Course:

Optimization I
Introduction to Linear Optimization
ISyE 6661 A Fall 2024

Instructor: **Dr. Arkadi Nemirovski**

nemirovs@isye.gatech.edu 404-385-0769

Office hours: virtual (zoom) Tuesday 10:00-11:55 am and by appointment

Groseclose 446

Teaching Assistant: TBA

Office location: TBA

Office hours: TBA

Classes: Monday & Wednesday 11:00-12:15 IC 109

Lecture Notes, Transparencies, Assignments:

Course website and

https://www2.isye.gatech.edu/~nemirovs/OPTITR2024.pdf

https://www2.isye.gatech.edu/~nemirovs/OPTILN2024.pdf

Grading Policy:

Assignments	5%
Midterm exam	30%
Final exam	65%

♣ To make decisions optimally is one of the most basic desires of a human being.

Whenever the candidate decisions, design restrictions and design goals can be properly quantified, optimal decision-making reduces to solving an *optimization problem*, most typically, a *Mathematical Programming* one:

$$
\begin{array}{lll}
\text{minimize} & & \\
 & f(x) & [\text{ objective }] \\
\text{subject to} & & \\
h_i(x) & = 0,\ i = 1,...,m & \left[\begin{array}{l}\text{equality}\\ \text{constraints}\end{array}\right] \\
g_j(x) & \leq 0,\ j = 1,...,k & \left[\begin{array}{l}\text{inequality}\\ \text{constraints}\end{array}\right] \\
x & \in X & [\text{ domain }]
\end{array}
\qquad \text{(MP)}
$$

♣ In (MP),

♢ a *solution* $x \in \mathbf{R}^n$ represents a candidate decision,

♢ the *constraints* express restrictions on the meaningful decisions (balance and state equations, bounds on resources, etc.),

♢ the *objective* to be minimized represents the losses (minus profit) associated with a decision.

$$
\begin{array}{lll}
\text{minimize} & & \\
& f(x) & [\text{ objective }] \\
\text{subject to} & & \\
h_i(x) & = 0,\ i = 1,...,m & \left[\begin{array}{l}\text{equality}\\ \text{constraints}\end{array}\right] \\
g_j(x) & \leq 0,\ j = 1,...,k & \left[\begin{array}{l}\text{inequality}\\ \text{constraints}\end{array}\right] \\
x & \in X & [\text{ domain }]
\end{array}
\qquad \text{(MP)}
$$

♣ To solve problem (MP) means to find its *optimal solution* x_*, that is, a *feasible* (i.e., satisfying the constraints) solution with the value of the objective $\leq$ its value at any other feasible solution:

$$
x_* : \left\{\begin{array}{l}
h_i(x_*) = 0\,\forall i \ \&\ g_j(x_*) \leq 0\,\forall j \ \&\ x_* \in X \\
h_i(x) = 0\,\forall i \ \&\ g_j(x) \leq 0\,\forall j \ \&\ x \in X \\
\qquad\qquad\qquad\qquad \Rightarrow f(x_*) \leq f(x)
\end{array}\right.
$$

$$\begin{array}{rcl} \min\limits_{x} f(x) & & \\ \text{s.t.} & & \\ h_i(x) & = & 0,\ i = 1, ..., m \\ g_j(x) & \leq & 0,\ j = 1, ..., k \\ x & \in & X \end{array} \qquad \text{(MP)}$$

♣ In *Combinatorial* (or *Discrete*) Optimization, the domain X is a discrete set, like the set of all integral or 0/1 vectors.

In contrast to this, in *Continuous* Optimization we will focus on, X is a "continuum" set like the entire $\mathbf{R}^n$, a *box* $\{x : a \leq x \leq b\}$, or *simplex* $\{x \geq 0 : \sum\limits_j x_j = 1\}$, etc., and the objective and the constraints are (at least) continuous on X.

♣ In *Linear Optimization*, $X = \mathbf{R}^n$ and the objective and the constraints are linear functions of x.
In contrast to this, in *Nonlinear Continuous Optimization*, the objective and the constraints can be nonlinear functions.

♣ Our course is on *Linear Optimization*, the simplest and the most frequently used in applications part of Mathematical Programming. Some of the reasons for LO to be popular are:

• reasonable "expressive power" of LO — while the world we live in is mainly nonlinear, linear dependencies in many situations can be considered as quite satisfactory approximations of actual nonlinear dependencies. At the same time, a linear dependence is easy to specify, which makes it realistic to specify data of Linear Optimization models with many variables and constraints;

• existence of extremely elegant, rich and essentially complete mathematical theory of LO;

• last, but by far not least, existence of extremely powerful solution algorithms capable to solve to optimality in reasonable time LO problems with tens and hundreds of thousands of variables and constraints.

♣ In our course, we will focus primarily on "LO machinery" (LO Theory and Algorithms), leaving beyond our scope practical applications of LO which are by far too numerous and diverse to be even outlined in a single course. The brief outline of the contents is as follows:

• *LO Modeling*, including instructive examples of LO models and "calculus" of LO models – collection of tools allowing to recognize the possibility to pose an optimization problem as an LO program;

• *LO Theory* – geometry of LO programs, existence and characterization of optimal solutions, theory of systems of linear inequalities and duality;

• *LO Algorithms*, including Simplex-type and Interior Point ones, and the associated complexity issues.

PART I.
LO: Descriptive Theory

Lecture I.1
LO Models

Linear Optimization Models

♣ **An LO program.** A *Linear Optimization problem*, or program (LO), called also *Linear Programming* problem/program, is the problem of optimizing a *linear function* $c^T x$ of an n-dimensional vector x under *finitely many linear* equality and nonstrict inequality constraints.

♣ The Mathematical Programming problem

$$\min_x \left\{ x_1 : \left\{ \begin{array}{rcl} x_1 + x_2 & \leq & 20 \\ x_1 - x_2 & = & 5 \\ x_1, x_2 & \geq & 0 \end{array} \right\} \right\} \tag{1}$$

is an LO program.

♣ The problem

$$\min_x \left\{ \exp\{x_1\} : \left\{ \begin{array}{rcl} x_1 + x_2 & \leq & 20 \\ x_1 - x_2 & = & 5 \\ x_1, x_2 & \geq & 0 \end{array} \right\} \right\} \tag{1$'$}$$

is not an LO program, since the objective in (1′) is nonlinear.

♣ The problem

$$\max_x \left\{ x_1 + x_2 : \left\{ \begin{array}{rcl} 2x_1 & \geq & 20 - x_2 \\ x_1 - x_2 & = & 5 \\ x_1 & \geq & 0 \\ x_2 & \leq & 0 \end{array} \right\} \right. \tag{2}$$

is an LO program.

♣ The problem

$$\max_x \left\{ x_1 + x_2 : \left\{ \begin{array}{rcl} \forall i \geq 2: & & \\ ix_1 & \geq & 20 - x_2, \\ x_1 - x_2 & = & 5 \\ x_1 & \geq & 0 \\ x_2 & \leq & 0 \end{array} \right\} \right. \tag{2$'$}$$

is not an LO program – it has infinitely many linear constraints.

♠ **Note:** Property of an MP problem to be or not to be an LO program is the property of a *representation* of the problem. We classify optimization problems according to *how they are presented*, and not according to *what they can be equivalent/reduced to*.

Canonical and Standard forms of LO programs

♣ Observe that we can somehow unify the forms in which LO programs are written. Indeed

• every linear equality/inequality can be equivalently rewritten in the form where the left hand side is a weighted sum $\sum_{j=1}^{n} a_j x_j$ of variables x_j with coefficients, and the right hand side is a real constant:

$$2x_1 \geq 20 - x_2 \Leftrightarrow 2x_1 + x_2 \geq 20$$

• the sign of a nonstrict linear inequality always can be made "$\leq$", since the inequality $\sum_j a_j x_j \geq b$ is equivalent to $\sum_j [-a_j] x_j \leq [-b]$:

$$2x_1 + x_2 \geq 20 \Leftrightarrow -2x_1 - x_2 \leq -20$$

• a linear equality constraint $\sum_j a_j x_j = b$ can be represented equivalently by the pair of opposite inequalities $\sum_j a_j x_j \leq b$, $\sum_j [-a_j] x_j \leq [-b]$:

$$2x_1 - x_2 = 5 \Leftrightarrow \begin{cases} 2x_1 - x_2 \leq 5 \\ -2x_1 + x_2 \leq -5 \end{cases}$$

• to minimize a linear function $\sum_j c_j x_j$ is exactly the same to maximize the linear function $\sum_j [-c_j] x_j$.

♣ *Every LO program is equivalent to an LO program in the canonical form*, where the objective should be maximized, and all the constraints are $\leq$-inequalities:

$$\text{Opt} = \max_x \left\{ \sum_{j=1}^n c_j x_j : \sum_{j=1}^n a_{ij} x_j \leq b_i, \; 1 \leq i \leq m \right\}$$

["term-wise" notation]

$$\Leftrightarrow \quad \text{Opt} = \max_x \left\{ c^T x : a_i^T x \leq b_i, \; 1 \leq i \leq m \right\}$$

["constraint-wise" notation]

$$\Leftrightarrow \quad \text{Opt} = \max_x \left\{ c^T x : Ax \leq b \right\}$$

["matrix-vector" notation]

$c = [c_1; ...; c_n], \; b = [b_1; ...; b_m],$
$a_i = [a_{i1}; ...; a_{in}], A = [a_1^T; a_2^T; ...; a_m^T]$

Attention! In our course, all vectors are *column* vectors. However, to save space, they are written "in a row" – as sequences of entries

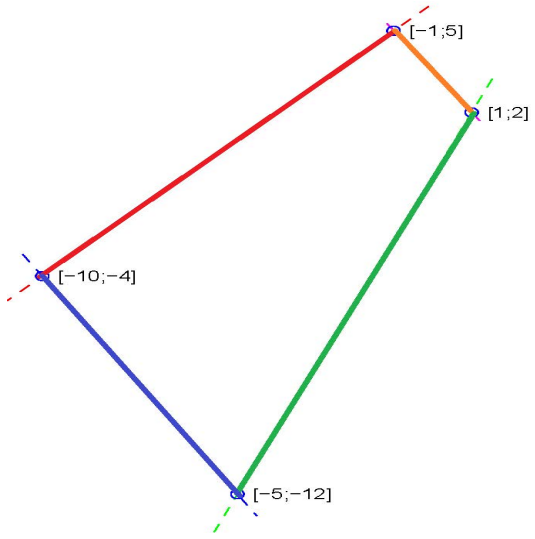
$$x = [x_1; ...; x_n]$$

separated by semicolons **;**

• Similarly for matrices: $A = [a_1^T; a_2^T; ...; a_m^T]$ means that the rows of A are the transposes a_i^T of column vectors a_i and separating semicolons say that there rows should be written beneath each other.

♠ A set $X \subset \mathbb{R}^n$ given by $X = \{x : Ax \leq b\}$ – the solution set of a finite system of nonstrict linear inequalities $a_i^T x \leq b_i$, $1 \leq i \leq m$ in variables $x \in \mathbb{R}^n$ – is called *polyhedral set*, or *polyhedron*. An LO program in the canonical form is to maximize a linear objective over a polyhedral set. ♠ **Note:** The solution set of an arbitrary finite system of linear equalities and nonstrict inequalities in variables $x \in \mathbb{R}^n$ is a polyhedral set.

$$\max_x \left\{ x_2 : \left\{ \begin{array}{rcl} -x_1 + x_2 & \leq & 6 \\ 3x_1 + 2x_2 & \leq & 7 \\ 7x_1 - 3x_2 & \leq & 1 \\ -8x_1 - 5x_2 & \leq & 100 \end{array} \right\} \right\}$$



LO program and its feasible domain

♣ **Standard form of an LO program** is to maximize a linear function over the intersection of the *nonnegative orthant* $\mathbb{R}^n_+ = \{x \in \mathbb{R}^n : x \geq 0\}$ and the *feasible plane* $\{x : Ax = b\}$:

$$\text{Opt} = \max_x \left\{ \sum_{j=1}^n c_j x_j : \begin{array}{r} \sum_{j=1}^n a_{ij} x_j = b_i, \\ 1 \leq i \leq m \\ x_j \geq 0,\ j = 1, ..., n \end{array} \right\}$$

["term-wise" notation]

$$\Leftrightarrow \quad \text{Opt} = \max_x \left\{ c^T x : \begin{array}{l} a_i^T x = b_i,\ 1 \leq i \leq m \\ x_j \geq 0,\ 1 \leq j \leq n \end{array} \right\}$$

["constraint-wise" notation]

$$\Leftrightarrow \quad \text{Opt} = \max_x \left\{ c^T x : Ax = b, x \geq 0 \right\}$$

["matrix-vector" notation]

$c = [c_1; ...; c_n],\ b = [b_1; ...; b_m],$
$a_i = [a_{i1}; ...; a_{in}], A = [a_1^T; a_2^T; ...; a_m^T]$

In the standard form LO program

- all variables are restricted to be nonnegative
- all "general-type" linear constraints are equalities.

♣ **Observation:** *The standard form of LO program is universal: every LO program is equivalent to an LO program in the standard form.*

Indeed, it suffices to convert to the standard form a canonical LO $\max_x\{c^Tx : Ax \leq b\}$. This can be done as follows:

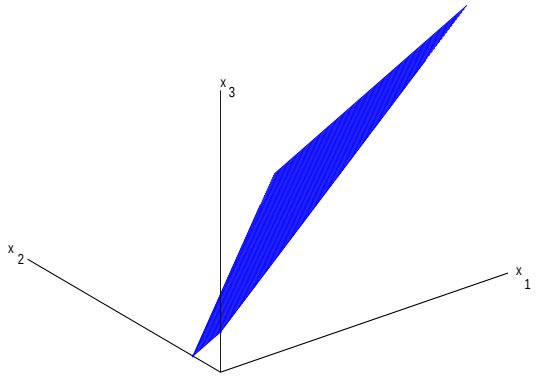
• we introduce *slack variables*, one per inequality constraint, and rewrite the problem equivalently as

$$\max_{x,s}\{c^Tx : Ax+s = b,\, s \geq 0\}$$

• we further represent x as the difference of two new *nonnegative* vector variables $x = u - v$, thus arriving at the program

$$\max_{u,v,s}\left\{c^Tu - c^Tv : Au - Av + s = b,\, [u; v; s] \geq 0\right\}.$$

$$\max_x \{-2x_1 + x_3 : -x_1 + x_2 + x_3 = 1,\, x \geq 0\}$$



Standard form LO program
and its feasible domain

LO Terminology

$$\begin{array}{c}\text{Opt} = \max\limits_{x}\{c^T x : Ax \leq b\} \\ [A : m \times n]\end{array} \qquad \text{(LO)}$$

- The variable vector x in (LO) is called the *decision vector* of the program; its entries x_j are called *decision variables*.
- The linear function $c^T x$ is called the *objective function* (or *objective*) of the program, and the inequalities $a_i^T x \leq b_i$ are called the *constraints*.
- The *structure* of (LO) reduces to the *sizes* m (number of constraints) and n (number of variables). The *data* of (LO) is the collection of numerical values of the coefficients in the *cost vector* c, in the *right hand side vector* b and in the *constraint matrix* A.

$$\text{Opt} = \max_x \{c^T x : Ax \leq b\} \qquad \text{(LO)}$$
$$[A : m \times n]$$

• A *solution* to (LO) is an arbitrary value of the decision vector. A solution x is called *feasible* if it satisfies the constraints: $Ax \leq b$. The set of all feasible solutions is called the *feasible set* of the program. The program is called *feasible*, if the feasible set is nonempty, and is called *infeasible* otherwise.

$$\mathrm{Opt} = \max_x \{c^T x : Ax \leq b\} \qquad \text{(LO)}$$
$$[A : m \times n]$$

• Given a program (LO), there are three possibilities:

— the program is infeasible. In this case, $\mathrm{Opt} = -\infty$ by definition.

— the program is feasible, and the objective is *not* bounded from above on the feasible set, i.e., for every $a \in \mathbb{R}$ there exists a feasible solution x such that $c^T x > a$. In this case, the program is called *unbounded*, and $\mathrm{Opt} = +\infty$ by definition.

The program which is not unbounded is called *bounded*; a program is bounded iff its objective is bounded from above on the feasible set (e.g., due to the fact that the latter is empty).

— the program is feasible, and the objective is bounded from above on the feasible set: there exists a real a such that $c^T x \leq a$ for all feasible solutions x. In this case, the optimal value Opt is the supremum, over the feasible solutions, of the values of the objective at a solution.

$$\mathrm{Opt} = \max_x \{c^T x : Ax \leq b\} \qquad [A : m \times n] \qquad \text{(LO)}$$

• a solution to the program is called *optimal*, if it is feasible, and the value of the objective at the solution equals to Opt. A program is called *solvable*, if it admits an optimal solution.

♠ **In the case of a minimization problem**

$$\mathrm{Opt} = \min_x \{c^T x : Ax \leq b\} \qquad [A : m \times n] \qquad \text{(LO)}$$

• the optimal value of an infeasible program is $+\infty$,

• the optimal value of a feasible and *unbounded* program (unboundedness now means that the objective to be minimized is not bounded *from below* on the feasible set) is $-\infty$

• the optimal value of a *bounded and feasible* program is the *infimum* of values of the objective at feasible solutions to the program.

♣ The notions of feasibility, boundedness, solvability and optimality can be straightforwardly extended from LO programs to arbitrary MP ones. With this extension, a solvable problem definitely is feasible and bounded, while the inverse not necessarily is true, as is illustrated by the program

$$\mathrm{Opt} = \max_x \{-\exp\{-x\} : x \geq 0\},$$

Opt $= 0$, but the optimal value is not achieved – there is no feasible solution where the objective is equal to 0! As a result, the program is unsolvable.

$\Rightarrow$ In general, the fact that an optimization program with a "legitimate" – real, and not $\pm\infty$ – optimal value, is *strictly weaker* that the fact that the program is solvable (i.e., has an optimal solution).

♠ In LO the situation is much better: we shall prove that *an LO program is solvable iff it is feasible and bounded*.

Examples of LO Models

♣ **Diet Problem:** There are n types of products and m types of nutrition elements. A unit of product # j contains p_{ij} grams of nutrition element # i and costs c_j. The daily consumption of a nutrition element # i should be within given bounds $[\underline{b}_i, \overline{b}_i]$. Find the cheapest possible "diet" – mixture of products – which provides appropriate daily amounts of every one of the nutrition elements.

Denoting x_j the amount of j-th product in a diet, the LO model reads

$$\min_x \sum_{j=1}^n c_j x_j \quad \text{[cost to be minimized]}$$

subject to

$$\left.\begin{array}{l} \sum_{j=1}^n p_{ij} x_j \geq \underline{b}_i \\ \sum_{j=1}^n p_{ij} x_j \leq \overline{b}_i \\ 1 \leq i \leq m \end{array}\right\} \quad \left[\begin{array}{l} \text{upper \& lower bounds on} \\ \text{the contents of nutrition} \\ \text{elements in a diet} \end{array}\right]$$

$$x_j \geq 0, 1 \leq j \leq n \quad \left[\begin{array}{l} \text{you cannot put into a} \\ \text{diet a negative amount} \\ \text{of a product} \end{array}\right]$$

- Diet problem is routinely used in nourishment of poultry, livestock, etc. As about nourishment of humans, the model is of no much use since it ignores factors like food's taste, food diversity requirements, etc.
- Here is the optimal daily human diet as computed by the software at

`https://neos-guide.org/case-studies/om/the-diet-problem/`

`(when solving the problem, I allowed to use all 68 kinds of food offered by the code):`

Food	Serving	Cost
Raw Carrots	0.12 cups shredded	0.02
Peanut Butter	7.20 Tbsp	0.25
Popcorn, Air-Popped	4.82 Oz	0.19
Potatoes, Baked	1.77 cups	0.21
Skim Milk	2.17 C	0.28

Daily cost $ 0.96

♣ **Production planning:** A factory

• consumes R types of resources (electricity, raw materials of various kinds, various sorts of manpower, processing times at different devices, etc.)

• produces P types of products.

• There are n possible production processes, j-th of them can be used with "intensity" x_j (intensities are fractions of the planning period during which a particular production process is used).

• Used at unit intensity, production process # j consumes A_{rj} units of resource r, $1 \leq r \leq R$, and yields C_{pj} units of product p, $1 \leq p \leq P$.

• The profit of selling a unit of product p is c_p.

♠ Given upper bounds $b_1, ..., b_R$ on the amounts of various recourses available during the planning period, and lower bounds $d_1, ..., d_P$ on the amount of products to be produced, find a production plan which maximizes the profit under the resource and the demand restrictions.

Denoting by x_j the intensity of production process j, the LO model reads:

$$\max_x \sum_{j=1}^{n} \left(\sum_{p=1}^{P} c_p C_{pj} \right) x_j \quad \text{[profit to be maximized]}$$

subject to

$$\sum_{j=1}^{n} A_{rj} x_j \leq b_r,\ 1 \leq r \leq R \quad \left[\begin{array}{l} \text{upper bounds on} \\ \text{resources should} \\ \text{be met} \end{array} \right]$$

$$\sum_{j=1}^{n} C_{pj} x_j \geq d_p,\ 1 \leq p \leq P \quad \left[\begin{array}{l} \text{lower bounds on} \\ \text{products should} \\ \text{be met} \end{array} \right]$$

$$\left. \begin{array}{l} \sum_{j=1}^{n} x_j \leq 1 \\ x_j \geq 0,\ 1 \leq j \leq n \end{array} \right\} \quad \left[\begin{array}{l} \text{total intensity should be} \leq 1 \\ \text{and intensities must be} \\ \text{nonnegative} \end{array} \right]$$

Implicit assumptions:

- all production can be sold
- there are no setup costs when switching between production processes
- the products are infinitely divisible

♣ **Inventory:** An inventory operates over time horizon of T days $1, ..., T$ and handles K types of products.

• Products share common warehouse with space C. Unit of product k takes space $c_k \geq 0$ and its day-long storage costs h_k.

• Inventory is replenished via ordering from a supplier; a replenishment order sent in the beginning of day t is executed immediately, and ordering a unit of product k costs $o_k \geq 0$.

• The inventory is affected by external demand of d_{tk} units of product k in day t. Backlog is allowed, and a day-long delay in supplying a unit of product k costs $p_k \geq 0$.

♠ *Given the initial amounts s_{0k}, $k = 1, ..., K$, of products in warehouse, all the (nonnegative) cost coefficients and the demands d_{tk}, we want to specify the replenishment orders v_{tk} (v_{tk} is the amount of product k which is ordered from the supplier at the beginning of day t) in such a way that at the end of day T there is no backlogged demand, and we want to meet this requirement at as small total inventory management cost as possible.*

Building the model

1. Let *state variable* s_{tk} be the amount of product k stored at warehouse at the end of day t. s_{tk} can be negative, meaning that at the end of day t the inventory owes the customers $|s_{tk}|$ units of product k.
Let also U be an upper bound on the total management cost. The problem reads:

$$\min_{U,v,s} \quad U$$

$$U \geq \sum_{\substack{1\leq k\leq K,\\ 1\leq t\leq T}} [o_k v_{tk} + \max[h_k s_{tk}, 0] + \max[-p_k s_{tk}, 0]]$$

[cost description]

$$s_{tk} = s_{t-1,k} + v_{tk} - d_{tk}, 1 \leq t \leq T, 1 \leq k \leq K$$

[state equations]

$$\sum_{k=1}^{K} \max[c_k s_{tk}, 0] \leq C,\ 1 \leq t \leq T$$

[space restriction should be met]

$$s_{Tk} \geq 0, 1 \leq k \leq K$$

[no backlogged demand at the end]

$$v_{tk} \geq 0, 1 \leq k \leq K, 1 \leq t \leq T$$

Implicit assumption: replenishment orders are executed, and the demands are shipped to customers at the beginning of day t.

♠ Our problem is *not* and LO program – it includes *nonlinear* constraints of the form

$$\sum_{k,t}[o_k v_{tk} + \max[h_k s_{tk}, 0] + \max[-p_k s_{tk}, 0]] \le U$$
$$\sum_k \max[c_k s_{tk}, 0] \le C,\ t = 1, ..., T$$

Let us show that these constraints can be *represented equivalently* by linear constraints.

♣ Consider a MP problem in variables x with linear objective $c^T x$ and constraints of the form

$$a_i^T x + \sum_{j=1}^{n_i} \mathrm{Term}_{ij}(x) \le b_i,\ 1 \le i \le m,$$

where $\mathrm{Term}_{ij}(x)$ are *convex piecewise linear* functions of x, that is, *maxima of affine functions of* x:

$$\mathrm{Term}_{ij}(x) = \max_{1 \le \ell \le L_{ij}} [\alpha_{ij\ell}^T x + \beta_{ij\ell}]$$

♣ **Observation:** MP problem

$$\max_x \left\{ c^T x : a_i^T x + \sum_{j=1}^{n_i} \mathrm{Term}_{ij}(x) \le b_i,\ i \le m \right\} \qquad (P)$$
$$\mathrm{Term}_{ij}(x) = \max_{1 \le \ell \le L_{ij}} [\alpha_{ij\ell}^T x + \beta_{ij\ell}]$$

is equivalent to the LO program

$$\max_{x,\tau_{ij}} \left\{ c^T x : \begin{array}{r} a_i^T x + \sum_{j=1}^{n_i} \tau_{ij} \le b_i \\ \tau_{ij} \ge \alpha_{ij\ell}^T x + \beta_{ij\ell}, \\ 1 \le \ell \le L_{ij} \end{array} , i \le m \right\} \qquad (P')$$

in the sense that both problems have the same objectives and x is feasible for (P) *iff* x can be extended, by properly chosen τ_{ij}, to a feasible solution to (P'). As a result,

- every feasible solution (x, τ) to (P') induces a feasible solution x to (P), with the same value of the objective;
- vice versa, every feasible solution x to (P) can be obtained in the above fashion from a feasible solution to (P').

♠ Applying the above construction to the Inventory problem, we end up with the following LO model:

$$
\begin{array}{l}
\min\limits_{U,v,s,x,y,z} \quad U \\
U \geq \sum\limits_{k,t} [o_k v_{tk} + x_{tk} + y_{tk}] \\
s_{tk} = s_{t-1,k} + v_{tk} - d_{tk}, 1 \leq t \leq T, 1 \leq k \leq K \\
\sum_{k=1}^{K} z_{tk} \leq C, \; 1 \leq t \leq T \\
x_{tk} \geq h_k s_{tk}, \; x_{tk} \geq 0, \; 1 \leq k \leq K, 1 \leq t \leq T \\
y_{tk} \geq -p_k s_{tk}, \; y_{tk} \geq 0, \; 1 \leq k \leq K, 1 \leq t \leq T \\
z_{tk} \geq c_k s_{tk}, \; z_{tk} \geq 0, \; 1 \leq k \leq K, 1 \leq t \leq T \\
s_{Tk} \geq 0, 1 \leq k \leq K \\
v_{tk} \geq 0, \; 1 \leq k \leq K, 1 \leq t \leq T
\end{array}
$$

♣ **Warning:** The outlined "eliminating piecewise linear nonlinearities" heavily exploits the facts that *after the nonlinearities are moved to the left hand sides of $\leq$-constraints, they can be written down as the maxima of affine functions.*

Indeed, the attempt to eliminate nonlinearity

$$\min_\ell [\alpha_{ij\ell}^T x + \beta_{ij\ell}]$$

in the constraint

$$... + \min_\ell [\alpha_{ij\ell}^T x + \beta_{ij\ell}] \leq b_i$$

by introducing upper bound τ_{ij} on the nonlinearity and representing the constraint by the pair of constraints

$$\begin{aligned} ... + \tau_{ij} &\leq b_i \\ \tau_{ij} &\geq \min_\ell [\alpha_{ij\ell}^T x + \beta_{ij\ell}] \end{aligned}$$

fails, since the red constraint in the pair, in general, is *not* representable by a system of linear inequalities.

♣ **Transportation problem:** There are I warehouses, i-th of them storing s_i units of product, and J customers, j-th of them demanding d_j units of product. Shipping a unit of product from warehouse i to customer j costs c_{ij}. Given the supplies s_i, the demands d_j and the costs C_{ij}, we want to decide on the amounts of product to be shipped from every warehouse to every customer. Our restrictions are that we cannot take from a warehouse more product than it has, and that all the demands should be satisfied; under these restrictions, we want to minimize the total transportation cost. Let x_{ij} be the amount of product shipped from warehouse i to customer j. The problem reads

$$\min_x \sum_{i,j} c_{ij} x_{ij} \qquad \left[\begin{array}{l}\text{transportation cost}\\ \text{to be minimized}\end{array}\right]$$

subject to

$$\sum_{j=1}^{J} x_{ij} \le s_i,\ 1 \le i \le I \qquad \left[\begin{array}{l}\text{bounds on supplies}\\ \text{of warehouses}\end{array}\right]$$

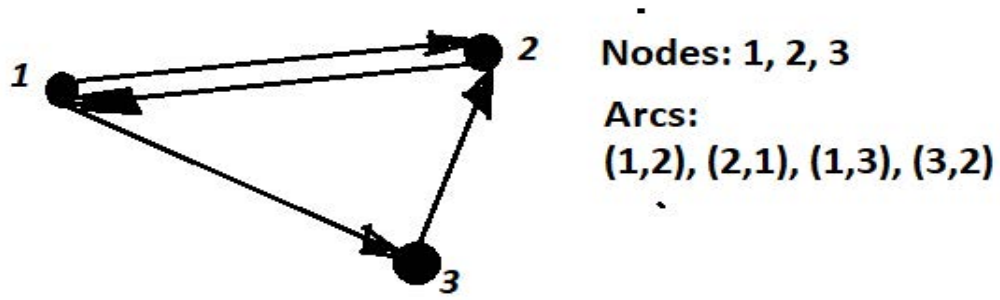
$$\sum_{i=1}^{I} x_{ij} = d_j,\ j = 1, ..., J \qquad \left[\begin{array}{l}\text{demands should}\\ \text{be satisfied}\end{array}\right]$$

$$x_{ij} \ge 0, 1 \le i \le I, 1 \le j \le J \qquad \left[\begin{array}{l}\text{no negative}\\ \text{shipments}\end{array}\right]$$

♣ **Multicommodity Flow:**

• We are given a *network* (an oriented graph), that is, a finite set of *nodes* $1, 2, ..., n$ along with a finite set Γ of *arcs* — ordered pairs $\gamma = (i, j)$ of distinct nodes. We say that an arc $\gamma = (i, j) \in \Gamma$ *starts* at node i, *ends* at node j and *links* nodes i and j.

♠ **Example: Road network** with road junctions as nodes and one-way road segments "from a junction to a neighbouring junction" as arcs.



• There are N types of "commodities" moving along the network, and we are given the "external supply" s_{ki} of k-th commodity at node i. When $s_{ki} \geq 0$, the node i "pumps" into the network s_{ki} units of commodity k; when $s_{ki} \leq 0$, the node i "drains" from the network $|s_{ki}|$ units of commodity k.

♠ k-th commodity in a road network with steady-state traffic can be composed of all cars leaving within an hour a particular origin (e.g., GaTech campus) towards a particular destination (e.g., Northside Hospital).

• Propagation of commodity k through the network is represented by a *flow vector* f^k. The entries in f^k are indexed by arcs, and f^k_γ is the flow of commodity k in an arc γ.

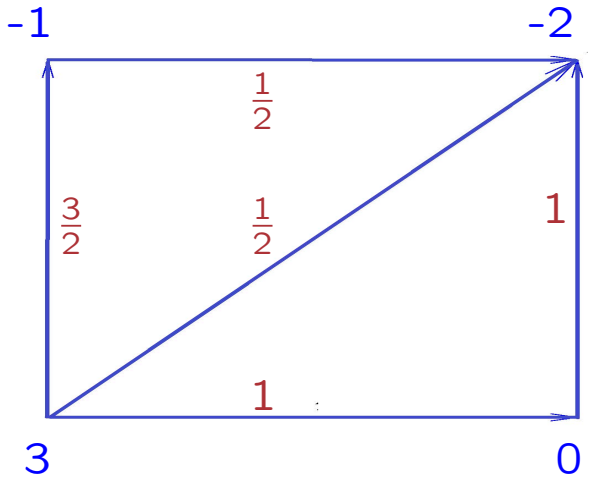
♠ In a road network with steady-state traffic, an entry f^k_γ of the flow vector f^k is the amount of cars from k-th origin-destination pair which move within an hour through the road segment γ.

• A flow vector f^k is called a *feasible flow*, if it is nonnegative and satisfies the

Conservation law: *for every node i, the total amount of commodity k entering the node plus the external supply s_{ki} of the commodity at the node is equal to the total amount of the commodity leaving the node:*

$$\sum_{p\in P(i)} f^k_{pi} + s_{ki} = \sum_{q\in Q(i)} f^k_{iq}$$

$$P(i) = \{p : (p,i) \in \Gamma\};\ Q(i) = \{q : (i,q) \in \Gamma\}$$



♣ **Multicommodity flow problem** reads: *Given*

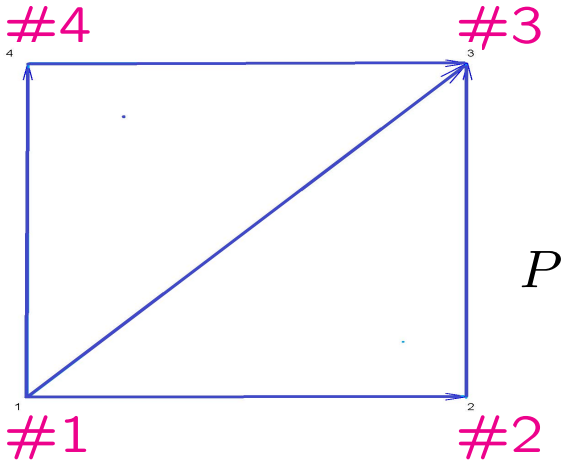
- *a network with n nodes $1, ..., n$ and a set Γ of arcs,*
- *a number K of commodities along with supplies s_{ki} of nodes $i = 1, ..., n$ to the flow of commodity k, $k = 1, ..., K$,*
- *the per unit cost $c_{k\gamma}$ of transporting commodity k through arc γ,*
- *the capacities h_γ of the arcs,*

find the flows $f^1, ..., f^K$ of the commodities which are nonnegative, respect the Conservation law and the capacity restrictions (that is, the total, over the commodities, flow through an arc does not exceed the capacity of the arc) *and minimize, under these restrictions, the total, over the arcs and the commodities, transportation cost.*

♠ In the Road Network illustration, interpreting $c_{k\gamma}$ as the travel time along road segment γ, the Multicommodity flow problem becomes the one of finding *social optimum*, where the total travelling time of all cars is as small as possible.

♣ We associate with a network with n nodes and m arcs an $n \times m$ *incidence matrix* P defined as follows:

- the rows of P are indexed by the nodes $1, ..., n$
- the columns of P are indexed by arcs γ
- $P_{i\gamma} = \begin{cases} 1, & \gamma \text{ starts at } i \\ -1, & \gamma \text{ ends at } i \\ 0, & \text{all other cases} \end{cases}$



$$P = \begin{array}{c} \begin{matrix} (1,2) & (1,3) & (1,4) & (2,3) & (4,3) \end{matrix} \\ \begin{bmatrix} 1 & 1 & 1 & & \\ -1 & & & 1 & \\ & -1 & & -1 & -1 \\ & & -1 & & 1 \end{bmatrix} \end{array} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}$$

♠ *In terms of the incidence matrix, the Conservation Law linking flow f and external supply s reads*

$$Pf = s$$

♠ **Multicommodity flow problem** reads

$$\min_{f^1,...,f^K} \sum_{k=1}^{K} \sum_{\gamma \in \Gamma} c_{k\gamma} f_\gamma^k$$

[transportation cost to be minimized]

subject to

$$Pf^k = s^k := [s_{k1}; ...; s_{kn}],\ k = 1, ..., K$$

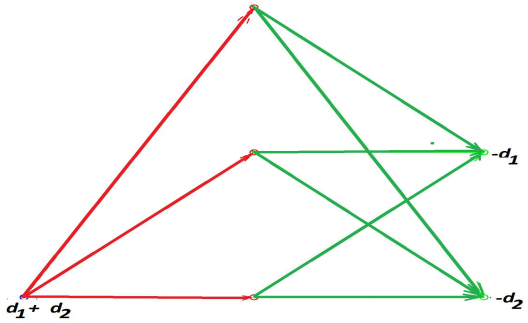
[flow conservation laws]

$$f_\gamma^k \geq 0,\ 1 \leq k \leq K, \gamma \in \Gamma$$

[flows must be nonnegative]

$$\sum_{k=1}^{K} f_\gamma^k \leq h_\gamma, \gamma \in \Gamma$$

[bounds on arc capacities]



red nodes: warehouses green nodes: customers

green arcs: transpostation costs c_{ij}, capacities $+\infty$

red arcs: transporation costs 0, capacities s_i

Note: Transportation problem is a particular case of the Multicommodity flow one. Here we

- start with I red nodes representing warehouses, J green nodes representing customers and IJ arcs "warehouse i – customer j;" these arcs have infinite capacities and transportation costs c_{ij};
- augment the network with *source node* which is linked to every warehouse node i by an arc with zero transportation cost and capacity s_i;
- consider the single-commodity case where the source node has external supply $\sum_j d_j$, the customer nodes have external supplies $-d_j$, $1 \leq j \leq J$, and the warehouse nodes have zero external supplies.

♣ **Maximum Flow problem:** Given a network with two selected nodes – a *source* and a *sink*, find the maximal flow from the source to the sink, that is, find the largest s such the external supply "s at the source node, $-s$ at the sink node, 0 at all other nodes" corresponds to a feasible flow respecting arc capacities.

The problem reads

$$\max_{f,s} \quad s \quad \left[\begin{array}{l} \text{total flow from source to} \\ \text{sink to be maximized} \end{array} \right]$$

subject to

$$\sum_\gamma P_{i\gamma} f_\gamma = \begin{cases} s, & i \text{ is the source node} \\ -s, & i \text{ is the sink node} \\ 0, & \text{for all other nodes} \end{cases} \quad \text{[flow conservation law]}$$

$f_\gamma \geq 0,\ \gamma \in \Gamma$ [arc flows should be ≥ 0]

$f_\gamma \leq h_\gamma, \gamma \in \Gamma$ [we should respect arc capacities]

LO Models in Engineering
A. "Flying Helicopter"

♣ **The story:** *A particle moves through $\mathbb{R}^n$. Given positions and velocities of the particle at times $t = 0$ and $t = 1$, we want to find trajectory minimizing the worst-case acceleration of the particle.*

♠ **In continuous time** the problem reads: *Find a curve $x(t) \in \mathbb{R}^n$, $0 \le t \le 1$ satisfying boundary conditions $x(0) = x^0, \dot{x}(0) = v^0, x(1) = x^1, \dot{x}(1) = v^1$ and minimizing* $\max_{0\le t\le 1} \|\ddot{x}(t)\|_2$ ($\|x\|_2 = \sqrt{x^T x}$: the standard Euclidean norm)

♠ **To process the problem numerically**, we pass to *discrete time model*:

— replace continuous time with the grid $\{t_\tau = \tau/N, \tau = 0, \pm 1, \pm 2, ...\}$ with resolution $dt = 1/N$

— restrict $x(t)$ on the grid, just getting collection of vectors $\{x_\tau = x(t_\tau), \tau = 0, \pm 1, \pm 2, ...\}$

— approximate the velocity $\dot{x}(x_\tau)$ by finite difference $v_\tau = [x_{\tau+1} - x_\tau]/dt$, and acceleration $\ddot{x}(t_\tau)$ – by the second order finite difference $[v_{\tau+1} - v_\tau]/dt = [x_{\tau+2} - 2x_{\tau+1} + x_\tau]/dt^2$

— finally, we translate the boundary conditions into the constraints $x_0 = x^0, x_1 = x^0 + dt \cdot v^0$, $x_N = x^1, x_{N+1} = x^1 + dt \cdot v^1$.

• The resulting discrete time problem reads

$$\min_{x_2,...,x_{N-1},\theta} \left\{\theta : \|x_{\tau+2} - 2x_{\tau+1} + x_\tau\|_2/dt^2 \le \theta,\ 0 \le \tau < N\right\}$$

$$\left[\begin{array}{rcl} x_0 &=& x^0 \\ x_1 &=& x^0 + dt \cdot v^0 \\ x_N &=& x^1 \\ x_{N+1} &=& x^1 + dt \cdot v^1 \end{array}\right]$$

$$\min_{x_2,\ldots,x_{N-1},\theta}\left\{\theta : \|x_{\tau+2}-2x_{\tau+1}+x_\tau\|_2/dt^2 \le \theta,\ 0\le\tau<N\right\} \qquad \left[\begin{array}{rcl} x_0 & = & x^0 \\ x_1 & = & x^0+dt\cdot v^0 \\ x_N & = & x^1 \\ x_{N+1} & = & x^1+dt\cdot v^1 \end{array}\right] \quad \text{(P)}$$

♠ (P) is *not* an LP problem!

Fact: We shall see in the mean time that *conic quadratic constraint*

$$\|\langle \textit{vector linearly depending on decision variables}\rangle\|_2 \le \langle \textit{linear function of decision variables}\rangle$$

"for al practical purposes" is a system of linear inequalities – it can approximated, to whatever high accuracy (say, with machine precision) *by a system of linear inequalities of quite moderate size.*

♠ 3D case of our problem can be interpreted as planning a maneuver of helicopter bringing it from one prescribed state (position & velocity) to another prescribed state.

- **Question:** *Why "maneuver of helicopter," and not of aircraft?*

- **Answer:** We did not impose bounds on the speed of our "particle"

✓imposing *upper* bounds on the speed is easy and does not change problem's complexity status

✓imposing *lower* bounds on the speed makes the problem *computationally intractable*...

✓*For aircraft*, lower bounding the speed is a must – with speed below certain level, aircraft will drop...

✓*For helicopter*, no lower bound on speed is needed, while controlling acceleration still is important due to possibility of high speeds:

  - top speeds of attack helicopters are in the range $180 - 255$ mph

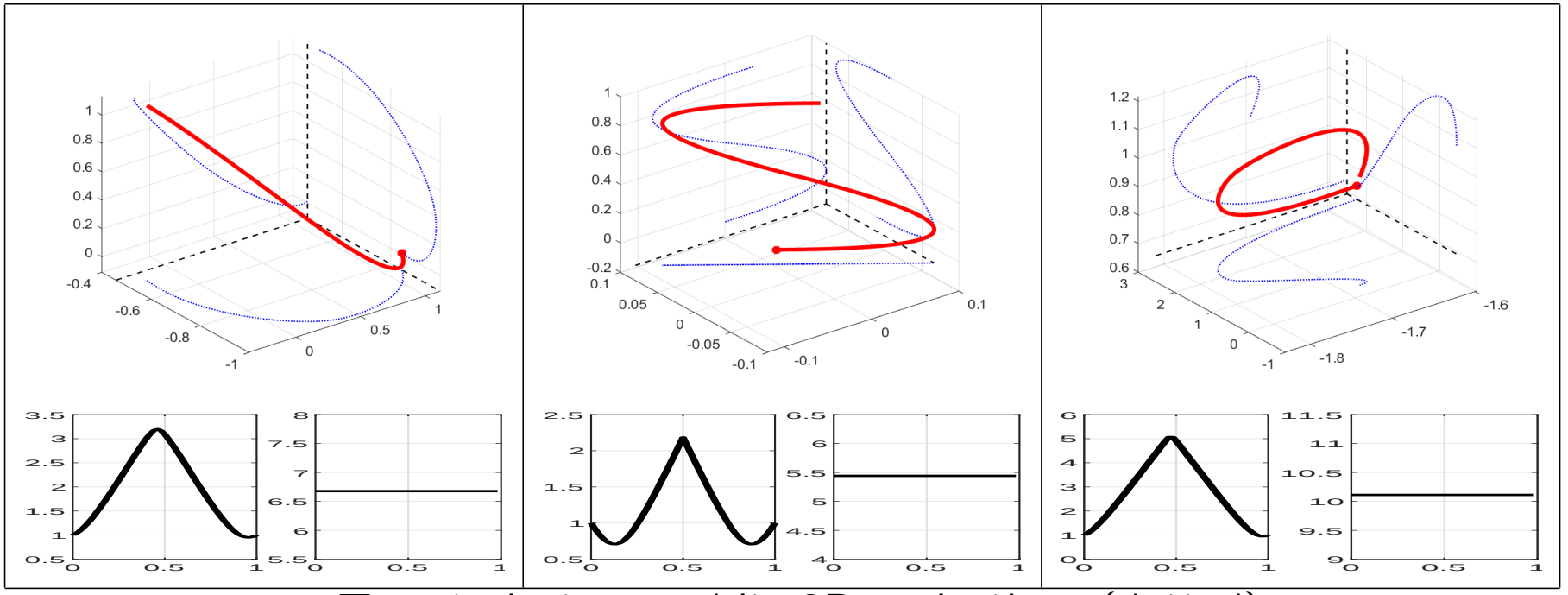
Compare with

  - top speeds of race cars are in the range $160 - 230$ mph

$$\min_{x_2,\ldots,x_{N-1},\theta}\left\{\theta : \|x_{\tau+2} - 2x_{\tau+1} + x_\tau\|_2/dt^2 \le \theta,\ 0 \le \tau < N\right\} \qquad \left[\begin{array}{rcl} x_0 &=& x^0 \\ x_1 &=& x^0 + dt\cdot v^0 \\ x_N &=& x^1 \\ x_{N+1} &=& x^1 + dt\cdot v^1 \end{array}\right] \quad \text{(P)}$$

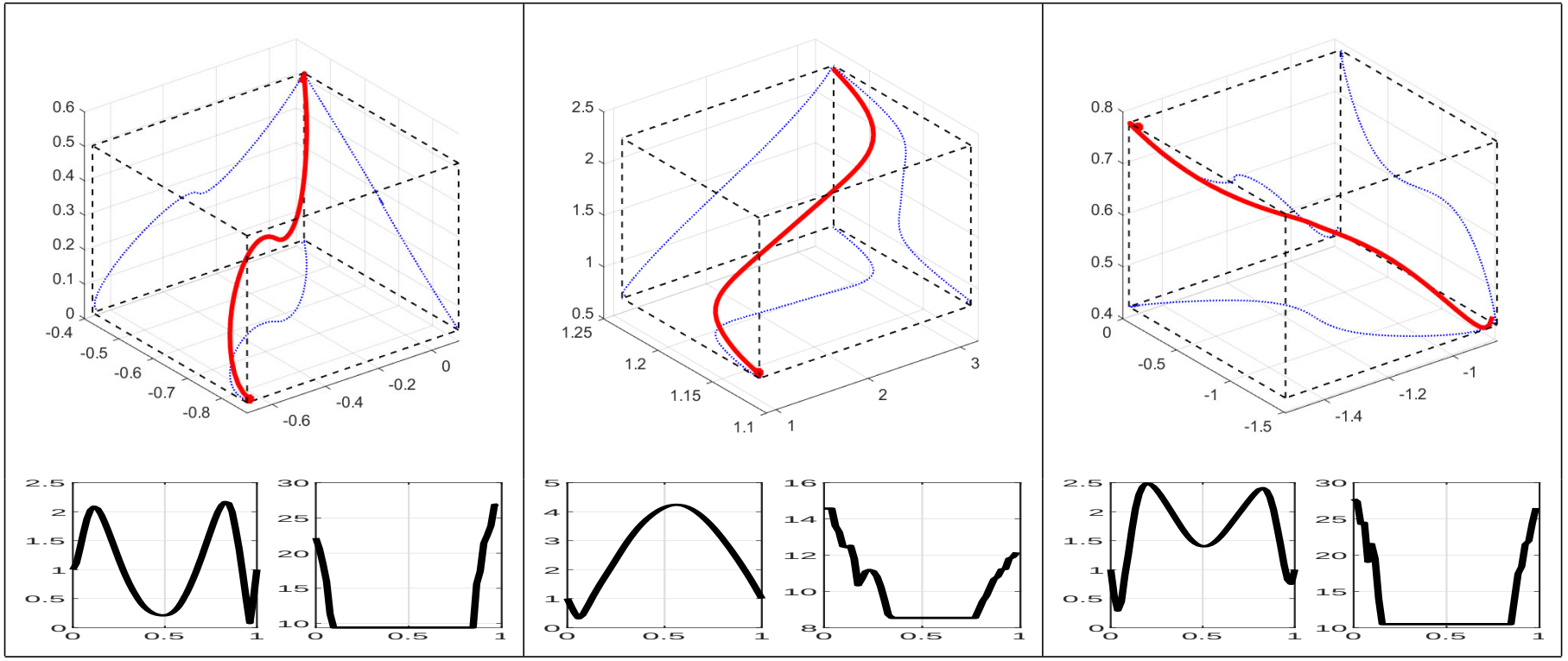
Note: We can add to (P) upper bounds on speed and (say) linear constraints on the allowed positions of the particle without violating problem's structure and computation-friendliness.

How it Works



- Top: trajectory and its 2D projections (dotted)
- Bottom: speed (left) and acceleration (right) vs time

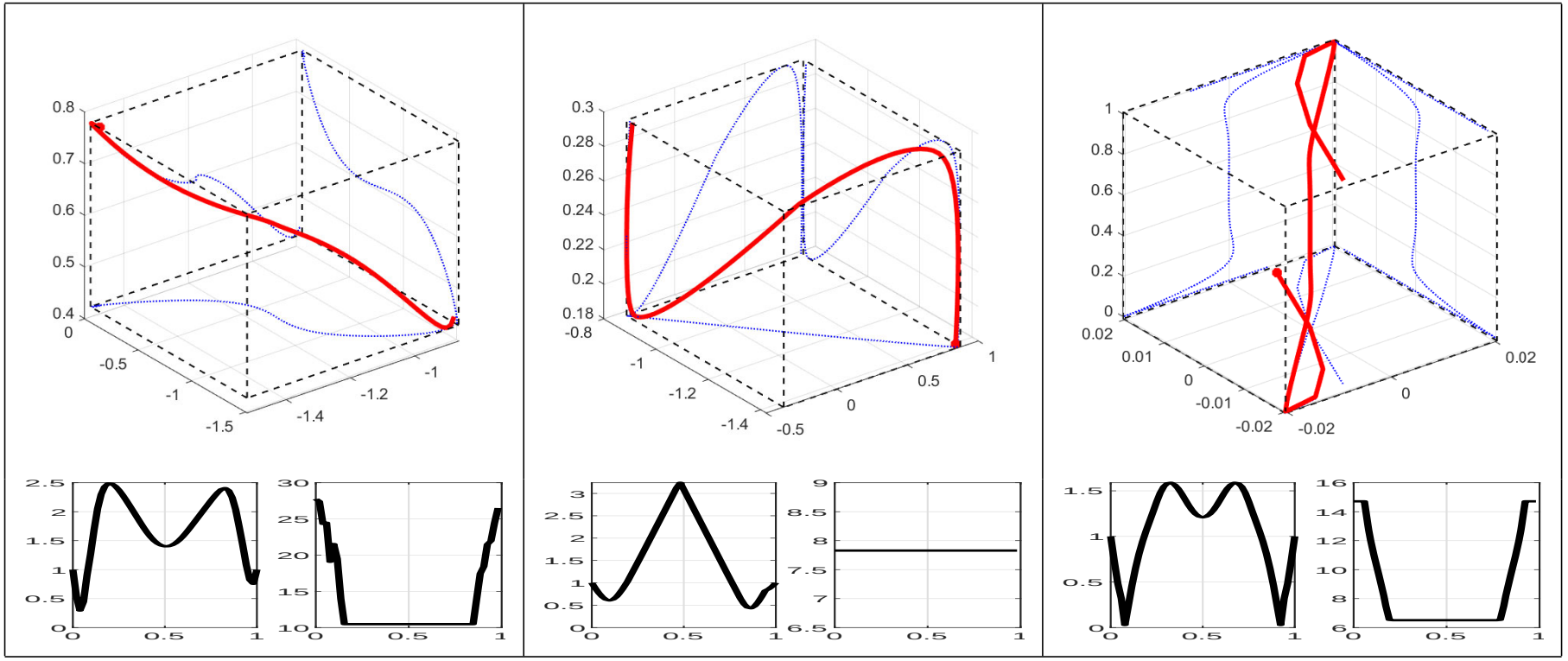
Maneuvers with "no-fly" zone



- Top: trajectory and its 2D projections (dotted)
- Bottom: speed (left) and acceleration (right) vs time

Outside of boxes: "no-fly" zones

Maneuvers with "no-fly" zone



- Top: trajectory and its 2D projections (dotted)
- Bottom: speed (left) and acceleration (right) vs time

Outside of boxes: "no-fly" zones

$$\min_{x_2,\ldots,x_{N-1},\theta} \left\{ \theta : \|x_{\tau+2} - 2x_{\tau+1} + x_\tau\|_2/dt^2 \le \theta,\ 0 \le \tau < N \right\} \qquad \left[\begin{array}{rcl} x_0 & = & x^0 \\ x_1 & = & x^0 + dt \cdot v^0 \\ x_N & = & x^1 \\ x_{N+1} & = & x^1 + dt \cdot v^1 \end{array} \right] \quad \text{(P)}$$

Quiz. Let $x_\tau \in \mathbb{R}^{2024}$. Can you reduce (P) to a similar problem with x's of smaller dimension? How small can you make this dimension?

$$\min_{x_2,\ldots,x_{N-1},\theta}\left\{\theta : \|x_{\tau+2}-2x_{\tau+1}+x_\tau\|_2/dt^2 \le \theta,\ 0\le\tau<N\right\} \quad \left[\begin{array}{rcl} x_0 &=& x^0 \\ x_1 &=& x^0+dt\cdot v^0 \\ x_N &=& x^1 \\ x_{N+1} &=& x^1+dt\cdot v^1 \end{array}\right] \quad \text{(P)}$$

Question. Let $x_\tau \in \mathbb{R}^{2024}$. Can you reduce (P) to a similar problem with x's of smaller dimension? How small can you make this dimension?

Answer. (P) *reduces to similar problem in dimension at most 3*.

Indeed, let our particle move through $\mathbb{R}^d$ from state x^0, v^0 to state x^1, v^1.

- Transformation $x \mapsto \underline{x} = U(x-x^0)$ with orthogonal U converts continuous time trajectory $x(t)$ obeying the original boundary conditions into the trajectory $\underline{x}(t) = U(x(t)-x^0)$ obeying boundary conditions

$$\underline{x}(0) = \underline{x}^0 := 0,\ \dot{\underline{x}}(0) = \underline{v}^0 := Uv^0,\ \underline{x}(1) = \underline{x}^1 := U(x^1-x^0),\ \dot{\underline{x}}(1) = \underline{v}^1 := Uv^1 \qquad (*)$$

and preserves the magnitude of acceleration, and similarly for discrete time trajectories.

$\Rightarrow$ *The original problem reduces to similar problem with boundary conditions* $(*)$.

- Let us select U to zero out all but the first $\overline{d} = \mathrm{Rank}\,([x^1-x^0, v^0, v^1])$ entries in the right hand sides of $(*)$. With this U, when zeroing out all but the first $\overline{d}$ entries in a trajectory obeying $(*)$, we preserve $(*)$ and can only decrease the magnitude of acceleration, and similarly for the discrete time trajectories.

$\Rightarrow$ *The "rotated" problem* (and therefore the original one) *reduces to similar problem in dimension* $\overline{d} \le 3$.

• So far, we measured magnitude of acceleration at time instant τ by the $\|\cdot\|_2$-norm of the acceleration vector $[x_{\tau+2}-2x_{\tau+1}+x_\tau]/dt^2$. We could use another norm, e.g., $\|\cdot\|_p$, resulting in the problem

$$\min_{x_2,\ldots,x_{N-1},\theta}\left\{\theta : \|x_{\tau+2}-2x_{\tau+1}+x_\tau\|_p/dt^2 \le \theta,\ 0\le\tau<N\right\} \qquad \left[\begin{array}{rcl} x_0 &=& x^0\\ x_1 &=& x^0+dt\cdot v^0\\ x_N &=& x^1\\ x_{N+1} &=& x^1+dt\cdot v^1\end{array}\right] \quad \text{(P)}$$

• For 3D "missile" (*Apologies to students from Mechanical/Aerospace Engineering, if any in the class*)

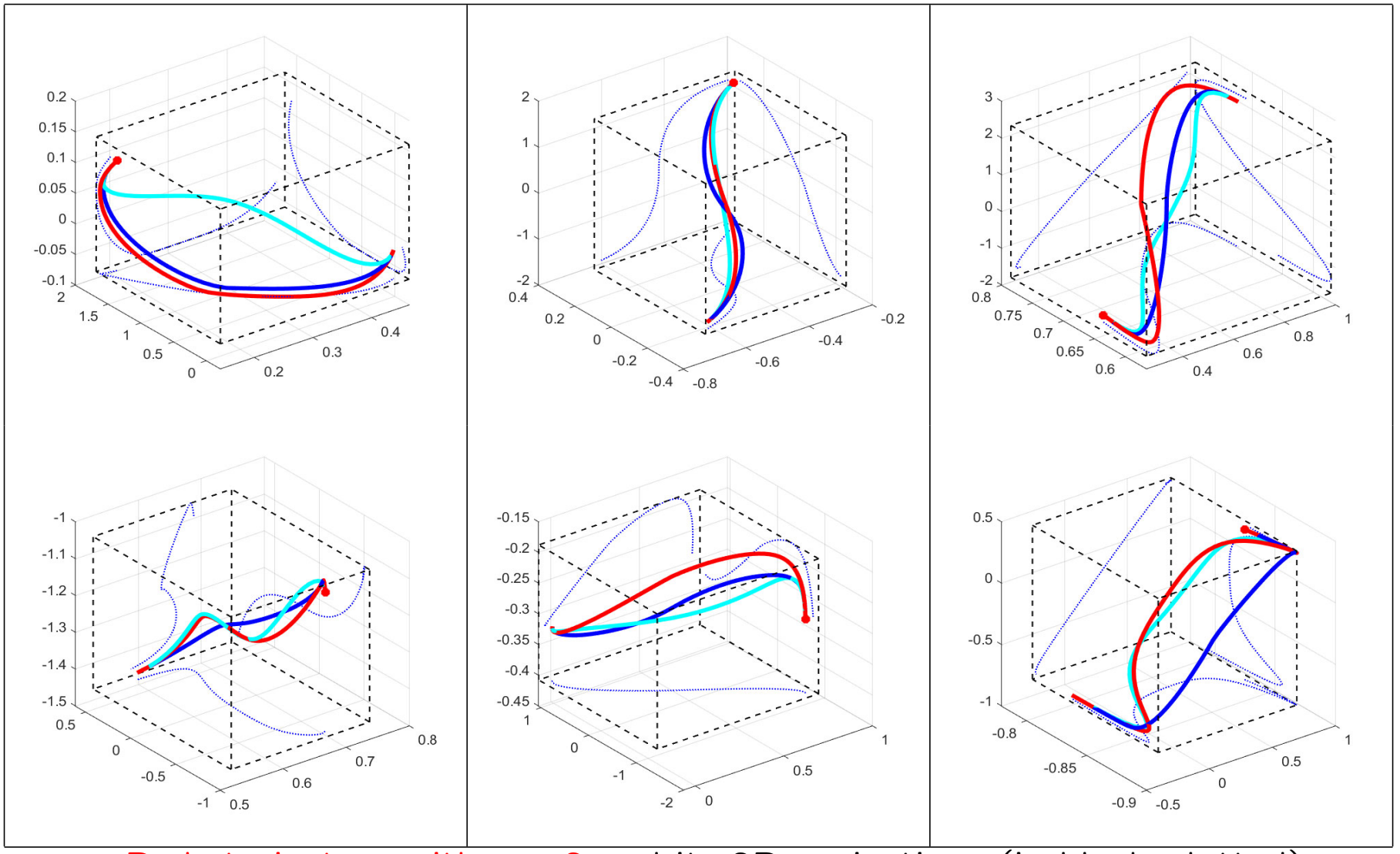
— $p=1$ corresponds to 6 engine nozzles aligned with coordinate rays; what matters, is the maximum, over time, instantaneous fuel consumption

— $p=\infty$ corresponds 6 engine nozzles aligned with the coordinate rays; what matters, is the maximum, over time and nozzles, pressure in the nozzle

— $p=2$ corresponds to the single rotating in 3D engine nozzle; what matters is the maximum, over time, instantaneous fuel consumption, a.k.a. pressure in the nozzle.

• When p differs from 1 and ∞, (P) is *not* an LP, albeit it can be rapidly approximated by LP to a whatever high accuracy. *When $p=1$ or $p=\infty$, the problem is an LP.*

Maneuvers with "no-fly" zone



- Red: trajectory with $p = 2$ and its 2D projections (in black, dotted)
- Blue: trajectory with $p = 1$ Cyan: trajectory with $p = \infty$

Outside of boxes: "no-fly" zones

LO Models in Engineering
B. LO Models in Signal Processing

♣ **Fitting Parameters in Linear Regression:** "In the nature" there exists a *linear dependence* between a variable vector of factors (*regressors*) $x \in \mathbb{R}^n$ and the "ideal output" $y \in \mathbb{R}$:

$$y = \theta_*^T x$$

$\theta_* \in \mathbb{R}^n$: vector of true parameters.
Given a collection $\{x_i, y_i\}_{i=1}^m$ of noisy observations of this dependence:

$$y_i = \theta_*^T x_i + \xi_i \qquad [\xi_i : \text{ observation noise}]$$

we want to recover the parameter vector θ_*.
♠ When $m \gg n$, a typical way to solve the problem is to choose a "discrepancy measure" $\phi(u, v)$ – a kind of distance between vectors $u, v \in \mathbb{R}^m$ – and to choose an estimate $\widehat{\theta}$ of θ_* by minimizing in θ the discrepancy

$$\phi\left([y_1; ...; y_m], [\theta^T x_1; ...; \theta^T x_m]\right)$$

between the observed outputs and the outputs of a hypothetic model $y = \theta^T x$ as applied to the observed regressors $x_1, ..., x_m$.

$$y_i = \theta_*^T x_i + \xi_i \qquad [\xi_i : \text{ observation noise}]$$

♠ Setting $X = [x_1^T; x_2^T; ...; x_m^T]$, the recovering routine becomes

$$\widehat{\theta} = \operatorname*{argmin}_{\theta} \phi(\mathbf{y}, X\theta) \qquad [\mathbf{y} = [y_1; ...; y_m]]$$

♠ The choice of $\phi(\cdot, \cdot)$ primarily depends on the type of observation noise.

• The most frequently used discrepancy measure is $\phi(u, v) = \|u - v\|_2$, corresponding to the case of White Gaussian Noise ($\xi_i \sim \mathcal{N}(0, \sigma^2)$ are independent) or, more generally, to the case when ξ_i are independent identically distributed with zero mean and finite variance. The corresponding *Least Squares* recovery

$$\widehat{\theta} = \operatorname*{argmin}_{\theta} \sum_{i=1}^{m} [y_i - x_i^T \theta]^2$$

reduces to solving a system of linear equations.

$$\widehat{\theta} = \operatorname{argmin}_{\theta} \phi(\mathbf{y}, X\theta) \qquad [\mathbf{y} = [y_1; ...; y_m]]$$

♠ There are cases when the recovery reduces to LO:

1. ℓ_1-fit: $\phi(u, v) = \|u - v\|_1 := \sum_{i=1}^m |u_i - v_i|$. Here the recovery problem reads

$$\min_{\theta} \sum_{i=1}^m |y_i - x_i^T\theta| \;\Leftrightarrow\; \min_{\theta,\tau} \left\{\tau : \sum_{i=1}^m |y_i - x_i^T\theta| \le \tau\right\} \qquad (!)$$

There are two ways to reduce (!) to an LO program:

- *Intelligent way:* Noting that $|y_i - x_i^T\theta| = \max[y_i - x_i^T\theta, x_i^T\theta - y_i]$, we can use the "eliminating piecewise linear nonlinearities" trick to convert (!) into the LO program

$$\min_{\theta,\tau,\tau_i} \left\{\tau : \begin{array}{l} y_i - x_i^T\theta \le \tau_i, x_i^T\theta - y_i \le \tau_i \; \forall i \\ \sum_{i=1}^m \tau_i \le \tau \end{array}\right\}$$

$1 + n + m$ variables, $2m + 1$ constraints

- *Stupid way:* Noting that

$$\sum_{i=1}^m |u_i| = \max_{\epsilon_1 = \pm 1, ..., \epsilon_m = \pm 1} \sum_i \epsilon_i u_i,$$

we can convert (!) into the LO program

$$\min_{\theta,\tau} \left\{\tau : \sum_{i=1}^m \epsilon_i [y_i - x_i^T\theta] \le \tau \; \forall \epsilon_1 = \pm 1, ..., \epsilon_m = \pm 1\right\}$$

$1 + n$ variables, 2^m constraints

$$\widehat{\theta} = \operatorname*{argmin}_{\theta} \phi(\mathbf{y}, X\theta) \qquad [\mathbf{y} = [y_1; ...; y_m]]$$

2. Uniform fit $\phi(u, v) = \|u - v\|_\infty := \max_i |u_i - v_i|$:

$$\min_{\theta,\tau} \left\{ \tau : y_i - x_i^T\theta \le \tau,\ x_i^T\theta - y_i \le \tau,\ 1 \le i \le m \right\}$$

Sparsity-oriented Signal Processing and ℓ_1 minimization

♣ **Compressed Sensing:** We have m-dimensional observation

$$y = [y_1; ...; y_m] = X\theta_* + \xi$$
$$[X \in \mathbb{R}^{m\times n}: \text{ sensing matrix}, \xi: \text{ observation noise}]$$

of unknown "signal" $\theta_* \in \mathbb{R}^n$ with $m \ll n$ and want to recover θ_*.

♠ Since $m \ll n$, the system $X\theta = y - \xi$ in variables θ, if solvable, has infinitely many solutions

$\Rightarrow$ Even in the noiseless case, θ_* cannot be recovered well, *unless additional information on θ_* is available.*

♠ In Compressed Sensing, the additional information on θ_* is that θ_* *is sparse* — *has at most a given number $s \ll m$ nonzero entries.*

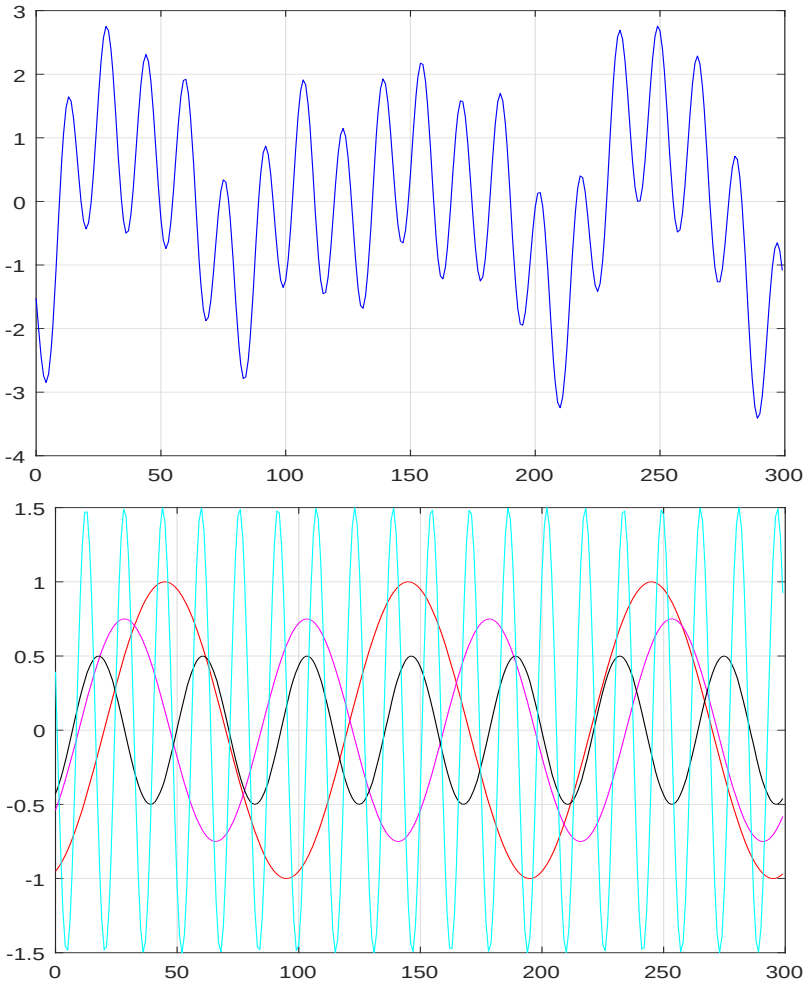
♠ **Fact:** *Many real-life signals x when presented by their coefficients in properly selected basis ("dictionary") B:*

$$x = Bu$$

- columns of B: vectors of basis B
- u: coefficients of x in basis B

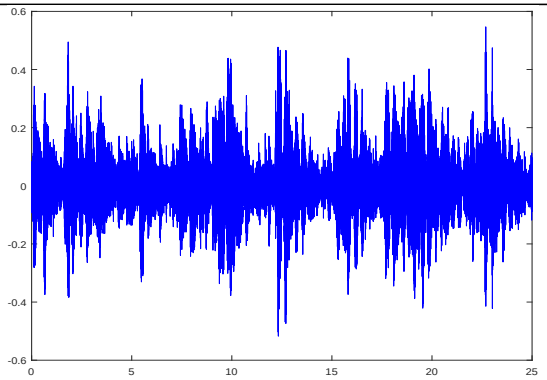
become sparse (or nearly so): u has just $s \ll n$ nonzero entries (or can be well approximated by vector with $s \ll n$ nonzero entries). We do *not* assume the location of "meaningful coefficients" known in advance.

Example I: Typical audio signals become sparse (or nearly so) when representing them "in frequency domain" – as sums of harmonic oscillations of different frequencies:

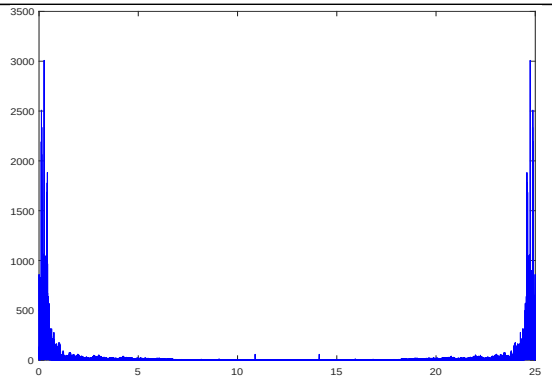


Top: singal in time domain
Bottom: decomposition of signal in harmonic oscillations

Illustration: 25 sec fragment of audio signal "Mail must go through" (dimension 1,058,400) and its "Fourier coefficients" – amplitudes of participating harmonic oscillations:



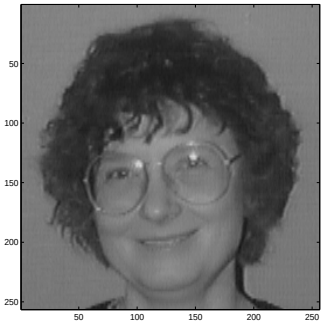
How mail goes through in time domain



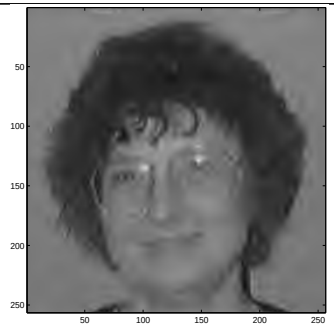
How mail goes through in frequency domain

% of leading Fourier coefficients kept	energy
100%	100%
25%	99.8%
15%	99.6%
5%	98.2%
1%	79.0%

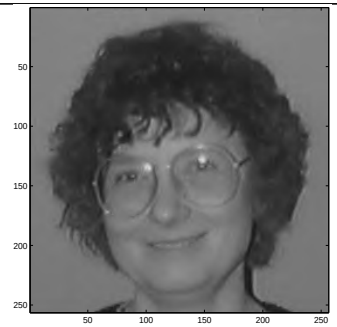
Example II: The 256×256 image



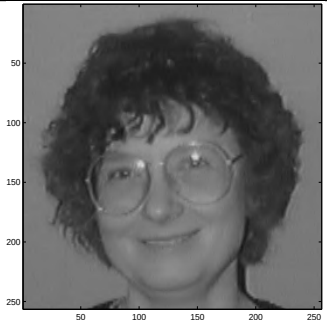
can be thought of as $256^2 = 65536$-dimensional vector (write down the intensities of pixels column by column). This image (same as other "non-pathological" images) is nearly sparse when represented in *wavelet* basis:



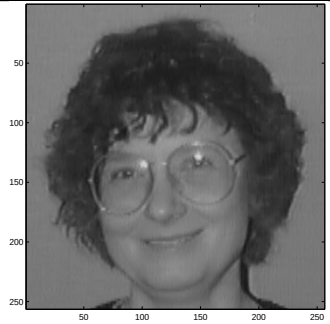
1% of leading wavelet coefficients kept (99.70% of energy)



5% of leading wavelet coefficients kept (99.93% of energy)



10% of leading wavelet coefficients kept (99.96% of energy)



25% of leading wavelet coefficients kept (99.99% of energy)

♠ When recovering a signal x_* admitting a sparse (or nearly so) representation Bu_* in a *known* basis B from observations

$$y = Ax_* + \eta,$$

the situation reduces to the one when the signal to be recovered is just sparse.

Indeed, we can first recover *sparse* u_* from observations

$$y = Ax_* + \eta = [AB]u_* + \eta.$$

After an estimate $\widehat{u}$ of u_* is built, we can estimate x_* by $B\widehat{u}$.

$\Rightarrow$ In fact, sparse recovery is about how to recover a *sparse* n-dimensional signal x from $m \ll n$ observations

$$y = Ax_* + \eta.$$

♣ **Observation:** *Assume that $\xi = 0$, and let every $m \times 2s$ submatrix of X be of rank $2s$* (which will be typically the case when $m \geq 2s$). *Then θ_* is the optimal solution to the optimization problem*

$$\min_\theta \{\mathrm{nnz}(\theta) : X\theta = y\}$$

$$[\mathrm{nnz}(\theta) = \mathrm{Card}\{j : \theta_j \neq 0\}]$$

♠ **Bad news:** Problem

$$\min_\theta \{\mathrm{nnz}(\theta) : X\theta = y\}$$

is heavily computationally intractable in the large-scale case.

Indeed, essentially the only known algorithm for solving the problem is a "brute force" one: *Look one by one at all finite subsets I of $\{1, ..., n\}$ of cardinality $0, 1, 2, ..., n$, trying each time to solve the linear system*

$$X\theta = y, \theta_i = 0,\ i \notin I$$

in variables θ. When the first solvable system is met, take its solution as $\widetilde{\theta}$.

- When $s = 5,\ n = 100$, the best known upper bound on the number of steps in this algorithm is $\approx 7.53\mathrm{e}7$, which perhaps is doable.
- When $s = 20,\ n = 200$, the bound blows up to $\approx 1.61\mathrm{e}27$, which is by many orders of magnitude beyond our "computational grasp."

♠ **Partial remedy:** Replace the difficult to minimize objective $\mathrm{nnz}(\theta)$ with an "easy-to-minimize" objective, specifically, $\|\theta\|_1 = \sum_i |\theta_i|$. As a result, we arrive at *ℓ_1-recovery*

$$\widehat{\theta} = \mathrm{argmin}_\theta \{\|\theta\|_1 : X\theta = y\}$$
$$\Leftrightarrow \min_{\theta,z} \left\{\sum_j z_j : X\theta = y, -z_j \le \theta_j \le z_j \, \forall j \le n\right\}.$$

♠ When the observation is noisy: $y = X\theta_* + \xi$ and we know an upper bound δ on a norm $\|\xi\|$ of the noise, the ℓ_1-recovery becomes

$$\widehat{\theta} = \mathrm{argmin}_\theta \{\|\theta\|_1 : \|X\theta - y\| \le \delta\}.$$

When $\|\cdot\|$ is $\|\cdot\|_\infty$, the latter problem is an LO program:

$$\min_{\theta,z} \left\{\sum_j z_j : \begin{array}{l} -\delta \le [X\theta - y]_i \le \delta \, \forall i \le m \\ -z_j \le \theta_j \le z_j \, \forall j \le n \end{array}\right\}.$$

♣ Compressed Sensing theory shows that *under some difficult to verify assumptions on* X which are satisfied with overwhelming probability when X is a large *randomly selected* matrix *ℓ_1-minimization recovers*

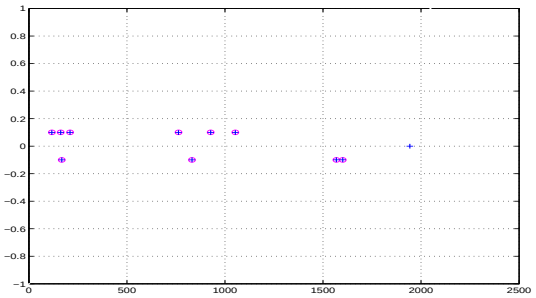
— *exactly in the noiseless case, and*

— *with error* $\leq C(X)\delta$ *in the noisy case*

all s*-sparse signals with* $s \leq O(1)m/\ln(n/m)$.

♠ **How it works:**

• X: 620×2048 • θ_*: 10 nonzeros • $\delta = 0.005$



ℓ_1-recovery, $\|\widehat{\theta} - \theta_*\|_\infty \leq$ 8.9e−4

♣ **Curious (and sad) fact:** Theory of Compressed Sensing states that "nearly all" large randomly generated $m \times n$ sensing matrices X are s-good with s as large as $O(1)\frac{m}{\ln(n/m)}$, meaning that for these matrices, ℓ_1-minimization in the noiseless case recovers *exactly* all s-sparse signals with the indicated value of s.

However: No individual sensing matrices with the outlined property are known. For all known $m \times n$ sensing matrices with $1 \ll m \ll n$, the provable level of goodness does not exceed $O(1)\sqrt{m}$... For example, for the 620×2048 matrix X from the above numerical illustration we have $m/\ln(n/m) \approx 518$ $\Rightarrow$ we could expect x to be s-good with s of order of hundreds. In fact we can certify

s-goodness of X with $s = 10$, and can certify that x is *not* s-good with $s = 59$.

Note: The best known *verifiable* sufficient condition for X to be s-good is

$$\min_Y \|\mathrm{Col}_j(I_n - Y^T X)\|_{s,1} < \frac{1}{2s}$$

- $\mathrm{Col}_j(A)$: j-th column of A
- $\|u\|_{s,1}$: the sum of s largest magnitudes of entries in u

This condition reduces to LO.

LO Models in Engineering
C. Synthesis of Linear Controllers

♣ Consider time-varying discrete time linear dynamical system

$$\begin{array}{rcll} x_0 & = & z & \text{[initial state]} \\ x_{t+1} & = & A_t x_t + B_t u_t + R_t d_t & \left[\begin{array}{l}\text{state equations}\\ \bullet\ x_t\text{: state}\quad \bullet\ u_t\text{: control}\\ \bullet\ d_t\text{: external disturbance}\end{array}\right] \\ y_t & = & C_t x_t + D_t d_t & \text{[observed output]} \end{array}$$

"closed" by *affine output-based control law*

$$u_t = g_t + \sum_{\tau=0}^{t} G_t^\tau y_\tau. \qquad (*)$$

♠ Given finite time horizon $0 \le t \le N$, we want to specify a control law $(*)$ which ensures that *the state-control trajectory* $w = [x_1; ...x_{N+1}; u_0; ...; u_N]$ *satisfies given design specifications*

$$a_i^T w \le b_i,\ 1 \le i \le I \qquad (!)$$

for all "perturbations" $\zeta = [z; d_0; ...; d_N]$ *from a given set* $\mathcal{Z}$ (equivalent wording: satisfies design specifications *robustly w.r.t.* $\zeta \in \mathcal{Z}$).

