x_S \leq 1 & \forall i \in M \\ & x_S \in \{0,1\} & \forall S \subset M. \end{array}$$

Set Packing is $\mathrm{NP}$-hard.

**Set Partitioning problem**:

$$\min \quad \text{or} \quad \max \left\{ \sum_{j=1}^{n} c_j x_j : A\underline{x} = \underline{1},\ \underline{x} \in \{0,1\}^n \right\}$$

where $c_j$s represent "costs" or "profits"

min/max $\sum_j c_j x_j$
st $\sum_j a_{ij} x_j = 1$ (partitioning: each $i$ must be chosen only one time)
$x_j \in \{0,1\}$

Application: Airline crew scheduling (see Computer Lab 3)

Given planning horizon.

eg (MXP → STC, 9.00 12.00) is compatible to (STC → BGY, 15.00 17.00)

$M = \{$ flight legs $\}$ single takeoff-landing phases to be carried out within a predefined time window.

$M_j = \{$ feasible subsets of flight legs $\}$ doable by same crew respecting all constraints (e.g., compatible flights, rest periods, total flight time,...).

Assign the crews to the flight legs so as to minimize total cost.

Other application: distribution planning (assign customers to routes)

Set Partitioning is $\mathrm{NP}$-hard.

# 2) Association between entities

Binary variables allow to model associations between two (several) entities.

**Example 3: Assignment problem** (or matching problem model)

Given

- $n$ projects and $n$ persons
- cost $c_{ij}$ for assigning project $i$ to person $j$, $\forall\, i, j \in \{1, \ldots, n\}$

decide which project to assign to each person so as to minimize the total cost while completing all projects.

Assumptions: every person can perform any project, and each person (project) must be assigned to a single project (person).

ILP formulation

Variables

$x_{ij} = \begin{cases} 1 & \text{if } i\text{-th project is assigned to the } j\text{-th person} \\ 0 & \text{oth.} \end{cases} \quad \forall i\ \forall j$

this problem is not NP hard

$\Rightarrow$ not all ILP problems are complex

Model

$\min \sum_{j=1}^{n} \sum_{i=1}^{n} c_{ij} x_{ij}$

st $\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i$ (each person must get a project)

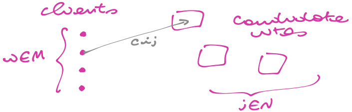
$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j$ (each project must get a person)

$x_{ij} \in \{0, 1\}$

# 3) Forcing constraints

To impose that "a decision X can be made $\boxed{\text{only if}}$ a decision Y has also been made".

**Example 4: Uncapacitated Facility Location (UFL)**




Given

- $M = \{1, 2, \ldots, m\}$ clients, $i \in M$
- $N = \{1, 2, \ldots, n\}$ candidate sites where a depot can be located, $j \in N$
- fixed cost $f_j$ for opening depot in $j$, $\forall j \in N$
- $c_{ij}$ transportation cost if the whole demand of client $i$ is served from depot $j$, $\forall i \in M$, $\forall j \in N$

localization & transportation (service)

decide where to locate the depots and how to serve the clients so as to minimize the total costs while satisfying all demands.

Illustration:

Variables

service: $x_{ij}$ = fraction of the demand of client $i$ served by depot $j$

$\in [0,1]$ $\forall i \; \forall j$

UFL is NP-hard.

$y_j = \begin{cases} 1 & \text{if depot } j \text{ is opened} \\ 0 & \text{otherwise} \end{cases}$

loc: $\in \{0,1\}$ $\forall j$

## MILP formulation

Variables:

- $x_{ij}$ = fraction of demand of client $i$ served by depot $j$, with $1 \leq i \leq m$, $1 \leq j \leq n$
- $y_j = 1$ if depot in $j$ is opened and $y_j = 0$ otherwise, with $1 \leq j \leq n$

Model $\quad \min \sum_i \sum_j \underbrace{c_{ij} x_{ij}}_{\text{cost for satisfying that fraction of the demand}} + \underbrace{\sum_j f_j y_j}_{\text{fixed cost of opening}}$

st $\quad \sum_{j \in M} x_{ij} = 1 \quad \forall i$ (satisfy all the demand of each client $i$)

$\sum_{i \in N} x_{ij} \leq m \cdot y_j \quad \forall j$ (variable link: $y_j = 0 \Rightarrow x_{ij} = 0 \ \forall i$; $\exists i: x_{ij} > 0 \Rightarrow y_j = 1$) we will discuss which one is more efficient

each of these at most is 1, and we are summing m times

$x_{ij} \leq y_j \quad \forall i \forall j$ (this also works, more constraints but simpler)

$0 \leq x_{ij} \leq 1 \quad \forall i \forall j$

$y_j \in \{0, 1\} \quad \forall j$

## Capacitated FL variant:

If $d_i$ demand of client $i$ and $k_j$ capacity of depot $j$, capacity constraints:

$\sum_{i \in N} d_i x_{ij} \leq k_j \cdot y_j$

# 4) Piecewise linear cost functions

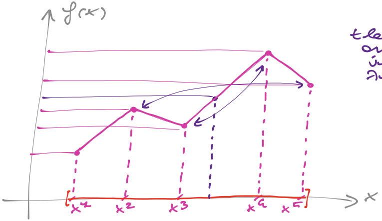
Continuous and binary variables allow to model nonconvex piecewise linear cost functions.

**Example 5: Minimization of piecewise linear cost functions**

Arbitrary such $f : \underbrace{[x^1, x^k]} \rightarrow \mathbb{R}$ specified by $(x^i, f(x^i))$ with $1 \leq i \leq k$ and $x^1 < \ldots < x^k$.

$f: I \to \mathbb{R}$ from that interval $I = [a,b]$

Illustration $\min_{x \in [x^1, x^k]} f(x)$:




the repr. is unique only if we allow just two consecutive λs to be non zero

0 0 0 λ λ 0 0 good
0 0 λ 0 0 λ 0 bad

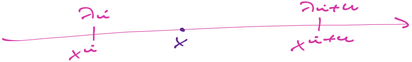
Any $x \in [x^1, x^k]$ and corresponding $f(x)$ can be expressed as

$$x = \sum_{i=1}^{k} \lambda_i x^i \quad \text{and} \quad f(x) = \sum_{i=1}^{k} \lambda_i f(x^i) \quad \text{with} \quad \sum_{i=1}^{k} \lambda_i = 1 \quad \text{and} \quad \underline{\lambda_1, \ldots, \lambda_k \geq 0},$$

convex comb.

Choice of $\lambda_i$s is unique if at most two consecutive $\lambda_i$ can be nonzero.

For any $x \in [x^i, x^{i+1}]$, $x = \lambda_i x^i + \lambda_{i+1} x^{i+1}$ with $\lambda_i + \lambda_{i+1} = 1$ and $\lambda_i \geq 0, \lambda_{i+1} \geq 0$.



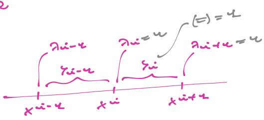

We can define $y_i = \begin{cases} 1 & \text{if } x \in [x^i, x^{i+1}] \\ 0 & \text{otherwise} \end{cases}$ $\forall i = 1 \to k-1$

(we are binary variable for each interval)

and the model of

$\min_{x \in [x^1, x^k]} f(x)$

can be reformulated as




$$\min \sum_{i=1}^{k} \lambda_i f(x^i)$$

st $\sum_{i=1}^{k} \lambda_i = 1$ (convex comb constraint)

$\sum_{i=1}^{k-1} y_i = 1$ (exactly one of the $y_i$ should be 1, just one interval must be lighted up)

$\lambda_i \leq \underbrace{y_{i-1} + y_i}_{\text{if this is } \begin{cases} 0 \Rightarrow \lambda_i = 0 \\ 1 \Rightarrow \lambda_i = 1 \end{cases}}$ $\forall i = 2, \dots, k-1$ (link among $\lambda_i$ and $y_i$)

$\lambda_1 \leq y_1$
$\lambda_k \leq y_{k-1}$ (boundary conditions)

$\lambda_i \geq 0$ $\forall i$
$y_i \in \{0, 1\}$ $\forall i$

# 5) Modeling with exponentially many constraints

**Example 6: Asymmetric Traveling Salesman Problem (ATSP)**

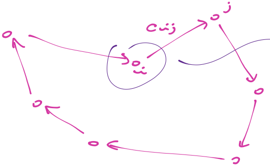
we have all the possible edges

Given

- a complete directed graph $G = (V, A)$ with $n = |V|$ nodes
- a cost $c_{ij} \in \mathbb{R}$ for each arc $(i, j) \in A$ (in case $c_{ij} = \infty$)

determine a *Hamiltonian circuit (tour)*, i.e., a circuit that visits exactly once each node, of minimum total cost.

Illustration:




hamiltonian circuit: start and come back at the same node

$(n-1)!$ Hamiltonian circuits

START NODE $\underbrace{\text{— — … — —}}_{}$ END NODE
(n-1 choices)

ATSP is $\mathrm{NP}$-hard.

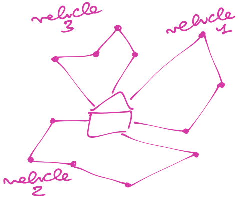
Applications: logistics, microchip manufacturing, scheduling, (DNA) sequencing,...

Also symmetric TSP version with undirected graph $G$.

**Website**: http://www.math.uwaterloo.ca/tsp/

Many variants with

- time windows (earliest and latest arrival time)
- precedence constraints
- capacity constraint
- several vehicles ("Vehicle Routing Problem" – VRP)
- ...

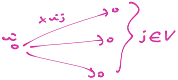


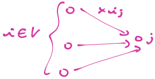
<u>Two ILP formulations</u>:

Variables

$x_{ij} = 1$ if the arc $(i,j)$ is included in the hamiltonian circuit

$\forall (i,j) \in A,\ x_{ij} \in \{0,1\}$



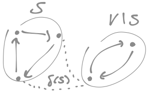




Model

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij} \qquad (2)$$

$$\text{st} \sum_{j \in V \setminus \{i\}} x_{ij} = 1 \quad \forall i \quad \text{(we select only 1 outgoing arc)} \qquad (3)$$

$$\sum_{i \in V \setminus \{j\}} x_{ij} = 1 \quad \forall j \quad \text{(select only 1 incoming arc)} \qquad (4)$$




there are missing arcs

we want to remove these sets, the non valid wrt constraints

we define

$\delta^+(S) = \{(i,j) \in A : i \in S, j \notin S\}$

directed cut induced by a set $S \subseteq V$

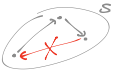
$$\sum_{(i,j)\in \delta^+(S)} x_{ij} \geq 1 \quad \forall S \subset V,\ S \neq \emptyset \quad \text{(cut set inequalities)}$$

we should pick at least one of the connecting arcs

an exponential # of constraints, $2^{|V|} - 1$

$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A$

equivalent idea:




we impose that if $S$ has $n$ vertices, at max we have $n-1$ arcs

## Alternative ILP formulation

Substitute cut-set inequalities with the *subtour elimination inequalities*:

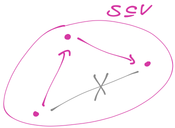
$$\sum_{(i,j)\in E(S)} x_{ij} \leq |S| - 1 \qquad \forall\ S \subseteq V, 2 \leq |S| \leq n-1 \tag{5}$$

where $E(S) = \{(i,j) \in A\ :\ i \in S, j \in S\}$ for $S \subseteq V$.

alternative formulation, but again we have an exp # of constraints

So which model is more efficient?

Illustration:




$|S|=3 \Rightarrow$ at most $\sum_{(i,j)\in E(S)} x_{ij} \leq 2$

# 6) Disjunctive constraints

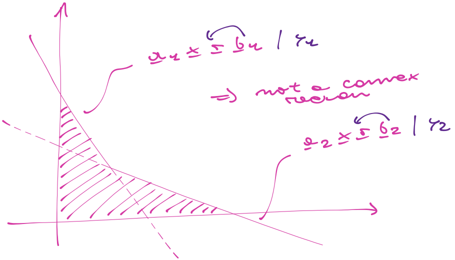
Binary variables allow to impose disjunctive constraints such as:

$$\textit{either} \quad \boxed{\underline{a}_1 \underline{x} \leq b_1} \quad \textit{or} \quad \boxed{\underline{a}_2 \underline{x} \leq b_2}$$

(or the disjunction of more inequalities)

with $\underline{x} \in \mathbb{R}$ and $\boxed{\underline{0} \leq \underline{x} \leq \underline{u}}$, where $\underline{u}$ is an upper bound vector.

Illustration:




idea: attach $y_i \in \{0,1\}$ for each of the constraints $\underline{a}_i \underline{x} \leq \underline{b}_i$ (here $i=1,2$)

then consider these constraints

$$\underline{a}_i \underline{x} - \underline{b}_i \leq M \cdot (1 - y_i)$$

only if $y_i = 1$, to let the constraint be as it was before if $y_i = 1$

big $M$, to make the inequality irrelevant (always true) if $y_i = 0$

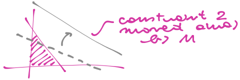
$y_1 + y_2 = 1$ (meet only 1 constraint)

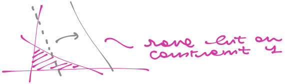
$y_i \in \{0,1\}$

$\underline{0} \leq \underline{x} \leq \underline{u}$

how to choose the big $M$?

$$M \geq \max_i \left( \underline{a}_i \underline{x} - \underline{b}_i : \underline{0} \leq \underline{x} \leq \underline{u} \right)$$



## **Example 7: Scheduling problem** (see Computer Lab 0)

Given

- $m$ machines and $n$ products
- for each product $j$, deadline $d_j$ and processing time $p_{jk}$ on machine $k$, with $1 \leq k \leq m$,

determine a schedule which minimizes the time needed to complete all products, while satisfying all deadlines.

Products cannot be processed simultaneously on the same machine.

# 7) Linearization of products of variables

very important

- Product of two (several) binary variables:

$\boxed{z = y_1 \cdot y_2}$, with $y_i \in \{0,1\}$ for $i = 1,2$ and $z \in \{0,1\}$, can be replaced by

introduce an auxiliary variable $z$ and then add the following constraints:

$y_1 = 0 \Rightarrow z = 0$
$y_2 = 0 \Rightarrow z = 0$
$y_1 = 1$ & $y_2 = 1 \Rightarrow z = 1$

$z \leq y_1$
$z \leq y_2$
$z \geq y_1 + y_2 - 1$

($2z \leq y_1 + y_2$ this one?)

extension to $n$ variables $\quad z = \prod_{i=1}^{n} y_i$

- Product of a binary variable and a bounded continuous variable:

$\boxed{z = x \cdot y}$, with $x \in [0,u]$, $y \in \{0,1\}$ and $z \in [0,u]$, can be replaced by

we can still do it without any approximation using the constraints:

$z = x \cdot y \leq x \cdot 1$
$x \in [0,u]$ & $y \in \{0,1\}$ $z = x \cdot y \leq u \cdot y$
$y = 0 \Rightarrow z = 0$

$z \leq x$ — lower bounds are easy
$0 \leq z \leq uy$
$z \geq \underline{x - (1-y)u}$
$= \begin{cases} \text{if } y=0 & x - u \leq 0 \Rightarrow z \geq 0 \\ \text{if } y=1 & x \text{ (constraint 1)} \end{cases}$

Question: If $x_1$ and $x_2$ are continuous and bounded, can $x_1 \cdot x_2$ be exactly linearized? NO

# 3.2 Strong and ideal formulations

LP

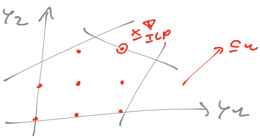
In linear optimization, good formulations contain a small number of variables $n$ and constraints $m$ because the complexity of algorithms grows polynomially in $n$ and $m$.

The choice of the formulation does not critically affect the possibility of solving LPs.

For ILPs and MILPs, the choice of the formulation is crucial.

# 3.2.1 Alternative and strong formulations

if there was no ≥ (so we go to ILP) the feasible region would look like




**Definition**: Given any MILP

$$
\begin{aligned}
z_{MILP} = \min \quad & \cancel{\underline{c}_1^t \underline{x}} + \underline{c}_2^t \underline{y} \\
s.t. \quad & \cancel{A_1 \underline{x}} + A_2 \underline{y} \geq \underline{b} \\
& \cancel{\underline{x} \geq \underline{0}} \\
& \boxed{\underline{y} \geq \underline{0} \text{ integer}}
\end{aligned}
$$

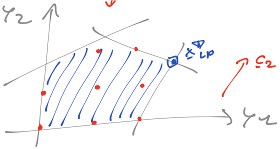
its **linear programming** (LP) **relaxation** is

we relax the integrality condition

$$
\begin{aligned}
z_{LP} = \min \quad & \underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y} \\
s.t. \quad & A_1 \underline{x} + A_2 \underline{y} \geq \underline{b} \\
& \underline{x} \geq \underline{0}, \ \boxed{\underline{y} \geq \underline{0},}
\end{aligned}
$$

while in the relaxation we would get

where $y_j \in \mathbb{Z}$ are omitted for all $j$.




If $y_j \in \mathbb{Z}$ with $0 \leq y_j \leq u_j$, then in LP relaxation $y_j \in [0, u_j]$.

Illustration:

$\Rightarrow \boxed{z_{MILP} \geq z_{LP}}$

as in the relaxation we get more chances so $z_{LP}$ is better

Obviously $X_{MILP} \subseteq X_{LP}$ where

$$X_{MILP} = \{(\underline{x}, \underline{y}) \in \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2} \ : \ A_1\underline{x} + A_2\underline{y} \geq \underline{b}, \ \underline{x} \geq \underline{0}, \ \underline{y} \geq \underline{0}\}$$
$$X_{LP} = \{(\underline{x}, \underline{y}) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \ : \ A_1\underline{x} + A_2\underline{y} \geq \underline{b}, \ \underline{x} \geq \underline{0}, \ \underline{y} \geq \underline{0}\}$$

zLP was better

**Proposition**: For any *minimization* MILP, we have:

- $\boxed{\mathbf{z_{LP}} \leq \mathbf{z_{MILP}},}$
- if optimal solution $(\underline{x}^*_{LP}, \underline{y}^*_{LP})$ of LP relaxation is integer (feasible for MILP), it is also optimal for MILP.

For *maximization* problems, $\mathbf{z_{MILP}} \leq \mathbf{z_{LP}}$.

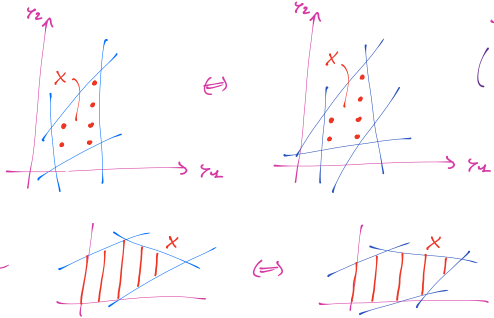
**Definition**:

Restricted to the integer MILP domain

A polyhedron $P = \{(\underline{x}, \underline{y}) \in \mathbb{R}^{n_1+n_2} \; : \; A_1\underline{x} + A_2\underline{y} \geq \underline{b}, \; \underline{x} \geq \underline{0}, \; \underline{y} \geq \underline{0}\} \subseteq \mathbb{R}^{n_1+n_2}$ is a **formulation** for a mixed integer set $X \subseteq \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ if and only if $\boxed{X = P \cap (\mathbb{R}^{n_1} \times \mathbb{Z}^{n_2})}$.

Illustrations:

ILP case




equivalent formulation
(but the LP relaxation is different)

both set of inequalities contain the same integer point set, but the LP relaxation will be different

MILP example

**Observation**: Any MILP admits an infinite number of *alternative formulations*.
Equivalent from MIP point of view but different LP relaxations.

Examples:

1) Two alternative formulations for A TSP (cut-set or subtour-elimination constraints).

2) Original formulation for UFL:

cost for serving clients — cost of depot opening

$$
\begin{array}{lll}
\min & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{j=1}^{n} f_j y_j & \\
s.t. & \sum_{j=1}^{n} x_{ij} = 1 & \forall i \in M \\
 & \sum_{i=1}^{m} x_{ij} \leq m y_j & \forall j \in N \qquad (1) \\
 & y_j \in \{0,1\} & \forall j \in N \\
 & 0 \leq x_{ij} \leq 1 & \forall i \in M, j \in N.
\end{array}
$$

Alternative formulation: $n$ linking constraints (1) are substituted with $mn$ ones

$$x_{ij} \leq y_j \quad \forall i \in M, j \in N. \qquad (2)$$

dis-aggregate formulation

question: which one is better from a computational pov?

**Definition**:


Given a mixed integer set $X \subseteq \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ and two formulations $P_1$ and $P_2$ for $X$, $P_1$ is **stronger** than $P_2$ if $\boxed{P_1 \subset P_2}$.


smaller region means stronger

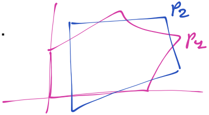
The lower bound provided by LP relaxation of $P_1$ is not smaller (weaker) than that of $P_2$:

we say $P_1$ is stronger as we are getting an opt set reached on lower values

$\Rightarrow$ is closer to the integer set then

$$
\begin{aligned}
z_{MILP} &= \min\{\underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y} \,:\, (\underline{x}, \underline{y}) \in X\} \\
&\geq \min\{\underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y} \,:\, (\underline{x}, \underline{y}) \in P_1\} \\
&\geq \min\{\underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y} \,:\, (\underline{x}, \underline{y}) \in P_2\}.
\end{aligned}
$$

Two formulations may not be comparable.



Examples:

1) **Uncapacitated Facility Location** (UFL)

**Proposition**: The LP relaxation of the MILP formulation with constraints $x_{ij} \leq y_j$ is stronger than that with aggregated constraints $\sum_{i=1}^{m} x_{ij} \leq m y_j$.

Proof:

dis-aggr

$$P_1 = \left\{(\underline{x}, \underline{y}) \in \mathbb{R}^{mn+n} \ : \ \sum_{j=1}^{n} x_{ij} = 1 \ \forall i, \ x_{ij} \leq y_j \ \forall i \forall j, \ \underbrace{0 \leq x_{ij} \leq 1 \ \forall i \forall j, \ 0 \leq y_j \leq 1 \ \forall j}_{\text{relaxation}} \right\}$$

$$P_2 = \left\{(\underline{x}, \underline{y}) \in \mathbb{R}^{mn+n} \ : \ \sum_{j=1}^{n} x_{ij} = 1 \ \forall i, \ \underbrace{\sum_{i=1}^{m} x_{ij} \leq m y_j \ \forall j}_{\text{aggregate}}, \ 0 \leq x_{ij} \leq 1 \ \forall i \forall j, \ 0 \leq y_j \leq 1 \ \forall j \right\}$$

Obviously $P_1 \subseteq P_2$. all vectors of $P_1$ satisfy the constraint of $P_2$. I) assume these constraints of $P_1$ $x_{ij} \leq y_j$ true in $P_1$ $\Rightarrow$ $|_+$ $\sum_i x_{ij} \leq m y_j$ also true in $P_2$

Exhibit $(\underline{x}, \underline{y}) \in P_2 \setminus P_1$:

Suppose that $m = kn$, with $k \in \mathbb{N}_2$ (i.e. $k \geq 2$ integer)

Example
$m = 6$ clients
$n = 3$ sites
$k = 2$

Let each depot serve $k$ clients. Then

$$x_{ij} = \begin{cases} 1 & \text{if } i = k(j-1)+1, \dots, k(j-1)+k \\ 0 & \text{otherwise} \end{cases}$$

and about the $y$ we can set

$y_j = \frac{k}{m} \quad \forall j = 1, \dots, n$

this point set $\in P_2$ but not on $P_1$, since each $x_{ij}$ is 1, so never $\leq y_j$

## 2) **Symmetric TSP (STSP)**

now we have edges, no more arcs

STSP: Given undirected $G = (V, E)$ and cost $c_e$ for every $e = \{i, j\} \in E$, determine a **Hamiltonian cycle** of $G$ (i.e., visiting each $i \in V$ exactly once) of minimum total cost.

no circuit so with edges there is no orientation

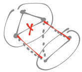
$x_e$ — $x_e = 1$ if edge is chosen

Two alternative formulations:

each node has 2 selected incident edges ($\Rightarrow$ degree constraint)

$$
\begin{array}{llll}
\min & \sum_{e \in E} c_e x_e & & \\
\text{s.t.} & \sum_{e \in \delta(i)} x_e = 2 & i \in V & (DEG) \\
 & \sum_{e \in \delta(S)} x_e \geq 2 & S \subset V,\ S \neq \emptyset & (CUT) \\
 & x_e \in \{0, 1\} & e \in E &
\end{array}
$$

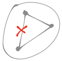
relax this [illegible] $(1 \geq)\ x_e \geq 0\ \forall e$



for the subtours case we need 2 edges (at least) or we no more create a circuit, just a cycle?)

where $\delta(S) = \{\{i, j\} \in E : i \in S, j \in V \setminus S\}$, $\delta(i) = \delta(\{i\})$

$$
\begin{array}{llll}
\min & \sum_{e \in E} c_e x_e & & \\
\text{s.t.} & \sum_{e \in \delta(i)} x_e = 2 & i \in V & (DEG) \\
 & \sum_{e \in E(S)} x_e \leq |S| - 1 & S \subset V, |S| \geq 2 & (SEC) \\
 & x_e \in \{0, 1\} & e \in E, &
\end{array}
$$



where $E(S) = \{\{i, j\} \in E : i \in S, j \in S\}$.