Good news: by linearity of the system and the control law, the trajectory is affine in ζ: $w = w_\gamma^0 + W_\gamma \zeta$, $\gamma = \{g_t, G_t^\tau : 0 \leq \tau \leq t \leq N\}$.

$\Rightarrow$ The *Analysis problem: check whether a given control law* $(*)$ *robustly meets the design specifications* reduces to verifying whether a system of affine constraints on ζ is satisfied by all $\zeta \in \mathcal{Z}$. This is easy, provided $\mathcal{Z}$ is "tractable."

•System:

$$\begin{array}{rcll} x_0 & = & z & \text{[initial state]} \\ x_{t+1} & = & A_t x_t + B_t u_t + R_t d_t & \left[\begin{array}{l}\text{state equations}\\ \bullet\ x_t\text{: state}\ \bullet\ u_t\text{: control}\\ \bullet\ d_t\text{: external disturbance}\end{array}\right] \\ y_t & = & C_t x_t + D_t d_t & \text{[observed output]} \end{array}$$

•Controller: $u_t = g_t + \sum_{\tau=0}^{t} G_t^\tau y_\tau$ $(*)$

•Trajectory: $w = [x_0; ...x_{N+1}; u_0; ...; u_N] = w_\gamma^0 + W_\gamma \zeta$ $[\gamma = \{g_t, G_t^\tau\}:$ control law$]$

•Design specifications: $a_i^T w \le b_i,\ 1 \le i \le I$ (!)

♠ From now on, assume that $\mathcal{Z}$ is given by polyhedral representation:

$$\mathcal{Z} = \{\zeta : \exists v : P\zeta + Qv \le r\}$$

Then to solve the *Analysis problem: given control law, check whether* $(*)$ *ensures* (!) *for all* $\zeta \in \mathcal{Z}$ is the same as to check whether

$$b_i \ge \max_{\zeta, v} \left\{ a_i^T [w_\gamma^0 + W_\gamma \zeta] : P\zeta + Qv \le r \right\},\ 1 \le i \le I.$$

⇒ *Verification requires solving* I *LO programs and is therefore easy.*

$$\begin{aligned} x_0 &= z \\ x_{t+1} &= A_t x_t + B_t u_t + R_t d_t \\ y_t &= C_t x_t + D_t d_t \end{aligned} \qquad (S)$$

$$u_t = g_t + \sum_{\tau=0}^{t} G_t^\tau y_\tau \qquad (*)$$

Bad news: the trajectory is highly nonlinear in the parameters $\gamma = \{g_t, G_t^\tau\}$ of the control law $(*)$. Indeed

- $x_0 = z$ is independent of $\gamma \Rightarrow y_0$ is independent of $\gamma \Rightarrow u_0$ is affine in $\gamma \Rightarrow x_1$ is affine in γ
- x_1 is affine in $\gamma \Rightarrow y_1$ is affine in $\gamma \Rightarrow u_1$ is quadratic in $\gamma \Rightarrow x_2$ is quadratic in γ
- x_2 is quadratic in $\gamma \Rightarrow y_2$ is quadratic in $\gamma \Rightarrow u_2$ is cubic in $\gamma \Rightarrow x_3$ is cubic in γ

  .....

  $\Rightarrow$ *x_k is polynomial of degree k in γ*

$\Rightarrow$ The *Synthesis problem: find control law $(*)$, if it exists, which robustly meets the design specifications* seems to be intractable.

$$\begin{array}{rcl} x_0 & = & z \\ x_{t+1} & = & A_t x_t + B_t u_t + R_t d_t \\ y_t & = & C_t x_t + D_t d_t \end{array} \qquad (S)$$

$$u_t = g_t + \sum_{\tau=0}^{t} G_t^\tau y_\tau \qquad (*)$$

Bad news: the trajectory is highly nonlinear in the parameters $\gamma = \{g_t, G_t^\tau\}$ of the control law $(*)$.

$\Rightarrow$ The *Synthesis problem: find control law* $(*)$*, if it exists, which robustly meets the design specifications* seems to be intractable.

Remedy: pass to affine *purified*-output-based control laws.

♠ Consider, along with system (S) "closed" by some control law, its *model*

$$\begin{array}{rcl} \widehat{x}_0 & = & 0 \\ \widehat{x}_{t+1} & = & A_t \widehat{x}_t + B_t u_t \\ \widehat{y}_t & = & C_t \widehat{x}_t \end{array} \qquad (M)$$

which we "feed" by the same controls u_t as (S). We can run the model in an on-line fashion, and thus at time t, before the decision on u_t should be made, we have at our disposal *purified output* $v_t = y_t - \widehat{y}_t$

Observation: *purified outputs are known in advance affine functions of* ζ *completely independent on the control law in use*

Indeed, setting $\Delta_t = x_t - \widehat{x}_t$, we clearly have

$$v_t = C_t \Delta_t + D_t d_t \text{ with } \Delta_{t+1} = A_t \Delta_t + R_t d_t,\ \Delta_0 = z,$$

System:		Model:	
$x_0 = z$		$\widehat{x}_0 = 0$	
$x_{t+1} = A_t x_t + B_t u_t + R_t d_t$	(S)	$\widehat{x}_{t+1} = A_t \widehat{x}_t + B_t u_t$	(M)
$y_t = C_t x_t + D_t d_t$		$\widehat{y}_t = C_t \widehat{x}_t$	
Purified outputs: $v_t = y_t - \widehat{y}_t$			
$u_t = \begin{cases} g_t + \sum_{\tau=0}^{t} G_t^\tau y_\tau & \text{[output-based affine law]} \\ h_t + \sum_{\tau=0}^{t} H_t^\tau v_\tau & \text{[purified-output-based affine law]} \end{cases}$			$(*)$ $(\#)$

Facts:

♡ *Affine purified-output-based and output-based controls laws are equivalent: every mapping $\zeta \to w$ which can be obtained when "closing" (S) by a law $(*)$, can be obtained by closing (S) by a law $(\#)$, and vice versa.*

♡ *When (S) is closed by an affine purified-output-based control law $(\#)$, the trajectory $w = W[\zeta, \eta]$ becomes bi-affine in ζ and in the parameters $\eta = \{h_t, H_t^\tau\}$ of the control law:*

$$w = w^0[\eta] + W[\eta]\zeta \text{ with } w^0[\eta],\ W[\eta] \textit{ affine in } \eta.$$

•... purified outputs v_t are known in advance linear functions of the external disturbances $[z; d_0; ...; d_N]$

$\Rightarrow$

•$u_t = h_t + \sum_{\tau=0}^{t} H_t^\tau v_t$ is bi-affine in $[z; d_0; ...; d_N]$ and in $\eta = \{h_t, H_t^\tau : 0 \leq \tau \leq t \leq N\}$

• By linearity of the system, the trajectory w is *linear* in the vector $[z; d_0; ...; d_N; u_0; ...; u_N]$, and with affine purified-output-based control, this vector is bi-affine in $[z; d_0; ...; d_N]$ and in η

$\Rightarrow$ *w is bi-affine in $[z; d_0; ...; d_N]$ and in η,*

as claimed.

The state-control trajectory of system "closed" with affine purified-output-based control law with parameters η is bi-affine in ζ and in η:

$$w = w^0[\eta] + W[\eta]\zeta \text{ with known affine } w^0[\cdot], W[\cdot]$$

What we want:

$$Aw \leq b \; \forall \zeta \in \mathcal{Z} = \{\zeta : \exists v : P\zeta + Qv \leq r\}$$

Facts (continued):

♡ *Sticking to purified-output-based control laws, the Synthesis problem*

Given design specifications $a_i^T w \leq b_i$, $i \leq I$, on the state-control trajectory, find a control law, if one exists, which meets these specifications robustly w.r.t. $\zeta = [z; d_0; ...; d_N] \in \mathcal{Z}$

becomes an infinite system of linear constraints on η:

$$a_i^T \left[w^0[\eta] + W[\eta]\zeta\right] \leq b_i \, \forall \zeta \in \mathcal{Z}, \, 1 \leq i \leq I.$$

which is fact is equivalent to an explicit finite "moderate size" system of linear constraints on ζ and additional variables.

Question: *What the infinite system of linear constraints on η:*

$$\forall(\zeta : \exists v : P\zeta + Qv \leq r) : a_i^T \left[w^0[\eta] + W[\eta]\zeta\right] \leq b_i,\ i \leq I$$

"wants" from η ?

Answer: *It wants the optimal values in I feasible parametric LP's:*

$$\begin{array}{rcl} \mathrm{Opt}_i[\eta] & = & \max\limits_{\zeta,v}\left\{a_i^T W[\eta]\zeta : P\zeta + Qv \leq r\right\} \\ & = & \min\limits_{y^i}\left\{r^T y^i : P^T y^i = W^T[\eta]a_i, Q^T y^i = 0, y^i \geq 0\right\} \end{array}$$

[by LP duality to be studied in details in the mean time]

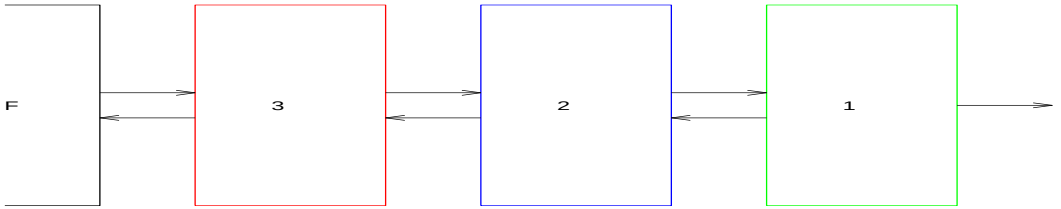
to satisfy the constraints $a_i^T w^0[\eta] + \mathrm{Opt}_i[\eta] \leq b_i$, $i \leq I$,

$\Rightarrow$ *the set of desirable η admits polyhedral representation*

$$\left\{\eta : \exists y^1, ..., y^I : \underbrace{\left\{\begin{array}{c} P^T y^i = W^T[\eta]a_i, Q^T y^i = 0, y^i \geq 0 \\ a_i^T w^0[\eta] + r^T y^i \leq b_i \end{array}\right\}}_{(H)}\right\}$$

Bottom line: *A purified-output-based affine control law with parameters η meets the design specifications $a_i^T w \leq b_i$, $1 \leq i \leq I$, robustly in $\zeta \in \mathcal{Z}$ iff η can be extended by properly chosen y^i, $i \leq I$, to a feasible solution of (H).*

How it Works: Controlling 3-Level Serial Inventory



3–LEVEL SERIAL INVENTORY

- Level 1 supplies external demand
- Level 2 supplies Level 1
- Level 3 supplies Level 2 and is supplied from Factory
- There is 2-period delay in executing replenishment orders

♠ **Normal operation:**

- Demand: 300 units per period
- Replenishment orders of every level: 300 units per period
- Inventory levels: 500 units every period at every level

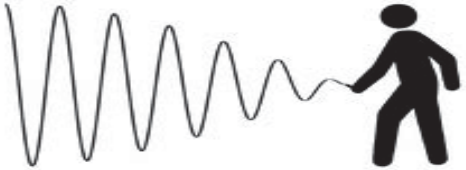
♠ **"In reality"** demands, orders, and inventory levels deviate from their "normal operation" values.

To save words, *in the sequel "demands," "orders," and "inventory levels" stand for the deviations of actual demands, orders, and inventory levels from their "normal operation" values.*

♠ On a close inspection, the Inventory can be modeled as the 9-state LDS

$$
\begin{array}{rcll}
x_1(t+1) & = & x_1(t) + x_{1,1}(t) & -d_t \\
x_{1,1}(t+1) & = & x_{1,2}(t) & \\
x_{1,2}(t+1) & = & u_1(t) & \\
x_2(t+1) & = & x_2(t) + x_{2,1}(t) & -u_1(t) \\
x_{2,1}(t+1) & = & x_{2,2}(t) & \\
x_{2,2}(t+1) & = & u_2(t) & \\
x_3(t+1) & = & x_3(t) + x_{3,1}(t) & -u_2(t) \\
x_{3,1}(t+1) & = & x_{3,2}(t) & \\
x_{3,2}(t+1) & = & u_3(t) & \\
\hline
y(t) & = & x(t) &
\end{array}
$$

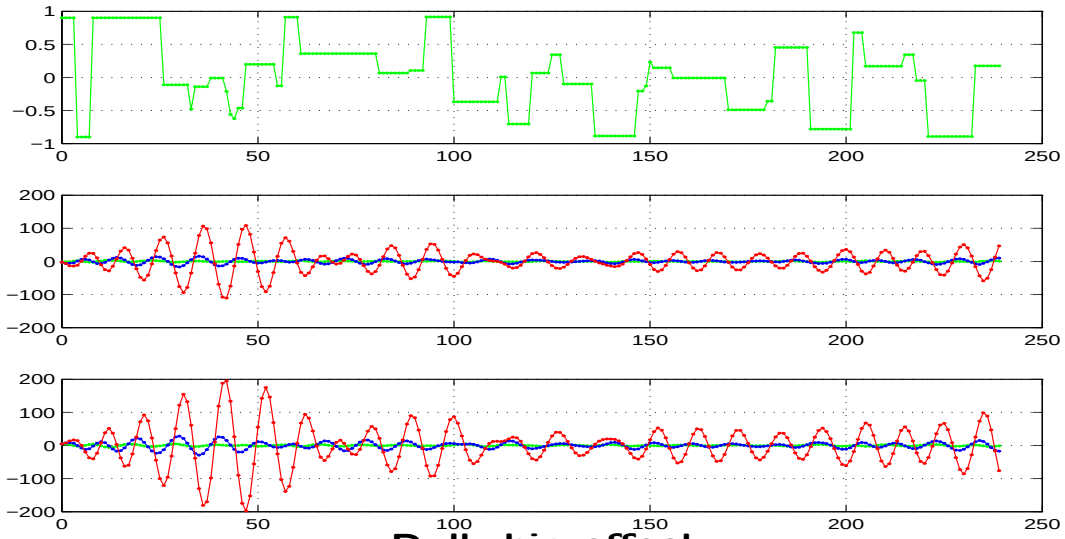
- $x_1(t)$, $x_2(t)$, $x_3(t)$ — inventory levels at the beginning of period t
- $u_1(t)$, $u_2(t)$, $u_3(t)$ — replenishment orders of period t
- $x_{p,1}(t) := u_p(t-2)$, $x_{p,2}(t) := u_p(t-1)$, $p = 1, 2, 3$
- d_t — demands



Bullwhip

♣ It is well known that serial inventories with delays (and supply chains in general) suffer from *bullwhip effect*: variations of states (e.g., inventory levels) are *severely amplified* when moving upward from external demand to production units along the supply chain. This phenomenon badly affects the production.

• This is what happens with "naive" linear controller:



Bullwhip effect

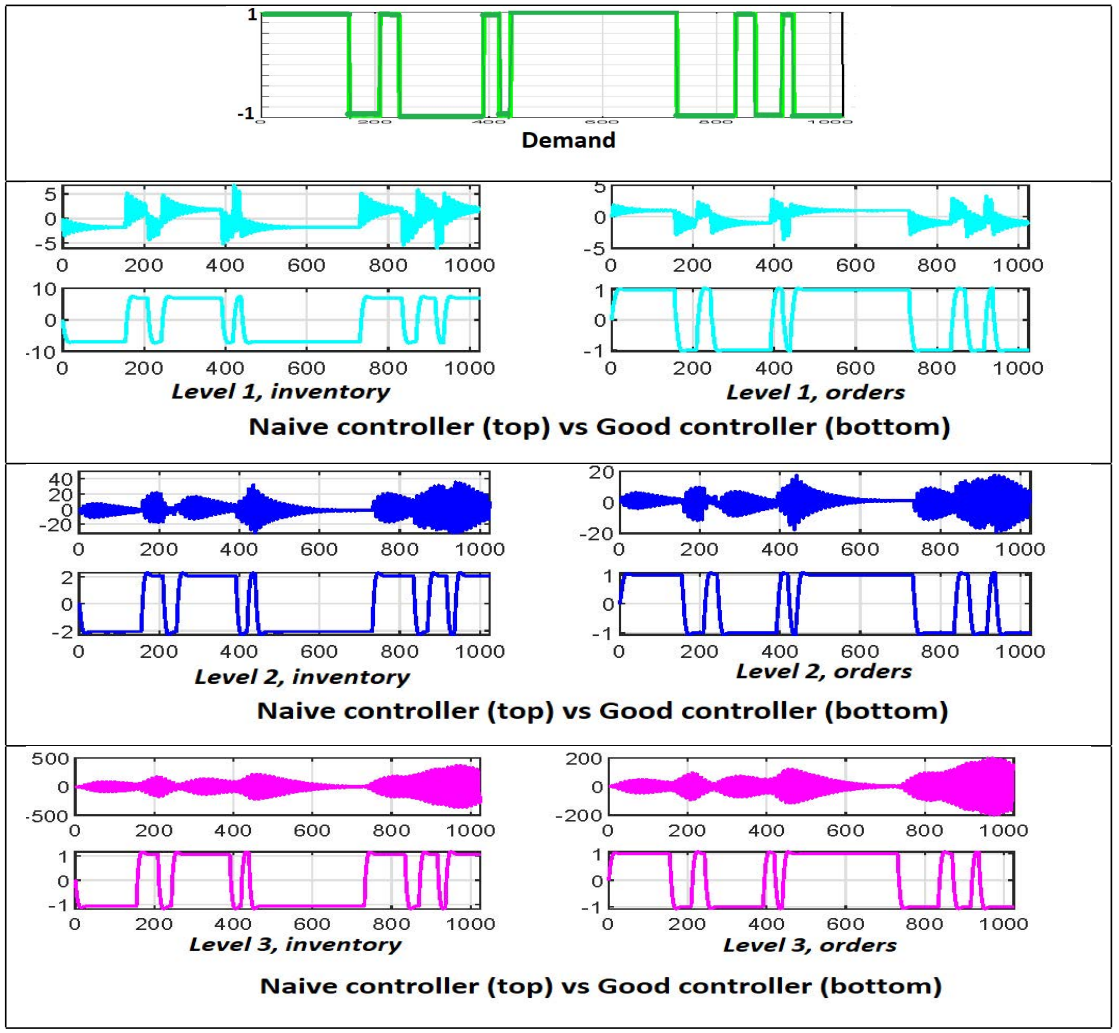
Top: time-dependent demand $d_t \in [-1, 1]$
Middle: replenishment orders $u_1(t)$, $u_2(t)$, $u_3(t) \in [-110, 110]$
Bottom: inventory levels $x_1(t)$, $x_2(t)$, $x_3(t) \in [-200, 200]$

Note: variations of the demand in the range $[-1, 1]$ result in huge (hundreds!) oscillations in the level #3 and in the replenishment orders.

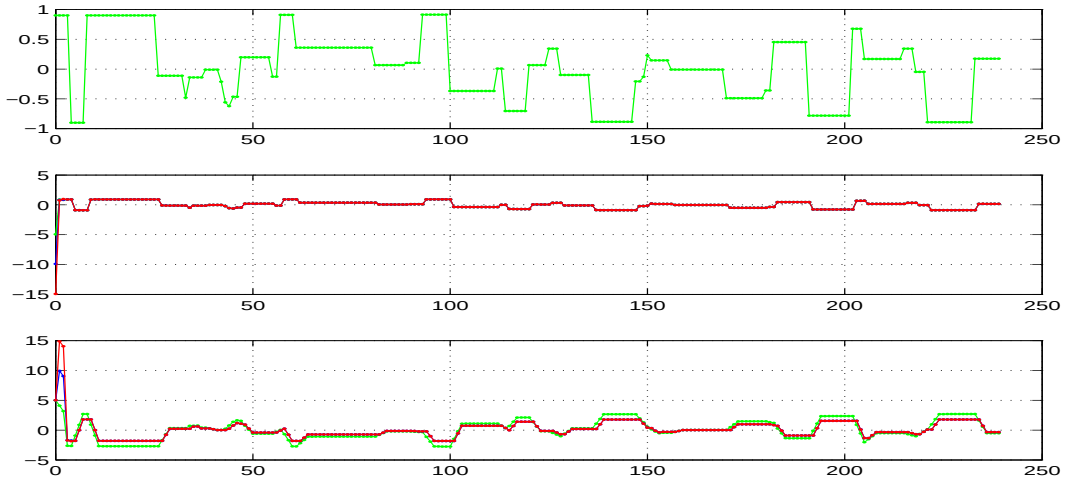
♡ To reduce the bullwhip effect, we can look for the best — with the largest decay rate as certified by Lyapunov Stability Certificate, whatever it means — linear feedback control law

$$u(t) = Ky(t) \ [= Kx(t)].$$

With this control, the picture looks much better:



Linear controller: Naive vs Good, long run

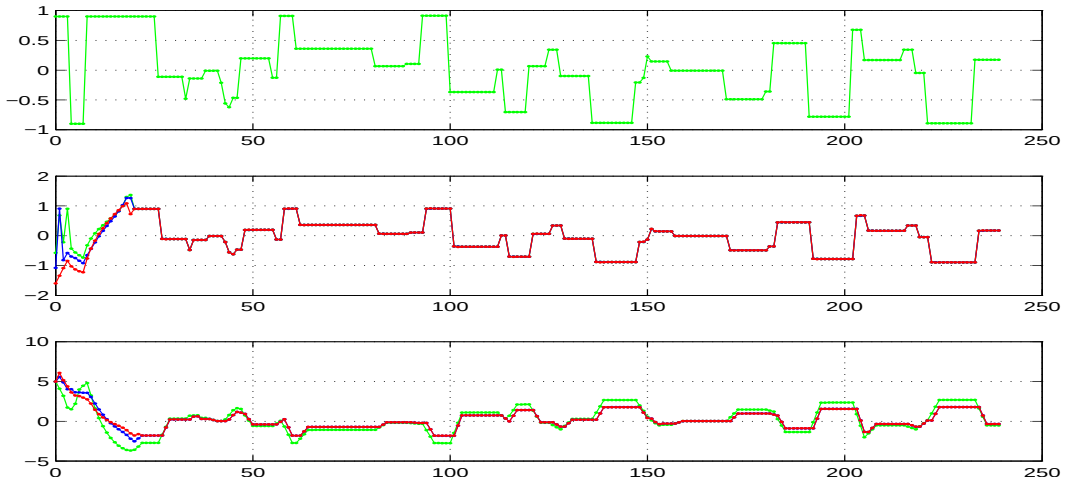


Good linear controller, short run

Top: time-dependent demand $d_t \in [-1, 1]$
Middle: replenishment orders $u_1(t)$, $u_2(t)$, $u_3(t) \in [-15, 5]$
Bottom: inventory levels $x_1(t)$, $x_2(t)$, $x_3(t) \in [-5, 15]$

But: At the very beginning, we still have unpleasant jumps in the inventory levels and replenishment orders.

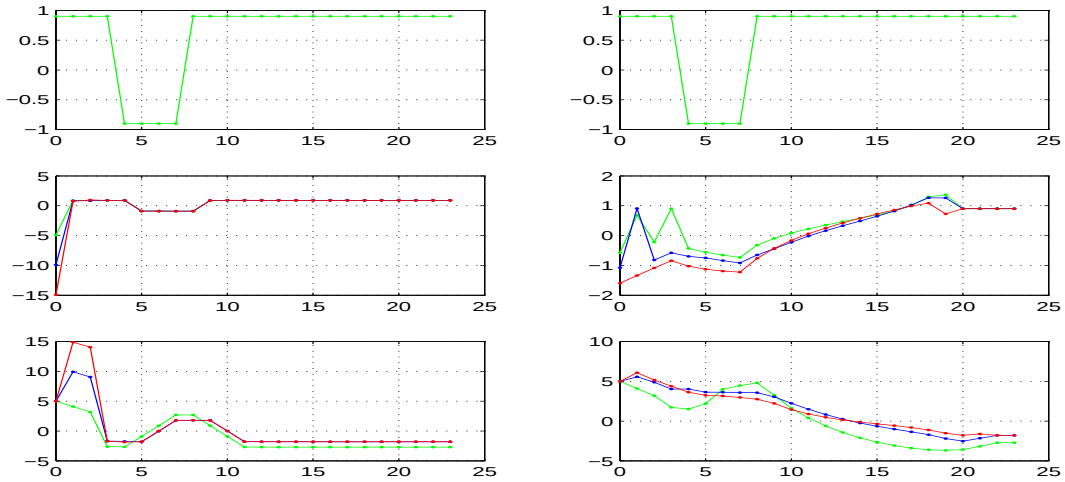
♡ To improve the behaviour of the process in the beginning, we can use purified-output-based affine control aimed at minimizing the initial jumps and eventually switching to the above feedback control. This is what we get:



Combined p.o.b./feedback control

Top: time-dependent demand $d_t \in [-1, 1]$
Middle: replenishment orders $u_1(t)$, $u_2(t)$, $u_3(t) \in [-1.5, 1.2]$
Bottom: inventory levels $x_1(t)$, $x_2(t)$, $x_3(t) \in [-4, 6]$

♡ This is what we gain in the beginning, while loosing nothing in the long run:



Pure feedback control (left)
vs.
combined p.o.b/feedback control (right)

Top: time-dependent demand $\in [-1, 1]$
Middle: replenishment orders $u_1(t)$, $u_2(t)$, $u_3(t)$
Bottom: inventory levels $x_1(t)$, $x_2(t)$, $x_3(t)$

Lecture I.2
What can be reduced to LO?
Polyhedral Representations

What Can Be Reduced to LO?

♣ We have seen numerous examples of optimization programs which can be reduced to LO, *although in its original "maiden" form the program is not an LO one.* Typical "maiden form" of a MP problem is

$$\text{(MP)}: \quad \max_{x \in X \subset \mathbb{R}^n} f(x)$$
$$X = \{x \in \mathbb{R}^n : g_i(x) \le 0,\ 1 \le i \le m\}$$

In LO,

- The objective is linear
- The constraints are affine

♠ **Observation:** Every MP program is equivalent to a program with linear objective.

Indeed, adding slack variable τ, we can rewrite (MP) equivalently as

$$\max_{y=[x;\tau] \in Y} c^T y := \tau,$$
$$Y = \{[x;\tau] : g_i(x) \le 0, \tau - f(x) \le 0\}$$

⇒ *we lose nothing when assuming from the very beginning that the objective in* (MP) *is linear:* $f(x) = c^T x$.

$$\text{(MP)}: \quad \max_{x \in X \subset \mathbb{R}^n} c^T x$$
$$X = \{x \in \mathbb{R}^n : g_i(x) \leq 0,\ 1 \leq i \leq m\}$$

♣ **Definition:** A polyhedral representation of a set $X \subset \mathbb{R}^n$ is a representation of X of the form:

$$X = \{x : \exists w : Px + Qw \leq r\},$$

that is, a representation of X as the a projection onto the space of x-variables of a polyhedral set $X^+ = \{[x; w] : Px + Qw \leq r\}$ in the space of x, w-variables.

♠ **Observation:** *Given a polyhedral representation of the feasible set X of* (MP)*, we can pose* (MP) *as the LO program*

$$\max_{[x;w]} \Big\{c^T x : Px + Qw \leq r\Big\}.$$

♠ Examples of polyhedral representations:

- The set $X = \{x \in \mathbb{R}^n : \sum_i |x_i| \leq 1\}$ admits the p.r.

$$X = \left\{ x \in \mathbb{R}^n : \exists w \in \mathbb{R}^n : \begin{array}{r} -w_i \leq x_i \leq w_i, \\ 1 \leq i \leq n, \\ \sum_i w_i \leq 1 \end{array} \right\}.$$

- The set

$$X = \left\{ x \in \mathbb{R}^6 : \max[x_1, x_2, x_3] + 2\max[x_4, x_5, x_6] \leq x_1 - x_6 + 5 \right\}$$

admits the p.r.

$$X = \left\{ x \in \mathbb{R}^6 : \exists w \in \mathbb{R}^2 : \begin{array}{l} x_1 \leq w_1, x_2 \leq w_1, x_3 \leq w_1 \\ x_4 \leq w_2, x_5 \leq w_2, x_6 \leq w_2 \\ w_1 + 2w_2 \leq x_1 - x_6 + 5 \end{array} \right\}.$$

Whether a Polyhedrally Represented Set is Polyhedral?

♣ **Question:** Let X be given by a polyhedral representation:

$$X = \{x \in \mathbb{R}^n : \exists w : Px + Qw \leq r\},$$

that is, as the *projection* of the solution set

$$Y = \{[x; w] : Px + Qw \leq r\} \qquad (*)$$

of a finite system of linear inequalities in variables x, w onto the space of x-variables.

Is it true that X is polyhedral, i.e., X is a solution set of finite system of linear inequalities *in variables x only?*

Theorem. *Every polyhedrally representable set is polyhedral.*

Proof is given by the *Fourier — Motzkin elimination scheme* which demonstrates that the projection of the set $(*)$ onto the space of x-variables is a polyhedral set.

$$Y = \{[x; w] : Px + Qw \leq r\}, \qquad (*)$$

Elimination step: eliminating a *single* slack variable. Given set $(*)$, assume that $w = [w_1; ...; w_m]$ is nonempty, and let Y^+ be the projection of Y on the space of variables $x, w_1, ..., w_{m-1}$:

$$Y^+ = \{[x; w_1; ...; w_{m-1}] : \exists w_m : Px + Qw \leq r\} \qquad (!)$$

Let us prove that Y^+ is polyhedral. Indeed, let us split the linear inequalities $p_i^T x + q_i^T w \leq r_i,\ 1 \leq i \leq I$, defining Y into three groups:

- **black** – the coefficient at w_m is 0
- **red** – the coefficient at w_m is > 0
- **green** – the coefficient at w_m is < 0

Then

$$\begin{aligned} Y = \{[x; w] :\ & \\ & a_i^T x + b_i^T [w_1; ...; w_{m-1}] \leq c_i,\ \ i \text{ is black} \\ & w_m \leq a_i^T x + b_i^T [w_1; ...; w_{m-1}] + c_i,\ \ i \text{ is red} \\ & w_m \geq a_i^T x + b_i^T [w_1; ...; w_{m-1}] + c_i,\ \ i \text{ is green}\} \end{aligned}$$

$\Rightarrow$

$$\begin{aligned} Y^+ = \{[x; w_1; ...; w_{m-1}] :\ & \\ & a_i^T x + b_i^T [w_1; ...; w_{m-1}] \leq c_i, i \text{ is black} \\ & a_\mu^T x + b_\mu^T [w_1; ...; w_{m-1}] + c_\mu \geq a_\nu^T x + b_\nu^T [w_1; ...; w_{m-1}] + c_\nu \\ & \qquad \text{whenever } \mu \text{ is red and } \nu \text{ is green}\} \end{aligned}$$

and thus Y^+ is polyhedral.

We have seen that the projection

$$Y^+ = \{[x; w_1; ...; w_{m-1}] : \exists w_m : [x; w_1; ...; w_m] \in Y\}$$

of the polyhedral set $Y = \{[x, w] : Px + Qw \leq r\}$ is polyhedral. Iterating the process, we conclude that the set $X = \{x : \exists w : [x, w] \in Y\}$ is polyhedral, Q.E.D.

♣ Given an LO program

$$\text{Opt} = \max_x \left\{c^T x : Ax \leq b\right\}, \qquad (!)$$

observe that the set of values of the objective at feasible solutions can be represented as

$$\begin{aligned} T &= \{\tau \in \mathbb{R} : \exists x : Ax \leq b, c^T x - \tau = 0\} \\ &= \{\tau \in \mathbb{R} : \exists x : Ax \leq b, c^T x \leq \tau, c^T x \geq \tau\} \end{aligned}$$

that is, *T is polyhedrally representable*. By Theorem, T is polyhedral, that is, T can be represented by a finite system of nonstrict linear inequalities *in variable τ only.* It immediately follows that *if T is nonempty and is bounded from above, T has the largest element*. Thus, we have proved

Corollary. *A feasible and bounded LO program admits an optimal solution and thus is solvable.*

$$\begin{aligned} T &= \{\tau \in \mathbb{R} : \exists x : Ax \leq b, c^T x - \tau = 0\} \\ &= \{\tau \in \mathbb{R} : \exists x : Ax \leq b, c^T x \leq \tau, c^T x \geq \tau\} \end{aligned}$$

♣ Fourier-Motzkin Elimination Scheme suggests a finite algorithm for solving an LO program, where we

• first, apply the scheme to get a representation of T by a finite system S of linear inequalities in variable τ,

• second, analyze S to find out whether the solution set is nonempty and bounded from above, and when it is the case, to find out the optimal value $\mathrm{Opt} \in T$ of the program,

• third, use the Fourier-Motzkin elimination scheme in the backward fashion to find x such that $Ax \leq b$ and $c^T x = \mathrm{Opt}$, thus recovering an optimal solution to the problem of interest.

Bad news: The resulting algorithm is completely impractical, since the number of inequalities we should handle at an elimination step usually rapidly grows with the step number and can become astronomically large when eliminating just tens of variables.

Polyhedrally Representable Functions

♣ **Definition:** *Let f be a real-valued function on a set $\mathrm{Dom} f \subset \mathbb{R}^n$. The epigraph of f is the set*

$$\mathrm{Epi}\{f\} = \{[x;\tau] \in \mathbb{R}^n \times \mathbb{R} : x \in \mathrm{Dom} f, \tau \geq f(x)\}.$$

A polyhedral representation of $\mathrm{Epi}\{f\}$ is called a polyhedral representation of f. Function f is called polyhedrally representable, if it admits a polyhedral representation.

♠ **Observation:** *A Lebesque set $\{x \in \mathrm{Dom} f : f(x) \leq a\}$ of a polyhedrally representable function is polyhedral, with a p.r. readily given by a p.r. of $\mathrm{Epi}\{f\}$:*

$$\mathrm{Epi}\{f\} = \{[x;\tau] : \exists w : Px + \tau p + Qw \leq r\} \Rightarrow$$
$$\left\{x : \begin{array}{c} x \in \mathrm{Dom} f \\ f(x) \leq a \end{array}\right\} = \{x : \exists w : Px + ap + Qw \leq r\}.$$

Examples: • The function $f(x) = \max\limits_{1\le i\le I}[\alpha_i^T x + \beta_i]$ is polyhedrally representable:

$$\mathrm{Epi}\{f\} = \{[x;\tau] : \alpha_i^T x + \beta_i - \tau \le 0,\ 1 \le i \le I\}.$$

• **Extension:** Let $D = \{x : Ax \le b\}$ be a polyhedral set in $\mathbb{R}^n$. A function f with the domain D given in D as $f(x) = \max\limits_{1\le i\le I}[\alpha_i^T x + \beta_i]$ is polyhedrally representable:

$$\mathrm{Epi}\{f\} = \{[x;\tau] : x \in D, \tau \ge \max_{1\le i\le I} \alpha_i^T x + \beta_i\} =$$
$$\{[x;\tau] : Ax \le b,\ \alpha_i^T x - \tau + \beta_i \le 0,\ 1 \le i \le I\}.$$

In fact, *every polyhedrally representable function f is of the form stated in Extension.*

Calculus of Polyhedral Representations

♣ In principle, speaking about polyhedral representations of sets and functions, we could restrict ourselves with representations which do not exploit slack variables, specifically,

- *for sets* — with representations of the form

$$X = \{x \in \mathbb{R}^n : Ax \leq b\};$$

- *for functions* — with representations of the form

$$\mathrm{Epi}\{f\} = \{[x;\tau] : Ax \leq b, \tau \geq \max_{1\leq i\leq I} \alpha_i^T x + \beta_i\}$$

♠ However, "general" – involving slack variables – polyhedral representations of sets and functions are much more flexible and can be much more "compact" that the straightforward – without slack variables – representations.

Examples:

- The function $f(x) = \|x\|_1 : \mathbb{R}^n \to \mathbb{R}$ admits the p.r.

$$\mathrm{Epi}\{f\} = \left\{ [x;\tau] : \exists w \in \mathbb{R}^n : \begin{array}{r} -w_i \le x_i \le w_i, \\ 1 \le i \le n \\ \sum_i w_i \le \tau \end{array} \right\}$$

which requires n slack variables and $2n+1$ linear inequality constraints. In contrast to this, the straightforward — without slack variables — representation of f

$$\mathrm{Epi}\{f\} = \left\{ [x;\tau] : \begin{array}{l} \sum_{i=1}^n \epsilon_i x_i \le \tau \\ \forall (\epsilon_1 = \pm 1, ..., \epsilon_n = \pm 1) \end{array} \right\}$$

requires 2^n inequality constraints.

- The set $X = \{x \in \mathbb{R}^n : \sum_{i=1}^n \max[x_i, 0] \le 1\}$ admits the p.r.

$$X = \{x \in \mathbb{R}^n : \exists w : 0 \le w,\ x_i \le w_i\, \forall i, \sum_i w_i \le 1\}$$

which requires n slack variables and $2n+1$ inequality constraints. Every straightforward — without slack variables — p.r. of X requires at least $2^n - 1$ constraints

$$\sum_{i \in I} x_i \le 1,\ \emptyset \ne I \subset \{1, ..., n\}$$

♣ Polyhedral representations admit a kind of simple and "fully algorithmic" calculus which, essentially, demonstrates that all *convexity-preserving* operations with polyhedral sets produce polyhedral results, and a p.r. of the result is readily given by p.r.'s of the operands.

♠ **Role of Convexity:** A set $X \subset \mathbb{R}^n$ is called *convex*, if whenever two points x, y belong to X, the entire segment $[x, y]$ linking these points belongs to X:

$$\forall (x, y \in x, \lambda \in [0,1]) : \\ x + \lambda(y - x) = (1 - \lambda)x + \lambda y \in X \ .$$

A function $f : \mathrm{Dom} f \to \mathbb{R}$ is called *convex*, if its epigraph $\mathrm{Epi}\{f\}$ is a convex set, or, equivalently, if

$$x, y \in \mathrm{Dom} f, \lambda \in [0,1] \\ \Rightarrow f((1 - \lambda)x + \lambda y) \leq (1 - \lambda) f(x) + \lambda f(y).$$

Fact: *A polyhedral set $X = \{x : Ax \leq b\}$ is convex. In particular, a polyhedrally representable function is convex.*

Indeed,

$$\begin{array}{rl} & Ax \leq b, Ay \leq b, \lambda \geq 0, 1-\lambda \geq 0 \\ \Rightarrow & \begin{array}{rcl} A(1-\lambda)x & \leq & (1-\lambda)b \\ A\lambda y & \leq & \lambda b \end{array} \\ \Rightarrow & A[(1-\lambda)x + \lambda y] \leq b \end{array}$$

Consequences:

- lack of convexity makes impossible polyhedral representation of a set/function,
- consequently, operations with functions/sets allowed by "calculus of polyhedral representability" we intend to develop should be convexity-preserving operations.

Calculus of Polyhedral Sets

♠ **Raw materials:** $X = \{x \in \mathbb{R}^n : a^T x \leq b\}$ (when $a \neq 0$, or, which is the same, the set is nonempty and differs from the entire space, such a set is called *half-space*)

♠ **Calculus rules:**

S.1. Taking finite intersections: *If the sets $X_i \subset \mathbb{R}^n$, $1 \leq i \leq k$, are polyhedral, so is their intersection, and a p.r. of the intersection is readily given by p.r.'s of the operands.*

Indeed, if

$$X_i = \{x \in \mathbb{R}^n : \exists w^i : P_i x + Q_i w^i \leq r_i\},\ i = 1, ..., k,$$

then

$$\bigcap_{i=1}^{k} X_i = \left\{ x : \exists w = [w^1; ...; w^k] : \begin{array}{r} P_i x + Q_i w^i \leq r_i, \\ 1 \leq i \leq k \end{array} \right\},$$

which is a polyhedral representation of $\bigcap\limits_i X_i$.

S.2. Taking direct products. Given k sets $X_i \subset \mathbb{R}^{n_i}$, their *direct product* $X_1 \times ... \times X_k$ is the set in $\mathbb{R}^{n_1+...+n_k}$ composed of all block-vectors $x = [x^1; ...; x^k]$ with blocks x^i belonging to X_i, $i = 1, ..., k$. E.g., the direct product of k segments $[-1, 1]$ on the axis is the unit k-dimensional box $\{x \in \mathbb{R}^k : -1 \leq x_i \leq 1, i = 1, ..., k\}$.

If the sets $X_i \subset \mathbb{R}^{n_i}$, $1 \leq i \leq k$, are polyhedral, so is their direct product, and a p.r. of the product is readily given by p.r.'s of the operands.

Indeed, if

$$X_i = \{x^i \in \mathbb{R}^{n_i} : \exists w^i : P_i x^i + Q_i w^i \leq r_i\}, \; i = 1, ..., k,$$

then

$$\begin{array}{l} X_1 \times ... \times X_k \\ = \Big\{x = [x^1; ...; x^k] : \exists w = [w^1; ...; w^k] : \\ \qquad\qquad\qquad\qquad P_i x^i + Q_i w^i \leq r_i, \\ \qquad\qquad\qquad\qquad 1 \leq i \leq k \Big\}. \end{array}$$

S.3. Taking affine image. *If $X \subset \mathbb{R}^n$ is a polyhedral set and $y = Ax + b : \mathbb{R}^n \to \mathbb{R}^m$ is an affine mapping, then the set*

$$Y = AX + b := \{y = Ax + b : x \in X\} \subset \mathbb{R}^m$$

is polyhedral, with p.r. readily given by the mapping and a p.r. of X.
Indeed, if $X = \{x : \exists w : Px + Qw \leq r\}$, then

$$\begin{aligned} Y &= \{y : \exists [x; w] : Px + Qw \leq r, y = Ax + b\} \\ &= \left\{ y : \exists [x; w] : \begin{array}{l} Px + Qw \leq r, \\ y - Ax \leq b, Ax - y \leq -b \end{array} \right\} \end{aligned}$$

Since Y admits a p.r., Y is polyhedral.

S.4. Taking inverse affine image. *If $X \subset \mathbb{R}^n$ is polyhedral, and $x = Ay + b : \mathbb{R}^m \to \mathbb{R}^n$ is an affine mapping, then the set*

$$Y = \{y \in \mathbb{R}^m : Ay + b \in X\} \subset \mathbb{R}^m$$

is polyhedral, with p.r. readily given by the mapping and a p.r. of X.

Indeed, if $X = \{x : \exists w : Px + Qw \leq r\}$, then

$$\begin{aligned} Y = & \{y : \exists w : P[Ay + b] + Qw \leq r\} \\ & = \{y : \exists w : [PA]y + Qw \leq r - Pb\}. \end{aligned}$$

S.5. Taking arithmetic sum: *If the sets $X_i \subset \mathbb{R}^n$, $1 \le i \le k$, are polyhedral, so is their arithmetic sum $X_1 + ... + X_k := \{x = x_1 + ... + x_k : x_i \in X_i, 1 \le i \le k\}$, and a p.r. of the sum is readily given by p.r.'s of the operands.*

Indeed, the arithmetic sum of $X_1, ..., X_k$ is the image of $X_1 \times ... \times X_k$ under the linear mapping $[x^1; ...; x^k] \mapsto x^1 + ... + x^k$, and both operations preserve polyhedrality. Here is an explicit p.r. for the sum:

if $X_i = \{x : \exists w^i : P_i x + Q_i w^i \le r_i\}$, $1 \le i \le k$, then

$$\begin{aligned} & X_1 + ... + X_k \\ &= \left\{ x : \exists x^1, ..., x^k, w^1, ..., w^k : \begin{array}{r} P_i x^i + Q_i w^i \le r_i, \\ 1 \le i \le k, \\ x = \sum_{i=1}^k x^i \end{array} \right\}, \end{aligned}$$

and it remains to replace the vector equality in the right hand side by a system of two opposite vector inequalities.

Calculus of Polyhedrally Representable Functions

♣ **Preliminaries:** Arithmetics of partially defined functions.

• a scalar function f of n variables is specified by indicating its *domain* $\mathrm{Dom} f$– the set where the function is well defined, and by the description of f as a real-valued function in the domain.

When speaking about convex functions f, *it is very convenient to think of f as of a function defined everywhere on $\mathbb{R}^n$ and taking real values in $\mathrm{Dom} f$ and the value $+\infty$ outside of $\mathrm{Dom} f$.*

With this convention, f becomes an everywhere defined function on $\mathbb{R}^n$ taking values in $\mathbb{R} \cup \{+\infty\}$, and $\mathrm{Dom} f$ becomes the set where f takes real values.

♠ In order to allow for basic operations with partially defined functions, like their addition or comparison, we augment our convention with the following agreements on the arithmetics of the "extended real axis" $\mathbb{R} \cup \{+\infty\}$:

- *Addition*: for a real a, $a + (+\infty) = (+\infty) + (+\infty) = +\infty$.
- *Multiplication by a nonnegative real λ:* $\lambda \cdot (+\infty) = +\infty$ when $\lambda > 0$, and $0 \cdot (+\infty) = 0$.
- *Comparison:* for a real a, $a < +\infty$ (and thus $a \leq +\infty$ as well), and of course $+\infty \leq +\infty$.

Note: Our arithmetic is incomplete — operations like $(+\infty) - (+\infty)$ and $(-1) \cdot (+\infty)$ remain undefined.

♠ **Raw materials:** $f(x) = a^T x + b$ (*affine functions*)

$$\mathrm{Epi}\{a^T x + b\} = \{[x;\tau] : a^T x + b - \tau \leq 0\}$$

♠ **Calculus rules:**

F.1. Taking linear combinations with positive coefficients. *If* $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ *are p.r.f.'s and* $\lambda_i > 0$, $1 \leq i \leq k$, *then* $f(x) = \sum_{i=1}^k \lambda_i f_i(x)$ *is a p.r.f., with a p.r. readily given by those of the operands.*

Indeed, if

$$\begin{aligned}&\{[x;\tau] : \tau \geq f_i(x)\}\\ &= \{[x;\tau] : \exists w^i : P_i x + \tau p_i + Q_i w^i \leq r_i\}, 1 \leq i \leq k,\end{aligned}$$

then

$$\begin{aligned}&\{[x;\tau] : \tau \geq \textstyle\sum_{i=1}^k \lambda_i f_i(x)\}\\ &= \left\{[x;\tau] : \exists t_1, ..., t_k : \begin{array}{l} t_i \geq f_i(x), 1 \leq i \leq k, \\ \sum_i \lambda_i t_i \leq \tau \end{array}\right\}\\ &= \left\{[x;\tau] : \exists t_1, ..., t_k, w^1, ..., w^k : \right.\\ &\qquad\qquad \left.\begin{array}{r} P_i x + t_i p_i + Q_i w^i \leq r_i, \\ 1 \leq i \leq k, \\ \sum_i \lambda_i t_i \leq \tau \end{array}\right\}.\end{aligned}$$

F.2. Direct summation. *If $f_i : \mathbb{R}^{n_i} \to \mathbb{R} \cup \{+\infty\}$, $1 \le i \le k$, are p.r.f.'s, then so is their direct sum*

$$f([x^1; ...; x^k]) = \sum_{i=1}^{k} f_i(x^i) : \mathbb{R}^{n_1+...+n_k} \to \mathbb{R} \cup \{+\infty\}$$

and a p.r. for this function is readily given by p.r.'s of the operands.

Indeed, if

$$\begin{array}{l}\{[x^i; \tau] : \tau \ge f_i(x^i)\} \\ = \{[x^i; \tau] : \exists w^i : P_i x^i + \tau p_i + Q_i w^i \le r_i\},\ 1 \le i \le k,\end{array}$$

then

$$\begin{array}{l}\{[x^1; ...; x^k; \tau] : \tau \ge \sum_{i=1}^k f_i(x^i)\} \\ = \left\{[x^1; ...; x^k; \tau] : \exists t_1, ..., t_k : \begin{array}{r} t_i \ge f_i(x^i), \\ 1 \le i \le k, \\ \sum_i t_i \le \tau \end{array}\right\} \\ = \left\{[x^1; ...; x^k; \tau] : \exists t_1, ..., t_k, w^1, ..., w^k : \right. \\ \qquad\qquad \left. \begin{array}{r} P_i x^i + t_i p_i + Q_i w^i \le r_i, \\ 1 \le i \le k, \\ \sum_i t_i \le \tau \end{array}\right\}.\end{array}$$

F.3. Taking maximum. *If $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are p.r.f.'s, so is their maximum $f(x) = \max[f_1(x), ..., f_k(x)]$, with a p.r. readily given by those of the operands.*

Indeed, if

$$\begin{aligned}&\{[x;\tau] : \tau \geq f_i(x)\}\\ &= \{[x;\tau] : \exists w^i : P_i x + \tau p_i + Q_i w^i \leq r_i\},\ 1 \leq i \leq k,\end{aligned}$$

then

$$\begin{aligned}&\{[x;\tau] : \tau \geq \max_i f_i(x)\}\\ &= \left\{[x;\tau] : \exists w^1, ..., w^k : \begin{array}{r} P_i x + \tau p_i + Q_i w^i \leq r_i,\\ 1 \leq i \leq k \end{array}\right\}.\end{aligned}$$

F.4. Affine substitution of argument. *If a function $f(x) : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a p.r.f. and $x = Ay + b : \mathbb{R}^m \to \mathbb{R}^n$ is an affine mapping, then the function $g(y) = f(Ay + b) : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ is a p.r.f., with a p.r. readily given by the mapping and a p.r. of f.*

Indeed, if

$$\begin{array}{l}\{[x;\tau] : \tau \geq f(x)\}\\ = \{[x;\tau] : \exists w : Px + \tau p + Qw \leq r\},\end{array}$$

then

$$\begin{array}{l}\{[y;\tau] : \tau \geq f(Ay + b)\}\\ = \{[y;\tau] : \exists w : P[Ay + b] + \tau p + Qw \leq r\}\\ = \{[y;\tau] : \exists w : [PA]y + \tau p + Qw \leq r - Pb\}.\end{array}$$

F.5. Theorem on superposition. *Let*

- $f_i(x) : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ *be p.r.f.'s, and let*
- $F(y) : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ *be a p.r.f. which is nondecreasing w.r.t. every one of the variables* $y_1, ..., y_m$. *Then the superposition*

$$g(x) = \begin{cases} F(f_1(x), ..., f_m(x)), & f_i(x) < +\infty \, \forall i \\ +\infty, & \text{otherwise} \end{cases}$$

of F *and* $f_1, ..., f_m$ *is a p.r.f., with a p.r. readily given by those of* f_i *and* F.

Indeed, let

$$\{[x;\tau] : \tau \geq f_i(x)\} = \{[x;\tau] : \exists w^i : P_i x + \tau p + Q_i w^i \leq r_i\},$$
$$\{[y;\tau] : \tau \geq F(y)\} = \{[y;\tau] : \exists w : Py + \tau p + Qw \leq r\}.$$

Then

$$\{[x;\tau] : \tau \geq g(x)\} \underbrace{=}_{(*)} \left\{[x;\tau] : \exists y_1, ..., y_m : \begin{array}{l} y_i \geq f_i(x), 1 \leq i \leq m, \\ F(y_1, ..., y_m) \leq \tau \end{array}\right\}$$

$$= \left\{[x;\tau] : \exists y, w^1, ..., w^m, w : \begin{array}{l} P_i x + y_i p_i + Q_i w^i \leq r_i, 1 \leq i \leq m, \\ Py + \tau p + Qw \leq r \end{array}\right\},$$

where $(*)$ is due to the monotonicity of F.

Note: if some of f_i, say, $f_1, ..., f_k$, are affine, then the Superposition Theorem remains valid when we require the monotonicity of F w.r.t. the variables $y_{k+1}, ..., y_m$ only; a p.r. of the superposition in this case reads

$$\{[x;\tau] : \tau \geq g(x)\} = \left\{[x;\tau] : \exists y_{k+1}..., y_m : \begin{array}{l} y_i \geq f_i(x),\ k+1 \leq i \leq m, \\ F(f_1(x), ..., f_k(x), y_{k+1}, ..., y_m) \leq \tau \end{array}\right\}$$

$$= \left\{[x;\tau] : \exists y_1, ..., y_m, w^{k+1}, ..., w^m, w : \begin{array}{r} y_i = f_i(x),\ 1 \leq i \leq k, \\ P_i x + y_i p_i + Q_i w^i \leq r_i, \\ k+1 \leq i \leq m, \\ Py + \tau p + Qw \leq r \end{array}\right\},$$

and the linear equalities $y_i = f_i(x)$, $1 \leq i \leq k$, can be replaced by pairs of opposite linear inequalities.

Fast Polyhedral Approximation of the Second Order Cone

♠ **Fact:** The canonical polyhedral representation $X = \{x \in \mathbb{R}^n : Ax \leq b\}$ of the projection

$$X = \{x : \exists w : Px + Qw \leq r\}$$

of a polyhedral set $X^+ = \{[x; w] : Px + Qw \leq r\}$ given by a moderate number of linear inequalities in variables x, w can require a huge number of linear inequalities in variables x.

Question: Can we use this phenomenon in order to *approximate* to high accuracy a non-polyhedral set $X \subset \mathbb{R}^n$ by projecting onto $\mathbb{R}^n$ a higher-dimensional *polyhedral and simple* (given by a moderate number of linear inequalities) set X^+ ?

Theorem: *For every n and every ϵ, $0 < \epsilon < 1/2$, one can point out a polyhedral set L^+ given by an explicit system of homogeneous linear inequalities in variables $x \in \mathbb{R}^n$, $t \in \mathbb{R}$, $w \in \mathbb{R}^k$:*

$$X^+ = \{[x; t; w] : Px + tp + Qw \leq 0\} \qquad (!)$$

such that

- *the number of inequalities in the system ($\approx 2n\ln(1/\epsilon)$) and the dimension of the slack vector w ($\approx 0.7n\ln(1/\epsilon)$) do not exceed $O(1)n\ln(1/\epsilon)$*
- *the projection*

$$L = \{[x; t] : \exists w : Px + tp + Qw \leq 0\}$$

of L^+ on the space of x, t-variables is in-between the Second Order Cone and $(1+\epsilon)$-extension of this cone:

$$\begin{array}{c}\mathbf{L}^{n+1} := \{[x; t] \in \mathbb{R}^{n+1} : \|x\|_2 \leq t\} \subset L \\ \subset \mathbf{L}_\epsilon^{n+1} := \{[x; t] \in \mathbb{R}^{n+1} : \|x\|_2 \leq (1+\epsilon)t\}.\end{array}$$

In particular, we have

$$\begin{array}{c}B_n^1 \subset \{x : \exists w : Px + p + Qw \leq 0\} \subset B_n^{1+\epsilon} \\ B_n^r = \{x \in \mathbb{R}^n : \|x\|_2 \leq r\}\end{array}$$

Note: When $\epsilon = 1.e\text{-}17$, a usual computer does not distinguish between $r = 1$ and $r = 1 + \epsilon$. Thus, *for all practical purposes*, the n-dimensional Euclidean ball admits polyhedral representation with $\approx 28n$ slack variables and $\approx 79n$ linear inequality constraints.

Note: A straightforward representation $X = \{x : Ax \leq b\}$ of a polyhedral set X satisfying

$$B_n^1 \subset X \subset B_n^{1+\epsilon}$$

requires at least $N = O(1)\epsilon^{-\frac{n-1}{2}}$ linear inequalities. With $n = 100$, $\epsilon = 0.01$, we get

$$N \geq 3.0\text{e}85 \approx 300,000 \times [\#\text{ of atoms in universe}]$$

With "fast polyhedral approximation" of B_n^1, a 0.01-approximation of B_{100} requires just 922 linear inequalities on 100 original and 325 slack variables.

♣ With fast polyhedral approximation of the cone $\mathbf{L}^{n+1} = \{[x;t] \in \mathbb{R}^{n+1} : \|x\|_2 \le t\}$, *Conic Quadratic* Optimization programs

$$\max_x \left\{ c^T x : \|A_i x - b_i\|_2 \le c_i^T x + d_i,\ 1 \le i \le m \right\} \qquad \text{(CQI)}$$

"for all practical purposes" become LO programs. Note that numerous highly nonlinear optimization problems, like

$$\begin{array}{c}
\text{minimize } c^T x \text{ subject to} \\
Ax = b \\
x \ge 0 \\
\left(\sum_{i=1}^{8} |x_i|^3\right)^{1/3} \le x_2^{1/7} x_3^{2/7} x_4^{3/7} + 2 x_1^{1/5} x_5^{2/5} x_6^{1/5} \\
5x_2 \ge \frac{1}{x_1^{1/2} x_2^2} + \frac{2}{x_2^{1/3} x_3^3 x_4^{5/8}} \\
\begin{bmatrix} x_2 & x_1 & & \\ x_1 & x_4 & x_3 & \\ & x_3 & x_6 & x_3 \\ & & x_3 & x_8 \end{bmatrix} \succeq 5I \\
\exp\{x_1\} + 2\exp\{2x_2 - x_3 + 4x_4\} \\
\qquad +3\exp\{x_5 + x_6 + x_7 + x_8\} \le 12
\end{array}$$

can be *in a systematic fashion* converted to/rapidly approximated by problems of the form (CQI) and thus "for all practical purposes" are just LO programs.

Lecture I.3
Geometry of Polyhedral Sets

Geometry of a Polyhedral Set

♣ An LO program $\max\limits_{x\in\mathbb{R}^n}\left\{c^Tx : Ax\le b\right\}$ is the problem of maximizing a linear objective over a *polyhedral set* $X=\{x\in\mathbb{R}^n : Ax\le b\}$ – the solution set of a *finite* system of *nonstrict linear* inequalities

⇒ *Understanding geometry of polyhedral sets is the key to LO theory and algorithms.*

♣ Our ultimate goal is to establish the following fundamental

Theorem. *A nonempty polyhedral set*

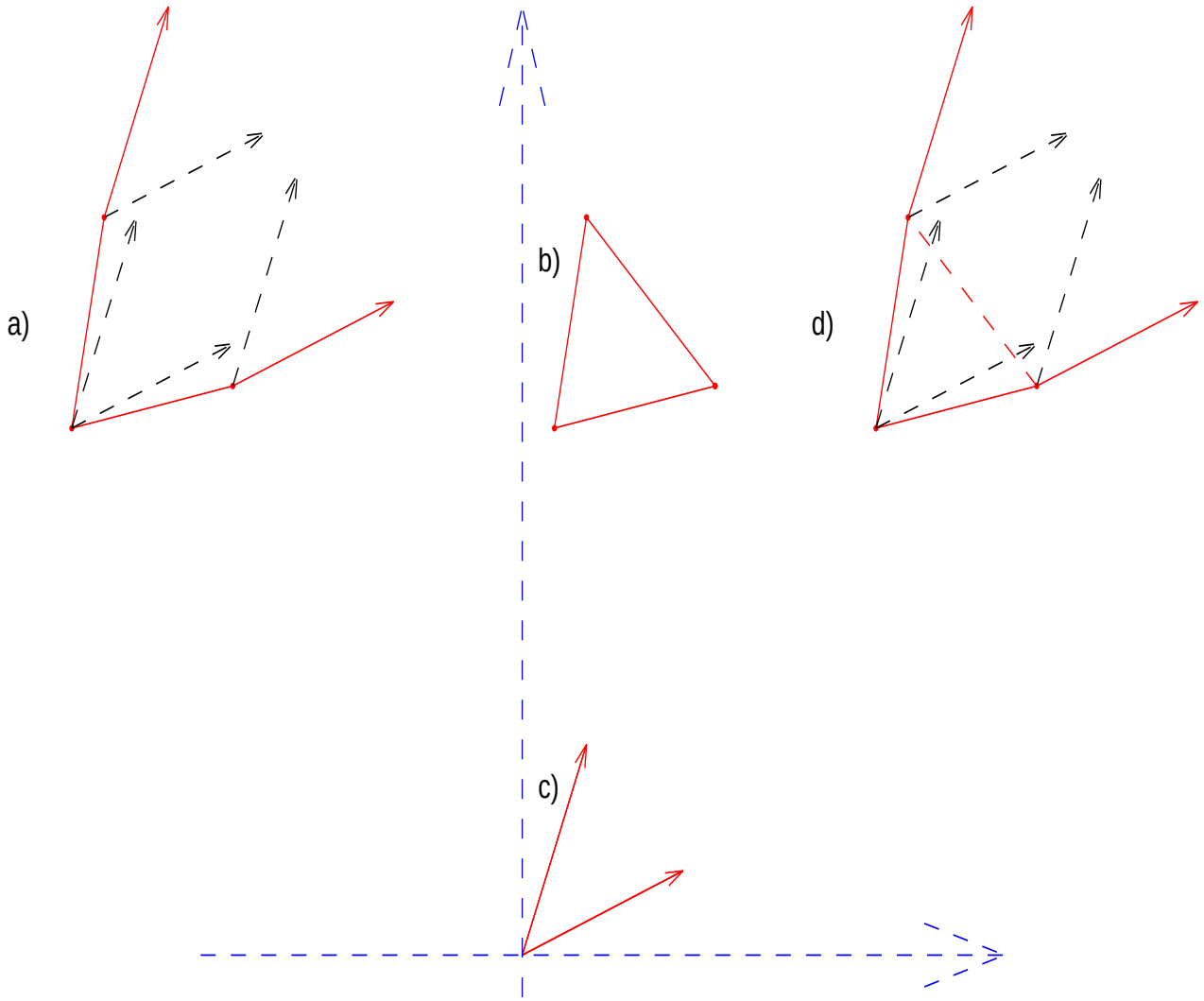
$$X=\{x\in\mathbb{R}^n : Ax\le b\}$$

admits a representation of the form

$$X=\left\{x=\sum_{i=1}^M\lambda_iv_i+\sum_{j=1}^N\mu_jr_j : \begin{array}{l}\lambda_i\ge 0\,\forall i\\ \sum\limits_{i=1}^M\lambda_i=1\\ \mu_j\ge 0\,\forall j\end{array}\right\} \qquad (!)$$

where $v_i\in\mathbb{R}^n$, $1\le i\le M$ *and* $r_j\in\mathbb{R}^n$, $1\le j\le N$ *are properly chosen "generators."*

Vice versa, every set X *representable in the form of* (!) *is polyhedral.*



a): a polyhedral set

b): $\{\sum_{i=1}^{3} \lambda_i v_i : \lambda_i \geq 0, \sum_{i=1}^{3} \lambda_i = 1\}$

c): $\{\sum_{j=1}^{2} \mu_j r_j : \mu_j \geq 0\}$

d): The set a) is the sum of sets b) and c)

Note: shown are the boundaries of the sets.

$$\emptyset \neq X = \{x \in \mathbb{R}^n : Ax \leq b\}$$
$$\Updownarrow$$
$$X = \left\{ x = \sum_{i=1}^{M} \lambda_i v_i + \sum_{j=1}^{N} \mu_j r_j : \begin{array}{l} \lambda_i \geq 0 \, \forall i \\ \sum_{i=1}^{M} \lambda_i = 1 \\ \mu_j \geq 0 \, \forall j \end{array} \right\} \quad (!)$$

♠ $X = \{x \in \mathbb{R}^n : Ax \leq b\}$ is an "outer" description of a polyhedral set X: it says what should be cut off $\mathbb{R}^n$ to get X.

♠ (!) is an "inner" description of a polyhedral set X: it explains how can we get all points of X, starting with two finite sets of vectors in $\mathbb{R}^n$.

♡ Taken together, these two descriptions offer a powerful "toolbox" for investigating polyhedral sets. For example,

• To see that the intersection of two polyhedral subsets X, Y in $\mathbb{R}^n$ is polyhedral, we can use their outer descriptions:

$$X = \{x : Ax \leq b\}, Y = \{x : Bx \leq c\}$$
$$\Rightarrow X \cap Y = \{x : Ax \leq b, Bx \leq c\} \quad .$$

$$\emptyset \neq X = \{x \in \mathbb{R}^n : Ax \leq b\}$$
$$\Updownarrow$$
$$X = \left\{ \sum_{i=1}^{M} \lambda_i v_i + \sum_{j=1}^{N} \mu_j r_j : \begin{array}{l} \lambda_i \geq 0 \, \forall i \\ \sum_{i=1}^{M} \lambda_i = 1 \\ \mu_j \geq 0 \, \forall j \end{array} \right\} \text{ (!)}$$

• To see that the image $Y = \{y = Px + p : x \in X\}$ of a polyhedral set $X \subset \mathbb{R}^n$ under an affine mapping $x \mapsto Px + p : \mathbb{R}^n \to \mathbb{R}^m$ is polyhedral, we can use the inner descriptions:

$$X \text{ is given by (!)}$$
$$\Rightarrow Y = \left\{ \sum_{i=1}^{M} \lambda_i (Pv_i + p) + \sum_{j=1}^{N} \mu_j P r_j : \begin{array}{l} \lambda_i \geq 0 \, \forall i \\ \sum_{i=1}^{M} \lambda_i = 1 \\ \mu_j \geq 0 \, \forall j \end{array} \right\}$$

Preliminaries: Linear Subspaces

♣ **Definition:** *A linear subspace in $\mathbb{R}^n$ is a nonempty subset L of $\mathbb{R}^n$ which is closed w.r.t. taking linear combinations of its elements:*

$$x_i \in L, \lambda_i \in \mathbb{R}, 1 \le i \le I \Rightarrow \sum_{i=1}^{I} \lambda_i x_i \in L$$

♣ **Examples:**

- $L = \mathbb{R}^n$
- $L = \{0\}$
- $L = \{x \in \mathbb{R}^n : x_1 = 0\}$
- $L = \{x \in \mathbb{R}^n : Ax = 0\}$
- Given a set $X \subset \mathbb{R}^n$, let $\mathrm{Lin}(X)$ be set of all finite linear combinations of vectors from X. This set – *the linear span of X* – is a linear subspace which contains X, and this is the intersection of all linear subspaces containing X.

Convention: A sum of vectors from $\mathbb{R}^n$ with empty set of terms is well defined and is the zero vector. In particular, $\mathrm{Lin}(\emptyset) = \{0\}$.

♠ **Note:** The last two examples are "universal:" *Every linear subspace L in $\mathbb{R}^n$ can be represented as $L = \mathrm{Lin}(X)$ for a properly chosen finite set $X \subset \mathbb{R}^n$, same as can be represented as $L = \{x : Ax = 0\}$ for a properly chosen matrix A.*

♣ **Dimension of a linear subspace.** Let L be a linear subspace in $\mathbb{R}^n$.

♠ For properly chosen $x_1, ..., x_m$, we have

$$L = \mathrm{Lin}(\{x_1, ..., x_m\}) = \left\{\sum_{i=1}^m \lambda_i x_i\right\};$$

whenever this is the case, we say that $x_1, ..., x_m$ *linearly span* L.

♠ **Facts:**

♡ *All minimal w.r.t. inclusion collections* $x_1, ..., x_m$ *linearly spanning* L (they are called *bases* of L) *have the same cardinality* m, called the *dimension* $\dim L$ of L.

♡ *Vectors* $x_1, ..., x_m$ *forming a basis of* L *always are linearly independent,* that is, every nontrivial (not all coefficients are zero) linear combination of the vectors is a nonzero vector.

♠ **Facts:**

♡ *All collections* $x_1, ..., x_m$ *of linearly independent vectors from* L *which are maximal w.r.t. inclusion* (i.e., extending the collection by any vector *from* L, we get a linearly dependent collection) *have the same cardinality, namely,* $\dim L$*, and are bases of* L*.*

♡ *Let* $x_1, ..., x_m$ *be vectors from* L*. Then the following four properties are equivalent:*

- $x_1, ..., x_m$ *is a basis of* L
- $m = \dim L$ *and* $x_1, ..., x_m$ *linearly span* L
- $m = \dim L$ *and* $x_1, ..., x_m$ *are linearly independent*
- $x_1, ..., x_m$ *are linearly independent and linearly span* L

♠ **Examples**:

• $\dim\{0\} = 0$, and the only basis of $\{0\}$ is the empty collection.

• $\dim \mathbb{R}^n = n$. When $n > 0$, there are infinitely many bases in $\mathbb{R}^n$, e.g., one composed of *standard basic orths* $e_i = [0; ...; 0; 1; 0; ...; 0]$ ("1" in i-th position), $1 \leq i \leq n$.

• $L = \{x \in \mathbb{R}^n : x_1 = 0\} \Rightarrow \dim L = n - 1\}$. An example of a basis in L is $e_2, e_3, ..., e_n$.

Facts: ♡ *if* $L \subset L'$ *are linear subspaces in* $\mathbb{R}^n$, *then* $\dim L \leq \dim L'$, *with equality taking place is and only if* $L = L'$.

⇒ *Whenever* L *is a linear subspace in* $\mathbb{R}^n$, *we have* $\{0\} \subset L \subset \mathbb{R}^n$, *whence* $0 \leq \dim L \leq n$

♡ *In every representation of a linear subspace as* $L = \{x \in \mathbb{R}^n : Ax = 0\}$, *the number of rows in* A *is at least* $n - \dim L$. *This number is equal to* $n - \dim L$ *if and only if the rows of* A *are linearly independent.*

"Calculus" of linear subspaces

♡ [taking intersection] When L_1, L_2 are linear subspaces in $\mathbb{R}^n$, so is the set $L_1 \cap L_2$.

Extension: The intersection $\bigcap_{\alpha\in\mathcal{A}} L_\alpha$ of an arbitrary family $\{L_\alpha\}_{\alpha\in\mathcal{A}}$ of linear subspaces of $\mathbb{R}^n$ is a linear subspace.

♡ [summation] When L_1, L_2 are linear subspaces in $\mathbb{R}^n$, so is their *arithmetic sum*

$$L_1 + L_2 = \{x = u + v : u \in L_1, v \in L_2\}.$$

Note "dimension formula:"

$$\dim L_1 + \dim L_2 = \dim (L_1 + L_2) + \dim (L_1 \cap L_2)$$

♡ [taking orthogonal complement] When L is a linear subspace in $\mathbb{R}^n$, so is its *orthogonal complement* $L^\perp = \{y \in \mathbb{R}^n : y^T x = 0 \, \forall x \in L\}$.

Note:

- $(L^\perp)^\perp = L$
- $L + L^\perp = \mathbb{R}^n$, $L \cap L^\perp = \{0\}$, whence $\dim L + \dim L^\perp = n$
- $L = \{x : Ax = 0\}$ if and only if the (transposes of) the rows in A linearly span $L^\perp$
- $x \in \mathbb{R}^n \Rightarrow \exists!(x_1 \in L, x_2 \in L^\perp) : x = x_1 + x_2$, and for these x_1, x_2 one has $x^T x = x_1^T x_1 + x_2^T x_2$.

♡ [taking direct product] When $L_1 \subset \mathbb{R}^{n_1}$ and $L_2 \subset \mathbb{R}^{n_2}$ are linear subspaces, the *direct product* (or *direct sum*) of L_1 and L_2 – the set

$$L_1 \times L_2 := \{[x_1; x_2] \in \mathbb{R}^{n_1+n_2} : x_1 \in L_1, x_2 \in L_2\}$$

is a linear subspace in $\mathbb{R}^{n_1+n_2}$, and

$$\dim(L_1 \times L_2) = \dim L_1 + \dim L_2 \quad .$$

♡ [taking image under linear mapping] When L is a linear subspace in $\mathbb{R}^n$ and $x \mapsto Px : \mathbb{R}^n \to \mathbb{R}^m$ is a linear mapping, the image $PL = \{y = Px : x \in L\}$ of L under the mapping is a linear subspace in $\mathbb{R}^m$.

♡ [taking inverse image under linear mapping] When L is a linear subspace in $\mathbb{R}^n$ and $x \mapsto Px : \mathbb{R}^m \to \mathbb{R}^n$ is a linear mapping, the inverse image $P^{-1}(L) = \{y : Py \in L\}$ of L under the mapping is a linear subspace in $\mathbb{R}^m$.

Preliminaries: Affine Subspaces

♣ **Definition:** *An affine subspace* (or *affine plane*, or simply *plane*) *in* $\mathbb{R}^n$ *is a nonempty subset* M *of* $\mathbb{R}^n$ *which can be obtained from a linear subspace* $L \subset \mathbb{R}^n$ *by a shift:*

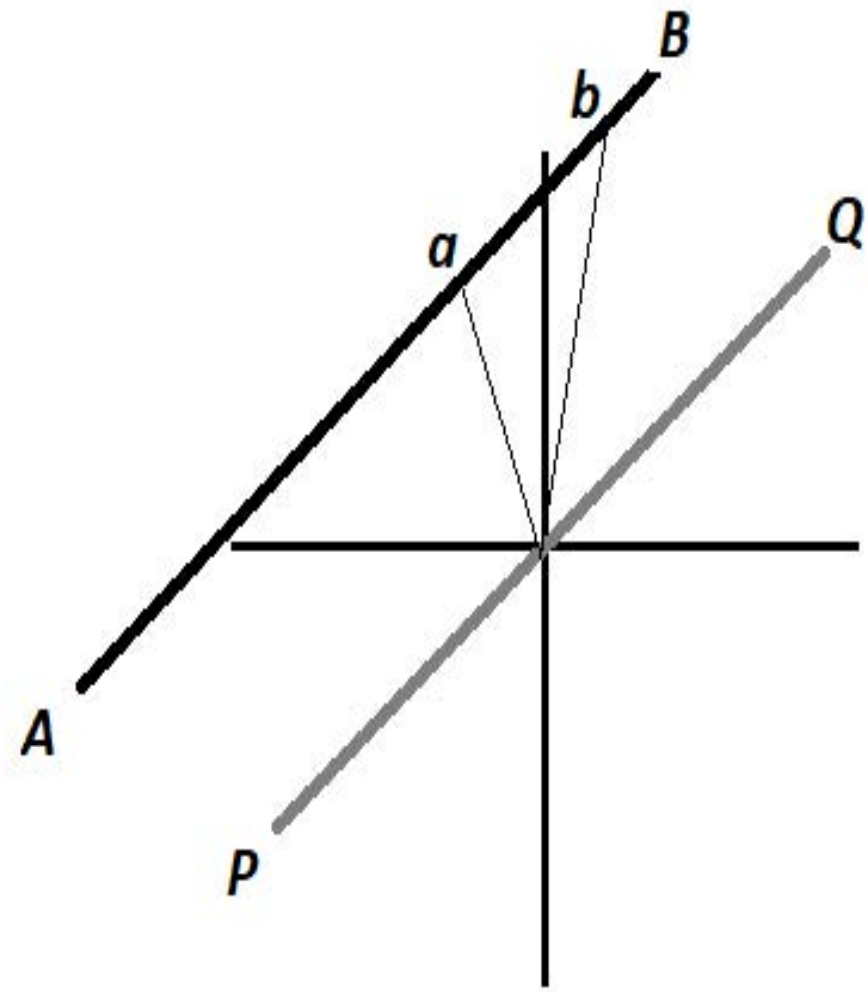
$$M = a + L = \{x = a + y : y \in L\} \qquad (*)$$

Note: *In a representation* $(*)$,

- L *us uniquely defined by* M*:* $L = M - M = \{x = u - v : u, v \in M\}$. L *is called the linear subspace which is parallel to* M*;*
- a *can be chosen as an arbitrary element of* M*, and only as an element from* M*.*

♠ **Equivalently:** *An affine subspace in* $\mathbb{R}^n$ *is a nonempty subset* M *of* $\mathbb{R}^n$ *which is closed with respect to taking affine combinations* (linear combinations with coefficients summing up to 1) *of its elements:*

$$\left(x_i \in M, \lambda_i \in \mathbb{R}, \sum_{i=1}^{I} \lambda_i = 1 \right) \Rightarrow \sum_{i=1}^{I} \lambda_i x_i \in M$$



$\mathbf{AB}$: affine plane (line) in $\mathbb{R}^2$
$\mathbf{PQ}$: parallel linear subspace
$\mathbf{a}, \mathbf{b}$: possible shift vectors

♣ **Examples:**

- $M = \mathbb{R}^n$. The parallel linear subspace is $\mathbb{R}^n$
- $M = \{a\}$ (singleton). The parallel linear subspace is $\{0\}$
- $M = \{a + \lambda \underbrace{[b-a]}_{\neq 0} : \lambda \in \mathbb{R}\} = \{(1-\lambda)a + \lambda b : \lambda \in \mathbb{R}\}$ – (straight) *line* passing through two distinct points $a, b \in \mathbb{R}^n$. The parallel linear subspace is the linear span $\mathbb{R}[b-a]$ of $b-a$.

Fact: *A nonempty subset $M \subset \mathbb{R}^n$ is an affine subspace if and only if with any pair of distinct points a, b from M M contains the entire line $\ell = \{(1-\lambda) + \lambda b : \lambda \in \mathbb{R}\}$ spanned by a, b.*

- $\emptyset \neq M = \{x \in \mathbb{R}^n : Ax = b\}$. The parallel linear subspace is $\{x : Ax = 0\}$.
- Given a *nonempty* set $X \subset \mathbb{R}^n$, let $\mathrm{Aff}(X)$ be the set of all finite *affine* combinations of vectors from X. This set – *the affine span* (or *affine hull*) *of* X – is an affine subspace, contains X, and is the intersection of all affine subspaces containing X. The parallel linear subspace is $\mathrm{Lin}(X - a)$, where a is an arbitrary point from X.

♠ **Note:** The last two examples are "universal:" *Every affine subspace M in $\mathbb{R}^n$ can be represented as $M = \mathrm{Aff}(X)$ for a properly chosen finite and nonempty set $X \subset \mathbb{R}^n$, same as can be represented as $M = \{x : Ax = b\}$ for a properly chosen matrix A and vector b such that the system $Ax = b$ is solvable.*

♣ **Affine bases and dimension.** Let M be an affine subspace in $\mathbb{R}^n$, and L be the parallel linear subspace.

♠ *By definition*, the *affine dimension* (or simply *dimension*) $\dim M$ of M is the (linear) dimension $\dim L$ of the linear subspace L to which M is parallel.

♠ We say that vectors $x_0, x_1..., x_m$, $m \geq 0$,

• *are affinely independent*, if no nontrivial (not all coefficients are zeros) linear combination of these vectors *with zero sum of coefficients* is the zero vector

Equivalently: $x_0, ..., x_m$ are affinely independent if and only if the coefficients in an *affine* combination $x = \sum_{i=0}^{m} \lambda_i x_i$ are uniquely defined by the value x of this combination.

• *affinely span* M, if

$$M = \mathrm{Aff}(\{x_0, ..., x_m\}) = \left\{\sum_{i=0}^m \lambda_i x_i : \sum_{i=0}^m \lambda_i = 1\right\}$$

• *form an affine basis in* M, if $x_0, ..., x_m$ are affinely independent and affinely span M.

♠ **Facts:** Let M be an affine subspace in $\mathbb{R}^n$, and L be the parallel linear subspace. Then

♡ *A collection $x_0, x_1, ..., x_m$ of vectors is an affine basis in M if and only if $x_0 \in M$ and the vectors $x_1 - x_0, x_2 - x_0, ..., x_m - x_0$ form a (linear) basis in L*

♡ *The following properties of a collection $x_0, ..., x_m$ of vectors from M are equivalent to each other:*

- *$x_0, ..., x_m$ is an affine basis in M*
- *$x_0, ..., x_m$ affinely span M and $m = \dim M$*
- *$x_0, ..., x_m$ are affinely independent and $m = \dim M$*
- *$x_0, ..., x_m$ affinely span M and is a minimal, w.r.t. inclusion, collection with this property*
- *$x_0, ..., x_m$ form a maximal, w.r.t. inclusion, affinely independent collection of vectors from M* (that is, the vectors $x_0, ..., x_m$ are affinely independent, and extending this collection by any vector *from M* yields an affinely dependent collection of vectors).

♠ **Facts:**

♡ *Let* $L = \mathsf{Lin}(X)$. *Then* L *admits a linear basis composed of vectors from* X.

♡ *Let* $X \neq \emptyset$ *and* $M = \mathsf{Aff}(X)$. *Then* M *admits an affine basis composed of vectors from* X.

♡ *Let* L *be a linear subspace. Then every linearly independent collection of vectors from* L *can be extended to a linear basis of* L.

♡ *Let* M *be an affine subspace. Then every affinely independent collection of vectors from* M *can be extended to an affine basis of* M.

Examples:

• $\dim\{a\} = 0$, and the only affine basis of $\{a\}$ is $x_0 = a$.

• $\dim \mathbb{R}^n = n$. When $n > 0$, there are infinitely many affine bases in $\mathbb{R}^n$, e.g., one composed of the zero vector and the n standard basic orths.

• $M = \{x \in \mathbb{R}^n : x_1 = 1\} \Rightarrow \dim M = n - 1$. An example of an affine basis in M is $e_1, e_1 + e_2, e_1 + e_3, ..., e_1 + e_n$.

Extension: M is an affine subspace in $\mathbb{R}^n$ of the dimension $n-1$ if and only if M can be represented as $M = \{x \in \mathbb{R}^n : e^T x = b\}$ with $e \neq 0$. Such a set is called *hyperplane*.

♠ **Note:** A hyperplane $M = \{x : e^T x = b\}$ $(e \neq 0)$ splits $\mathbb{R}^n$ into two *half-spaces*

$$\Pi_+ = \{x : e^T x \geq b\}, \Pi_- = \{x : e^T x \leq b\}$$

and is the common boundary of these half-spaces.

A polyhedral set is the intersection of a finite (perhaps empty) family of half-spaces.

♠ **Facts:**

♡ $M \subset M'$ *are affine subspaces in* $\mathbb{R}^n \Rightarrow \dim M \leq \dim M'$, *with equality taking place if and only if* $M = M'$.

$\Rightarrow$ *Whenever* M *is an affine subspace in* $\mathbb{R}^n$, *we have* $0 \leq \dim M \leq n$

♡ *In every representation of an affine subspace as* $M = \{x \in \mathbb{R}^n : Ax = b\}$, *the number of rows in* A *is at least* $n - \dim M$. *This number is equal to* $n - \dim M$ *if and only if the rows of* A *are linearly independent.*

"Calculus" of affine subspaces

♡ [taking intersection] When M_1, M_2 are affine subspaces in $\mathbb{R}^n$ *and* $M_1 \cap M_2 \neq \emptyset$, so is the set $M_1 \cap M_2$.

Extension: *If nonempty*, the intersection $\bigcap\limits_{\alpha\in\mathcal{A}} M_\alpha$ of an arbitrary family $\{M_\alpha\}_{\alpha\in\mathcal{A}}$ of affine subspaces in $\mathbb{R}^n$ is an affine subspace. The parallel linear subspace is $\bigcap_{\alpha\in\mathcal{A}} L_\alpha$, where L_α are the linear subspaces parallel to M_α.

♡ [summation] When M_1, M_2 are affine subspaces in $\mathbb{R}^n$, so is their *arithmetic sum*

$$M_1 + M_2 = \{x = u + v : u \in M_1, v \in M_2\}.$$

The linear subspace parallel to $M_1 + M_2$ is $L_1 + L_2$, where the linear subspaces L_i are parallel to M_i, $i = 1, 2$

♡ [taking direct product] When $M_1 \subset \mathbb{R}^{n_1}$ and $M_2 \subset \mathbb{R}^{n_2}$, the *direct product* (or *direct sum*) of M_1 and M_2 – the set

$$M_1 \times M_2 := \{[x_1; x_2] \in \mathbb{R}^{n_1+n_2} : x_1 \in M_1, x_2 \in M_2\}$$

is an affine subspace in $\mathbb{R}^{n_1+n_2}$. The parallel linear subspace is $L_1 \times L_2$, where linear subspaces $L_i \subset \mathbb{R}^{n_i}$ are parallel to M_i, $i = 1, 2$.

♡ [taking image under affine mapping] When M is an affine subspace in $\mathbb{R}^n$ and $x \mapsto Px + p : \mathbb{R}^n \to \mathbb{R}^m$ is an affine mapping, the image $PM + p = \{y = Px + p : x \in M\}$ of M under the mapping is an affine subspace in $\mathbb{R}^m$. The parallel subspace is $PL = \{y = Px : x \in L\}$, where L is the linear subspace parallel to M

♡ [taking inverse image under affine mapping] When M is a linear subspace in $\mathbb{R}^n$, $x \mapsto Px + p : \mathbb{R}^m \to \mathbb{R}^n$ is an affine mapping *and the inverse image* $Y = \{y : Py + p \in M\}$ *of* M *under the mapping is nonempty*, Y is an affine subspace. The parallel linear subspace if $P^{-1}(L) = \{y : Py \in L\}$, where L is the linear subspace parallel to M.

Convex Sets and Functions

♣ **Definitions:**

♠ A *set* $X \subset \mathbb{R}^n$ is called *convex*, if along with every two points x, y it contains the entire segment linking the points:

$$\boxed{x, y \in X, \lambda \in [0, 1] \Rightarrow (1 - \lambda)x + \lambda y \in X.}$$

♡ **Equivalently:** $X \in \mathbb{R}^n$ is convex, if X is closed w.r.t. taking all *convex combinations* of its elements (i.e., *linear combinations with nonnegative coefficients summing up to 1*):

$$\boxed{\begin{array}{c}\forall k \geq 1 : x_1, ..., x_k \in X, \lambda_1 \geq 0, ..., \lambda_k \geq 0, \sum_{i=1}^k \lambda_i = 1 \\ \Rightarrow \sum_{i=1}^k \lambda_i x_i \in X\end{array}}$$

Example: *A polyhedral set* $X = \{x \in \mathbb{R}^n : Ax \leq b\}$ *is convex. In particular, linear and affine subspaces are convex sets.*

♠ A *function* $f(x) : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is called *convex*, if its *epigraph*

$$\mathrm{Epi}\{f\} = \{[x; \tau] : \tau \geq f(x)\}$$

is convex.

♡ **Equivalently:** f is convex, if

$$\boxed{\begin{array}{c} x, y \in \mathbb{R}^n, \lambda \in [0, 1] \\ \Rightarrow f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y) \end{array}}$$

♡ **Equivalently:** f is convex, if f satisfies the *Jensen's Inequality:*

$$\boxed{\begin{array}{c} \forall k \geq 1 : x_1, ..., x_k \in \mathbb{R}^n, \lambda_1 \geq 0, ..., \lambda_k \geq 0, \sum_{i=1}^k \lambda_i = 1 \\ \Rightarrow f\left(\sum_{i=1}^k \lambda_i x_i\right) \leq \sum_{i=1}^k \lambda_i f(x_i) \end{array}}$$

Example: A piecewise linear function

$$f(x) = \begin{cases} \max\limits_{i \leq I}[a_i^T x + b_i], & Px \leq p \\ +\infty, & \text{otherwise} \end{cases}$$

is convex.

♠ **Convex hull:** For a nonempty set $X \subset \mathbb{R}^n$, its *convex hull* is the set composed of all convex combinations of elements of X:

$$\boxed{\mathrm{Conv}(X) = \left\{ x = \sum_{i=1}^{m} \lambda_i x_i : \begin{array}{l} x_i \in X,\, 1 \le i \le m \in \mathbf{N} \\ \lambda_i \ge 0\, \forall i, \sum_i \lambda_i = 1 \end{array} \right\}}$$

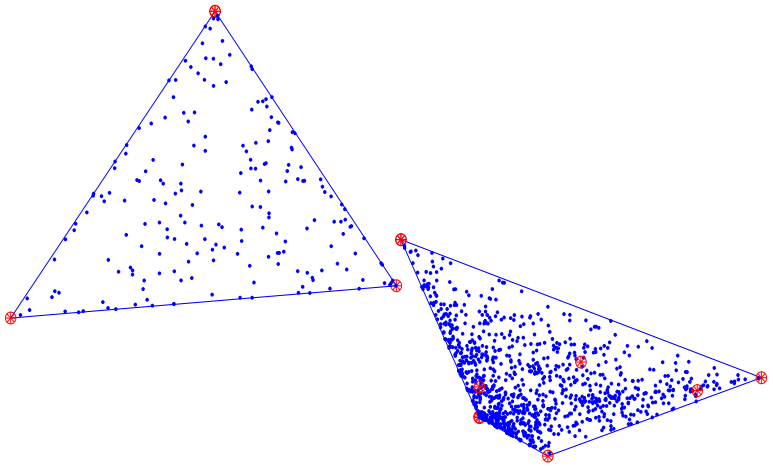
By definition, $\mathrm{Conv}(\emptyset) = \emptyset$.

Fact: *The convex hull of X is convex, contains X and is the intersection of all convex sets containing X and thus is the smallest, w.r.t. inclusion, convex set containing X.*

Note: a convex combination is an affine one, and an affine combination is a linear one, whence

$$\boxed{\begin{array}{c} X \subset \mathbb{R}^n \Rightarrow \mathrm{Conv}(X) \subset \mathrm{Lin}(X) \\ \emptyset \ne X \subset \mathbb{R}^n \Rightarrow \mathrm{Conv}(X) \subset \mathrm{Aff}(X) \subset \mathrm{Lin}(X) \end{array}}$$

Example: Convex hulls of a 3- and an 8-point sets (red dots) on the 2D plane:



♣ **Dimension of a** *nonempty* **set** $X \in \mathbb{R}^n$**:**

♡ When X is a linear subspace, $\dim X$ is the linear dimension of X (the cardinality of (any) linear basis in X)

♡ When X is an *affine subspace*, $\dim X$ is the *linear dimension of the linear subspace parallel to* X (that is, *the cardinality of* (any) *affine basis of* X *minus 1*)

♡ When X is an arbitrary nonempty subset of $\mathbb{R}^n$, $\dim X$ *is the dimension of the affine hull* $\mathrm{Aff}(X)$ of X.

Note: Some sets X are in the scope of more than one of these three definitions. For these sets, all applicable definitions result in the same value of $\dim X$.

Calculus of Convex Sets

♠ [taking intersection]: if X_1, X_2 are convex sets in $\mathbb{R}^n$, so is their intersection $X_1 \cap X_2$. In fact, *the intersection* $\bigcap_{\alpha \in \mathcal{A}} X_\alpha$ *of a whatever family of convex subsets in* $\mathbb{R}^n$ *is convex.*

♠ [taking arithmetic sum]: if X_1, X_2 are convex sets $\mathbb{R}^n$, so is the set $X_1 + X_2 = \{x = x_1 + x_2 : x_1 \in X_1, x_2 \in X_2\}$.

♠ [taking affine image]: if X is a convex set in $\mathbb{R}^n$, A is an $m \times n$ matrix, and $b \in \mathbb{R}^m$, then the set $AX + b := \{Ax + b : x \in X\} \subset \mathbb{R}^m$ – the image of X under the affine mapping $x \mapsto Ax + b : \mathbb{R}^n \to \mathbb{R}^m$ – is a convex set in $\mathbb{R}^m$.

♠ [taking inverse affine image]: if X is a convex set in $\mathbb{R}^n$, A is an $n \times k$ matrix, and $b \in \mathbb{R}^m$, then the set $\{y \in \mathbb{R}^k : Ay + b \in X\}$ – the inverse image of X under the affine mapping $y \mapsto Ay + b : \mathbb{R}^k \to \mathbb{R}^n$ – is a convex set in $\mathbb{R}^k$.

♠ [taking direct product]: if the sets $X_i \subset \mathbb{R}^{n_i}$, $1 \le i \le k$, are convex, so is their direct product $X_1 \times ... \times X_k \subset \mathbb{R}^{n_1 + ... + n_k}$.

Calculus of Convex Functions

♠ [taking linear combinations with positive coefficients] if functions $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are convex and $\lambda_i > 0$, $1 \le i \le k$, then the function

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

is convex.

♠ [direct summation] if functions $f_i : \mathbb{R}^{n_i} \to \mathbb{R} \cup \{+\infty\}$, $1 \le i \le k$, are convex, so is their direct sum

$$f([x^1; ...; x^k]) = \sum_{i=1}^{k} f_i(x^i) : \mathbb{R}^{n_1+...+n_k} \to \mathbb{R} \cup \{+\infty\}$$

♠ [taking supremum] the supremum $f(x) = \sup_{\alpha \in \mathcal{A}} f_\alpha(x)$ of a whatever (nonempty) family $\{f_\alpha\}_{\alpha \in \mathcal{A}}$ of convex functions is convex.

♠ [affine substitution of argument] if a function $f(x) : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is convex and $x = Ay + b : \mathbb{R}^m \to \mathbb{R}^n$ is an affine mapping, then the function $g(y) = f(Ay + b) : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ is convex.

♠ **Theorem on superposition:** *Let* $f_i(x) : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ *be convex functions, and let* $F(y) : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ *be a convex function which is nondecreasing w.r.t. every one of the variables* $y_1, ..., y_m$*. Then the superposition*

$$g(x) = \begin{cases} F(f_1(x), ..., f_m(x)), & f_i(x) < +\infty, 1 \le i \le m \\ +\infty, & \text{otherwise} \end{cases}$$

of F *and* $f_1, ..., f_m$ *is convex.*

Note: if some of f_i, say, $f_1, ..., f_k$, are affine, then the Theorem on superposition theorem remains valid when we require the monotonicity of F w.r.t. $y_{k+1}, ..., y_m$ only.

Cones

♣ **Definition:** *A set $X \subset \mathbb{R}^n$ is called a cone, if X is nonempty, convex and is homogeneous*, that is,

$$x \in X, \lambda \geq 0 \Rightarrow \lambda x \in X$$

Equivalently: *A set $X \subset \mathbb{R}^n$ is a cone, if X is nonempty and is closed w.r.t. addition of its elements and multiplication of its elements by nonnegative reals:*

$$x, y \in X, \lambda, \mu \geq 0 \Rightarrow \lambda x + \mu y \in X$$

Equivalently: *A set $X \subset \mathbb{R}^n$ is a cone, if X is nonempty and is closed w.r.t. taking conic combinations of its elements* (that is, linear combinations with nonnegative coefficients):

$$\forall m : x_i \in X, \lambda_i \geq 0, 1 \leq i \leq m \Rightarrow \sum_{i=1}^{m} \lambda_i x_i \in X.$$

Examples: • Every linear subspace in $\mathbb{R}^n$ (i.e., every solution set of a *homogeneous* system of linear equations with n variables) is a cone

• The solution set $X = \{x \in \mathbb{R}^n : Ax \leq 0\}$ of a *homogeneous* system of linear inequalities is a cone. Such a cone is called *polyhedral*.

♣ A cone X is called *pointed*, if it does not contain lines passing through the origin, or, equivalently, if the only vector d such that $d \in X$ and $-d \in X$ is the zero vector: $d = 0$.

♣ **Conic hull:** *For a nonempty set $X \subset \mathbb{R}^n$, its conic hull* $\mathrm{Cone}\,(X)$ *is defined as the set of all conic combinations of elements of X:*

$$\boxed{\begin{array}{l} X \neq \emptyset \\ \Rightarrow \mathrm{Cone}\,(X) = \left\{ x = \sum\limits_i \lambda_i x_i : \begin{array}{l} \lambda_i \geq 0, 1 \leq i \leq m \in \mathbf{N} \\ x_i \in X, 1 \leq i \leq m \end{array} \right\} \end{array}}$$

By definition, $\mathrm{Cone}\,(\emptyset) = \{0\}$.

Fact: $\mathrm{Cone}\,(X)$ *is a cone, contains X and is the intersection of all cones containing X, and thus is the smallest, w.r.t. inclusion, cone containing X.*

Example: *The conic hull of the set $X = \{e_1, ..., e_n\}$ of all basic orths in $\mathbb{R}^n$ is the nonnegative orthant $\mathbb{R}^n_+ = \{x \in \mathbb{R}^n : x \geq 0\}$. This cone is pointed.*

Calculus of Cones

♠ [taking intersection] if X_1, X_2 are cones in $\mathbb{R}^n$, so is their intersection $X_1 \cap X_2$. In fact, *the intersection* $\bigcap_{\alpha\in\mathcal{A}} X_\alpha$ *of a whatever family* $\{X_\alpha\}_{\alpha\in\mathcal{A}}$ *of cones in* $\mathbb{R}^n$ *is a cone.*

♠ [taking arithmetic sum] if X_1, X_2 are cones in $\mathbb{R}^n$, so is the set $X_1 + X_2 = \{x = x_1 + x_2 : x_1 \in X_1, x_2 \in X_2\}$;

♠ [taking linear image] if X is a cone in $\mathbb{R}^n$ and A is an $m \times n$ matrix, then the set $AX := \{Ax : x \in X\} \subset \mathbb{R}^m$ – the image of X under the linear mapping $x \mapsto Ax : \mathbb{R}^n \to \mathbb{R}^m$ – is a cone in $\mathbb{R}^m$.

♠ [taking inverse linear image] if X is a cone in $\mathbb{R}^n$ and A is an $n \times k$ matrix, then the set $\{y \in \mathbb{R}^k : Ay \in X\}$ – the inverse image of X under the linear mapping $y \mapsto Ay\mathbb{R}^k \to \mathbb{R}^n$ – is a cone in $\mathbb{R}^k$.

♠ [taking direct products] if $X_i \subset \mathbb{R}^{n_i}$ are cones, $1 \le i \le k$, so is the direct product $X_1 \times ... \times X_k \subset \mathbb{R}^{n_1+...+n_k}$.

♠ [passing to the dual cone] if X is a cone in $\mathbb{R}^n$, so is its *dual cone* defined as

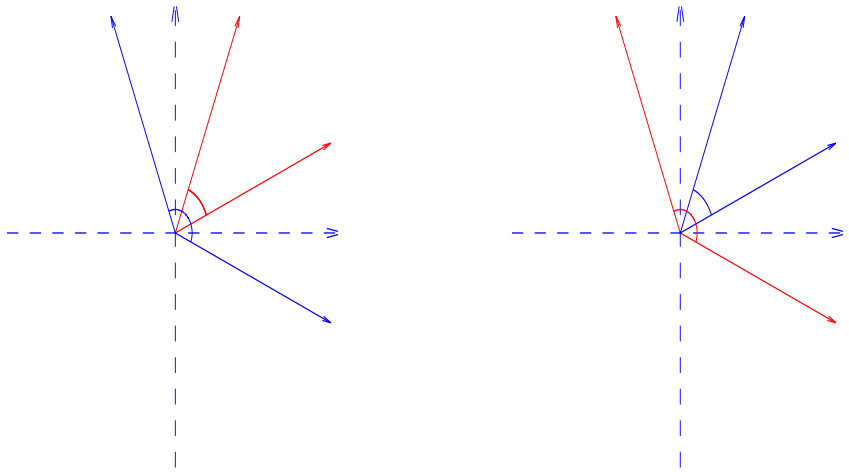
$$X_* = \{y \in \mathbb{R}^n : y^T x \geq 0 \ \forall x \in X\}.$$

Examples:

- The cone dual to a linear subspace L is the orthogonal complement $L^\perp$ of L
- The cone dual to the nonnegative orthant $\mathbb{R}^n_+$ is the nonnegative orthant itself:

$$(\mathbb{R}^n_+)_* := \{y \in \mathbb{R}^n : y^T x \geq 0 \, \forall x \geq 0\} = \{y \in \mathbb{R}^n : y \geq 0\}.$$

- 2D cones bounded by blue rays are dual to cones bounded by red rays:



Preparing Tools: Caratheodory Theorem

Theorem. *Let $x_1, ..., x_N \in \mathbb{R}^n$ and $m = \dim\{x_1, ..., x_N\}$. Then every point x which is a convex combination of $x_1, ..., x_N$ can be represented as a convex combination of at most $m+1$ of the points $x_1, ..., X_N$.*

Proof.

• Let $M = \mathrm{Aff}\{x_1, ..., x_N\}$, so that $\dim M = m$. By shifting M (which does not affect the statement we intend to prove) we can make M a m-dimensional linear subspace in $\mathbb{R}^n$. Representing points from the linear subspace M by their m-dimensional vectors of coordinates in a basis of M, we can identify M and $\mathbb{R}^m$, and this identification does not affect the statement we intend to prove. Thus, assume w.l.o.g. that $m = n$.

• Let $x = \sum_{i=1}^N \mu_i x_i$ be a representation of x as a convex combination of $x_1, ..., x_N$ *with as small number of nonzero coefficients as possible.* Reordering $x_1, ..., x_N$ and omitting terms with zero coefficients, assume w.l.o.g. that $x = \sum_{i=1}^M \mu_i x_i$, so that $\mu_i > 0,\ 1 \le i \le M$, and $\sum_{i=1}^M \mu_i = 1$. It suffices to show that $M \le n+1$. Let, on the contrary, $M > n+1$.

• Consider the system of linear equations in variables $\delta_1, ..., \delta_M$:

$$\sum_{i=1}^M \delta_i x_i = 0;\ \sum_{i=1}^M \delta_i = 0$$

This is a homogeneous system of $n+1$ linear equations in $M > n+1$ variables, and thus it has a nontrivial solution $\bar{\delta}_1, ..., \bar{\delta}_M$. Setting $\mu_i(t) = \mu_i + t\bar{\delta}_i$, we have

$$\forall t : x = \sum_{i=1}^M \mu_i(t) x_i, \sum_{i=1}^M \mu_i(t) = 1.$$

• Since $\bar{\delta}$ is nontrivial and $\sum_i \bar{\delta}_i = 0$, the set $I = \{i : \delta_i < 0\}$ is nonempty. Let $\bar{t} = \min\limits_{i \in I} \mu_i / |\bar{\delta}_i|$. Then all $\mu_i(\bar{t})$ are ≥ 0, at least one of $\mu_i(\bar{t})$ is zero, and

$$x = \sum_{i=1}^M \mu_i(\bar{t}) x_i,\ \sum_{i=1}^M \mu_i(\bar{t}) = 1.$$

We get a representation of x as a convex combination of x_i with *less than* M nonzero coefficients, which is impossible. □

Quiz:

- In the nature, there are 26 "pure" types of tea, denoted A, B,..., Z; all other types are mixtures of these "pure" types. In the market, 111 blends of pure types, rather than the pure types of tea themselves, are sold.
- John prefers a specific blend of tea which is not sold in the market; from experience, he found that in order to get this blend, he can buy 93 of the 111 market blends and mix them in certain proportion.
- An OR student pointed out that to get his favorite blend, John could mix appropriately just 27 properly selected market blends. Another OR student found that just 26 of market blends are enough.
- John does not believe the students, since no one of them asked what exactly is his favorite blend. Is John right?

Quiz:

• In the nature, there are 26 "pure" types of tea, denoted A, B,..., Z. In the market, 111 blends of these types are sold.

• John knows that his favorite blend can be obtained by mixing in appropriate proportion 93 of the 111 market blends. Is it true that the same blend can be obtained by mixing

  • 27 market blends?

  • 26 market blends?

Both answers are true. Let us speak about *unit weight* portions of tea blends. Then

• a blend can be identified with 26-dimensional vector

$$x = [x_A; ...; x_Z]$$

where $x_?$ is the weight of pure tea ? in the unit weight portion of the blend. The 26 entries in x are nonnegative and sum up to 1;

• denoting the marked blends by $x^1, ..., x^{111}$ and the favorite blend of John by $\bar{x}$, we know that

$$\bar{x} = \sum_{i=1}^{111} \lambda_i x^i$$

with nonnegative coefficients λ_i. Comparing the weights of both sides, we conclude that $\sum_{i=1}^{111} \lambda_i = 1$

$\Rightarrow$ $\bar{x}$ is a convex combination of $x^1, ..., x^{111}$

$\Rightarrow$ [by Caratheodory and due to $\dim x^i = 26$] $\bar{x}$ *is a convex combination of just* $26 + 1 = 27$ *of the market blends*, thus the first student is right.

• The vectors $x^1, ..., x^{111}$ have unit sums of entries thus belong to the hyperplane

$$M = \{[x_A; ...; x_Z] : x_A + ... + x_Z = 1\}$$

which has dimension 25

$\Rightarrow$ The dimension of the set $\{x^1, x^2, ..., x^{111}\}$ is at most $m = 25$

$\Rightarrow$ *By Caratheodory,* $\bar{x}$ *is a convex combination of just* $m + 1 = 26$ *vectors from* $\{x^1, ..., x^{111}\}$, thus the second student also is right.

Preparing Tools: Helly Theorem

Theorem. *Let $A_1, ..., A_N$ be convex sets in $\mathbb{R}^n$ which belong to an affine subspace M of dimension m. Assume that every $m+1$ sets of the collection have a point in common. then all N sets have a point in common.*

Proof. • Same as in the proof of Caratheodory Theorem, we can assume w.l.o.g. that $m = n$.

• We need the following fact:

Theorem [Radon] *Let $x_1, ..., x_N$ be points in $\mathbb{R}^n$. If $N \geq n+2$, we can split the index set $\{1, ..., N\}$ into two nonempty non-overlapping subsets I, J such that*

$$\text{Conv}\{x_i : i \in I\} \cap \text{Conv}\{x_i : i \in J\} \neq \emptyset.$$

From Radon to Helly: Let us prove Helly's theorem by indiction in N. There is nothing to prove when $N \leq n+1$. Thus, assume that $N \geq n+2$ and that the statement holds true for all collections of $N-1$ sets, and let us prove that the statement holds true for N-element collections of sets as well.

• Given $A_1, ..., A_N$, we define the N sets

$$B_i = A_1 \cap A_2 \cap ... \cap A_{i-1} \cap A_{i+1} \cap ... \cap A_N.$$

By inductive hypothesis, all B_i are nonempty. Choosing a point $x_i \in B_i$, we get $N \geq n+2$ points x_i, $1 \leq i \leq N$.

• By Radon Theorem, after appropriate reordering of the sets $A_1, ..., A_N$, we can assume that for certain k, $\text{Conv}\{x_1, ..., x_k\} \cap \text{Conv}\{x_{k+1}, ..., x_N\} \neq \emptyset$. We claim that *if* $b \in \text{Conv}\{x_1, ..., x_k\} \cap \text{Conv}\{x_{k+1}, ..., x_N\}$, *then* b *belongs to all* A_i, which would complete the inductive step.

To support our claim, note that

— when $i \leq k$, $x_i \in B_i \subset A_j$ for all $j = k+1, ..., N$, that is,

$$i \leq k \Rightarrow x_i \in \cap_{j=k+1}^{N} A_j.$$

Since the latter set is convex and b is a convex combination of $x_1, ..., x_k$, we get $b \in \cap_{j=k+1}^{N} A_j$.

— when $i > k$, $x_i \in B_i \subset A_j$ for all $1 \leq j \leq k$, that is,

$$i \geq k \Rightarrow x_i \in \cap_{j=1}^{k} A_j.$$

Similarly to the above, it follows that $b \in \cap_{j=1}^{k} A_j$.

Thus, our claim is correct.

Proof of Radon Theorem: Let $x_1, ..., x_N \in \mathbb{R}^n$ and $N \geq n+2$. We want to prove that we can split the set of indexes $\{1, ..., N\}$ into non-overlapping nonempty sets I, J such that $\mathrm{Conv}\{x_i : i \in I\} \cap \mathrm{Conv}\{x_i : i \in J\} \neq \emptyset$.
Indeed, consider the system of $n+1 < N$ homogeneous linear equations in N variables $\delta_1, ..., \delta_N$:

$$\sum_{i=1}^{N} \delta_i x_i = 0, \sum_{i=1}^{N} \delta_i = 0. \qquad (*)$$

This system has a nontrivial solution $\bar{\delta}$. Let us set $I = \{i : \bar{\delta}_i > 0\}$, $J = \{i : \bar{\delta}_i \leq 0\}$. Since $\bar{\delta} \neq 0$ and $\sum_{i=1}^{N} \bar{\delta}_i = 0$, both I, J are nonempty, do not intersect and $\mu := \sum_{i \in I} \bar{\delta}_i = \sum_{i \in J} [-\bar{\delta}_i] > 0$. $(*)$ implies that

$$\underbrace{\sum_{i \in I} \frac{\bar{\delta}_i}{\mu} x_i}_{\in \mathrm{Conv}\{x_i : i \in I\}} = \underbrace{\sum_{i \in J} \frac{[-\bar{\delta}_i]}{\mu} x_i}_{\in \mathrm{Conv}\{x_i : i \in J\}} \qquad \square$$

Quiz: The daily functioning of a plant is described by the linear constraints

$$\begin{array}{lrcl} (a) & Ax & \leq & f \in \mathbb{R}^{10}_{+} \\ (b) & Bx & \geq & d \in \mathbb{R}^{2013} \\ (c) & Cx & \leq & c \in \mathbb{R}^{2000} \end{array} \qquad (!)$$

- x: decision vector
- $f \in \mathbb{R}^{10}_{+}$: vector of resources
- d: vector of demands
- There are N demand scenarios d^i. In the evening of day $t-1$, the manager knows that the demand of day t will be one of the N scenarios, but he does *not* know which one. The manager should arrange a vector of resources f for the next day, at a price $c_\ell \geq 0$ per unit of resource f_ℓ, in order to make the next day production problem feasible.
- It is known that every one of the demand scenarios can be "served" by \$1 purchase of resources.

(?) *How much should the manager invest in resources to make the next day problem feasible when*

- $N = 1$ • $N = 2$ • $N = 10$ • $N = 11$
- $N = 12$ • $N = 2013$?

$$(a): Ax \le f \in \mathbb{R}^{10}_+; \quad (b): Bx \ge d \in \mathbb{R}^{2013}; \quad (c): Cx \le c \in \mathbb{R}^{2000}$$

Quiz answer: With N scenarios, \$ $\min[N, 11]$ is enough!

Indeed, the vector of resources $f \in \mathbb{R}^{10}_+$ appears only in the constraints (a)

$\Rightarrow$ surplus of resources makes no harm

$\Rightarrow$ with N scenarios d^i, \$ N in resources is enough: every d^i can be "served" by \$ 1 purchase of appropriate resource vector $f^i \ge 0$, thus it suffices to buy the vector $f^1 + ... + f^N$ which costs \$ N and is $\ge f^i$ for every $i = 1, ..., n$.

To see than \$ 11 is enough, let F_i be the set of all resource vectors f which cost at most \$11 and allow to "serve" demand $d^i \in D$.

A. $F_i \in \mathbb{R}^{10}$ is convex (and even polyhedral): it admits polyhedral representation

$$F_i = \{f \in \mathbb{R}^{10} : \exists x : Cx \le c, Bx \ge d^i, Ax \le f, f \ge 0, \sum_{\ell=1}^{10} c_\ell f_\ell \le 11\}$$

B. Every 11 sets $F_{i_1}, ..., F_{i_{11}}$ of the family $F_1, ..., F_i$ have a point in common.

Indeed, scenario d^{i_s} can be "served" by \$ 1 vector $f^s \ge 0$

$\Rightarrow$ every one of the scenarios $d^{i_1}, ..., d^{i_{11}}$ can be served by the \$ 11 vector of resources $f = f^1 + ... + f^{11}$

$\Rightarrow$ f belongs to every one of $F_{i_1}, ..., F_{i_{11}}$

• By Helly, **A** and **B** imply that all the sets $F_1, ..., F_N$ have a point f in common. f costs at most \$ 11 (the description of F_i) and allows to "serve" *every one* of the demands d^1,...,d^N.

Preparing Tools: Homogeneous Farkas Lemma

♣ **Question:** When a *homogeneous* linear inequality

$$a^T x \geq 0 \qquad (*)$$

is a consequence of a system of *homogeneous* linear inequalities

$$a_i^T x \geq 0,\ i = 1, ..., m \qquad (!)$$

i.e., when $(*)$ is satisfied at every solution to (!)?

Observation: If a is a conic combination of $a_1, ..., a_m$:

$$\exists \lambda_i \geq 0 : a = \sum_i \lambda_i a_i, \qquad (+)$$

then $(*)$ is a consequence of (!).

Indeed, $(+)$ implies that

$$a^T x = \sum_i \lambda_i a_i^T x \ \forall x,$$

and thus for every x with $a_i^T x \geq 0 \, \forall i$ one has $a^T x \geq 0$.

♣ **Homogeneous Farkas Lemma:** $(+)$ *is a consequence of* (!) *if and only if* a *is a conic combination of* $a_1, ..., a_m$.

♣ **Equivalently:** *Given vectors* $a_1, ..., a_m \in \mathbb{R}^n$, *let* $K = \text{Cone}\{a_1, ..., a_m\} = \{\sum_i \lambda_i a_i : \lambda \geq 0\}$ *be the conic hull of the vectors. Given a vector* a,

• *it is easy to certify that* $a \in \text{Cone}\{a_1, ..., a_m\}$*: a certificate is a collection of weights* $\lambda_i \geq 0$ *such that* $\sum_i \lambda_i a_i = a$;

• *it is easy to certify that* $a \notin \text{Cone}\{a_1, ..., a_m\}$*: a certificate is a vector* d *such that* $a_i^T d \geq 0 \, \forall i$ *and* $a^T d < 0$.

Proof of HFL: All we need to prove is that *If a is not a conic combination of $a_1, ..., a_m$, then there exists d such that $a^T d < 0$ and $a_i^T d \geq 0$, $i = 1, ..., m$.*

Fact: The set $K = \text{Cone}\{a_1, ..., a_m\}$ is polyhedrally representable:

$$\text{Cone}\{a_1, ..., a_m\} = \left\{ x : \exists \lambda \in \mathbb{R}^m : \begin{array}{c} x = \sum_i \lambda_i a_i \\ \lambda \geq 0 \end{array} \right\}.$$

$\Rightarrow$ By Fourier-Motzkin, K is polyhedral:

$$K = \{x : d_\ell^T x \geq c_\ell, 1 \leq \ell \leq L\}.$$

Observation I: $0 \in K \Rightarrow c_\ell \leq 0 \, \forall \ell$

Observation II: $\lambda a_i \in \text{Cone}\{a_1, ..., a_m\} \, \forall \lambda > 0 \Rightarrow \lambda d_\ell^T a_i \geq c_\ell \, \forall \lambda \geq 0 \Rightarrow d_\ell^T a_i \geq 0 \, \forall i, \ell$.

Now, $a \notin \text{Cone}\{a_1, ..., a_m\} \Rightarrow \exists \ell = \ell_* : d_{\ell_*}^T a < c_{\ell_*} \leq 0 \Rightarrow d_{\ell_*}^T a < 0$.

$\Rightarrow d = d_{\ell_*}$ *satisfies* $a^T d < 0$, $a_i^T d \geq 0$, $i = 1, ..., m$, *Q.E.D.*

Corollary: *Let $a_1, ..., a_m \in \mathbb{R}^n$ and $K = \text{Cone}\{a_1, ..., a_m\}$, and let $K_* = \{x \in \mathbb{R}^n : x^T u \geq 0 \forall u \in K\}$ be the dual cone. Then K itself is the cone dual to K_*:*

$$\begin{aligned}(K_*)_* &:= \{u : u^T x \geq 0 \ \forall u \in K_*\} \\ &= K := \{\textstyle\sum_i \lambda_i a_i : \lambda_i \geq 0\}.\end{aligned}$$

Proof. • We clearly have

$$\boxed{K_* = (\text{Cone}\{a_1, ..., a_m\})_* = \{d : d^T a_i \geq 0 \, \forall i = 1, ..., m\}}$$

• If K is a cone, then, by definition of K_*, every vector from K has nonnegative inner products with all vectors from K_* and thus $K \subset (K_*)_*$ for every cone K.

• To prove the opposite inclusion $(K_*)_* \subset K$ in the case of $K = \text{Cone}\{a_1, ..., a_m\}$, let $a \in (K_*)_*$, and let us verify that $a \in K$. Assuming this is *not* the case, by HFL there exists d such that $a^T d < 0$ and $a_i^T d \geq 0 \forall i$ $\Rightarrow d \in K_*$, that is, $a \notin (K_*)_*$, which is a contradiction.

Understanding Structure of a Polyhedral Set

♣ **Situation:** We consider a polyhedral set

$$X = \{x \in \mathbb{R}^n : Ax \le b\}, A = \begin{bmatrix} a_1^T \\ \cdots \\ a_m^T \end{bmatrix} \in \mathbb{R}^{m\times n} \qquad (X)$$
$$\Leftrightarrow X = \left\{x \in \mathbb{R}^n : a_i^T x \le b_i,\ i \in \mathcal{I} = \{1, ..., m\}\right\}.$$

Standing assumption: $X \neq \emptyset$.

♠ **Faces of X.** Let us pick a subset $I \subset \mathcal{I}$ and replace in (X) the inequality constraints $a_i^T x \le b_i$, $i \in I$ with their equality versions $a_i^T x = b_i$, $i \in I$. The resulting set

$$X_I = \{x \in \mathbb{R}^n : a_i^T x \le b_i, i \in \mathcal{I}\backslash I,\ a_i^T x = b_i,\ i \in I\}$$

if nonempty, is called a *face* of X.

Examples: • X is a face of itself: $X = X_\emptyset$.

• Let $\Delta_n = \mathrm{Conv}\{0, e_1, ..., e_n\} = \{x \in \mathbb{R}^n : x \ge 0, \sum_{i=1}^n x_i \le 1\}$
$\Rightarrow \mathcal{I} = \{1, ..., n+1\}$ with $a_i^T x := -x_i \le 0 =: b_i$, $1 \le i \le n$, and $a_{n+1}^T x := \sum_i x_i \le 1 =: b_{n+1}$
Every subset $I \subset \mathcal{I}$ *different from* $\mathcal{I}$ defines a face. For example, $I = \{1, n+1\}$ defines the face $\{x \in \mathbb{R}^n : x_1 = 0, x_i \ge 0, \sum_i x_i = 1\}$.

$$X = \left\{ x \in \mathbb{R}^n : a_i^T x \leq b_i,\ i \in \mathcal{I} = \{1, ..., m\} \right\}$$

Facts:

- A face

$$\emptyset \neq X_I = \{x \in \mathbb{R}^n : a_i^T x \leq b_i, i \in \mathcal{I}, a_i^T x = b_i, i \in I\}$$

of X is a nonempty polyhedral set

- A face of a face of X can be represented as a face of X
- if X_I and $X_{I'}$ are faces of X and their intersection is nonempty, this intersection is a face of X:

$$\emptyset \neq X_I \cap X_{I'} \Rightarrow X_I \cap X_{I'} = X_{I \cup I'}.$$

♣ A face X_I is called *proper*, if $X_I \neq X$.

Theorem: *A face X_I of X is proper if and only if* $\dim X_I < \dim X$.

$$X = \left\{x \in \mathbb{R}^n : a_i^T x \le b_i,\ i \in \mathcal{I} = \{1, ..., m\}\right\}$$
$$X_I = \{x \in \mathbb{R}^n : a_i^T x \le b_i, i \in \mathcal{I}\backslash I,\ a_i^T x = b_i,\ i \in I\}$$

Proof. One direction is evident. Now assume that X_I is a proper face, and let us prove that $\dim X_I < \dim X$.

• Since $X_I \neq X$, there exists $i_* \in I$ such that $a_{i_*}^T x \not\equiv b_{i_*}$ on X and thus on $M = \mathrm{Aff}(X)$.

$\Rightarrow$ The set $M_+ = \{x \in M : a_{i_*}^T x = b_{i_*}\}$ contains X_I (and thus is an affine subspace containing $\mathrm{Aff}(X_I)$), and is $\subsetneq M$.

$\Rightarrow$ $\mathrm{Aff}(X_I) \subsetneq M$, whence

$$\dim X_I = \dim \mathrm{Aff}(X_I) < \dim M = \dim X. \qquad \square$$

Extreme Points of a Polyhedral Set

$$X = \left\{x \in \mathbb{R}^n : a_i^T x \leq b_i,\ i \in \mathcal{I} = \{1, ..., m\}\right\}$$

Definition. *A point $v \in X$ is called an extreme point, or a vertex of X, if it can be represented as a face of X:*

$$\begin{array}{l} \exists I \subset \mathcal{I}: \\ X_I := \{x \in \mathbb{R}^n : a_i^T x \leq b_i,\ i \in \mathcal{I}, a_i^T x = b_i,\ i \in I\} = \{v\}. \end{array} \qquad (*)$$

Geometric characterization: *A point $v \in X$ is a vertex of X iff v is not the midpoint of a nontrivial segment contained in X:*

$$v \pm h \in X \Rightarrow h = 0. \qquad (!)$$

Proof. • Let v be a vertex, so that $(*)$ takes place for certain I, and let h be such that $v \pm h \in X$; we should prove that $h = 0$. We have $\forall i \in I$:

$$\begin{array}{l} \{b_i \geq a_i^T(v - h) = b_i - a_i^T h \ \&\ b_i \geq a_i^T(v + h) = b_i + a_i^T h\} \\ \Rightarrow a_i^T h = 0 \Rightarrow a_i^T[v \pm h] = b_i. \end{array}$$

Thus, $v \pm h \in X_I = \{v\}$, whence $h = 0$. We have proved that $(*)$ implies $(!)$.

$$v \pm h \in X \Rightarrow h = 0. \qquad (!)$$

$$\begin{array}{l} \exists I \subset \mathcal{I}: \\ X_I := \{x \in \mathbb{R}^n : a_i^T x \le b_i,\ i \in \mathcal{I}, a_i^T x = b_i,\ i \in I\} = \{v\}. \end{array} \qquad (*)$$

• Let us prove that (!) implies (∗). Indeed, let $v \in X$ be such that (!) takes place; we should prove that (∗) holds true for certain I. Let

$$I = \{i \in \mathcal{I} : a_i^T v = b_i\},$$

so that $v \in X_I$. It suffices to prove that $X_I = \{v\}$. Let, on the opposite, $\exists e \neq 0 : v + e \in X_I$. Then $a_i^T(v + e) = b_i = a_i^T v$ for all $i \in I$, that is, $a_i^T e = 0\, \forall i \in I$, that is,

$$a_i^T(v \pm te) = b_i\ \forall (i \in I, t > 0).$$

When $i \in \mathcal{I} \backslash I$, we have $a_i^T v < b_i$ and thus $a_i^T(v \pm te) \le b_i$ provided $t > 0$ is small enough.

$\Rightarrow$ *There exists* $\bar{t} > 0$*:* $a_i^T(v \pm \bar{t}e) \le b_i\, \forall i \in \mathcal{I}$*, that is* $v \pm \bar{t}e \in X$, which is a desired contradiction. □

Algebraic characterization: *A point*

$$v \in X = \{x \in \mathbb{R}^n : a_i^T x \leq b_i, i \in \mathcal{I} = \{1, ..., m\}\}$$

is a vertex of X *iff among the inequalities* $a_i^T x \leq b_i$ *which are active at* v (i.e., $a_i^T v = b_i$) *there are* n *with linearly independent* a_i*:*

$$\text{Rank}\,\{a_i : i \in I_v\} = n,\ I_v = \{i \in \mathcal{I} : a_i^T v = b_i\} \qquad (!)$$

Proof. • Let v be a vertex of X; we should prove that among the vectors a_i, $i \in I_v$ there are n linearly independent. Assuming that this is not the case, the linear system $a_i^T e = 0$, $i \in I_v$ in variables e has a *nonzero* solution e. We have

$$\begin{array}{l} i \in I_v \Rightarrow a_i^T[v \pm te] = b_i\, \forall t, \\ i \in \mathcal{I} \backslash I_v \Rightarrow a_i^T v < b_i \\ \Rightarrow a_i^T[v \pm te] \leq b_i \text{ for all small enough } t > 0 \end{array}$$

whence $\exists \bar{t} > 0 : a_i^T[v \pm \bar{t}e] \leq b_i\, \forall i \in \mathcal{I}$, that is $v \pm \bar{t}e \in X$, which is impossible due to $\bar{t}e \neq 0$. □

$$v \in X = \{x \in \mathbb{R}^n : a_i^T x \leq b_i, i \in \mathcal{I}\}$$
$$\text{Rank}\,\{a_i : i \in I_v\} = n,\ I_v = \{i \in \mathcal{I} : a_i^T v = b_i\} \qquad (!)$$

• Now let $v \in X$ satisfy (!); we should prove that v is a vertex of X, that is, that the relation $v \pm h \in X$ implies $h = 0$. Indeed, when $v \pm h \in X$, we should have

$$\forall i \in I_v : b_i \geq a_i^T[v_i \pm h] = b_i \pm a_i^T h$$
$$\Rightarrow a_i^T h = 0\ \forall i \in I_v.$$

Thus, $h \in \mathbb{R}^n$ is orthogonal to n linearly independent vectors from $\mathbb{R}^n$, whence $h = 0$. □

♠ **Observation:** *The set* $\mathrm{Ext}(X)$ *of extreme points of a polyhedral set is finite.*

Indeed, there could be no more extreme points than faces.

♠ **Observation:** *If* X *is a polyhedral set and* X_I *is a face of* X*, then* $\mathrm{Ext}(X_I) \subset \mathrm{Ext}(X)$*.*

Indeed, extreme points are singleton faces, and a face of a face of X can be represented as a face of X itself.

♡ **Note:** Geometric characterization of extreme points allows to define this notion for every convex set X: *A point $x \in X$ is called extreme, if* $x \pm h \in X \Rightarrow h = 0$

♡ **Fact:** Let X be convex and $x \in X$. Then $x \in \mathrm{Ext}(X)$ iff in every representation

$$x = \sum_{i=1}^{m} \lambda_i x_i$$

of x as a convex combination of points $x_i \in X$ with positive coefficients one has

$$x_1 = x_2 = ... = x_m = x$$

♡ **Fact:** *For a convex set X and $x \in X$, $x \in \mathrm{Ext}(X)$ iff the set $X \backslash \{x\}$ is convex.*

♡ **Fact:** *For every $X \subset \mathbb{R}^n$,*

$$\mathrm{Ext}(\mathrm{Conv}(X)) \subset X$$

Example: *Let us describe extreme points of the set*

$$\Delta_{n,k} = \{x \in \mathbb{R}^n : 0 \le x_i \le 1, \sum_i x_i \le k\} \qquad [k \in \mathbf{N}, 0 \le k \le n]$$

Description: *These are exactly 0/1-vectors from $\Delta_{n,k}$.*
In particular, the vertices of the box $\Delta_{n,n} = \{x \in \mathbb{R}^n : 0 \le x_i \le 1 \, \forall i\}$ are all 2^n 0/1-vectors from $\mathbb{R}^n$.

Proof. • If x is a 0/1 vector from $\Delta_{n,k}$, then the set of active at x bounds $0 \le x_i \le 1$ is of cardinality n, and the corresponding vectors of coefficients are linearly independent

$\Rightarrow x \in \mathrm{Ext}(\Delta_{n,k})$

• If $x \in \mathrm{Ext}(\Delta_{n,k})$, then among the active at x constraints defining $\Delta_{n,k}$ should be n linearly independent. The only options are:

— all n active constraints are among the bounds $0 \le x_i \le 1$

$\Rightarrow x$ is a 0/1 vector

— $n-1$ of the active constraints are among the bounds, and the remaining active constraint is $\sum_i x_i \le k$: $\sum_i x_i = k$.

$\Rightarrow x$ has $(n-1)$ coordinates 0/1 & the sum of all n coordinates is integral

$\Rightarrow x$ is 0/1 vector.

Example: *An $n \times n$ matrix A is called double stochastic, if the entries are nonnegative and all the row and the column sums are equal to 1. Thus, the set of double-stochastic matrices is a polyhedral set in $\mathbb{R}^{n \times n}$:*

$$\Pi_n = \left\{ x = [x_{ij}]_{i,j} \in \mathbb{R}^{n \times n} : \begin{array}{l} x_{ij} \geq 0 \, \forall i, j \\ \sum_{i=1}^{n} x_{ij} = 1 \, \forall j \\ \sum_{j=1}^{n} x_{ij} = 1 \, \forall i \end{array} \right\}$$

What are the extreme points of Π_n ?

Birkhoff's Theorem *The vertices of Π_n are exactly the $n \times n$ permutation matrices (exactly one nonzero entry, equal to 1, in every row and every column).*

Proof • A permutation matrix P can be viewed as 0/1 vector of dimension n^2 and as such is an extreme point of the box

$$\{[x_{ij}] : 0 \leq x_{ij} \leq 1\}$$

which contains Π_n. Therefore P is an extreme point of Π_n since by geometric characterization of extreme points, *an extreme point x of a convex set is an extreme point of every smaller convex set to which x belongs*.

• Let P be an extreme point of Π_n. To prove that Π_n is a permutation matrix, note that Π_n is cut off $\mathbb{R}^{n^2}$ by n^2 inequalities $x_{ij} \geq 0$ and $2n-1$ linearly independent linear equations (since if all but one row and column sums are equal to 1, the remaining sum also is equal to 1). By algebraic characterization of extreme points, *at least* $n^2-(2n-1) = (n-1)^2 > (n-2)n$ *entries in* P *should be zeros.*

$\Rightarrow$ *there is a column in* P *with at least* $n-1$ *zero entries*

$\Rightarrow \exists i_*, j_* : P_{i_*j_*} = 1.$

$\Rightarrow$ P *belongs to the face* $\{P \in \Pi_n : P_{i_*j_*} = 1\}$ *of* Π_n and thus is an extreme point of the face. In other words, *the matrix obtained from* P *by eliminating* i_**-th row and* j_**-th column is an extreme point in the set* Π_{n-1}. Iterating the reasoning, we conclude that P is a permutation matrix. $\square$

Recessive Directions and Recessive Cone

$$X = \{x \in \mathbb{R}^n : Ax \leq b\} \text{ is nonempty}$$

♣ **Definition.** A vector $d \in \mathbb{R}^n$ is called a *recessive direction of* X, if X contains a ray directed by d:

$$\exists \bar{x} \in X : \bar{x} + td \in X \ \forall t \geq 0.$$

♠ **Observation:** *d is a recessive direction of X iff $Ad \leq 0$.*

♠ **Corollary:** *Recessive directions of X form a polyhedral cone, namely, the cone* $\mathrm{Rec}(X) = \{d : Ad \leq 0\}$, *called the recessive cone of X.*
Whenever $x \in X$ and $d \in \mathrm{Rec}(X)$, one has $x + td \in X$ for all $t \geq 0$. In particular,

$$X + \mathrm{Rec}(X) = X.$$

♠ **Observation:** *The larger is a polyhedral set, the larger is its recessive cone:*

$$X \subset Y \text{ are polyhedral} \Rightarrow \mathrm{Rec}(X) \subset \mathrm{Rec}(Y).$$

Recessive Subspace of a Polyhedral Set

$$X = \{x \in \mathbb{R}^n : Ax \leq b\} \text{ is nonempty}$$

♠ **Observation:** *Directions d of lines contained in X are exactly the vectors from the recessive subspace*

$$L = \mathrm{Ker}A := \{d : Ad = 0\} = \mathrm{Rec}(X) \cap [-\mathrm{Rec}(X)]$$

of X, and $X = X + \mathrm{Ker}A$. In particular,

$$X = \bar{X} + \mathrm{Ker}A,$$
$$\bar{X} = \{x \in \mathbb{R}^n : Ax \leq b, x \in [\mathrm{Ker}A]^{\perp}\}$$

Note: *$\bar{X}$ is a polyhedral set which does not contain lines.*

Pointed Polyhedral Cones & Extreme Rays

$$K = \{x : Ax \leq 0\}$$

♣ **Definition.** *Polyhedral cone K is called pointed, if it does not contain lines.* **Equivalently:** *K is pointed iff $K \cap \{-K\} = \{0\}$.*
Equivalently: *K is pointed iff* Ker$A = \{0\}$..

♣ **Definition** *An extreme ray of K is a face of K which is a nontrivial ray (i.e., the set $\mathbb{R}_+ d = \{td : t \geq 0\}$ associated with a nonzero vector, called a generator of the ray).*

♠ **Geometric characterization:** *A vector $d \in K$ is a generator of an extreme ray of K (in short: d is an extreme direction of K) iff it is nonzero and whenever d is a sum of two vectors from K, both vectors are nonnegative multiples of d:*

$$d = d_1 + d_2,\ d_1, d_2 \in K \Rightarrow$$
$$\exists t_1 \geq 0, t_2 \geq 0 : d_1 = t_1 d, d_2 = t_2 d.$$

Example: $K = \mathbb{R}^n_+$ This cone is pointed, and its extreme directions are positive multiples of basic orths. There are n extreme rays $R_i = \{x \in \mathbb{R}^n : x_i \geq 0, x_j = 0 \forall (j \neq i)\}$.

♠ **Observation:** d is an extreme direction of K iff some (and then – all) positive multiples of d are extreme directions of K.

$$\begin{array}{rcl} K & = & \{x \in \mathbb{R}^n : Ax \leq 0\} \\ & = & \left\{x \in \mathbb{R}^n : a_i^T x \leq 0,\ i \in \mathcal{I} = \{1, ..., m\}\right\} \end{array}$$

♠ **Algebraic characterization of extreme directions:** Let K be a *pointed* cone. *Direction $d \in K \backslash \{0\}$ is an extreme direction of K iff among the homogeneous inequalities $a_i^T x \leq 0$ which are active at d* (i.e., are satisfied at d as equations) *there are $n-1$ inequalities with linearly independent a_i's.*

Proof. Let $0 \neq d \in K$, $I = \{i \in \mathcal{I} : a_i^T d = 0\}$.

• Let the set $\{a_i : i \in I\}$ contain $n-1$ linearly independent vectors, say, $a_1, ..., a_{n-1}$. Let us prove that then d is an extreme direction of K. Indeed, the set

$$L = \{x : a_i^T x = 0,\ 1 \leq i \leq n-1\} \supset K_I$$

is a one-dimensional linear subspace in $\mathbb{R}^n$. Since $0 \neq d \in L$, we have $L = \mathbb{R}d$. Since $d \in K$, the ray $\mathbb{R}_+ d$ is contained in K_I: $\mathbb{R}_+ d \subset K_I$. Since $K_I \subset L$, all vectors from K_I are real multiples of d. Since K is pointed, no negative multiples of d belong to K_I.

$\Rightarrow$ $K_I = R_+ d$ and $d \neq 0$, i.e., K_I is an extreme ray of K, and d is a generator of this ray. □

• Let d be an extreme direction of K, that is, $\mathbb{R}_+ d$ is a face of K, and let us prove that the set $\{a_i : i \in I\}$ contains $n-1$ linearly independent vectors. Assuming the opposite, the solution set L of the homogeneous system of linear equations

$$a_i^T x = 0,\ i \in I$$

is of dimension ≥ 2 and thus contains a vector h which is *not* proportional to d. When $i \notin I$, we have $a_i^T d < 0$ and thus $a_i^T(d+th) \leq 0$ when $|t|$ is small enough.

$\Rightarrow \exists \bar{t} > 0 : |t| \leq \bar{t} \Rightarrow a_i^T[d+th] \leq 0\ \forall i \in \mathcal{I}$

$\Rightarrow$ *the face K_I of K* (which is the smallest face of K containing d)*contains two non-proportional nonzero vectors $d, d+\bar{t}h$, and thus is strictly larger than $\mathbb{R}_+ d$.*

$\Rightarrow \mathbb{R}_+ d$ is *not* a face of K, which is a desired contradiction. □

Base of a Cone

$$K = \{x \in \mathbb{R}^n : Ax \leq 0\}.$$

♣ **Definition.** *A set B of the form*

$$B = \{x \in K : f^T x = 1\} \qquad (*)$$

is called a base of K, if it is nonempty and intersects with every (nontrivial) ray in K:

$$\forall 0 \neq d \in K \ \exists! t \geq 0 : td \in B.$$

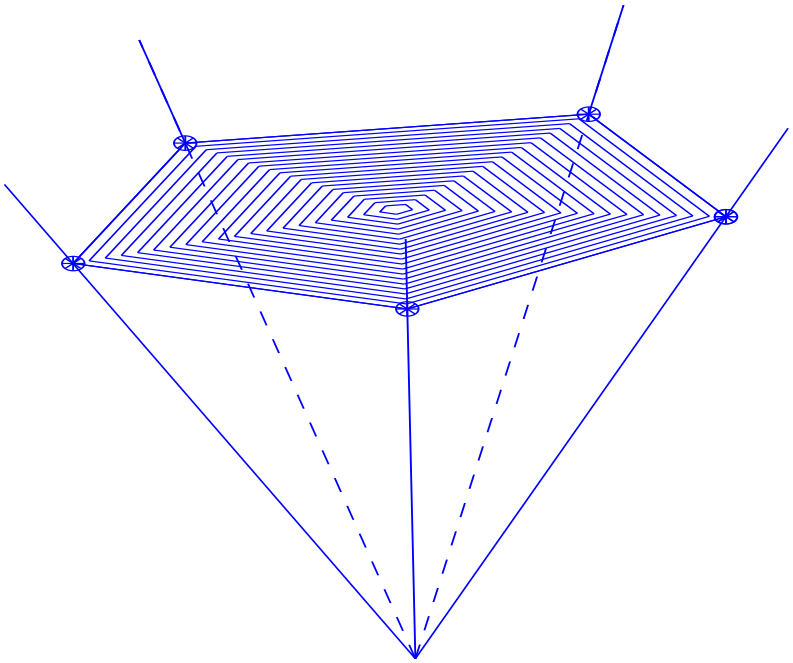
Example: The set

$$\Delta_n = \{x \in \mathbb{R}^n_+ : \sum_{i=1}^n x_i = 1\}$$

is a base of $\mathbb{R}^n_+$.

♠ **Observation:** *Set $(*)$ is a base of K iff $K \neq \{0\}$ and f makes strictly positive inner products with all nonzero vectors from K:*

$$0 \neq x \in K \Rightarrow f^T x > 0.$$



3D cone K and its base B (pentagon)

Note: *extreme rays of K are generated by extreme points of B*

♠ Facts:

- *K possesses a base iff $K \neq \{0\}$ and K is pointed.*
- *K possesses a base B iff K possesses extreme rays, and there is one-to-one correspondence between extreme rays of K and extreme points of B: extreme directions of K are exactly positive multiples of extreme points of B.*
- *The recessive cone of a base B of K is trivial:* $\mathrm{Rec}(B) = \{0\}$.

$$K = \{x \in \mathbb{R}^n : Ax \leq 0\}.$$

Observations:

- *If K possesses extreme rays, then K is nontrivial* ($K \neq \{0\}$) *and pointed* ($K \cap [-K] = \{0\}$).

In fact, the inverse is also true.

- *The set of extreme rays of K is finite.*

Indeed, there are no more extreme rays than faces.

Towards the Main Theorem: First Step

Theorem. *Let*

$$X = \{x \in \mathbb{R}^n : Ax \leq b\}$$

be a nonempty polyhedral set which does not contain lines. Then

(i) *The set* $V = \mathrm{Ext}\{X\}$ *of extreme points of* X *is nonempty and finite:*

$$V = \{v_1, ..., v_N\}$$

(ii) *The set of extreme rays of the recessive cone* $\mathrm{Rec}(X)$ *of* X *is finite; let*

$$R = \{r_1, ..., r_M\}$$

be a collection of generators of the extreme rays, one generator per ray.

(iii) *One has*

$$X = \mathrm{Conv}(V) + \mathrm{Cone}\,(R)$$
$$= \left\{ x = \sum_{i=1}^N \lambda_i v_i + \sum_{j=1}^M \mu_j r_j : \begin{array}{l} \lambda_i \geq 0 \, \forall i \\ \sum_{i=1}^N \lambda_i = 1 \\ \mu_j \geq 0 \, \forall j \end{array} \right\}$$

Main Lemma: *Let*

$$X = \{x \in \mathbb{R}^n : a_i^T x \leq b_i,\ 1 \leq i \leq m\}$$

be a nonempty polyhedral set which does not contain lines. Then the set $V = \mathrm{Ext}\{X\}$ *of extreme points of* X *is nonempty and finite, and*

$$X = \mathrm{Conv}(V) + \mathrm{Rec}(X).$$

Proof: Induction in $m = \dim X$**.**

Base $m = 0$ is evident: here

$$X = \{a\},\ V = \{a\},\ \mathrm{Rec}(X) = \{0\}.$$

Inductive step $m \Rightarrow m + 1$**:** Let the statement be true for polyhedral sets of dimension $\leq m$, and let $\dim X = m + 1$, $\mathcal{M} = \mathrm{Aff}(X)$, L be the linear subspace parallel to $\mathcal{M}$.

• Take a point $\bar{x} \in X$ and a nonzero direction $e \in L$. Since X does not contain lines, either e, or $-e$, or both are *not* recessive directions of X. Swapping, if necessary, e and $-e$, assume that $-e \notin \mathrm{Rec}(X)$.

$$X = \{x \in \mathbb{R}^n : a_i^T x \leq b_i,\ 1 \leq i \leq m\}$$
nonempty and does not contain lines

• Let us look at the ray $\bar{x} - \mathbb{R}_+ e = \{x(t) = \bar{x} - te : t \geq 0\}$. Since $-e \notin \mathrm{Rec}(X)$, this ray is *not* contained in X, that is, *all* linear inequalities $0 \leq b_i - a_i^T x(t) \equiv \alpha_i - \beta_i t$ are satisfied when $t = 0$, and *some* of them are violated at certain values of $t \geq 0$. In other words,

$$\alpha_i \geq 0\, \forall i \ \&\ \exists i : \beta_i > 0$$

Let $\bar{t} = \min\limits_{i:\beta_i>0} \frac{\alpha_i}{\beta_i}$, $I = \{i : \alpha_i - \beta_i^T \bar{t} = 0\}$.
Then by construction

$$x(\bar{t}) = \bar{x} - \bar{t}e \in X_I = \{x \in X : a_i^T x = b_i\, i \in I\}$$

so that X_I is a face of X. We claim that this face is proper. Indeed, *every* constraint $b_i - a_i^T x$ with $\beta_i > 0$ is nonconstant on $\mathcal{M} = \mathrm{Aff}(X)$ and thus is non-constant on X. (At least) one of these constraints $b_{i_*} - a_{i_*}^T x \geq 0$ becomes active at $x(\bar{t})$, that is, $i_* \in I$. This constraint is active everywhere on X_I and is nonconstant on X, whence $X_I \subsetneqq X$.

$$X = \{x \in \mathbb{R}^n : a_i^T x \le b_i,\ 1 \le i \le m\}$$
nonempty and does not contain lines

Situation: $X \ni \bar{x} = x(\bar{t}) + \bar{t}e$ with $\bar{t} \ge 0$ and $x(\bar{t})$ belonging to a proper face X_I of X.

• X_i is a proper face of $X \Rightarrow \dim X_I < \dim X \Rightarrow$ (by inductive hypothesis) $\mathrm{Ext}(X_I) \neq \emptyset$ and

$$X_I = \mathrm{Conv}(\mathrm{Ext}(X_I)) + \mathrm{Rec}(X_I) \subset \mathrm{Conv}(\mathrm{Ext}(X)) + \mathrm{Rec}(X).$$

In particular, $\mathrm{Ext}(X) \neq \emptyset$; as we know, this set is finite.

• It is possible that $e \in \mathrm{Rec}(X)$. Then

$$\bar{x} = x(\bar{t}) + \bar{t}e \in [\mathrm{Conv}(\mathrm{Ext}(X)) + \mathrm{Rec}(X)] + \bar{t}e \subset \mathrm{Conv}(\mathrm{Ext}(X)) + \mathrm{Rec}(X).$$

• It is possible that $e \notin \mathrm{Rec}(X)$. In this case, applying the same "moving along the ray" construction to the ray $\bar{x} + \mathbb{R}_+ e$, we get, along with $x(\bar{t}) := \bar{x} - \bar{t}e \in X_I$, a point $x(-\widehat{t}) := x + \widehat{t}e \in X_{\widehat{I}}$, where $\widehat{t} \ge 0$ and $X_{\widehat{I}}$ is a proper face of X, whence, same as above,

$$x(-\widehat{t}) \in \mathrm{Conv}(\mathrm{Ext}(X)) + \mathrm{Rec}(X).$$

By construction, $\bar{x} \in \mathrm{Conv}\{x(\bar{t}), x(-\widehat{t})\}$, whence $\bar{x} \in \mathrm{Ext}(X) + \mathrm{Rec}(X)$.

♣ Reasoning in pictures:

♠ Case A: e is a recessive direction of X.

• Let us move from $\bar{x}$ along the direction $-e$.

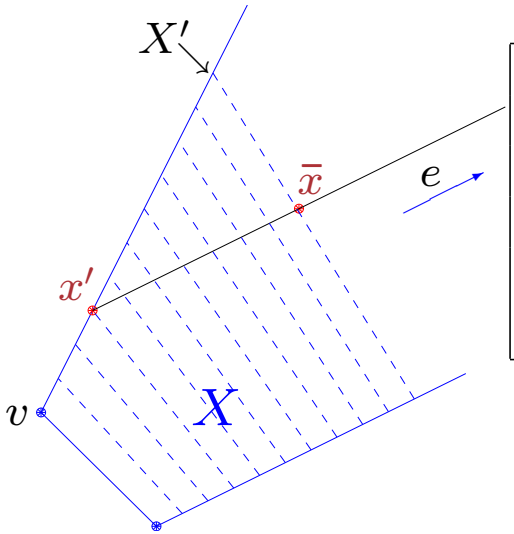
— since $e \in L$, we all the time will stay in $\mathrm{Aff}(X)$

— since $-e$ is not a recessive direction of X, eventually we will be about to leave X. *When it happens, our position* $x' = \bar{x} - \lambda e$, $\lambda \geq 0$, *will belong to a proper face* X' *of* X

$\Rightarrow$ $\dim(X') < \dim(X)$

$\Rightarrow$ [Ind. Hype.] $x' = v + r$ with $v \in \mathrm{Conv}(\mathrm{Ext}(X')) \subset \mathrm{Conv}(\mathrm{Ext}(X))$, $r \in \mathrm{Rec}(X') \subset \mathrm{Rec}(X)$

$\Rightarrow$ $\bar{x} = \underbrace{v}_{\in \mathrm{Conv}(\mathrm{Ext}(X))} + \underbrace{[r + \lambda e]}_{\in \mathrm{Rec}(X)} \subset \mathrm{Conv}(\mathrm{Ext}(X)) + \mathrm{Rec}(X)$



$\exists$ $v \in \mathrm{Conv}(\mathrm{Ext}(X')) \subset \mathrm{Conv}(\mathrm{Ext}(X))$, $r \in \mathrm{Rec}(X') \subset \mathrm{Rec}(X)$:

$$x' = v + r$$

$\Rightarrow$ for some $\lambda \geq 0$

$$\bar{x} = \underbrace{v}_{\in \mathrm{Conv}(\mathrm{Ext}(X))} + \underbrace{r + \lambda e}_{\in \mathrm{Rec}(X)}$$

♣ Reasoning in pictures (continued):

♠ Case B: both e and $-e$ are not recessive directions of X.

• As in Case A, we move from $\bar{x}$ along the direction $-e$ until hitting a proper face X' of X at a point x'.

$\Rightarrow$ [Ind. Hype.] $x' = v + r$ with

$v \in \mathrm{Conv}(\mathrm{Ext}(X')) \subset \mathrm{Conv}(\mathrm{Ext}(X))$, $r \in \mathrm{Rec}(X') \subset \mathrm{Rec}(X)$

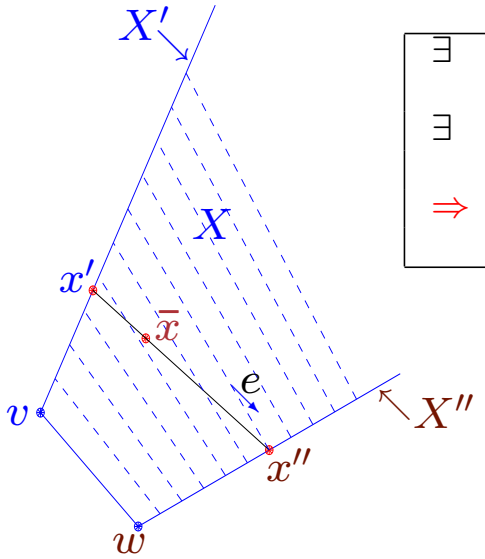
• *Since e is not a recessive direction of X*, when moving from $\bar{x}$ along the direction e, we eventually hit a proper face X'' of X at a point x''

$\Rightarrow$ [Ind. Hyp.] $x'' = w + s$ with

$w \in \mathrm{Conv}(\mathrm{Ext}(X'')) \subset \mathrm{Conv}(\mathrm{Ext}(X))$, $s \in \mathrm{Rec}(X'') \subset \mathrm{Rec}(X)$

• $\bar{x}$ is a convex combination of x' and x'': $\bar{x} = \lambda x' + (1-\lambda)x''$

$$\Rightarrow \bar{x} = \underbrace{\lambda v + (1-\lambda)w}_{\in \mathrm{Conv}(\mathrm{Ext}(X))} + \underbrace{\lambda r + (1-\lambda)s}_{\in \mathrm{Rec}(X)} \in \mathrm{Conv}(\mathrm{Ext}(X)) + \mathrm{Rec}(X)$$



$$\begin{array}{ll} \exists & v \in \mathrm{Conv}(\mathrm{Ext}(X')) \subset \mathrm{Conv}(\mathrm{Ext}(X)): \\ & \quad x' \in v + \mathrm{Rec}(X') \subset v + \mathrm{Rec}(X) \\ \exists & w \in \mathrm{Conv}(\mathrm{Ext}(X'')) \subset \mathrm{Conv}(\mathrm{Ext}(X)): \\ & \quad x'' \in w + \mathrm{Rec}(X'') \subset w + \mathrm{Rec}(X) \\ \Rightarrow & \bar{x} \in \mathrm{Conv}\{x', x''\} \subset \mathrm{Conv}\{v, w\} + \mathrm{Rec}(X) \\ & \quad \subset \mathrm{Conv}(\mathrm{Ext}(X)) + \mathrm{Rec}(X) \end{array}$$

♠ **Summary:** We have proved that $\mathrm{Ext}(X)$ is nonempty and finite, and every point $\bar{x} \in X$ belongs to

$$\mathrm{Conv}(\mathrm{Ext}(X)) + \mathrm{Rec}(X),$$

that is, $X \subset \mathrm{Conv}(\mathrm{Ext}(X)) + \mathrm{Rec}(X)$. Since

$$\mathrm{Conv}(\mathrm{Ext}(X)) \subset X$$

and $X + \mathrm{Rec}(X) = X$, we have also

$$X \supset \mathrm{Conv}(\mathrm{Ext}(X)) + \mathrm{Rec}(X)$$

$\Rightarrow X = \mathrm{Conv}(\mathrm{Ext}(X)) + \mathrm{Rec}(X)$. Induction is complete. □

Corollaries of Main Lemma: *Let X be a nonempty polyhedral set which does not contain lines.*

A. *If X is has a trivial recessive cone, then*

$$X = \text{Conv}(\text{Ext}(X)).$$

B. *If K is a nontrivial pointed polyhedral cone, the set of extreme rays of K is nonempty and finite, and if $r_1, ..., r_M$ are generators of the extreme rays of K, then*

$$K = \text{Cone}\,\{r_1, ..., r_M\}.$$

Proof of B: Let B be a base of K, so that B is a nonempty polyhedral set with $\text{Rec}(B) = \{0\}$. By **A**, $\text{Ext}(B)$ is nonempty, finite and $B = \text{Conv}(\text{Ext}(B))$, whence $K = \text{Cone}\,(\text{Ext}(B))$. It remains to note every nontrivial ray in K intersects B, and a ray is extreme iff this intersection is an extreme point of B.

♠ Augmenting Main Lemma with Corollary **B**, we get the Theorem.

♣ We have seen that if X is a nonempty polyhedral set not containing lines, then X admits a representation

$$X = \mathrm{Conv}(V) + \mathrm{Cone}\,\{R\} \qquad (*)$$

where

- $V = V_*$ is the nonempty finite set of all extreme points of X;
- $R = R_*$ is a finite set composed of generators of the extreme rays of $\mathrm{Rec}(X)$ (this set can be empty).

♠ It is easily seen that this representation is "minimal:" *Whenever X is represented in the form of $(*)$ with finite sets V, R,*
— V contains all vertices of X
— R contains generators of all extreme rays of $\mathrm{Rec}(X)$.

Structure of a Polyhedral Set

Main Theorem (i) *Every nonempty polyhedral set $X \subset \mathbb{R}^n$ can be represented as*

$$X = \mathrm{Conv}(V) + \mathrm{Cone}\,(R) \qquad (*)$$

where $V \subset \mathbb{R}^n$ is a nonempty finite set, and $R \subset \mathbb{R}^n$ is a finite set.

(ii) *Vice versa, if a set X given by representation $(*)$ with a nonempty finite set V and finite set R, X is a nonempty polyhedral set.*

Proof. (i): We know that (i) holds true when X does not contain lines. Now, *every* nonempty polyhedral set X can be represented as

$$X = \widehat{X} + L,\ L = \mathrm{Lin}\{f_1, ..., f_k\},$$

where $\widehat{X}$ is a nonempty polyhedral set which does not contain lines. In particular,

$$\Rightarrow \begin{array}{rcl} \widehat{X} & = & \mathrm{Conv}\{v_1, ..., v_N\} + \mathrm{Cone}\,\{r_1, ..., r_M\} \\ X & = & \mathrm{Conv}\{v_1, ..., v_N\} \\ & & + \mathrm{Cone}\,\{r_1, ..., r_M, f_1, -f_1, ..., f_K, -f_K\} \end{array}$$

Note: *In every representation $(*)$ of X, $\mathrm{Cone}\,(R) = \mathrm{Rec}(X)$.*

(ii): Let

$$X = \mathrm{Conv}\{v_1, ..., v_N\} + \mathrm{Cone}\,\{r_1, .., r_M\} \subset \mathbb{R}^n \qquad [N \geq 1, M \geq 0]$$

We want to prove that X is a polyhedral set.
Indeed, X admits polyhedral representation:

$$X = \left\{ x : \exists \lambda, \mu : \begin{array}{l} x = \sum_{i=1}^N \lambda_i v_i + \sum_{j=1}^M \mu_j r_j \\ \lambda \geq 0, \mu \geq 0, \sum_{i=1}^N \lambda_i = 1 \end{array} \right\}$$

and as such is polyhedral.

Immediate Corollaries

Corollary I. *A nonempty polyhedral set X possesses extreme points iff X does not contain lines. In addition, the set of extreme points of X is finite.*
Indeed, if X does not contain lines, X has extreme points and their number is finite by Main Lemma. When X contains lines, *every* point of X belongs to a line contained in X, and thus X has no extreme points.

Corollary II. (i) *A nonempty polyhedral set X is bounded iff its recessive cone is trivial:* $\mathrm{Rec}(X) = \{0\}$, *and in this case X is the convex hull of the (nonempty and finite) set of its extreme points:*

$$\emptyset \neq \mathrm{Ext}(X) \text{ is finite and } X = \mathrm{Conv}(\mathrm{Ext}(X)).$$

(ii) *The convex hull of a nonempty finite set V is a bounded polyhedral set, and* $\mathrm{Ext}(\mathrm{Conv}(X)) \subset V$.

Proof of (i): if $\mathrm{Rec}(X) = \{0\}$, then X does not contain lines and therefore $\emptyset \neq \mathrm{Ext}(X)$ is finite and

$$\begin{array}{rcll} X & = & \mathrm{Conv}(\mathrm{Ext}(X)) + \mathrm{Rec}(X) & \\ & = & \mathrm{Conv}(\mathrm{Ext}(X)) + \{0\} & (*) \\ & = & \mathrm{Conv}(\mathrm{Ext}(X)), & \end{array}$$

and thus X is bounded as the convex hull of a finite set.

Vice versa, if X is bounded, then X clearly does not contain nontrivial rays and thus $\mathrm{Rec}(X) = \{0\}$.

Proof of (ii): By Main Theorem (ii),

$$X := \mathrm{Conv}(\{v_1, ..., v_m\})$$

is a polyhedral set, and this set clearly is bounded. Besides this, $X = \mathrm{Conv}(V)$ always implies that $\mathrm{Ext}(X) \subset V$.

Application examples:

- *Every vector x from the set*

$$\{x \in \mathbb{R}^n : 0 \leq x_i \leq 1,\ 1 \leq i \leq n, \sum_{i=1}^{n} x_i \leq k\}$$

(k is an integer) is a convex combination of Boolean vectors from this set.

- *Every double-stochastic matrix is a convex combination of permutation matrices.*

Application example: Polar of a polyhedral set. Let $X \subset \mathbb{R}^n$ be a polyhedral set containing the origin. The polar $\text{Polar}\,(X)$ of X is given by

$$\text{Polar}\,(X) = \{y : y^T x \leq 1 \, \forall x \in X\}.$$

Examples:

- $\text{Polar}\,(\mathbb{R}^n) = \{0\}$
- $\text{Polar}\,(\{0\}) = \mathbb{R}^n$
- If L is a linear subspace, then $\text{Polar}\,(L) = L^{\perp}$
- If $K \subset \mathbb{R}^n$ is a cone, then $\text{Polar}\,(K) = -K_*$
- $\text{Polar}\,(\{x : \sum_i |x_i| \leq 1\}) = \{x : |x_i| \leq 1 \, \forall i\}$
- $\text{Polar}\,(\{x : |x_i| \leq 1 \, \forall i\}) = \{x : \sum_i |x_i| \leq 1\}$

Theorem. *When X is a polyhedral set containing the origin, so is* Polar (X), *and* Polar (Polar $(X)) = X$.

Proof. • Representing

$$X = \{x = \sum_i \lambda_i v_i + \sum_j \mu_j r_j : \lambda \geq 0, \sum_i \lambda_i = 1, \mu \geq 0\}$$

we clearly get

$$\text{Polar}\,(X) = \{y : y^T v_i \leq 1\,\forall i, y^T r_j \leq 0\,\forall j\}$$

$\Rightarrow$ Polar (X) *is a polyhedral set.* Inclusion $0 \in$ Polar (X) is evident.

• Let us prove that Polar (Polar $(X)) = X$. By definition of the polar, we have $X \subset$ Polar (Polar (X)). To prove the opposite inclusion, let $\bar{x} \in$ Polar (Polar (X)), and let us prove that $\bar{x} \in X$. X is polyhedral:

$$X = \{x : a_i^T x \leq b_i, 1 \leq i \leq K\}$$

and $b_i \geq 0$ due to $0 \in X$. By scaling inequalities $a_i^T x \leq b_i$, we can assume further that all nonzero b_i's are equal to 1. Setting $I = \{i : b_i = 0\}$, we have

$$\begin{aligned}
&i \in I \Rightarrow a_i^T x \leq 0\,\forall x \in X \Rightarrow \lambda a_i \in \text{Polar}\,(X)\,\forall \lambda \geq 0\\
&\Rightarrow \bar{x}^T[\lambda a_i] \leq 1\,\forall \lambda \geq 0 \Rightarrow a_i^T \bar{x} \leq 0\\
&i \notin I \Rightarrow a_i^T x \leq b_i = 1\,\forall x \in X \Rightarrow a_i \in \text{Polar}\,(X)\\
&\Rightarrow \bar{x}^T a_i \leq 1,
\end{aligned}$$

whence $\bar{x} \in X$. □

Corollary III. (i) *A cone K is polyhedral iff it is the conic hull of a finite set:*

$$K = \{x \in \mathbb{R}^n : Bx \leq 0\}$$
$$\Leftrightarrow \exists R = \{r_1, ..., r_M\} \subset \mathbb{R}^n : K = \mathrm{Cone}\,(R)$$

(ii) *When K is a nontrivial and pointed polyhedral cone, one can take as R the set of generators of the extreme rays of K.*

Proof of (i): If K is a polyhedral cone, then $K = \widehat{K} + L$ with a linear subspace L and a pointed cone $\widehat{K}$. By Main Theorem (i), we have

$$\begin{aligned} \widehat{K} &= \mathrm{Conv}(\mathrm{Ext}(\widehat{K})) + \mathrm{Cone}\,(\{r_1, ..., r_M\}) \\ &= \mathrm{Cone}\,(\{r_1, ..., r_M\}) \end{aligned}$$

since $\mathrm{Ext}(\widehat{K}) = \{0\}$, whence

$$K = \mathrm{Cone}\,(\{r_1, ..., r_M, f_1, -f_1, ..., f_s, -f_s\})$$

where $\{f_1, ...f_s\}$ is a basis in L.