(DEG), (SEC) and (CUT) are, respectively, the *degree*, *subtour-elimination* and *cut-set* constraints.

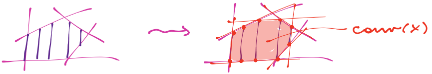
Let $P_{sec}$ and $P_{cut}$ be the polyhedra (feasible regions) of the respective LP relaxations.

**Proposition**: The two formulations are equally strong ($P_{sec} = P_{cut}$).

See Exercise 2.3

## 3.2.2 Ideal ILP formulations

**Theorem (Meyer)**: Let $X \subseteq \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ be mixed integer feasible set of any MILP with rational coefficients, then $\boxed{conv(X)}$ is a rational polyhedron. Moreover, all extreme points of $conv(X)$ belong to $X$.




For bounded integer $X$, intuitive and no need for rational coefficients assumption.

**Consequence**:

$$\min\{\underline{c}^t\underline{x} \ : \ \underline{x} \in X\} = \min\{\underline{c}^t\underline{x} \ : \ \underline{x} \in conv(X)\}$$

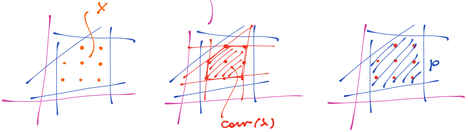
but often clearly determine what is conv(X) is complex

If we knew $conv(X)$ explicitly, we could solve the (M)ILP by solving a *single* Linear Program!

⇒ is a term to simplify the art of a MILP, to solve it directly (without relaxation eg)

so for the ideal formulation we want to squeeze P to be conv(X)

Clearly feasible region $P$ of LP relaxation of any formulation satisfies $X \subseteq conv(X) \subseteq P$.



**Definition**: Let $X \subseteq \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ be any mixed integer feasible set, the **ideal** (perfect) **formulation** for $X$ is the polyhedron $P \subseteq \mathbb{R}^{n_1+n_2}$ with $\boxed{P = conv(X)}$

Since it is often of exponential size or difficult to determine, we strive for strong formulations.

Examples:

1) **Assignment problem**

Natural ILP formulation:

$$\begin{array}{lll} \min & \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} & \\ s.t. & \sum_{i=1}^{n} x_{ij} = 1 & \forall j \\ & \sum_{j=1}^{n} x_{ij} = 1 & \forall i \\ & x_{ij} \in \{0,1\} & \forall i, \forall j \end{array}$$

**Proposition**:

$$P = \{\underline{x} \in \mathbb{R}^{n^2} : \sum_{i=1}^{n} x_{ij} = 1 \; \forall j, \; \sum_{j=1}^{n} x_{ij} = 1 \; \forall i, \; \underbrace{0 \le x_{ij} \le 1}_{\text{LP relaxation}} \; \forall i, j\}$$

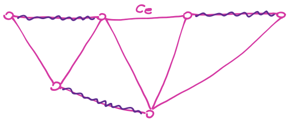
is an *ideal formulation* for the Assignment problem.

Proof later

2) **Perfect Matching problem (PM)**

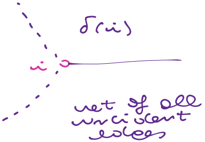
PM: Given an undirected $G = (V, E)$ with $n = |V|$ even and a cost $c_e$ for each $e = \{i, j\} \in E$, determine a **perfect matching** (i.e., subset of edges without common nodes but incident to all nodes) of minimum total cost.

Illustration:




solution exists - intersecting couples of vertices, by choosing some edges

Constraint:
for each vertex/node we must select just one incident node




actually we also add another constraint

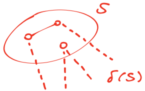
Natural ILP formulation:

$$
\begin{array}{lll}
\min & \sum_{e \in E} c_e x_e & \\
s.t. & \sum_{e \in \delta(i)} x_e = 1 & \forall i \in V \\
 & x_e \in \{0, 1\} & \forall e \in E,
\end{array}
$$

where $x_e = 1$ if $e$ is selected, and $x_e = 0$ otherwise.

Clearly all $\underline{x} \in \{0,1\}^{|E|}$ corresponding to perfect matchings satisfy:

$$\sum_{e \in \delta(S)} x_e \geq 1 \qquad \forall S \subset V \;\text{ with } |S| \text{ odd.}$$




**Theorem** (Edmonds):

$$P_M = \{\underline{x} \in \mathbb{R}^{|E|} \;:\; \underbrace{\textstyle\sum_{e \in \delta(i)} x_e = 1}_{\text{constr 1}} \;\forall i \in V,\; \underbrace{\textstyle\sum_{e \in \delta(S)} x_e \geq 1 \;\forall S \subset V,\; |S| \text{ odd},}_{\text{constr 2}} \; 0 \leq x_e \leq 1 \;\forall e \in E\}$$

is an ideal formulation for the Perfect Matching problem.

## 3.2.3 Extended formulations

Alternative formulations can use additional and/or different variables.

**Definition**: The formulations including additional variables, are **extended formulations**.

Example: **Uncapacitated Lot-Sizing (ULS)**

One type of product and $n$ periods.

time span of the problem

Given

- $f_t$ fixed cost for producing during period $t$ → needed to set a binary variable (like to tell (which) if we produced something or not)
- $p_t$ unit production cost in period $t$
- $h_t$ unit storage cost in period $t$
- $d_t$ demand in period $t$

determine a production plan for the next $n$ periods that minimizes the total costs, while satisfying demands.

Assumption: stock is empty at the beginning and at the end.

## MILP formulation

cost and natural formulation

$p_1$

Variables:

- $x_t$ = amount produced in period $t$, with $1 \le t \le n$
- $y_t = 1$ if production occurs in period $t$ and $y_t = 0$ otherwise, with $1 \le t \le n$
- $s_t$ = amount in stock at the end of period $t$, with $0 \le t \le n$

our modeling choice

Model

$$\min \sum_t \big( \underbrace{p_t x_t}_{\text{production}} + \underbrace{h_t s_t}_{\text{storage}} + \underbrace{f_t y_t}_{\text{fixed cost}} \big)$$

st $\quad s_t = s_{t-1} + x_t - d_t \quad$ (balance constraint & demand guarantee) $\forall t \ge 1$

$\left(\begin{array}{l} x_t > 0 \Rightarrow y_t = 1 \\ y_t = 0 \Rightarrow x_t = 0 \end{array}\right) \quad x_t \le M \cdot y_t, \quad M = \sum d_t \quad \forall t \quad$ ($x_t$ and $y_t$ link)

issue: this big M can be really large, so we could get into numerical problems

$s_0 = 0$
$s_n = 0$

$s_t, x_t \ge 0 \quad \forall t$
$y_t \in \{0,1\} \quad \forall t$

idea of this model

$x_t$

1 $\quad$ t $\quad$ n

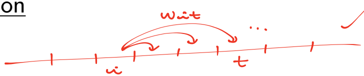
what's the idea of the extended formulation?

Extension with minimum lot sizes.

# MILP extended formulation

$P_2$

## Variables:




we we consider disaggregated variables (which are "complete", also more for a real production)

$w_{it}$ = amount produced in period $i$ to satisfy the demand of period $t$, for $1 \le i \le t \le n+1$

$\ge 0$

$y_t = \begin{cases} 1 & \text{if production occurs at period } t \\ 0 & \text{otherwise} \end{cases}$

( $x_i = \sum_{t=i}^{n} w_{it}$ $\Rightarrow$ this is why it is an extended formulation )

Model

$$\min \sum_{i=1}^{n} \sum_{t=i}^{n} c_{it} w_{it} + \sum_{t=1}^{n} f_t y_t$$

$$\left( c_{it} = p_i + \sum_{k=i}^{t-1} h_k \right)$$

st $\sum_{i=1}^{t} w_{it} = d_t \quad \forall t$ (demand)

$\sum_{i=1}^{n} w_{i,n+1} = 0$ (stock empty at the end)

$w_{it} \le \underbrace{d_t}_{\text{much better big M constraint}} y_t$ (variable link)

$w_{it} \ge 0 \quad \forall i \; \forall t$

$y_t \in \{0,1\} \quad \forall t$

the extended ILP obtained formulation in each period is obtained w the constraints

$$x_i = \sum_{t=i}^{n} w_{it} \quad \forall i$$

$$s_i = \sum_{j=1}^{i} \sum_{t=i+1}^{n+1} w_{jt} \quad \forall i$$

# 3.2.4 Comparison between formulations

Consider an ILP formulation

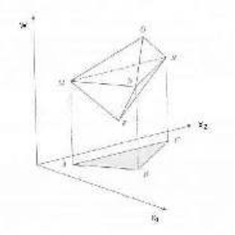
$$\min\{\underline{c}^t\underline{x} \ : \ \underline{x} \in P_1 \cap \mathbb{Z}^n\}$$

with $\boxed{P_1 \subseteq \mathbb{R}^n}$, and an extended formulation

$$\min\{\underline{c}^t(\underline{x}, \underline{w}) \ : \ (\underline{x}, \underline{w}) \in P_2 \cap (\mathbb{Z}^n \times \mathbb{R}^{n'})\}$$

with $\boxed{P_2 \subseteq \mathbb{R}^n \times \mathbb{R}^{n'}}$. the new added variables

**Definition**: Given a polyhedron $P \subseteq \mathbb{R}^n \times \mathbb{R}^{n'}$, the **orthogonal projection of** $P$ onto the $x$-subspace $\mathbb{R}^n$ is the polyhedron $\text{proj}_{\underline{x}}(P) = \{\underline{x} \in \mathbb{R}^n \ : \ \exists\, \underline{w} \in \mathbb{R}^{n'} \text{ s.t. } (\underline{x}, \underline{w}) \in P\ \}$.



To compare $P_1$ and extended formulation $P_2$, we compare $P_1$ and $\text{proj}_{\underline{x}}(P_2)$.

One way to determine the orthogonal projection of polyhedra onto subspaces:

problem: how to derive/compute the orthogonal projection?

**Fourier-Motzkin elimination method** (1827)

Goal: find a feasible solution of $A\underline{x} \geq \underline{b}$ with $A \in \mathbb{R}^{m \times n}$.

Idea: At each iteration eliminate one variable $x_i$ (derive an equivalent linear system without $x_i$), stop when a single variable is left.

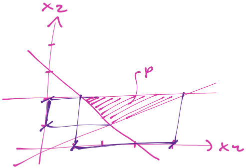
Given $A\underline{x} \geq \underline{b}$, suppose we wish to eliminate $x_i$.

The equivalent system without $x_i$ includes

- all inequalities of $A\underline{x} \geq \underline{b}$ in which $x_i$ does not appear,
- the inequalities resulting from all the possible combinations of the upper and lower bounds on $x_i$ implied by $A\underline{x} \geq \underline{b}$.

Example: $P$ defined by

$$\begin{array}{rrl} x_1 & +x_2 & \geq 3 \qquad (3) \\ -\frac{1}{2}x_1 & +x_2 & \geq 0 \qquad (4) \\ & -x_2 & \geq -2 \qquad (5) \end{array}$$




Eliminate $x_2$ (project $P$ onto subspace of $x_1$):

equivalent expr, but now we want to remove x2 (it's just a rewriting for now)

$$\left\{ \begin{array}{rcl} 3 - x_1 \leq & x_2 & \\ \frac{1}{2}x_1 \leq & x_2 & \\ & x_2 & \leq 2 \end{array} \right.$$

then we consider all the possible pairs of lower and upper bounds

and obtain

$$\left. \begin{array}{rcl} 3 - x_1 & \leq & 2 \\ \frac{1}{2}x_1 & \leq & 2, \end{array} \right\}$$

$x_1 \geq 1$
$x_1 \leq 4$

hence the projection $[1, 4]$.

Eliminate $x_1$ (project $P$ onto subspace of $x_2$): obtain $1 \leq x_2 \leq 2$, hence the projection $[1, 2]$.

Complexity: Since at each step an inequality is derived for each pair of upper-lower bounds, the number of constraints can grow exponentially in $n$.

so actually, there are also other methods

**Comparing ULS formulations**:

Consider the formulation $\boxed{P_1}$:

$M > \sum_{t=1}^{n} d_t$

$$
\begin{aligned}
s_t &= s_{t-1} + x_t - d_t && \forall t \\
x_t &\leq M y_t && \forall t \qquad (6)\\
s_0 = 0, s_t \geq 0, x_t \geq 0,\ & \underline{0 \leq y_t \leq 1} && \forall t
\end{aligned}
$$

$P_1$ relaxed version

and $\text{proj}_{\underline{x},\underline{s},\underline{y}}(P_2)$, with $\boxed{P_2}$ defined by

$$
\begin{aligned}
\textstyle\sum_{i=1}^{t} w_{it} &= d_t && \forall t \\
w_{it} &\leq d_t y_i && \forall i, t, 1 \leq i \leq t \qquad (7)\\
x_i &= \textstyle\sum_{t=i}^{n} w_{it} && \forall i \qquad (8)\\
s_i &= \textstyle\sum_{l=1}^{i} \sum_{t=i+1}^{n+1} w_{lt} && \forall i \qquad (9)\\
w_{it} &\geq 0 && \forall i, t, 1 \leq i \leq t \\
\underline{0 \leq y_t \leq 1} & && \forall t.
\end{aligned}
$$

$x_i = \sum_{t=i}^{n} w_{it} \leq \underbrace{\sum_{t=i}^{n} d_t}_{< M} y_i < M y_i$

$\Rightarrow$ $P_2$ is more/less contained in $P_1$

eliminating the $w_{it}$ as we did here

Easy to verify that $\boxed{\text{proj}_{\underline{x},\underline{s},\underline{y}}(P_2) \subset P_1}$;

eg the point $x_t = d_t$, $y_t = d_t/M$ $\forall t$
- is an (extreme) point of $P_1$
- but $\notin \text{proj}_{x,s,y}(P_2)$

$x_i = d_t \overset{?}{\leq} M y_t = M \frac{d_t}{M} = d_t$
$\Rightarrow$ no

**Proposition**: $P_2$ is the ideal formulation of ULS.

# 3.2.5 Stronger extended formulations

Look for an extended formulation whose projection is a better approximation of the ideal formulation.

Example: **Fixed charge network flow problem (FCNF)**:

Given a directed $G = (V, A)$ with

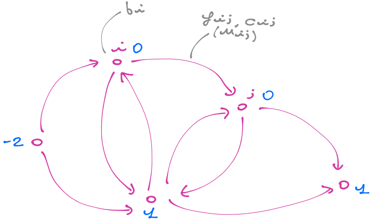
linear, nonnegative

all flow sources goes into the destinations

- for each $(i, j) \in A$ a fixed cost $f_{ij} > 0$, unit cost $c_{ij}$ and a capacity $u_{ij}$,
- for each $i \in V$ a demand $b_i$ ($b_i < 0$ sources, $b_i > 0$ destinations) with $\sum_{i \in V} b_i = 0$,

determine a feasible flow of minimum total cost which satisfies all demands and capacity constraints.

Illustration:




problem like min cut, max flow-min cut, but this is an extension (more complex, NP-hard)

FCNF is $\mathrm{NP}$-hard.

Natural MILP formulation:

i.e. of where actually that amount will go. In the extended notes we take with unit / we specify in a different way the flow distribution

Variables:

- $x_{ij}$ = amount of flow through $(i,j)$, for all $(i,j) \in A$
- $y_{ij} = 1$ if $(i,j)$ is used and $y_{ij} = 0$ otherwise, for all $(i,j) \in A$

Model

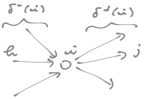
we should always have a balance between incoming and outcoming flow

$$\min \sum_{(i,j)\in A} \left[ c_{ij} x_{ij} + f_{ij} y_{ij} \right]$$

$$\text{st} \quad \forall i \in V \quad \underbrace{\sum_{\substack{(h,i)\in \\ \delta^-(i)}} x_{hi}}_{\text{what enters}} - \underbrace{\sum_{\substack{(i,j)\in \\ \delta^+(i)}} x_{ij}}_{\text{what exits}} = b_i \quad \text{(balance constr)} \qquad (10)$$

$$x_{ij} \leq c_{ij} y_{ij} \quad \text{(capacity constr \& link variables)} \qquad (11)$$

$$x_{ij} \geq 0$$
$$y_{ij} \in \{0,1\}$$




LP relaxation yields poor bounds because of weak coupling between $x_{ij}$s and $y_{ij}$s via (11).

Multi-commodity extended MILP formulation:

Idea: Suppose w.l.o.g. $\exists$ single source node $s$ ($b_s = -\sum_{i \in V \setminus \{s\}} b_i$) and decompose the flows according to their destinations.

destination nodes

Denote $K = \{i \in V \,:\, b_i > 0\} \subseteq V$.

flow passing through arc (i,j) that is destined to node k

Define one "commodity" for each $k \in K$, with the flow variables $x_{ij}^k$ for all $(i,j) \in A$.

Define $d_i^k = -b_k$ if $i = s$, $d_i^k = b_k$ if $i = k$, and $d_i^k = 0$ otherwise.

... see **Computer Lab 1**

Significantly stronger formulation of FCNF with $|K|$ times more variables/constraints.

# 3.2.6 Remarks on the strength and size of formulations

**Definition**: A **compact formulation** is a formulation with a number of variables/constraints polynomial w.r.t. the instance size.

**Remark 1**: A compact extended formulation can be much weaker than an alternative exponential-size formulation. models having more constraints are (often) better

Example: ATSP

To exclude subtours, instead of (SEC) one can add, for each $i \in V$, a variable $t_i$ representing the "position" in which node $i$ is visited in the tour and a set of constraints.

... see **Computer Lab 1**

**Remark 2**: A compact extended formulation can have a projection into the space of the natural variables that is of exponential size.

Example: ATSP

# 3.3 "Easy" ILP problems and totally unimodular matrices

Generic ILP

$$\min\{\underline{c}^t\underline{x} \;:\; \boxed{A\underline{x} = \underline{b},\ \underline{x} \in \mathbb{Z}^n_+}\} \tag{1}$$

more variables than constraints

where $A \in \mathbb{Z}^{m\times n}$ with $n \geq m$, and $\underline{b} \in \mathbb{Z}^m$.

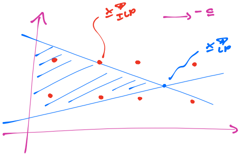
$P(\underline{b}) = \{\underline{x} \in \mathbb{R}^n \;:\; A\underline{x} = \underline{b}, \underline{x} \geq \underline{0}\}$ polyhedron of LP relaxation.

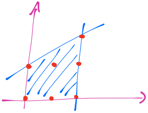
Assumption: rank$(A)=m$, i.e, $\nexists$ redundant constraints.

In general, optimal solutions of LP relaxation are far away from those of (1).

e.g. here rounding leads even to an unfeasible set

Illustration:



ideal if when the vertices of the LP relaxation (i.e. of P) are integer

If all vertices of $P(\underline{b})$ are integral, ideal formulation and just need to solve LP relaxation.

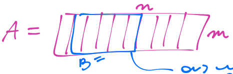
According to **Linear Programming theory**:

- Any LP $\min\{\underline{c}^t \underline{x} \ : \ A\underline{x} = \underline{b}, \ \underline{x} \geq \underline{0}\}$ with a finite optimal solution has an optimal vertex (extreme point). but this was a geometric characterization, now we now define an algebraic one

- To each vertex of $P(\underline{b})$ corresponds (at least) one **basic feasible solution**

$$\underline{x} = (\underline{x}_B, \underline{x}_N) = (B^{-1}\underline{b}, \underline{0}),$$

where $B$ is a **basis** of $A$, i.e., an $m \times m$ non-singular submatrix of $A$.




$\rightarrow A = \left( B \ \vdots \ N \right) \quad \underline{x} = \begin{pmatrix} \underline{x}_B \\ \cdots \\ \underline{x}_N \end{pmatrix}$

any subset of m cols which are linearly ⊥

we can do this by swapping the order of the variables and constraints

Consider any basis $B$.

By partitioning columns of $A$ into basic and non basic, $A\underline{x} = \underline{b}$, $\underline{x} \geq \underline{0}$ can be written as

$$B\underline{x}_B + N\underline{x}_N = \underline{b} \ \text{ with } \underline{x}_B \geq \underline{0} \ \text{ and } \ \underline{x}_N \geq \underline{0},$$

and in canonical form:

$$\underline{x}_B = B^{-1}\underline{b} - B^{-1}N\underline{x}_N \ \text{ with } \underline{x}_B \geq \underline{0} \ \text{ and } \ \underline{x}_N \geq \underline{0},$$

which emphasizes the basic feasible solution $\boxed{(\underline{x}_B, \underline{x}_N) = (B^{-1}\underline{b}, \underline{0})}$.

**Observation**: If an optimal basis $B$ of LP relaxation of (1) has $det(B) = \pm 1$, then $(\underline{x}_B, \underline{x}_N) = (B^{-1}\underline{b}, \underline{0})$ is integral and also optimal for ILP (1).

Proof: Recall that $B^{-1} = \frac{1}{det(B)} \cdot C^T$, where C is the cofactor matrix we set as

$C = [\alpha_{ij}] = (-1)^{i+j} det(B_{ij})$ where $B_{ij}$ is B removed of row i and col j

Since B contains integer entries (as the coeffs of A were integers), then also the cofactors $\alpha_{ij}$ are integers.

if $det(B) = \pm 1 \Rightarrow B^{-1}$ is also integer, and since also $\underline{b}$ is integer then we get that $(x_B, x_N) = (B^{-1}\underline{b}, \underline{0})$ is integer.

Only a sufficient condition for integrality of $(\underline{x}_B, \underline{x}_N) = (B^{-1}\underline{b}, \underline{0})$.

$B^{-1}\underline{b}$ integral also if $det(B) = 2$ and all $b_i \in \mathbb{Z}$ are even.

# 3.3.1 Totally unimodular matrices and optimal integer solutions

**Definition**: $A \in \mathbb{Z}^{m \times n}$ is **totally unimodular** (TU) if every squared submatrix has a determinant $-1$, 0 or 1.

Clearly, if $A$ is TU, $a_{ij} \in \{-1, 0, 1\}$ for all $i$ and $j$.

Examples:

$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ yes TU

$\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$ no TU

Recall: For any $B \in \mathbb{R}^{m \times m}$, *Laplace expansion* along any row $i$, $1 \leq i \leq m$, is

$det(B) = \sum_{j=1}^{m} b_{ij}\alpha_{ij}$, where $\alpha_{ij} = (-1)^{i+j} det(B_{ij})$ are the cofactors of $B$.

Expansion also along any column $j$.

**Proposition**:

- $A$ is TU if and only if $\boxed{A^t}$ is TU.
- $A$ is TU if and only if $\boxed{(A \,|\, I_m)}$ is TU. $\left( A \;\vdots\; \begin{matrix}1 & & \\ & \ddots & \\ & & 1\end{matrix} \right)$
- $A'$ obtained from $A$ by permuting/changing the sign of some columns/rows is TU if and only if $A$ is TU.

**Theorem 1**:

the feasible region polyhedron

If $\boxed{A} \in \mathbb{Z}^{m \times n}$ TU, $\boxed{\underline{b}}$ integral and $\boxed{P(\underline{b})} = \{\underline{x} \in \mathbb{R}^n \; : \; A\underline{x} = \underline{b}, \underline{x} \geq \underline{0}\} \neq \emptyset$, then all extreme points of $P(\underline{b})$ are integral.

Proof: See observation.

From ILP point of view, if $A$ is TU it suffices to solve the LP relaxation.

**Corollary**:

If $A \in \mathbb{Z}^{m \times n}$ TU, $\underline{b}$ integral and

so TU is very useful, works both in equality and inequality constraints

(=) or (≥)

$$\boxed{P(\underline{b})} = \{\underline{x} \in \mathbb{R}^n \ : \ \boxed{A\underline{x} \geq \underline{b}}, \underline{x} \geq \underline{0}\} \neq \emptyset,$$

then all vertices of $P(\underline{b})$ are integral.

So now, how do we quickly check if a matrix is TU?

Proof*:

Let $\tilde{\underline{x}}$ be any vertex of $P(\underline{b})$.

First we show that $(\tilde{\underline{x}}, \tilde{\underline{s}})$ with $\tilde{\underline{s}} := A\tilde{\underline{x}} - \underline{b}$ is a vertex of

$$P'(\underline{b}) := \{(\underline{x}, \underline{s}) \in \mathbb{R}^{n+m} \ : \ A\underline{x} - \underline{s} = \underline{b}, (\underline{x}, \underline{s}) \geq \underline{0}\}.$$

If not, there would exist two distinct $(\underline{x}_1, \underline{s}_1)$ and $(\underline{x}_2, \underline{s}_2)$ of $P'(\underline{b})$ such that $(\tilde{\underline{x}}, \tilde{\underline{s}}) = \alpha(\underline{x}_1, \underline{s}_1) + (1-\alpha)(\underline{x}_2, \underline{s}_2)$ for some $\alpha$ with $0 < \alpha < 1$.