Vice versa, if $K = \mathrm{Cone}\,(\{r_1, ..., r_M\})$, then K is a polyhedral set (Main Theorem (ii)). that is, $K = \{x : Ax \leq b\}$ for certain A, b. Since K is a cone, we have

$$K = \mathrm{Rec}(K) = \{x : Ax \leq 0\},$$

that is, K is a polyhedral cone. □

(ii) is Corollary B of Main Lemma.

Applications in LO

♣ **Theorem.** *Consider a LO program*

$$\mathrm{Opt} = \max_x \left\{ c^T x : Ax \le b \right\},$$

and let the feasible set $X = \{x : Ax \le b\}$ be nonempty and thus representable as

$$X = \mathrm{Conv}(\{v_1, ..., v_N\}) + \mathrm{Cone}\,(\{r_1, ..., r_M\})$$

with properly selected $v_1, ..., v_N$, $r_1, ..., r_M$.

(i) *The program is solvable iff c has nonpositive inner products with all r_j.*

(ii) *If X does not contain lines and the program is bounded, then among its optimal solutions, if any exist, there are vertices of X.*

' **Proof.** • We have

$$\mathrm{Opt} := \sup_{\lambda,\mu} \left\{ \sum_i \lambda_i c^T v_i + \sum_j \mu_j c^T r_j : \begin{array}{l} \lambda_i \ge 0 \\ \sum_i \lambda_i = 1 \\ \mu_j \ge 0 \end{array} \right\} < +\infty$$

$$\Leftrightarrow c^T r \le 0 \, \forall r \in \mathrm{Cone}\,(\{r_1, ..., r_M\}) = \mathrm{Rec}(X)$$

$$\Rightarrow \mathrm{Opt} = \max_i c^T v_i$$

Thus, $\mathrm{Opt} < +\infty$ implies that the best of the points $v_1, ..., v_M$ is an optimal solution.

• It remains to note that when X does not contain lines, we can set $\{v_i\}_{i=1}^N = \mathrm{Ext}(X)$. □

Application to Knapsack problem. *A knapsack can store k items. You have $n \geq k$ items, j-th of value $c_j \geq 0$. How to select items to be placed into the knapsack in order to get the most valuable selection?*

Solution: Assuming for a moment that we can put to the knapsack fractions of items, let x_j be the fraction of item j we put to the knapsack. The most valuable selection then is given by an optimal solution to the LO program

$$\max_x \left\{ \sum_j c_j x_j : 0 \leq x_j \leq 1, \sum_j x_j \leq k \right\}$$

The feasible set is nonempty, polyhedral and bounded, and all extreme points are Boolean vectors from this set

$\Rightarrow$ There is a Boolean optimal solution.

In fact, the optimal solution is evident: we should put to the knapsack k most valuable of the items.

Application to Assignment problem. *There are n jobs and n workers. Every job takes one man-hour. The profit of assigning worker i with job j is c_{ij}. How to assign workers with jobs in such a way that every worker gets exactly one job, every job is carried out by exactly one worker, and the total profit of the assignment is as large as possible?*

Solution: Assuming for a moment that a worker can distribute his time between several jobs and denoting x_{ij} the fraction of activity of worker i spent on job j, we get a *relaxed* problem

$$\max_x \left\{ \sum_{i,j} c_{ij} x_{ij} : x_{ij} \geq 0, \sum_i x_{ij} = 1 \, \forall j, \sum_j x_{ij} = 1 \, \forall i \right\}$$

The feasible set is polyhedral, nonempty and bounded

$\Rightarrow$ Program is solvable, and among the optimal solutions there are extreme points of the set of double stochastic matrices, i.e., permutation matrices

$\Rightarrow$ Relaxation is exact!

Lecture I.4
Duality

Theory of Systems of Linear Inequalities and Duality

♣ We still do not know how to answer some most basic questions about polyhedral sets, e.g.:

♠ *How to recognize that a polyhedral set* $X = \{x \in \mathbb{R}^n : Ax \leq b\}$ *is/is not empty?*

♠ *How to recognize that a polyhedral set* $X = \{x \in \mathbb{R}^n : Ax \leq b\}$ *is/is not bounded?*

♠ *How to recognize that two polyhedral sets* $X = \{x \in \mathbb{R}^n : Ax \leq b\}$ *and* $X' = \{x : A'x \leq b'\}$ *are/are not distinct?*

♠ *How to recognize that a given LO program is feasible/bounded/solvale?*

♠ *.....*

Our current goal is to find answers to these and similar questions, and these answers come from *Linear Programming Duality Theorem* which is the second main theoretical result in LO.

Theorem on Alternative

♣ Consider a system of m strict and nonstrict linear inequalities in variables $x \in \mathbb{R}^n$:

$$a_i^T x \begin{cases} < b_i, & i \in I \\ \leq b_i, & i \in \overline{I} \end{cases} \qquad (\mathcal{S})$$

- $a_i \in \mathbb{R}^n, b_i \in \mathbb{R}$, $1 \leq i \leq m$,
- $I \subset \{1, ..., m\}$, $\overline{I} = \{1, ..., m\} \backslash I$.

Note: $(\mathcal{S})$ is a universal form of a finite system of linear inequalities in n variables.

♣ **Main questions on $(\mathcal{S})$ [operational form]:**

- *How to find a solution to the system if one exists?*
- *How to find out that $(\mathcal{S})$ is infeasible?*

♠ **Main questions on $(\mathcal{S})$ [descriptive form]:**

- *How to certify that $(\mathcal{S})$ is solvable?*
- *How to certify that $(\mathcal{S})$ is infeasible?*

$$a_i^T x \begin{cases} < b_i, & i \in I \\ \leq b_i, & i \in \bar{I} \end{cases} \qquad (\mathcal{S})$$

♠ The simplest certificate for solvability of $(\mathcal{S})$ is *a* solution: plug a candidate certificate into the system and check that the inequalities are satisfied.

Example: The vector $\bar{x} = [10; 10; 10]$ is a solvability certificate for the system

$$\begin{array}{rrrcr} -x_1 & -x_2 & -x_3 & < & -29 \\ x_1 & +x_2 & & \leq & 20 \\ & x_2 & +x_3 & \leq & 20 \\ x_1 & & +x_3 & \leq & 20 \end{array}$$

– when plugging it into the system, we get valid numerical inequalities.

But: *How to certify that $(\mathcal{S})$ has no solution?* E.g., how to certify that the system

$$\begin{array}{rrrcr} -x_1 & -x_2 & -x_3 & < & -30 \\ x_1 & +x_2 & & \leq & 20 \\ & x_2 & +x_3 & \leq & 20 \\ x_1 & & +x_3 & \leq & 20 \end{array}$$

has no solutions?

$$a_i^T x \begin{cases} < b_i, & i \in I \\ \leq b_i, & i \in \overline{I} \end{cases} \qquad (\mathcal{S})$$

♣ *How to certify that* $(\mathcal{S})$ *has no solutions?*

♠ **A recipe:** *Take a weighted sum, with nonnegative weights, of the inequalities from the system, thus getting strict or nonstrict scalar linear inequality which, due its origin, is a consequence of the system – it must be satisfied at every solution to* $(\mathcal{S})$*. If the resulting inequality has no solutions at all, then* $(\mathcal{S})$ *is unsolvable.*

Example: To certify that the system

$2\times$	$-x_1$	$-x_2$	$-x_3$	$<$	-30
$1\times$	x_1	$+x_2$		$\leq$	20
$1\times$		x_2	$+x_3$	$\leq$	20
$1\times$	x_1		$+x_3$	$\leq$	20

has no solutions, take the weighted sum of the inequalities with the weights marked in red, thus arriving at the inequality

$$0 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 < 0.$$

This is a contradictory inequality which is a consequence of the system

$\Rightarrow$ *weights* $\lambda = [2; 1; 1; 1]$ *certify insolvability.*

$$a_i^T x \begin{cases} < b_i, & i \in I \\ \leq b_i, & i \in \overline{I} \end{cases} \qquad (\mathcal{S})$$

A recipe for certifying insolvability:

- *assign inequalities of* $(\mathcal{S})$ *with weights* $\lambda_i \geq 0$ *and sum them up, thus arriving at the inequality*

$$[\textstyle\sum_{i=1}^m \lambda_i a_i]^T x \ \Omega \ \sum_{i=1}^m \lambda_i b_i$$
$$\left[\begin{array}{l} \Omega = \text{"}<\text{"} \ \textit{when} \ \sum_{i\in I} \lambda_i > 0 \\ \Omega = \text{"}\leq\text{"} \ \textit{when} \ \sum_{i\in I} \lambda_i = 0 \end{array}\right] \qquad (!)$$

- *If* (!) *has no solutions,* $(\mathcal{S})$ *is insolvable.*

♠ **Observation:** *Inequality* (!) *has no solution iff* $\sum_{i=1}^m \lambda_i a_i = 0$ *and, in addition,*

- $\sum_{i=1}^m \lambda_i b_i \leq 0$ *when* $\sum_{i\in I} \lambda_i > 0$
- $\sum_{i=1}^m \lambda_i b_i < 0$ *when* $\sum_{i\in I} \lambda_i = 0$

♣ *We have arrived at*

Proposition: *Given system* $(\mathcal{S})$*, let us associate with it two systems of linear inequalities in variables* $\lambda_1, ..., \lambda_m$*:*

$$\text{(I)}: \begin{cases} \lambda_i \geq 0 \, \forall i \\ \sum_{i=1}^m \lambda_i a_i = 0 \\ \sum_{i=1}^m \lambda_i b_i \leq 0 \\ \sum_{i\in I} \lambda_i > 0 \end{cases}, \qquad \text{(II)}: \begin{cases} \lambda_i \geq 0 \, \forall i \\ \sum_{i=1}^m \lambda_i a_i = 0 \\ \sum_{i=1}^m \lambda_i b_i < 0 \\ \lambda_i = 0, i \in I \end{cases}$$

If at least one of the systems (I), (II) *has a solution, then* $(\mathcal{S})$ *has no solutions.*

General Theorem on Alternative: *Consider, along with system of linear inequalities*

$$a_i^T x \begin{cases} < b_i, & i \in I \\ \le b_i, & i \in \overline{I} \end{cases} \qquad (\mathcal{S})$$

in variables $x \in \mathbb{R}^n$, two systems of linear inequalities in variables $\lambda \in \mathbb{R}^m$:

$$\text{(I)}: \begin{cases} \lambda_i \ge 0 \, \forall i \\ \sum_{i=1}^m \lambda_i a_i = 0 \\ \sum_{i=1}^m \lambda_i b_i \le 0 \\ \sum_{i \in I} \lambda_i > 0 \end{cases}, \qquad \text{(II)}: \begin{cases} \lambda_i \ge 0 \, \forall i \\ \sum_{i=1}^m \lambda_i a_i = 0 \\ \sum_{i=1}^m \lambda_i b_i < 0 \\ \lambda_i = 0, i \in I \end{cases}$$

System $(\mathcal{S})$ has no solutions if and only if at least one of the systems (I), (II) *has a solution.*

Remark: Strict inequalities in $(\mathcal{S})$ in fact do not participate in (II). As a result, (II) *has a solution iff the "nonstrict" subsystem*

$$a_i^T x \le b_i, \; i \in \overline{I} \qquad (\mathcal{S}')$$

of $(\mathcal{S})$ has no solutions.

Remark: GTA says that a finite system of linear inequalities has no solutions if and only if (one of two) other systems of linear inequalities has a solution. Such a solution can be considered as a certificate of insolvability of $(\mathcal{S})$: $(\mathcal{S})$ is insolvable if and only if such an insolvability certificate exists.

$$a_i^T x \begin{cases} < b_i, & i \in I \\ \le b_i, & i \in \overline{I} \end{cases} \qquad (\mathcal{S})$$

$$(\mathrm{I}): \begin{cases} \lambda_i \ge 0 \, \forall i \\ \sum_{i=1}^m \lambda_i a_i = 0 \\ \sum_{i=1}^m \lambda_i b_i \le 0 \\ \sum_{i \in I} \lambda_i > 0 \end{cases}, \qquad (\mathrm{II}): \begin{cases} \lambda_i \ge 0 \, \forall i \\ \sum_{i=1}^m \lambda_i a_i = 0 \\ \sum_{i=1}^m \lambda_i b_i < 0 \\ \lambda_i = 0, i \in I \end{cases}$$

Proof of GTA: In one direction: "If (I) or (II) has a solution, then $(\mathcal{S})$ has no solutions" the statement is already proved. Now assume that $(\mathcal{S})$ has no solutions, and let us prove that one of the systems (I), (II) has a solution. Consider the system of homogeneous linear inequalities in variables x, t, ϵ:

$$\begin{array}{rrcl} a_i^T x & -b_i t & \le & 0, i \in I \\ a_i^T x & -b_i t & \le & 0, \ i \in \overline{I} \\ & -t & \le & 0 \\ & -\epsilon & < & 0 \end{array}$$

We claim that *this system has no solutions.* Indeed, assuming that the system has a solution $\bar{x}, \bar{t}, \bar{\epsilon}$, we have $\bar{\epsilon} > 0$, whence

$$\bar{t} > 0 \ \& \ a_i^T \bar{x} < b_i \bar{t}, i \in I \ \& \ a_i^T \bar{x} \le b_i \bar{t}, \ i \in \overline{I},$$

$\Rightarrow x = \bar{x}/\bar{t}$ is well defined and solves *un*solvable system $(\mathcal{S})$, which is impossible.

Situation: System

$$\begin{array}{rrrcl} a_i^T x & -b_i t & +\epsilon & \leq & 0, i \in I \\ a_i^T x & -b_i t & & \leq & 0, \, i \in \overline{I} \\ & -t & +\epsilon & \leq & 0 \\ & & -\epsilon & < & 0 \end{array}$$

has no solutions, or, equivalently, the homogeneous linear inequality

$$-\epsilon \geq 0$$

is a consequence of the system of homogeneous linear inequalities

$$\begin{array}{rrrcl} -a_i^T x & +b_i t & -\epsilon & \geq & 0, \, i \in I \\ -a_i^T x & +b_i t & & \geq & 0, \, i \in \overline{I} \\ & t & -\epsilon & \geq & 0 \end{array}$$

in variables x, t, ϵ. By Homogeneous Farkas Lemma, there exist $\mu_i \geq 0$, $1 \leq i \leq m$, $\mu \geq 0$ such that

$$\sum_{i=1}^m \mu_i a_i = 0 \;\; \& \;\; \sum_{i=1}^m \mu_i b_i + \mu = 0 \;\; \& \;\; \sum_{i \in I} \mu_i + \mu = 1$$

When $\mu > 0$, setting $\lambda_i = \mu_i / \mu$, we get

$$\lambda \geq 0, \textstyle\sum_{i=1}^m \lambda_i a_i = 0, \sum_{i=1}^m \lambda_i b_i = -1,$$

$\Rightarrow$ when $\sum_{i \in I} \lambda_i > 0$, λ solves (I), otherwise λ solves (II).

When $\mu = 0$, setting $\lambda_i = \mu_i$, we get

$$\lambda \geq 0, \textstyle\sum_i \lambda_i a_i = 0, \sum_i \lambda_i b_i = 0, \sum_{i \in I} \lambda_i = 1,$$

and λ solves (I). $\square$

♣ GTA is equivalent to the following

Principle: *A finite system of linear inequalities has no solution iff one can get, as a legitimate (i.e., compatible with the common rules of operating with inequalities) weighted sum of inequalities from the system, a contradictory inequality, i.e., either inequality* $0^T x \leq -1$, *or the inequality* $0^T x < 0$.

The advantage of this Principle is that it does not require converting the system into a standard form. For example, to see that the system of linear constraints

$$\begin{array}{rcl} x_1 + 2x_2 & < & 5 \\ 2x_1 + 3x_2 & \geq & 3 \\ 3x_1 + 4x_2 & = & 1 \end{array}$$

has no solutions, it suffices to take the weighted sum of these constraints with the weights $-1, 2, -1$, thus arriving at the contradictory inequality

$$0 \cdot x_1 + 0 \cdot x_2 > 0$$

♣ Specifying the system in question and applying GTA, we can obtain various particular cases of GTA, e.g., as follows:

Inhomogeneous Farkas Lemma: *A nonstrict linear inequality*

$$a^T x \le \alpha \qquad (!)$$

is a consequence of a solvable system of nonstrict linear inequalities

$$a_i^T x \le b_i,\ 1 \le i \le m \qquad (S)$$

if and only if (!) *can be obtained by taking weighted sum, with nonnegative coefficients, of the inequalities from the system and the identically true inequality* $0^T x \le 1$*:* (S) *implies* (!) *is and only if there exist nonnegative weights* $\lambda_0, \lambda_1, ..., \lambda_m$ *such that*

$$\lambda_0 \cdot [1 - 0^T x] + \sum_{i=1}^m \lambda_i [b_i - a_i^T x] \equiv \alpha - a^T x,$$

or, which is the same, iff there exist nonnegative $\lambda_0, \lambda_1, ..., \lambda_m$ *such that*

$$\sum_{i=1}^m \lambda_i a_i = a,\ \sum_{i=1}^m \lambda_i b_i + \lambda_0 = \alpha$$

or, which again is the same, iff there exist nonnegative $\lambda_1, ..., \lambda_m$ *such that*

$$\sum_{i=1}^m \lambda_i a_i = a,\ \sum_{i=1}^m \lambda_i b_i \le \alpha.$$

$$a_i^T x \le b_i,\ 1 \le i \le m \qquad (S)$$
$$a^T x \le \alpha \qquad (!)$$

Proof. If (!) can be obtained as a weighted sum, with nonnegative coefficients, of the inequalities from (S) and the inequality $0^T x \le 1$, then (!) clearly is a corollary of (S) independently of whether (S) is or is not solvable. Now let (S) be solvable and (!) be a consequence of (S); we want to prove that (!) is a combination, with nonnegative weights, of the constraints from (S) and the constraint $0^T x \le 1$. Since (!) is a consequence of (S), the system

$$-a^T x < -\alpha,\ a_i^T x \le b_i,\ 1 \le i \le m \qquad (M)$$

has no solutions, whence, by GTA, a legitimate weighted sum of the inequalities from the system is contradictory, that is, there exist $\mu \ge 0$, $\lambda_i \ge 0$:

$$\begin{array}{l} -\mu a + \sum_{i=1}^m \lambda_i a_i = 0, \\ 0 \left\{ \begin{array}{ll} \ge & -\mu\alpha + \sum_{i=1}^m \lambda_i b_i \ \&\ \mu > 0 \\ > & -\mu\alpha + \sum_{i=1}^m \lambda_i b_i \ \&\ \mu = 0 \end{array} \right. \end{array} \qquad (!!)$$

Situation: the system

$$-a^T x < -\alpha,\ a_i^T x \le b_i\, 1 \le i \le m \qquad (M)$$

has no solutions, whence there exist $\mu \ge 0, \lambda \ge 0$ such that

$$\begin{array}{l} -\mu a + \sum_{i=1}^m \lambda_i a_i = 0, \\ 0 \begin{cases} \ge & -\mu\alpha + \sum_{i=1}^m \lambda_i b_i \ \&\ \mu > 0 \\ > & -\mu\alpha + \sum_{i=1}^m \lambda_i b_i \ \&\ \mu = 0 \end{cases} \end{array} \qquad (!!)$$

Claim: $\mu > 0$. Indeed, otherwise the inequality $-a^T x < -\alpha$ does not participate in the weighted sum of the constraints from (M) which is a contradictory inequality

$\Rightarrow$ (S) can be led to a contradiction by taking weighted sum of the constraints

$\Rightarrow$ (S) is infeasible, which is a contradiction.

- When $\mu > 0$, setting $\lambda_i = \mu_i/\mu$, we get from (!!)

$$\sum_i \lambda_i a_i = a \,\&\, \sum_{i=1}^m \lambda_i b_i \le \alpha. \qquad \square$$

Why GTA is a deep fact?

♣ Consider the system of four linear inequalities in variables u, v:

$$-1 \le u \le 1,\ -1 \le v \le 1$$

and let us derive its consequence as follows:

$$\begin{aligned}
& -1 \le u \le 1,\ -1 \le v \le 1 \\
\Rightarrow\ & u^2 \le 1, v^2 \le 1 \\
\Rightarrow\ & u^2 + v^2 \le 2 \\
\Rightarrow\ & u + v = 1 \cdot u + 1 \cdot v \le \sqrt{1^2 + 1^2}\sqrt{u^2 + v^2} \\
\Rightarrow\ & u + v \le \sqrt{2}\sqrt{2} = 2
\end{aligned}$$

We have derived from solvable system of nonstrict linear inequalities a consequence which is a nonstrict *linear* inequality, and the derivation was "highly nonlinear."

A statement which says that *every* derivation of this type can be replaced by just taking weighted sum of the original inequalities and the trivial inequality $0^T x \le 1$ is a deep statement indeed!

♣ For *every* system $\mathcal{S}$ of inequalities, linear or nonlinear alike, taking weighted sums of inequalities of the system and trivial – identically true – inequalities always results in a consequence of $\mathcal{S}$. However, GTA heavily exploits the fact that the inequalities of the original system and a consequence we are looking for are *linear.* Already for quadratic inequalities, the statement similar to GTA fails to be true. For example, the quadratic inequality

$$x^2 \leq 1 \qquad (!)$$

is a consequence of the system of linear (and thus quadratic) inequalities

$$-1 \leq x \leq 1 \qquad (*)$$

Nevertheless, (!) can*not* be represented as a weighted sum of the inequalities from $(*)$ and identically true linear and quadratic inequalities, like

$$0 \cdot x \leq 1, x^2 \geq 0, x^2 - 2x + 1 \geq 0, ...$$

Answering Questions

♣ *How to certify that a polyhedral set* $X = \{x \in \mathbb{R}^n : Ax \leq b\}$ *is empty/nonempty?*

♠ A certificate for X to be *nonempty* is a solution $\bar{x}$ to the system $Ax \leq b$.

♠ A certificate for X to be *empty* is a solution $\bar{\lambda}$ to the system $\lambda \geq 0, A^T\lambda = 0, b^T\lambda < 0$.

In both cases, X possesses the property in question iff it can be certified as explained above ("the certification schemes are complete").

Note: *All certification schemes to follow are complete!*

Examples: • The vector $x = [1; ...; 1] \in \mathbb{R}^n$ certifies that the polyhedral set

$$X = \begin{array}{l}\{x \in \mathbb{R}^n : x_1 + x_2 \le 2, x_2 + x_3 \le 2, ..., x_n + x_1 \le 2, \\ -x_1 - ... - x_n \le -n\}\end{array}$$

is nonempty.

• The vector $\lambda = [1; 1; ...; 1; 2] \in \mathbb{R}^{n+1} \ge 0$ certifies that the polyhedral set

$$X = \begin{array}{l}\{x \in \mathbb{R}^n : x_1 + x_2 \le 2, x_2 + x_3 \le 2, ..., x_n + x_1 \le 2, \\ -x_1 - ... - x_n \le -n - 0.01\}\end{array}$$

is empty. Indeed, summing up the $n+1$ constraints defining $X = \{x : Ax \le b\}$ with weights λ_i, we get the contradictory inequality

$$\underbrace{0 \equiv 2(x_1 + ... + x_n) - 2[x_1 + ... + x_n]}_{[A^T\lambda]^T x \equiv 0}$$

$$\le \underbrace{2n - 2(n + 0.01) = -0.02}_{b^T\lambda = -0.02 < 0}$$

♣ *How to certify that a linear inequality* $c^T x \leq d$ *is violated somewhere on a polyhedral set*

$$X = \{x \in \mathbb{R}^n : Ax \leq b\},$$

that is, the inequality is not a consequence of the system $Ax \leq b$*?*

A certificate is $\bar{x}$ such that $A\bar{x} \leq b$ and $c^T \bar{x} > d$.

♣ *How to certify that a linear inequality $c^Tx \le d$ is satisfied everywhere on a polyhedral set*

$$X = \{x \in \mathbb{R}^n : Ax \le b\},$$

that is, the inequality is a consequence of the system $Ax \le b$?

♠ The situation in question arises in two cases:

A. X is empty, the target inequality is an arbitrary one

B. X is nonempty, the target inequality is a consequence of the system $Ax \le b$

Consequently, to certify the fact in question means

— either to certify that X is empty, the certificate being λ such that $\lambda \ge 0, A^T\lambda = 0, b^T\lambda < 0$,

— or to certify that X is nonempty by pointing out a solution $\bar{x}$ to the system $Ax \le b$ *and* to certify the fact that $c^Tx \le d$ is a consequence of the solvable system $Ax \le b$ by pointing out a λ which satisfies the system $\lambda \ge 0, A^T\lambda = c, b^T\lambda \le d$ (we have used Inhomogeneous Farkas Lemma).

Note: In the second case, we can omit the necessity to certify that $X \neq \emptyset$, since the existence of λ satisfying $\lambda \geq 0, A^T\lambda = c, b^T\lambda \leq d$ *always is sufficient* for $c^T x \leq d$ to be a consequence of $Ax \leq b$.

Example:• To certify that the linear inequality

$$c^T x := x_1 + ... + x_n \leq d := n - 0.01$$

is violated somewhere on the polyhedral set

$$\begin{array}{rcl} X & = & \{x \in \mathbb{R}^n : x_1 + x_2 \leq 2, x_2 + x_3 \leq 2, ..., x_n + x_1 \leq 2\} \\ & = & \{x : Ax \leq b\} \end{array}$$

it suffices to note that $x = [1; ...; 1] \in X$ and $n = c^T x > d = n - 0.01$

• To certify that the linear inequality

$$c^T x := x_1 + ... + x_n \leq d := n$$

is satisfied everywhere on the above X, it suffices to note that when taking weighted sum of inequalities defining X, the weights being $1/2$, we get the target inequality.

Equivalently: for $\lambda = [1/2; ...; 1/2] \in \mathbb{R}^n$ it holds $\lambda \geq 0, A^T \lambda = [1; ...; 1] = c, b^T \lambda = n \leq d$

♣ *How to certify that a polyhedral set* $X = \{x \in \mathbb{R}^n : Ax \leq b\}$ *does not contain a polyhedral set* $Y = \{x \in \mathbb{R}^n : Cx \leq d\}$*?*

A certificate is a point $\bar{x}$ such that $C\bar{x} \leq d$ (i.e., $\bar{x} \in Y$) and $\bar{x}$ does *not* solve the system $Ax \leq b$ (i.e., $\bar{x} \notin X$).

♣ *How to certify that a polyhedral set*

$$X = \{x \in \mathbb{R}^n : Ax \leq b\}$$

contains a polyhedral set

$$Y = \{x \in \mathbb{R}^n : Cx \leq d\}?$$

This situation arises in two cases:

— $Y = \emptyset$, X is arbitrary. To certify that this is the case, it suffices to point out λ such that $\lambda \geq 0, C^T\lambda = 0, d^T\lambda < 0$

— Y is nonempty and every one of the m linear inequalities $a_i^T x \leq b_i$ defining X is satisfied everywhere on Y. To certify that this is the case, it suffices to point out $\bar{x}$, $\lambda^1, ..., \lambda^m$ such that

$$C^T\bar{x} \leq d \;\&\; \lambda^i \geq 0, C^T\lambda_i = a_i, d^T\lambda^i \leq b_i,\ 1 \leq i \leq m.$$

Note: Same as above, we can omit the necessity to point out $\bar{x}$.

Examples. • To certify that the set

$$Y = \{x \in \mathbb{R}^3 : -2 \le x_1 + x_2 \le 2, -2 \le x_2 + x_3 \le 2, \\ -2 \le x_3 + x_1 \le 2\}$$

is *not* contained in the box

$$X = \{x \in \mathbb{R}^3 : |x_i| \le 2, 1 \le i \le 3\},$$

it suffices to note that the vector $\bar{x} = [3; -1; -1]$ belongs to Y and does not belong to X.

• To certify that the above Y is contained in

$$X' = \{x \in \mathbb{R}^3 : |x_i| \le 3, 1 \le i \le 3\}$$

note that summing up the 6 inequalities

$$x_1 + x_2 \le 2,\ -x_1 - x_2 \le 2, x_2 + x_3 \le 2, -x_2 - x_3 \le 2, \\ x_3 + x_1 \le 2, -x_3 - x_1 \le 2$$

defining Y with the nonnegative weights

$$\lambda_1 = 1, \lambda_2 = 0, \lambda_3 = 0, \lambda_4 = 1, \lambda_5 = 1, \lambda_6 = 0$$

we get

$$[x_1 + x_2] + [-x_2 - x_3] + [x_3 + x_1] \le 6 \Rightarrow x_1 \le 3$$

— with the nonnegative weights

$$\lambda_1 = 0, \lambda_2 = 1, \lambda_3 = 1, \lambda_4 = 0, \lambda_5 = 0, \lambda_6 = 1$$

we get

$$[-x_1 - x_2] + [x_2 + x_3] + [-x_3 - x_1] \le 6 \Rightarrow -x_1 \le 3$$

The inequalities $-3 \le x_2, x_3 \le 3$ can be obtained similarly $\Rightarrow Y \subset X'$.

♣ *How to certify that a polyhedral set* $Y = \{x \in \mathbb{R}^n : Ax \leq b\}$ *is bounded/unbounded?*

• Y is bounded iff for properly chosen R it holds

$$Y \subset X_R = \{x : |x_i| \leq R,\ 1 \leq i \leq n\}$$

To certify this means

— either to certify that Y is empty, the certificate being λ: $\lambda \geq 0, A^T\lambda = 0, b^T\lambda < 0$,

— or to point out vectors R and vectors $\lambda^i_\pm$ such that $\lambda^i_\pm \geq 0, A^T\lambda^i_\pm = \pm e_i, b^T\lambda^i_\pm \leq R$ for all i. Since R can be chosen arbitrary large, the latter amounts to pointing out vectors $\lambda^i_\pm$ such that $\lambda^i_\pm \geq 0,\ A^T\lambda^i_\pm = \pm e_i$, $i = 1, ..., n$.

• Y is *un*bounded iff Y is nonempty and the recessive cone $\mathrm{Rec}(Y) = \{x : Ax \leq 0\}$ is nontrivial. To certify that this is the case, it suffices to point out $\bar{x}$ satisfying $A\bar{x} \leq b$ and $\bar{d}$ satisfying $\bar{d} \neq 0, A\bar{d} \leq 0$.

Note: *When* $Y = \{x \in \mathbb{R}^n : Ax \leq b\}$ *is known to be nonempty, its boundedness/unboundedness is independent of the particular value of* b*!*

Examples: • To certify that the set

$$\begin{aligned} X &= \{x \in \mathbb{R}^3 : -2 \le x_1 + x_2 \le 2, -2 \le x_2 + x_3 \le 2, \\ &\quad -2 \le x_3 + x_1 \le 2\} \end{aligned}$$

is bounded, it suffices to certify that it belongs to the box $\{x \in \mathbb{R}^3 : |x_i| \le 3, 1 \le i \le 3\}$, which was already done.

• To certify that the set

$$\begin{aligned} X &= \{x \in \mathbb{R}^4 : -2 \le x_1 + x_2 \le 2, -2 \le x_2 + x_3 \le 2, \\ &\quad -2 \le x_3 + x_4 \le 2, -2 \le x_4 + x_1 \le 2\} \end{aligned}$$

is *un*bounded, it suffices to note that the vector $\bar{x} = [0; 0; 0; 0]$ belongs to X, and the vector $\bar{d} = [1; -1; 1; -1]$ when plugged into the inequalities defining X make the bodies of the inequalities zero and thus is a recessive direction of X.

Certificates in LO

♣ Consider LO program in the form

$$\mathrm{Opt} = \max_x \left\{ c^T x : \left\{ \begin{array}{rcll} Px & \leq & p & (\ell) \\ Qx & \geq & q & (g) \\ Rx & = & r & (e) \end{array} \right\} \right. \qquad (P)$$

♠ *How to certify that (P) is feasible/infeasible?*

• To certify that (P) is feasible, it suffices to point out a feasible solution $\bar{x}$ to the program.

• To certify that (P) is *in*feasible, it suffices to point out aggregation weights $\lambda_\ell, \lambda_g, \lambda_e$ such that

$$\begin{gathered} \lambda_\ell \geq 0, \lambda_g \leq 0 \\ P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e = 0 \\ p^T\lambda_\ell + q^T\lambda_g + r^T\lambda_e < 0 \end{gathered}$$

$$\mathrm{Opt} = \max_x \left\{ c^T x : \left\{ \begin{array}{rcll} Px & \leq & p & (\ell) \\ Qx & \geq & q & (g) \\ Rx & = & r & (e) \end{array} \right\} \right. \qquad (P)$$

♠ *How to certify that (P) is bounded/unbounded?*

• (P) is bounded if either (P) is infeasible, or (P) is feasible and there exists a such that the inequality $c^T x \leq a$ is consequence of the system of constraints. Consequently, to certify that (P) is bounded, we should

— either point out an infeasibility certificate $\lambda_\ell \geq 0, \lambda_g \leq 0, \lambda_e : P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e = 0, p^T\lambda_\ell + q^T\lambda_g + r^T\lambda_e < 0$ for (P),

— or point out a feasible solution $\bar{x}$ and $a, \lambda_\ell, \lambda_g, \lambda_e$ such that

$$\lambda_\ell \geq 0, \lambda_g \leq 0 \ \& \ P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e = c \\ \& \ p^T\lambda_\ell + q^T\lambda_g + r^T\lambda_e \leq a$$

which, since a can be arbitrary, amounts to

$$\lambda_\ell \geq 0, \lambda_g \leq 0, P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e = c$$

Note: We can skip the necessity to certify that (P) is feasible.

$$\mathrm{Opt} = \max_x \left\{ c^T x : \left\{ \begin{array}{rcll} Px & \leq & p & (\ell) \\ Qx & \geq & q & (g) \\ Rx & = & r & (e) \end{array} \right\} \right. \qquad (P)$$

• (P) is *un*bounded iff (P) is feasible and there is a recessive direction d such that $c^T d > 0$

$\Rightarrow$ to certify that (P) is unbounded, we should point out a feasible solution $\bar{x}$ to (P) *and* a vector d such that

$$Pd \leq 0, Qd \geq 0, Rd = 0, c^T d > 0.$$

Note: *If (P) is known to be feasible, its boundedness/unboundedness is independent of a particular value of $[p; q; r]$.*

$$\mathrm{Opt} = \max_x \left\{ c^T x : \left\{ \begin{array}{rcll} Px & \leq & p & (\ell) \\ Qx & \geq & q & (g) \\ Rx & = & r & (e) \end{array} \right\} \right. \qquad (P)$$

♠ *How to certify that* $\mathrm{Opt} \geq a$ *for a given* $a \in \mathbb{R}$*?*

a certificate is a feasible solution $\bar{x}$ with $c^T x \geq a$.

♠ *How to certify that* $\mathrm{Opt} \leq a$ *for a given* $a \in \mathbb{R}$*?*

$\mathrm{Opt} \leq a$ iff the linear inequality $c^T x \leq a$ is a consequence of the system of constraints. To certify this, we should

— either point out an infeasibility certificate $\lambda_\ell, \lambda_g, \lambda_e$:

$$\begin{array}{c} \lambda_\ell \geq 0, \lambda_g \leq 0, \\ P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e = 0, \\ p^T\lambda_e + q^T\lambda_g + r^T\lambda_e < 0 \end{array}$$

for (P),

— or point out $\lambda_\ell, \lambda_g, \lambda_e$ such that

$$\begin{array}{ll} \lambda_\ell \geq 0, \lambda_g \leq 0 & (a) \\ P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e = c & (b) \\ p^T\lambda_\ell + q^T\lambda_g + r^T\lambda_e \leq a & (c) \end{array}$$

Note: *Whenever* λ_ℓ, λ_g, λ_e *satisfy* (a) *and* (b), *the left hand side in* (c) *upper-bounds* $c^T x$ *for every* x *feasible for* (P)

⇒ *If* $c^T x \leq a$ *for every feasible* x *and* $c^T\bar{x} = a$ *for some feasible* $\bar{x}$, *then "*$\leq$*" in* (c) *is in fact "=".*

$$\mathrm{Opt} = \max_x \left\{ c^T x : \left\{ \begin{array}{rcll} Px & \leq & p & (\ell) \\ Qx & \geq & q & (g) \\ Rx & = & r & (e) \end{array} \right\} \right. \qquad (P)$$

♣ *How to certify that $\bar{x}$ is an optimal solution to (P)?*

• $\bar{x}$ is optimal solution iff it is feasible and $\mathrm{Opt} \leq c^T \bar{x}$. The latter amounts to existence of $\lambda_\ell, \lambda_g, \lambda_e$ such that

$$\overbrace{\lambda_\ell \geq 0, \lambda_g \leq 0}^{(a)} \;\&\; \overbrace{P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e = c}^{(b)}$$
$$\&\; \underbrace{p^T\lambda_\ell + q^T\lambda_g + r^T\lambda_e = c^T\bar{x}}_{(c)}$$

Multiplying both sides in (b) by $\bar{x}^T$ and subtracting the resulting equality from (c) results in

$$0 = \underbrace{\lambda_\ell^T}_{\geq 0} \overbrace{[p - P\bar{x}]}^{\geq 0} + \underbrace{\lambda_g^T}_{\leq 0} \overbrace{[q - Q\bar{x}]}^{\leq 0} + \lambda_e^T \overbrace{[r - R\bar{x}]}^{=0}$$

which is possible iff $(\lambda_\ell)_i[p_i - (P\bar{x})_i] = 0$ for all i and $(\lambda_g)_j[q_j - (Q\bar{x})_j] = 0$ for all j.

$$\mathrm{Opt} = \max_x \left\{ c^T x : \left\{ \begin{array}{rcll} Px & \leq & p & (\ell) \\ Qx & \geq & q & (g) \\ Rx & = & r & (e) \end{array} \right\} \right. \qquad (P)$$

♠ We have arrived at the *Karush-Kuhn-Tucker Optimality Conditions in LO:*

A feasible solution $\bar{x}$ to (P) is optimal iff the constraints of (P) can be assigned with vectors of Lagrange multipliers $\lambda_\ell, \lambda_g, \lambda_e$ in such a way that

- [signs of multipliers] *Lagrange multipliers associated with $\leq$-constraints are nonnegative, and Lagrange multipliers associated with $\geq$-constraints are nonpositive,*
- [complementary slackness] *Lagrange multipliers associated with non-active at $\bar{x}$ constraints are zero, and*
- [KKT equation] *One has*

$$P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e = c \qquad .$$

Example. To certify that the feasible solution $\bar{x} = [1; ...; 1] \in \mathbb{R}^n$ to the LO program

$$\max_x \left\{ x_1 + ... + x_n : \begin{array}{llll} x_1 + x_2 \le 2, & x_2 + x_3 \le 2 & , ..., & x_n + x_1 \le 2 \\ x_1 + x_2 \ge -2, & x_2 + x_3 \ge -2 & , ..., & x_n + x_1 \ge -2 \end{array} \right\}$$

is optimal, it suffices to assign the constraints with Lagrange multipliers $\lambda_\ell = [1/2; 1/2; ...; 1/2]$, $\lambda_g = [0; ...; 0]$ and to note that

$$\begin{array}{c} \\ P^T\lambda_\ell + Q^T\lambda_g = \end{array} \begin{array}{c} \lambda_\ell \ge 0, \lambda_g \le 0 \\ \left[\begin{array}{ccccc} 1 & & & & 1 \\ 1 & 1 & & & \\ & 1 & 1 & & \\ & & 1 & \ddots & \\ & & & \ddots & 1 \\ & & & 1 & 1 \end{array} \right] \end{array} \lambda_\ell = c := [1; ...; 1]$$

and complementary slackness takes place.

♣ **Application:** Faces of polyhedral set revisited. Recall that a face of a nonempty polyhedral set

$$X = \{x \in \mathbb{R}^n : a_i^T x \le b_i, 1 \le i \le m\}$$

is a nonempty set of the form

$$X_I = \{x \in \mathbb{R}^n : a_i^T x = b_i, i \in I, a_i^T x \le b_i, i \notin I\}$$

This definition is *not* geometric.

Geometric characterization of faces:

(i) *Let $c^T x$ be a linear function bounded from above on X. Then the set*

$$\mathrm{Argmax}_X\, c^T x := \{x \in X : c^T x = \max_{x' \in X} c^T x'\}$$

is a face of X. In particular, if the maximizer of $c^T x$ over X exists and is unique, it is an extreme point of X.

(ii) *Vice versa, every face of X admits a representation as $\mathrm{Argmax}_{x \in X}\, c^T x$ for properly chosen c. In particular, every vertex of X is the unique maximizer, over X, of some linear function.*

Proof, (i): Let $c^T x$ be bounded from above on X. Then the set $X_* = \mathrm{Argmax}_{x \in X} c^T x$ is nonempty. Let $x_* \in X_*$. By KKT Optimality conditions, there exist $\lambda \geq 0$ such that

$$\textstyle\sum_i \lambda_i a_i = c \;\&\; \lambda_i > 0 \Rightarrow a_i^T x_* = b_i$$

Let $I_* = \{i : \lambda_i > 0\}$, so that

$$\text{(a): } \textstyle\sum_{i \in I_*} \lambda_i a_i = c \text{ and (b): } a_i^T x_* = b_i,\ i \in I_*.$$

We claim that

$$X_* = X_{I_*} := \{x \in X : a_i^T x = b_i,\ i \in I_*\}.$$

Indeed,

— $x \in X_{I_*} \Rightarrow c^T x = [\sum_{i \in I_*} \lambda_i a_i]^T x$ [by (a)]

$= \sum_{i \in I_*} \lambda_i a_i^T x$

$= \sum_{i \in I_*} b_i$ [by (b)]

$= \sum_{i \in I_*} \lambda_i a_i^T x_* = c^T x_*$, [by (b) and (a)]

$\Rightarrow x \in X_* := \mathrm{Argmax}_{y \in X} c^T y$, and

— $x \in X_* \Rightarrow c^T(x_* - x) = 0$

$\Rightarrow \sum_{i \in I_*} \lambda_i (a_i^T x_* - a_i^T x) = 0$ [by (a)]

$\Rightarrow \sum_{i \in I_*} \underbrace{\lambda_i}_{>0} (b_i - \underbrace{a_i^T x}_{\leq b_i}) = 0$ [by (b) and due to $x \in X$]

$\Rightarrow a_i^T x = b_i \, \forall i \in I_* \Rightarrow x \in X_{I_*}$.

Proof, (ii): Let $X_I = \{x \in X : a_i^T x = b_i, i \in I\}$ be a face of X, and let us set $c = \sum_{i \in I} a_i$. Same as above, it is immediately seen that $X_I = \mathrm{Argmax}_{x \in X} c^T x$.

LO Duality

♣ Consider an LO program

$$\mathrm{Opt}(P) = \max_x \left\{ c^T x : Ax \leq b \right\} \qquad (P)$$

The dual problem stems from the desire to bound from above the optimal value of the *primal* problem (P), To this end, we use our aggregation technique, specifically,

• *assign the constraints* $a_i^T x \leq b_i$ *with nonnegative aggregation weights* λ_i ("Lagrange multipliers") *and sum them up with these weights, thus getting the inequality*

$$[A^T\lambda]^T x \leq b^T\lambda \qquad (!)$$

Note: by construction, this inequality is a consequence of the system of constraints in (P) and thus is satisfied at every feasible solution to (P).

• *We may be lucky to get in the left hand side of* $(!)$ *exactly the objective* $c^T x$*:*

$$A^T\lambda = c.$$

In this case, $(!)$ *says that* $b^T\lambda$ *is an upper bound on* $c^T x$ *everywhere in the feasible domain of* (P)*, and thus* $b^T\lambda \geq \mathrm{Opt}(P)$*.*

$$\mathrm{Opt}(P) = \max_x \left\{ c^T x : Ax \le b \right\} \qquad (P)$$

♠ *We arrive at the problem of finding the best – the smallest – upper bound on* $\mathrm{Opt}(P)$ *achievable with our bounding scheme. This new problem is*

$$\mathrm{Opt}(D) = \min_\lambda \left\{ b^T \lambda : A^T \lambda = c, \lambda \ge 0 \right\}. \qquad (D)$$

It is called the problem dual to (P)*.*

♣ **Note:** *Our "bounding principle" can be applied to every LO program, independently of its format. For example, as applied to the primal LO program in the form*

$$\mathrm{Opt}(P) = \max_x \left\{ c^T x : \left\{ \begin{array}{rcll} Px & \le & p & (\ell) \\ Qx & \ge & q & (g) \\ Rx & = & r & (e) \end{array} \right\} \right. \qquad (P)$$

it leads to the dual problem in the form of

$$\mathrm{Opt}(D) = \min_{[\lambda_\ell;\lambda_g,\lambda_e]} \left\{ p^T \lambda_\ell + q^T \lambda_g + r^T \lambda_e : \left\{ \begin{array}{l} \lambda_\ell \ge 0, \lambda_g \le 0 \\ P^T \lambda_\ell + Q^T \lambda_g + R^T \lambda_e = c \end{array} \right\} \right. \qquad (D)$$

$$\mathrm{Opt}(P) = \max_x \left\{ c^T x : \left\{ \begin{array}{rcll} Px & \leq & p & (\ell) \\ Qx & \geq & q & (g) \\ Rx & = & r & (e) \end{array} \right\} \right. \qquad (P)$$

$$\mathrm{Opt}(D) = \min_{[\lambda_\ell;\lambda_g;\lambda_e]} \left\{ p^T\lambda_\ell + q^T\lambda_g + r^T\lambda_e : \left\{ \begin{array}{l} \lambda_\ell \geq 0, \lambda_g \leq 0 \\ P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e = c \end{array} \right\} \right. \qquad (D)$$

LO Duality Theorem: *Consider a primal LO program (P) along with its dual program (D). Then*

(i) [Primal-dual symmetry] *The duality is symmetric: (D) is an LO program, and the program dual to (D) is (equivalent to) the primal problem (P).*

(ii) [Weak duality] *We always have $\mathrm{Opt}(D) \geq Opt(P)$.*

(iii) [Strong duality] *The following 3 properties are equivalent to each other:*

- *one of the problems is feasible and bounded*
- *both problems are solvable*
- *both problems are feasible*

and whenever these equivalent to each other properties take place, we have

$$\mathrm{Opt}(P) = \mathrm{Opt}(D).$$

$$\mathrm{Opt}(P) = \max_x \left\{ c^T x : \left\{ \begin{array}{lcll} Px & \leq & p & (\ell) \\ Qx & \geq & q & (g) \\ Rx & = & r & (e) \end{array} \right\} \right. \qquad (P)$$

$$\mathrm{Opt}(D) = \min_{[\lambda_\ell;\lambda_g,\lambda_e]} \left\{ p^T\lambda_\ell + q^T\lambda_g + r^T\lambda_e : \left\{ \begin{array}{l} \lambda_\ell \geq 0, \lambda_g \leq 0 \\ P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e = c \end{array} \right\} \right. \qquad (D)$$

Proof of Primal-Dual Symmetry: *We rewrite* (D) *is exactly the same form as* (P)*, that is, as*

$$-\mathrm{Opt}(D) = \max_{[\lambda_\ell;\lambda_g,\lambda_e]} \left\{ -p^T\lambda_\ell - q^T\lambda_g - r^T\lambda_e : \left\{ \begin{array}{l} \lambda_g \leq 0, \lambda_\ell \geq 0 \\ P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e = c \end{array} \right\} \right.$$

and apply the recipe for building the dual, resulting in

$$\min_{[x_\ell;x_g;x_e]} \left\{ c^T x_e : \left\{ \begin{array}{l} x_\ell \geq 0, x_g \leq 0 \\ Px_e + x_g = -p \\ Qx_e + x_\ell = -q \\ Rx_e = -r \end{array} \right\} \right.$$

whence, setting $x_e = -x$ *and eliminating* x_g *and* x_e*, the problem dual to dual becomes*

$$\min_x \left\{ -c^T x : Px \leq p, Qx \geq q, Rx = r \right\}$$

which is equivalent to (P)*.* □

$$\mathrm{Opt}(P) = \max_x \left\{ c^T x : \left\{ \begin{array}{lcll} Px & \leq & p & (\ell) \\ Qx & \geq & q & (g) \\ Rx & = & r & (e) \end{array} \right\} \right. \qquad (P)$$

$$\mathrm{Opt}(D) = \min_{[\lambda_\ell;\lambda_g,\lambda_e]} \left\{ p^T\lambda_\ell + q^T\lambda_g + r^T\lambda_e : \left\{ \begin{array}{l} \lambda_\ell \geq 0, \lambda_g \leq 0 \\ P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e = c \end{array} \right\} \right. \qquad (D)$$

Proof of Weak Duality $\mathrm{Opt}(D) \geq \mathrm{Opt}(P)$**:** *by construction of the dual.*

$$\mathrm{Opt}(P) = \max_x \left\{ c^T x : \left\{ \begin{array}{rcll} Px & \leq & p & (\ell) \\ Qx & \geq & q & (g) \\ Rx & = & r & (e) \end{array} \right\} \right. \qquad (P)$$

$$\mathrm{Opt}(D) = \min_{[\lambda_\ell;\lambda_g,\lambda_e]} \left\{ p^T\lambda_\ell + q^T\lambda_g + r^T\lambda_e : \left\{ \begin{array}{l} \lambda_\ell \geq 0, \lambda_g \leq 0 \\ P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e = c \end{array} \right\} \right. \qquad (D)$$

Proof of Strong Duality:

Main Lemma: *Let one of the problems (P), (D) be feasible and bounded. Then both problems are solvable with equal optimal values.*

Proof of Main Lemma: By Primal-Dual Symmetry, we can assume w.l.o.g. that the feasible and bounded problem is (P). By what we already know, (P) is solvable. Let us prove that (D) is solvable, and the optimal values are equal to each other.

• Observe that the linear inequality $c^T x \leq \mathrm{Opt}(P)$ is a consequence of the (solvable!) system of constraints of (P). By Inhomogeneous Farkas Lemma

$$\exists \lambda_\ell \geq 0, \lambda_g \leq 0, \lambda_e : \\ P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e = c \ \& \ p^T\lambda_\ell + q^T\lambda_g + r^T\lambda_e \leq \mathrm{Opt}(P).$$

$\Rightarrow$ *λ is feasible for (D) with the value of dual objective $\leq$*$\mathrm{Opt}(P)$*. By Weak Duality, this value should be $\geq$*$\mathrm{Opt}(P)$

$\Rightarrow$ *the dual objective at λ equals to* $\mathrm{Opt}(P)$

$\Rightarrow$ *λ is dual optimal and* $\mathrm{Opt}(D) = \mathrm{Opt}(P)$. $\square$

Main Lemma $\Rightarrow$ Strong Duality:

• By Main Lemma, if one of the problems (P), (D) is feasible and bounded, then both problems are solvable with equal optimal values

• If both problems are solvable, then both are feasible

• If both problems are feasible, then both are bounded by Weak Duality, and thus one of them (in fact, both of them) is feasible and bounded.

Immediate Consequences

$$\mathrm{Opt}(P) = \max_x \left\{ c^T x : \left\{ \begin{array}{rcll} Px & \leq & p & (\ell) \\ Qx & \geq & q & (g) \\ Rx & = & r & (e) \end{array} \right\} \right. \qquad (P)$$

$$\mathrm{Opt}(D) = \min_{[\lambda_\ell;\lambda_g,\lambda_e]} \left\{ p^T\lambda_\ell + q^T\lambda_g + r^T\lambda_e : \left\{ \begin{array}{l} \lambda_\ell \geq 0, \lambda_g \leq 0 \\ P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e = c \end{array} \right\} \right. \qquad (D)$$

♣ **Optimality Conditions in LO:** *Let x and $\lambda = [\lambda_\ell; \lambda_g; \lambda_e]$ be a pair of feasible solutions to (P) and (D). This pair is composed of optimal solutions to the respective problems*

- [zero duality gap] *if and only if the duality gap, as evaluated at this pair, vanishes:*

$$\begin{array}{rcl} \mathrm{DualityGap}(x,\lambda) & := & [p^T\lambda_\ell + q^T\lambda_g + r^T\lambda_e] - c^T x \\ & = & 0 \end{array}$$

- [complementary slackness] *if and only if the products of all Lagrange multipliers λ_i and the residuals in the corresponding primal constrains are zero:*

$$\forall i : [\lambda_\ell]_i [p - Px]_i = 0 \ \& \ \forall j : [\lambda_g]_j [q - Qx]_j = 0.$$

Proof: We are in the situation when both problems are feasible and thus both are solvable with equal optimal values. Therefore

$$\begin{array}{rcl}\mathsf{DualityGap}(x,\lambda) & := & \big[[p^T\lambda_\ell + q^T\lambda_g + r^T\lambda_e] - \mathsf{Opt}(D)\big] \\ & & +\big[\mathsf{Opt}(P) - c^Tx\big]\end{array}$$

For a primal-dual pair of feasible solutions the expressions in the magenta and the red brackets are nonnegative

$\Rightarrow$ *Duality Gap, as evaluated at a primal-dual feasible pair, is nonnegative and can vanish iff both the expressions in the magenta and the red brackets vanish, that is, iff x is primal optimal and λ is dual optimal.*

- Observe that

$$\begin{array}{l}\mathsf{DualityGap}(x,\lambda) = [p^T\lambda_\ell + q^T\lambda_g + r^T\lambda_e] - c^Tx \\ = [p^T\lambda_\ell + q^T\lambda_g + r^T\lambda_e] - [P^T\lambda_\ell + Q^T\lambda_g + R^T\lambda_e]^Tx \\ = \lambda_\ell^T[p - Px] + \lambda_g^T[q - Qx] + \lambda_e^T[r - Rx] \\ = \sum_i[\lambda_\ell]_i[p - Px]_i + \sum_j[\lambda_g]_j[q - Qx]_j\end{array}$$

Al terms in the resulting sums are nonnegative

$\Rightarrow$ *Duality Gap vanishes iff the complementary slackness holds true.*

Geometry of a Primal-Dual Pair of LO Programs

$$\mathrm{Opt}(P) = \max_x \left\{ c^T x : \left\{ \begin{array}{rcll} Px & \leq & p & (\ell) \\ Rx & = & r & (e) \end{array} \right\} \right. \qquad (P)$$

$$\mathrm{Opt}(D) = \min_{[\lambda_\ell;\lambda_e]} \left\{ p^T\lambda_\ell + r^T\lambda_e : \left\{ \begin{array}{l} \lambda_\ell \geq 0 \\ P^T\lambda_\ell + R^T\lambda_e = c \end{array} \right\} \right. \qquad (D)$$

Standing Assumption: *The systems of linear equations in (P), (D) are solvable:*

$$\exists \bar{x}, \bar{\lambda} = [\bar{\lambda}_\ell; \bar{\lambda}_e] : R\bar{x} = r,\ P^T\bar{\lambda}_\ell + R^T\bar{\lambda}_e = -c$$

♣ **Observation:** Whenever $Rx = r$, we have

$$\begin{array}{rcl} c^T x & = & -[P^T\bar{\lambda}_\ell + R^T\bar{\lambda}_e]^T x = -\bar{\lambda}_\ell^T[Px] - \bar{\lambda}_e^T[Rx] \\ & = & \bar{\lambda}_\ell^T[p - Px] + \left[-\bar{\lambda}_\ell^T p - \bar{\lambda}_e^T r\right] \end{array}$$

$\Rightarrow$ (P) is equivalent to the problem

$$\max_x \left\{ \bar{\lambda}_\ell^T[p - Px] : p - Px \geq 0, Rx = r \right\}.$$

♠ Passing to the new variable ("primal slack") $\xi = p - Px$, the problem becomes

$$\max_\xi \left\{ \bar{\lambda}_\ell^T \xi : \xi \geq 0,\ \xi \in \mathcal{M}_P = \bar{\xi} + \mathcal{L}_P \right\}$$

$$\left[\begin{array}{rcl} \mathcal{L}_P & = & \{\xi = Px : Rx = 0\} \\ \bar{\xi} & = & p - P\bar{x} \end{array} \right]$$

$\mathcal{M}_P$: primal feasible affine plane

$$\mathrm{Opt}(P) = \max_x \left\{ c^T x : \left\{ \begin{array}{rcll} Px & \leq & p & (\ell) \\ Rx & = & r & (e) \end{array} \right\} \right. \qquad (P)$$

$$\mathrm{Opt}(D) = \min_{[\lambda_\ell;\lambda_e]} \left\{ p^T\lambda_\ell + r^T\lambda_e : \left\{ \begin{array}{l} \lambda_\ell \geq 0 \\ P^T\lambda_\ell + R^T\lambda_e = c \end{array} \right\} \right. \qquad (D)$$

♣ Let us express (D) in terms of the *dual slack* λ_ℓ. If $[\lambda_\ell; \lambda_e]$ satisfy the equality constraints in (D), then

$$\begin{array}{l} p^T\lambda_\ell + r^T\lambda_e = p^T\lambda_\ell + [R\bar{x}]^T\lambda_e = p^T\lambda_\ell + \bar{x}^T[R^T\lambda_e] \\ = p^T\lambda_\ell + \bar{x}^T[c - P^T\lambda_\ell] = [p - P\bar{x}]^T\lambda_\ell + \bar{x}^T c \\ = \bar{\xi}^T\lambda_\ell + \bar{x}^T c \end{array}$$

Besides this,

$$\begin{array}{ll} & P^T\lambda_\ell + R^T\lambda_e = c \\ \Leftrightarrow & P^T\lambda_\ell + R^T\lambda_e = -P^T\bar{\lambda}_\ell - R^T\bar{\lambda}_e \\ \Leftrightarrow & P^T[\lambda_\ell + \bar{\lambda}_\ell] + R^T[\lambda_e + \bar{\lambda}_e] = 0 \\ \Leftrightarrow & \lambda_\ell \in \mathcal{M}_D := \mathcal{L}_D - \bar{\lambda}_\ell, \\ & \mathcal{L}_D = \{\mu_\ell : \exists \mu_e : P^T\mu_\ell + R^T\mu_e = 0\}. \end{array}$$

$\Rightarrow$ (D) is equivalent to the problem

$$\min_{\lambda_\ell} \left\{ \bar{\xi}^T\lambda_\ell : \lambda_\ell \geq 0, \lambda_\ell \in \mathcal{M}_D = \mathcal{L}_D - \bar{\lambda}_\ell \right\}$$

$$\left[\mathcal{L}_D = \{\mu_\ell : \exists \mu_e : P^T\mu_\ell + R^T\mu_e = 0\} \right]$$

$\mathcal{M}_D$: dual feasible affine plane

$$\mathrm{Opt}(P) = \max_x \left\{ c^T x : \left\{ \begin{array}{rcll} Px & \leq & p & (\ell) \\ Rx & = & r & (e) \end{array} \right\} \right. \qquad (P)$$

$$\mathrm{Opt}(D) = \min_{[\lambda_\ell;\lambda_e]} \left\{ p^T\lambda_\ell + r^T\lambda_e : \left\{ \begin{array}{l} \lambda_\ell \geq 0 \\ P^T\lambda_\ell + R^T\lambda_e = c \end{array} \right\} \right. \qquad (D)$$

Bottom line: Problems (P), (D) are equivalent to problems

$$\max_\xi \left\{ \bar{\lambda}_\ell^T \xi : \xi \geq 0, \xi \in \mathcal{M}_P = \mathcal{L}_P + \bar{\xi} \right\} \qquad (\mathcal{P})$$

$$\min_{\lambda_\ell} \left\{ \bar{\xi}^T \lambda_\ell : \lambda_\ell \geq 0, \lambda_\ell \in \mathcal{M}_D = \mathcal{L}_D - \bar{\lambda}_\ell \right\} \qquad (\mathcal{D})$$

where

$$\mathcal{L}_P = \{\xi : \exists x : \xi = Px, Rx = 0\},$$
$$\mathcal{L}_D = \{\lambda_\ell : \exists \lambda_e : P^T\lambda_\ell + R^T\lambda_e = 0\}$$

Note: • Linear subspaces $\mathcal{L}_P$ and $\mathcal{L}_D$ are orthogonal complements of each other

• The minus primal objective $-\bar{\lambda}_\ell$ belongs to the dual feasible plane $\mathcal{M}_D$, and the dual objective $\bar{\xi}$ belongs to the primal feasible plane $\mathcal{M}_P$. Moreover, replacing $\bar{\lambda}_\ell$, $\bar{\xi}$ with any other pair of points from $-\mathcal{M}_D$ and $\mathcal{M}_P$, problems remain essentially intact – on the respective feasible sets, the objectives get constant shifts

Problems (P), (D) are equivalent to problems

$$\begin{array}{|rl|} \hline \max_{\xi} \left\{ \bar{\lambda}_\ell^T \xi : \xi \geq 0, \xi \in \mathcal{M}_P = \mathcal{L}_P + \bar{\xi} \right\} & (\mathcal{P}) \\ \hline \min_{\lambda_\ell} \left\{ \bar{\xi}^T \lambda_\ell : \lambda_\ell \geq 0, \lambda_\ell \in \mathcal{M}_D = \mathcal{L}_D - \bar{\lambda}_\ell \right\} & (\mathcal{D}) \\ \hline \end{array}$$

where

$$\mathcal{L}_P = \{\xi : \exists x : \xi = Px, Rx = 0\},$$
$$\mathcal{L}_D = \{\lambda_\ell : \exists \lambda_e : P^T \lambda_\ell + R^T \lambda_e = 0\}$$

• A primal-dual feasible pair $(x, [\lambda_\ell; \lambda_e])$ of solutions to (P), (D) induces a pair of feasible solutions $(\xi = p - Px, \lambda_\ell)$ to $(\mathcal{P}, \mathcal{D})$, and

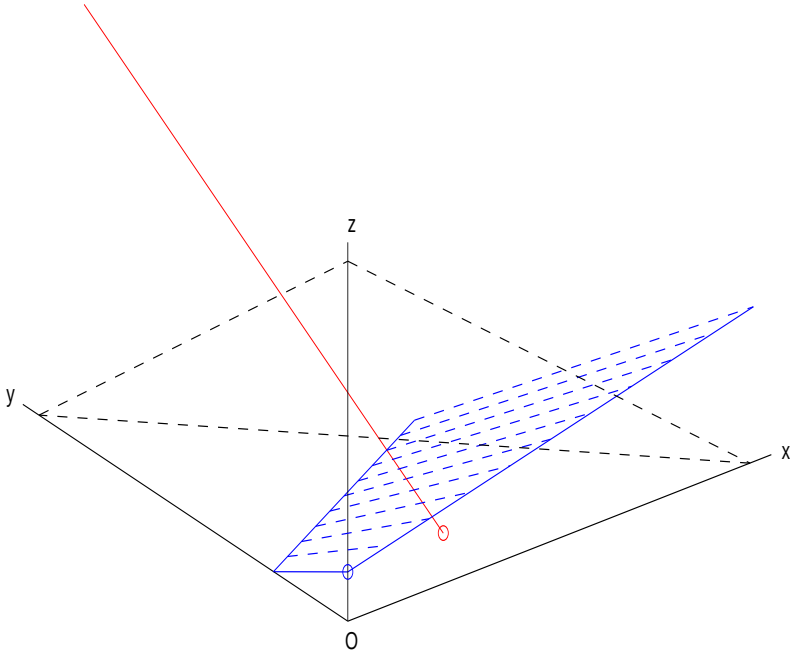
$$\mathsf{DualityGap}(x, [\lambda_\ell, \lambda_e]) = \lambda_\ell^T \xi.$$

Thus, *to solve $(P), (D)$ to optimality is the same as to pick a pair of orthogonal to each other feasible solutions to $(\mathcal{P})$, $(\mathcal{D})$.*

♣ We arrive at a wonderful perfectly symmetric and transparent geometric picture:

Geometrically, a primal-dual pair of LO programs is given by a pair of affine planes $\mathcal{M}_P$ and $\mathcal{M}_D$ in certain $\mathbb{R}^N$; these planes are shifts of linear subspaces $\mathcal{L}_P$ and $\mathcal{L}_D$ which are orthogonal complements of each other.
We intersect $\mathcal{M}_P$ and $\mathcal{M}_D$ with the nonnegative orthant $\mathbb{R}^N_+$, and our goal is to find in these intersections two orthogonal to each other vectors.

♠ Duality Theorem says that this task is feasible if and only if both $\mathcal{M}_P$ and $\mathcal{M}_D$ intersect the nonnegative orthant.



Geometry of primal-dual pair of LO programs:

Blue area: feasible set of $(\mathcal{P})$ — intersection of the 2D primal feasible plane $\mathcal{M}_P$ with the nonnegative orthant $\mathbb{R}^3_+$.

Red segment: feasible set of $(\mathcal{D})$ — intersection of the 1D dual feasible plane $\mathcal{M}_D$ with the nonnegative orthant $\mathbb{R}^3_+$.

Blue dot: primal optimal solution ξ^*.

Red dot: dual optimal solution λ^*_ℓ.

Pay attention to orthogonality of the primal solution (which is on the z-axis) and the dual solution (which is in the xy-plane).

The Cost Function of an LO program, I

♣ Consider an LO program

$$\mathrm{Opt}(b) = \max_x \left\{ c^T x : Ax \le b \right\}. \qquad (P[b])$$

Note: we treat the data A, c as fixed, and b as varying, and are interested in the properties of the optimal value $\mathrm{Opt}(b)$ as a function of b.

♠ **Fact:** *When b is such that $(P[b])$ is feasible, the property of problem to be/not to be bounded is independent of the value of b.*

Indeed, a feasible problem $(P[b])$ is unbounded iff there exists d: $Ad \le 0, c^T d > 0$, and this fact is independent of the particular value of b.

Standing Assumption: There exists b such that $P([b])$ is feasible and bounded

$\Rightarrow$ $P([b])$ is bounded whenever it is feasible.

Theorem *Under Assumption, $-\mathrm{Opt}(b)$ is a polyhedrally representable function with the polyhedral representation*

$$\begin{aligned} \{[b;\tau] : -\mathrm{Opt}(b) \le \tau\} &= \{[b;\tau] : \tau + \mathrm{Opt}(b) \ge 0\} \\ &= \{[b;\tau] : \exists x : Ax \le b, \tau + c^T x \ge 0\}. \end{aligned}$$

The function $\mathrm{Opt}(b)$ is monotone in b:

$$b' \le b'' \Rightarrow \mathrm{Opt}(b') \le \mathrm{Opt}(b'').$$

$$\mathrm{Opt}(b) = \max_x \left\{ c^T x : Ax \le b \right\}. \qquad (P[b])$$

♠ Additional information can be obtained from Duality. The problem dual to $(P[b])$ is

$$\min_\lambda \left\{ b^T \lambda : \lambda \ge 0, A^T \lambda = c \right\}. \qquad (D[b])$$

By LO Duality Theorem, under our Standing Assumption $(D[b])$ *is feasible for every* b *for which* $P([b])$ *is feasible, and*

$$\mathrm{Opt}(b) = \min_\lambda \left\{ b^T \lambda : \lambda \ge 0, A^T \lambda = c \right\}. \qquad (*)$$

Observation: *Let* $\bar{b}$ *be such that* $\mathrm{Opt}(\bar{b}) > -\infty$*, so that* $(D[\bar{b}])$ *is solvable, and let* $\bar{\lambda}$ *be an optimal solution to* $(D[\bar{b}])$*. Then* $\bar{\lambda}$ *is a supergradient of* $\mathrm{Opt}(b)$ *at* $b = \bar{b}$*, meaning that*

$$\forall b : \mathrm{Opt}(b) \le \mathrm{Opt}(\bar{b}) + \bar{\lambda}^T [b - \bar{b}]. \qquad (!)$$

Indeed, by $(*)$ we have $\mathrm{Opt}(\bar{b}) = \bar{\lambda}^T \bar{b}$ and $\mathrm{Opt}(b) \le \bar{\lambda}^T b$, that is, $\mathrm{Opt}(b) \le \bar{\lambda}^T \bar{b} + \bar{\lambda}^T [b - \bar{b}] = \mathrm{Opt}(\bar{b}) + \bar{\lambda}^T [b - \bar{b}]$. □

$$\begin{array}{rcll}
\mathrm{Opt}(b) & = & \max\limits_x \left\{c^T x : Ax \le b\right\} & (P[b]) \\
& = & \min\limits_\lambda \left\{b^T\lambda : \lambda \ge 0, A^T\lambda = c\right\} & (D[b])
\end{array}$$

$$\mathrm{Opt}(\bar{b}) > -\infty,\ \bar{\lambda} \in \mathrm{Argmin}_\lambda\{\bar{b}^T\lambda : \lambda \ge 0, A^T\lambda = c\}$$

$$\Rightarrow \mathrm{Opt}(b) \le \mathrm{Opt}(\bar{b}) + \bar{\lambda}[b - \bar{b}] \quad (!)$$

$\mathrm{Opt}(b)$ is a polyhedrally representable and thus piecewise-linear function with the *full-dimensional* domain:

$$\mathrm{Dom}\,\mathrm{Opt}(\cdot) = \{b : \exists x : Ax \le b\}.$$

Representing the feasible set $\Lambda = \{\lambda : \lambda \ge 0, A^T\lambda = c\}$ of $(D[b])$ as

$$\Lambda = \mathrm{Conv}(\{\lambda_1, ..., \lambda_N\}) + \mathrm{Cone}\,(\{r_1, ..., r_M\})$$

we get

$$\begin{array}{l}
\mathrm{Dom}\,\mathrm{Opt}(b) = \{b : b^T r_j \ge 0,\ 1 \ge j \le M\}, \\
b \in \mathrm{Dom}\,\mathrm{Opt}(b) \Rightarrow \mathrm{Opt}(b) = \min\limits_{1 \le i \le N} \lambda_i^T b
\end{array}$$

$$\begin{aligned}
\mathrm{Opt}(b) &= \max_x \left\{c^T x : Ax \le b\right\} && (P[b])\\
&= \min_\lambda \left\{b^T\lambda : \lambda \ge 0, A^T\lambda = c\right\} && (D[b])
\end{aligned}$$

$$\mathrm{Opt}(\bar{b}) > -\infty,\ \bar{\lambda} \in \mathrm{Argmin}_\lambda\{\bar{b}^T\lambda : \lambda \ge 0, A^T\lambda = c\}$$

$$\Rightarrow \mathrm{Opt}(b) \le \mathrm{Opt}(\bar{b}) + \bar{\lambda}[b - \bar{b}] \quad (!)$$

$$\mathrm{Dom}\,\mathrm{Opt}(b) = \{b : b^T r_j \ge 0,\ 1 \le j \le M\},$$
$$b \in \mathrm{Dom}\,\mathrm{Opt}(b) \Rightarrow \mathrm{Opt}(b) = \min_{1\le i\le N} \lambda_i^T b$$

$\Rightarrow$ *Under our Standing Assumption,*

- $\mathrm{Dom}\,\mathrm{Opt}(\cdot)$ *is a full-dimensional polyhedral cone,*
- *Assuming w.l.o.g. that* $\lambda_i \ne \lambda_j$ *when* $i \ne j$, *the finitely many hyperplanes* $\{b : \lambda_i^T b = \lambda_j^T b\}$, $1 \le i < j \le N$, *split this cone into finitely many cells, and in the interior of every cell* $\mathrm{Opt}(b)$ *is a linear function of* b.
- *By* (!), *when* b *is in the interior of a cell, the optimal solution* $\lambda(b)$ *to* $(D[b])$ *is unique, and* $\lambda(b) = \nabla\mathrm{Opt}(b)$.

Law of Diminishing Marginal Returns

♠ Consider a function of the form

$$\mathrm{Opt}(\beta) = \max_x \left\{ c^T x : Px \le p, q^T x \le \beta \right\} \qquad (P_\beta)$$

Interpretation: x is a production plan, $q^T x$ is the price of resources required by x, β is our investment in the recourses, $\mathrm{Opt}(\beta)$ is the maximal return for an investment β. Assume that *(P_β) is feasible for some β.*

♠ As above, for β such that (P_β) is feasible, the problem is either always bounded, or is always unbounded. Assume that the first is the case. Then

- The domain $\mathrm{Dom}\,\mathrm{Opt}(\cdot)$ of $\mathrm{Opt}(\cdot)$ is a nonempty ray $\underline{\beta} \le \beta < \infty$ with $\underline{\beta} \ge -\infty$, and
- $\mathrm{Opt}(\beta)$ is nondecreasing and *concave*.

Monotonicity and concavity imply that if

$$\underline{\beta} \le \beta_1 < \beta_2 < \beta_3,$$

then

$$\frac{\mathrm{Opt}(\beta_2) - \mathrm{Opt}(\beta_1)}{\beta_2 - \beta_1} \ge \frac{\mathrm{Opt}(\beta_3) - \mathrm{Opt}(\beta_2)}{\beta_3 - \beta_2},$$

that is, *the reward for an extra \$1 in the investment can only decrease* (or remain the same) *as the investment grows.*

In Economics, this is called *the law of diminishing marginal returns.*

The Cost Function of an LO program, II

♣ Consider an LO program

$$\mathrm{Opt}(c) = \max_x \left\{ c^T x : Ax \leq b \right\}. \qquad (P[c])$$

Note: we treat the data A, b as fixed, and c as varying, and are interested in the properties of $\mathrm{Opt}(c)$ as a function of c.

Standing Assumption: $(P[\cdot])$ is feasible (this fact is independent of the value of c).

Theorem *Under Assumption,* $\mathrm{Opt}(c)$ *is a polyhedrally representable function with the polyhedral representation*

$$\{[c;\tau] : \mathrm{Opt}(c) \leq \tau\} = \{[c;\tau] : \exists \lambda : \lambda \geq 0, A^T\lambda = c, b^T\lambda \leq \tau\}.$$

Proof. Since $(P[c])$ is feasible, by LO Duality Theorem the program is solvable if and only if the dual program

$$\min_\lambda \{b^T\lambda : \lambda \geq 0, A^T\lambda = c\} \qquad (D[c])$$

is feasible, and in this case the optimal values of the problems are equal
$\Rightarrow \tau \geq \mathrm{Opt}(c)$ iff $(D[c])$ has a feasible solution with the value of the objective $\leq \tau$. □

$$\mathrm{Opt}(c) = \max_x \left\{ c^T x : Ax \le b \right\}. \qquad (P[c])$$

Theorem *Let $\bar{c}$ be such that $\mathrm{Opt}(\bar{c}) < \infty$, and $\bar{x}$ be an optimal solution to $(P[\bar{c}])$. Then $\bar{x}$ is a subgradient of $\mathrm{Opt}(\cdot)$ at the point $\bar{c}$:*

$$\forall c : \mathrm{Opt}(c) \ge \mathrm{Opt}(\bar{c}) + \bar{x}^T [c - \bar{c}]. \qquad (!)$$

Proof: We have $\mathrm{Opt}(c) \ge c^T \bar{x} = \bar{c}^T \bar{x} + [c - \bar{c}]^T \bar{x} = \mathrm{Opt}(\bar{c}) + \bar{x}^T [c - \bar{c}]$. □

♠ Representing

$$\{x : Ax \le b\} = \mathrm{Conv}(\{v_1, ..., v_N\}) + \mathrm{Cone}\,(\{r_1, ..., r_M\}),$$

we see that

- $\mathrm{Dom}\,\mathrm{Opt}(\cdot) = \{c : r_j^T c \le 0,\ 1 \le j \le M\}$ is a polyhedral cone, and
- $c \in \mathrm{Dom}\,\mathrm{Opt}(\cdot) \Rightarrow \mathrm{Opt}(c) = \max\limits_{1 \le i \le N} v_i^T c$.

In particular, *if $\mathrm{Dom}\,\mathrm{Opt}(\cdot)$ is full-dimensional and v_i are distinct from each other, everywhere in $\mathrm{Dom}\,\mathrm{Opt}(\cdot)$ outside finitely many hyperplanes $\{c : v_i^T c = v_j^T c\}$, $1 \le i < j \le N$, the optimal solution $x = x(c)$ to $(P[c])$ is unique and $x(c) = \nabla \mathrm{Opt}(c)$.*

♣ Let $X = \{x \in \mathbb{R}^n : Ax \leq b\}$ be a *nonempty* polyhedral set. The function

$$\mathrm{Opt}(c) = \max_{x \in X} c^T x : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$$

has a name - it is called the *support function* of X. Along with already investigated properties of the support function, an important one is as follows:

♠ *The support function of a nonempty polyhedral set X "remembers" X: if*

$$\mathrm{Opt}(c) = \max_{x \in X} c^T x,$$

then

$$X = \{x \in \mathbb{R}^n : c^T x \leq \mathrm{Opt}(c)\ \forall c\}.$$

Proof. Let $X^+ = \{x \in \mathbb{R}^n : c^T x \leq \mathrm{Opt}(c)\, \forall c\}$. We clearly have $X \subset X^+$. To prove the inverse inclusion, let $\bar{x} \in X^+$; we want to prove that $x \in X$. To this end let us represent $X = \{x \in \mathbb{R}^n : a_i^T x \leq b_i,\ 1 \leq i \leq m\}$. For every i, we have

$$a_i^T \bar{x} \leq \mathrm{Opt}(a_i) \leq b_i,$$

and thus $\bar{x} \in X$. □

Applications of Duality in Robust LO

♣ Data uncertainty: Sources

Typically, the data of real world LOs

$$\max_x \left\{c^T x : Ax \leq b\right\} \quad [A = [a_{ij}] : m \times n] \qquad \text{(LO)}$$

is not known exactly when the problem is being solved. The most common reasons for data uncertainty are:

• Some of data entries (future demands, returns, etc.) do not exist when the problem is solved and hence are replaced with their forecasts. These data entries are subject to *prediction errors*

• Some of the data (parameters of technological devices/processes, contents associated with raw materials, etc.) cannot be measured exactly, and their true values drift around the measured "nominal" values. These data are subject to *measurement errors*

• Some of the decision variables (intensities with which we intend to use various technological processes, parameters of physical devices we are designing, etc.) cannot be implemented exactly as computed. The resulting *implementation errors* are equivalent to appropriate artificial data uncertainties.

A typical implementation error can be modeled as $x_j \mapsto (1+\xi_j)x_j+\eta_j$, and effect of these errors on a linear constraint

$$\sum_{j=1}^{n} a_{ij}x_j \le b_j$$

is *as if* there were no implementation errors, but the data a_{ij} got the multiplicative perturbations:

$$a_{ij} \mapsto a_{ij}(1+\xi_j)$$

,

and the data b_i got the perturbation

$$b_i \mapsto b_i - \textstyle\sum_j \eta_j a_{ij}.$$

Data uncertainty: Dangers.

In the traditional LO methodology, a small data uncertainty (say, 0.1% or less) is just ignored: the problem is solved *as if* the given ("nominal") data were exact, and the resulting *nominal* optimal solution is what is recommended for use.

Rationale: we hope that small data uncertainties will not affect too badly the feasibility/optimality properties of the nominal solution when plugged into the "true" problem.

Fact: *The above hope can be by far too optimistic, and the nominal solution can be practically meaningless.*

♣ **Example: Antenna Design**

♠ [Physics:] *Directional density of energy transmitted by a monochromatic antenna placed at the origin is proportional to* $|D(\delta)|^2$, *where the antenna's diagram* $D(\delta)$ *is a complex-valued function of 3-D direction (unit 3-D vector)* δ.

♠ [Physics:] For an *antenna array* — a complex antenna composed of a number of antenna elements, the diagram is

$$D(\delta) = \sum_j x_j D_j(\delta) \qquad (*)$$

- $D_j(\cdot)$: diagrams of elements
- x_j: complex *weights* – design parameters responsible for how the elements in the array are invoked.