Since $\underline{s}_1 = A\underline{x}_1 - \underline{b} \geq \underline{0}$ and $\underline{s}_2 = A\underline{x}_2 - \underline{b} \geq \underline{0}$, $\underline{x}_1$ and $\underline{x}_2$ belong to $P(\underline{b})$.

Moreover, $(\underline{x}_1, \underline{s}_1) \neq (\underline{x}_2, \underline{s}_2)$ would imply $\underline{x}_1 \neq \underline{x}_2$ and hence $\tilde{\underline{x}} = \alpha \underline{x}_1 + (1-\alpha)\underline{x}_2$ could not be a vertex of $P(\underline{b})$.

Since A is TU, also $(A \,|\, -I_m)$ is TU. According to Theorem 1 for $P'(\underline{b})$, $(\tilde{\underline{x}}, \tilde{\underline{s}})$ is integral, in particular $\tilde{\underline{x}}$.

□

only this, so there are TU matrices which violate some of those assumptions

**Proposition** (Sufficient conditions):

$A \in \mathbb{Z}^{m\times n}$ is TU $\boxed{\text{if}}$

i) $\boxed{a_{ij} \in \{-1, 0, +1\}}$ for all $i$ and $j$,

ii) each column of $A$ contains $\boxed{\text{at most two}}$ nonzero coefficients,

iii) set $I$ of all row indices of $A$ can be partitioned into $I_1$ and $I_2$ such that,

for each column $j$ with $\boxed{\text{two}}$ nonzero coefficients, we have $\boxed{\sum_{i\in I_1} a_{ij} - \sum_{i \in I_2} a_{ij} = 0.}$

N.B.: If a column has two nonzero coefficients of the same (different) sign, their rows must belong to different (same) subsets $I_1$ and $I_2$.

basically, entries need to balance, so for example
- same nonzero entries ⟶ different subsets
- different nonzero entries ⟶ same subset (so they cancel each other)

Examples of TU matrices (not) satisfying these conditions:

$$\begin{pmatrix} 1 & 1 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

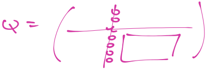
yes, TU, and $\left.\begin{array}{l} I_1 = I \\ I_2 = \emptyset \end{array}\right\}$ extreme case

$$A = I\left\{\begin{pmatrix} & -1 & \\ 1 & 1 & \\ -1 & & 1 \\ \hline & & 1 \end{pmatrix}\right. \begin{array}{l} \}\, I_1 \\ \}\, I_2 \end{array}$$

Proof: Suppose $A$ is not TU (but the three assumptions are met, so we try to run into a contradiction).

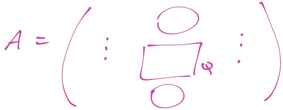
Let $Q$ be the smallest square submatrix of $A$ among the ones which $\det(Q) \notin \{-1,0,1\}$.

Being the smallest it can't contain a col with a single non zero coeff, otherwise $Q$ would not be the smallest.



Thus the cols of $Q$ must contain exactly two non zero coeffs. ?

and so above and below $Q$ we just have zeros.



According to the assumptions on $A$ we have that

$$\sum_{i\in I_1} a_{ij} = \sum_{i\in I_2} a_{ij}$$

and so, since $a_{ij}=0$ $\forall j \in Q$ and $\forall i \notin Q$ we would have that the rows of $Q$ would be linear dependent, and so $\det(Q)=0$ which is contradiction.

## Characterization of TU matrices

they ∃ but are not handy to use

**Theorem 2**: $A \in \mathbb{Z}^{m \times n}$ is TU if and only if every subset $\forall I' \subseteq I = \{1, \ldots, m\}$ of indices of the rows of $A$ can be partitioned into $I'_1$ and $I'_2$ such that $\left(\sum_{i \in I'_1} a_{ij} - \sum_{i \in I'_2} a_{ij}\right) \in \{-1, 0, +1\}$ for every column $j$, with $1 \leq j \leq n$.

relaxed computation but needs to be verified for all possible subsets of rows I

moral.

If $A$ is TU it suffices to solve the LP relaxation.

**Proposition**: $\min\{\underline{c}^t \underline{x} \; : \; A\underline{x} = \underline{b}, \underline{x} \in \mathbb{R}^n_+\}$ has an optimal integer solution for any integer $\underline{b}$ (for which it admits a finite optimal solution) if and only if $A$ is TU.

we ask the $\underline{b}$ so that it was not unbounded

Given $A$ and a basis $B$ with $det(B) \neq \pm 1$, there always exists a LP $\min\{\underline{c}^t \underline{x} \; : \; A\underline{x} = \underline{b}, \underline{x} \in \mathbb{R}^n_+\}$ with a fractional optimal solution.

# 3.3.2 Some ideal natural formulations

## 1) Assignment problem

Given $n$ jobs and $n$ machines with costs $c_{ij}$ for all $i, j \in \{1, \ldots, n\}$, decide which job to assign to which machine so as to minimize the total cost to complete all the jobs.

<u>ILP formulation</u>:

$$\begin{array}{lll} \min & \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} & \\ & \sum_{i=1}^{n} x_{ij} = 1 & \forall j \quad (2) \\ & \sum_{j=1}^{n} x_{ij} = 1 & \forall i \quad (3) \\ & x_{ij} \in \{0, 1\} & \forall i, \forall j \end{array}$$

in these constraints each variable $x_{ij}$ appears only once, with coeff of 1

of course the position of the 1s will change (incrementable) with the variables

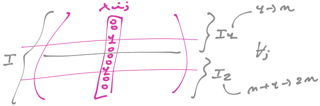
where $x_{ij} = 1$ if job $i$ is assigned to machine $j$, $1 \le i, j \le n$.

the full relaxation should be $0 \le x_{ij} \le 1$, but we can put only $x_{ij} \ge 0$ as the constraints will prevent $x_{ij}$ to be $\ge 1$

N.B.: In LP relaxation, $x_{ij} \ge 0 \ \forall i, j$ suffice

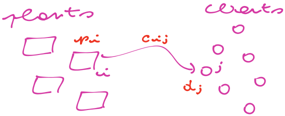
**Property**: Constraints matrix (2)-(3) is TU.

<u>Proof</u>: for each $x_{ij}$ we have two constraints:
- for C(2) we will have a 1 somewhere
- same for C(3)
- the rest entries will be zero




<u>Consequence</u>: All vertices of the LP relaxation are integral, and formulation is ideal.

## 2) Transportation problem




Single type of product.

Given

- $m$ production plants $(1 \leq i \leq m)$
- $n$ clients $(1 \leq j \leq n)$
- $c_{ij}$ = unit transportation cost from plant $i$ to client $j$
- $p_i$ = maximum amount that can be produced (capacity) at plant $i$
- $d_j$ = demand of client $j$
- $q_{ij}$ = maximum amount that can be transported from plant $i$ to client $j$

determine a transportation plan so as to minimize total transportation costs while satisfying all client demands and plant capacities.

Assumption: $\sum_{i=1}^{m} p_i \geq \sum_{j=1}^{n} d_j$

Natural ILP formulation:

Variables: $x_{ij}$ = amount of product transported from plant $i$ to client $j$, with $1 \leq i \leq m$, $1 \leq j \leq n$

$$\min \quad \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

the TU term worked with (=) or (≥) even

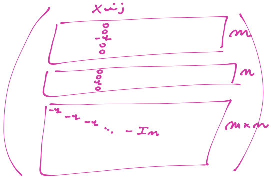
$$\sum_{j=1}^{n} -x_{ij} \overset{\geq}{\cancel{\leq}} p_i \qquad \forall i \rightarrow m \text{ constr} \tag{4}$$

$$\sum_{i=1}^{m} x_{ij} \geq d_j \qquad \forall j \rightarrow n \text{ constr} \tag{5}$$

$$-x_{ij} \overset{\geq}{\cancel{\leq}} q_{ij} \qquad \forall i, \forall j \rightarrow m \times n \text{ constr} \tag{6}$$

$$x_{ij} \geq 0 \ \text{ integer} \qquad \forall i, \forall j$$

**Property**: Constraints matrix (4)-(6) is TU.



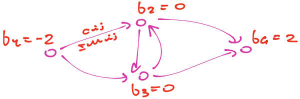

Proof: adding an Id matrix doesn't change the TUness, so we just focus on the upper part, which indeed is TU

it's not $I_m$ but $-I_m$
but it's still true since we also know the property of switch signs

**Consequence**: If all $p_i$, $d_j$ and $q_{ij}$ are integer, every vertex is integral, and hence the formulation is ideal.

## 3) Minimum cost flow problem

Given directed $G = (V, A)$ with a capacity $u_{ij}$ and a unit cost $c_{ij}$ for each $(i,j) \in A$, and a "demand" $b_i$ for each $i \in V$ ($b_i < 0$ for sources, $b_i > 0$ for destinations, $\sum_{i \in V} b_i = 0$), determine a feasible flow of minimum total cost satisfying all $b_i$.




Natutal ILP formulation:

$$
\begin{array}{llr}
\min & \sum_{(i,j) \in A} c_{ij} x_{ij} & \\
 & \sum_{(h,i) \in \delta^-(i)} x_{hi} - \sum_{(i,j) \in \delta^+(i)} x_{ij} = b_i \quad \forall i \in V & (7) \\
 & x_{ij} \leq u_{ij} \quad \forall (i,j) \in A & (8) \\
 & x_{ij} \geq 0 \ \text{ integer} \quad \forall (i,j) \in A &
\end{array}
$$

**Property**: Constraints matrix (7)-(8) is TU.

Proof:

**Consequence**: If all $b_i$ and capacities $u_{ij}$ are integer, every extreme point is integral, and the formulation is ideal.

Exercise:

Verify that the following problems are special cases of Min cost flow problem.

- Shortest path: Given directed $G = (V, A)$ with cost $c_{ij}$ for each $(i, j) \in A$, and two prescribed nodes $s$ and $t$, determine a minimum cost path from $s$ to $t$.

- Maximum flow: Given directed $G = (V, A)$ with a capacity $u_{ij}$ for each $(i, j) \in A$, a $s$ and a sink $t$, determine a feasible flow of maximum value from $s$ to $t$.

## Ad hoc more efficient algorithms

For the three above problems, the formulations are ideal but there exist better polynomial-time algorithms which exploit the problem's structure.

## Rounding optimal solutions of LP relaxation

In general, when constraint matrix of ILP is not TU, $\underline{x}^*_{LP}$ is fractional.

Rounding $\underline{x}^*_{LP}$ does rarely work because

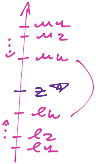
- rounded solutions are often infeasible for ILP,
- the error with respect to w.r.t. an optimal ILP solution may be arbitrarily large.

In general, rounding $\underline{x}^*_{LP}$ yields a good approximation of $\underline{x}^*_{IP}$ only when the components of $\underline{x}^*_{LP}$ have large values.

# 3.4 Relaxations, heuristics and bounds

Generic Discrete Optimization problem

$$z^* = \min\{c(\underline{x}) \; : \; \underline{x} \in X\}$$




we stop when $u_k$ and $l_k$ gets close enough

and an optimal solution $\underline{x}^* \in X$.

like the heuristic or Monte Carlo algos

Algorithms generate: a decreasing sequence of upper bounds $u_1 > \ldots > u_k \geq z^*$ and an increasing sequence of lower bounds $l_1 < \ldots < l_k \leq z^*$.

Termination criterion: $\boxed{(u_k - l_k) \leq \varepsilon}$ for $\varepsilon > 0$.

$\Rightarrow$ at the end of the alg we have also a precise estimate of the quality of the sol (a guarantee)

**Primal bounds** (min)

Any $\overline{x} \in X$ yields an upper bound $\overline{u} = c(\overline{x}) \geq z^*$.

Even finding an $\overline{x} \in X$ may be challenging ($\mathrm{NP}$-hard).

we "which was feasible"

**Dual bounds** (min)

as (for a minimization problem) we know that relaxation provides better (so $l_k \leq z^*$) sols

Lower bounds are obtained via a relaxation.

**Quality guarantee**:

If $\underline{x}_k$ is best feasible solution found so far and $l_k$ best dual bound,

$$(c(\underline{x}_k) - l_k) \leq \varepsilon$$

guarantees $(c(\underline{x}_k) - z^*) \leq \varepsilon$.

For maximization problems, primal (dual) bounds are lower (upper) bounds.

- for primal bounds we can use any heuristic that we can have
- for dual bounds we need to talk about (and generalize) the concept of relaxation

**Definition**: Given

$$(P) \qquad z^* = \min\{c(\underline{x}) \ : \ \underline{x} \in X \subseteq \mathbb{R}^n\},$$

a problem

$$(RP) \qquad \tilde{z} = \min\{\tilde{c}(\underline{x}) \ : \ \underline{x} \in \tilde{X} \subseteq \mathbb{R}^n\}$$

is a **relaxation** of $(P)$ if

- $X \subseteq \tilde{X}$ feasible region is larger
- $\tilde{c}(\underline{x}) \leq c(\underline{x})$ for each $\underline{x} \in X$. but we can also relax the obj function (but on $X$)

**Proposition**: If $(RP)$ is a relaxation of $(P)$ then $\boxed{\tilde{z} \leq z^*}$.

Proof:

Let $x^*$ be an optimal sol of P.
Then
- $x^* \in X \subseteq \tilde{X} \Rightarrow x^* \in \tilde{X}$ $\Rightarrow$ $\tilde{z}^* \leq z^*$
- $\tilde{c}(x^*) \leq c(x^*) = z^*$

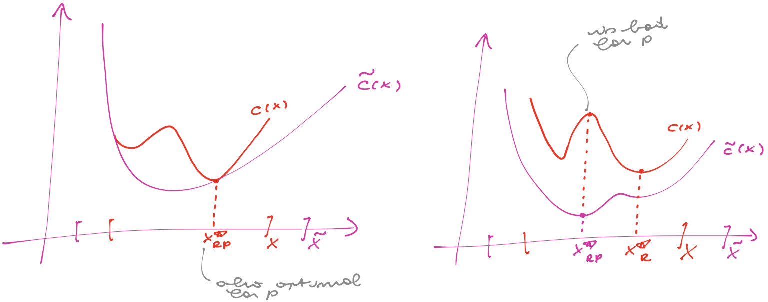
**Proposition**: Let $\underline{x}^*_{RP}$ be an optimal solution of $(RP)$. If $\underline{x}^*_{RP}$ is feasible for $(P)$ $(\underline{x}^*_{RP} \in X)$ $\boxed{\text{and}}$ $\tilde{c}(\underline{x}^*_{RP}) = c(\underline{x}^*_{RP})$, then $\underline{x}^*_{RP}$ is also optimal for $(P)$.

x*RP was feasible for p

the obj values coincide

now we see why we need this additional assumption

Illustrations:




Aim at tradeoff between the bound quality $(z^* - \tilde{z})$ and the computational load of $(RP)$.

# 3.4.1 Different types of relaxations

**1) Linear programming relaxation**

For any (M)ILP

$$\begin{aligned} z_{ILP} = \min \quad & \underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y} \\ & A_1 \underline{x} + A_2 \underline{y} \geq \underline{b} \\ & \underline{x} \geq \underline{0}, \underline{y} \geq \underline{0}, \text{ integer} \end{aligned}$$

and its LP relaxation

$$\begin{aligned} z_{LP} = \min \quad & \underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y} \\ & A_1 \underline{x} + A_2 \underline{y} \geq \underline{b} \\ & \underline{x} \geq \underline{0}, \underline{y} \geq \underline{0}, \end{aligned}$$

we have $z_{LP} \leq z_{ILP}$. The *stronger* the formulation, the *tighter the dual bound* $z_{LP}$.

## 2) Relaxation by elimination

Simply delete one or more constraints.

Examples:

1) Asymmetric TSP

Delete the subtour elimination (cut-set) constraints.

2) *Multi-dimensional binary knapsack problem*

$$
\begin{array}{llll}
\max & \sum_{j=1}^{n} p_j x_j & & \\
\text{s.t.} & \sum_{j=1}^{n} w_{ij} x_j \leq W_i & \forall i \in \{1, 2, \ldots, m\} & (1) \\
& x_j \in \{0, 1\} & \forall j \in \{1, 2, \ldots, n\} & (2)
\end{array}
$$

Delete all but one constraint.

Very weak relaxations.

## 3) Surrogate relaxation (SR)

idea: aggregate constraints rather than deleting them
not in an equivalent way
(in fact it's a relaxation)

Idea: Replace a subset of constraints with the surrogate constraint, i.e., their linear combination with multipliers $\lambda_i \geq 0$.

Example: *Multiple binary knapsack*

Given $m$ knapsacks of capacities $W_i$, select $m$ disjoint subsets of $n$ items fitting in the knapsacks so as to maximize total profit.

item $j$ goes to knapsack $i$

$$
\begin{aligned}
z_{mKP} = \max \quad & \textstyle\sum_{i=1}^{m} \sum_{j=1}^{n} p_j x_{ij} & & \\
\text{s.t.} \quad & \textstyle\sum_{j=1}^{n} w_j x_{ij} \leq W_i & \forall i \in \{1,2,\ldots,m\} & \qquad (3)\\
& \textstyle\sum_{i=1}^{m} x_{ij} \leq 1 & \forall j \in \{1,2,\ldots,n\} & \qquad (4)\\
& x_{ij} \in \{0,1\} & \forall i, \forall j & \qquad (5)
\end{aligned}
$$

Surrogate relaxation of (3):

surrogate relaxation

$$
\begin{aligned}
z_S(\lambda) = \max \quad & \sum_i \sum_j p_j x_{ij} & \\
\text{st} \quad & \sum_i \lambda_i \Big( \sum_j w_j x_{ij} \Big) \leq \sum_i \lambda_i (W_i) & (6)\\
& \sum_i x_{ij} \leq 1 \quad \forall j & (7)\\
& x_{ij} \in \{0,1\} \quad \forall i \, \forall j & (8)
\end{aligned}
$$

we get a whole constraint

we have more$^{(m)}$ copies of each item, with different weights according to the $\lambda_i$'s

$$
\begin{array}{rlr}
z_{S(\underline{\lambda})} = \max & \sum_{i=1}^m \sum_{j=1}^n p_j x_{ij} & \\
\text{s.t.} & \sum_{i=1}^m \sum_{j=1}^n (\lambda_i w_j) x_{ij} \leq \sum_{i=1}^m \lambda_i W_i & (9) \\
& \sum_{i=1}^m x_{ij} \leq 1 \qquad \forall j \in \{1, 2, \dots, n\} & (10) \\
& x_{ij} \in \{0, 1\} \qquad \forall i, \forall j & (11)
\end{array}
$$

Since for each item $j$ a copy $i$ with smallest $\lambda_i$ is more convenient, it is a standard binary knapsack problem with capacity $\sum_{i=1}^m \lambda_i W_i$.

relaxation provides an upper bound since the problem was a maximization

Clearly $z_{mKP} \leq z_{S(\underline{\lambda})}$.

Look for smallest upper bound by solving *surrogate dual*:

$$\boxed{\min_{\underline{\lambda} \geq \underline{0}} z_{S(\underline{\lambda})}}$$

solving this provides the best possible bound

unfortunately surrogate duals are often complex to solve

## 4) Lagrangian relaxation (LR)

Often LP relaxation and relaxation by elimination yield weak bounds (e.g., TSP, UFL).

Idea: Eliminate the "difficult" constraints and add, for each one of them, a term in the objective function with a multiplier $u$ which penalizes its violation.

For max: terms $\geq 0$ for all feasible solutions.

Example: *Multiple binary knapsack*

this constr seems difficult as it relates all the knapsacks

$$
\begin{aligned}
z_{mKP} = \max \quad & \textstyle\sum_{i=1}^{m} \sum_{j=1}^{n} p_j x_{ij} & & \\
\text{s.t.} \quad & \textstyle\sum_{j=1}^{n} w_j x_{ij} \leq W_i & & \forall i \in \{1, 2, \ldots, m\} \\
& \textstyle\sum_{i=1}^{m} x_{ij} \leq 1 & & \forall j \in \{1, 2, \ldots, n\} \\
& x_{ij} \in \{0, 1\} & & \forall i, \forall j
\end{aligned}
\qquad (12)
$$

knapsacks

items

Lagrangian relaxation of (12):

$$z_{LR(u)} = \max \sum_i \sum_j p_j x_{ij} + \sum_{j=1}^{m} \underbrace{u_j}_{\geq 0} \underbrace{\Big(1 - \sum_{i=1}^{m} x_{ij}\Big)}_{<0} \qquad (13)$$

< 0 if constr is not satisfied

$$\text{st} \quad \sum_j w_j x_{ij} \leq W_i \quad \forall i \qquad (14)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \, \forall j \qquad (15)$$

we keep the other constraints

=> we will subtract something, makes sense, we are penalizing since it is a maximization problem

Since

$$\sum_{i=1}^{m}\sum_{j=1}^{n} p_j x_{ij} + \sum_{j=1}^{n} u_j\left(1 - \sum_{i=1}^{m} x_{ij}\right) = \sum_{i=1}^{m}\sum_{j=1}^{n}(p_j - u_j)x_{ij} + \sum_{j=1}^{n} u_j,$$

in Lagrangian subproblem (13)-(15) each item $j$ has profit $\tilde{p}_j = p_j - u_j$, weight $w_j$ and can be inserted in several knapsacks.

if this happens we just done a rendition

this connection is simpler since we reduced (more equivalent) this problem to m 1 knapsack problems

$$z_{L(u)} = \sum_{i=1}^{m} z_i + \sum_{j=1}^{n} u_j \quad \text{where} \quad z_i = \max \sum_{j=1}^{n} \tilde{p}_j x_j \tag{16}$$

$$\text{st} \quad \sum_{j=1}^{n} w_j x_j \le W_i \tag{17}$$

$$x_j \in \{0,1\}$$

*Lagrangian dual*:

$$\min_{\underline{u}\geq\underline{0}} z_{L(\underline{u})}.$$

LR discussed in detail later.

# Simple dominance relations among relaxations

Compare the quality of three relaxations in terms of dual bound (relaxing same constraints with optimal multipliers).

surrogate lagrangian

**Proposition**: SR and LR dominate the relaxation by elimination.

The latter (R. by elimination) is equivalent to taking
- $\lambda = 0$ in SR
- $u = 0$ in LR

**Proposition**: SR dominates LR.

SR ⇒ we combine constr
LR ⇒ we (penalize and) delete constr

LR can be viewed as the lagrangian relaxation of the SR obtained by relaxing the surrogate/aggregated constr with $u = 1$

So we can take the SR and lagrangiate it (model)

So we should prefer SR, but

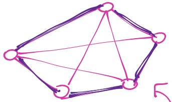
In practice LR is widely used because

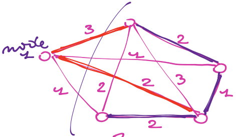
- Lagrangian subproblem is easier to solve than surrogate one,
- ∃ efficient methods to determine "good" Lagrangian multipliers, unlike for SR.

## 5) Combinatorial relaxations: Symmetric TSP

**Definition**: Given undirected $G = (V, E)$ with $V = \{1, \ldots, n\}$, a **1-tree** is a subgraph containing two edges incident to node 1, and the edges of a spanning tree on $\{2, \ldots, n\}$.

Illustration:






Clearly { Hamiltonian cycles of $G$ } $\subset$ { 1-trees of $G$ } — a wider set, as the choice of the two edges incident one on each cycle

Exact algorithm for minimum cost 1-tree:

- we determine the min cost ST on the subgraph of the tree $\{2, \ldots, n\}$ via the Kruskal alg (optimal alg)
- we select two edges incident with the special node 1 with the smallest cost

Recall Kruskal's greedy algorithm:

Consider edges in the order of non-decreasing cost.

At each step, discard edge if it creates a cycle with previously selected edges.

Stop when selected edges "cover" all the nodes. ~ we stop when we have
- $n$ vertices, and
- $n-1$ selected edges

# 3.4.2 Heuristics for primal bounds

## 1) **Greedy methods**

Construct a feasible solution piece by piece.

At each step, select an available "piece" that yields the best "local profit", without reconsidering previous choices.

Example 1: Binary Knapsack Problem

$$
\begin{aligned}
z_{ILP} = \max \quad & 16x_1 + 22x_2 + 12x_3 + 8x_4 \\
\text{s.t.} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\
& x_1, \ldots, x_4 \in \{0, 1\}
\end{aligned}
$$

Order items by non-increasing profit-weight ratios $(p_j / w_j)$ :

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|---|---|---|---|---|
| $p_j$ | 16 | 22 | 12 | 8 |
| $w_j$ | 5 | 7 | 4 | 3 |
| $p_j/w_j$ | 3.2 | 3.14 | 3 | 2.7 |

residual capacity → 14 ⇒ 9 ⇒ 2 ⇒ 2 capacity available in the end

Consider items in that order, select $(x_j = 1)$ those not violating the residual capacity, skip the others $(x_j = 0)$.

$x = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$ $z = 38$ (is it opt? no $x^* = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ $z^* = 42$)

Feasible solution of greedy procedure: $\underline{\overline{x}} = (1, 1, 0, 0)$ with $\overline{z}_{greedy} = 38$.

Optimal integer solution: $\underline{x}^* = (0, 1, 1, 1)$ with $z_{ILP} = 42$.

Clearly $\overline{z}_{greedy} \leq z_{ILP}$.

How bad can a greedy solution be w.r.t. an optimal one?

Worst case example:

item 1 : $w_1 = 1$, $p_1 = 2$ $\Rightarrow$ ratio$_1 = 2$ best

item 2 : $w_2 = W$, $p_2 = W$ ratio$_2 = 1$ worst

$\Rightarrow$ $\underline{x}_{greedy} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $z_{greedy} = 2$

$\underline{x}^* = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ $z^* = W$

$\Rightarrow$ we may be arbitrarily far away from the optimal sol

Example 2: Symmetric TSP with complete graph

Nearest neighbor heuristic: Start from any node, at each step insert the closest node not yet visited, come back to the starting node.

Complexity: $O(n^2)$, where $n = |V|$.

For animation see https://www.youtube.com/watch?v=fFfizorMPuk

Empirical performance: on TSPLIB(rary) instances it yields tours whose average cost is about 1.26 times that of optimal tours. is not too bad (on average)

Worst-case performance: there are instances for which the found tours are arbitrarily worse than the optimal ones.

## 2) **Local search methods**

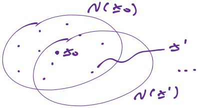
Generic

$$\min_{\underline{x} \in X} c(\underline{x})$$

and try to iteratively improve a current feasible solution.

Define, for any feasible solution $\underline{x}$, a neighborhood $N(\underline{x})$, i.e., a subset of "nearby" feasible solutions.

Start from an initial $\underline{x}_0$. as obtained from a greedy alg




we move around the $N(\underline{x}_k)$ evaluating for improved set

if we don't find any then we return $\underline{x}_k$ as it was the local optimum (wrt $N(\underline{x})$)

At iteration $k$:

- find a best solution $\underline{x}'$ in $N(\underline{x}_k)$
- if $c(\underline{x}') < c(\underline{x}_k)$ then $\underline{x}_{k+1} := \underline{x}'$ and perform iteration $k+1$,

  otherwise return $\underline{x}_k$ which is a local minimum w.r.t. $N(\underline{x})$.

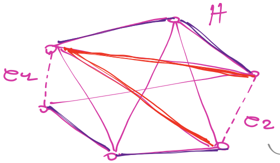
as we are replacing two edges at each time / iteration

Example: 2-opt heuristic for Symmetric TSP

Given $G = (V, E)$ and a current tour $H \subseteq E$.

For any nonadjacent $e_1$ and $e_2$ in $H$, try to replace them with the two (unique) alternative edges recombining the two paths into a new tour $H'$.

Illustration:




- start from H
- eliminate two non consecutive edges
- select two other edges that will produce another ham cycle H'

anyway edges can overlap in an ham cycle, as we think them as a jump

$N(H) = \{$ tours obtainable from $H$ with such a "2-interchange" $\}$.

If $c(H') < c(H)$ then set $H = H'$, otherwise $H$ is a local minimum w.r.t 2-opt neighborhood.

For animation see: https://www.youtube.com/watch?v=UGGPZnAUjPU
http://www.youtube.com/watch?v=SC5CX8drAtU

Complexity: $O(n^2)$ with $n = |V|$.

Also $k$-opt for $k = 3$, with complexity $O(n^3)$.

Empirical performance: on TSPLIB instances 2-opt (3-opt) provides tours about 1.06 (1.04) times the optimum.

(3)
# **Metaheuristics** (for minimization problems)

To try to escape from local optima and improve upon local search heuristics.
E.g., tabu search, simulated annealing or genetic algorithms.


## Tabu Search:

Idea: Allow moves to the best neighbor even if it has a worse objective function value.
Use a tabu list to avoid cycling.

to avoid reconsidering previous feasible sets that we already considered
- we forbid to take choices which would make us come back to previous steps
- rather than storing all the past steps values

Start from feasible $\underline{x}_0$.

At iteration $k$, $\underline{x}_{k+1} := \underline{x}'$ where $\underline{x}'$ is the best solution in $N(\underline{x}_k)$, even if $c(\underline{x}') \geq c(\underline{x}_k)$.

Prevent to undo recent moves for a certain number of iterations.
Once a move is peformed the opposite move is made tabu for the $l$ successive iterations.

Best solution found is stored and returned after a prescribed maximum number of iterations.

Example: Uncapacitated Facility Location (UFL) problem

$m$ clients ($i \in M$) and $n$ depots ($j \in N$)

we have to decide
- which depot where to open
- from which depot to serve the clients

For any $S \subseteq N$, consider the feasible solution where the depots with indices in $S$ are open and all clients are served by the "cheapest" open depot.

we can do this as we don't have any capacity constraint

Corresponding objective function value:

what is a simple neighborhood construction?

from a set $S$ we could
- remove a depot $\Rightarrow S \to S \setminus \{i\},\ i \in S$
- add a depot $\Rightarrow S \to S \cup \{i\},\ i \notin S$

$$\Rightarrow \quad N(S) = \left\{ T \subseteq N : \begin{array}{l} T = S \cup \{i\},\ i \notin S \text{ or} \\ T = S \setminus \{i\},\ i \in S \end{array} \right\}$$

$m = 6$ clients, $n = 4$ depots

$S_0$

$$(c_{ij}) = \begin{pmatrix} 6 & \underline{2} & 3 & 4 \\ \underline{1} & 9 & 4 & 11 \\ 15 & \underline{2} & 6 & 3 \\ \underline{9} & 11 & 4 & 8 \\ \underline{7} & 23 & 2 & 9 \\ 4 & \underline{3} & 1 & 5 \end{pmatrix}$$

service costs

fixed costs

$\underline{f} = (21, 16, 11, 24)^t$

Initial solution: $S_0 = \{1, 2\}$ of cost $c(S_0) = 61$.

Three iterations of Local search (Tabu Search):...

try $T = 1$, $T = 2$, $T = 123$, $T = 429$

# 3.5 Branch and Bound – Review

Generic Discrete Optimization problem:

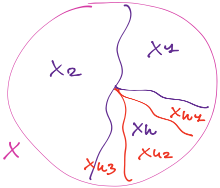
$$(P) \qquad z = \max\{c(\underline{x}) \, : \, \underline{x} \in X\}.$$

Branch and Bound is a general semi-enumerative approach (Land and Doig 1960) to explore the feasible region $X$.

See chapter 7 of L. Wolsey, Integer Programming, Wiley 1998, p. 91-111.

Two main components:

- "divide and conquer" strategy (branching)
- implicit enumeration exploiting bounds (bounding).




By exploiting bounds

- it avoids explicitly exploring certain subregions of $X$
- it is guaranteed to find an optimal solution.

- partition X into subregions
- solve the problem (or their relaxation) on the regions $X_h$
- if it is not integer we either branch that region while integers it provide the bounds

## 1) "Divide and conquer" strategy

Idea: Recursively partition $X$ so as to reduce the solution of $(P)$ to the solution of a sequence of smaller/easier subproblems.

**Observation**: Let $X = X_1 \cup \ldots \cup X_k$ be a *partition* of $X$ in $k$ subsets ($X_i \cap X_j = \emptyset$ for each pair of indices $i \neq j$) and

$$z^i = \max\{c(\underline{x}) \,:\, \underline{x} \in X_i\}$$

for $1 \leq i \leq k$. Obviously $z = \max_{1 \leq i \leq k} z^i$.

Partition of X or $X_i \equiv$ branching operation.

Procedure represented by a **enumeration tree** with root node associated to $X$ and other nodes to the subsets $X_i$.

Examples:

- $X \subseteq \{0,1\}^3$ – binary branching

- $X$ set of all Hamiltonian circuits of a given digraph $G = (V, A)$ – multiway branching

## 2) Implicit enumeration

Explicit enumeration is too heavy computationally, recursive partition of the feasible region does not suffice.

Idea: Exploit **upper** and **lower bounds** (primal and dual bounds) on $z^i$, with $1 \leq i \leq k$, in order to avoid explicit exploration of some subregions $X$.

**Observation**: Let $X = X_1 \cup \ldots \cup X_k$ be a partition of $X$ and

$$z^i = \max\{c(\underline{x}) \,:\, \underline{x} \in X_i\}$$

for $1 \leq i \leq k$.

Moreover, let $l^i$ be a lower bound and $u^i$ an upper bound on $z^i$, namely $l^i \leq z^i \leq u^i$.

Then $l = \max_{1 \leq i \leq k} l^i$ is a lower bound and $u = \max_{1 \leq i \leq k} u^i$ is an upper bound on $z$, that is $l \leq z \leq u$.

## Pruning criteria

Cases in which primal and dual bounds for $i$-th subproblem can be used to avoid exploring (discard) $X_i$ (to prune the corresponding node of the B&B tree):

- **Optimality criterion**: If $u_i = l_i$, no need to further explore $X_i$ since we found an optimal solution in $X_i$ of value $z^i = u_i = l_i$.
- **Bounding criterion**: If the upper bound $u_i$ is lower than
  - the objective function value $LB$ of the best solution $\underline{x}_{LB}$ found so far

  or

  - any lower bound $l_j$ for $j \neq i$,

  no need to explore $X_i$ because it cannot contain any better feasible solution.
- **Feasibility criterion**: $X_i = \emptyset$

Four examples of subproblems (node) configurations, including one whose feasible region must be further explored.

If a subproblem is not "solved", recursively generate subproblems (branching step).

**Main ingredients of Branch and Bound method** (max problems)

- *Upper bounds*: Efficient method to determine a good quality dual bound $u$ on $z$.

- *Lower bounds*: Efficient heuristic to look for a feasible solution $\underline{\tilde{x}}$ with a value $c(\underline{\tilde{x}})$, which provides a good lower bound $c(\underline{\tilde{x}})$ on $z$.

- *Branching rule*: Procedure to (recursively) partition the feasible region $X$ into smaller subregions.

<u>To be stored and updated</u>:

- list $\mathcal{L}$ of active subproblems with lower and upper bounds on $z^i$: $l^i \leq z^i \leq u^i$,

- global upper bound $UB$ on $z$,

- global lower bound $LB$ on $z$ provided by the best feasible solution $\underline{x}_{LB}$ found so far.

**General method**, we "just" need to specify:

1. how to choose the next subproblem (active node) to be "processed"
2. how to generate the subproblems of a given subproblem (the "children" nodes)
3. how to efficiently compute the primal and dual bounds.

The performance of a Branch-and-Bound algorithm strongly depends on the efficiency of the branching rule and the quality of primal and dual bounds.

A Branch-and-Bound approach is applicable to MILPs and to Nonlinear Optimization problems.

# 3.5.1 Branch and Bound for ILP problems

Find an optimal solution $\underline{x}^*_{ILP}$ of

$$z_{ILP} = \max\{\underline{c}^t\underline{x} \,:\, A\underline{x} = \underline{b}, \underline{x} \geq \underline{0} \text{ integer}\}. \qquad (1)$$

Solve its **linear relaxation** and let $\underline{x}^*_{LP}$ be an optimal solution of value $z_{LP}$.
Obviously $z_{ILP} = \underline{c}^t\underline{x}^*_{ILP} \leq z_{LP} = \underline{c}^t\underline{x}^*_{LP}$.

If $\underline{x}^*_{LP}$ is integral, it is also optimal for (1). Otherwise $\underline{x}^*_{LP}$ is fractional.

**Branching**

If $\underline{x}^*_{LP}$ is not integral, choose a fractional component $x^*_h$ and generate the two suproblems:

$$z^1_{ILP} = \max\{\underline{c}^t\underline{x} \,:\, A\underline{x} = \underline{b}, x_h \leq \lfloor x^*_h \rfloor, \underline{x} \geq \underline{0} \text{ integer}\}$$

$$z^2_{ILP} = \max\{\underline{c}^t\underline{x} : A\underline{x} = \underline{b}, x_h \geq \lfloor x^*_h \rfloor + 1, \underline{x} \geq \underline{0} \text{ integer}\}$$

with the corresponding subregions $X_1$ and $X_2$ of $X$, which are exhaustive and mutually exclusive.

Clearly $z_{ILP} = \max\{z^1_{ILP}, z^2_{ILP}\}$.

Recursive process: solve the linear relaxation of each subproblem and, if needed, carry out a branching step.

**Bounding**

Consider the $i$-th subproblem with feasible subregion $X_i$.

Solve its **linear relaxation**, let $\underline{x}^*_{LP}$ be an optimal solution and $z^i_{LP}$ its value.

Clearly, if all $c_i$s are integer, every feasible solution of the ILP in $X_i$ has value $\leq \lfloor z^i_{LP} \rfloor$.

In Branch and Bound, branching and bounding operations are alternated, while storing and updating the best feasible solution found.

We need to decide:

1. criterion to select the next subproblem (node) to explore,
2. how to generate the "children" nodes for the node under consideration (choice of the branching variable),
3. heuristic to determine the lower bounds on the optimal objective function value.

1. Choice of the subproblem (node) to be processed

- *Depth first search strategy* ("deepest" node first): easy to implement but costly if wrong choice.
- *Best bound first strategy* (most "promising" node first): tend to generate less nodes but the subproblems are less constrained (we rarely update the best solution found so far).

2. Choice of the fractional variable for branching

- Branching first on a fractional variable whose fractional part is closest to 0.5 (in an attempt to generate two subproblems that are "equally" constrained) is often not the best choice.
- *Strong branching* ("estimate" the bound improvement if branching on several candidate fractional variables, and branch w.r.t. the best one) is costly but effective for some hard instances.

**Exponential example for Branch and Bound**:

Let $n$ be an odd positive integer and consider the ILP problem:

$$\begin{array}{rcl} \max & -x_n & \\ \text{s.t.} & x_0 + 2\sum_{j=1}^{n} x_j = n & \\ & 0 \leq x_j \leq 1 & \forall j \in \{0, 1, 2, \ldots, n\} \\ & x_j \in \mathbb{Z}^+ & \forall j \in \{0, 1, 2, \ldots, n\}. \end{array}$$

It can be verified that, when Branch and Bound is applied to this ILP instance, at least $2^{\frac{n-1}{2}}$ ILP subproblems are inserted in the list $\mathcal{L}$.

Example 1:

Find an optimal solution of the ILP

$$\begin{array}{rc} \max & 4x_1 - x_2 \\ \text{s.t.} & 4x_1 + 2x_2 \geq 19 \\ & 10x_1 - 4x_2 \leq 25 \\ & x_2 \leq \frac{9}{2} \\ & x_1, x_2 \in \mathbb{Z}^+ \end{array}$$

with the Branch and Bound method by solving graphically the linear relaxation of the subproblems. Branch first with respect to $x_1$.

Example 2:

Solve the binary knapsack problem

$$\begin{array}{rl} \max & 10x_1 + 12x_2 + 5x_3 + 7x_4 + 9x_5 \\ \text{s.t.} & 5x_1 + 8x_2 + 6x_3 + 2x_4 + 7x_5 \leq 14 \\ & x_1, \ldots, x_5 \in \{0, 1\} \end{array}$$

with the Branch and Bound method. Use a simple greedy heuristic to determine the optimal solutions of the linear relaxations.

# 3.6 Cutting plane methods

Generic ILP

$$\min\{\, \underline{c}^t \underline{x} \;:\; \underline{x} \in X = \{\underline{x} \in \mathbb{Z}^n_+ \;:\; A\underline{x} \leq \underline{b}\}\,\}$$

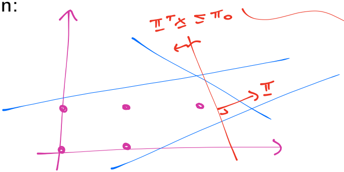
with rational $A$ and $\underline{b}$.

An ideal formulation always exists (Meyer's theorem). But for $NP$-hard problems, it is unknown and/or it contains a huge number of constraints.

Idea: Improve initial formulation (approximation of $conv(X)$) by adding valid inequalities.

**Definition**: $\boxed{\underline{\pi}^t \underline{x} \leq \pi_0}$ is a **valid inequality** for $X \subseteq \mathbb{R}^n$ if $\boxed{\underline{\pi}^t \underline{x} \leq \pi_0 \text{ for each } \underline{x} \in X.}$

Illustration:




this is a valid inequality as it is satisfied by all the points of X

⇒ idea: keep adding these new constraints till the relaxed set is integer

Use of valid inequalities:

- add them a priori
- generate them as needed – via a cutting plane method.

## 1) Addition a priori

**Advantage**: Branch and Bound method with stronger formulation is more efficient (tighter dual bounds).

Example: Given weak UFL formulation with $\sum_{i \in M} x_{ij} \leq m y_j \; \forall j \in N$, add stronger $x_{ij} \leq y_j, \; \forall i \in M, j \in N$.

**Disadvantage**: If huge number of valid inequalities, the LP relaxation is extremely heavy and/or standard Branch and Bound is impossible.

## 2) Cutting plane methods

Generic ILP:

$$\min\{\underline{c}^t\underline{x} \ : \ \underline{x} \in X = P \cap \mathbb{Z}^n\}$$

where $P = \{\underline{x} \in \mathbb{R}^n_+ \ : \ A\underline{x} \leq \underline{b}\}$ is the feasible region of LP relaxation.

A family $\mathcal{F}$ of inequalities $\underline{\pi}^t\underline{x} \leq \pi_0$ valid for $X$, $(\underline{\pi}, \pi_0) \in \mathcal{F}$.

Often $|\mathcal{F}|$ is very large (e.g. cut-set for ATSP).

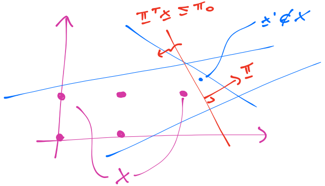
**Definition**: Given $\underline{x}' \in P$ with $\underline{x}' \notin X$, a **cutting plane** is an $\underline{\pi}^t\underline{x} \leq \pi_0$ s.t.

a (candidate) valid inequality

- $\underline{\pi}^t\underline{x} \leq \pi_0$ is valid for $X = P \cap \mathbb{Z}^n$
- $\underline{\pi}^t\underline{x}' > \pi_0$ not valid outside of $X$

Illustration:

once we take $\underline{x}' \in P$ (the relaxation solution) but $\underline{x}' \notin X$

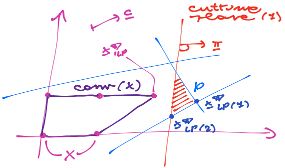


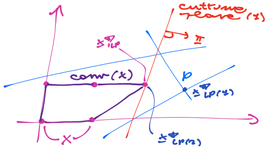
Idea of cutting plane methods:

No need for $conv(X)$, iteratively add cutting planes providing a good description around $\underline{x}^*_{ILP}$, i.e., bringing it out as optimal vertex of LP relaxation polyhedron.

Illustration:

$\min \underline{c}^T \underline{x}$
$\underline{x} \in X$
s.t. ...




this other cutting makes immediately appear an integer vertex as the optimal LP vertex

**Separation problem**:

Given any $\underline{x}' \notin X$ and a family $\mathcal{F}$ of valid inequalities for $X$, find one which separates $\underline{x}'$ from $conv(X)$ or establish that no such cutting plane exists.

Illustration:

we need to solve this at each iteration

nel senso che probabilmente un cutting plane esiste sempre, solo che magari esiste un una famiglia diversa da quella $\mathcal{F}$ che stavamo analizzando

Example: Gomory fractional cutting planes for ILPs – see Foundations of O.R. and 3.6.3.

## Cutting plane method

Initialization $P' := P = \{\underline{x} \in \mathbb{R}^n_+ \; : \; A\underline{x} \leq \underline{b}\}$ the initial LP relaxation polyhedron

1. Solve current LP relaxation $\min\{\underline{c}^t\underline{x} \; : \; \underline{x} \in P'\}$ and let $\boxed{\underline{x}^*_{LP}}$ be an optimal solution.

2. $\boxed{\text{IF } \underline{x}^*_{LP} \in \mathbb{Z}^n}$ THEN terminate because $\underline{x}^*_{LP}$ is also optimal for ILP

   $\boxed{\text{ELSE}}$ Solve the separation problem for $\underline{x}^*_{LP}$, $\mathcal{F}$ and $X = P' \cap \mathbb{Z}^n$

   IF $\underline{\pi}^t\underline{x} \leq \pi_0$ is found THEN $P' := P' \cap \{\underline{x} \in \mathbb{R}^n \; : \; \underline{\pi}^t\underline{x} \leq \pi_0\}$ and go back to (1).

   update P' with the new constraint

   ELSE stop

   we stop cause if was no more able to improve the problem cuts generation

Observation: If $\underline{x}^*_{LP}$ is not integer, $P'$ is anyway stronger than $P$.

since $P' \subset P$, we remove some region of P

## 3.6.1 Simple valid inequalities

1) **Binary set**

$$X = \{x \in \{0,1\}^5 : 3x_1 \underline{- 4x_2} + 2x_3 \underline{- 3x_4} + x_5 \leq \underline{-2}\}$$

immediately we notice that $x_1 = 1$ and $x_2 = 0$ is unfeasible, so a VI is

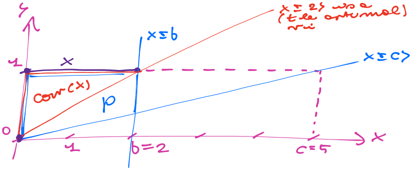
$x_1 \leq x_2$

$x_2$ and $x_4$ are the only variables with negative values, and the RHS is negative ⇒ at least one of them surely will be 1 and therefore a valid inequality is $x_2 + x_4 \geq 1$

2) **Mixed 0-1 set**

$$X = \{(x, y) : x \leq cy, 0 \leq x \leq b,\ y \in \{0,1\}\} \text{ with } c > b$$

Illustration: $c = 5$ and $b = 2$

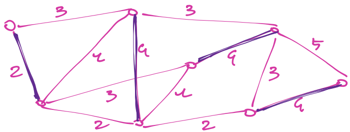



$x \leq by$ is valid and, with $x \geq 0$ and $y \leq 1$, describe $conv(X)$.

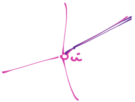
3) **Combinatorial set**

Maximum Matching problem: Given undirected $G = (V, E)$ with profit $p_e \in \mathbb{R}$ for each $e = \{i, j\} \in E$, determine a **matching**, i.e., a subset of edges without common nodes, of maximum total profit.

Illustration:




actually it does not need to be a perfect matching

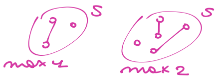



we can select at most one of the incident edges to i

$X = \{\underline{x} \in \{0,1\}^{|E|} : \sum_{e\in\delta(i)} x_e \le 1,\ i \in V\}$ all incidence vectors of matchings in $G$

but this is a very weak formulation, so we can try to add some v.i.

For any $S \subseteq V$ with $|S|$ odd and $|S| \ge 3$,




$$\sum_{e \in E(S)} x_e \le \frac{|S|-1}{2}$$

$E(S) = \{(i, j) : i \in S, j \in S\}$

is valid for $X$.

## 3.6.2 Chvátal cutting planes for ILP

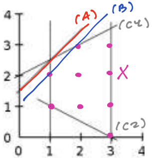
Generate valid inequalities via linear combination and rounding.

=> ex $x \le 2.3$ rational not integer

**Integer rounding principle**: Given $X = \{x \in \mathbb{Z} \ : \ x \le b\}$ where $b \in \mathbb{Q} \setminus \mathbb{Z}$, then $\boxed{x \le \lfloor b \rfloor}$ is valid for $X$.

Example 1:

$$X = \{(x_1, x_2)^t \in \mathbb{Z}^2_+ \ : \ -x_1 + \overset{(c1)}{2x_2} \le 4, \quad -x_1 - \overset{(c2)}{2x_2} \le -3, \quad 1 \le x_1 \le 3\}$$




By adding $-x_1 \le -1$ and $-x_1 + 2x_2 \le 4$ multiplied by $1/2$, we have: $-x_1 + x_2 \overset{(A)}{\le} 3/2$.

which was valid since we obtained it from a linear combination

Then

$$-x_1 + x_2 \le \lfloor 3/2 \rfloor = \overset{(B)}{1}$$

is valid for $X$ and needed to describe $conv(X)$.

we can round this down since the $x_i$ are integers

rounding down implies moving the constr line down till we pass through an integer point

## Chvátal-Gomory (CG) procedure:

Consider $X = P \cap \mathbb{Z}^n$ with $P = \{\underline{x} \in \mathbb{R}^n_+ \;:\; A\underline{x} \leq \underline{b}\}$

$X = \{\underline{x} \in \mathbb{Z}^n_+ \;:\; \sum_{j=1}^n A_j x_j \leq \underline{b}\}$ where $A_j$ is $j$-th column of $A$

the multiplier vector (like 1/2, 1/2 before)

1) Choose $\boxed{\underline{u} \in R^m_+}$ and consider $\boxed{\sum_{j=1}^n (\underline{u}^t A_j) x_j \leq \underline{u}^t \underline{b}}$ the choice of u is an important question

tante componenti quanti constraints

2) Since $\lfloor \underline{u}^t A_j \rfloor \leq \underline{u}^t A_j$ and $x_j \geq 0$,

$$\sum_{j=1}^n \overbrace{\lfloor \underline{u}^t A_j \rfloor}^{\text{int}} \overset{\text{int}}{x_j} \leq \underline{u}^t \underline{b}$$

⇒ this LHS is integer

is valid for $P$ and for $conv(X)$ and $X$.