♠ **Antenna Design problem:** *Given diagrams*

$$D_1(\cdot), ..., D_n(\cdot)$$

and a target diagram $D_*(\cdot)$, *find complex weights* x_i *which make the synthesized diagram* $(*)$ *as close as possible to the target diagram* $D_*(\cdot)$.

♡ When $D_j(\cdot)$, $D_*(\cdot)$ and the weights are real and the "closeness' is quantified by the maximal deviation along a finite grid Γ of directions, Antenna Design becomes the LO problem

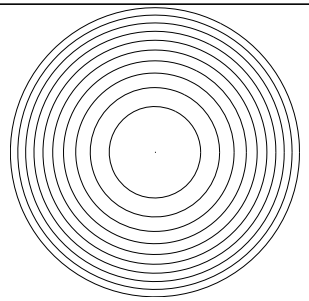
$$\min_{x\in\mathbb{R}^n,\tau} \left\{\tau : -\tau \le D_*(\delta) - \sum_j x_j D_j(\delta) \le \tau \ \forall \delta \in \Gamma\right\}.$$

♠ **Example:** Consider planar antenna array composed of 10 elements (circle surrounded by 9 rings of equal areas) in the plane XY (Earth's surface"), and our goal is to send most of the energy "up," along the 12^o cone around the Z-axis.

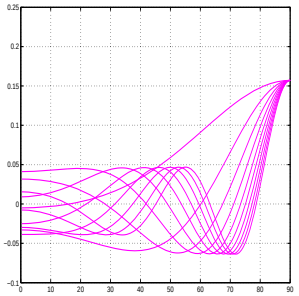
- Diagram of a ring $\{z = 0, a \le \sqrt{x^2 + y^2} \le b\}$:

$$D_{a,b}(\theta) = \frac{1}{2} \int_a^b \left[\int_0^{2\pi} r \cos\left(2\pi r \lambda^{-1} \cos(\theta) \cos(\phi)\right) d\phi \right] dr,$$

- θ: altitude angle
- λ: wavelength



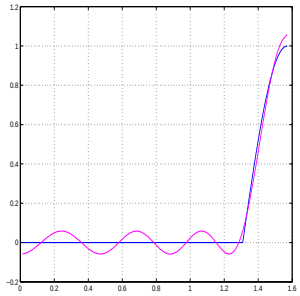
10 elements,
equal areas,
outer radius 1 m



Diagrams of the elements vs the altitude angle θ, λ =50 cm

- Nominal design problem:

$$\tau_* = \min_{x \in \mathbb{R}^{10}, \tau} \Big\{ \tau : -\tau \le D_*(\theta_i) - \sum_{j=1}^{10} x_j D_j(\theta_i) \le \tau,$$
$$1 \le i \le 240 \Big\}, \quad \theta_i = \frac{i\pi}{480}$$

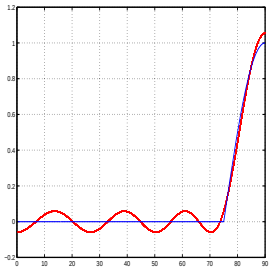


Target (blue) and nominal optimal (magenta) diagrams,
$\tau_* = 0.0589$

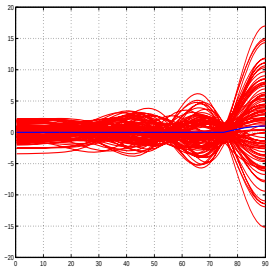
But: The design variables are characteristics of physical devices and as such they cannot be implemented exactly as computed. *What happens when there are implementation errors:*

$$x_j^{\text{fact}} = (1+\epsilon_j)x_j^{\text{comp}},\ \epsilon_j \sim \text{Uniform}[-\rho,\rho]$$

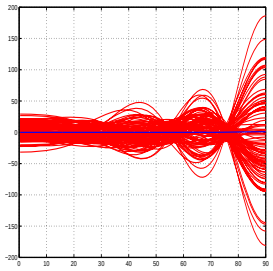
with small ρ?



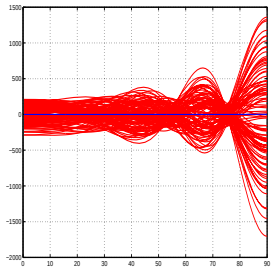
$\rho = 0$



$\rho = 0.0001$



$\rho = 0.001$



$\rho = 0.01$

"Dream and reality," nominal optimal design: samples of 100 actual diagrams (red) for different uncertainty levels. Blue: the target diagram

	Dream	Reality		
	$\rho = 0$	$\rho = 0.0001$	$\rho = 0.001$	$\rho = 0.01$
	value	mean	mean	mean
$\|\cdot\|_\infty$-distance to target	0.059	5.671	56.84	506.5
energy concentration	85.1%	16.4%	16.5%	14.9%

Quality of nominal antenna design: dream and reality. Data over 100 samples of actuation errors per each uncertainty level.

♠ Conclusion: *Nominal optimal design is completely meaningless...*

NETLIB Case Study: Diagnosis

♣ NETLIB is a collection of about 100 not very large LPs, mostly of real-world origin. To motivate the methodology of our "case study", here is constraint # 372 of the NETLIB problem PILOT4:

$$\begin{array}{rcl} a^T x & \equiv & -15.79081x_{826} - 8.598819x_{827} - 1.88789x_{828} - 1.362417x_{829} \\ & & -1.526049x_{830} - 0.031883x_{849} - 28.725555x_{850} - 10.792065x_{851} \\ & & -0.19004x_{852} - 2.757176x_{853} - 12.290832x_{854} + 717.562256x_{855} \\ & & -0.057865x_{856} - 3.785417x_{857} - 78.30661x_{858} - 122.163055x_{859} \\ & & -6.46609x_{860} - 0.48371x_{861} - 0.615264x_{862} - 1.353783x_{863} \\ & & -84.644257x_{864} - 122.459045x_{865} - 43.15593x_{866} - 1.712592x_{870} \\ & & -0.401597x_{871} + x_{880} - 0.946049x_{898} - 0.946049x_{916} \\ \geq & b \equiv 23.387405 & \end{array}$$

The related *nonzero* coordinates in the optimal solution x^* of the problem, as reported by CPLEX, are:

$$\begin{array}{ll} x^*_{826} = 255.6112787181108 & x^*_{827} = 6240.488912232100 \\ x^*_{828} = 3624.613324098961 & x^*_{829} = 18.20205065283259 \\ x^*_{849} = 174397.0389573037 & x^*_{870} = 14250.00176680900 \\ x^*_{871} = 25910.00731692178 & x^*_{880} = 104958.3199274139 \end{array}$$

This solution makes the constraint an equality within machine precision.

♣ Most of the coefficients in the constraint are "ugly reals" like -15.79081 or -84.644257. We can be sure that these coefficients characterize technological devices/processes, and as such *hardly are known to high accuracy*.

⇒ "ugly coefficients" can be assumed uncertain and coinciding with the "true" data within accuracy of 3-4 digits.

The only exception is the coefficient 1 of x_{880}, which perhaps reflects the structure of the problem and is exact.

$$
\begin{array}{rcl}
a^T x & \equiv & -15.79081x_{826} - 8.598819x_{827} - 1.88789x_{828} - 1.362417x_{829} \\
& & -1.526049x_{830} - 0.031883x_{849} - 28.725555x_{850} - 10.792065x_{851} \\
& & -0.19004x_{852} - 2.757176x_{853} - 12.290832x_{854} + 717.562256x_{855} \\
& & -0.057865x_{856} - 3.785417x_{857} - 78.30661x_{858} - 122.163055x_{859} \\
& & -6.46609x_{860} - 0.48371x_{861} - 0.615264x_{862} - 1.353783x_{863} \\
& & -84.644257x_{864} - 122.459045x_{865} - 43.15593x_{866} - 1.712592x_{870} \\
& & -0.401597x_{871} + x_{880} - 0.946049x_{898} - 0.946049x_{916} \\
& \geq & b \equiv 23.387405
\end{array}
$$

♣ Assume that the uncertain entries of a are 0.1%-accurate approximations of unknown entries in the "true" data $\tilde{a}$. How does data uncertainty affect the validity of the constraint *as evaluated at the nominal solution x^*?*

• *The worst case*, over all 0.1%-perturbations of uncertain data, violation of the constraint is as large as 450% of the right hand side!

• With *random* and *independent* of each other 0.1% perturbations of the uncertain coefficients, the statistics of the "relative constraint violation"

$$V = \frac{\max[b - \tilde{a}^T x^*, 0]}{b} \times 100\%$$

also is disastrous:

Prob$\{V > 0\}$	Prob$\{V > 150\%\}$	Mean(V)
0.50	0.18	125%

Relative violation of constraint # 372 in PILOT4
(1,000-element sample of 0.1% perturbations)

♣ We see that *quite small (just 0.1%) perturbations of "obviously uncertain" data coefficients can make the "nominal" optimal solution x^* heavily infeasible and thus – practically meaningless.*

♣ In Case Study, we choose a "perturbation level" $\rho \in \{1\%, 0.1\%, 0.01\%\}$, and, for every one of the NETLIB problems, measure the "reliability index" of the nominal solution at this perturbation level:

- We compute the optimal solution x^* of the program
- For every one of the *in*equality constraints

$$a^T x \leq b$$

— we split the left hand side coefficients a_j into "certain" (rational fractions p/q with $|q| \leq 100$) and "uncertain" (all the rest). Let J be the set of all uncertain coefficients of the constraint.

— we compute the *reliability index* of the constraint

$$\frac{\max[a^T x^* + \rho\sqrt{\sum_{j\in J} a_j^2 (x_j^*)^2} - b, 0]}{\max[1, |b|]} \times 100\%$$

Note: *the reliability index is of order of typical violation* (measured in percents of the right hand side) *of the constraint, as evaluated at x^*, under independent random perturbations, of relative magnitude ρ, of the uncertain coefficients*.

- We treat the nominal solution as *unreliable*, and the problem - as *bad*, the level of perturbations being ρ, if the worst, over the inequality constraints, reliability index is worse than 5%.

♣ The results of the Diagnosis phase of Case Study are as follows. From the 90 NETLIB problems,
— in 27 problems the nominal solution turned out to be unreliable when $\rho = 1\%$;
— 19 of these 27 problems were already bad at the $\rho = 0.01\%$-level of uncertainty
— in 13 problems, 0.01% perturbations of the uncertain data can make the nominal solution more than 50%-infeasible for some of the constraints.

Problem	Size[a]	$\rho = 0.01\%$		$\rho = 0.1\%$	
		#bad[b]	Index[c]	#bad	Index
80BAU3B	2263×9799	37	84	177	842
25FV47	822×1571	14	16	28	162
ADLITTLE	57×97			2	6
AFIRO	28×32			1	5
CAPRI	272×353			10	39
CYCLE	1904×2857	2	110	5	1,100
D2Q06C	2172×5167	107	1,150	134	11,500
FINNIS	498×614	12	10	63	104
GREENBEA	2393×5405	13	116	30	1,160
KB2	44×41	5	27	6	268
MAROS	847×1443	3	6	38	57
PEROLD	626×1376	6	34	26	339
PILOT	1442×3652	16	50	185	498
PILOT4	411×1000	42	210,000	63	2,100,000
PILOT87	2031×4883	86	130	433	1,300
PILOTJA	941×1988	4	46	20	463
PILOTNOV	976×2172	4	69	13	694
PILOTWE	723×2789	61	12,200	69	122,000
SCFXM1	331×457	1	95	3	946
SCFXM2	661×914	2	95	6	946
SCFXM3	991×1371	3	95	9	946
SHARE1B	118×225	1	257	1	2,570

[a] # of linear constraints (excluding the box ones) plus 1 and # of variables
[b] # of constraints with index $> 5\%$
[c] The worst, over the constraints, reliability index, in %

♣ **Conclusions:**

♢ *In real-world applications of Linear Programming one cannot ignore the possibility that a small uncertainty in the data* (intrinsic for the majority of real-world LP programs) *can make the usual optimal solution of the problem completely meaningless from practical viewpoint.*

Consequently,

♢ *In applications of LP, there exists a real need of a technique capable of detecting cases when data uncertainty can heavily affect the quality of the nominal solution, and in these cases to generate a "reliable" solution, one which is immune against uncertainty.*

Robust LO is aimed at meeting this need.

Robust LO: Paradigm

♣ In Robust LO, one considers an *uncertain LO problem*

$$\mathcal{P} = \left\{ \max_x \left\{ c^T x : Ax \le b \right\} : (c, A, b) \in \mathcal{U} \right\},$$

— a *family* of *all* usual LO instances of common sizes m (number of constraints) and n (number of variables) with the *data* (c, A, b) running through a given *uncertainty set* $\mathcal{U} \subset \mathbb{R}^n_c \times \mathbb{R}^{m\times n}_A \times \mathbb{R}^m_b$.

♠ We consider the situation where

• *The solution should be built before the "true" data reveals itself and thus cannot depend on the true data.* All we know when building the solution is the uncertainty set $\mathcal{U}$ to which the true data belongs.

• *The constraints are hard: we cannot tolerate their violation.*

♠ In the outlined "decision environment," the only meaningful candidate solutions x are the *robust feasible ones* – those *which remain feasible whatever be a realization of the data from the uncertainty set:*

$$x \in \mathbb{R}^n \text{ is robust feasible for } \mathcal{P} \Leftrightarrow Ax \leq b \, \forall (c, A, b) \in \mathcal{U}$$

♡ We characterize the objective at a candidate solution x by the *guaranteed value*

$$t(x) = \min\{c^T x : (c, A, b) \in \mathcal{U}\}$$

of the objective.

$$\mathcal{P} = \left\{\max_x \left\{c^T x : Ax \le b\right\} : (c, A, b) \in \mathcal{U}\right\},$$

♡ Finally, we associate with the uncertain problem $\mathcal{P}$ its *Robust Counterpart*

$$\mathrm{ROpt}(\mathcal{P}) = \max_{t,x} \left\{t : t \le c^T x,\ Ax \le b \,\forall (c, A, b) \in \mathcal{U}\right\} \qquad \text{(RC)}$$

where one seeks for the best (with the largest *guaranteed* value of the objective) *robust feasible* solution to $\mathcal{P}$.

The optimal solution to the RC is treated as the best among "immunized against uncertainty" solutions and is recommended for actual use.

Basic question: *Unless the uncertainty set $\mathcal{U}$ is finite, the RC is not an LO program, since it has infinitely many linear constraints. Can we convert* (RC) *into an explicit LO program?*

$$\mathcal{P} = \left\{\max_x \left\{c^T x : Ax \le b\right\} : (c, A, b) \in \mathcal{U}\right\}$$
$$\Rightarrow \max_{t,x} \left\{t : t \le c^T x,\ Ax \le b \,\forall (c, A, b) \in \mathcal{U}\right\} \quad \text{(RC)}$$

Observation: *The RC remains intact when the uncertainty set $\mathcal{U}$ is replaced with its convex hull.*

Theorem: *The RC of an uncertain LO program with nonempty polyhedrally representable uncertainty set is equivalent to an LO program. Given a polyhedral representation of $\mathcal{U}$, the LO reformulation of the RC is easy to get.*

$$\mathcal{P} = \left\{\max_x \left\{c^T x : Ax \le b\right\} : (c, A, b) \in \mathcal{U}\right\}$$
$$\Rightarrow \max_{t,x} \left\{t : t \le c^T x,\ Ax \le b \,\forall (c, A, b) \in \mathcal{U}\right\} \quad \text{(RC)}$$

Proof of Theorem. Let

$$\mathcal{U} = \{\zeta = (c, A, b) \in \mathbb{R}^N : \exists w : P\zeta + Qw \le r\}$$

be a polyhedral representation of the uncertainty set. Setting $y = [x; t]$, the constraints of (RC) become

$$q_i(\zeta) - p_i^T(\zeta) y \le 0 \,\forall \zeta \in \mathcal{U},\ 0 \le i \le m \quad (C_i)$$

with $p_i(\cdot)$, $q_i(\cdot)$ affine in ζ. We have

$$q_i(\zeta) - p_i^T(\zeta) y \equiv \pi_i^T(y)\zeta - \theta_i(y),$$

with $\theta_i(y)$, $\pi_i(y)$ affine in y. Thus, i-th constraint in (RC) reads

$$\max_{\zeta, w_i} \left\{\pi_i^T(y)\zeta : P\zeta + Qw_i \le r\right\} = \max_{\zeta \in \mathcal{U}} \pi_i^T(y)\zeta \le \theta_i(y).$$

Since $\mathcal{U} \neq \emptyset$, by the LO Duality we have

$$\max_{\zeta, w_i} \left\{\pi_i^T(y)\zeta : P\zeta + Qw_i \le r\right\}$$
$$= \min_{\eta_i} \left\{r^T \eta_i : \eta_i \ge 0, P^T \eta_i = \pi_i(y), Q^T \eta_i = 0\right\}$$

$\Rightarrow$ *y satisfies (C_i) if and only if there exists η_i such that*

$$\eta_i \ge 0, P^T \eta_i = \pi_i(y), Q^T \eta_i = 0, r^T \eta_i \le \theta_i(y) \qquad (R_i)$$

$\Rightarrow$ (RC) *is equivalent to the LO program of maximizing $e^T y \equiv t$ in variables $y, \eta_0, \eta_1, ..., \eta_m$ under the linear constraints (R_i), $0 \le i \le m$.*

♠ **Example:** The Robust Counterpart of uncertain LO with *interval uncertainty*:

$$\begin{array}{rcl} \mathcal{U}_{\mathsf{obj}} & = & \{c : |c_j - c_j^0| \leq \delta c_j,\, j = 1, ..., n\} \\ \mathcal{U}_i & = & \{(a_{i1}, , , .a_{im}, b_i) : |a_{ij} - a_{ij}^0| \leq \delta a_{ij}, |b_i - b_i^0| \leq \delta b_i\} \end{array}$$

is the program

$$\max_{x,t} \left\{ t : \begin{array}{c} t \leq \sum_j c_j^0 x_j - \sum_j \delta c_j |x_j| \\ \sum_j a_{ij}^0 x_j + \sum_j \delta a_{ij} |x_j| \leq b_i - \delta b_i^0 \end{array} \right\}$$

which is equivalent to the LO program

$$\max_{x,y,t} \left\{ t : \begin{array}{c} t \leq \sum_j c_j^0 x_j - \sum_j \delta c_j y_j \\ \sum_j a_{ij}^0 x_j + \sum_j \delta a_{ij} y_j \leq b_i - \delta b_i^0 \\ -y_j \leq x_j \leq y_j \end{array} \right\}$$

How it works? – Antenna Example

$$\min_{x,\tau}\left\{\tau : -\tau \le D_*(\theta_\ell) - \textstyle\sum_{j=1}^{10} x_j D_j(\theta_\ell) \le \tau,\ \ell = 1,...,L\right\}$$

$$\Updownarrow$$

$$\boxed{\min_{x,\tau}\{\tau : Ax + \tau a + b \ge 0\} \qquad \text{(LO)}}$$

- The influence of "implementation errors"

$$x_j \mapsto (1+\epsilon_j)x_j$$

with $|\epsilon_j| \le \rho \in [0,1]$ is *as if* there were no implementation errors, but the part A of the constraint matrix was uncertain and known "up to multiplication by a diagonal matrix with diagonal entries from $[1-\rho, 1+\rho]$":

$$\mathcal{U} = \left\{A = A^{\text{nom}}\text{Diag}\{1+\epsilon_1,...,1+\epsilon_{10}\} : |\epsilon_j| \le \rho\right\} \qquad \text{(U)}$$

Note that *as far as a particular constraint is concerned, the uncertainty is an interval one with $\delta A_{ij} = \rho|A_{ij}|$. The remaining coefficients (and the objective) are certain.*

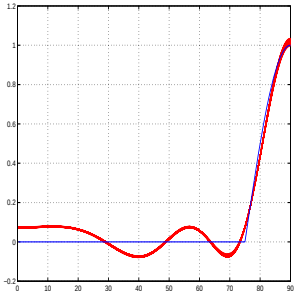
♣ To improve reliability of our design, we replace the uncertain LO program (LO), (U) with its robust counterpart, which is nothing but an explicit LO program.

How it Works: Antenna Design (continued)

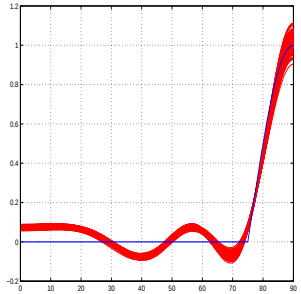
$$\min_{\tau,x}\left\{\tau : -\tau \le D_*(\theta_i) - \textstyle\sum_{j=1}^{10} x_j D_j(\theta_i) \le \tau,\ 1 \le i \le I\right\}, \quad x_j \mapsto (1+\epsilon_j)x_j,\ -\rho \le \epsilon_j \le \rho$$

$$\Rightarrow \min_{\tau,x}\left\{\tau : \begin{array}{l} D_*(\theta_i) - \sum_j x_j D_j(\theta_i) - \rho \sum_j |x_j||D_j(\theta_i)| \ge -\tau \\ D_*(\theta_i) - \sum_j x_j D_j(\theta_i) + \rho \sum_j |x_j||D_j(\theta_i)| \le \tau \end{array}, 1 \le i \le I\right\}$$

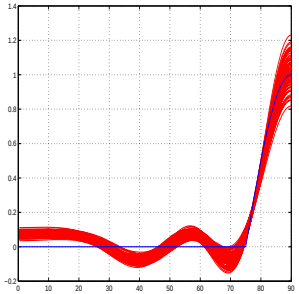
♠ Solving the Robust Counterpart at uncertainty level $\rho = 0.01$, we arrive at *robust design*. The robust optimal value is 0.0815 (39% more than the nominal optimal value 0.0589).



$\rho = 0.01$



$\rho = 0.05$



$\rho = 0.1$

Robust optimal design: samples of 100 actual diagrams (red).

	Reality	
	$\rho = 0.01$	$\rho = 0.1$
$\|\cdot\|_\infty$ distance to target	max = 0.081 mean = 0.077	max = 0.216 mean = 0.113
energy concentration	min = 70.3% mean = 72.3%	min = 52.2% mean = 70.8%

Robust optimal design, data over 100 samples of actuation errors.

- For *nominal* design with $\rho = 0.001$, the average $\|\cdot\|_\infty$-distance to target is 56.8, and average energy concentration is 16.5%.

♣ *Why the "nominal design" is that unreliable?*

• The basic diagrams $D_j(\cdot)$ are "nearly linearly dependent". As a result, the nominal problem is "ill-posed" – it possesses a huge domain comprised of "nearly optimal" solutions. Indeed, look what are the optimal values in the nominal Antenna Design LO with added box constraints $|x_j| \leq L$ on the variables:

L	1	10	10^2	10^3	10^4	10^5	10^6
Opt_Val	0.0945	0.0800	0.0736	0.0696	0.0649	0.0607	0.0589

The "exactly optimal" solution to the nominal problem is very large, and therefore even small *relative* implementation errors may completely destroy the design.

• In the robust counterpart, magnitudes of candidate solutions are penalized, and RC implements a smart trade-off between the optimality and the magnitude (i.e., the stability) of the solution.

j	1	2	3	4	5	6	7	8	9	10
x_j^{nom}	2e3	-1e4	6e4	-1e5	1e5	2e4	-1e5	1e6	-7e4	1e4
x_j^{rob}	-0.3	5.0	-3.4	-5.1	6.9	5.5	5.3	-7.5	-8.9	13

How it works? NETLIB Case Study

♣ When applying the RO methodology to the bad NETLIB problems, assuming interval uncertainty of (relative) magnitude $\rho \in \{1\%, 0.1\%, 0.01\%\}$ in "ugly coefficients" of *in*equality constraints (*no uncertainty in equations*!), it turns out that

- Reliable solutions do exist, except for 4 cases corresponding to the highest ($\rho = 1\%$) perturbation level.
- The "price of immunization" in terms of the objective value is surprisingly low: when $\rho \leq 0.1\%$, it never exceeds 1% and it is less than 0.1% in 13 of 23 cases. Thus, *passing to the robust solutions, we gain a lot in the ability of the solution to withstand data uncertainty, while losing nearly nothing in optimality.*

Problem	Nominal optimal value	Objective at robust solution	
		$\rho = 0.1\%$	$\rho = 1\%$
80BAU3B	987224.2		1009229 (2.2%)
25FV47	5501.846	5502.191 (0.0%)	5505.653 (0.1%)
ADLITTLE	225495.0		228061.3 (1.1%)
AFIRO	-464.7531	-464.7500 (0.0%)	-464.2613 (0.1%)
BNL2	1811.237	1811.237 (0.0%)	1811.338 (0.0%)
BRANDY	1518.511		1518.581 (0.0%)
CAPRI	1912.621	1912.738 (0.0%)	1913.958 (0.1%)
CYCLE	1913.958	1913.958 (0.0%)	1913.958 (0.0%)
D2Q06C	122784.2	122893.8 (0.1%)	Infeasible
E226	-18.75193		-18.75173 (0.0%)
FFFFF800	555679.6		555715.2 (0.0%)
FINNIS	172791.1	173269.4 (0.3%)	178448.7 (3.3%)
GREENBEA	-72555250	-72192920 (0.5%)	-68869430 (5.1%)
KB2	-1749.900	-1749.638 (0.0%)	-1746.613 (0.2%)
MAROS	-58063.74	-58011.14 (0.1%)	-57312.23 (1.3%)
NESM	14076040		14172030 (0.7%)
PEROLD	-9380.755	-9362.653 (0.2%)	Infeasible
PILOT	-557.4875	-555.3021 (0.4%)	Infeasible
PILOT4	-64195.51	-63584.16 (1.0%)	-58113.67 (9.5%)
PILOT87	301.7109	302.2191 (0.2%)	Infeasible
PILOTJA	-6113.136	-6104.153 (0.2%)	-5943.937 (2.8%)
PILOTNOV	-4497.276	-4488.072 (0.2%)	-4405.665 (2.0%)
PILOTWE	-2720108	-2713356 (0.3%)	-2651786 (2.5%)
SCFXM1	18416.76	18420.66 (0.0%)	18470.51 (0.3%)
SCFXM2	36660.26	36666.86 (0.0%)	36764.43 (0.3%)
SCFXM3	54901.25	54910.49 (0.0%)	55055.51 (0.3%)
SHARE1B	-76589.32	-76589.32 (0.0%)	-76589.29 (0.0%)

Objective values at nominal and robust solutions to bad NETLIB problems.
Percent in $(\cdot)$: Excess of robust optimal value over the nominal optimal value

Affinely Adjustable Robust Counterpart

♣ The rationale behind the Robust Optimization paradigm as applied to LO is based on two assumptions:

A. *Constraints of an uncertain LO program are a "must": a meaningful solution should satisfy all realizations of the constraints allowed by the uncertainty set.*

B. *All decision variables should be defined before the true data become known and thus should be independent of the true data.*

♣ In many cases, Assumption **B** is too conservative:

• In dynamical decision-making, only part of decision variables correspond to "here and now" decisions, while the remaining variables represent "wait and see" decisions to be made when part of the true data will be already revealed.

(!) *"Wait and see" decision variables may – and should! – depend on the corresponding part of the true data.*

• Some of decision variables do not represent actual decisions at all; they are artificial "analysis variables" introduced to convert the problem into the LO form.

(!) *Analysis variables may – and should! – depend on the entire true data.*

Example: Consider the problem of the best $\|\cdot\|_1$-approximation

$$\min_{x,t}\left\{t : \sum_i |b_i - \sum_j a_{ij}x_j| \le t\right\}. \qquad \text{(P)}$$

When the data are certain, this problem is equivalent to the LP program

$$\min_{x,y,t}\left\{t : \sum_i y_i \le t,\ -y_i \le b_i - \sum_j a_{ij}x_j \le y_i\ \forall i\right\}. \qquad \text{(LP)}$$

With uncertain data, the Robust Counterpart of (P) becomes the semi-infinite problem

$$\min_{x,t}\left\{t : \sum_i |b_i - \sum_j a_{ij}x_j| \le t\ \forall (b_i, a_{ij}) \in \mathcal{U}\right\},$$

or, which is the same, the problem

$$\min_{x,t}\left\{t : \forall (b_i, a_{ij}) \in \mathcal{U} :\ \exists y : \sum_i y_i \le t,\ -y_i \le b_i - \sum_j a_{ij}x_j \le y_i\right\},$$

while the RC of (LP) is the much more conservative problem

$$\min_{x,t}\left\{t : \exists y :\ \forall (b_i, a_{ij}) \in \mathcal{U} : \sum_i y_i \le t,\ -y_i \le b_i - \sum_j a_{ij}x_j \le y_i\right\}.$$

Adjustable Robust Counterpart of an Uncertain LO

♣ Consider an uncertain LO. Assume w.l.o.g. that the data of LO are affinely parameterized by a "perturbation vector" ζ running through a given *perturbation set* $\mathcal{Z}$:

$$\mathcal{LP} = \left\{\max_x \left\{c^T[\zeta]x : A[\zeta]x - b[\zeta] \leq 0\right\} : \zeta \in \mathcal{Z}\right\}$$
$$\left[c_j[\zeta], A_{ij}[\zeta], b_i[\zeta] : \text{ affine in } \zeta\right]$$

♠ Assume that every decision variable may depend on a given "portion" of the true data. Since the latter is affine in ζ, this assumption says that x_j *may depend on* $P_j\zeta$, where P_j are given matrices.

- $P_j = 0 \Rightarrow x_j$ *is non-adjustable*: x_j represents an independent of the true data "here and now" decision;
- $P_j \neq 0 \Rightarrow x_j$ *is adjustable*: x_j represents a "wait and see' decision or an analysis variable which may adjust itself – fully or partially, depending on P_j – to the true data.

$$\mathcal{LP} = \left\{\max_x \left\{c^T[\zeta]x : A[\zeta]x - b[\zeta] \le 0\right\} : \zeta \in \mathcal{Z}\right\}$$
$$\left[c_j[\zeta], A_{ij}[\zeta], b_i[\zeta] : \text{ affine in } \zeta\right]$$

♣ Under the circumstances, a natural Robust Counterpart of $\mathcal{LP}$ is the problem

Find t and functions $\phi_j(\cdot)$ such that the decision rules $x_j = \phi_j(P_j\zeta)$ make all the constraints feasible for all perturbations $\zeta \in \mathcal{Z}$, while minimizing the guaranteed value t of the objective:

$$\max_{t,\{\phi_j(\cdot)\}} \left\{ t : \begin{array}{l} \sum_j c_j[\zeta]\phi_j(P_j\zeta) \ge t \,\forall \zeta \in \mathcal{Z} \\ \sum_j \phi_j(P_j\zeta)A_j[\zeta] - b[\zeta] \le 0 \,\forall \zeta \in \mathcal{Z} \end{array} \right\} \qquad \text{(ARC)}$$

♣ **Bad news:** The *Adjustable Robust Counterpart*

$$\max_{t,\{\phi_j(\cdot)\}} \left\{ t : \begin{array}{l} \sum_j c_j[\zeta]\phi_j(P_j\zeta) \geq t \, \forall \zeta \in \mathcal{Z} \\ \sum_j \phi_j(P_j\zeta)A_j[\zeta] - b[\zeta] \leq 0 \, \forall \zeta \in \mathcal{Z} \end{array} \right\} \qquad \text{(ARC)}$$

of uncertain LP is an *infinite-dimensional* optimization program and as such typically is absolutely intractable: How could we represent efficiently general-type functions of many variables, not speaking about how to optimize with respect to these functions?

♠ **Partial Remedy (???):** Let us restrict the decision rules $x_j = \phi_j(P_j\zeta)$ to be easily representable – specifically, *affine* – functions:

$$\phi_j(P_j\zeta) \equiv \mu_j + \nu_j^T P_j \zeta.$$

With this dramatic simplification, (ARC) becomes a *finite-dimensional* (still semi-infinite) *optimization problem in new non-adjustable variables* μ_j, ν_j

$$\max_{t,\{\mu_j,\nu_j\}} \left\{ t : \begin{array}{l} \sum_j c_j[\zeta](\mu_j + \nu_j^T P_j\zeta) \geq t \, \forall \zeta \in \mathcal{Z} \\ \sum_j (\mu_j + \nu_j^T P_j\zeta)A_j[\zeta] - b[\zeta] \leq 0 \, \forall \zeta \in \mathcal{Z} \end{array} \right\} \qquad \text{(AARC)}$$

♣ We have associated with uncertain LO

$$\mathcal{LP} = \left\{\max_x \left\{c^T[\zeta]x : A[\zeta]x - b[\zeta] \le 0\right\} : \zeta \in \mathcal{Z}\right\}$$
$$\left[c_j[\zeta], A_{ij}[\zeta], b_i[\zeta] : \text{ affine in } \zeta\right]$$

and the "information matrices" $P_1, ..., P_n$ the *Affinely Adjustable Robust Counterpart*

$$\max_{t,\{\mu_j,\nu_j\}} \left\{ t : \begin{array}{l} \sum_j c_j[\zeta](\mu_j + \nu_j^T P_j \zeta) \ge t \, \forall \zeta \in \mathcal{Z} \\ \sum_j (\mu_j + \nu_j^T P_j \zeta) A_j[\zeta] - b[\zeta] \le 0 \, \forall \zeta \in \mathcal{Z} \end{array} \right\} \qquad \text{(AARC)}$$

♠ **Relatively good news:**

- AARC is by far more flexible than the usual (non-adjustable) RC of $\mathcal{LP}$.
- As compared to ARC, AARC has much more chances to be computationally tractable:

— *In the case of simple recourse*, where the coefficients of adjustable variables are certain, AARC has the same tractability properties as RC:
If the perturbation set $\mathcal{Z}$ is given by polyhedral representation, (AARC) can be straightforwardly converted into an explicit LO program.

— *In the general case*, (AARC) may be computationally intractable; however, under mild assumptions on the perturbation set, (AARC) admits "tight" computationally tractable approximation.

♣ **Example: simple Inventory model.** There is a single-product inventory system with

• a single warehouse which should at any time store at least $V_{\min}$ and at most $V_{\max}$ units of the product;

• *uncertain* demands d_t of periods $t = 1, ..., T$ known to vary within given bounds:

$$d_t \in [d_t^*(1-\theta), d_t^*(1+\theta)],\ t = 1, ..., T$$

• $\theta \in [0, 1]$: uncertainty level

No backlogged demand is allowed!

• I factories from which the warehouse can be replenished:

— at the beginning of period t, you may order $p_{t,i}$ units of product from factory i. Your orders should satisfy the constraints

$$\begin{array}{ll} 0 \le p_{t,i} \le P_i(t) & \text{[bounds on capacities per period]} \\ \sum_t p_{t,i} \le Q_i & \text{[bounds on cumulative capacities]} \end{array}$$

— an order is executed with no delay

— order $p_{t,i}$ costs you $c_i(t)p_{t,i}$.

• The goal: *to minimize the total cost of the orders.*

♠ *With certain demand*, the problem can be modeled as the LO program

$$\min_{\substack{p_{t,i},i\le I,t\le T,\\ v_t,2\le t\le T+1}} \sum_{t,i} c_i(t)p_{t,i} \qquad \text{[total cost]}$$

s.t.

$$\begin{aligned}
&v_{t+1} = v_t + \sum_i p_{t,i} - d_t,\ t = 1,...,T && \left[\begin{array}{c}\text{state equations}\\ (v_1 \text{ is given})\end{array}\right]\\
&V_{\min} \le v_t \le V_{\max}, 2 \le t \le T+1 && \text{[bounds on states]}\\
&0 \le p_{t,i} \le P_i(t),\ i \le I, t \le T && \text{[bounds on orders]}\\
&\sum_t p_{t,i} \le Q_i,\ i \le I && \left[\begin{array}{c}\text{cumulative bounds}\\ \text{on orders}\end{array}\right]
\end{aligned}$$

♠ *With uncertain demand*, it is natural to assume that the orders $p_{t,i}$ may depend on the demands of the preceding periods $1,...,t-1$. The *analysis variables* v_t are allowed to depend on the entire actual data. In fact, it suffices to allow for v_t to depend on $d_1,...,d_{t-1}$.

♠ Applying the AARC methodology, we make $p_{t,i}$ and v_t affine functions of past demands:

$$\begin{aligned}
p_{t,i} &= \phi^0_{t,i} + \sum_{1\le\tau<t} \phi^\tau_{t,i} d_\tau\\
v_t &= \psi^0_t + \sum_{1\le\tau<t} \psi^\tau_t d_\tau
\end{aligned}$$

- *ϕ's and ψ's are our new decision variables...*

$$\begin{array}{|c|}\hline \min_{\{p_{t,i},v_t\}} \sum_{t,i} c_i(t)p_{t,i} \quad \text{s.t.} \\ v_{t+1} = v_t + \sum_i p_{t,i} - d_t,\ t = 1,...,T \\ V_{\min} \le v_t \le V_{\max}, 2 \le t \le T+1 \\ 0 \le p_{t,i} \le P_i(t),\ i \le I, t \le T \\ \sum_t p_{t,i} \le Q_i,\ i \le I \\ \hline \begin{array}{rcl} p_{t,i} & = & \phi^0_{t,i} + \sum_{1\le\tau<t} \phi^\tau_{t,i} d_\tau \\ v_t & = & \psi^0_t + \sum_{1\le\tau<t} \psi^\tau_t d_\tau \end{array} \\ \hline \end{array}$$

♠ The AARC is the *semi-infinite* LO in non-adjustable variables ϕ's and ψ's:

$$\min_{C,\{\phi^\tau_{t,i},\psi^\tau_t\}} C \quad \text{s.t.} \quad \left\{ \begin{array}{l} \sum_{t,i} c_i(t) \left[\phi^0_{t,i} + \sum_{1\le\tau<t} \phi^\tau_{t,i} d_\tau\right] \le C \\ \left[\psi^0_{t+1} + \sum_{\tau=1}^{t} \psi^\tau_{t+1} d_\tau\right] = \left[\psi^0_t + \sum_{\tau=1}^{t-1} \psi^\tau_t d_\tau\right] + \sum_i \left[\phi^0_{t,i} + \sum_{\tau=1}^{t-1} \phi^\tau_{t,i} d_\tau\right] - d_t \\ V_{\min} \le \left[\psi^0_t + \sum_{\tau=1}^{t-1} \psi^\tau_t d_\tau\right] \le V_{\max} \\ 0 \le \left[\phi^0_{t,i} + \sum_{\tau=1}^{t-1} \phi^\tau_{t,i} d_\tau\right] \le P_i(t),\ \sum_t \left[\phi^0_{t,i} + \sum_{\tau=1}^{t-1} \phi^\tau_{t,i} d_\tau\right] \le Q_i \end{array} \right.$$

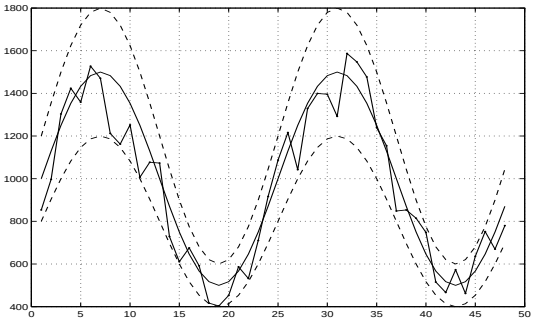
• The constraints should be valid for all values of "free" indexes and *all demand trajectories* $d = \{d_t\}_{t=1}^T$ *from the "demand uncertainty box"*

$$\mathcal{D} = \{d : d^*_t(1-\theta) \le d_t \le d^*_t(1+\theta), 1 \le t \le T\}.$$

♠ *The AARC can be straightforwardly converted to a usual LP and easily solved.*

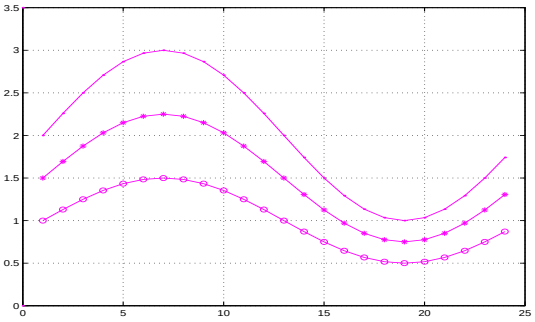
♣ In the numerical illustration to follow:

- the planning horizon is $T = 24$
- there are $I = 3$ factories with per period capacities $P_i(t) = 567$ and cumulative capacities $Q_i = 13600$
- the nominal demand d_t^* is seasonal:



$$d_t^* = 1000\left(1 + 0.5\sin\left(\frac{\pi(t-1)}{12}\right)\right)$$

- the ordering costs also are seasonal:



$$c_i(t) = c_i\left(1 + 0.5\sin\left(\frac{\pi(t-1)}{12}\right)\right),\ c_1 = 1, c_2 = 1.5, c_3 = 2$$

- $v_1 = V_{\min} = 500$, $V_{\max} = 2000$
- demand uncertainty $\theta = 20\%$

♣ **Results:**

• Opt(AARC) = 35542.

Note: The *non-adjustable* RC is infeasible already at 5% uncertainty level!

• With uniformly distributed in the range ±20% demand perturbations, the average, over 100 simulations, AARC management cost is 35121.

Note: Over the same 100 simulations, the average *"utopian"* management cost (optimal for *a priori known* demand trajectories) is 33958, i.e., is by just 3.5% (!) less than the average AARC management cost.

♣ **Comparison with Dynamic Programming.** *When applicable*, DP is the technique for dynamical decision-making under uncertainty – in (worst-case-oriented) DP, one solves the Adjustable Robust Counterpart of uncertain LO, *with no ad hoc simplifications like "let us restrict ourselves with affine decision rules."*

♠ Unfortunately, DP suffers from *"curse of dimensionality"* – with DP, the computational effort blows up rapidly as the state dimension of the dynamical process grows. Usually state dimension 4 is already "too big".

Note: There is no "curse of dimensionality" in AARC!

However: Reducing the number of factories to 1, increasing the per period capacity of the remaining factory to 1800 and making its cumulative capacity $+\infty$, we reduce the state dimension to 1 and make DP easily implementable. With this setup,

- the DP (that is, the "absolutely best") optimal value is 31270
- the AARC optimal value is 31514 – just by 0.8% worse!

PART II.
LO: Pivoting Algorithms

ALGORITHMS OF LINEAR OPTIMIZATION

♣ The existing algorithmic "working horses" of LO fall into two major categories:

♠ Pivoting methods, primarily the *Simplex-type algorithms* which heavily exploit the polyhedral structure of LO programs, in particular, move along the vertices of the feasible set.

♠ Interior Point algorithms, primarily the *Primal-Dual Path-Following Methods*, much less "polyhedrally oriented" than the pivoting algorithms and, in particular, traveling along interior points of the feasible set of LO rather than along its vertices. In fact, IPM's have a much wider scope of applications than LO.

♠ Theoretically speaking (and modulo rounding errors), pivoting algorithms solve LO programs *exactly* in *finitely many* arithmetic operations. The operation count, however, can be astronomically large already for small LO's. In contrast to the disastrously bad theoretical worst-case-oriented performance estimates, *Simplex-type algorithms seem to be extremely efficient in practice*. In 1940's — early 1990's these algorithms were, essentially, *the only* LO solution techniques.

♠ Interior Point algorithms, discovered in 1980's, entered LO practice in 1990's. These methods combine high practical performance (quite competitive with the one of pivoting algorithms) with nice theoretical worst-case-oriented efficiency guarantees.

Lecture II.1
Primal and Dual Simplex Methods

The Primal and the Dual Simplex Algorithms

♣ Recall that a primal-dual pair of LO problems geometrically is as follows:

Given are:

• *A pair of linear subspaces* $\mathcal{L}_P$, $\mathcal{L}_D$ *in* $\mathbb{R}^n$ *which are orthogonal complements to each other:* $\mathcal{L}_P{}^\perp = \mathcal{L}_D$

• *Shifts of these linear subspaces – the primal feasible plane* $\mathcal{M}_P = \mathcal{L}_P - b$ *and the dual feasible plane* $\mathcal{M}_D = \mathcal{L}_D + c$

The goal:

• *We want to find a pair of orthogonal to each other vectors, one from the primal feasible set* $\mathcal{M}_P \cap \mathbb{R}^n_+$, *and another one from the dual feasible set* $\mathcal{M}_D \cap \mathbb{R}^n_+$.

This goal is achievable iff the primal and the dual feasible sets are nonempty, and in this case the members of the desired pair can be chosen among extreme points of the respective feasible sets.

♣ In the Primal Simplex method, the goal is achieved via generating a *sequence* $x^1, x^2, ...$ *of vertices of the primal feasible set* accompanied by a sequence $c^1, c^2, ...$ of solutions to the dual problem which belong to the dual feasible plane, satisfy the orthogonality requirement $[x^t]^T c^t = 0$, but do not belong to $\mathbb{R}^n_+$ (and thus are not dual feasible).

The process lasts until

— either a feasible dual solution c^t is generated, in which case we end up with a pair of primal-dual optimal solutions (x^t, c^t),

— or a certificate of primal unboundedness (and thus - dual infeasibility) is found.

♣ In the Dual Simplex method, the goal is achieved via generating a *sequence $c^1, c^2, ...$ of vertices of the dual feasible set* accompanied by a sequence $x^1, x^2, ...$ of solutions to the primal problem which belong to the primal feasible plane, satisfy the orthogonality requirement $[x^t]^T c^t = 0$, but do not belong to $\mathbb{R}^n_+$ (and thus are not primal feasible).
The process lasts until
— either a feasible primal solution x^t is generated, in which case we end up with a pair of primal-dual optimal solutions (x^t, c^t),
— or a certificate of dual unboundedness (and thus – primal infeasibility) is found.

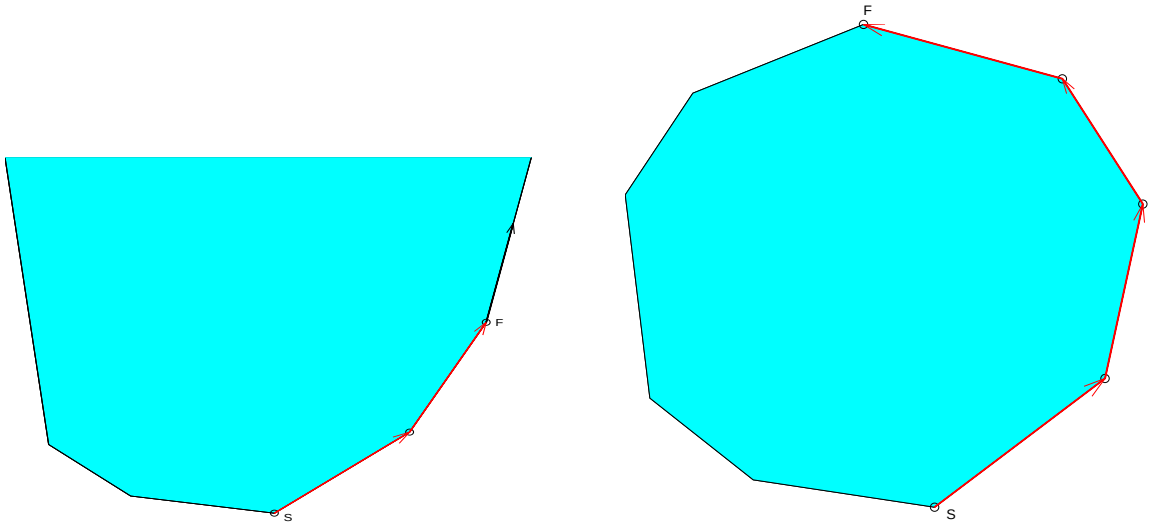
♣ Both methods work with the primal LO program *in the standard form*. As a result, the dual problem is *not* in the standard form, which makes the implementations of the algorithms different from each other, in spite of the "geometrical symmetry" of the algorithms.

Primal Simplex Method

♣ PSM works with an LO program *in the standard form*

$$\mathrm{Opt}(P) = \max_x \left\{c^T x : Ax = b, x \geq 0\right\} \ [A : m \times n] \qquad (P)$$

Geometrically, PSM moves from a vertex x^t of the feasible set of (P) to a neighboring vertex x^{t+1}, improving the value of the objective, until either an optimal solution, or an unboundedness certificate are built. This process is "guided" by the dual solutions c^t.



Geometry of PSM. The objective is the ordinate ("height"). **Left:** method starts from vertex S and ascends to vertex F where an improving ray is discovered, meaning that the problem is unbounded.

Right: method starts from vertex S and ascends to the optimal vertex F.

PSM: Preliminaries

$$\mathrm{Opt}(P) = \max_x \left\{c^T x : Ax = b, x \geq 0\right\} \ [A : m \times n] \qquad (P)$$

♣ **Standing assumption:** *The m rows of A are linearly independent.*

Note: When the rows of A are linearly dependent, the system of linear equations in (P) is either infeasible (this is so when $\mathrm{Rank}\, A < \mathrm{Rank}\, [A, b]$), or, eliminating "redundant" linear equations, the system $Ax = b$ can be reduced to an equivalent system $A'x = b$ with linearly independent rows in A'. When possible, this transformation is an easy Linear Algebra task

$\Rightarrow$ *the assumption* $\mathrm{Rank}\, A = m$ *is w.l.o.g.*

♣ **Bases and basic solutions.** A set I of m distinct from each other indexes from $\{1, ..., n\}$ is called a *base* (or *basis*) of A, if the columns of A with indexes from I are linearly independent, or, equivalently, *when the $m \times m$ submatrix A_I of A composed of columns with indexes from I is nonsingular.*

$$\mathrm{Opt}(P) = \max_x \left\{ c^T x : Ax = b, x \geq 0 \right\} \ [A : m \times n] \qquad (P)$$

Observation I: *Let I be a basis. Then the system $Ax = b, x_i = 0, i \notin I$ has a unique solution*

$$x^I = (A_I^{-1} b, 0_{n-m});$$

the m entries of x^I with indexes from I form the vector $A_I^{-1} b$, and the remaining $n - m$ entries are zero. x^I is called the basic solution associated with basis I.

$$\mathrm{Opt}(P) = \max_x \left\{c^T x : Ax = b, x \geq 0\right\} \ [A : m \times n] \qquad (P)$$

Theorem: *Extreme points of the feasible set $X = \{x : Ax = b, x \geq 0\}$ of (P) are exactly the basic solutions which are feasible for (P).*

Proof. A. Let x^I be a basic solution which is feasible. The following n constraints specifying X are active at x^I:

— m equality constraints $Ax = b$;

— $n - m$ bounds $x_j = 0$, $j \notin I$.

Let us prove that the vectors of coefficients of these n constraints are linearly independent; by algebraic characterization of extreme points, this would imply that x^I is an extreme point of X.

Assuming the vectors of coefficients of the constraints to be linearly dependent, there exists a nontrivial solution h to the homogeneous system of linear equations

$$Ah = 0,\ h_j = 0, j \notin I,$$

implying that the corresponding to I $m \times m$ basic submatrix of A is degenerate, which is impossible.

B. Let $\bar{x}$ be an extreme point of X, and let J be the set of indexes of positive entries in $\bar{x}$. We claim that the columns of A with indexes from J are linearly independent.

Indeed, otherwise we could find a nonzero vector h with entries $h_j = 0$ for $j \notin J$ such that $Ah = 0$. For small positive t, vectors $\bar{x} + th$ and $\bar{x} - th$ are nonnegative, and we always have $A[\bar{x} + th] = A[\bar{x} - th] = b$, so that $\bar{x}$ is a midpoint of a nontrivial segment in X, which is impossible.

• Since the columns of A with indexes in J are linearly independent and the linear span of the system of all columns of A is $\mathbb{R}^m$ (the m rows of A are linearly independent!), we can extend J to an m-element set I of indexes such that the columns of A with indexes from I are linearly independent. Clearly, $\bar{x}$ is the basic solution associated with the basis I, that is, every extreme point of X is a basic feasible solution.

$$\mathrm{Opt}(P) = \max_x \left\{ c^T x : Ax = b, x \geq 0 \right\} \ [A : m \times n] \qquad (P)$$

Note: Theorem says that extreme points of $X = \{x : Ax = b, x \geq 0\}$ can be "parameterized" by bases I of (P): every *feasible* basic solution x^I is an extreme point, and vice versa. Note that this parameterization in general is *not* a one-to-one correspondence: while there exists *at most* one extreme point with entries vanishing outside of a given basis, and *every* extreme point can be obtained from appropriate basis, there could be several bases defining the same extreme point. The latter takes place when the extreme point v in question is *degenerate* – it has less than m positive entries. Whenever this is the case, *all* bases containing all the indexes of the nonzero entries of v specify the same vertex v.

$$\mathrm{Opt}(P) = \max_x \left\{c^T x : Ax = b, x \geq 0\right\} \ [A : m \times n] \qquad (P)$$

♣ The problem dual to (P) reads

$$\mathrm{Opt}(D) = \min_{\lambda=[\lambda_e;\lambda_g]} \left\{b^T\lambda_e : \lambda_g \leq 0, \lambda_g = c - A^T\lambda_e\right\} \quad (\mathrm{D})$$

Observation II: *Given a basis I for (P), we can uniquely define a solution $\lambda^I = [\lambda_e^I; c^I]$ to (D) which satisfies the equality constraints in (D) and is such that c^I vanishes on I. Specifically, setting $I = \{i_1 < i_2 < ... < i_m\}$ and $c_I = [c_{i_1}; c_{i_2}; ...; c_{i_m}]$, one has*

$$\lambda_e^I = A_I^{-T} c_I, \quad c^I = c - A^T A_I^{-T} c_I.$$

Vector c^I is called the vector of reduced costs associated with basis I.

Observation III: *Let I be a basis of A. Then for every x satisfying the equality constraints of (P) one has*

$$c^T x = [c^I]^T x + \mathrm{const}(I)$$

Indeed, $[c^I]^T x = [c - A^T\lambda_e^I]^T x = c^T x - [\lambda_e^I]^T Ax = c^T x - [\lambda_e^I]^T b$. □

Observation IV: Let I be a basis of A which corresponds to a *feasible* basic solution x^I and *nonpositive* vector of reduced costs c^I. Then x^I is an optimal solution to (P), and λ^I is an optimal solution to (D).

Indeed, in the case in question x^I and λ^I are feasible solutions to (P) and (D) satisfying the complementary slackness condition. $\square$

Step of the PSM

$$\mathrm{Opt}(P) = \max_x \left\{c^T x : Ax = b, x \geq 0\right\} \ [A : m \times n] \qquad (P)$$

At the beginning of a step of PSM we have at our disposal

- current basis I along with the corresponding *feasible* basic solution x^I, and
- the vector c^I of reduced costs associated with I.

We call variables x_i *basic*, if $i \in I$, and *non-basic* otherwise. Note that all non-basic variables in x^I are zero, while basic ones are nonnegative.

At a step we act as follows:

♠ We check whether $c^I \leq 0$. If it is the case, we terminate with optimal primal solution x^I and "optimality certificate" – optimal dual solution $[\lambda_e^I; c^I]$

$$\mathrm{Opt}(P) = \max_x \left\{c^T x : Ax = b, x \geq 0\right\} \ [A : m \times n] \qquad (P)$$

When c^I is *not* nonpositive, we proceed as follows:

♠ We select an index j such that $c^I_j > 0$. Since $c^I_i = 0$ for $i \in I$, we have $j \notin I$; our intention is to update the current basis I into a new basis I^+ (which, same as I is associated with a *feasible* basic solution), by adding to the basis the index j ("*non-basic variable x_j enters the basis*") and discarding from the basis an appropriately chosen index $i_* \in I$ ("*basic variable x_{i_*} leaves the basis*"). Specifically,

- We look at the solutions $x(t)$, $t \geq 0$ is a parameter, defined by

$$Ax(t) = b \ \& \ x_j(t) = t \ \& \ x_\ell(t) = 0 \, \forall \ell \notin [I \cup \{j\}]$$

$$\Leftrightarrow x_i(t) = \begin{cases} x^I_i - t[A_I^{-1} A_j]_i, & i \in I \\ t, & i = j \\ 0, & \text{all other cases} \end{cases}$$

Observation VI: (a): $x(0) = x^I$ and (b): $c^T x(t) - c^T x^I = c^I_j t$.

(a) is evident. By Observation IV

$$c^T[x(t) - x(0)] = [c^I]^T[x(t) - x(0)] = \sum_{i \in I} \overbrace{c^I_i}^{=0} [x_i(t) - x_i(0)] + c^I_j[x_j(t) - x_j(0)] = c^I_j t.$$

$$\text{Opt}(P) = \max_x \left\{ c^T x : Ax = b, x \geq 0 \right\} \ [A : m \times n] \qquad (P)$$

Situation: • I, x^I, c^I: a basis of A and the associated basic feasible solution and reduced costs.

• $j \notin I$: $c_j^I > 0$

• $Ax(t) = b$ & $x_j(t) = t$ & $x_\ell(t) = 0 \, \forall \ell \notin [I \cup \{j\}]$

$$\Leftrightarrow x_i(t) = \begin{cases} x_i^I - t[A_I^{-1} A_j]_i, & i \in I \\ t, & i = j \\ 0, & \text{all other cases} \end{cases}$$

• $c^T[x(t) - x^I] = c_j^I t$

There are two options:

A. All quantities $[A_I^{-1} A_j]_i$, $i \in I$, are ≤ 0. Here $x(t)$ is feasible for all $t \geq 0$ and

$$c^T[x(t) - x(0)] = c_j^I t \to \infty \text{ as } t \to \infty.$$

We claim (P) unbounded and terminate.

B. $I_* := \{i \in I : [A_I^{-1} A_j]_i > 0\} \neq \emptyset$. We set

$$\bar{t} = \min \left\{ \frac{x_i^I}{[A_I^{-1} A_j]_i} : i \in I_* \right\} = \frac{x_{i_*}^I}{[A_I^{-1} A_j]_{i_*}} \text{ with } i_* \in I_* \ .$$

Observe that $x(\bar{t}) \geq 0$ and $x_{i_*}(\bar{t}) = 0$. We set $I^+ = I \cup \{j\} \backslash \{i_*\}$, $x^{I^+} = x(\bar{t})$, compute the vector of reduced costs c^{I^+} and pass to the next step of the PSM, with I^+, x^{I^+}, c^{I^+} in the roles of I, x^I, c^I.

Summary

♠ The PSM works with a standard form LO

$$\max_x \left\{ c^T x : Ax = b, x \geq 0 \right\} \qquad [A : m \times n] \qquad (P)$$

♠ At the beginning of a step, we have

• current *basis* I - a set of m indexes such that the columns of A with these indexes are linearly independent

• current *basic feasible solution* x^I such that $Ax^I = b$, $x^I \geq 0$ and all nonbasic – with indexes not in I – entries of x^I are zeros

♠ At a step, we

• A. Compute the *vector of reduced costs* $c^I = c - A^T\lambda_e^I$ such that the basic entries in c^I are zeros.

• If $c^I \leq 0$, we terminate — x^I is an optimal solution.

• Otherwise we pick j with $c_j^I > 0$ and build a "ray of solutions" $x(t) = x^I + th$ such that $Ax(t) \equiv b$, *and* $x_j(t) \equiv t$ *is the only nonbasic entry in* $x(t)$ *which can be* $\neq 0$.

• We have $x(0) = x^I$, $c^T x(t) - c^T x^I = c_j^I t$.

• If no basic entry in $x(t)$ decreases as t grows, we terminate: (P) is unbounded.

• If some basic entries in $x(t)$ decrease as t grows, we choose the largest $t = \bar{t}$ such that $x(t) \geq 0$. When $t = \bar{t}$, at least one of the basic entries of $x(t)$ is about to become negative, and we eliminate its index i_* from I, adding to I the index j instead ("variable x_j enters the basis, variable x_{i_*} leaves it").

♠ $I^+ = [I \cup \{j\}]\backslash\{i_*\}$ is our new basis, $x^{I^+} = x(\bar{t})$ is our new basic feasible solution, and we pass to the next step.

Remarks:

• *I^+, when defined, is a basis of A, and in this case $x(\bar{t})$ is the corresponding basic feasible solution. $\Rightarrow$ The PSM is well defined*

• *Upon termination* (if any), *the PSM correctly solves the problem and, moreover,*

— either returns extreme point optimal solutions to the primal and the dual problems,

— or returns a certificate of unboundedness - a (vertex) feasible solution x^I along with a direction $d = \frac{d}{dt}x(t) \in \mathrm{Rec}(\{x : Ax = b, x \geq 0\})$ which is an improving direction of the objective: $c^T d > 0$. On a closest inspection, d is an extreme direction of $\mathrm{Rec}(\{x : Ax = b, x \geq 0\})$.

• *The method is monotone: $c^T x^{I^+} \geq c^T x^I$. The latter inequality is strict, unless $x^{I^+} = x^I$, which may happen only when x^I is a degenerate basic feasible solution.*

• *If all basic feasible solutions to (P) are nondegenerate, the PSM terminates in finite time.*

Indeed, in this case the objective strictly grows from step to step, meaning that the same basis cannot be visited twice. Since there are finitely many bases, the method must terminate in finite time.

Tableau Implementation of the PSM

$$\mathrm{Opt}(P) = \max_x \left\{ c^T x : Ax = b, x \geq 0 \right\} \ [A : m \times n] \qquad (P)$$

♣ In computations by hand it is convenient to implement the PSM in the tableau form. The tableau summarizes the information we have at the beginning of the step, and is updated from step to step by easy to memorize rules.

♠ The structure of a tableau is as follows:

	x_1	x_2	$\cdots$	x_n
$-c^T x^I$	c_1^I	c_2^I	$\cdots$	c_n^I
$x_{i_1} =< \ldots >$	$[A_I^{-1}A]_{1,1}$	$[A_I^{-1}A]_{1,2}$	$\cdots$	$[A_I^{-1}A]_{1,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x_{i_n} =< \ldots >$	$[A_I^{-1}A]_{m,1}$	$[A_I^{-1}A]_{m,2}$	$\cdots$	$[A_I^{-1}A]_{m,n}$

- Zeroth row: minus the current value of the objective and the current reduced costs
- Rest of Zeroth column: the names and the values of current basic variables
- Rest of the tableau: the $m \times n$ matrix $A_I^{-1}A$

Note: In the Tableau, all columns but the Zeroth one are labeled by decision variables, and *all rows but the Zeroth one are labeled by the current basic variables.*

♣ To illustrate the Tableau implementation, consider the LO problem

$$\begin{array}{ll} \max & 10x_1 + 12x_2 + 12x_3 \\ \text{s.t.} & \end{array}$$

$$\begin{array}{rcrcrcl} x_1 & + & 2x_2 & + & 2x_3 & \leq & 20 \\ 2x_1 & + & x_2 & + & 2x_3 & \leq & 20 \\ 2x_1 & + & 2x_2 & + & x_3 & \leq & 20 \end{array}$$

$$x_1, x_2, x_3 \geq 0$$

♠ Adding slack variables, we convert the problem into the standard form

$$\begin{array}{ll} \max & 10x_1 + 12x_2 + 12x_3 \\ \text{s.t.} & \end{array}$$

$$\begin{array}{rcrcrcrcrcl} x_1 & + & 2x_2 & + & 2x_3 & + & x_4 & & & = & 20 \\ 2x_1 & + & x_2 & + & 2x_3 & & & + & x_5 & = & 20 \\ 2x_1 & + & 2x_2 & + & x_3 & & & & & = & 20 \end{array} \quad \begin{array}{rcrcl} & & & + & \\ & & & & \\ & & & + & x_6 \end{array} \quad \begin{array}{l} \\ \\ = \ 20 \end{array}$$

$$x_1, ..., x_6 \geq 0$$

• We can take $I = \{4, 5, 6\}$ as the initial basis, $x^I = [0; 0; 0; 20; 20; 20]$ as the corresponding basic feasible solution, and c as c^I. The first tableau is

	x_1	x_2	x_3	x_4	x_5	x_6
0	10	12	12	0	0	0
$x_4 = 20$	1	2	2	1	0	0
$x_5 = 20$	2	1	2	0	1	0
$x_6 = 20$	2	2	1	0	0	1

	x_1	x_2	x_3	x_4	x_5	x_6
0	10	12	12	0	0	0
$x_4 = 20$	1	2	2	1	0	0
$x_5 = 20$	2	1	2	0	1	0
$x_6 = 20$	2	2	1	0	0	1

- Not all reduced costs (Zeroth row of the Tableau) are nonpositive.

A: Detecting variable to enter the basis:

We choose a variable with positive reduced cost, specifically, x_1, which is about to enter the basis, and call the corresponding column in the Tableau the pivoting column.

	x_1	x_2	x_3	x_4	x_5	x_6
0	10	12	12	0	0	0
$x_4 = 20$	1	2	2	1	0	0
$x_5 = 20$	2	1	2	0	1	0
$x_6 = 20$	2	2	1	0	0	1

B: Detecting variable to leave the basis:

B.1. We select the positive entries in the pivoting column (not looking at the Zeroth row). If there is nothing to select, (P) is unbounded, and we terminate.

B.2. We divide by the selected entries of the pivoting column the corresponding entries of the Zeroth column, thus getting ratios $20 = 20/1$, $10 = 20/2$, $10 = 20/2$.

B.3. We select the smallest among the ratios in B.2 (this quantity is the above $\bar{t}$) and call the corresponding row the pivoting one. The basic variable labeling this row leaves the basis. In our case, we choose as pivoting row the one labeled by x_5; this variable leaves the basis.

	x_1	x_2	x_3	x_4	x_5	x_6
0	10	12	12	0	0	0
$x_4 = 20$	1	2	2	1	0	0
$x_5 = 20$	2	1	2	0	1	0
$x_6 = 20$	2	2	1	0	0	1

C. Updating the tableau:

C.1. We divide all entries in the pivoting row by the *pivoting element* (the one which is in the pivoting row and pivoting column) and change the label of the pivoting row to the name of the variable entering the basis:

	x_1	x_2	x_3	x_4	x_5	x_6
0	10	12	12	0	0	0
$x_4 = 20$	1	2	2	1	0	0
$x_1 = 10$	1	0.5	1	0	0.5	0
$x_6 = 20$	2	2	1	0	0	1

C.2. We subtract from all non-pivoting rows (including the Zeroth one) multiples of the (updated) pivoting row to zero out the entries in the pivoting column:

	x_1	x_2	x_3	x_4	x_5	x_6
-100	0	7	2	0	-5	0
$x_4 = 10$	0	1.5	1	1	-0.5	0
$x_1 = 10$	1	0.5	1	0	0.5	0
$x_6 = 0$	0	1	-1	0	-1	1

The step is over.

	x_1	x_2	x_3	x_4	x_5	x_6
-100	0	7	2	0	-5	0
$x_4 = 10$	0	1.5	1	1	-0.5	0
$x_1 = 10$	1	0.5	1	0	0.5	0
$x_6 = 0$	0	1	-1	0	-1	1

A: There still are positive reduced costs in the Zeroth row. We choose x_3 to enter the basis; the column of x_3 is the pivoting column.

B: We select positive entries in the pivoting column (except for the Zeroth row), divide by the selected entries the corresponding entries in the zeroth column, the getting the ratios 10/1, 10/1, and select the minimal ratio, breaking the ties arbitrarily. Let us select the first ratio as the minimal one. The corresponding – the first – row becomes the pivoting one, and the variable x_4 labeling this row leaves the basis.

	x_1	x_2	x_3	x_4	x_5	x_6
-100	0	7	2	0	-5	0
$x_4 = 10$	0	1.5	1	1	-0.5	0
$x_1 = 10$	1	0.5	1	0	0.5	0
$x_6 = 0$	0	1	-1	0	-1	1

	x_1	x_2	x_3	x_4	x_5	x_6
-100	0	7	2	0	-5	0
$x_4 = 10$	0	1.5	$\underline{1}$	1	-0.5	0
$x_1 = 10$	1	0.5	1	0	0.5	0
$x_6 = 0$	0	1	-1	0	-1	1

C: We identify the pivoting element (underlined), divide by it the pivoting row and update the label of this row:

	x_1	x_2	x_3	x_4	x_5	x_6
-100	0	7	2	0	-5	0
$x_3 = 10$	0	1.5	$\underline{1}$	1	-0.5	0
$x_1 = 10$	1	0.5	1	0	0.5	0
$x_6 = 0$	0	1	-1	0	-1	1

We then subtract from non-pivoting rows the multiples of the pivoting row to zero the entries in the pivoting column:

	x_1	x_2	x_3	x_4	x_5	x_6
-120	0	4	0	-2	-4	0
$x_3 = 10$	0	1.5	$\underline{1}$	1	-0.5	0
$x_1 = 0$	1	-1	0	-1	1	0
$x_6 = 10$	0	2.5	0	1	-1.5	1

The step is over.

	x_1	x_2	x_3	x_4	x_5	x_6
-120	0	4	0	-2	-4	0
$x_3 = 10$	0	1.5	1	1	-0.5	0
$x_1 = 0$	1	-1	0	-1	1	0
$x_6 = 10$	0	2.5	0	1	-1.5	1

A: There still is a positive reduced cost. Variable x_2 enters the basis, the corresponding column becomes the pivoting one:

	x_1	x_2	x_3	x_4	x_5	x_6
-120	0	4	0	-2	-4	0
$x_3 = 10$	0	1.5	1	1	-0.5	0
$x_1 = 0$	1	-1	0	-1	1	0
$x_6 = 10$	0	2.5	0	1	-1.5	1

B: We select the positive entries in the pivoting column (aside of the Zeroth row) and divide by them the corresponding entries in the Zeroth column, thus getting ratios 10/1.5, 10/2.5. The minimal ratio corresponds to the row labeled by x_6. This row becomes the pivoting one, and x_6 leaves the basis:

	x_1	x_2	x_3	x_4	x_5	x_6
-120	0	4	0	-2	-4	0
$x_3 = 10$	0	1.5	1	1	-0.5	0
$x_1 = 0$	1	-1	0	-1	1	0
$x_6 = 10$	0	2.5	0	1	-1.5	1

	x_1	x_2	x_3	x_4	x_5	x_6
-120	0	4	0	-2	-4	0
$x_3 = 10$	0	1.5	1	1	-0.5	0
$x_1 = 0$	1	-1	0	-1	1	0
$x_6 = 10$	0	$\underline{2.5}$	0	1	-1.5	1

C: We divide the pivoting row by the pivoting element (underlined), change the label of this row:

	x_1	x_2	x_3	x_4	x_5	x_6
-120	0	4	0	-2	-4	0
$x_3 = 10$	0	1.5	1	1	-0.5	0
$x_1 = 0$	1	-1	0	-1	1	0
$x_2 = 4$	0	$\underline{1}$	0	0.4	-0.6	0.4

and then subtract from all non-pivoting rows multiples of the pivoting row to zero the entries in the pivoting column:

	x_1	x_2	x_3	x_4	x_5	x_6
-136	0	0	0	-3.6	-1.6	-1.6
$x_3 = 4$	0	0	1	0.4	0.4	-0.6
$x_1 = 4$	1	0	0	-0.6	0.4	0.4
$x_2 = 4$	0	$\underline{1}$	0	0.4	-0.6	0.4

The step is over.

	x_1	x_2	x_3	x_4	x_5	x_6
-136	0	0	0	-3.6	-1.6	-1.6
$x_3 = 4$	0	0	1	0.4	0.4	-0.6
$x_1 = 4$	1	0	0	-0.6	0.4	0.4
$x_2 = 4$	0	$\underline{1}$	0	0.4	-0.6	0.4

• The new reduced costs are nonpositive $\Rightarrow$ the solution $x_* = [4; 4; 4; 0; 0; 0; 0]$ is optimal. The dual optimal solution is

$$\lambda_e = [3.6; 1.6; 1.6],\ c^I = [0; 0; -3.6; -1.6; -1.6],$$

the optimal value is 136.

$$\mathrm{Opt}(P) = \max_x \left\{c^T x : Ax = b, x \geq 0\right\} \ [A : m \times n] \qquad (P)$$

♣ **Getting started.**

In order to run the PSM, we should initialize it with some *feasible* basic solution. The standard way to do it is as follows:

♠ Multiplying appropriate equality constraints in (P) by -1, we ensure that $b \geq 0$.

♠ We then build an auxiliary LO program

$$\mathrm{Opt}(P^0) = \min_{x,s} \left\{\sum_{i=1}^m s_i : Ax + s = b,\ x \geq 0, s \geq 0\right\} \qquad (P^0)$$

Observations:

- $\mathrm{Opt}(P^0) \geq 0$, *and* $\mathrm{Opt}(P^0) = 0$ *iff* (P) *is feasible;*
- (P^0) *admits an evident basic feasible solution* $x = 0, s = b$*, the basis being* $I = \{n+1, ..., n+m\}$*;*
- *When* $\mathrm{Opt}(P^0) = 0$*, the* x*-part* x^* *of every vertex optimal solution* $(x^*, s^* = 0)$ *of* (P^0) *is a feasible basic solution to* (P)*.*

Indeed, the columns A_i with indexes i such that $x_i^* > 0$ are linearly independent $\Rightarrow$ the set $I' = \{i : x_i^* > 0\}$ can be extended to a basis I of A, x^* being the associated basic feasible solution x^I.

$$\mathrm{Opt}(P) = \max_x \left\{c^T x : Ax = b \geq 0, x \geq 0\right\} \qquad (P)$$
$$\mathrm{Opt}(P^0) = \min_{x,s} \left\{\sum_{i=1}^m s_i : Ax + s = b,\ x \geq 0, s \geq 0\right\} \qquad (P^0)$$

♠ In order to initialize solving (P) by the PSM, we start with *phase I* where we solve (P^0) by the PSM, the initial feasible basic solution being $x = 0, s = b$. Upon termination of Phase I, we have at our disposal

— either an optimal solution $x^*, s^* \neq 0$ to (P^0); it happens when $\mathrm{Opt}(P^0) > 0$

$\Rightarrow$ (P) is infeasible,

— or an optimal solution $x^*, s^* = 0$; in this case x^* is a feasible basic solution to (P), and we pass to Phase II, where we solve (P) by the PSM initialized by x^* and the corresponding basis I.

$$\mathrm{Opt}(P) = \max_x \left\{c^T x : Ax = b \geq 0, x \geq 0\right\} \qquad (P)$$
$$\mathrm{Opt}(P^0) = \min_{x,s} \left\{\sum_{i=1}^m s_i : Ax + s = b,\ x \geq 0, s \geq 0\right\} \qquad (P^0)$$

Note: Phase I yields, along with x^*, s^*, an optimal solution $[\lambda^*; \widetilde{c}]$ to the problem dual to (P^0):

$$\left\{\begin{array}{rcl} 0 \leq \widetilde{c} & := & [0_{n\times 1}; 1; ...; 1] - [A, I_m]^T \lambda^* \\ & = & \left[-A^T\lambda^*; [1; ...; 1] - \lambda^*\right] \\ 0 & = & \widetilde{c}^T [x^*; s^*] \end{array}\right.$$
$$\Rightarrow 0 = -[x^*]^T A^T \lambda^* + \mathrm{Opt}(P^0) - [s^*]^T \lambda^* \ \&\ A^T\lambda^* \leq 0$$
$$\Rightarrow \mathrm{Opt}(P^0) = b^T\lambda^* \ \&\ A^T\lambda^* \leq 0$$

Recall that the standard certificate of infeasibility for (P) is a pair $\lambda_g \leq 0, \lambda_e$ such that

$$\lambda_g \leq 0 \ \&\ A^T\lambda_e + \lambda_g = 0 \ \&\ b^T\lambda_e < 0.$$

We see that *when* $\mathrm{Opt}(P^0) > 0$, *the vectors* $\lambda_e = -\lambda^*, \lambda_g = A^T\lambda^*$ *form an infeasibility certificate for* (P).

Preventing Cycling

$$\mathrm{Opt}(P) = \max_x \left\{c^T x : Ax = b, x \geq 0\right\} \ [A : m \times n] \qquad (P)$$

♣ Finite termination of the PSM is fully guaranteed only when (P) is *nondegenerate*, that is, every basic feasible solution has *exactly* m nonzero entries, or, equivalently, *in every representation of* b *as a conic combination of linearly independent columns of* A*, the number of columns which get positive weights is* m*.*

♠ When (P) is degenerate, b belongs to the union of finitely many *proper* linear subspaces $E_J = \mathrm{Lin}(\{A_i : i \in J\}) \subset \mathbb{R}^m$, where J runs through the family of all $(m-1)$-element subsets of $\{1, ..., n\}$

$\Rightarrow$ *Degeneracy is a rare phenomenon: given* A*, the set of right hand side vectors* b *for which* (P) *is degenerate is of measure 0.*

However: There are specific problems with integer data where degeneracy can be typical.

♣ It turns out that cycling can be prevented by properly chosen pivoting rules which "break ties" in the cases when there are several candidates to entering the basis and/or several candidates to leave it.

Example 1: Bland's "smallest subscript" pivoting rule.

• *Given a current tableau containing positive reduced costs, find the smallest index j such that j-th reduced cost is positive; take x_j as the variable to enter the basis.*

• *Assuming that the above choice of the pivoting column does not lead to immediate termination due to problem's unboundedness, choose among all legitimate* (i.e., compatible with the description of the PSM) *candidates to the role of the pivoting row the one with the smallest index and use it as the pivoting row.*

Example 2: Lexicographic pivoting rules.

• $\mathbb{R}^N$ can be equipped with *lexicographic order*: $x >_L y$ if and only if $x \neq y$ and the first nonzero entry in $x - y$ is positive, and $x \geq_L y$ if and only if $x = y$ or $x >_L y$.

Examples: $[1; -1; -2] >_L [0; 100, 10]$, $[1; -1; -2] >_L [1; -2; 1000]$.

Note: $\geq_L$ and $>_L$ share the usual properties of the arithmetic $\geq$ and $>$, e.g.

- $x \geq_L x$
- $x \geq_L y$ & $y \geq_L x \Rightarrow y = x$
- $x \geq_L y \geq_L z \Rightarrow x \geq_L z$
- $x \geq_L y$ & $u \geq_L v \Rightarrow x + u \geq_L y + v$
- $x \geq_L y, \lambda \geq 0 \Rightarrow \lambda x \geq_L \lambda y$

In addition, the order $\geq_L$ is *complete:* every two vectors x, y from $\mathbb{R}^n$ are lexicographically comparable, i.e., either $x >_L y$, or $x = y$, or $y >_L x$.

Lexicographic pivoting rule (L)

• *Given the current tableau which contains positive reduced costs c^I_j, choose as the index of the pivoting column a j such that $c^I_j > 0$.*

• *Let u_i be the entry of the pivoting column in row $i = 1, ..., m$, and let some of u_i be positive* (that is, the PSM does not detect problem's unboundedness at the step in question). *Normalize every row i with $u_i > 0$ by dividing all its entries, including those in the zeroth column, by u_i, and choose among the resulting $(n+1)$-dimensional row vectors the smallest w.r.t the lexicographic order, let its index be i_** (it is easily seen that all rows to be compared to each other are distinct, so that the lexicographically smallest of them is uniquely defined). *Choose the row i_* as the pivoting one* (i.e., the basic variable labeling the row i_* leaves the basis).