3) Since $x_j \in \mathbb{Z}^n_+$, the stronger

$$\boxed{\sum_{j=1}^n \lfloor \underline{u}^t A_j \rfloor x_j \leq \lfloor \underline{u}^t \underline{b} \rfloor}$$

is valid for $conv(X)$ and $X$ (but not necessarily for $P$).

so we are (maybe) removing some fractional points from P

## Example 2: **Matching polytope**

Given an undirected $G = (V, E)$ and $X = \{\underline{x} \in \{0,1\}^{|E|} : \sum_{e \in \delta(i)} x_e \leq 1,\ i \in V\}$.

**Proposition 1**: For any $S \subseteq V$ with $|S|$ odd and $|S| \geq 3$,

$$\sum_{e \in E(S)} x_e \leq \frac{|S|-1}{2}$$

so this constraint that we derived is "valid" can be derived with the previous procedure

is a Chvátal-Gomory inequality w.r.t. the linear description

$$\sum_{e \in \delta(i)} x_e \leq 1 \quad \forall i \in V \qquad (1)$$

$|V|$ constraints $\Rightarrow$ we associate the multipliers $u_i \geq 0$ $\forall i = 1, \ldots, |V|$

$$x_e \geq 0 \quad \forall e \in E. \qquad (2)$$

Proof:

Consider any $S \subseteq V$ with $|S| \geq 3$.

Linear combination of (1) with $u_i = 0.5$ for $i \in S$ and $u_i = 0$ for $i \notin S$, yields

$$2 \cdot \sum_{e \in E(S)} \frac{1}{2} x_e + \frac{1}{2} \sum_{e \in \delta(S)} x_e \leq \frac{|S|}{2}$$

which is valid for $X$.

Since $x_e \geq 0$ and $x_e \in \mathbb{Z}$ for each $e \in E$, also

$$\sum_{e \in E(S)} x_e \leq \lfloor \frac{|S|}{2} \rfloor \tag{3}$$

is valid for $X$.

If $|S|$ is even, (3) is implied by (1) for $i \in S$ and by (2).

If $|S|$ is odd, $\lfloor \frac{|S|}{2} \rfloor = \frac{|S|-1}{2}$ and (3) is not implied.

**Theorem 1** (Chvátal): $\boxed{\text{Any}}$ valid inequality for $\boxed{\text{any}}$ $X$ can be obtained by applying Chvátal-Gomory procedure a finite number of times.

more just a theoretical result

Proof for case $X \subseteq \{0,1\}^n$ cf. L. Wolsey, Integer Programming, Wiley 2021, p. 145-146

$\underline{u} \geq 0 \left\{ \begin{array}{l} A\underline{x} \geq \underline{b} \\ \underline{x} \geq 0 \end{array} \right. \xrightarrow{CG}$ combining with $\underline{u}$ and rounding is one iteration $\Rightarrow$ we get $A\underline{x}_u \leq b_u$ and then we repeat the procedure

Given any fractional extreme point $\underline{x}^*_{LP}$ of $P$, $\boxed{\exists\ \underline{u} \geq \underline{0}}$ such that the CG inequality $\lfloor \underline{u}^t A \rfloor \underline{x} \leq \lfloor \underline{u}^t \underline{b} \rfloor$ is valid for $X$ and violated by $\underline{x}^*_{LP}$.

we $\exists$ a CG iteration that allows to eliminate any LP fractional extreme point

**Definition**: Denote by $A^1\underline{x} \leq \underline{b}^1$ all inequalities obtainable by varying $\underline{u}$ in $\mathbb{R}^m_+$.
$P_1 = \{\underline{x} \in \mathbb{R}^n_+ \ : \ A\underline{x} \leq \underline{b}, A^1\underline{x} \leq \underline{b}^1\}$ is the **first Chvátal closure** of $P$.

Obviously $P_1 \subseteq P$, and $P_1 = P$ if and only if $P$ has no fractional vertices, that is $P = conv(X)$.

If $P_1 \neq conv(X)$, we can iterate to obtain Chvátal closures $P_k$ of (higher) rank $k$, with $k \geq 2$.

ideal formulation

**Definition**: The smallest integer $k$ such that $\boxed{P_k = conv(X)}$ is the **Chvátal rank** of $conv(X)$ with respect to the formulation $P$.

how many times we have to apply CG procedure to get to the ideal formulation

Clearly $P_k = conv(X) \subset \ldots \subset P_2 \subset P_1 \subset P$.

# 3.6.3 Gomory fractional/integer cutting planes – Review

Generic ILP

$$\min\{\, \underline{c}^t\underline{x} \;:\; \overbrace{A\underline{x} = \underline{b}}^{\text{standard form}},\ \underline{x} \geq \underline{0},\ \underline{x} \in \mathbb{Z}^n \}$$

where $A \in \mathbb{Z}^{m\times n}$, $\underline{b} \in \mathbb{Z}^{m\times 1}$ and $n > m$.

Assumption: rank($A$)=m

Idea: At each iteration, generate C-G cuts exploiting the optimal basic feasible solution $\underline{x}^*_{LP}$ of the current LP relaxation.

$A = (\underbrace{B}_{m \text{ cols}} \,\vdots\, N)$ $\quad \underline{x} = \begin{pmatrix} \underline{x}_B \\ \hdashline \underline{x}_N \end{pmatrix}$

$B$ is a basis of $A$ associated with $\underline{x}^*_{LP}$.

$A\underline{x} = b$
$B\underline{x}_B + N\underline{x}_N = \underline{b}$
$B\underline{x}_B = \underline{b} - N\underline{x}_N$
$\underline{x}_B = B^{-1}(\dots)$

$A\underline{x} = \underline{b}$, $\underline{x} \geq \underline{0}$ can be expressed in canonical form as

$$\underline{x}_B = B^{-1}\underline{b} - \underbrace{B^{-1}N}_{\text{call it } \bar{A}}\underline{x}_N \quad \text{with } \underline{x}_B \geq \underline{0} \text{ and } \underline{x}_N \geq \underline{0},$$

which emphasizes $\underline{x}^*_{LP} = (\underline{x}_B, \underline{x}_N) = (B^{-1}\underline{b}, \underline{0})$.

If $\underline{x}^*_{LP} = B^{-1}\underline{b}$ integer, $\underline{x}^*_{LP}$ is also optimal for ILP.

If $\underline{x}^*_{LP}$ is fractional, generate a C-G cut violated by $\underline{x}^*_{LP}$.

∃ a component fractional

Let $x^*_h$ be a fractional basic variable and row $t$ of the canonical form

$$x_h + \sum_{j \in N} \overline{a}_{tj} x_j = \overline{b}_t \ (= x^*_h) \tag{4}$$

where $N$ corresponds to non basic variables.

Observation: Equation (4) amounts to take $\underline{u}^t = \underline{e}^t_t B^{-1}$ where $\underline{e}_t$ is the $t$-th $m$-dimensional unit vector.

Applying CG rounding to (4):

the **integer form** of the **Gomory** fractional **cut** generated from row $t$ of LP relaxation

$$\boxed{x_h + \sum_{j \in N} \lfloor \overline{a}_{tj} \rfloor x_j \leq \lfloor \overline{b}_t \rfloor.} \tag{5}$$

Valid for $X$ but violated by $\underline{x}^*_{LP}$.

Substracting (5) from (4):

the **fractional form** of the **Gomory cut** generated from row $t$ of LP relaxation

$$\sum_{j \in N} (\overline{a}_{tj} - \lfloor \overline{a}_{tj} \rfloor) x_j \geq \overline{b}_t - \lfloor \overline{b}_t \rfloor. \qquad (6)$$

If $\{a\} := a - \lfloor a \rfloor \geq 0$ denotes the *fractional part* of $a \in \mathbb{R}$, (6) is equivalent to

$$\sum_{j \in N} \{\overline{a}_{tj}\} \, x_j \geq \{\overline{b}_t\}.$$

Recall: $\{4/3\} = 1/3$ but $\{-4/3\} = -4/3 - (-2) = 2/3$

The fractional and integer forms of a Gomory cut are equivalent.

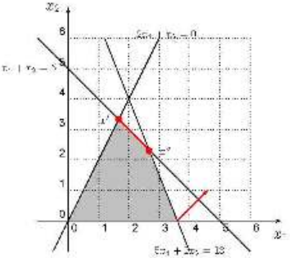
<u>Observation</u>: The difference (slack) between the lhs and rhs of (5) and hence of (6) is always integer when $\underline{x}$ is integer.

Minimal computational requirements.

**Example**:

$$\begin{array}{rl} \max & x_1 + x_2 \\ \text{s.t.} & x_1 + x_2 \leq 5 \\ & -2x_1 + x_2 \leq 0 \\ & 5x_1 + 2x_2 \leq 18 \\ & x_1, x_2 \in \mathbb{Z}^+ \end{array}$$

1. Graphical solution of LP relaxation:




Two optimal basic solutions: $\underline{x}' = (5/3, 10/3)$ and $\underline{x}'' = (8/3, 7/3)$ of value 5.

2. LP relaxation in standard form:

$$\begin{array}{rl} \max & x_1 + x_2 \\ \text{s.t.} & x_1 + x_2 + x_3 = 5 \\ & -2x_1 + x_2 + x_4 = 0 \\ & 5x_1 + 2x_2 + x_5 = 18 \\ & x_1, \ldots, x_5 \geq 0 \end{array}$$

3. Canonical form w.r.t. the optimal basic solution $\underline{x}'' = (8/3, 7/3, 0, 3, 0)$:

$$\begin{array}{l} x_1 - \frac{2}{3}x_3 + \frac{1}{3}x_5 = \frac{8}{3} \\ x_2 + \frac{5}{3}x_3 - \frac{1}{3}x_5 = \frac{7}{3} \\ -3x_3 + x_4 + x_5 = 3 \end{array}$$

Gomory cut derived from $x_1$ row:

- integer form: $x_1 - x_3 \leq 2$
- fractional form: $\frac{1}{3}x_3 + \frac{1}{3}x_5 \geq \frac{2}{3}$

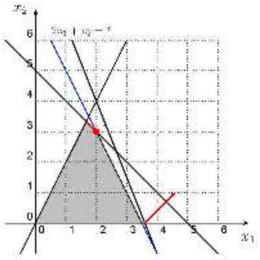
Gomory cut derived from $x_2$ row:

- integer form: $x_2 + x_3 - x_5 \leq 2$
- fractional form: $\frac{2}{3}x_3 + \frac{2}{3}x_5 \geq \frac{1}{3}$

4. Express Gomory cut associated with $x_1$ as a function of $x_1$ and $x_2$.

   Substituting $x_3 = 5 - x_1 - x_2$ in $x_1 - x_3 \leq 2$, we obtain the cut: $2x_1 + x_2 \leq 7$.

5. Add this Gomory cut to LP relaxation and find an optimal solution.




   Adding $2x_1 + x_2 \leq 7$ to the original formulation, we obtain an optimal solution of new LP relaxation $\underline{x}^*_{LP} = (2, 3)$ with $z^*_{LP}5$.

   Since $\underline{x}^*_{LP}$ is integer, it is also optimal for ILP.

moreover, Gomory cuts made in a certain order

**Theorem 2** (Gomory): A lexicographic cutting plane method based on Gomory fractional/integer cuts terminates after a finite number of iterations.

Provided a careful choice of (i) the basis defining the optimal solution we intend to cut off and (ii) the row of the tableau used to generate the cut.

In practice: Huge number of iterations and such cuts tend to become weaker after a few iterations.

Strategy: Introduce several cuts at each iterations, e.g., all those with $\{\bar{b}_t\} > \varepsilon = 0.01$

**Recall**: Gomory fractional/integer cuts are generated via simple integer rounding.

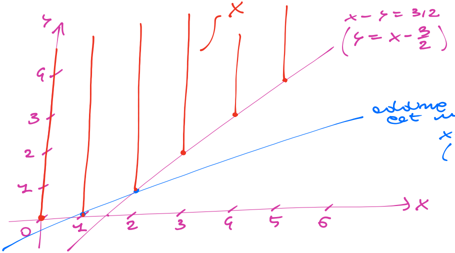
But these Gomory cuts are easy to implement, but not very effective. A better variant is the next one

## 3.6.4 Mixed integer rounding inequalities

Consider $X = \{(x, y)^t \in \boxed{\mathbb{Z}} \times \boxed{\mathbb{R}^+} : \ x - y \leq b\}$ where $b \in \mathbb{Q} \setminus \mathbb{Z}$.

mixed rounding, we have an integer variable and a continuous one

Illustration for $b = 3/2$:




adding this inequality will get us have $P = conv(X)$

$x - 2y = 1$

$(y = \frac{1}{2}x - \frac{1}{2})$

**Proposition 2**: The **mixed-integer rounding** (**MIR**) inequality

$$\boxed{x - \frac{1}{1 - \{b\}} y \leq \lfloor b \rfloor} \qquad (7)$$

is valid for *conv*$(X)$.

For $b \in \mathbb{R}$, $\{b\} := b - \lfloor b \rfloor \geq 0$ denotes the *fractional part* of $b$.

eg

$1{,}5 - 1 = 0{,}5$

Observation: $conv(\{(x,y)^t \in \mathbb{Z} \times \mathbb{R}^+ \ : \ x - y \leq b\})$ is defined by $x - y \leq b$, $y \geq 0$ and $x - \frac{1}{1-\{b\}} y \leq \lfloor b \rfloor$.

## 3.6.5 Gomory mixed integer cutting planes

Generic MILP

now MILP, before by with Gomory cuts we were with ILP

$$
\begin{aligned}
\min \quad & \underline{c}_1^t \underline{x} + \underline{c}_2^t \underline{y} & \\
s.t. \quad & A_1 \underline{x} + A_2 \underline{y} = \underline{b} & (8)\\
& \underline{x} \geq \underline{0},\ \underline{y} \geq \underline{0} & (9)\\
& \underline{x} \text{ integer.} & (10)
\end{aligned}
$$

$(\underline{x}^*_{LP}, \underline{y}^*_{LP})$ an optimal basic feasible solution of LP relaxation.

Denote by $N_1/N_2$ the indices in $N$ corresponding to integer/continuous variables.

If $\underline{x}^*_{LP}$ not integer ($(\underline{x}^*_{LP}, \underline{y}^*_{LP})$ not optimal), $\exists$ an index $h \in B$ such that $x^*_h \notin \mathbb{Z}$.

Canonical form w.r.t. optimal basis contains a row, say $t$-th one:

$$x_h + \sum_{j \in N_1} \overline{a}_{tj} x_j + \sum_{j \in N_2} \overline{a}_{tj} y_j = \overline{b}_t \qquad (11)$$

for appropriate $\overline{a}_{tj}$ and $\overline{b}_t$, with $\overline{b}_t \notin \mathbb{Z}$.

apply the MIR (now mixed as we have also the y, continuous)

Notation: For any $a \in \mathbb{R}$, $a^+ = \max\{a, 0\}$ and $a^- = \max\{-a, 0\}$.

similar to the classical Gomory cut:
$x_t + \sum_{j \in N} \lfloor \bar{a}_{tj} \rfloor x_j \leq \lfloor \bar{b}_t \rfloor$
but updated to account for the (possible) presence of continuous variables

when running OPT solvers, at the end we get feedback on how many GMI iterations were made

**Proposition 3**: The **Gomory mixed integer** (**GMI**) inequality

$$\boxed{x_h + \sum_{j \in N_1} \left( \lfloor \bar{a}_{tj} \rfloor + \frac{(\{\bar{a}_{tj}\} - \{\bar{b}_t\})^+}{1 - \{\bar{b}_t\}} \right) x_j \leq \lfloor \bar{b}_t \rfloor + \sum_{j \in N_2} \frac{(\bar{a}_{tj})^-}{1 - \{\bar{b}_t\}} y_j} \qquad (12)$$

is valid for the feasible region (8)-(10) and is violated by $(\underline{x}^*_{LP}, \underline{y}^*_{LP})$.

optimal sol of the LP relax

- we get a better constraint (wrt pure Gomory cuts)
- but the coeff now may be not integers

**Remarks**: For pure ILP

i) GMI cut (12) is potentially stronger than corresponding fractional Gomory cut $\left(\frac{(\{\bar{a}_{tj}\} - \{\bar{b}_t\})^+}{1 - \{\bar{b}_t\}} \geq 0 \text{ and } y_j = 0 \ \forall j \in N_2\right)$,

ii) coefficients are not integer anymore.

Unlike for fractional Gomory cuts in pure ILP, no finite termination guarantee for GMI cuts but very effective in practice (see later).

# 3.7 Strong valid inequalities for structured ILP problems

Studying the problem structure, we can derive strong valid inequalities yielding better approximations of $conv(X)$ and tighter bounds.

For any $P = \{\underline{x} \in \mathbb{R}^n_+ \; : \; A\underline{x} \leq \underline{b}\}$

a vi

**Definition**: Given $\underline{\pi}^t \underline{x} \leq \pi_0$ and $\underline{\mu}^t \underline{x} \leq \mu_0$ both valid for $P$, $\boxed{\underline{\pi}^t \underline{x} \leq \pi_0}$ **dominates** $\boxed{\underline{\mu}^t \underline{x} \leq \mu_0}$ if $\exists \; u > 0$ such that $\boxed{u\underline{\mu} \leq \underline{\pi} \text{ and } \pi_0 \leq u\mu_0}$ with $(\underline{\pi}, \pi_0) \neq (u\underline{\mu}, u\mu_0)$.

another one

the feasible reg of the 1st one was included in the feasible region of the 2nd one

⇒ again, smaller means better, dominating

Example: $x_1 + 3x_2 \leq 4$ dominates $2x_1 + 4x_2 \leq 9$

**Definition**: A valid $\underline{\pi}^t \underline{x} \leq \pi_0$ is **redundant** in the description of $P$ if

$\exists\ k \geq 2$ valid $\underline{\pi}^i \underline{x} \leq \pi_0^i$ for $P$ with $u_i > 0$, $1 \leq i \leq k$, such that

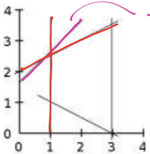
there exist k rows

corresponding multipliers

$$(\sum_{i=1}^{k} u_i \underline{\pi}^i)\underline{x} \leq \sum_{i=1}^{k} u_i \pi_0^i \quad \text{dominates} \quad \underline{\pi}^t \underline{x} \leq \pi_0.$$

Example:

$P = \{(x_1, x_2) \in \mathbb{R}^2_+ \ : \ -x_1 + 2x_2 \leq 4,\ -x_1 - 2x_2 \leq -3,\ -x_1 + x_2 \leq 5/3,\ 1 \leq x_1 \leq 3\}$




this one is dominated by the combination of the red ones, so it is redundant

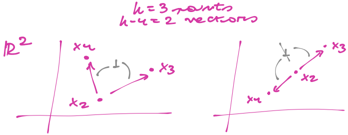
eliminating it won't change the region P description

$-x_1 \leq -1$
$-x_1 + 2x_2 \leq 4$ $\left( \& \begin{array}{l} u_1 = 1/2 \\ u_2 = 1/2 \end{array} \right) \Rightarrow$ domination of the constraint $-x_1 + x_2 \leq 5/3$

Observation: It can be very difficult to check redundancy. In practice, try to avoid dominated inequalities.

## 3.7.1 Faces and facets of polyhedra

Consider any $P = \{\underline{x} \in \mathbb{R}^n \ : \ A\underline{x} \leq \underline{b}\}$.



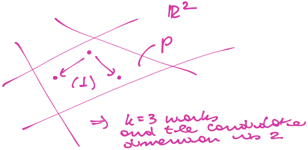

**Definitions**

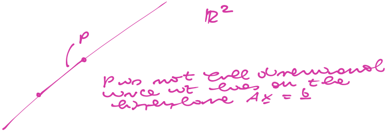
- k points in $\mathbb{R}^n$
  $\underline{x}_1, \ldots, \underline{x}_k \in \mathbb{R}^n$ are **affinely independent** if $k-1$ vectors $\underline{x}_2 - \underline{x}_1, \ldots, \underline{x}_k - \underline{x}_1$ (or $k$ vectors $(\underline{x}_1, 1), \ldots, (\underline{x}_k, 1)$ in $\mathbb{R}^{n+1}$) are linearly independent.

- The **dimension** of $P$, $\dim(P)$, is equal to the maximum number of affinely independent points of $P$ minus 1.

- $P$ is **full dimensional** if $\boxed{\dim(P) = n}$, i.e., no $\underline{a}^t \underline{x} \leq b$ is satisfied with equality by all points $\underline{x} \in P$.
  inequalities
  of the polyhedron

Illustrations:



Assumption: $dim(P) = n$

**Theorem**: If $dim(P) = n$, $P$ admits a unique minimal description

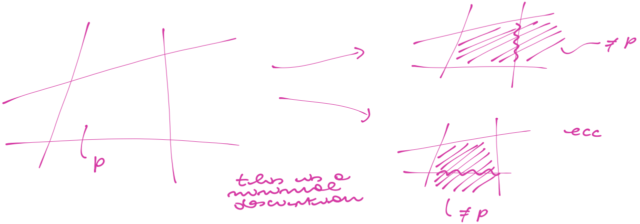
$$P = \{\underline{x} \in \mathbb{R}^n : \underline{a}_i^t \underline{x} \leq b_i, i = 1, \ldots, m\}$$

where each inequality is unique (within a positive multiple.)

what does minimal mean?

Each inequality is necessary (deletion yields a different polyhedron).

Moreover, each valid inequality for $P$ which is not a positive multiple of one $\underline{a}_i^t \underline{x} \leq b_i$ is redundant.



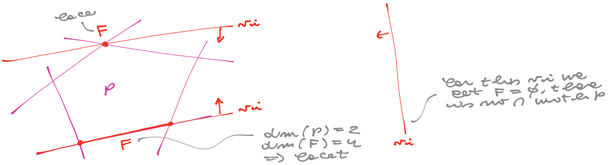
## 1) Alternative characterization of necessary valid inequalities

→ the ∩ of P with the vi

**Definitions** the set of points of P which satisfy the vi at equality

- Let $F = \{\underline{x} \in P : \underline{\pi}^t \underline{x} = \pi_0\}$ for any valid $\underline{\pi}^t \underline{x} \leq \pi_0$ (a vi) for $P$. Then $F$ is a **face** of $P$ and $\underline{\pi}^t \underline{x} \leq \pi_0$ *represents* or *defines* $F$.
- If $F$ is a face of $P$ and $\boxed{\dim(F) = \dim(P) - 1}$, then $F$ is a **facet** of $P$.

Illustrations:




Consequences: The faces of a polyhedron are polyhedra, a polyhedron has a finite number of faces.

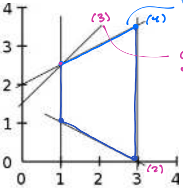
**Theorem**: If $P$ is full dimensional, a valid inequality is necessary to describe $P$ $\boxed{\text{if and only if}}$ it defines a facet of $P$, i.e., if $\exists\ n$ affinely independent points of $P$ satisfying it (the vi) at equality.

the vertices of the segment of the facet above

Example:

Consider $P \subset \mathbb{R}^2$ described by:

$$
\begin{aligned}
x_1 \quad + \quad 2x_2 &\leq 4 && (1)\\
-x_1 \quad - \quad 2x_2 &\leq -3 && (2)\\
-x_1 \quad + \quad x_2 &\leq \frac{3}{2} && (3)\\
x_1 &\leq 3 && (4)\\
x_1 &\geq 1 && (5)
\end{aligned}
$$




facet related to constr (4)

constr 3 instead defines a face, not a facet, so it is not necessary

Verify that $P$ is full dimensional (dim($P$)=2).

Which inequalities define facets of $P$ or are redundant?

## 2) Showing that a valid inequality is facet defining

Consider $X \subset \mathbb{Z}^n_+$ and a valid inequality $\underline{\pi}^t \underline{x} \leq \pi_0$ for $X$.

Assumption: $conv(X)$ is bounded and $dim(conv(X)) = n$.

Simple approaches to show that $\underline{\pi}^t \underline{x} \leq \pi_0$ defines a facet of $conv(X)$:

the min of equalities

1) Apply the definition: Find $n$ points $\underline{x}^1, \ldots, \underline{x}^n \in X$ satisfying $\underline{\pi}^t \underline{x} = \pi_0$ and prove that they are affinely independent.

the geometric idea we saw before

2) Indirect approach:

(i) Select $t$ points $\underline{x}^1, \ldots, \underline{x}^t \in X$, with $t \geq n$, satisfying $\underline{\pi}^t \underline{x} = \pi_0$.
Suppose that they all belong to a generic hyperplane $\underline{\mu}^t \underline{x} = \mu_0$.

this would imply being all in

params to be determined

(ii) Solve linear system

which defines the hyperplane of the pts

$$\sum_{j=1}^{n} \mu_j x_j^k = \mu_0 \qquad \text{for } k = 1, \ldots, t$$

in $n+1$ unknowns $\mu_0, \mu_1, \ldots, \mu_n$.

(iii) If the only solution is $\boxed{(\underline{\mu}, \mu_0) = \lambda(\underline{\pi}, \pi_0)}$ with $\lambda \neq 0$, then $\underline{\pi}^t \underline{x} \leq \pi_0$ defines a facet of $conv(X)$.

## Example:

Consider $X = \{(\underline{x}, y) \in \mathbb{R}^m \times \{0,1\} \,:\, \sum_{i=1}^m x_i \le my,\ 0 \le x_i \le 1\ \forall i\}$

i) Verify that $dim(conv(X)) = m + 1$.

- we should exhibit $m+2$ points $\in X$ and show that they are affinely $\perp$
- we can take $(\underline{0}|0)$ and $(\underline{e}_i|1)$ with $1 \le i \le m$ ; $(\underline{0}|1)$ $\quad (0 \cdots 0\, 1/_i\, 0 \cdots 0)$ — are $m+2$ points, affinely $\perp$ and belong to $conv(X)$

ii) Show (approach 2) that, for each $i$, valid $x_i \le y$ defines a facet of $conv(X)$.

Consider $m+1$ points which
- are feasible ($\in X$)
- and satisfy the ineq at equality

$\Rightarrow$ $(\underline{0}, 0)$ and $(\underline{e}_i, 1)$ work, the rest of these plus new ones of the form $(\underline{e}_i + \underline{e}_{i'}, 1)$ $\forall i' \ne i$

where $(\underline{0}, 0)$ means that $y$ is the coord $x_{m+1}$

These are the candidate $t$ $(= m+1)$ points.
Now we look for the hyperplane that contains them.

Since $(\underline{0}, 0) \in H$, the hyperplane defined by $\sum_{j=1}^m \pi_j x_j + \pi_{m+1} y = \pi_0$

$\Rightarrow \Sigma 0 + 0 = 0 = \pi_0$

Since $(\underline{e}_1, 1) \in H$, then $\Rightarrow \pi_i \cdot 1 + \pi_{m+1} \cdot 1 = 0 \Rightarrow \pi_{m+1} = -\pi_i$

about the points $(\underline{e}_i + \underline{e}_{i'}, 1)$ we get

$\Rightarrow \cancel{\pi_i} + \pi_{i'} + \underbrace{\pi_{m+1}}_{\text{this was } \cancel{-\pi_i}} = 0 \Rightarrow \pi_{i'} = 0\ \forall i' \ne i$

so we get $H$ to be $\pi_i x_i + \pi_i y = 0$

and the $x_i \le y$ defines then a facet of $conv(X)$

## 3.7.2 Cover inequalities for binary knapsack problem

Consider $X = \{\underline{x} \in \{0,1\}^n : \sum_{j=1}^{n} a_j x_j \leq b\}$ with $b > 0$ and $N = \{1, \ldots, n\}$.

the items set

Assumptions: For each $j$, $a_j \leq b$ and $a_j > 0$.

we cant fit wt(C) in the knapsack

**Definition**: A subset $C \subseteq N$ is a **cover** for $X$ if $\sum_{j \in C} a_j > b$.

removing any item will make C fit now

A cover is **minimal** if, for each $j \in C$, $C \setminus \{j\}$ is not a cover.

Example: For $X = \{\underline{x} \in \{0,1\}^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$

the minimal cover is $C = \{1, 2, 3\}$
while $\{3, 4, 5, 6, 7\}$ is a non-minimal cover

**Proposition**: If $C \subseteq N$ is a cover for $X$, the **cover inequality**

$$\boxed{\sum_{j \in C} x_j \leq |C| - 1}$$

is valid for $X$.

we need to delete at least one (S.. -1) item from the collection C

Example cont.:

$x_1 + x_2 + x_3 \leq 3 - 1 = 2$
$x_3 + x_4 + x_5 + x_6 + x_7 \leq 5 - 1 = 4$

**Proposition**: If $C \subseteq N$ is a cover for $X$, the cover inequality

$$\sum_{j \in C} x_j \leq |C| - 1$$

defines a facet of $\boxed{P_C := conv(X) \cap \{\underline{x} \in \mathbb{R}^n : \ x_j = 0, j \in N \setminus C\}}$ if and only if $C$ is a minimal cover.

we look for optimal set only in $C$ (as items $\notin C$ are set to be with $x_j = 0$)

## 1) Separation of cover inequalities

eg $\underline{x}^*_{LP}$

Separation problem: Given a fractional $\underline{\overline{x}}$ with $0 \leq \overline{x}_j \leq 1$, $1 \leq j \leq n$, find a cover inequality violated by $\underline{\overline{x}}$ (or establish that none exists.)

Since $\sum_{j \in C} x_j \leq |C| - 1$ can be written as $\sum_{j \in C}(1 - x_j) \geq 1$, it amounts to question:

$$\exists\ C \subseteq N \text{ such that } \underbrace{\sum_{j \in C} a_j > b}_{\text{it is a cover}} \text{ and } \underbrace{\sum_{j \in C}(1 - \overline{x}_j) < 1}_{\text{that constr is violated}}?$$

we can formulate the question in an ILP problem

If $\underline{z} \in \{0,1\}^n$ incidence vector of $C \subseteq N$, it is equivalent to:

$$\zeta = \min\{\sum_{j \in N}(1 - \overline{x}_j)z_j : \underbrace{\sum_{j \in N} a_j z_j > b}_{\text{the selected items should correspond to a cover}}, \underline{z} \in \{0,1\}^n\} \boxed{< 1?}$$

**Proposition**:

(i) $\boxed{\text{If } \zeta \geq 1}$, $\underline{\overline{x}}$ satisfies all cover inequalities.

(ii) $\boxed{\text{If } \zeta < 1}$ with optimal solution $\underline{z}^*$, then $\sum_{j \in C} x_j \leq |C| - 1$ with $C = \{j : z_j^* = 1, 1 \leq j \leq n\}$ is violated by $\underline{\overline{x}}$ by a quantity $1 - \zeta$.

Example:

$$\begin{array}{rl} \max & 5x_1 + 2x_2 + x_3 + 8x_4 \\ s.t. & 4x_1 + 2x_2 + 2x_3 + 3x_4 \leq 4 \\ & x_j \in \{0,1\} \quad \forall j \in \{1,\dots,4\} \end{array}$$

Optimal solution of LP relaxation $\underline{x}^*_{LP} = (1/4, 0, 0, 1)^t$ of value 9.25.

- we solve the LP relaxation, we see that it is not integer, so we look for a cutting plane
- we have to solve the separation problem:

$$\zeta = \min \Big\{ \sum_{j \in N} (1 - \bar{x}_j) z_j \quad st \ \sum_{j \in N} a_j z_j > b \quad z_j \in \{0,1\} \ \forall j \Big\}$$

$$\min \ \tfrac{3}{4} z_1 + z_2 + z_3$$

$$st \ 4z_1 + 2z_2 + 2z_3 + 3z_4 \geq 4$$

Obs: we have seen a knapsack problem, but we don't have to solve it to optimality

- now we get $z^* = (1\ 0\ 0\ 1)$ with $\zeta = 3/4$ and so we get the cover inequality: $x_1 + x_4 \leq 1$
  which cuts the $\underline{x}^*_{LP}$ as $1 - \zeta = 1/4$

Separation problem is $NP$-hard, in practice fast heuristics.

## 2) Strengthening cover inequalities

**Proposition**: If $C \subseteq N$ is a cover for $X$, the **extended cover inequality**

$$\boxed{\sum_{\boxed{j \in E(C)}} x_j \leq |C| - 1}$$

still $|C|$, not of $E$

is valid for $X$, where $E(C) = C \cup \{j \in N \,:\, a_j \geq a_i \text{ for all } i \in C\}$.

C unito a

tutti gli altri items che occupano più spazio degli elementi in C

Example cont.: $X = \{\underline{x} \in \{0,1\}^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$

cover $C = \{3, 4, 5, 6\}$

$\max_{i \in C}(\text{size}_i) = 6$, but $x_1$ and $x_2$ have sizes $(11, 6) \geq 6$

$\Rightarrow E(C) = C \cup \{1, 2\}$ and the extended inequality is

$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq |C| - 1 = 3$

Systematic way to strengthen a cover inequality to obtain a facet defining one.

Example of $\boxed{\text{lifting procedure}}$:

if $=1$

$$X = \{\underline{x} \in \{0,1\}^7 : 11x_1 + 6\cancel{x_2} + 6x_3 + 5x_4 + 5x_5 + 4x_6 + \cancel{x_7} \le 19\}$$

$19 - 11 = 8$

Minimal cover $C = \{3, 4, 5, 6\}$ with $x_3 + x_4 + x_5 + x_6 \le 3$.

Consider $x_j$ with $j \in N \setminus C$ in the order $x_1$, $x_2$ and $x_7$.

The largest $\alpha_1$ such that $\alpha_1 x_1 + x_3 + x_4 + x_5 + x_6 \le 3$ is valid for $X$ is

- if $x_1 = 0 \Rightarrow \forall \alpha_1$
- if $x_1 = 1$ we have that

max of that quantity st we still get a feasible set when also $x_1 = 1$

$$\alpha_1 \le 3 - x_3 - x_4 - x_5 - x_6 \le$$

$$\le 3 - \left\{ \begin{array}{l} \max\ x_3 + x_4 + x_5 + x_6 \\ \text{st } 6x_3 + 5x_4 + 5x_5 + 4x_6 \le 19 - 11 = 8 \\ x_i \in \{0,1\}^4 \end{array} \right\}$$

$$= \ldots = 3 - 1 = 2$$

now about $\alpha_2$ we have that $\alpha_2 x_2 + 2x_1 + x_3 + x_4 + x_5 + x_6 \le 3$ is valid for $X$ if

$$\alpha_2 \le 3 - \left\{ \begin{array}{l} \max\ 2x_1 + x_3 + x_4 + x_5 + x_6 \\ \text{st } 11x_1 + 6x_3 + 5x_4 + 5x_5 + 4x_6 \le 19 - 6 = 13 \\ x_i \in \{0,1\}^5 \end{array} \right\}$$

$$= \ldots$$

and similarly for $x_7$ ...

## Lifting procedure for cover inequalities

Let $j_1, \ldots, j_r$ be an ordering of $N \setminus C$ and set $t = 1$.

$\sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j \leq |C| - 1$ valid inequality obtained at iteration $t-1$.

<u>Iteration $t$</u>: Determine the maximum $\alpha_{j_t}$ such that

$$\alpha_{j_t} x_{j_t} + \sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j \leq |C| - 1$$

is valid for $X$ by solving (binary knapsack) problem

$$\begin{array}{rl} \sigma_t = \max & \sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j \\ s.t. & \sum_{i=1}^{t-1} a_{j_i} x_{j_i} + \sum_{j \in C} a_j x_j \leq b - a_{j_t} \\ & \underline{x} \in \{0,1\}^{|C|+t-1} \end{array}$$

and setting $\alpha_t = |C| - 1 - \sigma_t$.

Terminate when $t = r$.

Note: $\sigma_t =$ maximum amount of "space" used up by the variables of indices in $C \cup \{j_1, \ldots, j_{t-1}\}$ when $x_{j_t} = 1$.

**Proposition**: If $C \subseteq N$ is a minimal cover and $a_j \leq b$ for all $j \in N$, the lifting procedure is guaranteed to yield a facet defining inequality of $conv(X)$.

Example cont.:

$$X = \{\underline{x} \in \{0,1\}^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$$

the valid inequality

$$2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$$

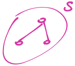
defines a facet of $conv(X)$.

The resulting facet defining inequality depends on the order of variables $N \setminus C$, that is, on the lifting sequence.

## 3.7.3 Strong valid inequalities for TSP

STSP: Given undirected $G = (V, E)$ with $n = |V|$ nodes and a cost $c_e$ for every $e = \{i, j\} \in E$, determine a Hamiltonian cycle of minimal total cost.

$$
\begin{array}{llll}
\min & \sum_{e \in E} c_e x_e & & \\
\text{s.t.} & \sum_{e \in \delta(i)} x_e = 2 & i \in V & \\
& \sum_{e \in E(S)} x_e \leq |S| - 1 & S \subset V,\ S \neq \emptyset & \\
& x_e \in \{0, 1\} & e \in E. &
\end{array}
$$




k nodes in S

$\Rightarrow$ at most k-1 edges

$conv(X)$ with $X = \{\underline{x} \in \{0,1\}^{|E|}$ of Hamiltonian cycles $\}$ is the STSP polytope

**Proposition**: For every $S \subseteq V$ with $2 \leq |S| \leq n/2$ and $n \geq 4$,

the subtours elimination ~~the cut set~~ inequalities

$$\sum_{e \in E(S)} x_e \leq |S| - 1$$

defines a facet of conv(X).

STSP polytope has a very complicated structure. Many classes of facet defining inequalities are known but its complete description is unknown.

## Separation of cut-set inequalities for the ATSP

ILP formulation:

$$
\begin{array}{llll}
\min & \sum_{(i,j)\in A} c_{ij} x_{ij} & & (6)\\
s.t. & \sum_{(i,j)\in\delta^-(j)} x_{ij} = 1 & \forall j & (7)\\
 & \sum_{(i,j)\in\delta^+(i)} x_{ij} = 1 & \forall i & (8)\\
 & \sum_{(i,j)\in\delta^+(S)} x_{ij} \geq 1 & \forall S \subset V \,:\, 1 \in S & (9)\\
 & x_{ij} \in \{0,1\} & \forall (i,j) \in A & (10)
\end{array}
$$

for the cut set version we remove this constr but then [illegible] adjust for it

Cutting plane approach:

Start solving LP relaxation of (6)-(10) without (9), namely

$$
\begin{array}{llll}
\min & \sum_{(i,j)\in A} c_{ij} x_{ij} & & (11)\\
s.t. & \sum_{(i,j)\in\delta^-(j)} x_{ij} = 1 & \forall j & (12)\\
 & \sum_{(i,j)\in\delta^+(i)} x_{ij} = 1 & \forall i & (13)\\
 & x_{ij} \geq 0 & \forall (i,j) \in A, & (14)
\end{array}
$$

and iteratively add some cut set inequalities (we reintroduce some of the constr of (9)) which substantially violate the current $\underline{x}^*_{LP}$.

**Proposition**:

Given $\underline{x}^*_{LP}$ of the current LP relaxation ((11)-(14) with (9) generated so far), a cut-set inequality (9) violated by $\underline{x}^*_{LP}$ can be obtained (if $\exists$) by solving a sequence of instances of the minimum cut problem.

Separation algorithm:

Given $\underline{x}^*_{LP}$, look for $S^* \subseteq V$ with $1 \in S^*$ such that $\overbrace{\sum_{(i,j)\in\delta^+(S^*)} x^*_{ij} < 1}^{\text{the cut inequality (9) is violated}}$.

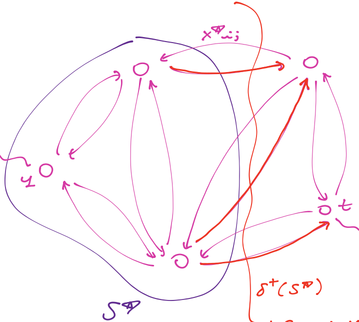
Consider $G^* = (V, A^*)$ with 1 node and $x_{ij}^*$ as capacity $\forall (i,j) \in A^*$

For any choice of the sink $t \in V \setminus \{1\}$ we look for a cut $\delta^+(S^*)$, separating $1 \in S^*$ from $t \notin S^*$, of minimum capacity.

- if cut capacity $< 1$, then $S^*$ describes a cut set inequality violated by $\underline{x}^*_{LP}$
- otherwise it does not exists

For each $t \in V \setminus \{1\}$ a min 1-t cut can be found (also quickly), in polynomial time) and adopted

Illustration:




source $1 \in S^*$

sink $t \notin S^*$ } loop on all the other nodes and treat them as the sink t

consider $G^* = (V, A^*)$
$A^* = \{(i,j) \in A : x^*_{ij} > 0\}$

on the outcome ($\to t$) arcs

the sum of those capacities will be the maximum flow also (by duality), max flow/min cut)

$\Rightarrow$ so we look for a cut of minimum capacity, and

the min cap cut will have capacity

$\sum_{(i,j) \in \delta^+(S^*)} x^*_{ij} \gtrless 1$

- if min cut cap $> 1$ $\Rightarrow$ all the cut set inequalities (for that t) are satisfied
- if min cut cap $< 1$ $\Rightarrow$ we found the violated constraint

Observations:

- The separation problem can be solved in polynomial time.
- The procedure may yield a number of violated cut-set inequalities (one for each $t$).

we add them on the fly, when are needed, since they would be too much all at the beginning

## 3.7.4 Equivalence between separation and optimization

like the set of all possible graphs

A family of LPs $\min\{\, \underline{c}^t \underline{x} \;:\; \underline{x} \in P_o\}$ with $o \in \mathcal{O}$, where $P_o = \{\, \underline{x} \in \mathbb{R}^{n_o} \;:\; A_o \underline{x} \geq \underline{b}_o \,\}$ polytope with rational (integer) coefficients and a very large number of constraints.

Examples:

1) LP relaxation of ATSP with cut-set inequalities ($\mathcal{O}$ set of all graphs)

2) Maximum Matching problem: For each $G = (V, E)$, the matching polytope

$$conv(\{\underline{x} \in \{0,1\}^{|E|} \;:\; \sum_{e \in \delta(i)} x_e \leq 1,\ \forall i \in V\,\})$$

coincides (Edmonds) with

$$\{\underline{x} \in \mathbb{R}_+^{|E|} \;:\; \sum_{e \in \delta(i)} x_e \leq 1,\ \forall i \in V,\ \sum_{e \in E(S)} x_e \leq \frac{|S|-1}{2},\ \forall S \subseteq V \text{ with } |S| \geq 3 \text{ odd}\}.$$

Consider a cutting plane approach.

**Assumption**: The number of constraints $m_o$ of $P_o$ is exponential in $n_o$ but $A_o$ and $\underline{b}_o$ are specified in a concise way (as function of a polynomial number of parameters w.r.t. $n_o$).

size is large, but can be quickly represented

Optimization problem: Given rational polytope $P \subseteq \mathbb{R}^n$ and $\underline{c} \in \mathbb{Q}^n$, find $\underline{x}^* \in P$ minimizing $\underline{c}^t\underline{x}$ over $\underline{x} \in P$ or establish that $P$ is empty.

N.B.: $P$ assumed to be bounded just to avoid unbounded problems.

corresponding

Separation problem: Given rational polytope $P \subseteq \mathbb{R}^n$ and $\underline{x}' \in \mathbb{Q}^n$, establish that $\underline{x}' \in P$ or determine a cut that separates $\underline{x}'$ from $P$.

**Theorem**: (consequence of Grötschel, Lovász, Schriver 1988 theorem)

The separation problem (for a family of polyhedra) can be solved in polynomial time in $n$ and $\log U$ if and only if the optimization problem (for that family) can be solved in polynomial time in $n$ and $\log U$, where $U$ is an upper bound on all $a_{ij}$ and $b_i$.

Proof based on *Ellipsoid method*, first polynomial algorithm for LP.

**Corollary**: The LP relaxation of ILP formulation with cut-set inequalities for ATSP can be solved in polynomial time in spite of its exponential size.

as we now that the separation problem was solvable in polinomial time

## 3.7.5 Remarks on cutting plane methods

Generic Discrete Optimization problem

$$\min\{ \underline{c}^t \underline{x} \; : \; \underline{x} \in X \subseteq \mathbb{R}^n_+ \}$$

When designing a cutting plane method

- Describing families of strong (possibly facet defining) valid inequalities for $conv(X)$ can be difficult.
- The separation problem for a given family $\mathcal{F}$ may be computationally challenging (if $\mathrm{NP}$-hard devise heuristics).

  but remember: we dont need to solve sep to optimality!

- Even when finite convergence is guaranteed (e.g., Gomory cuts), pure cutting plane methods tend to be very slow.

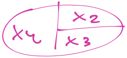
  just find a violated v.i. and add it to the formulation

Polyhedral Combinatorics is the subfield studying the polyhedral structure of ideal formulations.

# 3.8 Branch and bound and Cut

**Idea**: Embed strong valid inequalities into a Branch-and-Bound framework to be able to solve hard/large problems to optimality.

→ *Branch-and-Cut* method




- solve the LP relaxation on each of the $X_i$
- if all are integers stop, otherwise branch and so on

(Strong) valid inequalities are generated throughout the branching tree.

idea: branching and relaxing often was not good, so instead we add cuts rather than directly branching/splitting the $X_i$

Advantages:

- stronger LP relaxations of subproblems yield tighter dual bounds which improve Branch and Bound efficiency,
- slow convergence of pure cutting plane method is contrasted by branching steps.

so when after adding cuts we reach a stall, a no improvement, we branch

Trade-off between computational load of reoptimization and quality of the formulations (bounds).

## **Main components of Branch and Cut** (min problem)

Preprocessing

Delete redundant constraints, strengthen the constraint coefficients and r.h.s. terms, fix variables (whenever possible).

Primal heuristics — before starting the B&C we run for one time a heuristic (to get a first integer upper bound)

Tighter upper bounds lead to a more efficient implicit enumeration.

Cutting planes pool

Violated valid inequalities and facets are added by solving corresponding separation problems exactly or heuristically. Many of them are simultaneously added at each node.

Branching strategy

Choice of the fractional branching variable based on one/mix of criteria (with largest cost coefficient, "most promising" one based on estimate,...).

Postprocessing — like a massage/refinement on the last LP relaxation

When $\underline{x}^*_{LP}$ of value $z_{LP}$ is not integer, primal heuristic yields a feasible $\underline{x}_{heur}$ such that $z_{LP} \leq z^* \leq z_{heur}$ ($\underline{x}_{heur}$ often derived by "smart" rounding).

exploiting the structure of the problem

For **flow chart** of Branch and Cut, see L. Wolsey, Integer Programming, p. 158.

(we with resource constr)

For an **example** of application to the generalized assignment problem

$$
\begin{aligned}
\min\ z = \quad & \textstyle\sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} & \\
\text{s.t.} \quad & \textstyle\sum_{j \in J} x_{ij} = 1 & \forall i \in I \\
& \textstyle\sum_{i \in I} w_{ij} x_{ij} \leq b_j & \forall j \in J \\
& x_{ij} \in \{0,1\} & \forall i \in I, \forall j \in J,
\end{aligned}
$$

see computer lab 2 and L. Wolsey, Integer Programming, p. 157-160.

**Computer lab 2**: separate cover inequalities and evaluate the impact of adding them at the root node of the branching tree (Cut and Branch).

we strengthen only the root problem formulation, while we could alternatively polish also the branched problems

Branch and Cut methods solve to optimality a wide range of discrete optimization problems.

Example: Concorde algorithm for TSP (see http://www.math.uwaterloo.ca/tsp/)

# Impact of different features in a MILP solver

From R. Bixby, M. Fenelon, Z. Gu, E. Rothberg and R. Wunderling, Mixed integer programming: A progress report, M. Grötschel ed., The sharpest cut, MPS/SIAM Series in Optimization (2004) 309-326.

2002 "new generation" Cplex solver for MILPs

Computational experiments on set of 106 benchmark instances

**Different features**

| Feature | Speedup factor |
|---|---:|
| Cuts | **54** |
| Preprocessing | **11** |
| Variable fixing | 3 |
| Heuristics | 1.5 |

Average speedup for each feature (enabling that feature versus disabling it, while keeping all others active).

## Different types of cutting planes

| Cut type | Speedup factor |
|---|---|
| GMI | **2.5** |
| MIR | **1.8** |
| Knapsack cover | 1.4 |
| Flow cover | 1.2 |
| Implied bounds | 1.2 |
| Path | 1.04 |
| Clique | 1.02 |
| GUB cover | 1.02 |

MIR cuts: heuristic aggregation of constraints with mixed integer rounding.

Gomory mixed integers cuts

GMI and MIR cuts implementations account for finite precision (avoid invalid cuts or cuts that could slow down LP solution).

# 3.9 Column generation method

Many relevant decision-making problems can be formulated as ILP problems with a very large (exponential) number of variables.

Examples: cutting stock, crew scheduling, vehicle routing, combinatorial auctions,
multicommodity flows,...