Theorem: *Let the PSM be initialized by a basic feasible solution and use pivoting rule* (L). *Assume also that all rows in the initial tableau, except for the zeroth row, are lexicographically positive. Then*

- *the rows in the tableau, except for the zeroth one, remain lexicographically positive at all non-terminal steps,*
- *the vector of reduced costs strictly lexicographically decreases when passing from a tableau to the next one, whence no basis is visited twice,*
- *the PSM method terminates in finite time.*

Note: Lexicographical positivity of the rows in the initial tableau is easy to achieve: it suffices to reorder variables to make the initial basis to be $I = \{1, 2, ..., m\}$. In this case the non-zeroth rows in the initial tableau look as follows:

$$[A_I^{-1}b, A_I^{-1}A] = \left[\begin{array}{c|c|c|c|c|c|c|c|c} x_1^I \geq 0 & 1 & 0 & 0 & \cdots & 0 & ? & ? & \cdots \\ \hline x_2^I \geq 0 & 0 & 1 & 0 & \cdots & 0 & ? & ? & \cdots \\ \hline x_3^I \geq 0 & 0 & 0 & 1 & \cdots & 0 & ? & ? & \cdots \\ \hline \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \hline x_m^I \geq 0 & 0 & 0 & 0 & \cdots & 1 & ? & ? & \cdots \end{array}\right]$$

and we see that these rows are $>_L 0$.

Dual Simplex Method

♣ DSM is aimed at solving LO program in the standard form:

$$\mathrm{Opt}(P) = \max_x \left\{c^T x : Ax = b, x \geq 0\right\} \ [A : m \times n] \qquad (P)$$

under the same standing assumption that *the rows of A are linearly independent.*

♠ The problem dual to (P) reads

$$\mathrm{Opt}(D) = \min_{[\lambda_e;\lambda_g]} \left\{b^T \lambda_e : \lambda_g = c - A^T \lambda_e, \lambda_g \leq 0\right\} \qquad (D)$$

♠ Complementarity slackness reads

$$[\lambda_g]_i x_i = 0 \, \forall i.$$

♣ The PSM generates a sequence of neighboring *basic feasible solutions to* (P) (aka vertices of the primal feasible set) $x^1, x^2, ...$ augmented by *complementary infeasible solutions* $[\lambda_e^1; \lambda_g^1], [\lambda_e^2; \lambda_g^2], ...$ *to* (D) (these dual solutions satisfy, however, the equality constraints of (D)) until the infeasibility of the dual problem is discovered or until a feasible dual solution is built. In the latter case, we terminate with optimal solutions to both (P) and (D).

$$\mathrm{Opt}(P) = \max_x \left\{c^T x : Ax = b, x \geq 0\right\} \qquad (P)$$
$$[A : m \times n, \mathrm{Rank}\, A = m]$$

♠ The problem dual to (P) reads

$$\mathrm{Opt}(D) = \min_{[\lambda_e;\lambda_g]} \left\{b^T\lambda_e : \lambda_g = c - A^T\lambda_e, \lambda_g \leq 0\right\} \qquad (D)$$

♠ Complementarity slackness reads

$$[\lambda_g]_i x_i = 0 \,\forall i.$$

♠ The DSM generates a sequence of neighboring *basic feasible solutions to* (D) (aka vertices of the dual feasible set) $[\lambda_e^1; \lambda_g^1], [\lambda_e^2; \lambda_g^2], ...$ augmented by *complementary infeasible solutions* $x^1, x^2, ...$ *to* (P) (these primal solutions satisfy, however, the equality constraints of (P)) until the infeasibility of (P) is discovered or a feasible primal solution is built. In the latter case, we terminate with optimal solutions to both (P) and (D).

DSM: Preliminaries

$$\begin{array}{rcll} \mathrm{Opt}(P) & = & \max\limits_{x}\left\{c^Tx : Ax = b,\, x \geq 0\right\} & (P) \\ \mathrm{Opt}(D) & = & \min\limits_{[\lambda_e;\lambda_g]}\left\{b^T\lambda_e : \lambda_g + A^T\lambda_e = c, \lambda_g \leq 0\right\} & (D) \end{array}$$

♠ **Fact:** *Every basis I of matrix A induces*

— a uniquely defined by the basis primal basic solution x^I which solves primal equations, and has all nonbasic entries in x^I equal to zero;

— a uniquely defined by the basis dual basic solution $[\lambda_e^I, c^I]$ which solves dual equations, and has all basic entries in c^I equal to zero.

♠ **Fact:**

- *Extreme points of the primal feasible set are primal basic solutions x^I which are primal feasible: $x^I \geq 0$.*
- *Extreme points of the dual feasible set are dual basic solutions $[\lambda_e^I, c^I]$ which are dual feasible: $c^I \leq 0$.*

$$\mathrm{Opt}(D) = \min_{[\lambda_e;\lambda_g]} \left\{ b^T\lambda_e : \lambda_g + A^T\lambda_e = c, \lambda_g \leq 0 \right\} \qquad (D)$$

Observation: *Let x satisfy $Ax = b$. Replacing the dual objective $b^T\lambda_e$ with $-x^T\lambda_g$, we shift the dual objective on the dual feasible plane by a constant. Equivalently:*

$$[\lambda_e; \lambda_g], [\lambda'_e; \lambda'_g] \text{ satisfy the equality constraints in } (D)$$
$$\Leftrightarrow b^T[\lambda_e - \lambda'_e] = -x^T[\lambda_g - \lambda'_g]$$

Indeed,

$$\lambda_g + A^T\lambda_e = c = \lambda'_g + A^T\lambda'_e$$
$$\Rightarrow \lambda_g - \lambda'_g = A^T[\lambda'_e - \lambda_e]$$
$$\Rightarrow -x^T[\lambda_g - \lambda'_g] = -x^T A^T[\lambda'_e - \lambda_e] = b^T[\lambda_e - \lambda'_e].$$

$$\mathrm{Opt}(P) = \max_x \left\{c^T x : Ax = b, x \geq 0\right\} \ [A : m \times n] \qquad (P)$$
$$\mathrm{Opt}(D) = \min_{[\lambda_e;\lambda_g]} \left\{b^T\lambda_e : \lambda_g + A^T\lambda_e = c, \lambda_g \leq 0\right\} \qquad (D)$$

♣ **A step of the DSM:**

♠ At the beginning of a step, we have at our disposal a *basis* I of A along with corresponding *nonpositive vector* $c^I = c - A^T\overbrace{A_I^{-T}c_I}^{\lambda_e^I}$ of reduced costs and the *(not necessary feasible)* basic primal solution x^I. At a step we act as follows:

• We check whether $x^I \geq 0$. If it is the case, we terminate – x^I is an optimal primal solution, and $[\lambda_e^I = A_I^{-T}c_I; c^I]$ is the complementary optimal dual solution.

$$\mathrm{Opt}(D) = \min_{[\lambda_e;\lambda_g]} \left\{ b^T\lambda_e : \lambda_g + A^T\lambda_e = c, \lambda_g \le 0 \right\} \qquad (D)$$

• If x^I has negative entries, we pick one of them, let it be $x^I_{i_*}$: $x^I_{i_*} < 0$; note that $i_* \in I$. Variable x_{i_*} will leave the basis.

• We build a "ray of dual solutions"

$$[\lambda^I_e(t); c^I(t)] = [\lambda^I_e; c^I] - t[d_e; d_g],\ t \ge 0$$

from the requirements

$$\begin{aligned} c^I(t) + A^T\lambda^I_e(t) &= c, \\ c^I_{i_*}(t) &= -t, \\ c^I_i(t) &= c^I_i = 0 \text{ for all basic indexes } i \ne i_*. \end{aligned}$$

This results in

$$c^I(t) = c - A^T \overbrace{A_I^{-T}[c + te_{i_*}]_I}^{\lambda^I_e(t)} = c^I - tA^T A_I^{-T}[e_{i_*}]_I.$$

Observe that *along the ray $c^I(t)$, the dual objective strictly improves:*

$$b^T[\lambda^I_e(t) - \lambda^I_e] = -[x^I]^T[c^I(t) - c^I] = -x^I_{i_*}[-t] = \underbrace{x^I_{i_*}}_{<0} \cdot t.$$

♡ *It may happen that $c^I(t)$ remains nonpositive for all $t \ge 0$.* Then we have discovered a dual feasible ray along which the dual objective tends to $-\infty$. We terminate claiming that the dual problem is unbounded $\Rightarrow$ the primal problem is infeasible.

$$c^I(t) = c - A^T \overbrace{A_I^{-T}[c + te_{i_*}]_I}^{\lambda_e^I(t)} = c^I - tA^T A_I^{-T}[e_{i_*}]_I.$$

♡ *Alternatively, as t grows, some of the reduced costs $c_j^I(t)$ eventually become positive*. We identify the largest $t = \bar{t}$ such that $c^I(\bar{t}) \leq 0$ and index j such that $c_j^I(t)$ is about to become positive when $t = \bar{t}$; note that $j \notin I$. Variable x_j enters the basis. The new basis is $I^+ = [I \cup \{j\}]\backslash\{i_*\}$, the new vector of reduced costs is $c^I(\bar{t}) = c - A^T\lambda_e^I(\bar{t})$. We compute the new basic primal solution x^{I^+} and pass to the next step.

Note: As a result of a step, we
— either terminate with optimal primal and dual solutions,
— or terminate with correct claim that (P) is infeasible, (D) is unbounded,
— or pass to the new basis and new *feasible* basic dual solution and carry out a new step.
Along the basic dual solutions generated by the method, the values of dual objective never increase and strictly decrease at every step which does update the current dual solution (i.e., results in $\bar{t} > 0$). The latter definitely is the case when the current dual solution is *nondegenerate*, that is, all non-basic reduced costs are strictly negative.
In particular, *when all basic dual solutions are nondegenerate, the DSM terminates in finite time.*

♣ Same as PSM, *the DSM possesses Tableau implementation.* The structure of the tableau is exactly the same as in the case of PSM:

	x_1	x_2	$\cdots$	x_n
$-c^T x^I$	c_1^I	c_2^I	$\cdots$	c_n^I
$x_{i_1} =< \ldots >$	$[A_I^{-1}A]_{1,1}$	$[A_I^{-1}A]_{1,2}$	$\cdots$	$[A_I^{-1}A]_{1,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x_{i_n} =< \ldots >$	$[A_I^{-1}A]_{m,1}$	$[A_I^{-1}A]_{m,2}$	$\cdots$	$[A_I^{-1}A]_{m,n}$

We illustrate the DSM rules by example. The current tableau is

	x_1	x_2	x_3	x_4	x_5	x_6
8	-1	-1	-2	0	0	0
$x_4 = 20$	1	2	2	1	0	0
$x_5 = -20$	-2	-1	2	0	1	0
$x_6 = -1$	2	2	-1	0	0	1

which corresponds to $I = \{4, 5, 6\}$. The vector of reduced costs is nonpositive, and its basic entries are zero (a must for DSM).

A. Some of the entries in the basic primal solution (zeroth column) are negative. We select one of them, say, x_5, and call the corresponding row the *pivoting row*:

	x_1	x_2	x_3	x_4	x_5	x_6
8	-1	-1	-2	0	0	0
$x_4 = 20$	1	2	2	1	0	0
$x_5 = -20$	-2	-1	2	0	1	0
$x_6 = -1$	2	2	-1	0	0	1

Variable x_5 will leave the basis.

	x_1	x_2	x_3	x_4	x_5	x_6
8	-1	-1	-2	0	0	0
$x_4 = 20$	1	2	2	1	0	0
$x_5 = -20$	-2	-1	2	0	1	0
$x_6 = -1$	2	2	-1	0	0	1

B.1. *If all entries in the pivoting row, except for the one in zeroth column, are nonnegative, we terminate – the dual problem is unbounded, the primal is infeasible.*

B.2. *Alternatively, we*

— *select the negative entries in the pivoting row outside of the zeroth column* (all of them are nonbasic!) *and divide by them the corresponding entries in the zeroth row, thus getting nonnegative ratios*, in our example the ratios $10 = (-20)/(-2)$ and $20 = (-20)/(-1)$;

— *pick the smallest of the computed ratios and call the corresponding column the pivoting column. Variable which marks this column will enter the basis.*

	x_1	x_2	x_3	x_4	x_5	x_6
8	-1	-1	-2	0	0	0
$x_4 = 20$	1	2	2	1	0	0
$x_5 = -20$	-2	-1	2	0	1	0
$x_6 = -1$	2	2	-1	0	0	1

The pivoting column is the column of x_1. Variable x_5 leaves the basis, variable x_1 enters the basis.

	x_1	x_2	x_3	x_4	x_5	x_6
8	-1	-1	-2	0	0	0
$x_4 = 20$	1	2	2	1	0	0
$x_5 = -20$	-2	-1	2	0	1	0
$x_6 = -1$	2	2	-1	0	0	1

C. It remains to update the tableau, which is done *exactly as in the PSM*:
— we normalize the pivoting row by dividing its entries by the *pivoting element* (one in the intersection of the pivoting row and pivoting column) and change the label of this row (the new label is the variable which enters the basis):

	x_1	x_2	x_3	x_4	x_5	x_6
8	-1	-1	-2	0	0	0
$x_4 = 20$	1	2	2	1	0	0
$x_1 = 10$	1	0.5	-1	0	-0.5	0
$x_6 = -1$	2	2	-1	0	0	1

— we subtract from all non-pivoting rows multiples of the (normalized) pivoting row to zero the non-pivoting entries in the pivoting column:

	x_1	x_2	x_3	x_4	x_5	x_6
18	0	-0.5	-3	0	-0.5	0
$x_4 = 10$	0	1.5	3	1	0.5	0
$x_1 = 10$	1	0.5	-1	0	-0.5	0
$x_6 = -21$	0	1	1	0	1	1

The step of DSM is over.

	x_1	x_2	x_3	x_4	x_5	x_6
18	0	−0.5	−3	0	−0.5	0
$x_4 = 10$	0	1.5	3	1	0.5	0
$x_1 = 10$	1	0.5	−1	0	−0.5	0
$x_6 = -21$	0	1	1	0	1	1

The basis is $I = \{1, 4, 6\}$, and there still is a negative basic variable x_6. Its row is the pivoting one:

	x_1	x_2	x_3	x_4	x_5	x_6
18	0	−0.5	−3	0	−0.5	0
$x_4 = 10$	0	1.5	3	1	0.5	0
$x_1 = 10$	1	0.5	−1	0	−0.5	0
$x_6 = -21$	0	1	1	0	1	1

However: *all entries in the pivoting row, except for the one in the zeroth column, are nonnegative*

$\Rightarrow$ *Dual problem is unbounded, primal problem is infeasible.*

	x_1	x_2	x_3	x_4	x_5	x_6
18	0	−0.5	−3	0	−0.5	0
$x_4 = 10$	0	1.5	3	1	0.5	0
$x_1 = 10$	1	0.5	−1	0	−0.5	0
$x_6 = -21$	0	1	1	0	1	1

Explanation. In terms of the tableau, the vector d_g participating in the description of the "ray of dual solutions"

$$[\lambda_e^I(t); c^I(t)] = [\lambda_e^I; c^I] - t[d_e; d_g]$$

is just the (transpose of) the pivoting row (with entry in the zeroth column excluded). When this vector is nonnegative, we have $c^I(t) \le 0$ for all $t \ge 0$, that is, the *entire* ray of dual solutions is feasible. It remains to recall that along this ray the dual objective tends to $-\infty$.

Warm Start

♣ **Fact:** In many applications there is a necessity to solve a *sequence of "close" to each other* LO programs.

Example: In branch-and-bound methods for solving Mixed Integer Linear Optimization problems, one solves a sequence of LOs obtained from each other by adding/deleting variable/constraint, one at a time.

♣ **Question:** Assume we have solved an LO

$$\max_x \left\{ \bar{c}^T x : \bar{A}x = \bar{b}, x \geq 0 \right\} \qquad (\bar{P})$$

to optimality by PSM (or DSM) and have at our disposal the optimal basis I giving rise to optimal primal solution $\bar{x}^I$ and optimal dual solution $[\bar{\lambda}_e^I; \bar{c}^I]$:

$$\begin{array}{ll} [\bar{x}^I]_I = \bar{A}_I^{-1}\bar{b} \geq 0 & \text{[feasibility]} \\ \bar{c}^I = \bar{c} - \bar{A}^T \underbrace{[\bar{A}_I]^{-T}\bar{c}_I}_{\bar{\lambda}_e^I} \leq 0 & \text{[optimality]} \end{array}$$

How could we utilize I when solving an LO problem (P) "close" to $(\bar{P})$?

$$\max_x \left\{\bar{c}^T x : \bar{A}x = \bar{b}, x \geq 0\right\} \qquad (\bar{P})$$

Cost vector is updated: $\bar{c} \mapsto c$

♠ For the new problem, I remains a basis, and $\bar{x}^I$ remains a basic feasible solution. We can easily compute the basic dual solution $[\lambda_e; c^I]$: $c^I = c - \bar{A}^T \underbrace{[\bar{A}_I]^{-T} c_I}_{\lambda_e}$

• If nonbasic entries in c^I are nonpositive, $\bar{x}^I$ and $[\lambda_e; c^I]$ are optimal primal and dual solutions to the new problem.

Note: This definitely is the case when nonbasic entries in $\bar{c}^I$ are negative (i.e., $[\bar{\lambda}_e; \bar{c}^I]$ is *nondegenerate* dual basic solution to $(\bar{P})$) and c is close enough to $\bar{c}$. In this case,

$$\bar{x}^I = \nabla_c\Big|_{c=\bar{c}} \mathrm{Opt}(c).$$

• If some of the nonbasic entries in c^I are positive, I is non-optimal for (P), but I still is a basis producing a basic feasible primal solution for the new problem. *We can solve the new problem by PSM starting from this basis*, which usually allows to solve (P) much faster than "from scratch."

$$\max_x \left\{\bar{c}^T x : \bar{A}x = \bar{b}, x \geq 0\right\} \qquad (\bar{P})$$

Right hand side vector is updated: $\bar{b} \mapsto b$

♠ For the new problem, I remains a basis, and $[\bar{\lambda}_e; \bar{c}^I]$ remains a basic dual feasible solution. We can easily compute the new primal basic solution:

$$[x^I]_I = [\bar{A}_I]^{-1} b, \; [x^I]_i = 0 \text{ when } i \notin I$$

• If basic entries in x^I are nonnegative, x^I and $[\bar{\lambda}_e; \bar{c}^I]$ are optimal primal and dual solution to the new problem.

Note: This definitely is the case when basic entries in $\bar{x}^I$ are positive (i.e., $\bar{x}^I$ is *nondegenerate* primal basic solution to $(\bar{P})$) and b is close enough to $\bar{b}$. In this case,

$$\bar{\lambda}_e = \nabla_b \big|_{b=\bar{b}} \mathrm{Opt}(b).$$

• If some of the basic entries in x^I are negative, I is non-optimal for (P), but I still is a basis producing a dual feasible solution to the new problem. *We can solve the new problem by DSM starting from this basis*, which usually allows to solve (P) much faster than "from scratch."

$$\max_x \left\{ \bar{c}^T x : \bar{A}x = \bar{b}, x \geq 0 \right\} \qquad (\bar{P})$$

Some nonbasic columns in $\bar{A}$ are updated
and/or
new variables are added

♣ Assume that we update some nonbasic columns in $\bar{A}$ and extend the list of primal variables, adding to $\bar{A}$ several columns and extending c accordingly.

♠ For the new problem, I remains a basis, and $\bar{x}^I$, extended by appropriate number of zero entries, remains a basic primal feasible solution. *We can solve the new problem by PSM, starting with these basis and basic feasible solution.*

$$\max_x \left\{ \bar{c}^T x : \bar{A}x = \bar{b}, x \geq 0 \right\} \qquad (\bar{P})$$

A new equality constraint is added to $(\bar{P})$

♣ Assume that $m \times n$ matrix $\bar{A}$ with linearly independent rows is augmented by a new row a^T (linearly independent of the old ones), and $\bar{b}$ is augmented by a new entry.

♠ Augmenting $\bar{\lambda}_e$ by zero entry, we get a feasible dual solution $[\lambda_e; \bar{c}^I]$ to the new problem. This solution, however, is *not necessarily basic*, since I is *not* a basis for the constraint matrix A of the new problem.

♠ **Fact:** $[\lambda_e; \bar{c}^I]$ *can be easily converted into a basic dual feasible solution, and we can use this solution in DSM.*

$$\max_x \left\{\bar{c}^T x : \bar{A}x = \bar{b}, x \geq 0\right\} \qquad (\bar{P})$$

♣ Assume that $m \times n$ matrix $\bar{A}$ with linearly independent rows is augmented by a new row a^T (linearly independent of the old ones), and $\bar{b}$ is augmented by a new entry.

♠ Augmenting $\bar{\lambda}_e$ by zero entry, we get a feasible dual solution $[\lambda_e; \bar{c}^I]$ to the new problem. This solution, however, is *not necessarily basic*, since I is *not* a basis for the constraint matrix A of the new problem.

♠ **Fact:** $[\lambda_e; \bar{c}^I]$ *can be easily converted into a basic dual feasible solution, and we can use this solution in DSM.*

♡ Let $\Delta = a - \bar{A}^T[\bar{A}_I]^{-T} a_I = A^T\delta$.

Note: $\delta \neq 0$ and entries Δ_i, $i \in I$, are zero.

• Since the rows in A are independent and $\delta \neq 0$, we have $\Delta \neq 0$; replacing, if necessary, δ with $-\delta$, we can assume that one of the entries in Δ is negative.

Thus, $\Delta = A^T\delta \neq 0$, $\Delta_i = 0$ for $i \in I$ and $\Delta_j < 0$ for some $j \notin I$.

Situation: $\Delta = A^T\delta \neq 0$, $\Delta_i = 0$ for $i \in I$ and $\Delta_j < 0$ for some $j \notin I$.

Now let us look at the ray

$$\{[\lambda_e(t) := \lambda_e - t\delta, c^I(t) := c - A^T\lambda_e(t) = \bar{c}^I - t\Delta] : t \geq 0\}$$

so that

- $A^T\lambda_e(t) + c^I(t) \equiv c$
- all $[c^I(t)]_i$, $i \in I$, are identically zero
- $c^I(0) = \bar{c}^I \leq 0$
- entry $[c^I(t)]_j$ is positive when t is large

Let $\bar{t}$ be the largest $t \geq 0$ such that $c^I(t) \leq 0$. When $t = \bar{t}$, for some $\bar{j}$ the entry $[c^I(\bar{t})]_{\bar{j}}$ is about to become positive. Note that $\bar{j} \notin I$ since $[c^I(t)]_i \equiv 0$ for $i \in I$. *It is immediately seen that* $I^+ = I \cup \{\bar{j}\}$ *and* $[\lambda_e(\bar{t}); c^I(\bar{t})]$ *are a basis and the corresponding basic feasible dual solution for the new problem. We can now solve the new problem by DSP starting with this solution.*

$$\max_x \left\{ \bar{c}^T x : \bar{A}x = \bar{b}, x \geq 0 \right\} \qquad (\bar{P})$$

An inequality constraint $a^T x \leq \alpha$ is added to $(\bar{P})$

♣ The new problem *in the standard form* reads

$$\min_{x,s} \left\{ [\bar{c}; 0]^T [x; s] : \begin{array}{lll} \bar{A}x & & = \bar{b} \\ a^T x & +s & = \alpha \end{array}, x \geq 0, s \geq 0 \right\} \qquad (P)$$

♠ Transition from $(\bar{P})$ to (P) can be done in two steps:

♣ We augment x by new variable s, $\bar{A}$ by a zero column, and $\bar{c}$ - by zero entry, thus arriving at the problem

$$\min_{x,s} \left\{ [\bar{c}; 0]^T [\bar{x}; s] : \bar{A}x + 0 \cdot s = \bar{b}, x \geq 0, s \geq 0 \right\} \qquad (\widetilde{P})$$

For this problem, I is a basis, $\widetilde{x}^I = [x^I; 0]$ is the associated primal optimal basic feasible solution, and $[\bar{\lambda}_e; [\bar{c}^I; 0]]$ is the associated dual optimal basic solution.

• (P) is obtained from $(\widetilde{P})$ by adding a new equality constraint. *We already know how to warm-start solving (P) by DSM.*

Lecture II.2
Network Simplex Method

♣ Recall the (single-product) *Network Flow* problem:

Given

- *an oriented graph with arcs assigned transportation costs and capacities,*
- *a vector of external supplies at the nodes of the arc,*

find the cheapest flow fitting the supply and respecting arc capacities.

♣ **Network Flow problems** form an important special case in LO.

♠ As applied to Network Flow problems, the Simplex Method admits a dedicated implementation which has many advantages as compared to the general-purpose algorithm.

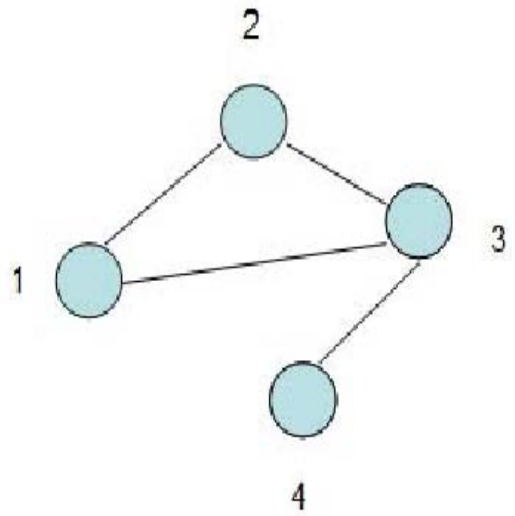
Preliminaries: Undirected Graphs

♣ **Undirected graph** is a pair $G = (\mathcal{N}, \mathcal{E})$ composed of

- finite nonempty *set of nodes* $\mathcal{N}$
- *set* $\mathcal{E}$ *of arcs, an arc being a 2-element subset of* $\mathcal{N}$*.*

Standard notation: we identify $\mathcal{N}$ with the set $\{1, 2, ..., m\}$, m being a number of nodes in the graph in question. Then every arc γ becomes an *unordered* pair $\{i, j\}$ of two *distinct from each other* ($i \neq j$) nodes.

We say that nodes i, j are *incident* to arc $\gamma = \{i, j\}$, and this arc *links* nodes i and j.



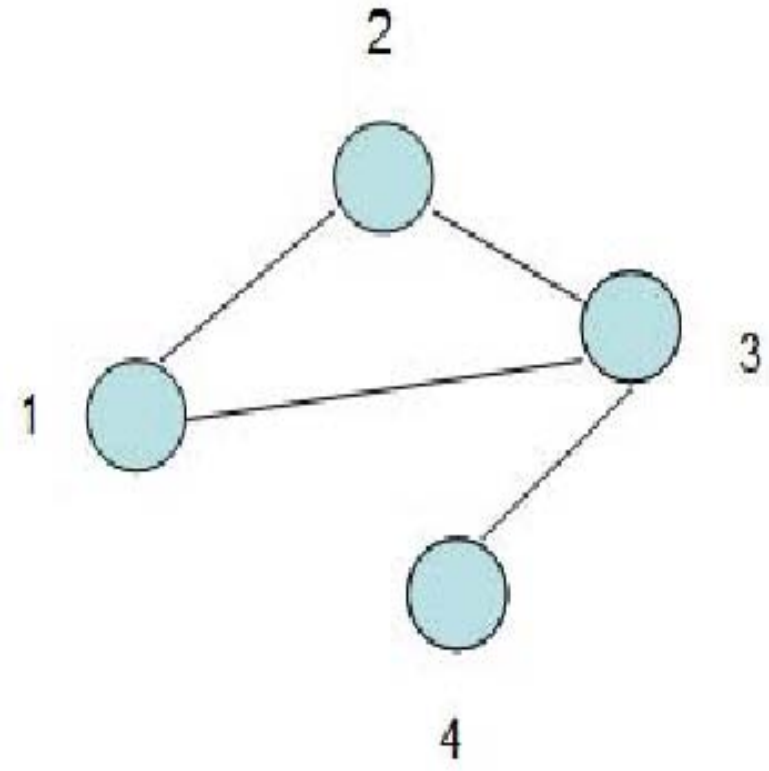
Undirected graph.
Nodes: $\mathcal{N} = \{1, 2, 3, 4\}$.
Arcs: $\mathcal{E} = \{\{1, 2\}, \{2, 3\}, \{1, 3\}, \{3, 4\}\}$

♣ **Walk:** an ordered sequence of nodes $i_1, ..., i_t$ with every two consecutive nodes linked by an arc: $\{i_s, i_{s+1}\} \in \mathcal{E}$, $1 \leq s < t$.

- $1, 2, 3, 1, 3, 4, 3$ is a walk.
- $1, 2, 4, 1$ is not a walk.

♠ **Connected graph:** there exists a walk which passes through all nodes.

- Graph on the picture is connected.



♣ **Path:** a walk with all nodes distinct from each other

• $1, 2, 3, 4$ is a path • $1, 2, 3, 1$ is not a path

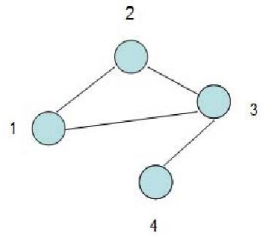
♣ **Cycle:** a walk $i_1, i_2, i_3, ..., i_t = i_1$ with the same first and last nodes, all nodes $i_1, ..., i_{t-1}$ distinct from each other *and* $t \geq 4$ (i.e., at least 3 distinct nodes).

• $1, 2, 3, 1$ is a cycle • $1, 2, 1$ is not a cycle

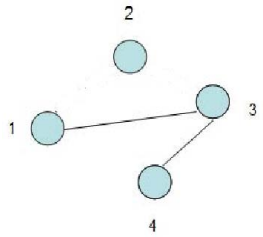
♣ **Leaf:** a node which is incident to exactly one arc

• Node 4 is a leaf, all other nodes are not leaves

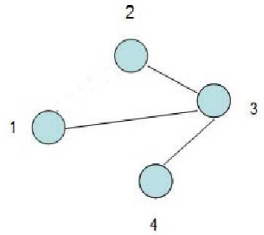
A:



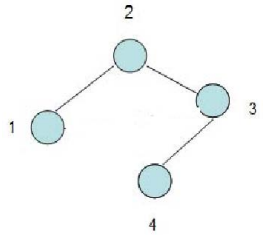
B:



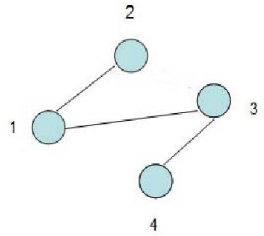
C:



D:



E:



♣ **Tree:** a connected graph without cycles

- **A** - non-tree (there is a cycle)
- **B** – non-tree (not connected)
- **C, D, E:** trees

♣ **Tree:** a connected graph without cycles

♠ Observation I: *Every nontrivial (with more than one node) tree has a leaf*

• Let a graph with $m > 1$ nodes be connected and without leaves. Then every node is incident to at least two arcs. Let us walk along the graph in such a way that when leaving a node, we do *not* use an arc which was used when entering the node (such a choice of the "exit" arc always is possible, since every node is incident to at least two arcs). Eventually certain node i will be visited for the second time. When this happens for the first time, the segment of our walk "from the first visit to i till the second visit to i" will be a cycle. Thus, *every connected graph with ≥ 2 nodes and without leaves has a cycle* and thus cannot be a tree. □

♠ **Observation II:** *A connected m-node graph is a tree iff it has $m-1$ arcs.*

Proof, I: *Let G be an m-node tree*, and let us prove that G has exactly $m-1$ arcs.

Verification is by induction in m. The case of $m=1$ is trivial. Assume that every m-node tree has exactly $m-1$ arcs, and let G be a tree with $m+1$ nodes. By Observation I, G has a leaf node; eliminating from G this node along with the unique arc incident to this node, we get an m-node graph G_- which is connected and has no cycles (since G is so). By Inductive Hypothesis, G_- has $m-1$ arcs, whence G has $m=(m+1)-1$ arcs. □

Proof, II: *Let G be a connected m-node graph with $m-1$ arcs*, and let us prove that G is a tree, that is, that G does not have cycles.

Assume that it is not the case. Take a cycle in G and eliminate from G one of the arcs on the cycle; note that the new graph G' is connected along with G. If G' still have a cycle, eliminate from G' an arc on a cycle, thus getting connected graph G'', and so on. Eventually a connected *cycle-free* m-node graph (and thus a tree) will be obtained; by part I, this graph has $m-1$ arcs, same as G. But our process reduces the number of arcs by 1 at every step, and we arrive at the conclusion that no steps of the process were in fact carried out, that is, G has no cycles. □

♠ **Observation III:** *Let G be a graph with m nodes and $m-1$ arcs and without cycles. Then G is a tree.*

Indeed, we should prove that G is connected. Assuming that this is not the case, we can split G into $k \geq 2$ *connected components*, that is, split the set $\mathcal{N}$ of nodes into k nonempty and non-overlapping subsets $\mathcal{N}_1, ..., \mathcal{N}_k$ such that no arc in G links two nodes belonging to different subsets. The graphs G_i obtained from G by reducing the nodal set to $\mathcal{N}_i$, and the set of arcs – to the set of those of the original arcs which link nodes from $\mathcal{N}_i$, are connected graphs without cycles and thus are trees. It follows that G_i has $\mathrm{Card}(\mathcal{N}_i) - 1$ arcs, and the total number of arcs in all G_i is $m - k < m - 1$. But this total is exactly the number $m - 1$ of arcs in G, and we get a contradiction. □

♠ **Bottom Line:** *Let G be a m-node graph. Consider the following properties of G*

- *G is connected*
- *G has no cycles*
- *G has exactly $m - 1$ arcs.*

Every two of these properties imply the third one, and all these properties together mean that G is a tree.

♠ **Observation IV:** *Let G be a tree, and i, j be two distinct nodes in G. Then there exists exactly one path which starts at i and ends at j.*

Indeed, since G is connected, there exists a walk which starts at i and ends at j; the walk of this type with minimum possible number of arcs clearly is a path.

Now let us prove that the path π which links i and j is unique. Indeed, assuming that there exists another path π' which starts at i and ends at j, we get a 'loop" – a walk which starts and ends at i and is obtained by moving first from i to j along π and then moving *backward* along π'. If $\pi' \neq \pi$, this loop clearly contains a cycle. Thus, G has a cycle, which is impossible since G is a tree. □

♠ **Observation V:** *Let G be a tree, and let we add to G a new arc $\gamma_* = \{i_*, j_*\}$. In the resulting graph, there will be exactly one cycle, and the added arc will be on this cycle.*

Note: When counting the number of different cycles in a graph, we do not distinguish between cycles obtained from each other by shifting the starting node along the cycle and/or reversing order of nodes. Thus, cycles a, b, c, d, a, b, c, d, a, b and a, d, c, b, a are counted as a *single* cycle.

Proof: When adding a new arc to a tree with m nodes, we get a connected graph with m nodes and m arcs. Such a graph cannot be a tree and therefore it has a cycle C. One of the arcs in C should be γ_* (otherwise C would be a cycle in G itself, while G is a tree and does not have cycles). The "outer" (aside of γ_*) part of C should be a path in G which links the nodes j_* and i_*, and by Observation IV such a path is unique. Thus, C is unique as well. □

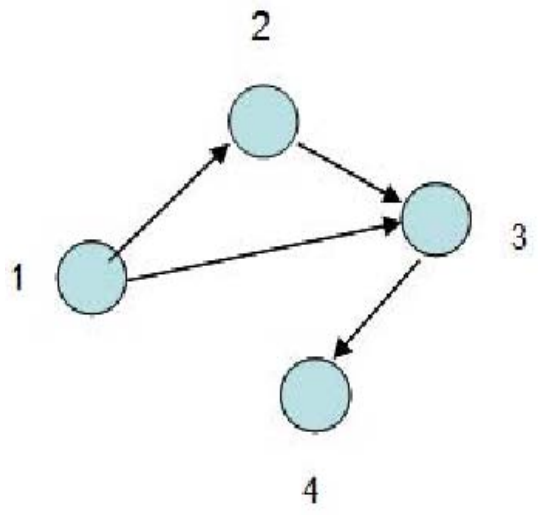
Oriented Graphs

♣ **Oriented graph** $G = (\mathcal{N}, \mathcal{E})$ is a pair composed of

- finite nonempty set of nodes $\mathcal{N}$
- *set $\mathcal{E}$ of arcs – ordered pairs of distinct nodes*

Thus, an arc γ of G is an ordered pair (i, j) composed of two distinct $(i \neq j)$ nodes. We say that arc $\gamma = (i, j)$

- *starts at node i*,
- *ends at node j*, and
- *links* nodes i and j.

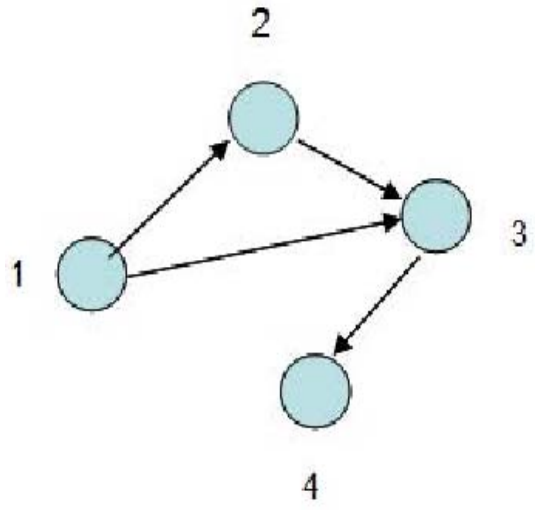


Directed graph.
Nodes: $\mathcal{N} = \{1, 2, 3, 4\}$.
Arcs: $\mathcal{E} = \{(1, 2), (2, 3), (1, 3), (3, 4)\}$

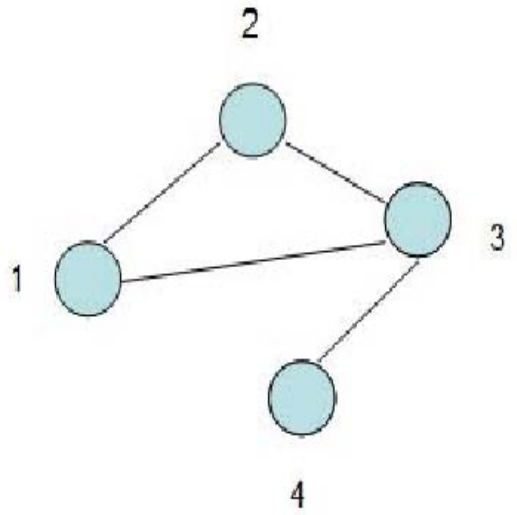
♠ Given a directed graph G and ignoring the directions of arcs (i.e., converting a directed arc (i, j) into undirected arc $\{i, j\}$), we get an undirected graph $\widehat{G}$.

Note: If G contains "inverse to each other" directed arcs (i, j), (j, i), these arcs yield a *single* arc $\{i, j\}$ in $\widehat{G}$.

♠ A directed graph G is called *connected*, if its undirected counterpart is connected.



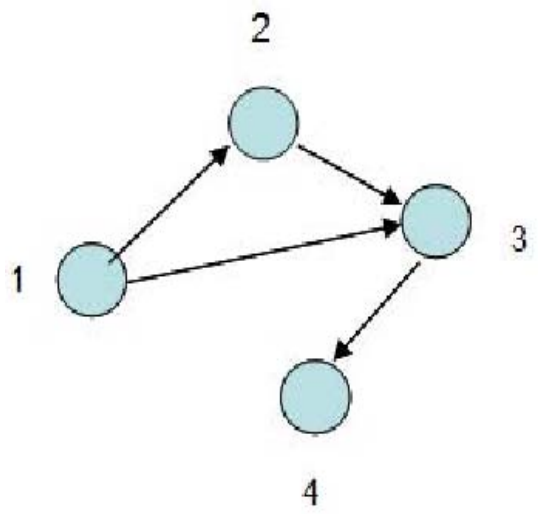
Directed graph G



Undirected counterpart $\widehat{G}$ of G

♠ **Incidence matrix** P of a directed graph $G = (\mathcal{N}, \mathcal{E})$:

- the rows are indexed by nodes $1, ..., m$
- the columns are indexed by arcs γ
- $P_{i\gamma} = \begin{cases} 1, & \gamma \text{ starts at } i \\ -1, & \gamma \text{ ends at } i \\ 0, & \text{all other cases} \end{cases}$



$$P = \left[\begin{array}{r|r|r|r} 1 & 0 & 1 & 0 \\ \hline -1 & 1 & 0 & 0 \\ \hline 0 & -1 & -1 & 1 \\ \hline 0 & 0 & 0 & -1 \end{array}\right]$$

Network Flow Problem

♣ **Network Flow Problem:** *Given*

- *an oriented graph* $G = (\mathcal{N}, \mathcal{E})$
- *costs* c_γ *of transporting a unit flow along arcs* $\gamma \in \mathcal{E}$,
- *capacities* $u_\gamma > 0$ *of arcs* $\gamma \in \mathcal{E}$ *– upper bounds on flows in the arcs,*
- *a vector* s *of external supplies at the nodes of* G,

find a flow $f = \{f_\gamma\}_{\gamma \in \mathcal{E}}$ *with as small cost* $\sum_\gamma c_\gamma f_\gamma$ *as it is possible under the restrictions that*

- *the flow* f *respects arc directions and capacities:* $0 \leq f_\gamma \leq u_\gamma$ *for all* $\gamma \in \mathcal{E}$
- *the flow* f *obeys the conservation law w.r.t. the vector of supplies* s*:*

  at every node i*, the total incoming flow* $\sum_{j:(j,i)\in\mathcal{E}} f_{ji}$ *plus the external supply* s_i *at the node is equal to the total outgoing flow* $\sum_{j:(i,j)\in\mathcal{E}} f_{ij}$.

♠ **Observation:** *The flow conservation law can be written as*

$$Pf = s$$

where P *is the incidence matrix of* G.

♠ Notation: From now on, m stands for the number of nodes, and n stands for the number of arcs in G.

♠ **The Network Flow problem reads:**

$$\min_f \left\{ \sum_{\gamma \in \mathcal{E}} c_\gamma f_\gamma : Pf = s,\, 0 \le f \le u \right\}. \qquad \text{(NWIni)}$$

The Network Simplex Method is a specialization of the Primal Simplex Method aimed at solving the Network Flow Problem.

♣ **Observation:** The column sums in P are equal to 0, that is, $[1; 1; ...; 1]^T P = 0$, whence $[1; ...; 1]^T Pf = 0$ for every f

⇒ *The rows in P are linearly dependent, and* (NWIni) *is infeasible unless* $\sum_{i=1}^m s_i = 0$.

♠ **Observation:** When $\sum_{i=1}^m s_i = 0$ (which is necessary for (NWIni) to be solvable), the last equality constraint in (NWIni) is minus the sum of the first $m-1$ equality constraints

⇒ (NWIni) *is equivalent to the LO program*

$$\min_f \left\{ \sum_{\gamma \in \mathcal{E}} c_\gamma f_\gamma : Af = b,\, 0 \le f \le u \right\}, \qquad \text{(NW)}$$

where A is the $(m-1) \times n$ matrix composed of the first $m-1$ rows of P, and $b = [s_1; s_2; ...; s_{m-1}]$.

♣ **Standing assumptions:**

♡ The total external supply is 0: $\sum_{i=1}^{m} s_i = 0$

♡ The graph G is connected

Under these assumptions,

- The Network Flow problem reduces to

$$\min_f \left\{ \sum_{\gamma \in \mathcal{E}} c_\gamma f_\gamma : Af = b,\ 0 \le f \le u \right\}. \qquad \text{(NW)}$$

- The rows of A are linearly independent (to be shown later), which makes (NW) well suited for Simplex-type algorithms.

♣ We start with the *uncapacitated* case where all capacities are $+\infty$, that is, the problem of interest is

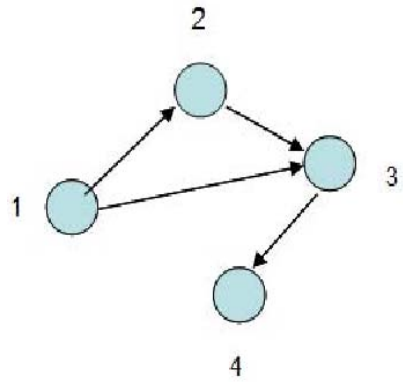
$$\min_f \left\{ \sum_{\gamma \in \mathcal{E}} c_\gamma f_\gamma : Af = b,\ f \ge 0 \right\}. \qquad \text{(NWU)}$$

$$\min_f \left\{ \sum_{\gamma \in \mathcal{E}} c_\gamma f_\gamma : Af = b,\ f \geq 0 \right\}. \qquad \text{(NWU)}$$

♣ Our first step is to understand *what are bases of* A.

♣ **Spanning trees.** Let I be a set of exactly $m-1$ arcs of G. Consider the *undirected* graph G_I with the same m nodes as G, and with arcs obtained from arcs $\gamma \in I$ by ignoring their directions. Thus, every arc in G_I is of the form $\{i,j\}$, where (i,j) or (j,i) is an arc from I.
Note that the inverse to each other arcs (i,j), (j,i), if present in I, induce a single arc $\{i,j\} = \{j,i\}$ in G_I.

♠ It may happen that G_I is a tree. In this case I is called a *spanning tree* of G; we shall express this situation by words *the* $m-1$ *arcs from* I *form a tree when their directions are ignored.*



- $I = \{(1,2),(2,3),(3,4)\}$ is a spanning tree
- $I = \{(2,3),(1,3),(3,4)\}$ is a spanning tree
- $I = \{(1,2),(2,3),(1,3)\}$ is not a spanning tree (G_I has cycles and is not connected)
- $I = \{(1,2),(2,3)\}$ is not a tree (less than $4-1=3$ arcs)

$$\min_f \left\{ \sum_{\gamma \in \mathcal{E}} c_\gamma f_\gamma : Af = b,\ f \geq 0 \right\}. \qquad \text{(NWU)}$$

♣ We are about to prove that *the bases of A are exactly the spanning trees of G.*

Observation 0: If I is a spanning tree, then I does not include pairs of opposite to each other arcs, and thus there is a one-to-one correspondence between arcs in I and induced arcs in G_I.

Indeed, I has $m-1$ arcs, and G_I is a tree and thus has $m-1$ arcs as well; the latter would be impossible if two arcs in I were opposite to each other and thus would induce a single arc in G_I. □

Observation I: Every set I' of arcs of G which does not contain inverse to each other arcs and is such that arcs of I' do not form cycles after their directions are ignored can be extended to a spanning tree I. *In particular, spanning trees do exist.*

Proof. It may happen (case A) that the set $\mathcal{E}$ of all arcs of G does not contain inverse arcs and arcs from $\mathcal{E}$ do not form cycles when their directions are ignored. Since G is connected, the undirected counterpart of G is a connected graph with no cycles, i.e., it has exactly $m-1$ arcs. But then $\mathcal{E}$ has $m-1$ arcs and is a spanning tree, and we are done.

"It may happen (case A) that the set $\mathcal{E}$ of all arcs of G does not contain inverse arcs and arcs from $\mathcal{E}$ do not form cycles when their directions are ignored. Since G is connected, the undirected counterpart of G is a connected graph with no cycles, i.e., it has exactly $m-1$ arcs. But then $\mathcal{E}$ has $m-1$ arcs and is a spanning tree, and we are done."

♡ If A is not the case, then either $\mathcal{E}$ has a pair of inverse arcs (i,j), (j,i) (case B.1), or $\mathcal{E}$ does not contain inverse arcs and has a cycle (case B.2).

• *In the case of B.1* one of the arcs (i,j), (j,i) is not in I' (since I' does not contain inverse arcs). Eliminating this arc from G, we get a *connected* graph G^1, and arcs from I' are among the arcs of G^1.

• *In the case of B.2* G has a cycle; this cycle includes an arc $\gamma \notin I'$ (since arcs from I' make no cycles). Eliminating this arc from G, we get a *connected* graph G^1, and arcs from I' are among the arcs of G^1.

♡ Assuming that A is not the case, we build G^1 and apply to this connected graph the same construction as to G. As a result, we

— either conclude that G^1 is the desired spanning tree,

— or eliminate from G^1 an arc, thus getting *connected* graph G^2 such that all arcs of I' are arcs of G^2.

♡ Proceeding in this way, we must eventually terminate; upon termination, the set of arcs of the current graph is the desired spanning tree containing I'. □

Note: Columns of A are indexed by arcs

$\Rightarrow$ *Bases of A are some collections I of $m-1$ arcs of G.*

♣ **Facts:**

A. *Bases of A are exactly the collections I of $m-1$ arcs from $\mathcal{E}$ which become spanning trees after their directions are ignored.*

B. *If I is a base of A, then the $(m-1)\times(m-1)$ submatrix A_I of A after permuting rows and columns becomes lower-triangular with all diagonal entries equal to ± 1.*

$$\min_f \left\{ \sum_{\gamma\in\mathcal{E}} c_\gamma f_\gamma : Af = b,\ f \geq 0 \right\}. \quad \text{(NWU)}$$

♣ **Corollary [Integrality of Network Flow Polytope]:** *Let b be integral. Then all basic solutions to* (NWU)*, feasible or not, are integral vectors.*

Indeed, the basic entries in a basic solution solve the system $Bu = b$ with integral right hand side and lower triangular nonsingular matrix B with integral entries and diagonal entries ± 1

$\Rightarrow$ all entries in u are integral. $\square$

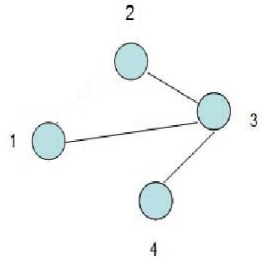
Proof of Facts

Claim I: *Let I be a spanning tree. Then the columns of A with indexes from I are linearly independent, that is, I is a basis of A.*

Proof. • Consider the graph G_I. This is a tree; therefore, for every node i there is a unique path in G_I leading from i to the *root node* m. *We can renumber the nodes j in such a way that the new indexes j' of nodes strictly increase along such a path. Note that the new number of the root node still is m: $m' = m$.*

Indeed, we can associate with a node i its "distance" from the root node, defined as the number of arcs in the unique path from i to m, and then reorder the nodes, making the distances nonincreasing in the new serial numbers of the nodes.

♡ For example, with G_I as shown:



we could set $2' = 1, 1' = 2, 3' = 3, 4' = 4$

Proofs of Facts, continued

• Now let us associate with an arc $\gamma = (i, j) \in I$ its *serial number* $p(\gamma) = \min[i', j']$. Observe that *different arcs from I get different serial numbers*. Indeed, assuming w.l.o.g. that $i' < j'$, the walk in G_I

"from i to j along γ and then from j to the root node m
along the unique path π in G_I leading from j to m"

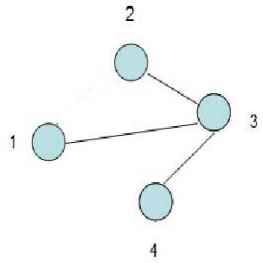
is a path in G_I, since $'$-indexes of the nodes in π are $\geq j'$ and thus differ from i'. If now two distinct from each other arcs from I, $\gamma = (i, j)$ $\bar{\gamma} = (\bar{i}, \bar{j})$ have equal serial numbers:

$$\min[i', j'] = \min[\bar{i}', \bar{j}'] = s',$$

then there is a path in G_I from s to the root node which starts with the arc γ, same as there is a path in G_i from s to the root node which starts with $\bar{\gamma} \neq \gamma$, which is impossible, since G_I is a tree.

Since the serial numbers take values $1, ..., m-1$ and the serial numbers of different arcs from I are different, serial numbers indeed form an ordering of the $m-1$ arcs from I.

♡ In our example, where G_I is the graph



and $2' = 1, 1' = 2, 3' = 3, 4' = 4$ we get

$$p(2, 3) = \min[2'; 3'] = 1,$$
$$p(1, 3) = \min[1', 3'] = 2,$$
$$p(3, 4) = \min[3', 4'] = 3$$

Proofs of Facts, continued

• Now let A_I be the $(m-1) \times (m-1)$ submatrix of A composed of columns with indexes from I. Let us reorder the columns according to the serial numbers of the arcs $\gamma \in I$, and rows – according to the new indexes i' of the nodes. The resulting matrix B_I is or is not singular simultaneously with A_I.
Observe that in B, same as in A_I, every column
— either has two nonzero entries (case A), one equal to 1 and one equal to -1 [this is the case when the arc γ indexing the column is *not* incident to the root node m],
— or has exactly one nonzero entry (case B), equal to +1 or to -1 [this is the case when the root node m is incident to the arc γ indexing the column]

• in case A, the index ν of column is the minimum of row indexes of the two nonzero entries in the column.

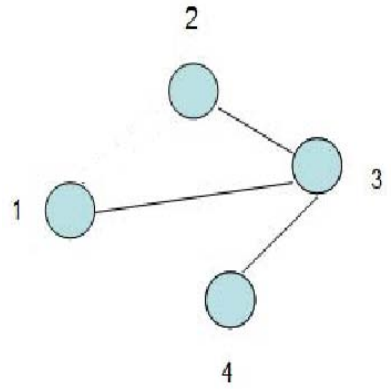
• in case B, the index ν of the column is the row index of the only nonzero entry in the column.

$\Rightarrow$ In both cases, ν is the minimum of row indexes of the nonzero entries in the column.

In other words, B is a lower triangular matrix with diagonal entries ± 1 and therefore it is nonsingular. $\square$

Proofs of Facts, continued

♡ In our example $I = \{(2,3),(1,3),(3,4)\}$, G_I is the graph



we have $1' = 2, 2' = 1, 3' = 3, 4' = 4$, $p(2,3) = 1$, $p(1,3) = 2$, $p(3,4) = 3$ and

$$A = \left[\begin{array}{r|r|r|r} 1 & 0 & 1 & 0 \\ \hline -1 & 1 & 0 & 0 \\ \hline 0 & -1 & -1 & 1 \end{array}\right]$$

arcs indexing columns, from left to right: (1,2),(2,3),(1,3),(3,4)

so that

$$A_I = \left[\begin{array}{r|r|r} 0 & 1 & 0 \\ \hline 1 & 0 & 0 \\ \hline -1 & -1 & 1 \end{array}\right], \quad B = \left[\begin{array}{r|r|r} 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline -1 & -1 & 1 \end{array}\right]$$

As it should be B is lower triangular with diagonal entries ± 1.

Proofs of Facts, continued

Corollary: *The $m-1$ rows of A are linearly independent.*

Indeed, by Observation I, G has a spanning tree I, and by Claim I, the corresponding $(m-1)\times(m-1)$ submatrix A_I of A is nonsingular.

♣ We have seen that spanning trees are the bases of A. The inverse is also true:

Claim II: *Let I be a basis of A. Then I is a spanning tree.*

Proof. A basis should be a set of $m-1$ indexes of columns in A — i.e., a set I of $m-1$ arcs — such that the columns A_γ, $\gamma\in I$, of A are linearly independent.

• Observe that I cannot contain inverse arcs (i,j), (j,i), since the sum of the corresponding columns in P (and thus in A) is zero.

Proofs of Facts, continued

• Consequently G_I has exactly $m-1$ arcs. We want to prove that the m-node graph G_I is a tree, and to this end it suffices to prove that G_I has no cycles. Let, on the contrary, $i_1, i_2, ..., i_t = i_1$ be a cycle in G_I ($t > 3$, $i_1, ..., i_{t-1}$ are distinct from each other). Consequently, I contains $t-1$ distinct arcs $\gamma_1, ..., \gamma_{t-1}$ such that for every ℓ

— either $\gamma_\ell = (i_\ell, i_{\ell+1})$ ("forward arc"),

— or $\gamma_\ell = (i_{\ell+1}, i_\ell)$ ("backward arc").

Setting $\epsilon_s = 1$ or $\epsilon_s = -1$ depending on whether γ_s is a forward or a backward arc and denoting by A_γ the column of A indexed by γ, we get $\sum_{s=1}^{t-1} \epsilon_s A_{\gamma_s} = 0$, which is impossible, since the columns A_γ, $\gamma \in I$, are linearly independent. □

Network Simplex Algorithm
Building Block I: Computing Basic Solution

$$\min_f \left\{ \sum_{\gamma \in \mathcal{E}} c_\gamma f_\gamma : Af = b,\ f \geq 0 \right\}. \quad \text{(NWU)}$$

♣ As a specialization of the Primal Simplex Method, the Network Simplex Algorithm works with basic feasible solutions.

There is a simple algorithm allowing to specify the basic feasible solution f^I associated with a given basis (i.e., a given spanning tree) I.

Note: f^I should be a flow (i.e., $Af^I = b$) which vanishes outside of I (i.e., $f^I_\gamma = 0$ whenever $\gamma \notin I$).

♠ The algorithm for specifying f^I is as follows:

• G_I is a tree and thus has a leaf i_*; let $\gamma \in I$ be an arc which is incident to node i_*. The flow conservation law specifies f^I_γ:

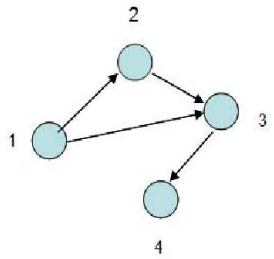
$$f^I_\gamma = \begin{cases} -s_{i_*}, & \gamma = (j_*, i_*) \\ s_{i_*}, & \gamma = (i_*, j_*) \end{cases}$$

We specify f^I_γ, eliminate from G_I node i_* and arc γ and update s_{j_*} to account for the flow in the arc γ:

$$s^+_{j_*} = s_{j_*} + s_{i_*}$$

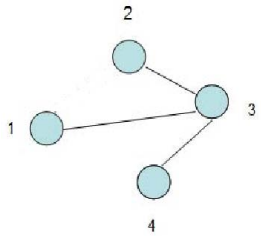
We end up with $(m-1)$-node graph G^1_I equipped with the updated $(m-1)$-dimensional vector of external supplies s^1 obtained from s by eliminating s_{i_*} and replacing s_{j_*} with $s_{j_*} + s_{i_*}$. Note that the total of updated supplies is 0.

• We apply to G^1_I the same procedure as to G^I, thus getting one more entry in f^I, reducing the number of nodes by one and updating the vector of external supplies, and proceed in this fashion until all entries f^I_γ, $\gamma \in I$, are specified.

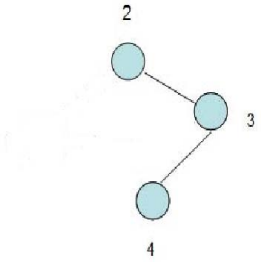


$s = [1; 2; 3; -6]$

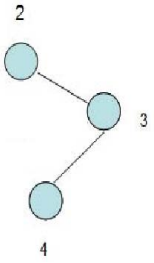
♠ **Illustration:** Let $I = \{(1,3),(2,3),(3,4)\}$. Graph G_I is



• We choose a leaf in G_I, specifically, node 1, and set $f^I_{1,3} = s_1 = 1$. We then eliminate from G_I the node 1 and the incident arc, thus getting the graph G^1_I, and convert s into s^1:

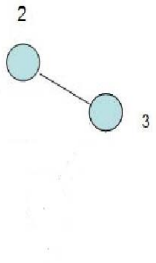


$s^1_2 = s_2 = 2; s^1_3 = s_3 + s_1 = 4; s^1_4 = s_4 = -6$



$$s_2^1 = s_2 = 2; s_3^1 = s_3 + s_1 = 4; s_4^1 = s_4 = -6$$

• We choose a leaf in the new graph, say, the node 4, and set $f^I_{3,4} = -s^1_4 = 6$. We then eliminate from G^1_I the node 4 and the incident arc, thus getting the graph G^2_I, and convert s^1 into s^2:



$$s_2^2 = s_2^1 = 2; s_3^2 = s_3^1 + s_4^1 = -2$$

• We choose a leaf, say, node 3, in the new graph and set $f^I_{2,3} = -s^2_3 = 2$. The algorithm is completed. The resulting basic flow is

$$f^I_{1,2} = 0, f^I_{2,3} = 2, f^I_{1,3} = 1, f^I_{3,4} = 6.$$

Building Block II: Computing Reduced Costs

There is a simple algorithm allowing to specify the reduced costs associated with a given basis (i.e., a given spanning tree) I.

Note: The reduced costs should form a vector $c^I = c - A^T\lambda^I$ and should satisfy $c^I_\gamma = 0$ whenever $\gamma \in I$. Observe that the span of the columns of A^T is the same as the span of the columns of P^T, thus, we lose nothing when setting $c^I = c - P^T\mu^I$. The requirement $c^I_\gamma = 0$ for $\gamma \in I$ reads

$$c_{ij} = \mu_i - \mu_j \, \forall \gamma = (i,j) \in I \quad (*),$$

while

$$c^I_{ij} = c_{ij} - \mu_i + \mu_j \; \forall \gamma = (i,j) \in \mathcal{E} \qquad (!)$$

♠ To achieve $(*)$, we act as follows. When building f^I, we build a sequence of trees $G^0_I := G_I$, G^1_I,...,G^{m-2}_I in such a way that G^{s+1}_I is obtained from G^s_I by eliminating a leaf and the arc incident to this leaf.

To build μ, we look through these graphs in the backward order.

• G_I^{m-2} has two nodes, say, i_* and j_*, and a single arc which corresponds to an arc $\gamma = (\gamma^{\mathsf{s}}, \gamma^{\mathsf{f}}) \in I$, where either $\gamma^{\mathsf{s}} = j_*, \gamma^{\mathsf{f}} = i_*$, or $\gamma^{\mathsf{s}} = i_*, \gamma^{\mathsf{f}} = j_*$. We choose μ_{i_*}, μ_{j_*} such that

$$c_\gamma = \mu_{\gamma^{\mathsf{s}}} - \mu_{\gamma^{\mathsf{f}}}$$

• Let we already have assigned all nodes i of G_I^s with the "potentials" μ_i in such a way that for every arc $\gamma \in I$ linking the nodes from G_I^s it holds

$$c_\gamma = \mu_{\gamma^{\mathsf{s}}} - \mu_{\gamma^{\mathsf{f}}} \qquad (*_s)$$

The graph G_I^{s-1} has exactly one node, let it be i_*, which is not in G_I^s, and exactly one arc $\{j_*, i_*\}$, obtained from an oriented arc $\bar{\gamma} \in I$, which is incident to i_*. Note that μ_{j_*} is already defined. We specify μ_{i_*} from the requirement

$$c_{\bar{\gamma}} = \mu_{\bar{\gamma}^{\mathsf{s}}} - \mu_{\bar{\gamma}^{\mathsf{f}}},$$

thus ensuring the validity of $(*_{s-1})$.

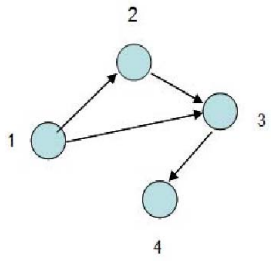
• After G_I^0 is processed, we get the potentials μ satisfying the target relation

$$c_{ij} = \mu_i - \mu_j \, \forall \gamma = (i,j) \in I \quad (*),$$

and then define the reduces costs according to

$$c_\gamma^I = c_\gamma + \mu_{\gamma^{\mathsf{f}}} - \mu_{\gamma^{\mathsf{s}}} \;\; \gamma \in \mathcal{E}.$$

Illustration: Let G and I be as in the previous illustrations:

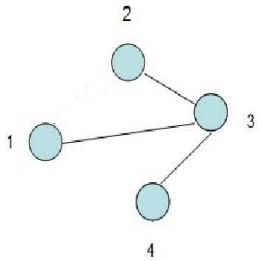


$$I = \{(1,3),(2,3),(3,4)\}$$

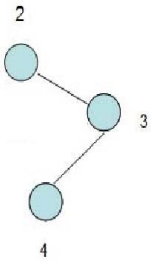
and let

$$c_{1,2} = 1, c_{2,3} = 4, c_{1,3} = 6, c_{3,4} = 8$$

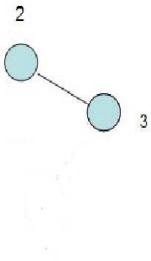
We have already found the corresponding graphs G_I^s:



G_I^0



G_I^1



G_I^2

- We look at graph G_I^2 and set $\mu_2 = 0, \mu_3 = -4$, thus ensuring $c_{2,3} = \mu_2 - \mu_3$
- We look at graph G_I^1 and set $\mu_4 = \mu_3 - c_{3,4} = -12$, thus ensuring $c_{3,4} = \mu_3 - \mu_4$
- We look at graph G_I^0 and set $\mu_1 = \mu_3 + c_{1,3} = 2$, thus ensuring $c_{1,3} = \mu_1 - \mu_3$

The reduced costs are

$$c_{1,2}^I = c_{1,2} + \mu_2 - \mu_1 = -1, c_{1,3}^I = c_{2,3}^I = c_{3,4}^I = 0.$$

Iteration of the Network Simplex Algorithm

$$\min_f \left\{ \sum_{\gamma \in \mathcal{E}} c_\gamma f_\gamma : Af = b,\ f \geq 0 \right\}. \quad \text{(NWU)}$$

♠ At the beginning of an iteration, we have at our disposal a basis – a spanning tree – I along with the associated *feasible* basic solution f^I and the vector of reduced costs c^I.

• It is possible that $c^I \geq 0$. In this case we terminate, f^I being an optimal solution.

Note: We are solving a *minimization* problem, so that sufficient condition for optimality is that the reduced costs are *nonnegative*.

• If c^I is not nonnegative, we identify an arc $\widehat{\gamma}$ in G such that $c^I_{\widehat{\gamma}} < 0$. Note: $\gamma \notin I$ due to $c^I_\gamma = 0$ for $\gamma \in I$. $f_{\widehat{\gamma}}$ enters the basis.

• It may happen that $\widehat{\gamma} = (i_1, i_2)$ is inverse to an arc $\widetilde{\gamma} = (i_2, i_1) \in I$. Setting $h_{\widehat{\gamma}} = h_{\widetilde{\gamma}} = 1$ and $h_\gamma = 0$ when γ is different from $\widetilde{\gamma}$ and $\widehat{\gamma}$, we have $Ah = 0$ and $c^T h = [c^I]^T h = c^I_{\widehat{\gamma}} < 0$

$\Rightarrow$ $f^I + th$ is feasible for all $t > 0$ and $c^T(f^I + th) \to -\infty$ as $t \to \infty$

$\Rightarrow$ The problem is unbounded, and we terminate.

• Alternatively, $\widehat{\gamma} = (i_1, i_2)$ is not inverse to an arc in I. In this case, adding to G_I the arc $\{i_1; i_2\}$, we get a graph with a single cycle C, that is, we can point out a sequence of nodes $i_1, i_2, ..., i_t = i_1$ such that $t > 3$, $i_1, ..., i_{t-1}$ are distinct, and for every s, $2 \leq s \leq t-1$,
— either the arc $\gamma_s = (i_s, i_{s+1})$ is in I ("forward arc γ_s"),
— or the arc $\gamma_s = (i_{s+1}, i_s)$ is in I ("backward arc γ_s").
We set $\gamma_1 = \widehat{\gamma} = (i_1, i_2)$ and treat γ_1 as a forward arc of C.
We define the flow h according to

$$h_\gamma = \begin{cases} 1, & \gamma = \gamma_s \text{ is forward} \\ -1, & \gamma = \gamma_s \text{ is backward} \\ 0, & \gamma \text{ is distinct from all } \gamma_s \end{cases}$$

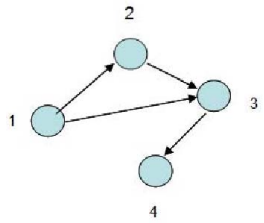
By construction, $Ah = 0$, $h_{\widehat{\gamma}} = 1$ and $h_\gamma = 0$ whenever $\gamma \neq \widehat{\gamma}$ and $\gamma \notin I$.
Setting $f^I(t) = f^I + th$, we have $Af^I(t) \equiv b$ and $c^T f^I(t) = c^T f^I + t[c^I]^T h = c^T[f^I] + tc^I_{\widehat{\gamma}} \to -\infty,\ t \to \infty$.
• If $h \geq 0$, $f^I(t)$ is feasible for all $t \geq 0$ and $c^T f^I(t) \to -\infty$, $t \to \infty$
$\Rightarrow$ the problem is unbounded, and we terminate

- If $h_\gamma = -1$ for some γ (all these γ are in I), we set

$$\widetilde{\gamma} = \operatorname*{argmin}_{\gamma \in I} \{ f^I_\gamma : h_\gamma = -1 \} \quad [f_{\widetilde{\gamma}} \text{ leaves the basis}]$$
$$\bar{t} = f^I_{\widetilde{\gamma}}$$
$$I^+ = (I \cup \widehat{\gamma}) \backslash \{\widetilde{\gamma}\}$$
$$f^{I^+} = f^I + \bar{t} h$$

We have defined the new basis I^+ along with the corresponding basic feasible flow f^{I+} and pass to the next iteration.

Illustration:



$I = \{(1,3),(2,3),(3,4)\}$ $f^I_{1,2} = 0, f^I_{2,3} = 2, f^I_{1,3} = 1, f^I_{3,4} = 6$ $c^I_{1,2} = -1, c^I_{1,3} = c^I_{2,3} = c^I_{3,4} = 0$

- $c^I_{1,2} < 0 \Rightarrow f_{1,2}$ enters the basis
- Adding $\widehat{\gamma} = (1,2)$ to I, we get the cycle

$$i_1 = 1 (1,2) i_2 = 2 (2,3) i_3 = 3 (1,3) i_4 = 1$$

(magenta: forward arcs, blue: backward arcs)

$\Rightarrow h_{1,2} = h_{2,3} = 1, h_{1,3} = -1, h_{3,4} = 0$

$\Rightarrow f^I_{1,2}(t) = t, f^I_{2,3}(t) = 2 + t, f^I_{1,3}(t) = 1 - t, f^I_{3,4}(t) = 6$

$\Rightarrow$ the largest t for which $f^I(t)$ is feasible is $t = 1$, variable $f_{1,3}$ leaves the basis

$\Rightarrow I^+ = \{(1,2),(2,3),(3,4)\}$, $f^{I^+}_{1,2} = 1, f^{I^+}_{2,3} = 3, f^{I^+}_{3,4} = 6, f^{I^+}_{1,3} = 0$, cost change is $c^I_{1,2} f^{I^+}_{1,2} = -1$.

The iteration is over.

- New potentials are given by

$$1 = c_{1,2} = \mu_1 - \mu_2, 4 = c_{2,3} = \mu_2 - \mu_3, 8 = c_{3,4} = \mu_3 - \mu_4$$
$$\Rightarrow \mu_1 = 1, \mu_2 = 0, \mu_3 = -4, \mu_4 = -12$$
$$c^{I^+}_{1,2} = c^{I^+}_{2,3} = c^{I^+}_{3,4} = 0,\ c^{I^+}_{1,3} = c_{1,3} - \mu_1 + \mu_3 = 6 - 1 - 4 = 1$$

$\Rightarrow$ New reduced costs are nonnegative

$\Rightarrow f^{I^+}$ is optimal.

Capacitated Network Flow Problem

♣ Capacitated Network Flow problem on a graph $G = (\mathcal{N}, \mathcal{E})$ is to find a minimum cost flow on G which fits given external supplies and obeys arc capacity bounds:

$$\min_f \left\{ c^T f : Af = b,\ 0 \le f \le u \right\}, \qquad \text{(NW)}$$

- A is the $(m-1) \times n$ matrix composed of the first $m-1$ rows of the incidence matrix P of G,
- $b = [s_1; s_2; ...; s_{m-1}]$ is the vector composed of the first $m-1$ entries of the vector s of external supplies,
- u is the vector of arc capacities.

♣ We are about to present a modification of the Network Simplex algorithm for solving (NW).

Primal Simplex Method for Problems with Bounds on Variables

♣ Consider an LO program

$$\min_x \left\{ c^T x : Ax = b,\ \ell_j \le x_j \le u_j,\ 1 \le j \le n \right\} \qquad (P)$$

Standing assumptions:

- for every j, $\ell_j < u_j$ ("no fixed variables")
- for every j, either $\ell_j > -\infty$, or $u_j < +\infty$, or both ("no free variables");
- The rows in A are linearly independent.

Note: (P) is not in the standard form unless $\ell_j = 0, u_j = \infty$ for all j.

However: The PSM can be adjusted to handle problems (P) in nearly the same way as it handles the standard form problems.

$$\min_x \left\{ c^T x : Ax = b,\ \ell_j \le x_j \le u_j,\ 1 \le j \le n \right\} \qquad (P)$$

♣ **Observation I:** *if x is a vertex of the feasible set X of (P), then there exists a basis I of A such that all nonbasic entries in x are on the bounds:*

$$\forall j \notin I : x_j = \ell_j \text{ or } x_j = u_j \qquad (*)$$

Vice versa, every feasible solution x which, for some basis I, has all nonbasic entries on the bounds, is a vertex of X.

Proof: identical to the one for the standard case.

$$\min_x \left\{ c^T x : Ax = b,\ \ell_j \le x_j \le u_j,\ 1 \le j \le n \right\} \quad (P)$$

Definition: *Vector x is called basic solution associated with a basis I of A, if $Ax = b$ and all non-basic (with indexes not in I) entries in x are on their bounds.*

Observation I says that *the vertices of the feasible set of (P) are exactly the basic solutions which are feasible.*

Difference with the standard case: In the standard case, a basis uniquely defines the associated basic solution. In the case of (P), there can be as many basic solutions associated with a given basis I as many ways there are to set nonbasic variables to their bounds. *After nonbasic variables are somehow set to their bounds, the basic variables are uniquely defined:*

$$x_I = A_I^{-1}[b - \textstyle\sum_{j \notin I} x_j A^j]$$

where A^j are columns of A and A_I is the submatrix of A composed of columns with indexes in I.

$$\min_x \left\{ c^T x : Ax = b,\ \ell_j \le x_j \le u_j,\ 1 \le j \le n \right\} \quad (P)$$

♣ **Reduced costs.** Given a basis I for A, the corresponding reduced costs are defined exactly as in the standard case: as the vector c^I uniquely defined by the requirements that it is of the form $c - A^T\lambda$ and all basic entries in the vector are zeros: $c^I_j = 0,\ j \in I$. The formula for c^I is

$$c^I = c - A^T \overbrace{A_I^{-T} c_I}^{\lambda^I}$$

where c_I is the vector composed of the basic entries in c.

Note: As in the standard case, *reduced costs associated with a basis are uniquely defined by this basis*.

♣ **Observation II:** *On the feasible affine plane $\mathcal{M} = \{x : Ax = b\}$ of (P) one has*

$$c^T x - [c^I]^T x = \text{const}$$

In particular,

$$x', x'' \in \mathcal{M} \Rightarrow c^T x' - c^T x'' = [c^I]^T x' - [c^I]^T x''$$

Proof. Indeed, if $x \in \mathcal{M}$, then

$$[c^I]^T x - c^T x = [c - A^T\lambda^I]^T x - c^T x = -[\lambda^I]^T Ax = -[\lambda^I]^T b$$

and the result is independent of $x \in \mathcal{M}$.

$$\min_x \left\{ c^T x : Ax = b,\ \ell_j \le x_j \le u_j,\ 1 \le j \le n \right\} \quad (P)$$

♣ **Corollary:** *Let I be a basis of A, x^* be an associated feasible basic solution, and c^I be the associated with I vector of reduced costs. Assume that nonbasic entries in c^I and nonbasic entries in x satisfy the relation*

$$\forall j \notin I : \begin{cases} x_j^* = \ell_j \Rightarrow c_j^I \ge 0 \\ x_j^* = u_j \Rightarrow c_j^I \le 0 \end{cases}$$

Then x^* is an optimal solution to (P).

Proof. By Observation II, problem

$$\min_x \left\{ [c^I]^T x : Ax = b,\ \ell_j \le x_j \le u_j,\ 1 \le j \le n \right\} \quad (P')$$

is equivalent to (P) $\Rightarrow$ it suffices to prove that x^* is optimal for (P'). This is evident:

$$\begin{array}{rcl} x \text{ is feasible } \Rightarrow & & \\ [c^I]^T[x - x^*] & = & \sum_{j \in I} \overbrace{c_j^I}^{=0} [x_j - x_j^*] \\ & & + \sum_{\substack{j \notin I, \\ x_j^* = \ell_j}} \overbrace{c_j^I}^{\ge 0} \underbrace{[x_j - \ell_j]}_{\ge 0} + \sum_{\substack{j \notin I, \\ x_j^* = u_j}} \overbrace{c_j^I}^{\le 0} \underbrace{[x_j - u_j]}_{\le 0} \\ & \ge & 0 \end{array}$$

$$\min_x \left\{c^T x : Ax = b,\, \ell_j \le x_j \le u_j,\, 1 \le j \le n\right\} \quad (P)$$

♣ **Iteration** of the PSM as applied to (P) is as follows:

♠ At the beginning of the iteration, we have at our disposal current basis I, an associated feasible basic solution x^I, and the associated reduced costs c^I and set

$$J_\ell = \{j \notin I : x^I_j = \ell_j\},\ J_u = \{j \notin I : x^I_j = u_j\}.$$

♠ At the iteration, we

A. Check whether $c^I_j \ge 0\, \forall j \in J_\ell$ and $c^I_j \le 0\, \forall j \in J_u$. If it is the case, we terminate – x is an optimal solution to (P) (by Corollary).

B. If we do not terminate according to **A**, we pick j_* such that $j_* \in J_\ell$ & $c^I_{j_*} < 0$ or $j_* \in J_u$ & $c^I_{j_*} > 0$. Further, we build the ray $x(t)$ of solutions such that

$$\begin{array}{l} Ax(t) = b\, \forall t \ge 0 \\ x_j(t) = x^I_j\, \forall j \notin [I \cup \{j_*\}] \\ x_{j_*}(t) = \left\{ \begin{array}{l} x_{j_*} + t = \ell_{j_*} + t,\ j_* \in J_\ell \\ x_{j_*} - t = u_{j_*} - t,\ j_* \in J_u \end{array} \right. \end{array}$$

Note that

$$x(t) = x^I - \epsilon t A_I^{-1} A^{j_*},\ \epsilon = \left\{ \begin{array}{rl} 1, & j_* \in J_\ell \\ -1, & j_* \in J_u \end{array} \right.$$

$$\min_x \left\{c^T x : Ax = b,\ \ell_j \le x_j \le u_j,\ 1 \le j \le n\right\} \qquad (P)$$
$$J_\ell = \{j \notin I : x^I_j = \ell_j\},\ J_u = \{j \notin I : x^I_j = u_j\}.$$

Situation: We have built a ray of solutions

$$x(t) = x^I - \epsilon t A_I^{-1} A^{j_*},\ \epsilon = \begin{cases} 1, & j_* \in J_\ell \\ -1, & j_* \in J_u \end{cases}$$

such that

- $Ax(t) = b$ for all $t \ge 0$
- the only nonzero entries in $x(t) - x^I$ are the basic entries and the j_*-th entry, the latter being equal to ϵt.

Note: By construction, the objective improves along the ray $x(t)$:

$$c^T[x(t) - x^I] = [c^I]^T[x(t) - x^I] = c^I_{j_*}[x_{j_*}(t) - x^I_{j_*}] = c^I_{j_*}\epsilon t = -|c^I_{j_*}|t \qquad \heartsuit$$

$x(0) = x^I$ is feasible. It may happen that as $t \ge 0$ grows, all entries $x_j(t)$ of $x(t)$ stay all the time in their feasible ranges $[\ell_j, u_j]$. If it is the case, we have discovered problem's unboundedness and terminate.

$\heartsuit$ Alternatively, when t grows, some of the entries in $x(t)$ eventually leave their feasible ranges. In this case, we specify the largest $t = \bar{t} \ge 0$ such that $x(\bar{t})$ still is feasible. As $t = \bar{t}$, one of the depending on t entries $x_{i_*}(t)$ is about to become infeasible — it is in the feasible range when $t = \bar{t}$ and leaves this range when $t > \bar{t}$. We set $I^+ = I \cup \{j_*\}\backslash\{i_*\}$, $x^{I^+} = x(\bar{t})$, update c^I into c^{I^+} and pass to the next iteration.

♠ As a result of an iteration, the following can occur:

♡ We terminate with optimal solution x^I and "optimality certificate" c^I

♡ We terminate with correct conclusion that the problem is unbounded

♡ We pass to the new basis I^+ and the new feasible basic solution x^{I^+}. If it happens,

- the objective either improves, or remains the same. The latter can happen only when $x^{I^+} = x^I$ (i.e., the above $\bar{t} = 0$). Note that this option is possible *only when some of the basic entries in* x^I *are on their bounds* ("degenerate basic feasible solution"), and in this case the basis definitely changes;
- the basis can change and can remain the same (the latter happens if $i_* = j_*$. i.e., as a result of the iteration, certain non-basic variable jumps from one endpoint of its feasible range to another endpoint of this range).

♠ In the nondegenerate case (i.e., there are no degenerate basic feasible solutions), the PSM terminates in finite time.

♠ Same as the basic PSM, the method can be initialized by solving auxiliary Phase I program with readily available basic feasible solution.

♣ The Network Simplex Method for the capacitated Network Flow problem

$$\min_f \left\{ c^T f : Af = b,\, \ell \leq f \leq u \right\}, \qquad \text{(NW)}$$

associated with a graph $G = (\mathcal{N} = \{1, ..., m\}, \mathcal{E})$ is the network flow adaptation of the above general construction. We assume that G is connected. Then

♠ *Bases of* A are spanning trees of G – the collections of $m-1$ arcs from $\mathcal{E}$ which form a tree after their orientations are ignored;

♠ *Basic solution* f^I associated with basis I is a (not necessarily feasible) flow such that

$$Af^I = b \text{ and } \forall \gamma \notin I : f^I_\gamma \text{ is either } \ell_\gamma, \text{ or } u_\gamma.$$

$$\min_f \left\{ c^T f : Af = b,\, \ell \le f \le u \right\}, \qquad \text{(NW)}$$

♠ At the beginning of an iteration, we have at our disposal a basis I along with associated *feasible* basic solution f^I.

♡ In course of an iteration,

- we build the reduced costs $c^I_{ij} = c_{ij} + \mu_j - \mu_i$, $(i,j) \in \mathcal{E}$, where the "potentials" $\mu_1, ..., \mu_m$ are given by the requirement

$$c_{ij} = \mu_i - \mu_j \, \forall (i,j) \in I$$

The algorithm for building the potentials is exactly the same as in the uncapacitated case.

- we check the signs of the non-basic reduced costs to find a "promising" arc – an arc $\gamma_* = (i_*, j_*) \notin I$ such that either $f^I_{\gamma_*} = \ell_{\gamma_*}$ and $c^I_{\gamma_*} < 0$, or $f^I_{\gamma_*} = u_{\gamma_*}$ and $c^I_{\gamma_*} > 0$.

♡ If no promising arc exists, we terminate — f^I is an optimal solution to (NW).

Alternatively, we find a promising arc $\gamma_* = (i_*, j_*)$. Note that $\gamma \notin I$.

Case A: *γ_* is not inverse to an arc in I.* We add arc γ_* to I. Since I is a spanning tree, there exists a cycle

$$C = i_1 = i_* \underbrace{(i_*, j_*)}_{\equiv \gamma_1} i_2 = j_* \gamma_2 i_3 \gamma_3 i_4 ... \gamma_{t-1} i_t = i_1$$

where

- $i_1, ..., i_{t-1}$ are distinct from each other nodes,
- $\gamma_2, ..., \gamma_{t-1}$ are distinct from each other arcs *from I*, and
- for every s, $1 \le s \le t-1$,
  - either $\gamma_s = (i_{s-1}, i_s)$ ("forward arc"),
  - or $\gamma_s = (i_s, i_{s-1})$ ("backward arc").

Note: $\gamma_1 = (i_*, j_*)$ is a forward arc.

$$C = i_1 = i_* \underbrace{(i_*, j_*)}_{\equiv \gamma_1} i_2 = j_* \gamma_2 i_3 \gamma_3 i_4 ... \gamma_{t-1} i_t = i_1$$
$$\gamma_2, ..., \gamma_{t-1} \in I$$

♡ We identify the cycle C and define the flow h as follows:

$$h_\gamma = \begin{cases} 0, & \gamma \text{ does not enter } C \\ \epsilon, & \gamma \text{ is a forward arc in } C \\ -\epsilon, & \gamma \text{ is a backward arc in } C \end{cases}$$

$$\epsilon = \begin{cases} 1, & f^I_{\gamma_*} = \ell_{\gamma_*} \ \& \ c^I_{\gamma_*} < 0 \\ -1, & f^I_{\gamma_*} = u_{\gamma_*} \ \& \ c^I_{\gamma_*} > 0 \end{cases}$$

Setting $f(t) = f^I + th$, we ensure that

- $Af(t) \equiv b$
- $c^T f(t)$ decreases as t grows,
- $f(0) = f^I$ is feasible,
- $f_{\gamma_*}(t) = f^I_{\gamma_*} + \epsilon t$ is in the feasible range for all small enough $t \geq 0$,
- $f_\gamma(t)$ are affine functions of t and are constant when γ is not in C.

Setting $f(t) = f^I + th$, we ensure that

- $Af(t) \equiv b$
- $c^T f(t)$ decreases as t grows,
- $f(0) = f^I$ is feasible,
- $f_{\gamma_*}(t) = f^I_{\gamma_*} + \epsilon t$ is in the feasible range for all small enough $t \geq 0$,
- $f_\gamma(t)$ are affine functions of t and are constant when γ is not in C.

♡ It is easy to check whether $f(t)$ remains feasible for all $t \geq 0$. If it is the case, the problem is unbounded, and we terminate.

Alternatively, it is easy to find the largest $t = \bar{t} \geq 0$ such that $f(\bar{t})$ still is feasible. When $t = \bar{t}$, at least one of the flows $f_\gamma(t)$ which do depend on t ($\Rightarrow$ γ belongs to C) is about to leave its feasible range. We specify the corresponding arc $\widehat{\gamma}$ and define

- the new basis as $I^+ = I \cup \{\gamma_*\} \backslash \{\widehat{\gamma}\}$
- the new basic feasible solution as $f^{I^+} = f(\bar{t})$

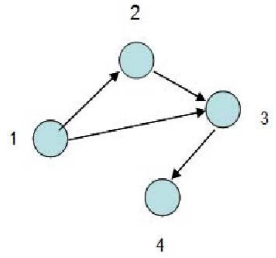
and proceed to the next iteration.

Case B: $\gamma_* = (i_*, j_*)$ is inverse to an arc $\gamma'_* = (j_*, i_*) \in I$. Here we act exactly as in Case A, but with

$$f(t) = \begin{cases} f^I_\gamma, & \gamma \neq \gamma_*, \gamma \neq \gamma'_* \\ f^I_\gamma + \epsilon t, & \gamma = \gamma_* \\ f^I_\gamma - \epsilon t, & \gamma = \gamma'_* \end{cases}$$

$$\epsilon = \begin{cases} 1, & f^I_{\gamma_*} = \ell_{\gamma_*} \ \& \ c^I_{\gamma_*} < 0 \\ -1, & f^I_{\gamma_*} = u_{\gamma_*} \ \& \ c^I_{\gamma_*} > 0 \end{cases}$$

♣ Illustration:



$$c_{1,2}=1, c_{2,3}=4, c_{1,3}=6, c_{3,4}=8,\ s=[1;2;3;-6]$$
$$\ell_{ij}=0\,\forall i,j; u_{1,2}=\infty, u_{1,3}=\infty, u_{2,3}=2, u_{3,4}=7$$

♡ Let $I=\{(1,3),(2,3),(3,4)\}$

$\Rightarrow$ $f^I_{1,2}=0, f^I_{2,3}=2, f^I_{1,3}=1, f^I_{3,4}=6$ — feasible!

$\Rightarrow$ $c^I_{1,2}=-1, c^I_{1,3}=c^I_{2,3}=c^I_{3,4}=0$

♡ $c^I_{1,2}=-1<0$ and $f^I_{1,2}=0=\ell_{1,2}$

$\Rightarrow$ $\gamma_*=(1,2)$ is a promising arc, and it is profitable to increase its flow ($\epsilon=1$)

♡ Adding $(1,2)$ to I, we get the cycle

$$1(1,2)2(2,3)3(1,3)1$$

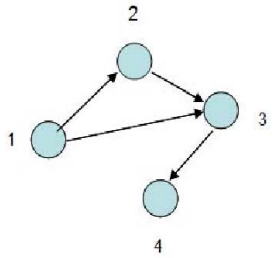
$\Rightarrow$ $h_{1,2}=1, h_{2,3}=1, h_{1,3}=-1, h_{3,4}=0$

$\Rightarrow$ $f_{1,2}(t)=t, f_{2,3}(t)=2+t, f_{1,3}(t)=1-t, f_{3,4}(t)=6$

♡ The largest $\bar{t}$ for which all $f_\gamma(t)$ are in their feasible ranges is $\bar{t}=0$, the corresponding arc being $\widehat{\gamma}=(2,3)$

$\Rightarrow$ The new basis is $I^+=\{(1,2),(1,3),(3,4)\}$, the new basic feasible flow is the old one:

$$f^{I^+}_{1,2}=0, f^{I^+}_{2,3}=2, f^{I^+}_{1,3}=1, f^{I^+}_{3,4}=6$$



♡ The new basis is $I^+ = \{(1,2),(1,3),(3,4)\}$, the new basic feasible flow is the old one:

$$f^{I^+}_{1,2} = 0, f^{I^+}_{2,3} = 2, f^{I^+}_{1,3} = 1, f^{I^+}_{3,4} = 6$$

♡ Computing the new reduced costs:

$$1 = c_{1,2} = \mu_1 - \mu_2, 6 = c_{1,3} = \mu_1 - \mu_3, 8 = c_{3,4} = \mu_3 - \mu_4$$
$$\Rightarrow \mu_1 = 1, \mu_2 = 0, \mu_3 = -5, \mu_4 = -13$$
$$\Rightarrow c^{I^+}_{1,2} = c^{I^+}_{1,3} = c^{I^+}_{3,4} = 0, c^{I^+}_{23} = c_{23} + \mu_3 - \mu_2 = 4 - 5 + 0 = -1$$

All positive reduced costs (in fact, there are none) correspond to flows on their lower bounds, and all negative reduced costs (namely, $c^{I^+}_{2,3} = -1$) correspond to flows on their upper bounds

$\Rightarrow f^{I^+}$ is optimal!

PART III.
LO and Beyond: Polynomial Time Algorithms

Lecture III.1
Ellipsoid Algorithm
and
Efficient Solvability of LO

Complexity of LO

♣ A *generic computational problem* $\mathcal{P}$ is a family of *instances* — particular problems of certain structure, identified within $\mathcal{P}$ by a (real or binary) *data vector*. For example, LO is a generic problem $\mathcal{LO}$ with instances of the form

$$\min_x \left\{c^T x : Ax \leq b\right\} \quad [A = [A_1; ...; A_n] : m \times n]$$

An $\mathcal{LO}$ instance – a particular LO program – is specified within this family by the *data* (c, A, b) which can be arranged in a single vector

$$[m; n; c; A_1; A_2; ...; A_n; b].$$

♣ Investigating *complexity* of a generic computational problem $\mathcal{P}$ is aimed at answering the question

How the computational effort of solving an instance P of $\mathcal{P}$ depends on the "size" Size(P) of the instance?

♠ Precise meaning of the above question depends on how we measure the computational effort and how we measure the sizes of instances.

♣ Let the generic problem in question be $\mathcal{LO}$ – Linear Optimization. There are two natural ways to treat the complexity of $\mathcal{LO}$:

♠ **Real Arithmetic Complexity Model:**

• We allow the instances to have real data and measure the size of an instance P by the sizes $m(P)$, $n(P)$ of the constraint matrix of the instance, say, set $\text{Size}(P) = m(P)n(P)$

• We take, as a model of computations, the Real Arithmetic computer – an idealized computer capable to store reals and carry out operations of precise Real Arithmetics (four arithmetic operations and comparison), each operation taking unit time.

• The computational effort in a computation is defined as the *running time* of the computation. Since all operations take unit time, this effort is just the total number of arithmetic operations in the computation.

$$\min_x \left\{ c^T x : Ax \leq b \right\} \qquad (P)$$

♠ **Rational Arithmetic ("Fully finite") Complexity Model:**

• We allow the instances to have rational data and measure the size of an instance P by its *binary length* $\mathcal{L}(P)$ — the total number of binary digits in the data c, A, b of the instance;

• We take, as a model of computations, the Rational Arithmetic computer – a computer which is capable to store rational numbers (represented by pairs of integers) and to carry out operations of precise Rational Arithmetics (four arithmetic operations and comparison), the time taken by an operation being a polynomial $\pi(\ell)$ in the total binary length ℓ of operands.

• The computational effort in a computation is again defined as the *running time* of the computation. However, now this effort is *not* just the total number of operations in the computation, since the time taken by an operation depends on the binary length of the operands.

♠ In both Real and Rational Arithmetics complexity models, a *solution algorithm* $\mathcal{B}$ for a generic problem $\mathcal{P}$ is a code for the respective computer such that executing the code on the data of every instance P of $\mathcal{P}$, the running time of the resulting computation is finite, and upon termination the computer outputs an exact solution to the instance P. In the LO case, the latter means that upon termination, the computer outputs

- either an optimal solution to P,
- or the correct claim "P is infeasible,"
- or the correct claim "P is unbounded."

For example, the Primal Simplex Method with anti-cycling pivoting rules is a solution algorithm for $\mathcal{LO}$. Without these rules, the PSM is not a solution algorithm for $\mathcal{LO}$.

♠ A solution algorithm $\mathcal{B}$ for $\mathcal{P}$ is called *polynomial time* ("computationally efficient"), if its running time on every instance P of $\mathcal{P}$ is bounded by a polynomial in the size of the input:

$$\exists a, b: \ \forall P \in \mathcal{P}: \langle \text{running time of } \mathcal{B} \text{ on } P \rangle \leq a \cdot \text{Size}^b(P).$$

♠ We say that a generic problem $\mathcal{P}$ is *polynomially solvable* ("computationally tractable"), if it admits a polynomial time solution algorithm.

Comments:

♠ *Polynomial solvability is a worst-case-oriented concept: we want the computational effort to be bounded by a polynomial of* Size(P) *for every* $P \in \mathcal{P}$. *In real life, we would be completely satisfied by a good bound on the effort of solving most, but not necessary all, of the instances. Why not to pass from the worst to the average running time?*

Answer: For a generic problem $\mathcal{P}$ like $\mathcal{LO}$, with instances coming from a wide spectrum of applications, there is no hope to specify in a meaningful way the distribution according to which instances arise, and thus *there is no meaningful way to say w.r.t. which distribution the probabilities and averages should be taken*.

♠ *Why computational tractability is interpreted as polynomial solvability? A high degree polynomial bound on running time is, for all practical purposes, not better than an exponential bound!*

"Why computational tractability is interpreted as polynomial solvability? A high degree polynomial bound on running time is, for all practical purposes, not better than an exponential bound!"

Answer:

- At the theoretical level, it is good to distinguish first of all between "definitely bad" and "possibly good." Typical non-polynomial complexity bounds are exponential, and these bounds definitely are bad from the worst-case viewpoint. High degree polynomial bounds also are bad, but as a matter of fact, they do not arise.
- The property of $\mathcal{P}$ to be/not to be polynomially solvable remains intact when changing "small details" like how exactly we encode the data of an instance to measure its size, what exactly is the polynomial dependence of the time taken by an arithmetic operation on the bit length of the operands, etc., etc. As a result, the notion of "computational tractability" becomes independent of technical "second order" details.

• After polynomial time solvability is established, we can investigate the complexity in more details — what are the degrees of the corresponding polynomials, what are the best under circumstances data structures and polynomial time solution algorithms, etc., etc. But first thing first...

• Proof of a pudding is in eating: the notion works! As applied in the framework of Rational Arithmetic Complexity Model, it results in fully compatible with common sense classification of Discrete Optimization problems into "difficult" and "easy" ones.

Classes P and NP

♣ Consider a generic discrete problem $\mathcal{P}$, that is, a generic problem with rational data and instances of the form "given rational data d of an instance, find a solution to the instance, that is, a rational vector x such that $\mathcal{A}(d,x)=1$." Here $\mathcal{A}$ is the associated with $\mathcal{P}$ *predicate* – function taking values 1 ("true") and 0 ("false").

Examples:

A. Finding rational solution to a system of linear inequalities with rational data.

Note: From our general theory, a solvable system of linear inequalities with rational data always has a rational solution.

B. Finding optimal rational primal and dual solutions to an LO program with rational data.

Note: Writing down the primal and the dual constraints and the constraint "the duality gap is zero," problem B reduces to problem A.

♣ Given a generic discrete problem $\mathcal{P}$, one can associate with it its *feasibility version* $\mathcal{P}_f$ *"given rational data d of an instance, check whether the instance has a solution."*

♣ Examples of generic feasibility problems:

• Checking feasibility in LO – generic problem with instances "given a finite system $Ax \leq b$ with rational data, check whether the system is solvable"

• Existence of a Hamiltonian cycle in a graph – generic problem with instances "given an undirected graph, check whether it admits a Hamiltonian cycle – a cycle which visits all nodes of the graph"

• Stone problem – generic problem with instances "given finitely many stones with integer weights $a_1, ..., a_n$, check whether it is possible to split them into two groups of equal weights," or, equivalently, check whether the single homogeneous linear equation $\sum_{i=1}^{n} a_i x_i = 0$ has a solution in variables $x_i = \pm 1$.

♣ A generic discrete problem $\mathcal{P}$ with instances "given rational vector of data d, find a rational solution vector x such that $\mathcal{A}(d,x) = 1$" is said to be in class NP, if

• *The predicate $\mathcal{A}$ is easy to compute:* in the Rational Arithmetic Complexity Model, it can be computed in time polynomial in the total bit length $\ell(d) + \ell(x)$ of d and x.

In other words, given the data d of an instance and a candidate solution x, it is easy to check whether x indeed is a solution.

• *If an instance with data d is solvable, then there exists a "short solution" x to the instance* – a solution with the binary length $\ell(x) \leq \pi(\ell(d))$, where $\pi(\cdot)$ is a polynomial which, same as the predicate $\mathcal{A}$, is specific for $\mathcal{P}$.

♠ In our examples:

• Checking existence of a Hamiltonian cycle and the Stones problems clearly are in NP – in both cases, the corresponding predicates are easy to compute, and the binary length of a candidate solution is not larger than the binary length of the data of an instance.

• Checking feasibility in LO also is in NP. Clearly, the associated predicate is easy to compute — indeed, given the rational data of a system of linear inequalities and a rational candidate solution, it takes polynomial in the total bit length of the data and the solution time to verify on a Rational Arithmetic computer whether the candidate solution is indeed a solution. What is *un*clear in advance, is whether a feasible system of linear inequalities with rational data admits a "short" rational solution – one of the binary length polynomial in the total binary length of system's data. As we shall see in the mean time, this indeed is the case.

Note: Class NP is extremely wide and covers the vast majority of, if not all, interesting discrete problems.

♣ We say that a generic discrete problem $\mathcal{P}$ is in class P, if it is in NP and its feasibility version is polynomially solvable in the Rational Arithmetic complexity model.

♠ **Note:** By definition of P, P is contained in NP. *If these classes were equal, there, for all practical purposes, would be no difficult problems at all*, since, as a matter of fact, finding a solution to a solvable instance of $\mathcal{P} \in$ NP reduces to solving a "short" series of related to $\mathcal{P}$ feasibility problems.

♣ *While we do not know whether* P=NP – and this is definitely the major open problem in Computer Science and one of the major open problems in Mathematics – we do know something extremely important – namely, that there exist NP-complete problems.

♠ Let $\mathcal{P}$, $\mathcal{P}^+$ be two generic problems from NP. We say that *$\mathcal{P}$ is polynomially reducible to $\mathcal{P}$, if there exists a polynomial time*, in the Rational Arithmetic complexity model, *algorithm which, given on input the data d of an instance $P \in \mathcal{P}$, converts d into the data d^+ of an instance $P^+ \in \mathcal{P}^+$ such that P^+ is solvable if and only if P is solvable.*

Note: If $\mathcal{P}$ is polynomially reducible to $\mathcal{P}^+$ and the feasibility version $\mathcal{P}_f^+$ of $\mathcal{P}^+$ is polynomially solvable, so is the feasibility version $\mathcal{P}_f$ of $\mathcal{P}$.

♠ A problem $\mathcal{P} \in$ NP is called *NP-complete*, if *every* problem from NP is polynomially reducible to $\mathcal{P}$. In particular:

- all NP-complete problems are polynomially reducible to each other;
- if the Feasibility version of one of NP-complete problems is polynomially solvable, then the Feasibility versions of *all* problems from NP are polynomially solvable.

Facts:

♠ *NP-complete problems do exist.* For example, both checking the existence of a Hamiltonian cycle and Stones are NP-complete.

Moreover, modulo a handful of exceptions, *all generic problems from NP for which no polynomial time solution algorithms are known turn out to be NP-complete.*

Practical aspects:

♡ Many discrete problems are of high practical interest, and basically all of them are in NP; as a result, a huge total effort was invested over the years in search of efficient algorithms for discrete problems. More often than not this effort did not yield efficient algorithms, and the corresponding problems finally turned out to be NP-complete and thus polynomially reducible to each other. Thus, *the huge total effort invested into various difficult combinatorial problems was in fact invested into a single problem and did not yield a polynomial time solution algorithm*.

⇒ *At the present level of our knowledge, it is highly unlikely that NP-complete problems admit efficient solution algorithms, that is, we believe that* P≠NP

♡ *Taking for granted that P≠NP, the interpretation of the real-life notion of computational tractability as the formal notion of polynomial time solvability works perfectly well at least in discrete optimization – as a matter of fact, polynomial solvability/insolvability classifies the vast majority of discrete problems into "difficult" and "easy" in exactly the same fashion as computational practice.*

♡ **However:** For a long time, there was an important exception from the otherwise solid rule "tractability ≡ polynomial solvability" – Linear Programming with rational data. On one hand, LO admits an extremely practically efficient computational tool – the Simplex method – and thus, practically speaking, is computationally tractable. On the other hand, it was unknown for a long time (and partially remains unknown) whether LO is polynomially solvable.

Complexity Status of LO

♣ The complexity of LO can be studied both in the Real and the Rational Arithmetic complexity models, and the related questions of polynomial solvability are as follows:

Real Arithmetic complexity model: *Whether there exists a Real Arithmetics solution algorithm for LO which solves every LO program with real data in a number of operations of Real Arithmetics polynomial in the sizes m (number of constraints) and n (number of variables) of the program?*

This is one of the major open questions in Optimization.

Rational Arithmetic complexity model: *Whether there exists a Rational Arithmetics solution algorithm for LO which solves every LO program with rational data in a number of "bit-wise" operations polynomial in the total binary length of the data of the program?*

This question remained open for nearly two decades until it was answered affirmatively by Leonid Khachiyan in 1978.

Complexity Status of the Simplex Method

♣ Equipped with appropriate anti-cycling rules, the Simplex method becomes a legitimate solution algorithm in both Real and Rational Arithmetic complexity models.

♠ **However,** the only known complexity bounds for the Simplex method in both models are exponential. Moreover, starting with mid-1960's, it is known that these "bad bounds" are not an artifact coming from a poor complexity analysis – they do reflect the bad worst-case properties of the algorithm.

♠ Specifically, Klee and Minty presented a series P_n, $n = 1, 2, ...$ of explicit LO programs as follows:

• P_n has n variables and $2n$ inequality constraints with rational data of the total bit length $O(n^2)$

• the feasible set X_n of P_n is a polytope (bounded nonempty polyhedral set) with 2^n vertices

• the 2^n vertices of X_n can be arranged into a sequence such that every two neighboring vertices are linked by an edge (belong to the same one-dimensional face of X_n), and the objective strictly increases when passing from a vertex to the next one.

$\Rightarrow$ *Unless a special care of the pivoting rules and the starting vertex is taken, the Simplex method, as applied to P_n, can visit all 2^n vertexes of X_n and thus its running time on P_n in both complexity models will be exponential in n, while the size of P_n in both models is merely polynomial in n.*

♠ Later, similar "exponential" examples were built for all standard pivoting rules. However, the family of all possible pivoting rules is too diffuse for analysis, and, strictly speaking, we do *not* know whether the Simplex method can be "cured" – converted to a polynomial time algorithm – by properly chosen pivoting rules.

♣ The question of "whether the Simplex method can be cured" is closely related to the famous *Hirsch Conjecture* as follows. Let $X_{m,n}$ be a polytope in $\mathbb{R}^n$ given by m linear inequalities. An *edge path* on $X_{m,n}$ is a sequence of distinct from each other edges (one-dimensional faces of $X_{m,n}$) in which every two subsequent edges have a point in common (this point is a vertex of X).

Note: The trajectory of the PSM as applied to an LO with the feasible set $X_{m,n}$ is an edge path on $X_{m,n}$ *whatever be the pivoting rules*.

♠ The *Hirsch Conjecture* suggests that every two vertices in $X_{m,n}$ can be linked by an edge path with at most $m-n$ edges, or, equivalently, that the "edge diameter" of $X_{m,n}$:

$$d(X_{m,n}) = \max_{u,v\in \mathrm{Ext}(X_{m,n})} \min_{\text{path}} \langle \#\text{ of edges in a path from } u \text{ to } v\rangle$$

is at most $m-n$.

After several decades of intensive research, this conjecture *in its "literal" form* was disproved in 2010 by pointing out (F. Santos) a 43-dimensional polytope given by 86 inequalities with edge diameter > 43.

Fact: No polynomial in m,n *upper bounds* on $d(x_{m,n})$ are known. If Hirsch Conjecture is "heavily wrong" and no polynomial in m,n upper bound on edge diameter exists, this would be an "ultimate" demonstration of inability to "cure" the Simplex method.

♣ Surprisingly, polynomial solvability of LO with rational data in Rational Arithmetic complexity model turned out to be a byproduct of a completely unrelated to LO Real Arithmetic algorithm for finding *approximate* solutions to *general-type convex* optimization programs – the *Ellipsoid algorithm.*

Solving Convex Problems: Ellipsoid Algorithm

♣ There is a wide spectrum of algorithms capable to approximate *global* solutions of convex problems to *high accuracy* in *"reasonable" time.*
We will start with one of the "universal" algorithms of this type – the *Ellipsoid method* imposing only minimal additional to convexity requirements on the problem.

♣ Consider an optimization program

$$f_* = \min_X f(x) \qquad \text{(P)}$$

- $X \subset \mathbb{R}^n$ is a closed and bounded convex set with a nonempty interior;
- f is a continuous convex function on $\mathbb{R}^n$.

♠ Assume that our "environment" when solving (P) is as follows:

A. We have access to a *Separation Oracle* Sep(X) for X – a routine which, given on input a point $x \in \mathbb{R}^n$, reports whether $x \in X$, and in the case of $x \notin X$, returns a *separator* – a vector $e \neq 0$ such that

$$e^T x \geq \max_{y \in X} e^T y$$

B. We have access to a *First Order* Oracle which, given on input a point $x \in X$, returns the value $f(x)$ and a *subgradient* $f'(x)$ of f at x:

$$\forall y : f(y) \geq f(x) + (y - x)^T f'(x).$$

Note: When f is differentiable, one can set $f'(x) = \nabla f(x)$.

C. We are given positive reals R, r, V such that for some (unknown) c one has

$$\{x : \|x - c\| \leq r\} \subset X \subset \{x : \|x\|_2 \leq R\}$$

and

$$\max_{x \in X} f(x) - \min_{x \in X} f(x) \leq V.$$

♠ **Example:** Consider an optimization program

$$\min_x \left\{ f(x) \equiv \max_{1\le\ell\le L} [p_\ell + q_\ell^T x] : x \in X = \{x : a_i^T x \le b_i,\ 1 \le i \le m\} \right\}$$

W.l.o.g. we assume that $a_i \neq 0$ for all i.

♠ A Separation Oracle can be as follows: given x, the oracle checks whether $a_i^T x \le b_i$ for all i. If it is the case, the oracle reports that $x \in X$, otherwise it finds $i = i_x$ such that $a_{i_x}^T x > b_{i_x}$, reports that $x \notin X$ and returns a_{i_x} as a separator. This indeed is a separator:

$$y \in X \Rightarrow a_{i_x}^T y \le b_{i_x} < a_{i_x}^T x$$

♠ A First Order Oracle can be as follows: given x, the oracle computes the quantities $p_\ell + q_\ell^T x$ for $\ell = 1, ..., L$ and identifies the largest of these quantities, which is exactly $f(x)$, along with the corresponding index ℓ, let it be ℓ_x: $f(x) = p_{\ell_x} + q_{\ell_x}^T x$. The oracle returns the computed $f(x)$ and, as a subgradient $f'(x)$, the vector q_{ℓ_x}. This indeed is a subgradient:

$$f(y) \ge p_{\ell_x} + q_{\ell_x}^T y = [p_{\ell_x} + q_{\ell_x}^T x] + (y - x)^T q_{\ell_x} = f(x) + (y - x)^T f'(x).$$

$$f_* = \min_X f(x) \qquad \text{(P)}$$

- $X \subset \mathbb{R}^n$ is a closed and bounded convex set with a nonempty interior;
- f is a continuous convex function on $\mathbb{R}^n$.
- We have access to a *Separation Oracle* which, given on input a point $x \in \mathbb{R}^n$, reports whether $x \in X$, and in the case of $x \notin X$, returns a separator $e \neq 0$:

$$e^T x \geq \max_{y \in X} e^T y$$

- We have access to a *First Order* Oracle which, given on input a point $x \in X$, returns the value $f(x)$ and a subgradient $f'(x)$ of f:

$$\forall y : f(y) \geq f(x) + (y - x)^T f'(x).$$

- We are given positive reals R, r, V such that for some (unknown) c one has

$$\{x : \|x - c\| \leq r\} \subset X \subset \{x : \|x\|_2 \leq R\}$$

and

$$\max_{x \in X} f(x) - \min_{x \in X} f(x) \leq V.$$

♠ *How to build a good solution method for* (P)*?*

To get an idea, let us start with univariate case.

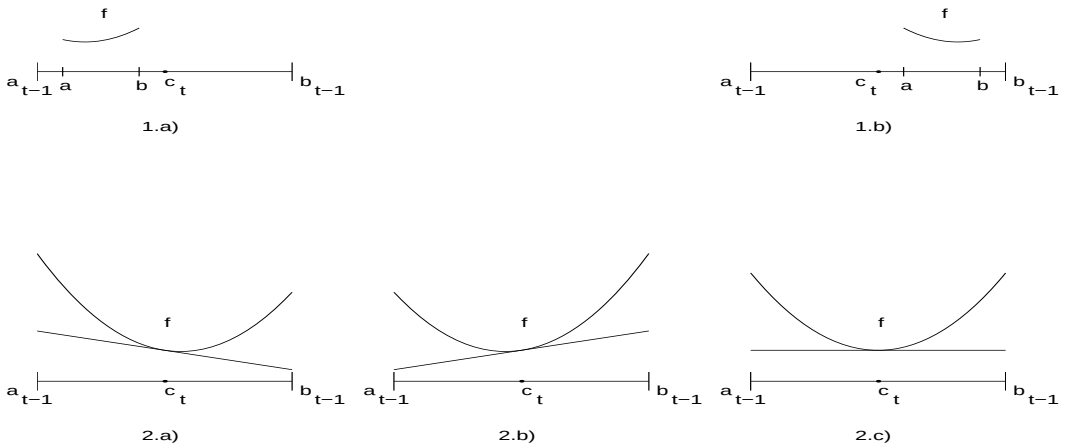
Univariate Case: Bisection

♣ When solving a problem $\min_x \{f(x) : x \in X = [a,b] \subset [-R,R]\}$, by bisection, we recursively update *localizers* – segments $\Delta_t = [a_t, b_t]$ containing the optimal set X_{opt}.

- **Initialization:** Set $\Delta_1 = [-R,R]$ $[\supset X_{\mathrm{opt}}]$

$$\min_x \{f(x) : x \in X = [a,b] \subset [-R,R]\},$$

• **Step t:** Given $\Delta_t \supset X_{\rm opt}$ let c_t be the midpoint of Δ_t. Calling Separation and First Order oracles at e_t, we replace Δ_t by *twice smaller* localizer Δ_{t+1}.



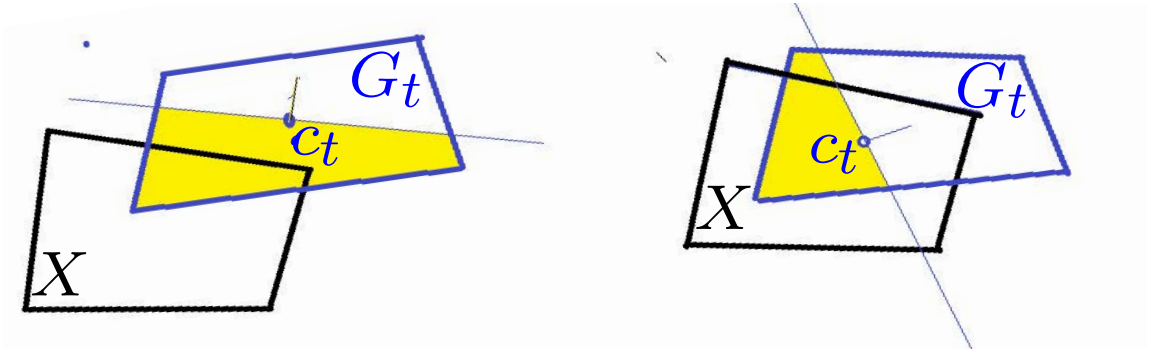
1)	Sep_X says that $c_t \notin X$ and reports, via separator e, on which side of c_t X is. 1.a): $\Delta_{t+1} = [a_t, c_t]$; 1.b): $\Delta_{t+1} = [c_t, b_t]$
2)	Sep_X says that $c_t \in X$, and $\mathcal{O}_f$ reports, via $\mathrm{sign} f'(c_t)$, on which side of c_t $X_{\rm opt}$ is. 2.a): $\Delta_{t+1} = [a_t, c_t]$; 2.b): $\Delta_{t+1} = [c_t, b_t]$; 2.c): $c_t \in X_{\rm opt}$

♠ *Since the localizers rapidly shrink and X is of positive length, eventually some of search points will become feasible, and the nonoptimality of the best found so far feasible search point will rapidly converge to 0 as process goes on.*

$$\mathrm{Opt}(P) = \min_{x \in X \subset \mathbb{R}^n} f(x) \qquad (P)$$

♠ Bisection admits multidimensional extension, called *Generic Cutting Plane Algorithm*, where one builds a sequence of "shrinking" *localizers* G_t – closed and bounded convex domains containing the optimal set X_{opt} of (P).
Generic Cutting Plane Algorithm is as follows:
♠ **Initialization** Select as G_1 a closed and bounded convex set containing X and thus being a localizer.

$$\mathrm{Opt}(P) = \min_{x \in X \subset \mathbb{R}^n} f(x) \qquad (P)$$



Left: $c_t \notin X$ (case A); right: $c_t \in X$ (case B). Yellow polygon: $\widehat{G}_t$.

♠ **Step** $t = 1, 2, ...$**:** Given current localizer G_t,

• Select current *search point* $c_t \in G_t$ and call Separation and First Order oracles to form a *cut* – to find $e_t \neq 0$ s.t. $X_{\mathrm{opt}} \subset \widehat{G}_t := \{x \in G_t : e_t^T x \leq e_t^T c_t\}$.

To this end

— call Sep_X, c_t being the input. If Sep_X says that $c_t \notin X$ and returns a separator, take it as e_t (case A on the picture).

Note: $c_t \notin X \Rightarrow$ *all points from* $G_t \backslash \widehat{G}_t$ *are infeasible*

— if $c_t \in X_t$, call $\mathcal{O}_f$ to compute $f(c_t)$, $f'(c_t)$. If $f'(c_t) = 0$, terminate, otherwise set $e_t = f'(c_t)$ (case B on the picture).

Note: *When* $f'(c_t) = 0$, c_t *is optimal for* (P), *otherwise* $f(x) > f(c_t)$ *at all feasible* $x \in G_t \backslash \widehat{G}_t$

• By the two "Note" above, $\widehat{G}_t$ is a localizer along with G_t. Select a closed and bounded convex set $G_{t+1} \supset \widehat{G}_t$ (it also will be a localizer) and pass to step $t + 1$.

$$\mathrm{Opt}(P) = \min_{x \in X \subset \mathbb{R}^n} f(x) \qquad (P)$$

♣ **Summary:** Given current localizer G_t, selecting a point $c_t \in G_t$ and calling the Separation and the First Order oracles, we can

♠ in the *productive case* $c_t \in X$, find e_t such that

$$e_t^T(x - c_t) > 0 \Rightarrow f(x) > f(c_t)$$

♠ in the *non-productive case* $c_t \notin X$, find e_t such that

$$e_t^T(x - c_t) > 0 \Rightarrow x \notin X$$

$\Rightarrow$ *the set* $\widehat{G}_t = \{x \in G_t : e_t^T(x - c_t) \leq 0\}$ *is a localizer*

♣ *We can select as the next localizer* G_{t+1} *any set containing* $\widehat{G}_t$.

♠ We define approximate solution x^t built in course of $t = 1, 2, ...$ steps as the best – with the smallest value of f – of the feasible search points $c_1, ..., c_t$ built so far.

If in course of the first t steps no feasible search points were built, x^t is undefined.

$$\mathrm{Opt}(P) = \min_{x \in X \subset \mathbb{R}^n} f(x) \qquad (P)$$

♣ Analysing Cutting Plane algorithm

• Let $\mathrm{Vol}(G)$ be the n-dimensional volume of a closed and bounded convex set $G \subset \mathbb{R}^n$.

Note: For convenience, we use, as the unit of volume, the volume of n-dimensional unit ball $\{x \in \mathbb{R}^n : \|x\|_2 \leq 1\}$, and not the volume of n-dimensional unit box.

• Let us call the quantity $\rho(G) = [\mathrm{Vol}(G)]^{1/n}$ the *radius* of G. $\rho(G)$ is the radius of n-dimensional ball with the same volume as G, and this quantity can be thought of as the average linear size of G.

Theorem. *Let convex problem (P) satisfying our standing assumptions be solved by Generic Cutting Plane Algorithm generating localizers G_1, G_2,... and ensuring that $\rho(G_t) \to 0$ as $t \to \infty$. Let $\bar{t}$ be the first step where $\rho(G_{t+1}) < \rho(X)$. Starting with this step, approximate solution x^t is well defined and obeys the "error bound"*

$$f(x^t) - \mathrm{Opt}(P) \leq \min_{\tau \leq t} \left[\frac{\rho(G_{\tau+1})}{\rho(X)}\right] \left[\max_X f - \min_X f\right]$$

$$\mathrm{Opt}(P) = \min_{x \in X \subset \mathbb{R}^n} f(x) \qquad (P)$$

Explanation: Since $\mathrm{int}\, X \neq \emptyset$, $\rho(X)$ is positive, and since X is closed and bounded, (P) is solvable. Let x_* be an optimal solution to (P).

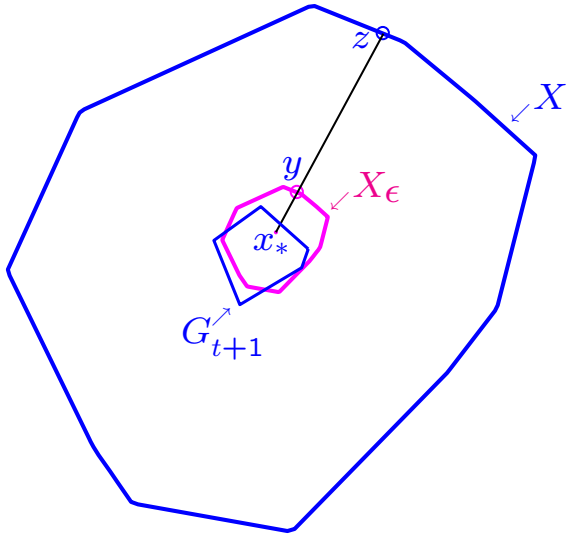
• Let us fix $\epsilon \in (0,1)$ and set $X_\epsilon = x_* + \epsilon(X - x_*)$.

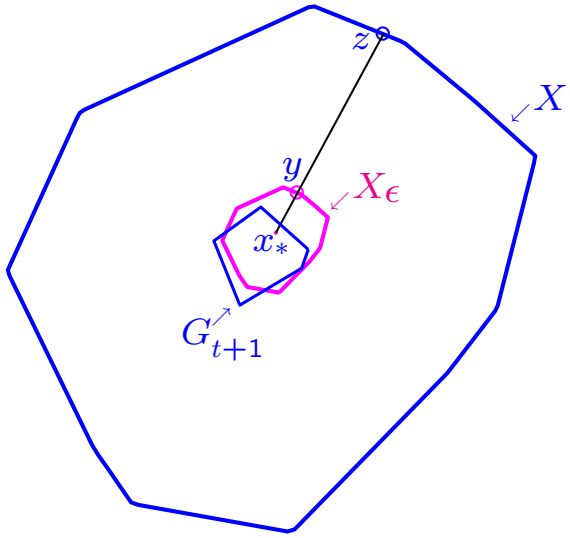
X_ϵ is obtained X by similarity transformation which keeps x_* intact and "shrinks" X towards x_* by factor ϵ. This transformation multiplies volumes by $\epsilon^n \Rightarrow \rho(X_\epsilon) = \epsilon\rho(X)$.

• Let t be such that $\rho(G_{t+1}) < \epsilon\rho(X) = \rho(X_\epsilon)$. Then $\mathrm{Vol}(G_{t+1}) < \mathrm{Vol}(X_\epsilon) \Rightarrow$ *the set* $X_\epsilon \backslash G_{t+1}$ *is nonempty* $\Rightarrow$ *for some* $z \in X$, *the point*

$$y = x_* + \epsilon(z - x_*) = (1-\epsilon)x_* + \epsilon z$$

does not belong to G_{t+1}.





• G_1 contains X and thus y, and G_{t+1} does not contain y, implying that *for some* $\tau \leq t$, *it holds*

$$e_\tau^T y > e_\tau^T c_\tau \qquad (!)$$

• We definitely have $c_\tau \in X$ – otherwise e_τ separates c_τ and $X \ni y$, and (!) witnesses otherwise.

$\Rightarrow c_\tau \in X \Rightarrow e_\tau = f'(c_\tau) \Rightarrow f(c_\tau) + e_\tau^T(y - c_\tau) \leq f(y)$

$\Rightarrow$ [by (!)]

$f(c_\tau) \leq f(y) = f((1-\epsilon)x_* + \epsilon z) \leq (1-\epsilon)f(x_*) + \epsilon f(z)$

$\Rightarrow f(c_\tau) - f(x_*) \leq \epsilon[f(z) - f(x_*)] \leq \epsilon\left[\max_X f - \min_X f\right].$

Bottom line: *If* $0 < \epsilon < 1$ *and* $\rho(G_{t+1}) < \epsilon\rho(X)$, *then* x^t *is well defined* (since $\tau \leq t$ and c_τ is feasible) *and* $f(x^t) - \mathrm{Opt}(P) \leq \epsilon\left[\max_X f - \min_X f\right].$

$$\mathrm{Opt}(P) = \min_{x \in X \subset \mathbb{R}^n} f(x) \qquad (P)$$

"Starting with the first step $\bar{t}$ where $\rho(G_{t+1}) < \rho(X)$, x^t is well defined, and

$$f(x^t) - \mathrm{Opt} \leq \underbrace{\min_{\tau \leq t} \left[\frac{\rho(G_{\tau+1})}{\rho(X)}\right]}_{\epsilon_t} \underbrace{\left[\max_X f - \min_X f\right]}_{V}"$$

♣ We are done. Let $t \geq \bar{t}$, so that $\epsilon_t < 1$, and let $\epsilon \in (\epsilon_t, 1)$. Then for some $t' \leq t$ we have

$$\rho(G_{t'+1}) < \epsilon \rho(X)$$

$\Rightarrow$ [by bottom line] *$x^{t'}$ is well defined and*

$$f(x^{t'}) - \mathrm{Opt}(P) \leq \epsilon V$$

$\Rightarrow$ [since $f(x^t) \leq f(x^{t'})$ due to $t \geq t'$] *x^t is well defined and $f(x^t) - \mathrm{Opt}(P) \leq \epsilon V$*

$\Rightarrow$ [passing to limit as $\epsilon \to \epsilon_t + 0$] *$x^t$ is well defined and $f(x^t) - \mathrm{Opt}(P) \leq \epsilon_t V$*

□

$$\mathrm{Opt}(P) = \min_{x \in X \subset \mathbb{R}^n} f(x) \qquad (P)$$

♠ **Corollary:** *Let (P) be solved by cutting Plane Algorithm which ensures, for some $\vartheta \in (0,1)$, that $\rho(G_{t+1}) \leq \vartheta \rho(G_t)$. Then, for every desired accuracy $\epsilon > 0$, finding feasible ϵ-optimal solution x_ϵ to (P)* (i.e., a feasible solution x_ϵ satisfying $f(x_\epsilon) - \mathrm{Opt} \leq \epsilon$) *takes at most*

$$N = \frac{1}{\ln(1/\vartheta)} \ln\left(\mathcal{R}\left[1 + \frac{V}{\epsilon}\right]\right) + 1$$

steps of the algorithm. Here

$$\mathcal{R} = \frac{\rho(G_1)}{\rho(X)}$$

says how well, in terms of volume, the initial localizer G_1 approximates X, and

$$V = \max_X f - \min_X f$$

is the variation of f on X.

Note: $\mathcal{R}$ and V/ϵ are under log, implying that high accuracy and poor approximation of X by G_1 cost "nearly nothing."

What matters, is the factor *at the log* which is the larger the closer $\vartheta < 1$ is to 1.

♠ In high dimensions, to ensure progress in volumes of subsequent localizers in a Cutting Plane algorithm is not an easy task: we do *not* know how the cut through c_t will pass, and thus should select c_t in G_t in such a way that *whatever be the cut*, it cuts off the current localizer G_t a "meaningful" part of its volume.

The difficulty with achieving this task stems from counterintuitive behaviour of high-dimensional volumes.

Illustration I: Let us model n-dimensional watermelon as unit Euclidean ball in $\mathbb{R}^n$; the concentric ball of radius 0.99 is the flesh, and the remaining "spherical layer" is the rind.

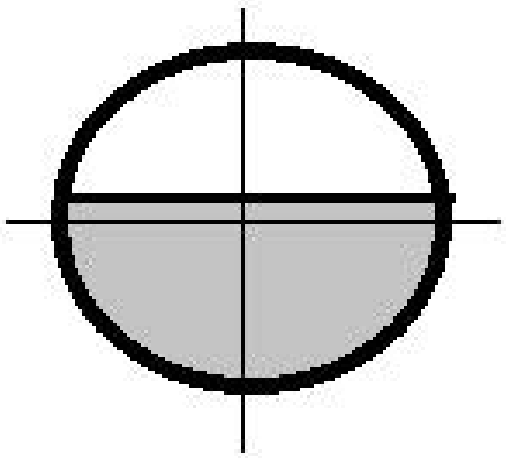
Question: Which portion of watermelon's volume is its flesh?

n	3	10	100	500	1000
$\frac{\mathrm{Vol(flesh)}}{\mathrm{Vol(watermellon)}}$	0.9703	0.9044	0.3660	0.0066	4.3e−5

Conclusion: *Large, in terms of linear sizes, parts of high-dimensional domains may be "invisible" in terms of their volume!*

Illustration 2: Current localizer G_t is unit n-dimensional Euclidean ball. Were we selecting as c_t its center, the origin, the reduction in volume, whatever the cut, is by factor 1/2.

Question: *What will be the reduction in volume when* $c_t = [0; ...; 0; 0.1]$ *and* $\widehat{G}_t = \{x \in G_t : x_n \leq 0.1\}$ *?*



n	32	64	128	256	512	1024	2048
$\frac{\mathrm{Vol}(\widehat{G}_t)}{\mathrm{Vol}(G_t)}$	0.7162	0.7896	0.8721	0.9458	0.9884	0.9993	1.0000

Conclusion: *In order to get a good cutting plane scheme, one needs to be very accurate with the choice of search points!*

"Academic" Implementation: Centers of Gravity

♠ *The most natural choice of c_t in G_t is the center of gravity:*

$$c_t = \left[\int_{G_t} x dx\right] / \left[\int_{G_t} 1 dx\right],$$

the expectation of the random vector uniformly distributed on G_t.

Good news: *The Center of Gravity policy with $G_{t+1} = \widehat{G}_t$ results in*

$$\vartheta = \left(1 - \left[\tfrac{n}{n+1}\right]^n\right)^{1/n} \leq [0.632...]^{1/n} \qquad (*)$$

This results in the complexity bound (# of steps needed to build ϵ-solution)

$$N = 2.2n \ln\left(\mathcal{R}\left[1 + \tfrac{V}{\epsilon}\right]\right) + 1$$

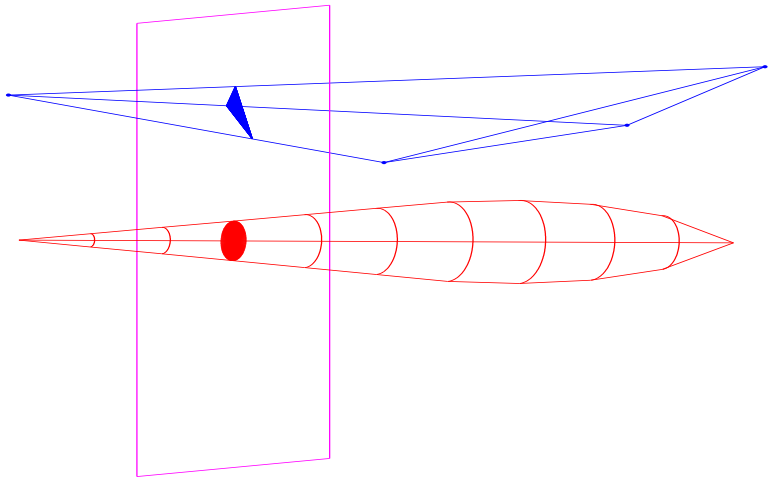
Note: *It can be proved that within absolute constant factor, like 4, this is the best complexity bound achievable by whatever algorithm for convex minimization which can "learn" the objective via First Order oracle only.*

♣ Reason for (*): Brunn-Minkowski Symmeterization Principle:

Let Y be a convex compact set in $\mathbb{R}^n$, e be a unit direction and Z be "equi-cross-sectional" to X body symmetric w.r.t. e, so that

- *Z is rotationally symmetric w.r.t. the axis e*
- *for every hyperplane $H = \{x : e^T x = \text{const}\}$, one has*

$$\text{Vol}_{n-1}(X \cap H) = \text{Vol}_{n-1}(Z \cap H)$$



Then Z is a convex compact set.

Equivalently: *Let U, V be convex compact nonempty sets in $\mathbb{R}^n$. Then*

$$\text{Vol}^{1/n}(U+V) \geq \text{Vol}^{1/n}(U) + \text{Vol}^{1/n}(V).$$

In fact, convexity of U, V is redundant!

Disastrously bad news: *Centers of Gravity are not implementable, unless the dimension n of the problem is like 2 or 3.*

Reason: *We have no control on the shape of localizers. When started with a polytope G_1 given by M linear inequalities (e.g., a box), G_t for $t \gg n$ can be a more or less arbitrary polytope given by $M + t - 1$ linear inequalities. Computing center of gravity of a general-type high-dimensional polytope is a computationally intractable task – it requires astronomically many computations already in the dimensions like 5 – 10.*

Remedy: *Maintain the shape of G_t simple and convenient for computing centers of gravity, sacrificing, if necessary, the value of ϑ.*

The most natural implementation of this remedy is enforcing G_t to be ellipsoids. As a result,

- *c_t becomes computable in $O(n^2)$ operations (nice!)*
- *$\vartheta = [0.632...]^{1/n} \approx \exp\{-0.367/n\}$ increases to $\vartheta \approx \exp\{-0.5/n^2\}$, spoiling the complexity bound*

$$N = 2.2n \ln\left(\mathcal{R}\left[1 + \frac{V}{\epsilon}\right]\right) + 1$$

to

$$N = 4n^2 \ln\left(\mathcal{R}\left[1 + \frac{V}{\epsilon}\right]\right) + 1$$

(unpleasant, but survivable...)

Practical Implementation - Ellipsoid Method

♠ *Ellipsoid in $\mathbb{R}^n$ is the image of the unit n-dimensional ball under one-to-one affine mapping:*

$$E = E(B, c) = \{x = Bu + c : u^T u \leq 1\}$$

where B is $n \times n$ nonsingular matrix, and $c \in \mathbb{R}^n$.

- *c is the center of ellipsoid $E = E(B, c)$: when $c + h \in E$, $c - h \in E$ as well*
- *When multiplying by $n \times n$ matrix B, n-dimensional volumes are multiplied by $|\mathrm{Det}(B)|$*

$\Rightarrow \mathrm{Vol}(E(B, c)) = |\mathrm{Det}(B)|$, $\rho(E(B, c)) = |\mathrm{Det}(B)|^{1/n}$.

$$E = E(B,c) = \{x = Bu + c : u^T u \le 1\}$$

Simple fact: *Let $E(B,c)$ be ellipsoid in $\mathbb{R}^n$ and $e \in \mathbb{R}^n$ be a nonzero vector. The "half-ellipsoid"*

$$\widehat{E} = \{x \in E(B,c) : e^T x \le e^T c\}$$

is covered by the ellipsoid $E^+ = E(B^+, c^+)$ given by

$$\begin{array}{rcl} c^+ & = & c - \frac{1}{n+1} Bp, \; p = B^T e / \sqrt{e^T B B^T e} \\ B^+ & = & \frac{n}{\sqrt{n^2-1}} B + \left(\frac{n}{n+1} - \frac{n}{\sqrt{n^2-1}} \right) (Bp) p^T, \end{array}$$

- *$E(B^+, c^+)$ is the ellipsoid of the smallest volume containing the half-ellipsoid $\widehat{E}$, and the volume of $E(B^+, c^+)$ is strictly smaller than the one of $E(B,c)$:*

$$\vartheta := \frac{\rho(E(B^+,c^+))}{\rho(E(B,c))} \le \exp\{-\frac{1}{2n^2}\}.$$

- *Given B, c, e, computing B^+, c^+ costs $O(n^2)$ arithmetic operations.*

$$\mathrm{Opt}(P) = \min_{x \in X \subset \mathbb{R}^n} f(x) \qquad (P)$$

♣ **Ellipsoid method** *is the Cutting Plane Algorithm where*

- *all localizers G_t are ellipsoids:*

$$G_t = E(B_t, c_t),$$

- *the search point at step t is c_t, and*
- *G_{t+1} is the smallest volume ellipsoid containing the half-ellipsoid*

$$\widehat{G}_t = \{x \in G_t : e_t^T x \le e_t^T c_t\}$$

Computationally, *at every step of the algorithm we once call the Separation oracle Sep_X, (at most) once call the First Order oracle $\mathcal{O}_f$ and spend $O(n^2)$ operations to update (B_t, c_t) into (B_{t+1}, c_{t+1}) by explicit formulas.*

♠ **Complexity bound** *of the Ellipsoid algorithm is*

$$N = 4n^2 \ln\left(\mathcal{R}\left[1 + \tfrac{V}{\epsilon}\right]\right) + 1$$
$$\mathcal{R} = \tfrac{\rho(G_1)}{\rho(X)} \le \tfrac{R}{r}, \; V = \max_{x \in X} f(x) - \min_{x \in X} f(x)$$

Pay attention:

- *$\mathcal{R}, V, \epsilon$ are under log $\Rightarrow$ large magnitudes in data entries and high accuracy are not issues*
- *the factor at the log depends only on the structural parameter of the problem (its design dimension n) and is independent of the remaining data.*

What is Inside Simple Fact

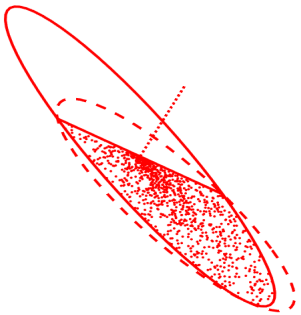
♠ *Messy formulas describing the updating*

$$(B_t, c_t) \to (B_{t+1}, c_{t+1})$$

in fact are easy to get.

• *Ellipsoid E is the image of the unit ball U under affine transformation. Affine transformation preserves ratio of volumes*

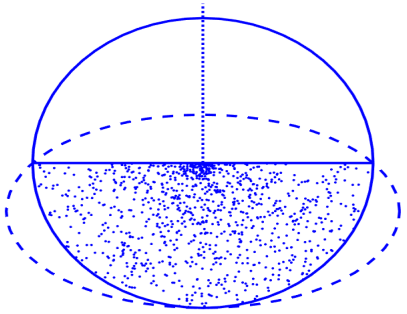
$\Rightarrow$ *Finding the smallest volume ellipsoid containing a given half-ellipsoid $\widehat{E}$ reduces to finding the smallest volume ellipsoid U^+ containing half-ball $\widehat{U}$:*



E, $\widehat{E}$ and E^+

$\Leftrightarrow$

$x = c + Bu$



U, $\widehat{U}$ and U^+

• *The "ball" problem is highly symmetric, and solving it reduces to a simple exercise in elementary Calculus.*

Why Ellipsoids?

(?) *When enforcing the localizers to be of "simple and stable" shape, why we make them ellipsoids (i.e., affine images of the unit Euclidean ball), and not something else, say parallelotopes (affine images of the unit box)?*

Answer: *In a "simple stable shape" version of Cutting Plane Scheme all localizers are affine images of some fixed n-dimensional solid $\mathbf{C}$ (closed and bounded convex set in $\mathbb{R}^n$ with a nonempty interior). To allow for reducing step by step volumes of localizers, $\mathbf{C}$ cannot be arbitrary. What we need is the following property of $\mathbf{C}$:*

One can fix a point $\mathbf{c}$ in $\mathbf{C}$ in such a way that whatever be a cut

$$\widehat{\mathbf{C}} = \{x \in \mathbf{C} : e^T x \leq e^T \mathbf{c}\} \quad [e \neq 0]$$

this cut can be covered by the affine image of $\mathbf{C}$ with the volume less than the one of $\mathbf{C}$:

$$\exists B, b : \widehat{\mathbf{C}} \subset B\mathbf{C} + b \;\&\; |\mathrm{Det}(B)| < 1 \qquad (!)$$

Note: *The Ellipsoid method corresponds to unit Euclidean ball in the role of $\mathbf{C}$ and to $\mathbf{c} = 0$, which allows to satisfy (!) with $|\mathrm{Det}(B)| \leq \exp\{-\frac{1}{2n}\}$, finally yielding $\vartheta \leq \exp\{-\frac{1}{2n^2}\}$.*

• *Solids* $\mathbf{C}$ *with the above property are "rare commodity." For example,* n*-dimensional box does not possess it.*

• *Another "good" solid is* n*-dimensional simplex (this is not that easy to see!). Here* (!) *can be satisfied with* $|\mathrm{Det}(B)| \leq \exp\{-O(1/n^2)\}$*, finally yielding* $\vartheta = (1 - O(1/n^3))$*.*

$\Rightarrow$ *From the complexity viewpoint, "simplex" Cutting Plane algorithm is worse than the Ellipsoid method.*

The same is true for handful of other known so far (and quite exotic) "good solids."

Ellipsoid Method: pro's & con's

♣ **Academically speaking,** *Ellipsoid method is an indispensable tool underlying basically all results on efficient solvability of generic convex problems*, most notably, the famous theorem of L. Khachiyan (1978) on *polynomial time solvability of Linear Programming with rational data in Rational Arithmetic Complexity model.*

♠ *What matters from theoretical perspective,* is "universality" of the algorithm (nearly no assumptions on the problem except for convexity) and complexity bound of the form *"structural parameter outside of log, all else, including required accuracy, under the log."*

♠ Another theoretical (and to some extent, also practical) advantage of the Ellipsoid algorithm is that *as far as the representation of the feasible set X is concerned, all we need is a Separation oracle, and not the list of constraints describing X.* The number of these constraints can be astronomically large, making impossible to check feasibility by looking at the constraints one by one; however, in many important situations the constraints are "well organized," allowing to implement Separation oracle efficiently.

♠ Theoretically, the only (and minor!) drawbacks of the algorithm is the necessity for the feasible set X to be bounded, with known "upper bound," and to possess nonempty interior.

As of now, there is not way to cure the first drawback without sacrificing universality. The second "drawback" is artifact: given nonempty set

$$X = \{x : g_i(x) \leq 0, 1 \leq i \leq m\},$$

we can extend it to

$$X^\epsilon = \{x : g_i(x) \leq \epsilon, 1 \leq i \leq m\},$$

thus making the interior nonempty, and minimize the objective within accuracy ϵ on this larger set, seeking for ϵ-optimal ϵ-feasible solution instead of ϵ-optimal and *exactly feasible* one.

This is quite natural: to find a feasible solution is, in general, not easier than to find an optimal one. Thus, *either ask for exactly feasible and exactly optimal solution* (which beyond LO is unrealistic), or allow for controlled violation in *both* feasibility and optimality!

♠ **From practical perspective**, theoretical drawbacks of the Ellipsoid method become irrelevant: for all practical purposes, bounds on the magnitude of variables like 10^{100} are the same as no bounds at all, and infeasibility like 10^{-10} is the same as feasibility. And since the bounds on the variables and the infeasibility are under log in the complexity estimate, 10^{100} and 10^{-10} are not a disaster.

♠ **Practical limitations** (rather severe!) of Ellipsoid algorithm stem from method's sensitivity to problem's design dimension n. Theoretically, with $\epsilon, V, \mathcal{R}$ fixed, the number of steps grows with n as n^2, and the effort per step is *at least* $O(n^2)$ a.o.

$\Rightarrow$ *Theoretically, computational effort grows with n at least as $O(n^4)$,*

$\Rightarrow$ *n like 1000 and more is beyond the "practical grasp" of the algorithm.*

Note: *Nearly all modern applications of Convex Optimization deal with n in the range of tens and hundreds of thousands!*

♠ By itself, growth of *theoretical* complexity with n as n^4 is not a big deal: for Simplex method, this growth is exponential rather than polynomial, and nobody dies – in reality, Simplex does *not* work according to its disastrous theoretical complexity bound.

Ellipsoid algorithm, unfortunately, works more or less according to its complexity bound.

⇒ *Practical scope of Ellipsoid algorithm is restricted to convex problems with few tens of variables.*

However: Low-dimensional convex problems from time to time do arise in applications. More importantly, these problems arise "on a permanent basis" as auxiliary problems within some modern algorithms aimed at solving *extremely large-scale* convex problems.

⇒ *The scope of practical applications of Ellipsoid algorithm is nonempty, and within this scope, the algorithm, due to its ability to produce high-accuracy solutions* (and surprising stability to rounding errors) *can be considered as the method of choice.*

How It Works

$$\text{Opt} = \min_x f(x),\ X = \{x \in \mathbb{R}^n : a_i^T x - b_i \le 0,\ 1 \le i \le m\}$$

♠ Real-life problem with $n = 10$ variables and $m = 81,963,927$ "well-organized" linear constraints:

CPU, sec	t	$f(x^t)$	$f(x^t) - \text{Opt} \le$	$\rho(G_t)/\rho(G_1)$
0.01	1	0.000000	6.7e4	1.0e0
0.53	63	0.000000	6.7e3	4.2e-1
0.60	176	0.000000	6.7e2	8.9e-2
0.61	280	0.000000	6.6e1	1.5e-2
0.63	436	0.000000	6.6e0	2.5e-3
1.17	895	-1.615642	6.3e-1	4.2e-5
1.45	1250	-1.983631	6.1e-2	4.7e-6
1.68	1628	-2.020759	5.9e-3	4.5e-7
1.88	1992	-2.024579	5.9e-4	4.5e-8
2.08	2364	-2.024957	5.9e-5	4.5e-9
2.42	2755	-2.024996	5.7e-6	4.1e-10
2.66	3033	-2.024999	9.4e-7	7.6e-11

♠ Similar problem with $n = 30$ variables and $m = 1,462,753,730$ "well-organized" linear constraints:

CPU, sec	t	$f(x^t)$	$f(x^t) - \text{Opt} \leq$	$\rho(G_t)/\rho(G_1)$
0.02	1	0.000000	5.9e5	1.0e0
1.56	649	0.000000	5.9e4	5.0e-1
1.95	2258	0.000000	5.9e3	8.1e-2
2.23	4130	0.000000	5.9e2	8.5e-3
5.28	7080	-19.044887	5.9e1	8.6e-4
10.13	10100	-46.339639	5.7e0	1.1e-4
15.42	13308	-49.683777	5.6e-1	1.1e-5
19.65	16627	-50.034527	5.5e-2	1.0e-6
25.12	19817	-50.071008	5.4e-3	1.1e-7
31.03	23040	-50.074601	5.4e-4	1.1e-8
37.84	26434	-50.074959	5.4e-5	1.0e-9
45.61	29447	-50.074996	5.3e-6	1.2e-10
52.35	31983	-50.074999	1.0e-6	2.0e-11

From Ellipsoid Method to Polynomial Time Solvability of Linear Programming

♣ **Theorem** [L. Khachiyan, 1978] *A Linear Programming problem*

$$\min\left\{c^T x : Ax \leq b\right\}$$

with rational data admits polynomial time solution algorithm: an optimal solution to the problem (or a correct conclusion that no solution exists) can be found in polynomial time, that is, in number of bitwise arithmetic operations polynomial in the bitlength L (total number of bits) in the data.

♠ **Main Lemma:** *Given a system $Ax \leq b$ of linear inequalities with rational data, one can decide whether or not the system is solvable in polynomial time.*

♠ **Proof of Main Lemma.** Eliminating from A columns which are linear combinations of the remaining columns does not affect solvability of the system $Ax \leq b$, and selecting the maximal linearly independent set of columns in A is a simple Linear Algebra problem which can be solved in polynomial time.

$\Rightarrow$ *We may assume without loss of generality that the columns of A are linearly independent*, or, which is the same, that *the solution set does not contain lines.* By similar argument, *we may assume without loss of generality that the data are integer*.

Step 1: Reformulating the problem. Assuming that A is $m \times n$, observe that system $Ax \leq b$ is feasible if and only if the optimal value Opt in the optimization problem

$$\text{Opt} = \min_{x \in \mathbb{R}^n} \left\{ f(x) = \max_{1 \leq i \leq m} [Ax - b]_i \right\} \qquad (*)$$

is nonpositive.

Strategy: $(*)$ is a convex minimization problem with easy to compute objective

$\Rightarrow$ *We could try to check whether or not* Opt ≤ 0 *by solving the problem by the Ellipsoid method.*

Immediate obstacles:

- The domain of $(*)$ is the entire space, while the Ellipsoid method requires the domain to be bounded.
- The Ellipsoid method allows to find high accuracy *approximate* solutions, while we need to distinguish between the cases Opt ≤ 0 and Opt > 0, which seems to require finding *precise* solution.

Removing the obstacles, I

$$Ax \le b \text{ is feasible}$$

$$\Leftrightarrow \quad \mathrm{Opt} = \inf_{x\in\mathbb{R}^n}\left\{f(x) = \max_{1\le i\le m}[Ax-b]_i\right\} \le 0 \qquad (*)$$

♠ **Fact I:** $\mathrm{Opt} \le 0$ *is and only if*

$$\mathrm{Opt}_* = \min_x\left\{f(x) = \max_{1\le i\le m}[Ax-b]_i : \|x\|_\infty \le 2^L\right\} \le 0 \ (!)$$

Indeed, the polyhedral set $\{x : Ax \le b\}$ does not contain lines and therefore is nonempty if and only if the set possesses an extreme point $\bar{x}$.

A. By characterization of extreme points of polyhedral sets, we should have $\bar{A}\bar{x} = \bar{b}$ where $\bar{A}$ is a *nonsingular* $n \times n$ submatrix of the $m \times n$ matrix A, and $\bar{b}$ is the respective subvector of b.

$\Rightarrow$ by Cramer's rule, $x_j = \frac{\Delta_j}{\Delta}$, where $\Delta \ne 0$ is the determinant of $\bar{A}$ and Δ_j is the determinant of the matrix $\bar{A}^j$ obtained from $\bar{A}$ when replacing j-th column with $\bar{b}$.

B. Since $\bar{A}$ is with integer entries, its determinant is integer; since it is nonzero, we have $|\Delta| \ge 1$.

Since $\bar{A}^j$ is with integer entries of total bit length $\le L$, the magnitude of its determinant is at most 2^L (an immediate corollary of the definition of the total bit length and the *Hadamard's Inequality* stating that *the magnitude of a determinant does not exceed the product of Euclidean lengths of its rows).*

Combining **A** and **B**, we get $|\bar{x}_j| \le 2^L$ for all j. □

Removing the obstacles, II

$$Ax \le b \text{ is feasible}$$

$$\Leftrightarrow \quad \mathrm{Opt} = \inf_{x\in\mathbb{R}^n}\left\{\max_{1\le i\le m}[Ax-b]_i\right\} \le 0 \qquad (*)$$

$$\Leftrightarrow \quad \mathrm{Opt}_* := \min_x\left\{\max_{1\le i\le m}[Ax-b]_i : \|x\|_\infty \le 2^L\right\} \le 0 \quad (!)$$

♠ **Fact II:** *Let A, b be with integer entries of the total bit length L and the columns of A be linearly independent. If* Opt *is positive,* Opt_* *is not too small, specifically,* $\mathrm{Opt}_* \ge 2^{-2L}$.

Indeed, assume that $\mathrm{Opt} > 0$. Note that $\mathrm{Opt}_* \ge \mathrm{Opt}$ and that

$$\mathrm{Opt} = \min_{t,x}\{t : Ax - t\mathbf{1} \le b\} > 0 \qquad [\mathbf{1} = [1;...;1]]$$

It is immediately seen that when Opt is positive, the feasible domain of this (clearly feasible) LP does not contain lines, and thus the problem has an extreme point solution. $\Rightarrow$ Opt_* is just a coordinate in an extreme point $\bar{y}$ of the polyhedral set $\{y = [x;t] : A_+y \le b\}$, $A_+ = [A, -\mathbf{1}]$. Note that the bit length of (A_+, b) is at most $2L$.

$\Rightarrow$ [the same argument as in Fact I] $\mathrm{Opt} = \frac{\Delta'}{\Delta''}$ with integer $\Delta'' \ne 0, \Delta'$ of magnitudes not exceeding 2^{2L}. Since in addition $\mathrm{Opt} > 0$, we conclude that $\mathrm{Opt} > 2^{-2L}$, as claimed.

♠ **Bottom line:** *When A, b are with integer data of total bit length L and the columns in A are linearly independent, checking whether the system $Ax \leq b$ is solvable reduces to deciding on two hypotheses about*

$$\mathrm{Opt}_* := \min_x \left\{ \max_{1 \leq i \leq m} [Ax - b]_i : \|x\|_\infty \leq 2^L \right\} \qquad (!)$$

the first stating that $\mathrm{Opt}_* \leq 0$, and the second stating that $\mathrm{Opt}_* \geq 2^{-2L}$.

• To decide correctly on the hypotheses, it clearly suffices to approximate Opt_* within accuracy $\epsilon = \frac{1}{3}2^{-2L}$.

Invoking the efficiency estimate of the Ellipsoid method and taking into account that by evident reasons $n \leq L$, it is immediately seen that *resolving the resulting task requires polynomial in L number of arithmetic operations*, including those to mimic Separation and First Order oracles.

However: The Ellipsoid algorithm uses precise real arithmetics, while we want to check feasibility in polynomial in L number of *bitwise* operations. What to do?

• Straightforward (albeit tedious) analysis shows that *we lose nothing when replacing the precise real arithmetic with imprecise one, where one keeps $O(nL)$ digits in the results before and after the dot.* With this implementation, the procedure becomes "fully finite" and requires *polynomial in L* number of bitwise operations. □

From Checking Feasibility to Finding Solution

♠ **Note:** Solving LP with rational data of bitlength L reduces to solving system of linear inequalities with rational data of bitlength $O(L)$ (write down the primal and the dual constraints and add the inequality "duality gap is ≤ 0")

$\Rightarrow$ The only thing which is still missing is how to reduce, in a polynomial time fashion, *finding a solution*, if any, *to a system of linear inequalities with rational data* to *checking feasibility* of systems of linear inequalities with rational data.

♣ *How to reduce in a polynomial time fashion finding a solution to checking feasibility?*

♠ **Reduction:** To find a solution, if any, to a system $\mathcal{S}$ of m linear inequalities and equalities with rational data, we

• Check in polynomial time whether $\mathcal{S}$ is solvable. If not, we are done, otherwise we proceed as follows.

• We convert the first inequality in $\mathcal{S}$, if any, into equality and check in polynomial time whether the resulting system is solvable. If yes, this is our new system $\mathcal{S}_1$, otherwise $\mathcal{S}_1$ is obtained from $\mathcal{S}$ by eliminating the first inequality.

Note: As it is immediately seen, *$\mathcal{S}_1$ solvable, and every feasible solution to $\mathcal{S}_1$ is feasible for $\mathcal{S}$.* Thus,

• *Given a solvable system of m linear inequalities and equalities, we can in polynomial time replace it with another solvable system of (at most) m linear inequalities and equalities, strictly reducing the number of inequalities, provided it was positive, and ensuring that every feasible solution to $\mathcal{S}_1$ is feasible for $\mathcal{S}$.* Besides this, *the bitlength of the data of $\mathcal{S}_1$ is (at most) the total bitlength L of the data of $\mathcal{S}$.*

• Iterating the above construction, we in at most m steps end up with a *solvable* system $\mathcal{S}^*$ of linear *equations* such that *every feasible solution to* $\mathcal{S}^*$ *is feasible for the original system* $\mathcal{S}$.

$\Rightarrow$ Finding a solution to a system $\mathcal{S}$ of linear inequalities and equations with rational data indeed reduces in a polynomial time fashion to polynomially solvable, via elementary Linear Algebra, problem of solving a system of linear *equations* with rational data.

Lecture III.2
From Linear to Conic Programming

What is Ahead

♣ The theorem on polynomial time solvability of Linear Optimization is "constructive" – we can explicitly point out the underlying polynomial time solution algorithm (e.g., the Ellipsoid method). However, from the practical viewpoint this is a kind of "existence theorem" – the resulting complexity bounds, although polynomial, are "too large" for practical large-scale computations.

The intrinsic drawback of the Ellipsoid method (and all other "universal" polynomial time methods in Convex Programming) is that the method utilizes just the convex structure of instances and is unable to facilitate our a priori knowledge of the particular analytic structure of these instances.

• In late 80's, a new family of polynomial time methods for "well-structured" generic convex programs was found – the *Interior Point* methods which indeed are able to facilitate our knowledge of the analytical structure of instances.

• $\mathcal{LO}$ and its extensions – *Conic Quadratic Optimization* $\mathcal{CQO}$ and *Semidefinite Optimization* $\mathcal{SDO}$ – are especially well-suited for processing by the IP methods, and these methods yield the best known so far theoretical complexity bounds for the indicated generic problems.

♣ As far as practical computations are concerned, the IP methods

• in the case of Linear Optimization, are competitive with the Simplex method

• in the case of Conic Quadratic and Semidefinite Optimization are the best known so far numerical techniques.

From Linear to Conic Optimization

Conic Optimization: Why?

♠ "Universal" Convex Optimization algorithm, like Ellipsoids method, are *blind* (scientifically: "black box oriented") – they do *not* utilize problem's structure, aside of convexity, and "learn" the problem via local information (values and (sub)gradients of objective and constraints along search points). At present level of our knowledge, this implies severe limitations on the sizes of convex problems amenable to "universal" algorithms.

Note: A convex program *always* has a lot of structure – otherwise how could we know that the problem is convex?

A good algorithm should utilize a priori knowledge of problem's structure in order to accelerate the solution process.

A good algorithm should utilize a priori knowledge of problem's structure in order to accelerate the solution process.

Example: The LP Simplex Method is fully adjusted to the particular structure of an LO problem. Although *by far* inferior to the Ellipsoid method *in the worst case*, Simplex Method in reality is capable to solve LO's with tens and hundreds of thousands of variables and constraints – a task which is by far out of reach of the theoretically efficient "universal" black box oriented algorithms.

♠ Before utilizing structure of a convex program, one should "reveal" it. Revealing structure is a highly challenging task: *it is unclear what we are looking for until we find it!*

♠ The most useful, as of now, "structure revealing" form of convex program – *Conic Optimization* – was found in early 1990's. The idea behind looks really striking (if not crazy):

- Traditionally, when passing from a LO problem

$$\min\{c^T x : Ax - b \leq 0\} \qquad (P)$$

to a convex one,

- linear objective $c^T x$ is replaced with convex objective, and
- affine in x left hand side $Ax - b$ in the vector inequality constraint $Ax - b \leq 0$ is replaced with entrywise convex vector-valued function $A(x)$, yielding the vector inequality constraint $A(x) \leq 0$. *In Conic Optimization, we keep the objective and the left hand side in the vector inequality $Ax - b \leq 0$ linear/affine, and "introduce nonlinearity" in what "≤ 0" means!*

Note: This is not as crazy as it looks. *When comparing numbers*, there is only one meaningful notion of $\leq$. Inequality $\leq$ in (P) is something different: it is specific "entrywise" inequality between *vectors*, with "$a \leq 0$" meaning "all entries in vector a are nonpositive." On a closed inspection, *the entrywise vector inequality "$\leq$" is neither the only possible, nor the only useful way to compare vectors,* so why to stick to the entrywise $\leq$?

♣ A *Conic Programming* optimization program is

$$\mathrm{Opt} = \min_x \left\{ c^T x : Ax - b \in \mathbf{K} \right\}, \qquad (C)$$

where $\mathbf{K} \subset \mathbb{R}^m$ is a *regular* cone.

♠ *Regularity* of $\mathbf{K}$ means that

- $\mathbf{K}$ *is convex cone:*
$$(x_i \in \mathbf{K}, \lambda_i \geq 0, 1 \leq i \leq p) \Rightarrow \textstyle\sum_i \lambda_i x_i \in \mathbf{K}$$
- $\mathbf{K}$ *is pointed:* $\pm a \in \mathbf{K} \Leftrightarrow a = 0$
- $\mathbf{K}$ *is closed:* $x_i \in \mathbf{K}, \lim_{i\to\infty} x_i = x \Rightarrow x \in \mathbf{K}$
- $\mathbf{K}$ *has a nonempty interior* int $\mathbf{K}$*:*
$$\exists(\bar{x} \in \mathbf{K}, r > 0) : \{x : \|x - \bar{x}\|_2 \leq r\} \subset \mathbf{K}$$

Example: The nonnegative orthant

$$\mathbb{R}^m_+ = \{x \in \mathbb{R}^m : x_i \geq 0,\ 1 \leq i \leq m\}$$

is a regular cone, and the associated conic problem (C) is just the usual LO program.