General idea:

- enumerate all partially feasible solutions and represent any additional constraints in a set covering/packing/partitionning type of formulation.
- do not consider all variables explicitly, new variables are generated when needed.

else we deal with ILPs with an exp # of constraints

## Example: 1-D cutting stock problem

A paper company produces large rolls of width $W$.

Demand: $b_i$ small rolls of width $w_i$ ($w_i \leq W$), $i \in I = \{1, \ldots, m\}$.

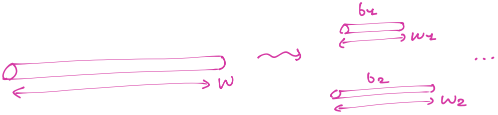
Small rolls obtained by cutting large rolls according to certain patterns.

Given

- large rolls of width $W$,
- demands for $b_i$ small rolls of width $w_i$, with $i \in I$

decide how to cut large rolls into small ones so as to minimize the number of large rolls used, while satisfying demand.

Illustration:




$\mathrm{NP}$-hard problem

Classical ILP formulation (Kantorovich)

$K$: index set of the large rolls

$x_{ik}$: # of times that the i-th small roll is cut in the k-th roll

(how many times we cut a small roll of length i from the large roll k)

$y_k = \begin{cases} 1 & \text{if the } k \text{ roll is cut} \\ 0 & \text{otherwise} \end{cases}$

model

$$z^k_{ILP} = \min \sum_{k \in K} y_k$$

$$\text{st} \quad \underbrace{\sum_{k \in K} x_{ik}}_{\text{the total \# of small rolls of size } i} \geq b_i \quad \forall i \in I = \{1, \dots, m\} \quad \text{(demand constr)}$$

$$\sum_{i \in I} w_i x_{ik} \leq W \cdot y_k \quad \forall k \in K \quad \text{(width constr \& non lin)}$$

$$x_{ik} \in \mathbb{Z}^+ \quad \forall i \; \forall k$$

$$y_k \in \{0, 1\} \quad \forall k$$

Very weak formulation

Trivial LP relaxation bound:

$$z^k_{LP} = \sum_{k \in K} y_k \Big|_{relax} = \sum_{k \in K} \sum_{i \in I} \frac{w_i x_{ik}}{W} = \sum_{i \in I} \left[ \frac{w_i}{W} \sum_{k \in K} x_{ik} \right] = \sum_{i \in I} \frac{w_i b_i}{W}$$

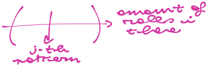
## Set covering ILP formulation (Gilmore and Gomory)

in order

Let $J = \{1, \ldots, n\}$ denote index set of the patterns,

eg $\begin{pmatrix} 2 \\ 4 \end{pmatrix} \begin{matrix} \to w_1 \\ \to w_2 \end{matrix}$ means that we cut 2 rolls of width $w_1$ and 4 of the $w_2$

– $\boxed{a_{ij}}$ the number of small rolls of width $w_i$ in $j$-th cutting pattern.




– $x_j$ the number of large rolls cut according to pattern $j$

Model

$$z_{ILP}^{GG} = \min \sum_{j \in J} x_j$$

$$\text{st} \quad \sum_{j \in J} a_{ij} x_j \geq b_i \quad \forall i \qquad \text{( demand constr )}$$

no explicit constr about the leftover (all of the cut due to the choice of the non-ole patterns according to $W, w_i$)

$$x_j \in \mathbb{Z}^+ \quad \forall j$$

Number $n$ of variables (patterns) grows exponentially with number $m$ of rows (types of small rolls).

but this formulation has a stronger LP relaxation


Observations:

(think about the structure of a basic set $\leq B$)

- at LP optimality at most $m$ of the $n$ variables have nonzero value; since $m \ll n$ only a very small subset of them (columns) is needed.
- for large integer $b_i$s, rounding optimal solutions of LP relaxation leads to satisfactory integer solutions,

# Column generation scheme

to solve efficiently a LP relaxation which involves an exp # of variables

Idea: no need to include all variables a priori, new variables are generated when needed.

like the dual approach of the cutting planes, where in general there
- we had an exp # of constr
- and we on the fly generate constr / cutting planes

Main steps:

1) consider LP relaxation of ILP, choose initial subset of variables $J_0 \subseteq J$, and set $k = 0$,

we solve the previous formulation of the problem, but using $J_k$ smaller than $J$

2) solve *LP Restricted Master problem* (LPRM) with subset $J_k$,

with the simplex method

3) solve *pricing subproblem* for LPRM with $J_k$ to search for an improving non basic variable $x_l$ (with negative reduced cost if min problem) and the associated column,

4) if $\exists$ such $x_l$, update $J_{k+1} := J_k \cup \{l\}$, set $k := k + 1$ and goto (2);
otherwise LPRM optimal solution is also optimal for LP relaxation of original ILP.

Observation: Column generation (CG) yields an optimal solution of LP relaxation and hence a **bound** on optimal ILP solution value.

## Example cont.: 1-D cutting stock problem

LP relaxation of Master problem (LPM):

not LPRM since for now it is not restricted, we have the full set $J$

$$
\begin{array}{rll}
z_{LPM} = \min & \sum_{j=1}^{n} x_j & y_i \\
s.t. & \sum_{j=1}^{n} a_{ij} x_j \geq b_i & \forall i \in I = \{1, \dots, m\} \\
& x_j \geq 0 & \forall j \in J = \{1, \dots, n\}.
\end{array}
$$

and its dual:

$$
\begin{array}{lll}
\max & \sum_{i \in I} y_i b_i & \\
\text{st} & \sum_{i \in I} a_{ij} y_i \leq 1 & \forall j \in J \\
& y_i \geq 0 & \forall i \in I
\end{array}
$$

When solving LPM with Simplex method:

- Since we have $\bar{c}_N^T = c_N^T - c_B^T B^{-1} N$, then the reduced cost of the $N$ (non basic) variable $x_j$ is $\boxed{\bar{c}_j = 1 - \sum_{i \in I} a_{ij} y_i}$ (remember the link dual constraints (at equality) and primal optimality)
  where $y^T = c^T B^{-1}$ is the complementary dual set
- The current basic sol is optimal iff $\bar{c}_j \geq 0 \quad \forall j \in J$

the method was:

Start with **LP Restricted Master problem** (LPRM) with $J_0 \subset J = \{1, \ldots, n\}$, guaranteeing a feasible solution.

LPRM with $J_0$:

$$
\begin{array}{rlll}
z_{LPRM} = \min & \sum_{j=1}^{n} x_j & & \rightsquigarrow x^* \\
s.t. & \sum_{j \in J_0} a_{ij} x_j \geq b_i & \forall i \in I = \{1, \ldots, m\} & \\
& x_j \geq 0 & \forall j \in J_0. &
\end{array}
$$

one difference wrt before

Reduced cost of non basic variable $x_j$ is still $\overline{c}_j = 1 - \sum_{i=1}^{m} a_{ij} y_i$.

Dual of LPRM with $J_0$:

$$
\begin{array}{rlll}
\max & \sum_{i=1}^{m} b_i y_i & & \rightsquigarrow y^* \\
s.t. & \sum_{i=1}^{m} a_{ij} y_i \leq 1 & \forall j \in J_0 & \\
& y_i \geq 0 & \forall i \in I = \{1, \ldots, m\}. &
\end{array}
$$

Let $\underline{x}^*$ and $\underline{y}^*$ be optimal solutions of LPRM and its dual, respectively.

Search for new improving non basic variables (columns/patterns)

Look for a non basic variable with smallest reduced cost and corresponding pattern $\underline{\alpha} \in \mathbb{Z}_+^m$ by solving the **pricing subproblem**:

we ask wif wt $\exists j \notin J_0$ (i.e. non basic) st $\bar{c}_j < 0$ (we would like to add to the basis)

$$\begin{aligned} \min \quad & \bar{c} = 1 - \sum_{i \in I} y_i^* \alpha_i \\ \text{st} \quad & \sum_{i \in I} w_i \alpha_i \leq W \quad \text{(pattern feasibility constraint)} \\ & \alpha_i \in \mathbb{Z}_+ \quad \forall i \in I = \{1, \dots, m\} \end{aligned} \tag{1}$$

Integer Knapsack problem that can be solved in $O(mW)$ using Dynamic Programming.

Two cases:

there are no non-basic variables with a negative reduced cost
$\Rightarrow$ we can't improve anymore the obj

- if $\bar{c}^* \geq 0$, the optimal solution of current LPRM is also optimal for LP relaxation,
- adding to current LPRM any non basic variable associated to a cutting pattern $\underline{\alpha} \in \mathbb{Z}_+^m$ with $\bar{c} < 0$, improves (decreases) the objective function value.

## Example cont.: 1-D cutting stock problem

small rolls sizes

demands for each of the small rolls

$$W = 3.9 \text{ m}, \quad \underline{w} = \begin{pmatrix} 1.25 \\ 1 \\ 0.8 \end{pmatrix} \text{ and } \underline{b} = \begin{pmatrix} 35 \\ 171 \\ 133 \end{pmatrix}.$$

Initial patterns: $A_1 = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$ waste of 0.05, $A_2 = \begin{pmatrix} 0 \\ 1 \\ 3 \end{pmatrix}$ waste of 0.5,

this choice is relevant:
- needs to guarantee a feasible set
- and quickly

$A_3 = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}$ waste of 0.6, $A_4 = \begin{pmatrix} 0 \\ 3 \\ 0 \end{pmatrix}$ waste of 0.9

From J. Lundgren, M. Rönnqvist, P. Värbrand, Optimization, Studentlitteratur AB, Lund, Sweden, 2010.

LP Restriced Master problem:

$$\begin{array}{ll} \min & z = \sum_{j=1}^{4} x_j \\ s.t. & \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} x_1 + \begin{pmatrix} 0 \\ 1 \\ 3 \end{pmatrix} x_2 + \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} x_3 + \begin{pmatrix} 0 \\ 3 \\ 0 \end{pmatrix} x_4 \geq \begin{pmatrix} 35 \\ 171 \\ 133 \end{pmatrix} \\ & x_j \geq 0 \qquad \forall j \in J_0 = \{1, 2, 3, 4\} \end{array}$$

Optimal solution of LPRM: $\underline{x}^* = (35, 21, 0, 38.33)^t$ with value $z^* = 94.33$

we then use as a LB on the opt val of the original ILP problem

Optimal dual solution: $\underline{y}^* = (\frac{2}{9}, \frac{1}{3}, \frac{2}{9})^t$

$\underline{y}^* = \underline{c}^T B^{-1}$

we need it to set up the pricing subproblem

Pricing subproblem:

$$\min \quad \bar{c} = 1 - \sum_{i \in I} y_i^* \alpha_i = 1 - \left(\frac{2}{9}\alpha_1 + \frac{1}{3}\alpha_2 + \frac{2}{9}\alpha_3\right)$$

$$\text{st} \quad \sum_{i \in I} w_i \alpha_i \leq W \iff 1{,}25\,\alpha_1 + 1\,\alpha_2 + 0{,}8\,\alpha_3 \leq 3{,}9$$

$\alpha_i$ integers

Optimal solution (integer knapsack): $\underline{\alpha}^* = (0, 3, 1)^t$ with value $\bar{c} = -\frac{2}{9}$. $\Rightarrow$ that pattern given by $\alpha^*$ is usable to include in the problem

Since $\bar{c} < 0$, adding new pattern $A_5 = (0, 3, 1)^t$ will improve (decrease) the objective function value.

Optimal solution of LPRM with $J_1 = \{1, 2, 3, 4, 5\}$: $\underline{x}^* = (35, 6.625, 0, 0, 43.125)^t$ with value $z^* = 84.75$.

Optimal dual solution: $\underline{y}^* = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4})^t$

and then we can repeat

Pricing subproblem:

$$\min \ \bar{c} = 1 - \sum_{i \in I} y_i^* \alpha_i = 1 - \left(\frac{1}{4}\alpha_1 + \frac{1}{4}\alpha_2 + \frac{1}{4}\alpha_3\right)$$

$$\text{st} \ \sum_{i \in I} w_i \alpha_i \leq W \iff 1{,}2\,\alpha_1 + 1\,\alpha_2 + 0{,}8\,\alpha_3 \leq 3{,}9$$

$\alpha_i$ integers

with optimal solution $\underline{\alpha}^* = (0, 3, 1)^t$ (as before!) and $\bar{c} = 0$. i.e. there is no way to further improve the obj (of the LP relaxation)

Thus $\underline{x}^* = (35, 6.625, 0, 0, 43.125)^t$ is an optimal sol. of LP relaxation of original ILP.

N.B.: in general many iterations!

ceil $\lceil \cdot \rceil$ as we have to satisfy a demand, i.e. demand we need) are $\geq$ to it

Rounding up: $\underline{x} = (35, 7, 0, 0, 44)^t$ with $z = 86$.

Since $z_{LPM} = 84.75$, lower bound is 85. since all coeffs and variables are integers

Optimal ILP solution: $\underline{x}_{ILP} = (36, 6, 0, 0, 43)^t$ with $z_{ILP} = 85$.

# General remarks

- Initial set of columns ($J_0$) has a strong impact: rich enough to guarantee initial feasible solution but not too large to reduce computational load.

for the pricing subproblem we want just to find quickly a good set (not nec optimal) so we can use heuristics

- Heuristics for pricing subproblem as long as an improving variable (column) is found. Exact method only to certify that LPRM solution is also optimal for LPM.

- CG methods can be viewed as cutting plane methods to solve the dual of LPM.

- Strong practical impact of CG due to great flexibility to model complicated restrictions.

- To find an optimal solution of original ILP, CG can be embedded in a Branch-and-Bound framework $\Rightarrow$ **Branch-and-Price method**.

**Computer Lab 3**: apply Column Generation to the airline crew pairing problem.

# 3.10 Lagrangian duality and relaxation

Generic ILP

$$\min \{ \underline{c}^t \underline{x} \; : \; \overbrace{A\underline{x} \geq \underline{b}}^{\text{easy constr}}, \; \underbrace{D\underline{x} \geq \underline{d}}_{\text{complex constr}}, \underline{x} \in \mathbb{Z}^n \}$$

with integer coefficients.

Suppose $D\underline{x} \geq \underline{d}$ are "complicating" constraints.

Idea: Delete $D\underline{x} \geq \underline{d}$ and, for each one of them, add to objective function a term with a multiplier $u_i$, which penalizes its violation.

More general setting:

$$\min \{ \underline{c}^t \underline{x} \; : \; D\underline{x} \geq \underline{d}, \; \underbrace{\underline{x} \in X \subseteq \mathbb{R}^n}_{\text{extension of the easy constr and domain (integrality eg) constr}} \} \qquad (1)$$

while the Lagrangianize constr need to be linear more problem, we just remove a constr and update the obj function

primal problem

**Definition**: Given

$$z^* = \min\{\underline{c}^t\underline{x} \;:\; D\underline{x} \geq \underline{d},\; \underline{x} \in X \subseteq \mathbb{R}^n\} \tag{2}$$

For each *multipliers* vector $\underline{u} \geq \underline{0}$, **Lagrangian subproblem** is

$$\boxed{w(\underline{u}) = \min\{\underline{c}^t\underline{x} + \underbrace{\underline{u}^t(\underline{d} - D\underline{x})}_{\sum \underbrace{u_i}_{\geq 0} (d_i - (Dx)_i)} \;:\; x \in X\}} \tag{3}$$

$\leq 0$ (we good for min prob) if constr satisfied

$\geq 0$ (we bad for min prob) if constr is violated

where

$L(\underline{x}, \underline{u}) = \underline{c}^t\underline{x} + \underline{u}^t(\underline{d} - D\underline{x})$ *Lagrangian function* of primal (2),

$w(\underline{u}) = \min\{L(\underline{x}, \underline{u}) \;:\; x \in X\}$ *dual function*.

**Proposition**: For any $\underline{u} \geq \underline{0}$, the Lagrangian subproblem (3) is a relaxation of (2).

Proof: Clearly $\{\underline{x} \in X : D\underline{x} \geq \underline{d}\} \subseteq X$

when $\forall \underline{u} \geq 0$ and $\underline{x} \in X$ (we feasible for (2)) we have that $w(\underline{u}) \leq \underline{c}^T\underline{x}$, since

$w(\underline{u}) \leq \underbrace{\underline{c}^T\underline{x}}_{\geq 0} + \underbrace{\underline{u}^T(\underline{d} - D\underline{x})}_{\leq 0} \leq \underline{c}^T\underline{x} \quad \forall \underline{x} \in X$, so it is a relaxation

**Corollary**: If $z^* = \min\{\underline{c}^t\underline{x} \;:\; D\underline{x} \geq \underline{d},\; \underline{x} \in X\}$ is finite, then $\quad w(\underline{u}) \leq z^* \quad \forall \underline{u} \geq \underline{0}$.

we with $w(\underline{u})$ we get a lower (better) value than the opt one $z^*$, but we expect (3) is a relaxation

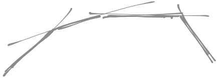
$w(\underline{u})$ — $z^*$ — $\underline{c}^T\underline{x}$ for $\underline{x} \in X$ we feasible for (2)

To determine tightest lower bound

**Definition**: **Lagrangian dual** of primal problem (2) is

$$w^* = \max\ w(\underline{u}) \quad \text{st } \underline{u} \geq \underline{0} \qquad (4)$$

we always get a piece-wise linear concave function since we take the lower envelop of the various functions of $\underline{u}$



Note: Relaxing linear constraints, $L(., \underline{u})$ is linear. Subproblem (3) must be "sufficiently easy". to make this approach interesting

For LPs Lagrangian dual coincides with LP dual.

**Corollary**: (Weak Duality)

For every pair of feasible solutions $\underline{x} \in \{\underline{x} \in X \,:\, D\underline{x} \geq \underline{d}\}$ of primal (2) and $\underline{u} \geq \underline{0}$ of Lagrangian dual (4), we have

$$\boxed{w(\underline{u}) \leq \underline{c}^t \underline{x}}$$

why is this useful? consider as feasible vectors $\bar{x}$ and $\bar{u}$. then

if we have equality, ie $w(\bar{u}) = c^T \bar{x}$ $\Rightarrow$ then both are optimal resp. for their problems

Consequences:

*i*) If $\underline{\tilde{x}}$ feasible for primal (2), $\underline{\tilde{u}}$ feasible for Lagrangian dual (4) and $\overbrace{\underline{c}^t\underline{\tilde{x}}}^{\tilde{w}} = \overbrace{w(\underline{\tilde{u}})}^{\tilde{z}}$, then $\underline{\tilde{x}}$ and $\underline{\tilde{u}}$ optimal for respectively (2) and (4).

*ii*) In particular $w^* = \max_{\underline{u} \geq \underline{0}} w(\underline{u}) \leq z^* = \min \{\underline{c}^t\underline{x} : D\underline{x} \geq \underline{d}, \ \underline{x} \in X\}$.
If one problem is unbounded, the other one is infeasible.

Recall: For any primal-dual pair of bounded LPs, we have strong duality ($w^* = z^*$).

**Observation**: In discrete optimization we can have a *duality gap*, i.e., $w^* < z^*$.

what happens if we want to ~~[illegible]~~ equality constraints?

ILP with equality constraints:

we now get free multipliers, i.e. $\underline{u} \in \mathbb{R}^m$, $u_i \gtreqless 0$

Lagrangian dual is

$$\max_{\underline{u} \in \mathbb{R}^m} w(\underline{u})$$

as we split $D\underline{x} = \underline{d}$ into
$D\underline{x} \geq \underline{d} \rightarrow \underline{u}^-$
$-D\underline{x} \geq -\underline{d} \rightarrow \underline{u}^+$

$$\underbrace{\underline{u}}_{\text{free}} = \underbrace{\underline{u}^+}_{\geq 0} - \underbrace{\underline{u}^-}_{\geq 0}$$

**Example: Uncapacitated Facility Location** (UFL)

Variant with profits $p_{ij}$, fixed costs $f_j$ for opening the depots in the candidate sites, and total profit to be maximized.

MILP formulation:

primal problem

$$z^* = \max \quad \sum_{i\in M}\sum_{j\in N} p_{ij}x_{ij} - \sum_{j\in N} f_j y_j$$

$$\begin{aligned}
s.t. \quad & \sum_{j\in N} x_{ij} = 1 & \forall i\in M \quad & (5)\\
& x_{ij} \le y_j & \forall i\in M, j\in N &\\
& y_j \in \{0,1\} & \forall j\in N &\\
& \underbrace{0 \le x_{ij} \le 1}_{\text{fraction of the demand satisfied}} & \forall i\in M, j\in N &
\end{aligned}$$

clients

depots

linking constr => these are the complicating ones

while these are in a sense (wrt i) we are easier

Relaxing constraints (5), Lagrangian subproblem:

$$w(u) = \max \sum_i\sum_j p_{ij}x_{ij} + \sum_j f_j y_j + \left[\sum_{i\in M} u_i\left(1 - \sum_{j\in N} x_{ij}\right)\right]$$

$$= \max \sum_i\sum_j (p_{ij} - u_i)x_{ij} + \sum_j f_j y_j + \cancel{\sum_{i\in M} u_i} \quad (6)$$

$(7)$

$$\text{st} \quad x_{ij} \le y_j \quad \forall i\in N \ \forall j\in M \quad (8)$$

$$y_j \in \{0,1\} \quad \forall j$$

$$0 \le x_{ij} \le 1 \quad \forall i \ \forall j$$

so we can update the objective function

this problem gets decomposed into $|N|$ subproblems, one for each candidate $j\in N$

Indeed $w(\underline{u}) = \sum_{j \in N} w_j(\underline{u}) + \sum_{i \in M} u_i$ where

$$
\begin{aligned}
w_j(\underline{u}) = \max \quad & \textstyle\sum_{i \in M} (p_{ij} - u_i) x_{ij} - f_j y_j & & \\
s.t. \quad & x_{ij} \leq y_j & & \forall i \in M \\
& y_j \in \{0, 1\} & & \\
& 0 \leq x_{ij} \leq 1 & & \forall i \in M
\end{aligned}
\tag{9}
$$

For each $j \in N$, the subproblem (9) can be solved by inspection:

- if $y_j = 0$ then $x_{ij} = 0 \ \forall i$ and $w_j$ (obj function value) is 0
- if $y_j = 1$ then we set $x_{ij} = 1$ for the clients $i$ for which the profit $(p_{ij} - u_i)$ is $> 0$

as we have a max problem we want to add positive stuff in the obj function

and now the obj function value is

$$w_j = \sum_i \max(p_{ij} - u_i, 0) - f_j$$

and the Lagrangian dual is $\max w_j(\underline{u})$ st $\underline{u} \in \mathbb{R}^m$

we unconstrained

See Chapter 10 of L. Wolsey, Integer Programming, p. 169-170

come mai? perché max e la altra problema originale

## Properties of Lagrangian subproblem and dual function

**Proposition**: If $\underline{u} \geq \underline{0}$ and

i) $\underline{x}(\underline{u})$ is an optimal solution of Lagrangian subproblem (3)

ii) $D\underline{x}(\underline{u}) \geq \underline{d}$ we (we are lucky and) that $\underline{x}(\underline{u})$ is feasible for the primal

iii) $(D\underline{x}(\underline{u}))_i = d_i$ for each $u_i > 0$ (complementary slackness conditions),

then $\underline{x}(\underline{u})$ is also optimal for primal (2).

otherwise the constr at equality with strictly positive mult

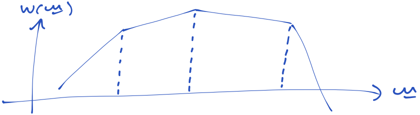
Proof:

⇒ in general the Lagrangian dual problem is easier to solve

well, in this setting of min/min/max ie when the dual is a max problem if dual is a min then $w(\underline{u})$ is convex

**Proposition**: Dual function $w(\underline{u})$ is concave. and it will be even piece-wise linear (it is not everywhere differentiable)

Illustration:



# 3.10.1 Strength and choice of the Lagrangian dual

Characterization in terms of an LP.

this allows us to see how much tight (we error) we get by dualizing

**Theorem**: Generic ILP

with linear eq.) constraints

$$\min\{\underline{c}^t\underline{x} \,:\, A\underline{x} \geq \underline{b},\ D\underline{x} \geq \underline{d},\ \underline{x} \in \mathbb{Z}^n\}$$

with integer coefficients.

Let $w(\underline{u}) = \min\{\underline{c}^t\underline{x} + \underline{u}^t(\underline{d} - D\underline{x}) \,:\, A\underline{x} \geq \underline{b},\ \underline{x} \in \mathbb{Z}^n\}$,

$w^* = \max_{\underline{u} \geq \underline{0}} w(\underline{u})$ and $X = \{\underline{x} \in \mathbb{Z}^n \,:\, A\underline{x} \geq \underline{b}\}$,

feasible region considering just
- easy constr
- integrality restriction

then

the Lag. dual val (optimal)

$$\boxed{w^* = \min\{\underline{c}^t\underline{x} \,:\, D\underline{x} \geq \underline{d},\ \underline{x} \in conv(X)\}.}$$

original obj funct

complex constr

convexification of $X$ (convex hull of the integer set of the remaining constraints)

"Convexification" of $X$.

**Corollary 1**: Since $conv(X) \subseteq \{\underline{x} \in \mathbb{R}^n \,:\, A\underline{x} \geq \underline{b}\}$,

$$\boxed{z_{LP}} = \min\{\underline{c}^t\underline{x} \,:\, A\underline{x} \geq \underline{b}, D\underline{x} \geq \underline{d}, \underline{x} \in \mathbb{R}^n\} \boxed{\leq w^* \leq z^*}$$

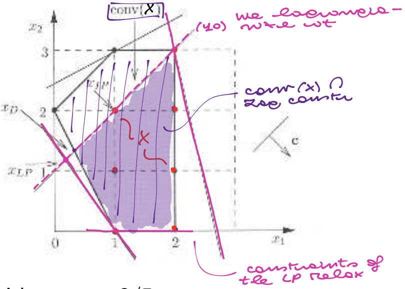
We may have $z_{LP} < w^* < z^*$.

since we are in discrete and we may have a duality gap

- remove the constr (Lag. set)
- consider the conv(x) obtained with the removal
- then consider that conv(x) found and ∩ set with the Lag. constr

Illustration: D. Bertsimas, R. Weismantel, Optimization over integers, Dynamic Ideas, 2005

$$\begin{array}{llll} \min & 3x_1 - x_2 & & \\ s.t. & x_1 - x_2 & \geq -1 & (10) \\ & -x_1 + 2x_2 & \leq 5 & (11) \\ & 3x_1 + 2x_2 & \geq 3 & (12) \\ & 6x_1 + x_2 & \leq 15 & (13) \\ & x_1, x_2 & \geq 0 \ \text{integer} & \end{array}$$




$\underline{x}_{ILP} = (1,2)^t$ with $z_{ILP} = 1$ and $\underline{x}_{LP} = (1/5, 6/5)^t$ with $z_{LP} = -3/5$.

- Dualize (10): For every $u \geq 0$, $w(u) = \min_{(x_1,x_2) \in X} 3x_1 - x_2 + u(-1 - x_1 + x_2)$
  where $X$ is the set of all integer solutions of (11)-(13).

- Find optimal solution $u^*$ of Lagrangian dual: $w^* = \max_{u \geq 0} w(u)$ and optimal solution $\underline{x}_D = \underline{x}(u^*)$.

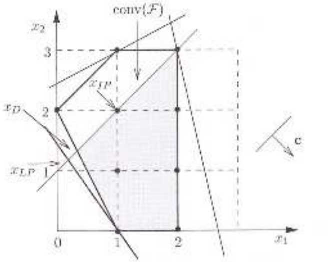
  Represent $conv(X) \cap \{(x_1, x_2) \in \mathbb{R}^2 : x_1 - x_2 \geq -1\}$ (in grey).

  Obtain $\underline{x}_D = (1/3, 4/3)$ with $w^* = -1/3$.

Thus $z_{LP} = -3/5 < w^* = -1/3 < z_{ILP} = 1$

Illustration: D. Bertsimas, R. Weismantel, Optimization over integers, Dynamic Ideas, 2005

$$
\begin{array}{llll}
\min & 3x_1 - x_2 & & \\
s.t. & x_1 - x_2 & \geq -1 & (10) \\
 & -x_1 + 2x_2 & \leq 5 & (11) \\
 & 3x_1 + 2x_2 & \geq 3 & (12) \\
 & 6x_1 + x_2 & \leq 15 & (13) \\
 & x_1, x_2 & \geq 0 \text{ integer} &
\end{array}
$$




$\underline{x}_{ILP} = (1,2)^t$ with $z_{ILP} = 1$ and $\underline{x}_{LP} = (1/5, 6/5)^t$ with $z_{LP} = -3/5$.

- Dualize (10): For every $u \geq 0$, $w(u) = \min_{(x_1,x_2)\in X} 3x_1 - x_2 + u(-1 - x_1 + x_2)$
  where $X$ is the set of all integer solutions of (11)-(13).

- Find optimal solution $u^*$ of Lagrangian dual: $w^* = \max_{u \geq 0} w(u)$ and optimal solution $\underline{x}_D = \underline{x}(u^*)$.

  Represent $conv(X) \cap \{(x_1,x_2) \in \mathbb{R}^2 \, : \, x_1 - x_2 \geq -1\}$ (in grey).

  Obtain $\underline{x}_D = (1/3, 4/3)$ with $w^* = -1/3$.

Thus $z_{LP} = -3/5 < w^* = -1/3 < z_{ILP} = 1$

Drawing $w(u)$ we can verify that $u^* = 5/3$ with $w^* = -1/3$.

(2) Lag dual    max $w(u)$
st $u \geq 0$

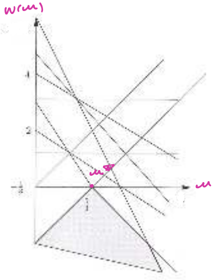


Illustration $w(\underline{u})$:

In some cases Lagrangian relaxation is as weak as LP relaxation.

**Corollary 2**: If $X = \{\underline{x} \in \mathbb{Z}^n \,:\, A\underline{x} \geq \underline{b}\}$ and $conv(X) = \{\underline{x} \in \mathbb{R}^n \,:\, A\underline{x} \geq \underline{b}\}$, then

$$\boxed{w^*} = \max_{\underline{u} \geq \underline{0}} w(\underline{u}) \boxed{= z_{LP}} = \min\{\underline{c}^t\underline{x} \,:\, A\underline{x} \geq \underline{b}, D\underline{x} \geq \underline{d}, \underline{x} \in \mathbb{R}^n\}.$$

but if we have this condition we can even just solve the LP relaxation then

Example: Binary knapsack problem

$$\begin{array}{lrl} \max & z = & \sum_{j=1}^n p_j x_j \\ \text{s.t.} & & \sum_{j=1}^n a_j x_j \leq b \\ & & x_j \in \{0,1\} \qquad \forall j \end{array}$$

and its LP relaxation

allowing fractional choices

$$z_{LP-KP} = \max_{\underline{x} \in [0,1]^n} \{\sum_{j=1}^n p_j x_j \,:\, \sum_{j=1}^n a_j x_j \leq b\}.$$

$X = \{\underline{x} \in \{0,1\}^n\}$ and $conv(X) = \{\underline{x} \in [0,1]^n\}$, and $0 \leq x_j \leq 1$ are already contained in LP relaxation.

all the points inside that n-hypercube

Corollary 2 implies: $w^* = z_{LP-KP}$.

## Choice of the Lagrangian dual

Which constraints to relax to get tighter bounds?

Choice **criteria**:

i) strength of the bound $w^*$ obtained by solving Lagrangian dual,

ii) difficulty of solving Lagrangian subproblems    as we are talking of a relaxation

$$w(\underline{u}) = \min \{\underline{c}^t \underline{x} + \underline{u}^t(\underline{d} - D\underline{x}) \, : \, x \in X \subseteq \mathbb{R}^n\},$$

iii) difficulty of solving Lagrangian dual:  $w^* = \max_{\underline{u} \geq \underline{0}} w(\underline{u})$.

For $(i)$ we have the LP characterization,

(ii) depends on the specific problem,

(iii) depends, among others, on the number of dual variables.

Look for a reasonable trade-off.

- if we relax too simple constr
  ⇒ we get too difficult sub-problems
- if we relax too complex constr
  ⇒ we get too difficult Lag duals

See **exercise 5.3** on the generalized assignment problem.

## 3.10.2 Solution of the Lagrangian duals

context here: min $f(x)$ s.t. $x \in X$

where $f$ convex, while before $w(u)$ was concave, but is equivalent

Generalization of the gradient method for $\mathcal{C}^1$ functions to convex piecewise $\mathcal{C}^1$ ones (not everywhere differentiable).

**Definition**: Let $C \subseteq \mathbb{R}^n$ and $f : C \to \mathbb{R}$ be convex.

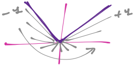
- $\underline{\gamma} \in \mathbb{R}^n$ is a *subgradient* of $f$ at $\overline{\underline{x}} \in C$ if

$\underline{\gamma} = \nabla f(\overline{\underline{x}})$ if $f$ is differentiable at $\overline{\underline{x}}$

$$f(\underline{x}) \geq f(\overline{\underline{x}}) + \underline{\gamma}^t(\underline{x} - \overline{\underline{x}}) \qquad \forall \underline{x} \in C$$

- the *subdifferential*, denoted by $\partial f(\underline{x})$, is the set of all subgradients of $f$ at $\underline{x}$.

Example: For $f(x) = |x|$, $\gamma = 1$ if $\overline{x} > 0$, $\gamma = -1$ if $\overline{x} < 0$, and $\partial f(\overline{x}) = [-1, 1]$ if $\overline{x} = 0$

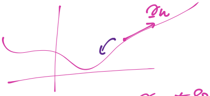



**Properties**:

A convex $f : C \to \mathbb{R}$ has at least one subgradient at each interior point $\overline{\underline{x}}$ of $C$.

$\underline{x}^*$ is a global minimum of $f$ if and only if $\underline{0} \in \partial f(\underline{x}^*)$.

## Subgradient method




Given $\min_{x \in \mathbb{R}^n} f(\underline{x})$ with $f(\underline{x})$ convex.

Start from an abitrary $\underline{x}_0$.

At $k$-th iteration: consider $\underline{\gamma}_k \in \partial f(\underline{x}_k)$ and set

$$\boxed{\underline{x}_{k+1} := \underline{x}_k - \alpha_k \, \underline{\gamma}_k}$$

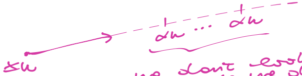
$\gamma_k$ the gradient describe the max increase direction, so we use $-\alpha_k$ to move in the opposite direction

well it is true in the classical case!! $C^1$ domain functions

gradient descent model

with $\alpha_k > 0$

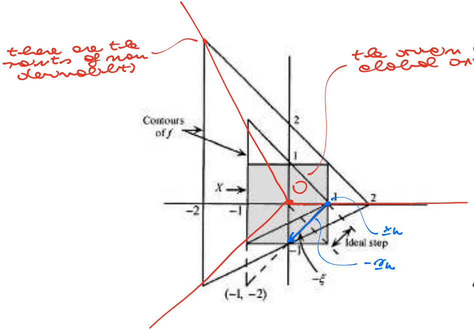
**Observation**: No 1-D search (optimization) because for nondifferentiable functions a subgradient $\underline{\gamma} \in \partial f(\underline{x})$ is not necessarily a descent direction!




we don't look for the "optimal" $\alpha_k$ since we don't have that $-\gamma_k$ points towards the descent direction, if we have not even differentiable functions like here

Example: $\min_{-1\le x_1,x_2\le 1}\ f(x_1,x_2)$ with $f(x_1,x_2)=\max\{-x_1, x_1+x_2, x_1-2x_2\}$

Level curves in black, points of nondifferentiability $(t,0)$, $(-t,2t)$ and $(-t,-t)$ for $t\ge 0$, global minimum $\underline{x}^*=(0,0)$.




Moral: keep the jump size $\alpha_k$ small

- let $x_k=\binom{1}{0}$ and consider $\xi_k=\binom{1}{1}$ we see that such a step is a worsening direction, but there still a good direction, that brings us closer to the real optimum (0,0)
- in the direction $\{x\in\mathbb{R}^2 : x = x_k-\alpha_k\xi_k,\ \alpha_k\ge 0\}$ is worsening, but if $\alpha_k$ is small enough we still get an improvement

From Chapter 8, Bazaraa et al., Nonlinear Programming, Wiley, 2006, p. 436-437 convergence guarantee

**Theorem**:

the $\alpha_k$ goes to 0 — but not too fast

If $f$ is convex, $\lim_{\|\underline{x}\|\to\infty} f(\underline{x}) = +\infty$, $\lim_{k\to\infty} \alpha_k = 0$ and $\sum_{k=0}^{\infty} \alpha_k = \infty$, the subgradient method terminates after a finite number of iterations with an optimal solution $\underline{x}^*$ or infinite sequence $\{\underline{x}_k\}$ admits a subsequence converging to $\underline{x}^*$.

Stepsize:

In practice $\{\alpha_k\}$ as above (e.g., $\alpha_k = 1/k$) are too slow.

An option: $\alpha_k = \alpha_0 \rho^k$ for a given $\rho < 1$. A more popular one (min problems):

$$\alpha_k = \varepsilon_k \frac{f(\underline{x}_k) - \hat{f}}{\|\underline{\gamma}_k\|^2},$$

where $0 < \varepsilon_k < 2$ and $\hat{f}$ is either the optimal value $f(\underline{x}^*)$ or an estimate.

eg in some problems we could know the optimal value but not how to reach it (ie the set)

Stopping criterion: prescribed maximum number of iterations

(even if $\underline{0} \in \partial f(\underline{x}_k)$ it may non be considered at $\underline{x}_k$).

Need to store the best solution $\underline{x}_k$ found.

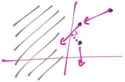
or also if we stall in the improvement of the obj function

Simple extension for bounds (projections).

## Subgradient method for Lagrangian dual

how do we deal with non-negativity constr in the subgradient method? just use a projection if needed

$$\max_{\underline{u} \geq \underline{0}} w(\underline{u})$$



where $w(\underline{u}) = \min \{\underline{c}^t \underline{x} + \underline{u}^t(\underline{d} - D\underline{x}) \ : \ x \in X \subseteq \mathbb{R}^n\}$ is concave and piecewise linear.

is it difficult to find subgradients for it?
No: there is a

Simple characterization of the subgradients of $w(\underline{u})$:

- find an optimal set of the LAG subproblem
- the vector of the constr violation, $\underline{d} - D\underline{x}_{\underline{\tilde{u}}}$, is a subgradient of $w(\underline{u})$

**Proposition**:

Consider $\underline{\tilde{u}} \geq \underline{0}$ and $X(\underline{\tilde{u}}) = \{\underline{x} \in X \ : \ w(\underline{\tilde{u}}) = \underline{c}^t \underline{x} + \underline{\tilde{u}}^t(\underline{d} - D\underline{x})\}$ set of optimal solutions of Lagrangian subproblem (3). Then

- For each $\underline{x}(\underline{\tilde{u}}) \in X(\underline{\tilde{u}})$, the vector $\boxed{(\underline{d} - D\underline{x}(\underline{\tilde{u}})) \in \partial w(\underline{\tilde{u}})}$.
- Each subgradient of $w(\underline{u})$ at $\underline{\tilde{u}}$ can be expressed as a convex combination of subgradients $(\underline{d} - D\underline{x}(\underline{\tilde{u}}))$ with $\underline{x}(\underline{\tilde{u}}) \in X(\underline{\tilde{u}})$.

and these subgradients are the ones define the subdifferential of $w(\underline{u})$

Procedure:

1) Select initial $\underline{u}_0$ and set $k := 0$.

2) Solve Lagrangian subproblem

$$w(\underline{u}_k) = \min\{\underline{c}^t \underline{x} + \underline{u}_k^t(\underline{d} - D\underline{x}) \,:\, x \in X\}.$$

If $\underline{x}(\underline{u}_k)$ optimal solution found, $(\underline{d} - D\underline{x}(\underline{u}_k))$ is a subgradient of $w(\underline{u})$ at $\underline{u}_k$.

3) Update Lagrange multipliers:

now we use (+) as we are maximizing, so we want to follow the (sub) gradient

$$\underline{u}_{k+1} = \max\{\underline{0}, \underline{u}_k \underline{+ \alpha_k}\,(\underline{d} - D\underline{x}(\underline{u}_k))\}$$

with, for instance, $\alpha_k = \varepsilon_k \frac{\hat{w} - w(\underline{u}_k)}{\|\underline{d} - D\underline{x}(\underline{u}_k)\|^2}$, where $\hat{w}$ is an estimate of optimal value $w^*$.

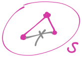
4) Set $k := k + 1$

# 3.10.3 Lagrangian relaxation for the STSP (Held & Karp)

Symmetric TSP: Given undirected $G = (V, E)$ with cost $c_e \in \mathbb{Z}^+$ for each $e \in E$, determine a Hamiltonian cycle of minimum total cost.

for the 2 see notes on them

$$
\begin{array}{llll}
\min & \sum_{e \in E} c_e x_e & & \\
s.t. & \sum_{e \in \delta(i)} x_e = 2 & \forall i \in V & (14) \\
& \sum_{e \in E(S)} x_e \leq |S| - 1 & \forall S \subseteq V, 2 \leq |S| \leq n-1 & (15) \\
& x_e \in \{0, 1\} & \forall e \in E &
\end{array}
$$


where $E(S) = \{\{i, j\} \in E \,:\, i \in S, j \in S\}$

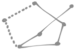
Observations:

the ones relevant to the sets $S^c$

*i*) Due to (14), half of the (15) are redundant:
$\sum_{e \in E(S)} x_e \leq |S| - 1$ if and only if $\sum_{e \in E(\overline{S})} x_e \leq |\overline{S}| - 1$, where $\overline{S} = V \setminus S$.
Thus all (15) with $1 \in S$ can be deleted.

*ii*) Summing over all (14) and dividing by 2, we obtain $\sum_{e \in E} x_e = n$ that can be added.

Recall: a Hamiltonian cycle is a 1-tree (i.e., a spanning tree on nodes $\{2, \ldots, n\}$ plus two edges incident to node 1) in which all nodes have exactly two incident edges.



≥0 (we will worsen the obj funct) if we select fewer than 2 incident edges

Since

$$\sum_{e \in E} c_e x_e + \sum_{i \in V} \underbrace{u_i}_{\geq 0}(2 - \sum_{e \in \delta(i)} x_e) = \sum_{e \in E}\Big[c_e x_e + \sum_{i \in V}(-u_i - u_j)x_e\Big] + \sum_{i \in V} 2u_i ,$$

relaxing the degree constraints (14) for all nodes except node 1,

each of the xe occurs twice since the edge e in {i,j} so write the outside sum over V will get ui and uj

Lagrangian subproblem:

$$
\begin{array}{llll}
w(\underline{u}) = \min & \sum_{e \in E}(c_e - u_i - u_j)x_e + 2\sum_{i \in V} u_i & \\
\quad s.t. & \sum_{e \in \delta(1)} x_e = 2 & \\
& \sum_{e \in E(S)} x_e \leq |S| - 1 & \forall S \subseteq V, 2 \leq |S| \leq n-1, 1 \notin S \\
& \sum_{e \in E} x_e = n & \\
& x_e \in \{0,1\} & \forall e \in E
\end{array}
$$

where $u_1 = 0$ and $E(S) = \{\{i,j\} \in E : i \in S, j \in S\}$.

and the problem becomes equivalent to
- finding a min cost 1-tree
- with a modified obj function

Note: Set of feasible solutions $\equiv$ set of all 1-trees.

Lagrangian dual: $\max_{\underline{u} \in \mathbb{R}^{|V|} \,:\, u_1 = 0} \; w(\underline{u})$

⇒ we can solve it at optimality with a greedy alg

# Example from L. Wolsey, Integer Programming, p. 175-177

Undirected $G = (V, E)$ with 5 nodes and cost matrix:

$c_{ij}$

$$\begin{pmatrix} - & 30 & 26 & 50 & 40 \\ - & - & 24 & 40 & 50 \\ - & - & - & 24 & 26 \\ - & - & - & - & 30 \\ - & - & - & - & - \end{pmatrix}$$

Dual function:

$$w(\underline{u}^k) = \min \left\{ \sum_{e=\{i,j\}\in E} (c_e - u_i^k - u_j^k)x_e^k + 2\sum_{i\in V} u_i^k : \underline{x}^k \text{ incidence vector of a 1-tree} \right\}$$

Notation: $c_{ij}^k = c_e - u_i^k - u_j^k$ for $e = \{i,j\} \in E$

i-th vector component

Subgradient $\underline{\gamma}^k$ with $\boxed{\gamma_i^k = (2 - \sum_{e\in\delta(i)} x_e^k)}$, where $\underline{x}^k = \underline{x}(\underline{u}^k)$ is an optimal solution of Lagrangian subproblem at $k$-th iteration.

Since $\sum_{e\in\delta(1)} x_e = 2$ is not relaxed, $\gamma_1^k = 0$ for all $k$.
Starting from $u_1^0 = 0$ we then have $u_1^k = 0$ for all $k \geq 1$.

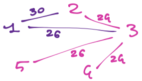
Feasible solution of cost 148 found with primal heuristic:

$$x_{12} = x_{23} = x_{34} = x_{45} = x_{51} = 1 \text{ and } x_{ij} = 0 \text{ for all other } \{i,j\} \in E$$

Solution of Lagrangian dual starting from $\underline{u}^0 = \underline{0}$ with $\varepsilon = 1$:

Solving Lagrangian subproblem with costs:

$$c^0 = c = \begin{pmatrix} - & 30 & 26 & 50 & 40 \\ - & - & 24 & 40 & 50 \\ - & - & - & 24 & 26 \\ - & - & - & - & 30 \\ - & - & - & - & - \end{pmatrix}$$




we apply eg. kruskal alg on that subgraph
- we get the min cost span tree
- and then we reconsider node 1

($c_e^0 = c_e$ for each $e \in E$ since $\underline{u}^0 = \underline{0}$),

we find $\underline{x}(\underline{u}^0)$ corresponding to 1-tree of cost 130:

$$x_{12} = x_{13} = x_{23} = x_{34} = x_{35} = 1 \text{ and } x_{ij} = 0 \text{ for all other } \{i,j\} \in E$$

and the subgradient is

$\gamma_k = [\gamma_i^k = 2 - (\#\text{incident edges on node } i)]$

$$\Rightarrow \quad \underline{\gamma}_0 = \begin{pmatrix} 2-2 \\ 2-2 \\ 2-4 \\ 2-1 \\ 2-1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -2 \\ 1 \\ 1 \end{pmatrix}$$

Knowing $\underline{x}(\underline{u}^0)$, we can compute $w(\underline{u}^0) = 130 + 0$ (cost of 1-tree + $2\sum_{i \in V} u_i^0$).

Subgradient

$$\underline{\gamma}^0 = \begin{pmatrix} 0 \\ 0 \\ -2 \\ 1 \\ 1 \end{pmatrix}$$

Update Lagrange multipliers:

$$\underline{u}^1 = \underline{u}^0 + \frac{(\hat{w} - w(u^0))}{\|\underline{\gamma}_0\|^2}\underline{\gamma}^0 = \underline{0} + \frac{(148-130)}{6}\begin{pmatrix} 0 \\ 0 \\ -2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -6 \\ 3 \\ 3 \end{pmatrix}$$

Since

$$C^0 = \begin{pmatrix} - & 30 & 26 & 50 & 40 \\ - & - & 24 & 40 & 50 \\ - & - & - & 24 & 26 \\ - & - & - & - & 30 \\ - & - & - & - & - \end{pmatrix}$$

we have

$c_{ij} \mapsto c_{ij} - u_i^k - u_j^k$

$$C^1 = \begin{pmatrix} - & 30 & 32 & 47 & 37 \\ - & - & 30 & 37 & 47 \\ - & - & - & 27 & 29 \\ - & - & - & - & 24 \\ - & - & - & - & - \end{pmatrix}$$

As optimal solution $\underline{x}(\underline{u}^1)$ of Lagrangian subproblem with matrix $C^1$ we find 1-tree of cost 143:

$$x_{12} = x_{13} = x_{23} = x_{34} = x_{45} = 1 \text{ and } x_{ij} = 0 \text{ for all other } \{i, j\} \in E$$

and $w(\underline{u}^1) = 143 + 2\sum_{i \in V} u_i^1 = 143$.

Since

$$\underline{\gamma}^1 = \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 1 \end{pmatrix},$$

we have

$$\underline{u}^2 = \underline{u}^1 + \frac{(\hat{w} - w(u^1))}{\|\underline{\gamma}_1\|^2}\underline{\gamma}^1 = \begin{pmatrix} 0 \\ 0 \\ -6 \\ 3 \\ 3 \end{pmatrix} + \frac{(148 - 143)}{2}\begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{-17}{2} \\ 3 \\ \frac{11}{2} \end{pmatrix}$$

Therefore

$$c^2 = \begin{pmatrix} - & 30 & 34.5 & 47 & 34.5 \\ - & - & 32.5 & 37 & 44.5 \\ - & - & - & 29.5 & 29 \\ - & - & - & - & 21.5 \\ - & - & - & - & - \end{pmatrix}$$

and we obtain $\underline{x}(\underline{u}^2)$ that corresponds to 1-tree of cost 147.5:

$$x_{12} = x_{15} = x_{23} = x_{35} = x_{45} = 1 \text{ and } x_{ij} = 0 \text{ for all other } \{i, j\} \in E$$

and $w(\underline{u}^2) = 147.5 + 0$.

Since all costs $c_e$ are integer, the feasible solution of cost 148 found by the heuristic is optimal!

we can say that since the Lag relaxation is a relaxation, it indeed it just provides us lower bounds.

and here we get
- feasible sol value: 148
- LB of Lag relaxation: 147,5

⇒ there is no room for improvement, so that 148 was optimal