Fact: *When passing from LO programs* (i.e., conic programs associated with nonnegative orthants) *to conic programs associated with properly chosen wider families of cones, we extend dramatically the scope of applications we can process, while preserving the major part of LO theory and preserving our abilities to solve problems efficiently.*

• Let $\mathbf{K} \subset \mathbb{R}^m$ be a regular cone. We can associate with $\mathbf{K}$ two relations between vectors of $\mathbb{R}^m$:

- *"nonstrict $\mathbf{K}$-inequality" $\geq_{\mathbf{K}}$:*

$$a \geq_{\mathbf{K}} b \Leftrightarrow a - b \in \mathbf{K}$$

- *"strict $\mathbf{K}$-inequality" $>_{\mathbf{K}}$:*

$$a >_{\mathbf{K}} b \Leftrightarrow a - b \in \mathrm{int}\,\mathbf{K}$$

Example: when $\mathbf{K} = \mathbb{R}^m_+$, $\geq_{\mathbf{K}}$ is the usual "coordinate-wise" nonstrict inequality "$\geq$" between vectors $a, b \in \mathbb{R}^m$:

$$a \geq b \Leftrightarrow a_i \geq b_i,\ 1 \leq i \leq m$$

while $>_{\mathbf{K}}$ is the usual "coordinate-wise" strict inequality "$>$" between vectors $a, b \in \mathbb{R}^m$:

$$a > b \Leftrightarrow a_i > b_i,\ 1 \leq i \leq m$$

♣ $\mathbf{K}$-inequalities share the basic algebraic and topological properties of the usual coordinate-wise $\geq$ and $>$, for example:

♠ $\geq_{\mathbf{K}}$ is a partial order:

- $a \geq_{\mathbf{K}} a$ (reflexivity),
- $a \geq_{\mathbf{K}} b$ and $b \geq_{\mathbf{K}} a \Rightarrow a = b$ (anti-symmetry)
- $a \geq_{\mathbf{K}} b$ and $b \geq_{\mathbf{K}} c \Rightarrow a \geq_{\mathbf{K}} c$ (transitivity)

♠ $\geq_{\mathbf{K}}$ is compatible with linear operations:

- $a \geq_{\mathbf{K}} b$ and $c \geq_{\mathbf{K}} d \Rightarrow a + c \geq_{\mathbf{K}} b + d$,
- $a \geq_{\mathbf{K}} b$ and $\lambda \geq 0 \Rightarrow \lambda a \geq_{\mathbf{K}} \lambda b$

♠ $\geq_{\mathbf{K}}$ is stable w.r.t. passing to limits:

$$a_i \geq_{\mathbf{K}} b_i, a_i \to a, b_i \to b \text{ as } i \to \infty \Rightarrow a \geq_{\mathbf{K}} b$$

♠ $>_{\mathbf{K}}$ satisfies the usual arithmetic properties, like

- $a >_{\mathbf{K}} b$ and $c \geq_{\mathbf{K}} d \Rightarrow a + c >_{\mathbf{K}} b + d$
- $a >_{\mathbf{K}} b$ and $\lambda > 0 \Rightarrow \lambda a >_{\mathbf{K}} \lambda b$

and is stable w.r.t perturbations: if $a >_{\mathbf{K}} b$, then $a' >_{\mathbf{K}} b'$ whenever a' is close enough to a and b' is close enough to b.

♣ **Note:** *Conic program associated with a regular cone* $\mathbf{K}$ *can be written down as*

$$\min_x \left\{ c^T x : Ax - b \geq_{\mathbf{K}} 0 \right\}$$

Note: Every convex program can be equivalently reformulated as a conic one.

Basic Operations with Cones

♣ *Given regular cones* $\mathbf{K}_\ell \subset \mathbb{R}^{m_\ell}$, $1 \le \ell \le L$, *we can form their direct product*

$$\begin{array}{l}\mathbf{K} = \mathbf{K}_1 \times ... \times \mathbf{K}_L := \{x = [x^1; ...; x^L] : x^\ell \in \mathbf{K}_\ell \, \forall \ell\} \\ \subset \mathbb{R}^{m_1+...+m_L} = \mathbb{R}^{m_1} \times ... \mathbb{R}^{m_L}\end{array},$$

and this direct product is a regular cone.

Example: $\mathbb{R}^m_+$ is the direct product of m nonnegative rays $\mathbb{R}_+ = \mathbb{R}^1_+$.

♣ *Given a regular cone* $\mathbf{K} \in \mathbb{R}^m$, *we can build its dual cone*

$$\mathbf{K}_* = \{x \in \mathbb{R}^m : x^T y \ge 0 \, \forall y \in \mathbf{K}\}$$

The cone $\mathbf{K}_*$ *is regular, and* $(\mathbf{K}_*)_* = \mathbf{K}$.

Example: $\mathbb{R}^m_+$ is self-dual: $(\mathbb{R}^m_+)_* = \mathbb{R}^m_+$.

♣ **Fact:** *The cone dual to a direct product of regular cones is the direct product of the dual cones of the factors:*

$$(\mathbf{K}_1 \times ... \times \mathbf{K}_L)_* = (\mathbf{K}_1)_* \times ... \times (\mathbf{K}_L)_*$$

Data and Structure of Conic Program

$$\min_{x\in\mathbb{R}^n}\left\{c^Tx : Ax-b\geq_{\mathbf{K}}0\right\} \qquad \text{(CP)}$$

♠ When asked "what is the data, and what is the structure in (CP)", everybody will give the same answer:
The structure "sits" in the cone K (and in n), *and the data are the entries in* c, A, b.
But: General type convex cone is as "unstructured" as a general type convex function. Why not to say that in a convex program of in the MP form

$$\min_{x\in\mathbb{R}^n}\{f(x) : g_i(x)\leq 0,\ 1\leq i\leq m\}$$

the structure "sits" in the convex functions $f, g_1, ..., g_m$ (and m, n) — definitely true and absolutely useless!
♠ **Fact:** *Conic problems associated with just three specific families of cones cover nearly all* (for all practical purposes – just all) *applications of Convex Optimization.*
Cones from the three "magic" families possess transparent structure fully utilized by theoretically (and practically!) efficient *Interior Point* methods "tailored" to these cones.
⇒ *Reformulating convex program as a conic program from a "magic family" allows to process the problem by highly efficient dedicated algorithms.*

Linear/Conic Quadratic/Semidefinite Optimization

♣ The three magic families of cones are

• Direct products of nonnegative rays – *nonnegative orthants* giving rise to *Linear Optimization*,

• Direct products of *Lorentz cones* giving rise to *Conic Quadratic Optimization*, a.k.a. *Second Order Cone Optimization*,

• Direct products of *Semidefinite cones* giving rise to *Semidefinite Optimization*.

Linear Optimization

♣ Let $\mathcal{K} = \mathcal{LO}$ be the family of all nonnegative orthants, i.e., all direct products of nonnegative rays. Conic programs associated with cones from $\mathcal{K}$ are exactly the LO programs

$$\min_x \left\{ c^T x : \underbrace{a_i^T x - b_i \geq 0, 1 \leq i \leq m}_{\Leftrightarrow Ax - b \geq_{\mathbb{R}^m_+} 0} \right\}$$

Conic Quadratic Optimization

♣ *Lorentz cone* $\mathbf{L}^m$ of dimension m is the regular cone in $\mathbb{R}^m$ given by

$$\mathbf{L}^m = \{x \in \mathbb{R}^m : x_m \geq \sqrt{x_1^2 + ... + x_{m-1}^2}\}$$

This cone is self-dual.

♠ Let $\mathcal{K} = \mathcal{CQP}$ be the family of all direct products of Lorentz cones. Conic programs associated with cones from $\mathcal{K}$ are called *conic quadratic* programs.

"Mathematical Programming" form of a conic quadratic program is

$$\min_x \left\{c^T x : \underbrace{\|P_i x - p_i\|_2 \leq q_i^T x + r_i}_{\Leftrightarrow [P_i x - p_i; q_i^T x - r_i] \in \mathbf{L}^{m_i}}, 1 \leq i \leq m\right\}$$

Note: According our convention "sum over empty set is 0", $\mathbf{L}^1 = \mathbb{R}_+$ is the nonnegative ray

⇒ *All LO programs are Conic Quadratic ones.*

Semidefinite Optimization

♣ *Semidefinite cone* $\mathbf{S}^m_+$ of order m "lives" in the space $\mathbf{S}^m$ of real symmetric $m \times m$ matrices and is composed of *positive semidefinite* $m \times m$ *matrices*, i.e., symmetric $m \times m$ matrices A such that $d^T A d \geq 0$ for all d.

♡ **Equivalent descriptions of positive semidefiniteness:** *A symmetric* $m \times m$ *matrix* A *is positive semidefinite (notation:* $A \succeq 0$*) if and only if it possesses any one of the following properties:*

- *All eigenvalues of* A *are nonnegative, that is,*

$$A = U\mathrm{Diag}\{\lambda\}U^T$$

with orthogonal U *and nonnegative* λ*.*

Note: *In the representation* $A = U\mathrm{Diag}\{\lambda\}U^T$ *with orthogonal* U*,* $\lambda = \lambda(A)$ *is the vector of eigenvalues of* A *taken with their multiplicities*

- $A = D^T D$ *for a rectangular matrix* D*, or, equivalently,* A *is the sum of dyadic matrices:* $A = \sum_\ell d_\ell d_\ell^T$
- *All principal minors of* A *are nonnegative.*

♡ *The semidefinite cone* $\mathbf{S}^m_+$ *is regular and self-dual*, provided that the inner product on the space $\mathbf{S}^m$ where the cone lives is inherited from the natural embedding $\mathbf{S}^m$ into $\mathbb{R}^{m\times m}$:

$$\forall A, B \in \mathbf{S}^m : \langle A, B\rangle = \sum_{i,j} A_{ij}B_{ij} = \mathrm{Tr}(AB)$$

♠ Let $\mathcal{K} = \mathcal{SDP}$ be the family of all direct products of Semidefinite cones. Conic programs associated with cones from $\mathcal{K}$ are called *semidefinite programs*. Thus, *a semidefinite program* is an optimization program of the form

$$\min_x \left\{ c^T x : \mathcal{A}_i x - B_i := \sum_{j=1}^n x_j A^{ij} - B_i \succeq 0,\ 1 \le i \le m \right\}$$
$$A^{ij}, B_i : \text{ symmetric } k_i \times k_i \text{ matrices}$$

Note: A collection of symmetric matrices $A_1, ..., A_m$ is composed of positive semidefinite matrices iff the block-diagonal matrix $\mathrm{Diag}\{A_1, ..., A_m\}$ is $\succeq 0$

$\Rightarrow$ an SDO program can be written down as a problem with a *single* $\succeq$ constraint (called also a *Linear Matrix Inequality* (LMI)):

$$\min_x \left\{ c^T x : \mathcal{A}x - B := \mathrm{Diag}\{\mathcal{A}_i x - B_i, 1 \le i \le m\} \succeq 0 \right\}.$$

♣ Three generic conic problems – Linear, Conic Quadratic and Semidefinite Optimization — posses intrinsic mathematical similarity allowing for deep unified theoretical and algorithmic developments, including design of theoretically *and practically* efficient polynomial time solution algorithms — *Interior Point Methods*.

♠ At the same time, *"expressive abilities" of Conic Quadratic and especially Semidefinite Optimization are incomparably stronger than those of Linear Optimization.* For all practical purposes, *the entire Convex Programming is within the grasp of Semidefinite Optimization*.

LO/CQO/SDO Hierarchy

♠ $\mathbf{L}^1 = \mathbb{R}_+ \Rightarrow \mathcal{LO} \subset \mathcal{CQO} \Rightarrow$ *Linear Optimization is a particular case of Conic Quadratic Optimization.*

♠ **Fact:** *Conic Quadratic Optimization is a particular case of Semidefinite Optimization.*

♡ **Explanation:** The relation $x \geq_{\mathbf{L}^k} 0$ is equivalent to the relation

$$\mathrm{Arrow}(x) = \left[\begin{array}{c|ccccc} x_k & x_1 & x_2 & \cdots & x_{k-1} \\ \hline x_1 & x_k & & & \\ x_2 & & x_k & & \\ \vdots & & & \ddots & \\ x_{k-1} & & & & x_k \end{array}\right] \succeq 0.$$

As a result, a system of conic quadratic constraints

$$A_i x - b_i \geq_{\mathbf{L}^{k_i}} 0,\ 1 \leq i \leq m$$

is equivalent to the system of LMIs

$$\mathrm{Arrow}(A_i x - b_i) \succeq 0,\ 1 \leq i \leq m.$$

Why

$$x \succeq_{\mathbf{L}^k} 0 \Leftrightarrow \mathrm{Arrow}(x) \succeq 0 \qquad (!)$$

Schur Complement Lemma: *A symmetric block matrix* $\begin{bmatrix} P & Q \\ Q^T & R \end{bmatrix}$ *with positive definite* R *is* $\succeq 0$ *if and only if the matrix* $P - QR^{-1}Q^T$ *is* $\succeq 0$.

Proof. We have

$$\begin{bmatrix} P & Q \\ Q^T & R \end{bmatrix} \succeq 0 \Leftrightarrow [u;v]^T \begin{bmatrix} P & Q \\ Q^T & R \end{bmatrix} [u;v] \geq 0 \, \forall [u;v]$$
$$\Leftrightarrow u^T P u + 2u^T Q v + v^T R v \geq 0 \, \forall [u;v]$$
$$\Leftrightarrow \forall u : u^T P u + \min_v \left\{ 2u^T Q v + v^T R v \right\} \geq 0$$
$$\Leftrightarrow \forall u : u^T P u - u^T Q R^{-1} Q^T u \geq 0$$
$$\Leftrightarrow P - QR^{-1}Q^T \succeq 0 \qquad \square$$

♠ **Schur Complement Lemma $\Rightarrow$ (!):**

• In one direction: Let $x \in \mathbf{L}^k$. Then either $x_k = 0$, whence $x = 0$ and $\mathrm{Arrow}(x) \succeq 0$, or $x_k > 0$ and $\sum_{i=1}^{k-1} \frac{x_i^2}{x_k} \leq x_k$, meaning that the matrix

$$\left[\begin{array}{c|ccc} x_k & x_1 & \cdots & x_{k-1} \\ \hline x_1 & x_k & & \\ \vdots & & \ddots & \\ x_{k-1} & & & x_k \end{array}\right]$$

satisfies the premise of the SCL and thus is $\succeq 0$.

- In another direction: let $\left[\begin{array}{c|ccc} x_k & x_1 & \cdots & x_{k-1} \\ \hline x_1 & x_k & & \\ \vdots & & \ddots & \\ x_{k-1} & & & x_k \end{array}\right] \succeq 0$. Then either $x_k = 0$, and then $x = 0 \in \mathbf{L}^k$, or $x_k > 0$ and $\sum_{i=1}^{k-1} \frac{x_i^2}{x_k} \leq x_k$ by the SCL, whence $x \in \mathbf{L}^k$. $\square$

♣ Example of CQO program: Control of Linear Dynamical system.

Consider a discrete time linear dynamical system given by

$$\begin{aligned} x(0) &= 0; \\ x(t+1) &= Ax(t) + Bu(t) + f(t), 0 \leq t \leq T-1 \end{aligned}$$

- $x(t)$: state at time t
- $u(t)$: control at time t
- $f(t)$: given external input

Goal: Given time horizon T, bounds on control $\|u(t)\|_2 \leq 1$ for all t and desired destination x_*, find a control which makes $x(T)$ as close as possible to x_*.

The model: From state equations,

$$x(T) = \sum_{t=0}^{T-1} A^{T-t-1}[Bu(t) + f(t)],$$

so that the problem in question is

$$\min_{\tau, u(0), \ldots, u(T-1)} \left\{ \tau : \begin{array}{l} \|x_* - \sum_{t=0}^{T-1} A^{T-t-1}[Bu(t) + f(t)]\|_2 \leq \tau \\ \|u(t)\|_2 \leq 1,\ 0 \leq t \leq T-1 \end{array} \right\}$$

♣ **Example of SDO program: Relaxation of a Combinatorial Problem.**

♠ Numerous NP-hard combinatorial problems can be posed as problems of quadratic minimization under quadratic constraints:

$$\begin{array}{c}\mathrm{Opt}(P) = \min\limits_{x} \{f_0(x) : f_i(x) \le 0,\ 1 \le i \le m\} \\ f_i(x) = x^T Q_i x + 2b_i^T x + c_i,\ 0 \le i \le m\end{array} \qquad (P)$$

Example: One can model Boolean constraints $x_i \in \{0;1\}$ as quadratic equality constraints $x_i^2 = x_i$ and then represent them by pairs of quadratic inequalities $x_i^2 - x_i \le 0$ and $-x_i^2 + x_i \le 0$

$\Rightarrow$ *Boolean Programming problems reduce to* (P).

$$\begin{array}{c}\mathrm{Opt}(P) = \min\limits_x \{f_0(x) : f_i(x) \le 0,\ 1 \le i \le m\} \\ f_i(x) = x^T Q_i x + 2b_i^T x + c_i,\ 0 \le i \le m\end{array} \qquad (P)$$

♠ In branch-and-bound algorithms, an important role is played by efficient bounding of $\mathrm{Opt}(P)$ from below. To this end one can use *Semidefinite relaxation* as follows:

• We set $F_i = \left[\begin{array}{c|c} Q_i & b_i \\ \hline b_i^T & c_i \end{array}\right]$, $0 \le i \le m$, and $X[x] = \left[\begin{array}{c|c} xx^T & x \\ \hline x^T & 1 \end{array}\right]$, so that

$$f_i(x) = \mathrm{Tr}(F_i X[x]).$$

$\Rightarrow$ (P) is equivalent to the problem

$$\min_x \{\mathrm{Tr}(F_0 X[x]) : \mathrm{Tr}(F_i X[x]) \le 0,\ 1 \le i \le m\} \qquad (P')$$

$$\begin{array}{l}\mathrm{Opt}(P) = \min\limits_x \{f_0(x) : f_i(x) \le 0,\ 1 \le i \le m\} \\ \quad \left[f_i(x) = x^T Q_i x + 2b_i^T x + c_i,\ 0 \le i \le m\right] \\ \Leftrightarrow \min\limits_x \{\mathrm{Tr}(F_0 X[x]) : \mathrm{Tr}(F_i X[x]) \le 0,\ 1 \le i \le m\} \\ \quad \left[F_i = \left[\begin{array}{c|c} Q_i & b_i \\ \hline b_i^T & c_i \end{array}\right],\ 0 \le i \le m\right]\end{array} \qquad \begin{array}{c}(P) \\ \\ \\ (P')\end{array}$$

• The objective and the constraints in (P') are linear in $X[x]$, and the only difficulty is that as x runs through $\mathbb{R}^n$, $X[x]$ runs through a difficult for minimization manifold $\mathcal{X} \subset \mathbf{S}^{n+1}$ given by the following restrictions:

A. $X \succeq 0$

B. $X_{n+1,n+1} = 1$

C. $\mathrm{Rank}\, X = 1$

• Restrictions **A**, **B** are simple constraints specifying a nice convex domain

• Restriction **C** is the "troublemaker" – it makes the feasible set of (P) difficult

♠ *In SDO relaxation, we just eliminate the rank constraint* **C**, *thus ending up with the SDO program*

$$\mathrm{Opt(SDO)} = \min_{X \in \mathbf{S}^{n+1}} \left\{ \mathrm{Tr}(F_0 X) : \begin{array}{l} \mathrm{Tr}(F_i X) \le 0,\ 1 \le i \le m, \\ X \succeq 0, X_{n+1,n+1} = 1 \end{array} \right\}.$$

♠ *When passing from $(P) \equiv (P')$ to the SDO relaxation, we extend the domain over which we minimize*

$\Rightarrow \mathrm{Opt(SDO)} \le \mathrm{Opt}(P)$.

What Can Be Expressed via $\mathcal{LO}/\mathcal{CQO}/\mathcal{SDO}$?

♣ Consider a family $\mathcal{K}$ of regular cones such that

• $\mathcal{K}$ is closed w.r.t. taking direct products of cones: $\mathbf{K}_1, ..., \mathbf{K}_m \in \mathcal{K} \Rightarrow \mathbf{K}_1 \times ... \times \mathbf{K}_m \in \mathcal{K}$

• $\mathcal{K}$ is closed w.r.t. passing from a cone to its dual: $\mathbf{K} \in \mathcal{K} \Rightarrow \mathbf{K}_* \in \mathcal{K}$

Examples: $\mathcal{LO}$, $\mathcal{CQO}$, $\mathcal{SDO}$.

Question: *When an optimization program*

$$\min_{x \in X} f(x) \qquad (P)$$

can be posed as a conic problem associated with a cone from $\mathcal{K}$?

Answer: This is the case when the set X and the function f are $\mathcal{K}$-*representable*, i.e., admit representations of the form

$$\begin{array}{rcl} X & = & \{x : \exists u : Ax + Bu + c \in \mathbf{K}_X\} \\ \mathrm{Epi}\{f\} & := & \{[x;\tau] : \tau \geq f(x)\} \\ & = & \{[x;\tau] : \exists w : Px + \tau p + Qw + d \in \mathbf{K}_f\} \end{array}$$

where $\mathbf{K}_X \in \mathcal{K}$, $\mathbf{K}_f \in \mathcal{K}$.

Indeed, if

$$\begin{array}{rcl} X & = & \{x : \exists u : Ax + Bu + c \in \mathbf{K}_X\} \\ \mathrm{Epi}\{f\} & := & \{[x;\tau] : \tau \geq f(x)\} \\ & = & \{[x;\tau] : \exists w : Px + \tau p + Qw + d \in \mathbf{K}_f\} \end{array}$$

then problem

$$\min_{x \in X} f(x) \qquad (P)$$

is equivalent to

$$\min_{x,\tau,u,w} \left\{ \tau : \begin{array}{c} \overbrace{Ax + Bu + c \in \mathbf{K}_X}^{\text{says that } x \in X} \\ \underbrace{Px + \tau p + Qw + d \in \mathbf{K}_F}_{\text{says that } \tau \geq f(x)} \end{array} \right\}$$

and the constraints read

$$[Ax + bu + c; Px + \tau p + Qw + d] \in \mathbf{K} := \mathbf{K}_X \times \mathbf{K}_f \in \mathcal{K} \quad .$$

♣ *$\mathcal{K}$-representable sets/functions always are convex.*

♣ *$\mathcal{K}$-representable sets/functions admit fully algorithmic calculus* completely similar to the one we have developed in the particular case $\mathcal{K} = \mathcal{LO}$.

♠ **Example of $\mathcal{CQO}$-representable function:** convex quadratic form

$$f(x) = x^T A^T A x + 2b^T x + c$$

Indeed,

$$\begin{aligned}
&\tau \geq f(x) \Leftrightarrow [\tau - c - 2b^T x] \geq \|Ax\|_2^2 \\
&\Leftrightarrow \left[\tfrac{1+[\tau-c-2b^T x]}{2}\right]^2 - \left[\tfrac{1-[\tau-c-2b^T x]}{2}\right]^2 \geq \|Ax\|_2^2 \\
&\Leftrightarrow \left[Ax; \tfrac{1-[\tau-c-2b^T x]}{2}, \tfrac{1+[\tau-c-2b^T x]}{2}\right] \in \mathbf{L}^{\dim b+2}
\end{aligned}$$

♠ **Examples of $\mathcal{SDO}$-representable functions/sets:**

- the maximal eigenvalue $\lambda_{\max}(X)$ of a symmetric $m \times m$ matrix X:

$$\tau \geq \lambda_{\max}(X) \Leftrightarrow \underbrace{\tau I_m - X \succeq 0}_{\text{LMI}}$$

- the sum of k largest eigenvalues of a symmetric $m \times m$ matrix X
- $-\mathrm{Det}^{1/m}(X)$, $X \in \mathbf{S}^m_+$
- the set P_d of (vectors of coefficients of) nonnegative on a given segment Δ algebraic polynomials $p(x) = p_d x^d + p_{d-1} x^{d-1} + ... + p_1 x + p_0$ of degree $\leq d$.

Conic Duality Theorem

♣ Consider a conic program

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : Ax \geq_{\mathbf{K}} b \right\} \qquad (P)$$

As in the LO case, the concept of the dual problem stems from the desire to find a systematic way to bound from below the optimal value $\mathrm{Opt}(P)$.

♠ In the LO case $\mathbf{K} = \mathbb{R}^m_+$ this mechanism was built as follows:

• We observe that for properly chosen vectors of "aggregation weights" λ (specifically, for $\lambda \in \mathbb{R}^m_+$) the aggregated constraint $\lambda^T Ax \geq \lambda^T b$ is the consequence of the vector inequality $Ax \geq_{\mathbf{K}} b$ and thus $\lambda^T Ax \geq \lambda^T b$ for all feasible solutions x to (P)

• In particular, when admissible vector of aggregation weights λ is such that $A^T\lambda = c$, then the aggregated constraint reads "$c^T x \geq b^T\lambda$ for all feasible x" and thus $b^T\lambda$ is a lower bound on $\mathrm{Opt}(P)$. The dual problem is the problem of maximizing this bound:

$$\mathrm{Opt}(D) = \max_\lambda \{ b^T\lambda : \begin{array}{l} A^T\lambda = c \\ \lambda \text{ is admissible for aggregation} \end{array} \}$$

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : Ax \geq_{\mathbf{K}} b \right\} \qquad (P)$$

♠ The same approach works in the case of a general cone $\mathbf{K}$. The only issue to be resolved is:

What are admissible weight vectors λ for (P)? When a valid vector inequality $a \geq_{\mathbf{K}} b$ always implies the inequality $\lambda^T a \geq \lambda^T b$?

Answer: is immediate: *the required λ's are exactly the vectors from the cone $\mathbf{K}_*$ dual to $\mathbf{K}$.*

Indeed,

- If $\lambda \in \mathbf{K}_*$, then

$$a \geq_{\mathbf{K}} b \Rightarrow a - b \in \mathbf{K} \Rightarrow \lambda^T(a-b) \geq 0 \Rightarrow \lambda^T a \geq \lambda^T b,$$

that is, λ is an admissible weight vector.

- If λ is admissible weight vector and $a \in \mathbf{K}$, that is, $a \geq_{\mathbf{K}} 0$, we should have $\lambda^T a \geq \lambda^T 0 = 0$, so that $\lambda^T a \geq 0$ for all $a \in \mathbf{K}$, i.e., $\lambda \in \mathbf{K}_*$.

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : Ax \geq_{\mathbf{K}} b \right\} \qquad (P)$$

♠ We arrive at the following construction:

• Whenever $\lambda \in \mathbf{K}_*$, the scalar inequality $\lambda^T Ax \geq \lambda^T b$ is a consequence of the constraint in (P) and thus is valid everywhere on the feasible set of (P).

• In particular, when $\lambda \in \mathbf{K}_*$ is such that $A^T\lambda = c$, the quantity $b^T\lambda$ is a lower bound on $\mathrm{Opt}(P)$, and the dual problem is to maximize this bound:

$$\mathrm{Opt}(D) = \max_\lambda \left\{ b^T\lambda : A^T\lambda = c, \lambda \geq_{\mathbf{K}_*} 0 \right\} \qquad (D)$$

As it should be, in the LO case, where $\mathbf{K} = \mathbb{R}^m_+ = (\mathbb{R}^m_+)_* = \mathbf{K}_*$, (D) is nothing but the LP dual of (P).

♣ Our "aggregation mechanism" can be applied to conic problems in a slightly more general format:

$$\mathrm{Opt}(P) = \min_{x \in X} \left\{ c^T x : \begin{array}{l} A_\ell x \geq_{\mathbf{K}^\ell} b_\ell,\ 1 \leq \ell \leq L \\ Px = p \end{array} \right\} \qquad (P)$$

Here the dual problem is built as follows:

- We associate with every vector inequality constraint

$$A_\ell x \geq_{\mathbf{K}^\ell} b_\ell$$

dual variable ("Lagrange multiplier") $\lambda_\ell \geq_{\mathbf{K}^\ell_*} 0$, so that the scalar inequality constraint $\lambda_\ell^T A_\ell x \geq \lambda_\ell^T b_\ell$ is a consequence of $A_\ell x \geq_{\mathbf{K}^\ell} b_\ell$ and $\lambda_\ell \geq_{\mathbf{K}^*_\ell} 0$;
- We associate with the system $Px = p$ a "free" vector μ of Lagrange multipliers of the same dimension as p, so that the scalar inequality $\mu^T Px \geq \mu^T p$ is a consequence of the vector equation $Px = p$;
- We sum up all the scalar inequalities we got, thus arriving at the scalar inequality

$$\left[\sum_{\ell=1}^L A_\ell^T \lambda_\ell + P^T \mu\right]^T x \geq \sum_{\ell=1}^L b_\ell^T \lambda_\ell + p^T \mu \qquad (*)$$

$$\mathrm{Opt}(P) = \min_{x \in X} \left\{ c^T x : \begin{array}{l} A_\ell x \geq_{\mathbf{K}^\ell} b_\ell,\ 1 \leq \ell \leq L \\ Px = p \end{array} \right\} \qquad (P)$$

Whenever x is feasible for (P) and $\lambda_\ell \geq_{\mathbf{K}^\ell_*} 0$, $1 \leq \ell \leq L$, we have

$$\left[\sum_{\ell=1}^L A_\ell^T \lambda_\ell + P^T \mu\right]^T x \geq \sum_{\ell=1}^L b_\ell^T \lambda_\ell + p^T \mu \qquad (*)$$

• If we are lucky to get in the left hand side of $(*)$ the expression $c^T x$, that is, if $\sum_{\ell=1}^L A_\ell^T \lambda_\ell + P^T \mu = c$, then the right hand side of $(*)$ is a lower bound on the objective of (P) everywhere in the feasible domain of (P) and thus is a lower bound on $\mathrm{Opt}(P)$. The dual problem is to maximize this bound:

$$\begin{array}{l} \mathrm{Opt}(D) \\ = \max\limits_{\lambda,\mu} \left\{ \sum_{\ell=1}^L b_\ell^T \lambda_\ell + p^T \mu : \begin{array}{l} \lambda_\ell \geq_{\mathbf{K}^\ell_*} 0,\ 1 \leq \ell \leq L \\ \sum_{\ell=1}^L A_\ell^T \lambda_\ell + P^T \mu = c \end{array} \right\} \end{array} \qquad (D)$$

Note: When all cones $\mathbf{K}^\ell$ are self-dual (as it is the case in Linear/Conic Quadratic/Semidefinite Optimization), the dual problem (D) involves *exactly the same cones* $\mathbf{K}^\ell$ as the primal problem.

Example: Dual of a Semidefinite program.

Consider a Semidefinite program

$$\min_x \left\{ c^T x : \begin{array}{l} \sum_{i=1}^n A_\ell^j x_j \succeq B_\ell, 1 \le \ell \le L \\ Px = p \end{array} \right\}$$

The cones $\mathbf{S}_+^k$ are self-dual, so that the Lagrange multipliers for the $\succeq$-constraints are matrices $\Lambda_\ell \succeq 0$ of the same size as the symmetric data matrices A_ℓ^j, B_ℓ. Aggregating the constraints of our SDO program and recalling that the inner product $\langle A, B\rangle$ in $\mathbf{S}^k$ is $\mathrm{Tr}(AB)$, the aggregated linear inequality reads

$$\sum_{j=1}^n x_j \left[\sum_{\ell=1}^L \mathrm{Tr}(A_\ell^j \Lambda_\ell) + \sum_{j=1}^n (P^T\mu)_j \right] \ge \sum_{\ell=1}^L \mathrm{Tr}(B_\ell \Lambda_\ell) + p^T\mu$$

The equality constraints of the dual should say that the left hand side expression, identically in $x \in \mathbb{R}^n$, is $c^T x$, that is, the dual problem reads

$$\max_{\{\Lambda_\ell\},\mu} \left\{ \sum_{\ell=1}^L \mathrm{Tr}(B_\ell \Lambda_\ell) + p^T\mu : \begin{array}{r} \mathrm{Tr}(A_\ell^j \Lambda_\ell) + (P^T\mu)_j = c_j, \\ 1 \le j \le n \\ \Lambda_\ell \succeq 0, 1 \le \ell \le L \end{array} \right\}$$

Symmetry of Conic Duality

$$\mathrm{Opt}(P) = \min_{x \in X} \left\{ c^T x : \begin{array}{l} A_\ell x \geq_{\mathbf{K}^\ell} b_\ell,\ 1 \leq \ell \leq L \\ Px = p \end{array} \right\} \qquad (P)$$

$$\mathrm{Opt}(D) = \max_{\lambda, \mu} \left\{ \sum_{\ell=1}^{L} b_\ell^T \lambda_\ell + p^T \mu : \begin{array}{l} \lambda_\ell \geq_{\mathbf{K}_*^\ell} 0,\ 1 \leq \ell \leq L \\ \sum_{\ell=1}^{L} A_\ell^T \lambda_\ell + P^T \mu = c \end{array} \right\} \qquad (D)$$

♠ Observe that (D) is, essentially, in the same form as (P), and thus we can build the dual of (D). To this end, we rewrite (D) as

$$-\mathrm{Opt}(D) = \min_{\lambda_\ell, \mu} \left\{ -\sum_{\ell=1}^{L} b_\ell^T \lambda_\ell - p^T \mu : \begin{array}{l} \lambda_\ell \geq_{\mathbf{K}_*^\ell} 0,\ 1 \leq \ell \leq L \\ \sum_{\ell=1}^{L} A_\ell^T \lambda_\ell + P^T \mu = c \end{array} \right\} \qquad (D')$$

$$-\mathrm{Opt}(D) = \min_{\lambda_\ell,\mu}\left\{-\sum_{\ell=1}^{L} b_\ell^T\lambda_\ell - p^T\mu : \begin{array}{l}\lambda_\ell \geq_{\mathbf{K}_*^\ell} 0,\ 1\leq\ell\leq L\\ \sum_{\ell=1}^{L} A_\ell^T\lambda_\ell + P^T\mu = c\end{array}\right\} \qquad (D')$$

Denoting by $-x$ the vector of Lagrange multipliers for the equality constraints in (D'), and by $\xi_\ell \geq_{[\mathbf{K}_*^\ell]_*} 0$ (i.e., $\xi_\ell \geq_{\mathbf{K}^\ell} 0$) the vectors of Lagrange multipliers for the $\geq_{\mathbf{K}_*^\ell}$-constraints in (D') and aggregating the constraints of (D') with these weights, we see that everywhere on the feasible domain of (D') it holds:

$$\textstyle\sum_\ell [\xi_\ell - A_\ell x]^T\lambda_\ell + [-Px]^T\mu \geq -c^T x$$

• When the left hand side in this inequality as a function of $\{\lambda_\ell\}, \mu$ is identically equal to the objective of (D'), i.e., when

$$\left\{\begin{array}{ll}\xi_\ell - A_\ell x = -b_\ell & 1\leq\ell\leq L,\\ -Px = -p & \end{array}\right.,$$

the quantity $-c^T x$ is a lower bound on $\mathrm{Opt}(D') = -\mathrm{Opt}(D)$, and the problem dual to (D) thus is

$$\max_{x,\xi_\ell}\left\{-c^T x : \begin{array}{l} A_\ell x = b_\ell + \xi_\ell, 1\leq\ell\leq L\\ Px = p\\ \xi_\ell \geq_{\mathbf{K}^\ell} 0,\ 1\leq\ell\leq L\end{array}\right\}$$

which is equivalent to (P).

$\Rightarrow$ *Conic duality is symmetric!*

Conic Duality Theorem

♠ A conic program in the form

$$\min_y \left\{ c^T y : Ry = r, \underbrace{Py \geq_{\mathbf{K}} p, Sy \geq s}_{Qy \geq_{\mathbf{M}} q} \right\}$$

is called *strictly feasible*, if there exists a *strictly feasible* solution $\bar{y}$ – a feasible solution where the vector inequality constraint is satisfied as strict: $Q\bar{y} >_{\mathbf{M}} q$. The program is called *essentially strictly feasible*, if there exists a feasible solution $\widehat{y}$ such that $P\widehat{y} >_{\mathbf{K}} p$.

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : Ax \geq_{\mathbf{K}} b,\ Px = p \right\} \qquad (P)$$
$$\mathrm{Opt}(D) = \max_{\lambda,\mu} \left\{ b^T\lambda + p^T\mu : \lambda \geq_{\mathbf{K}_*} 0,\ A^T\lambda + P^T\mu = c \right\} \qquad (D)$$

Conic Duality Theorem

♠ [Weak Duality] *One has* $\mathrm{Opt}(D) \leq \mathrm{Opt}(P)$.

♠ [Symmetry] *duality is symmetric:* (D) *is a conic program, and the program dual to* (D) *is (equivalent to)* (P).

♠ [Strong Duality] *Let one of the problems* (P),(D) *be essentially strictly feasible and bounded. Then the other problem is solvable, and* $\mathrm{Opt}(D) = \mathrm{Opt}(P)$.

In particular, if both (P) *and* (D) *are strictly feasible, then both the problems are solvable with equal optimal values.*

Example: Dual of the SDO relaxation. Recall that given a (difficult to solve!) quadratic quadratically constrained problem

$$\mathrm{Opt}_* = \min_x \{f_0(x) : f_i(x) \geq 0,\ 1 \leq i \leq m\}$$
$$f_i(x) = x^T Q_i x + 2b_i^T x + c_i$$

we can bound its optimal value from below by passing to the *semidefinite relaxation* of the problem:

$$\mathrm{Opt}_* \geq \mathrm{Opt}$$
$$:= \min_X \left\{ \mathrm{Tr}(F_0 X) : \begin{array}{l} \mathrm{Tr}(F_i X) \geq 0, 1 \leq i \leq m \\ X \succeq 0,\ X_{n+1,n+1} \equiv \mathrm{Tr}(GX) = 1 \end{array} \right\} \qquad (P)$$
$$G = \left[\begin{array}{c|c} & \\ \hline & 1 \end{array}\right],\ F_i = \left[\begin{array}{cc} Q_i & b_i \\ b_i^T & c_i \end{array}\right],\ 0 \leq i \leq m.$$

Let us build the dual to (P). Denoting by $\lambda_i \geq 0$ the Lagrange multipliers for the scalar inequality constraints, by $\Lambda \succeq 0$ the Lagrange multiplier for the LMI $X \succeq 0$, and by μ – the Lagrange multiplier for the equality constraint $X_{n+1,n+1} = 1$, and aggregating the constraints, we get the aggregated inequality

$$\mathrm{Tr}([\textstyle\sum_{i=1}^m \lambda_i F_i]X) + \mathrm{Tr}(\Lambda X) + \mu \mathrm{Tr}(GX) \geq \mu$$

Specializing the Lagrange multipliers to make the left hand side to be identically equal to $\mathrm{Tr}(F_0 X)$, the dual problem reads

$$\mathrm{Opt}(D) = \max_{\Lambda, \{\lambda_i\}, \mu} \left\{\mu : F_0 = \textstyle\sum_{i=1}^m \lambda_i F_i + \mu G + \Lambda, \lambda \geq 0, \Lambda \succeq 0\right\}$$

We can easily eliminate Λ, thus arriving at

$$\mathrm{Opt}(D) = \max_{\{\lambda_i\}, \mu} \left\{\mu : \textstyle\sum_{i=1}^m \lambda_i F_i + \mu G \preceq F_0, \lambda \geq 0\right\} \qquad (D)$$

Geometry of Primal-Dual Pair of Conic Problems

♣ Consider a primal-dual pair of conic problems in the form

$$\mathrm{Opt}(P) = \min_x \left\{c^T x : Ax \geq_{\mathbf{K}} b, Px = p\right\} \qquad (P)$$
$$\mathrm{Opt}(D) = \max_{\lambda,\mu} \left\{b^T\lambda + p^T\mu : \lambda \geq_{\mathbf{K}_*} 0, A^T\lambda + P^T\mu = c\right\} \qquad (D)$$

♠ **Assumption:** Linear equality constraints in (P) and (D) are feasible:

$$\exists \bar{x}, \bar{\lambda}, \bar{\mu} : P\bar{x} = p \;\&\; A^T\bar{\lambda} + P^T\bar{\mu} = c$$

♠ Let us pass in (P) from variable x to the slack variable $\xi = Ax - b$. For x satisfying the equality constraints $Px = p$ of (P) we have

$$c^T x = [A^T\bar{\lambda} + P^T\bar{\mu}]^T x = \bar{\lambda}^T Ax + \bar{\mu}^T Px = \bar{\lambda}^T\xi + \bar{\mu}^T p + \bar{\lambda}^T b$$

⇒ *(P) is equivalent to*

$$\begin{array}{rcll} \mathrm{Opt}(\mathcal{P}) & = & \min_\xi \left\{\bar{\lambda}^T\xi : \xi \in \mathcal{M}_P \cap \mathbf{K}\right\} & (\mathcal{P}) \\ & = & \mathrm{Opt}(P) - \left[b^T\bar{\lambda} + p^T\bar{\mu}\right] & \\ \mathcal{M}_P & = & \mathcal{L}_P - \bar{\xi}, \;\; \bar{\xi} = b - A\bar{x}, & \\ \mathcal{L}_P & = & \{\xi : \exists x : \xi = Ax, Px = 0\} & \end{array}$$

♠ Let us eliminate from (D) the variable μ. For $[\lambda;\mu]$ satisfying the equality constraint $A^T\lambda + P^T\mu = c$ of (D) we have

$$b^T\lambda + p^T\mu = b^T\lambda + \bar{x}^T P^T\mu = b^T\lambda + \bar{x}^T[c - A^T\lambda] = \underbrace{[b - A\bar{x}]}_{\bar{\xi}}{}^T\lambda + c^T\bar{x}$$

⇒ (D) is equivalent to

$$\mathrm{Opt}(\mathcal{D}) = \max_\lambda \left\{\bar{\xi}^T\lambda : \lambda \in \mathcal{M}_D \cap \mathbf{K}_*\right\} = \mathrm{Opt}(D) - c^T\bar{x} \qquad (\mathcal{D})$$
$$\mathcal{M}_D = \mathcal{L}_D + \bar{\lambda}, \; \mathcal{L}_D = \{\lambda : \exists \mu : A^T\lambda + P^T\mu = 0\}$$

$$\mathrm{Opt}(P) = \min_x \left\{c^T x : Ax \geq_{\mathbf{K}} b, Px = p\right\} \qquad (P)$$
$$\mathrm{Opt}(D) = \max_{\lambda,\mu} \left\{b^T\lambda + p^T\mu : \lambda \geq_{\mathbf{K}_*} 0, A^T\lambda + P^T\mu = c\right\} \qquad (D)$$

♣ **Intermediate Conclusion:** The primal-dual pair (C), (D) of conic problems with feasible equality constraints is equivalent to the pair

$$\mathrm{Opt}(\mathcal{P}) = \min_\xi \left\{\bar{\lambda}^T\xi : \xi \in \mathcal{M}_P \cap \mathbf{K}\right\} = \mathrm{Opt}(P) - \left[b^T\bar{\lambda} + p^T\bar{\mu}\right] \qquad (\mathcal{P})$$
$$\mathcal{M}_P = \mathcal{L}_P - \bar{\xi},\ \mathcal{L}_P = \{\xi : \exists x : \xi = Ax, Px = 0\}$$

$$\mathrm{Opt}(\mathcal{D}) = \max_\lambda \left\{\bar{\xi}^T\lambda : \lambda \in \mathcal{M}_D \cap \mathbf{K}_*\right\} = \mathrm{Opt}(D) - c^T\bar{\xi} \qquad (\mathcal{D})$$
$$\mathcal{M}_D = \mathcal{L}_D + \bar{\lambda},\ \mathcal{L}_D = \{\lambda : \exists\mu : A^T\lambda + P^T\mu = 0\}$$

Observation: *The linear subspaces $\mathcal{L}_P$ and $\mathcal{L}_D$ are orthogonal complements of each other.*

Observation: *Let x be feasible for (P) and $[\lambda, \mu]$ be feasible for (D), and let $\xi = Ax - b$ be the primal slack associated with x. Then*

$$\begin{aligned}\mathrm{DualityGap}(x, \lambda, \mu) &= c^T x - [b^T\lambda + p^T\mu] \\ &= [A^T\lambda + P^T\mu]^T x - [b^T\lambda + p^T\mu] \\ &= \lambda^T[Ax - b] + \mu^T[Px - p] = \lambda^T[Ax - b] = \lambda^T\xi.\end{aligned}$$

Note: To solve $(P), (D)$ $\Leftrightarrow$ to minimize the duality gap over primal feasible x and dual feasible λ, μ

$\Leftrightarrow$ *to minimize the inner product of $\xi^T\lambda$ over ξ feasible for $(\mathcal{P})$ and λ feasible for $(\mathcal{D})$.*

♣ **Conclusion:** *A primal-dual pair of conic problems*

$$\mathrm{Opt}(P) = \min_x \left\{c^T x : Ax \geq_{\mathbf{K}} b, Px = p\right\} \qquad (P)$$

$$\mathrm{Opt}(D) = \max_{\lambda,\mu} \left\{b^T\lambda + p^T\mu : \lambda \geq_{\mathbf{K}_*} 0, A^T\lambda + P^T\mu = c\right\} \qquad (D)$$

with feasible equality constraints is, geometrically, the problem as follows:

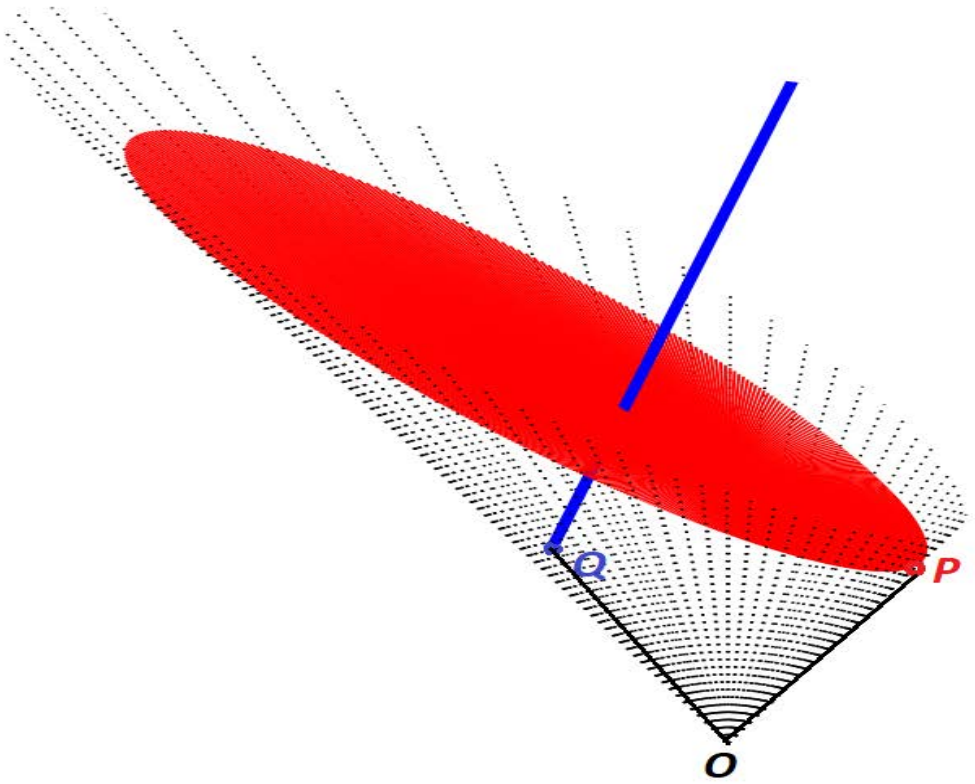
♠ **We are given**

- *a regular cone* $\mathbf{K}$ *in certain* $\mathbb{R}^N$ *along with its dual cone* $\mathbf{K}_*$
- *a linear subspace* $\mathcal{L}_P \subset \mathbb{R}^N$ *along with its orthogonal complement* $\mathcal{L}_D \subset \mathbb{R}^N$
- *a pair of vectors* $\bar{\xi}, \bar{\lambda} \in \mathbb{R}^N$.

These data define

- Primal feasible set $\Xi = [\mathcal{L}_P - \bar{\xi}] \cap \mathbf{K} \subset \mathbb{R}^N$
- Dual feasible set $\Lambda = [\mathcal{L}_D + \bar{\lambda}] \cap \mathbf{K}_* \subset \mathbb{R}^N$

♠ **We want** to find a pair $\xi \in \Xi$ and $\lambda \in \Lambda$ with as small as possible inner product. Whenever Ξ intersects $\mathrm{int}\,\mathbf{K}$ and Λ intersects $\mathrm{int}\,\mathbf{K}_*$, this geometric problem is solvable, and its optimal value is 0 (Conic Duality Theorem).



Primal-dual pair of conic problems on 3D Lorentz cone (self-dual)

Red: feasible set of $(\mathcal{P})$ Blue: feasible set of $(\mathcal{D})$

P – optimal solution to $(\mathcal{P})$; Q – optimal solution to $(\mathcal{D})$.

Pay attention to orthogonality of $\overrightarrow{OP}$ to $\overrightarrow{OQ}$

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : Ax \geq_{\mathbf{K}} b, Px = p \right\} \qquad (P)$$
$$\mathrm{Opt}(D) = \max_{\lambda,\mu} \left\{ b^T\lambda + p^T\mu : \lambda \geq_{\mathbf{K}_*} 0, A^T\lambda + P^T\mu = c \right\} \qquad (D)$$

♣ The data $\mathcal{L}_P$, $\bar{\xi}$, $\mathcal{L}_D$, $\bar{\lambda}$ of the geometric problem associated with $(P), (D)$ is as follows:

$$\mathcal{L}_P = \{\xi = Ax : Px = 0\}$$
$\bar{\xi}$: any vector of the form $Ax - b$ with $Px = p$
$$\mathcal{L}_D = \mathcal{L}_P^\perp = \{\lambda : \exists \mu : A^T\lambda + P^T\mu = 0\}$$
$\bar{\lambda}$: any vector λ such that $A^T\lambda + P^T\mu = c$ for some μ

• Vectors $\xi \in \Xi$ are exactly vectors of the form $Ax - b$ coming from feasible solutions x to (P), and vectors λ from Λ are exactly the λ-components of the feasible solutions $[\lambda; \mu]$ to (D).

• ξ_*, λ_* form an optimal solution to the geometric problem if and only if $\xi_* = Ax_* - b$ with $Px_* = p$, λ_* can be augmented by some μ_* to satisfy $A^T\lambda_* + P^T\mu_* = c$ and, in addition, x_* is optimal for (P), and $[\lambda_*; \mu_*]$ is optimal for (D).

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : Ax \geq_{\mathbf{K}} b, Px = p \right\} \qquad (P)$$
$$\mathrm{Opt}(D) = \max_{\lambda,\mu} \left\{ b^T\lambda + p^T\mu : \lambda \geq_{\mathbf{K}_*} 0, A^T\lambda + P^T\mu = c \right\} \qquad (D)$$

♣ **Conic Programming Optimality Conditions:** *Let both (P) and (D) be essentially strictly feasible. Then a pair $(x, [\lambda; \mu])$ of primal and dual feasible solutions is composed of optimal solutions to the respective problems if and only if*

- [Zero Duality Gap]

$$\mathrm{DualityGap}(x, [\lambda; \mu]) := c^T x - [b^T\lambda + p^T\mu] = 0$$

Indeed,

$$\mathrm{DualityGap}(x, [\lambda; \mu]) = \underbrace{[c^T x - \mathrm{Opt}(P)]}_{\geq 0} + \underbrace{[\mathrm{Opt}(D) - [b^T\lambda + p^T\mu]]}_{\geq 0}$$

and if and only if

- [Complementary Slackness]

$$[Ax - b]^T\lambda = 0$$

Indeed,

$$\begin{array}{rcl} [Ax - b]^T\lambda & = & (A^T\lambda)^T x - b^T\lambda = [c - P^T\mu]^T x - b^T\lambda \\ & = & c^T x - [b^T\lambda + \mu^T Px] \\ & = & c^T x - [b^T\lambda + p^T\mu] \\ & = & \mathrm{DualityGap}(x, [\lambda; \mu]) \end{array}$$

♣ **Conic Duality**, same as the LP one, is

- *fully algorithmic:* to write down the dual, given the primal, is a purely mechanical process
- *fully symmetric:* the dual problem "remembers" the primal one

♡ Cf. **Lagrange Duality:**

$$\min_x \{f(x) : g_i(x) \le 0,\ i = 1, ..., m\} \qquad (P)$$

$$\Downarrow$$

$$\max_{y \ge 0} \left[\underline{L}(y) := \min_x \left\{ f(x) + \sum_i y_i g_i(x) \right\} \right] \qquad (D)$$

- Dual "exists in the nature", but is given implicitly; its objective, typically, is not available in a closed form
- Duality is asymmetric: given $\underline{L}(\cdot)$, we, typically, cannot recover f and g_i...

♣ Conic Duality in the case of Magic cones:

- powerful tool to process problem, to some extent, "on paper", which in many cases provides extremely valuable insight and/or allows to end up with a problem much better suited for numerical processing
- is heavily exploited by efficient polynomial time algorithms for Magic conic problems

Illustration: Semidefinite Relaxation

♣ Recall that a quadratically constrained quadratic program

$$\begin{array}{c}\mathrm{Opt} = \min_{x\in\mathbb{R}^n}\{f_0(x) : f_i(x) \le 0, 1 \le i \le m\} \\ \left[f_i(x) = x^T Q_i x + 2b_i^T x + c_i,\ 0 \le i \le m\right]\end{array} \qquad (QP)$$

admits *semidefinite relaxation:* We associate with x the symmetric matrix

$$X[x] = [x;1][x;1]^T = \left[\begin{array}{cc} xx^T & x \\ x^T & 1\end{array}\right]$$

rewrite (QP) equivalently as

$$\mathrm{Opt} = \min_X \left\{ \mathrm{Tr}(F_0X) : \begin{array}{l} \mathrm{Tr}(F_iX) \le 0, 1 \le i \le m \\ X = X[x] \text{ for some } x \\ \qquad\qquad \Updownarrow \\ X \succeq 0, X_{n+1,n+1} = 1, \mathrm{Rank}\,(X) = 1 \end{array} \right\} \qquad (QP')$$

$$\left[F_i = \left[\begin{array}{cc} Q_i & b_i \\ b_i^T & c_i\end{array}\right]\right]$$

and remove the "troublemaking" rank restriction, arriving at the *semidefinite relaxation* of (QP) – the problem

$$\mathrm{Opt(SDO)} = \min_X \left\{ \mathrm{Tr}(F_0X) : \begin{array}{c} \mathrm{Tr}(F_iX) \le 0, 1 \le i \le M \\ X \succeq 0, X_{n+1,n+1} = 1 \end{array} \right\}$$

$$\boxed{\begin{array}{c}\mathrm{Opt} = \min\limits_{x\in\mathbb{R}^n}\{f_0(x) : f_i(x) \le 0, 1 \le i \le m\} \\ \left[f_i(x) = x^T Q_i x + 2b_i^T x + c_i,\ 0 \le i \le m\right]\end{array}} \quad (QP)$$

$$\Updownarrow$$

$$\boxed{\begin{array}{c}\mathrm{Opt} = \min\limits_{X}\left\{\mathrm{Tr}(F_0X) : \begin{array}{c}\mathrm{Tr}(F_iX) \le 0, 1 \le i \le m \\ X = \left[\begin{array}{cc} xx^T & x \\ x^T & 1\end{array}\right] \text{ for some } x\end{array}\right\} \quad (QP') \\ \left[F_i = \left[\begin{array}{cc} Q_i & b_i \\ b_i^T & c_i\end{array}\right]\right]\end{array}}$$

$$\Downarrow$$

$$\boxed{\mathrm{Opt(SDO)} = \min_X\left\{\mathrm{Tr}(F_0X) : \begin{array}{c}\mathrm{Tr}(F_iX) \le 0, 1 \le i \le M \\ X \succeq 0, X_{n+1,n+1} = 1\end{array}\right\} \text{(SDO)}}$$

♠ **Probabilistic Interpretation of** (SDO)**:**
Assume that instead of solving (QP) in deterministic variables x, we are solving the problem in *random vectors* ξ and want to minimize the *expected value* of the objective under the restriction that *the constraints are satisfied at average.*
Since f_i are quadratic, the expectations of the objective and the constraints are affine functions of the *moment matrix*

$$X = \mathbf{E}\left\{\left[\begin{array}{cc} \xi\xi^T & \xi \\ \xi^T & 1\end{array}\right]\right\}$$

which can be an arbitrary symmetric positive semidefinite matrix X with $X_{n+1,n+1} = 1$. *The "randomized" version of* (QP) *is exactly* (SDO) (check it!)

♣ With outlined interpretation, *an optimal solution to* (SDO) *gives rise to* (various) *randomized solutions to the problem of interest*.

In good cases, *we can extract from these randomized solutions feasible solutions to the problem of interest with reasonable approximation guarantees in terms of optimality.*

We can, e.g.,

— use X_* to generate a sample $\xi^1, ..., \xi^N$ of, say, $N = 100$ random solutions to (QP),

— "correct" ξ^t to get *feasible* solutions x^t to (QP).
The approach works when the correction is easy, e.g., when at some known point $\bar{x}$ the constraints of (QP) are satisfied *strictly*. Here we can take as x^t the closest to ξ^t *feasible* solution from the segment $[\bar{x}, \xi^t]$.

— select from the resulting N feasible solutions x^t to (QP) the best in terms of the objective.

♡ When applicable, the outlined approach can be combined with *local improvement* – N runs of any traditional algorithm for nonlinear optimization as applied to (QP), $x^1, ..., x^N$ being the starting points of the runs.

Lagrangian Relaxation

♣ Recall that for every MP problem

$$\mathrm{Opt}(P) = \min_{x \in X} \{f(x) : g_i(x) \le 0,\ 1 \le i \le m\} \qquad (P)$$

its Lagrange function

$$L(x, \lambda) = f(x) + \sum_i \lambda_i g_i(x)$$

underestimates $f(x)$ on the feasible set of (P), provided $\lambda \ge 0$: $\lambda \ge 0 \Rightarrow$

$$\mathrm{Opt}(D) = \max_{\lambda \ge 0} \left[\underline{L}(\lambda) := \inf_{x \in X} L(x, \lambda)\right] \le \mathrm{Opt}(P)$$

("Weak Lagrange duality").

♠ *Whenever $\underline{L}$ is efficiently computable, $\mathrm{Opt}(D)$ is an efficiently computable lower bound on $\mathrm{Opt}(P)$.*

♣ **Example:**

$$\text{Opt} = \min_{x \in \mathbb{R}^n} \{f_0(x) : f_i(x) \leq 0, 1 \leq i \leq m\} \quad (QP)$$

$$\left[f_i(x) = x^T Q_i x + 2b_i^T x + c_i,\ 0 \leq i \leq m\right]$$

- Applying Lagrange Relaxation Scheme, we get

$$\begin{aligned} \underline{L}(\lambda) &= \inf_x \left\{f_0(x) + \textstyle\sum_{i=1}^m \lambda_i f_i(x)\right\} \\ &= \inf_x \left\{ x^T \left[Q_0 + \textstyle\sum_{i=1}^m \lambda_i Q_i\right] x + 2\left[b_0 + \textstyle\sum_{i=1}^m \lambda_i b_i\right]^T x \right. \\ &\qquad \left. + \left[c_0 + \textstyle\sum_{i=1}^m \lambda_i c_i\right] \right\} \end{aligned}$$

Simple Fact: $x^T P x + 2q^T x + r \geq \tau$ *for all* $x \in \mathbb{R}^n$ *iff*

$$\begin{bmatrix} P & q \\ q^T & r - \tau \end{bmatrix} \succeq 0$$

- *Using Simple Fact, the Lagrange dual of* (QP) *becomes*

$$\text{Opt}(D) = \max_{\lambda, \tau} \left\{ \tau : \lambda \geq 0, \begin{bmatrix} Q_0 + \sum_{i=1}^m \lambda_i Q_i & b_0 + \sum_{i=1}^m \lambda_i b_i \\ b_0^T + \sum_{i=1}^m \lambda_i b_i^T & c_0 + \sum_{i=1}^m \lambda_i c_i - \tau \end{bmatrix} \succeq 0 \right\}$$

$$\mathrm{Opt} = \min_{x\in\mathbb{R}^n} \{f_0(x) : f_i(x) \le 0, 1 \le i \le m\} \quad (QP)$$
$$\left[f_i(x) = x^T Q_i x + 2b_i^T x + c_i,\ 0 \le i \le m\right]$$

♠ **Note:** *The SDO relaxations of (QP) resulting from our two relaxation schemes read*

$$\mathrm{Opt(SDO)} = \min_X \left\{ \mathrm{Tr}(F_0 X) : \begin{array}{c} \mathrm{Tr}(Q_i X) \le 0, 1 \le i \le M \\ X \succeq 0, X_{n+1,n+1} = 1 \end{array} \right\} \quad (P)$$

$$\mathrm{SDO} = \max_{\lambda,\tau} \left\{ \tau : \left[\begin{array}{c|c} Q_0 + \sum_{i=1}^m \lambda_i Q_i & b_0 + \sum_{i=1}^m \lambda_i b_i \\ \hline b_0^T + \sum_{i=1}^m \lambda_i b_i^T & c_0 + \sum_{i=1}^m c_i\lambda_i - \tau \end{array}\right] \succeq 0 \right\} \quad (D)$$
$$\lambda \ge 0$$

On a closest inspection, they are just semidefinite duals of each other!

♣ **Example: Quadratic Maximization over the box**

$$\text{Opt} = \max_x\{x^T Lx : x_i^2 \le 1,\ 1 \le i \le n\} \qquad (QP)$$
$$\Rightarrow \text{Opt(SDO)} = \max_X\{\text{Tr}(XL) : X \succeq 0, X_{ii} \le 1\ \forall i\} \quad \text{(SDO)}$$

Note: When $L \succeq 0$ or L has zero diagonal, Opt and Opt(SDO) remain intact when the inequality constraints are replaced with their equality versions.

♠ **MAXCUT:** The combinatorial problem *"given n-node graph with arcs assigned nonnegative weights $a_{ij} = a_{ij}$, $1 \le i, j \le n$, split the nodes into two non-overlapping subsets to maximize the total weight of the arcs linking nodes from different subsets"* is equivalent to (QP) with

$$L_{ij} = \begin{cases} \sum_k a_{ik} & , j = i \\ -a_{ij} & , j \ne i \end{cases}$$

♠ **Theorem of Goemans and Williamson** '94:

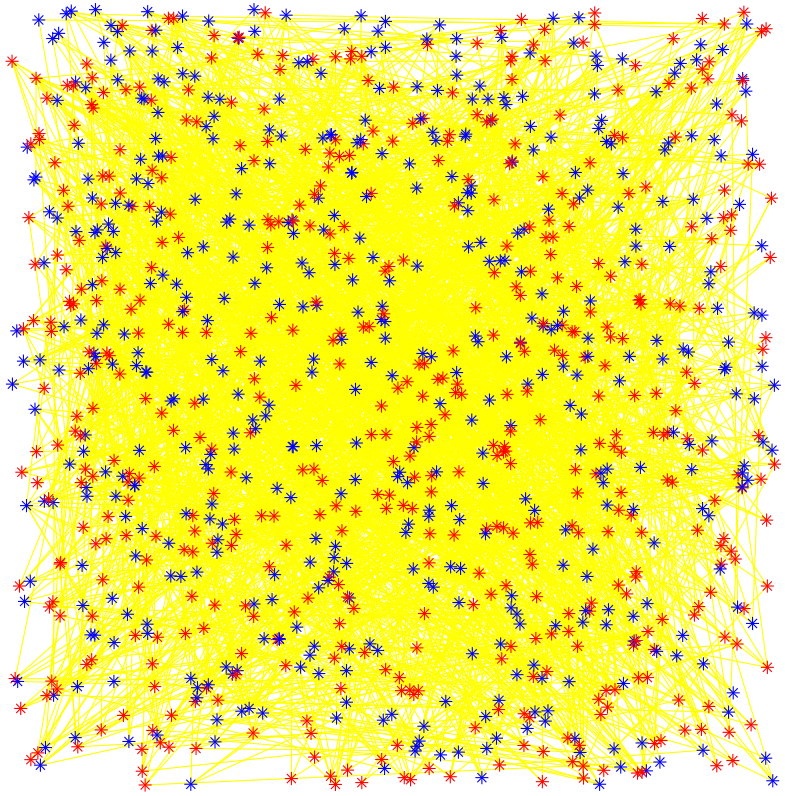
$$\text{Opt} \le \text{Opt(SDO)} \le 1.1383 \cdot \text{Opt} \qquad (!)$$

Note: To approximate Opt within 4% is NP-hard...

Sketch of the proof of (!): treat an optimal solution X_* of (SDO) as the covariance matrix of zero mean Gaussian random vector ξ and look at

$$\mathbf{E}\{\mathrm{sign}[\xi]^T L \mathrm{sign}[\xi]\}.$$

Illustration: MAXCUT, 1024 nodes, 2614 arcs.



Suboptimal cut, weight $\geq 0.9196 \cdot \text{Opt(SDO)} \geq 0.9196 \cdot \text{Opt}$

[Slightly better than Goemans-Williamson guarantee:
weight $\geq 0.8785 \cdot \text{Opt(SDO)} \geq 0.8785 \cdot \text{Opt}$]

$$\text{Opt} = \max_x\{x^T L x : x_i^2 \le 1,\ 1 \le i \le n\} \qquad (QP)$$
$$\Rightarrow \text{Opt(SDO)} = \max_X\{\text{Tr}(XL) : X \succeq 0, X_{ii} \le 1\ \forall i\}(\text{SDO})$$

♠ **Nesterov's $\pi/2$ Theorem.** Matrix L arising in MAXCUT is $\succeq 0$ (and possesses additional properties). What can be said about (SDO) under the only restriction $L \succeq 0$?

Answer [Nesterov'98]: $\text{Opt} \le \text{Opt(SDO)} \le \frac{\pi}{2} \cdot \text{Opt}$.

Illustration: L: randomly built positive semidefinite 1024×1024 matrix. Relaxation combined with local improvement yields a feasible solution $\bar{x}$ with

$$\bar{x}^T L \bar{x} \ge 0.7867 \cdot \text{Opt(SDO)} \ge 0.7867 \cdot \text{Opt}$$

$$\begin{array}{rl}\text{Opt} = \max_x\{x^TLx : x_i^2 \le 1,\ 1 \le i \le n\} & (QP)\\ \Rightarrow \text{Opt(SDO)} = \max_X\{\text{Tr}(XL) : X \succeq 0, X_{ii} \le 1\ \forall i\} & \text{(SDO)}\end{array}$$

♠ **The case of indefinite L:** When L is an arbitrary symmetric matrix, one has

$$\text{Opt} \le \text{Opt(SDO)} \le O(1)\ln(n)\text{Opt}.$$

This is a particular case of the following result: *The SDO relaxation*

$$\text{Opt(SDO)} = \max_X\{\text{Tr}(XL) : \text{Tr}(XQ_i) \le 1, i \le m\}$$

of the problem

$$\begin{array}{c}\text{Opt} = \max_x\left\{x^TLx : x^TQ_ix \le 1, i \le m\right\}\\ [Q_i \succeq 0\ \forall i, \sum_i Q_i \succ 0]\end{array} \qquad (P)$$

satisfies $\text{Opt} \le \text{Opt(SDO)} \le O(1)\ln(m)\text{Opt}$.

Illustration, A: Problem (QP) with randomly selected indefinite 1024×1024 matrix L. Relaxation combined with local improvement yields a feasible solution $\bar{x}$ with

$$\bar{x}^TL\bar{x} \ge 0.7649 \cdot \text{Opt(SDO)} \ge 0.7649 \cdot \text{Opt}$$

Illustration, B: Problem (P) with randomly selected indefinite 1024×1024 matrix L and 64 randomly selected positive semidefinite matrices Q_i of rank 64. Relaxation yields a feasible solution $\bar{x}$ with

$$\bar{x}^TL\bar{x} \ge 0.9969 \cdot \text{Opt(SDO)} \ge 0.9969 \cdot \text{Opt}$$

Illustration: Lyapunov Stability Analysis

♣ Consider an *uncertain* time varying linear dynamical system

$$\frac{d}{dt}x(t) = A(t)x(t) \qquad \text{(ULS)}$$

- $x(t) \in \mathbb{R}^n$: state at time t,
- $A(t) \in \mathbb{R}^{n\times n}$: known to take all values in a given *uncertainty set* $\mathcal{U} \subset \mathbb{R}^{n\times n}$.

♠ (ULS) is called *stable*, if all trajectories of the system go to 0 as $t \to \infty$:

$$A(t) \in \mathcal{U}\ \forall t \geq 0,\ \tfrac{d}{dt}x(t) = A(t)x(t) \Rightarrow \lim_{t\to\infty} x(t) = 0.$$

♣ **Question:** *How to certify stability?*

♠ Standard *sufficient* stability condition is *the existence of Lyapunov Stability Certificate – a matrix $X \succ 0$ such that the function $L(x) = x^T Xx$ for some $\alpha > 0$ satisfies $\frac{d}{dt}L(x(t)) \leq -\alpha L(x(t))$* for all trajectories *and thus goes to 0 exponentially fast along the trajectories:*

$$\begin{aligned}
&\tfrac{d}{dt}L(x(t)) \leq -\alpha L(x(t)) \Rightarrow \tfrac{d}{dt}\left[\exp\{\alpha t\}L(x(t))\right] \leq 0\\
\Rightarrow\ &\exp\{\alpha t\}L(x(t)) \leq L(x(0)),\ t \geq 0\\
\Rightarrow\ &L(x(t)) \leq \exp\{-\alpha t\}L(x(0))\\
\Rightarrow\ &\|x(t)\|_2^2 \leq \frac{\lambda_{\max}(X)}{\lambda_{\min}(X)}\exp\{-\alpha t\}\|x(0)\|_2^2
\end{aligned}$$

- For a *time-invariant* system, this condition is necessary and sufficient for stability.

♠ **Question:** *When $\alpha > 0$ is such that*
$\frac{d}{dt}L(x(t)) \leq -\alpha L(x(t))$ *for all trajectories $x(t)$ satisfying*
$\frac{d}{dt}x(t) = A(t)x(t)$ *with $A(t) \in \mathcal{U}$ for all t ?*

♡ **Answer:** We should have

$$\begin{array}{rcl}\frac{d}{dt}\left(x^T(t)Xx(t)\right) & = & (\frac{d}{dt}x(t))^T Xx(t) + x^T(t)X\frac{d}{dt}x(t) \\ & = & x^T(t)A^T(t)Xx(t) + x^T(t)XAx(t) \\ & = & x^T(t)\left[A^T(t)X + XA(t)\right]x(t) \\ & \leq & -\alpha x^T(t)Xx(t)\end{array}$$

Thus,

$\frac{d}{dt}L(x(t)) \leq -\alpha L(x(t))$ for all trajectories
$\Leftrightarrow$ $x^T(t)\left[A^T(t)X + XA(t)\right]x(t) \leq -\alpha x^T(t)Xx(t)$ for all trajectories
$\Leftrightarrow$ $x^T(t)[A^T(t)X + XA(t) + \alpha X]x(t) \leq 0$ for all trajectories
$\Leftrightarrow$ $A^TX + XA \preceq -\alpha X \ \forall A \in \mathcal{U}$

$\Rightarrow$ $X \succ 0$ is LSC for a given $\alpha > 0$ iff X solves semi-infinite LMI

$$A^TX + XA \preceq -\alpha X \ \forall A \in \mathcal{U}$$

$\Rightarrow$ *Uncertain linear dynamical system*

$$\frac{d}{dt}x(t) = A(t)x(t), \ A(t) \in \mathcal{U}$$

admits an LSC iff the semi-infinite system of LMI's

$$X \succeq I, \ A^TX + XA \preceq -I \ \forall A \in \mathcal{U}$$

in matrix variable X is solvable.

♠ **But:** *SDP is about finite, and not semi-infinite, systems of LMI's. Semi-infinite systems of LMI's typically are heavily computationally intractable...*

$$X \succeq I,\ A^T X + X A \preceq -I\ \forall A \in \mathcal{U} \qquad (!)$$

♠ **Solvable case I:** *Scenario* (a.k.a. *polytopic*) *uncertainty* $\mathcal{U} = \mathrm{Conv}\{A_1, ..., A_N\}$.
Here (!) is equivalent to the finite system of LMI's

$$X \succeq I,\ A_k^T X + X A_k \preceq -I,\ 1 \le k \le N$$

♠ **Solvable case II:** *Unstructured Norm-Bounded uncertainty*

$$\mathcal{U} = \{A = \bar{A} + B\Delta C :\ \|\Delta\|_{2,2} \le \rho\},$$

- $\|\cdot\|_{2,2}$: spectral norm of a matrix.

♡ **Example:** We close *open loop time invariant system*

$$\begin{array}{rcll} \frac{d}{dt}x(t) & = & Px(t) + Bu(t) & \text{[state equations]} \\ y(t) & = & Cx(t) & \text{[observed output]} \end{array}$$

with *linear feedback*

$$u(t) = Ky(t),$$

thus arriving at the *closed loop* system

$$\frac{d}{dt}x(t) = [P + BKC]x(t)$$

and want to certify stability of the closed loop system when the feedback matrix K is subject to time-varying norm-bounded perturbations:

$$K = K(t) \in \mathcal{V} = \{\bar{K} + \Delta : \|\Delta\|_{2,2} \le \rho\}.$$

This is exactly the same as to certify stability of the system

$$\frac{d}{dt}x(t) = A(t)x(t),\ A(t) \in \mathcal{U} = \{\underbrace{P + B\bar{K}C}_{\bar{A}} + B\Delta C\}$$

with unstructured norm-bounded uncertainty.

- **Observation:** The semi-infinite system of LMI's

$$X \succeq I \;\&\; A^T X + X A^T \preceq -I \,\forall (A = \bar{A} + B\Delta C : \|\Delta\|_{2,2} \le \rho)$$

is of the generic form

$$\begin{cases} (A): & \text{finite system of LMI's in variables } x \\ \hline & \text{semi-infinite LMI} \\ (!): & A(x) + L^T(x)\Delta R + R^T\Delta^T L(x) \succeq 0 \;\forall (\Delta : \|\Delta\|_{2,2} \le \rho) \\ & A(x),\ L(x): \text{ affine in } x \end{cases}$$

♠ **Fact:** [S. Boyd et al, early 90's] *Assuming w.l.o.g. that $R \neq 0$, the semi-infinite LMI (!) can be equivalently represented by the usual LMI*

$$\left[\begin{array}{c|c} A(x) - \lambda R^T R & \rho L^T(x) \\ \hline \rho L(x) & \lambda I \end{array}\right] \succeq 0 \quad (!!)$$

in variables x, λ, meaning that x satisfies (!) if and only x can be augmented by properly selected λ to satisfy (!!).

♣ Key argument when proving Fact:

S-Lemma: *A homogeneous quadratic inequality*

$$x^T Bx \geq 0 \qquad (B)$$

is a consequence of strictly feasible homogeneous quadratic inequality

$$x^T Ax \geq 0 \qquad (A)$$

if and only if (B) can be obtained by taking weighted sum, with nonnegative weights, of (A) and identically true homogeneous quadratic inequality:

$$\exists(\lambda \geq 0 \ \& \ C : \underbrace{x^T Cx \geq 0 \, \forall x}_{\Leftrightarrow C \succeq 0}) : x^T Bx \equiv \lambda x^T Ax + x^T Cx$$

or, which is the same, if and only if

$$\exists \lambda \geq 0 : B \succeq \lambda A.$$

Immediate corollary: *A quadratic inequality*

$$x^T Bx + 2b^T x + \beta \geq 0$$

is a consequence of strictly feasible quadratic inequality

$$x^T Ax + 2a^T x + \alpha \geq 0$$

iff

$$\exists \lambda \geq 0 : \left[\begin{array}{c|c} B - \lambda A & b^T - \lambda a^T \\ \hline b - \lambda a & \beta - \lambda \alpha \end{array}\right] \succeq 0$$

$\Rightarrow$ *We can efficiently optimize a quadratic function over the set given by a single strictly feasible quadratic constraint.*

♣ S-**Lemma:** *A homogeneous quadratic inequality*

$$x^T Bx \geq 0 \qquad (B)$$

is a consequence of strictly feasible homogeneous quadratic inequality

$$x^T Ax \geq 0 \qquad (A)$$

if and only if (B) *can be obtained by taking weighted sum, with nonnegative weights, of* (A) *and identically true homogeneous quadratic inequality:*

$$\exists(\lambda \geq 0 \ \& \ C : \underbrace{x^T Cx \geq 0 \, \forall x}_{\Leftrightarrow C \succeq 0}) : x^T Bx \equiv \lambda x^T Ax + x^T Cx$$

or, which is the same, if and only if

$$\exists \lambda \geq 0 : B \succeq \lambda A.$$

♠ **Note:** The "if" part of the claim is evident and remains true when we replace (A) with a *finite system* of quadratic inequalities: *Let a system of homogeneous quadratic inequalities*

$$x^T A_i x \geq 0, \ 1 \leq i \leq m,$$

and a "target" inequality $x^T Bx \geq 0$ *be given. If the target inequality can be obtained by taking weighted sum, with nonnegative coefficients, of the inequalities of the system and an identically true homogeneous quadratic inequality,* or, equivalently, If there exist $\lambda_i \geq 0$ such that

$$B \succeq \sum_i \lambda_i A_i,$$

then the target inequality is a consequence of the system.

$$\exists \lambda_i \geq 0 : B \succeq \sum_{i=1}^m \lambda_i A_i \qquad (!)$$
$$\Rightarrow x^T B x \geq 0 \text{ is a consequence of } x^T A_i x \geq 0, 1 \leq i \leq m$$

- If instead of homogeneous *quadratic* inequalities we were speaking about homogeneous *linear* ones, similar *sufficient* condition for the target inequality to be a consequence of the system would be also *necessary* (Homogeneous Farkas Lemma).
- The power of S-Lemma is in the claim that *when $m = 1$, the sufficient condition (!) for the target inequality $x^T B x \geq 0$ to be a consequence of the system $x^T A_i x \geq 0$, $1 \leq i \leq m$, is also necessary*, provided the "system" $x^T A_1 x \geq 0$ is strictly feasible.

The "necessity" part of S-Lemma *fails to be true* when $m > 1$.

From $\mathcal{S}$-Lemma to Fact

$$A + L^T \Delta R + R^T \Delta^T L \succeq 0 \ \forall (\Delta : \|\Delta\|_{2,2} \leq \rho)$$

$$\Updownarrow$$

$$A + [\rho L]^T \Delta R + R^T \Delta^T [\rho L] \succeq 0 \ \forall (\Delta : \|\Delta\|_{2,2} \leq 1)$$

$$\Updownarrow$$

$$\xi^T A \xi + 2[\underbrace{\Delta R \xi}_{\eta}]^T [\rho L \xi] \geq 0 \ \forall (\Delta, \xi : \|\Delta\|_{2,2} \leq 1)$$

$$\Updownarrow$$

$$\xi^T A \xi + 2\eta^T [\rho L \xi] \geq 0 \ \ \forall (\xi, \eta : \|\eta\|_2 \leq \|R\xi\|_2)$$

$$\Updownarrow$$

$$\xi^T R^T R \xi - \eta^T \eta \geq 0 \Rightarrow \xi^T A \xi + 2\eta^T \rho L \xi) \geq 0$$

$$\Updownarrow$$

$$\exists \lambda \geq 0 : \xi^T A \xi + 2\eta^T \rho L \xi - \lambda [\xi^T R^T R \xi - \eta^T \eta] \geq 0 \, \forall (\xi, \eta)$$

$$\Updownarrow$$

$$\exists \lambda \geq 0 : \left[\begin{array}{c|c} A - \lambda R^T R & \rho L^T \\ \hline \rho L & \lambda I \end{array}\right] \succeq 0.$$

Proof of the "only if" part of S-Lemma

• **Situation:** We are given two symmetric matrices A, B such that

(I): $$\exists \bar{x} : \bar{x}^T A \bar{x} > 0$$

and

(II): $$x^T A x \geq 0 \text{ implies } x^T B x \geq 0$$

or, equivalently,

(I-II): $$\text{Opt} := \min_x \{x^T B x : x^T A x \geq 0\} \geq 0$$
and the constraint $x^T A x \geq 0$ is strictly feasible

• **Goal:** To prove that

(III): $$\exists \lambda \geq 0 : B \succeq \lambda A$$

or, equivalently, that

(III′): $$\text{SDO} := \min_X \{\text{Tr}(BX) : \text{Tr}(AX) \geq 0,\ X \succeq 0\} \geq 0.$$

Equivalence of (III) and (III′): By (I), semidefinite program in (III′) is strictly feasible. Since the program is homogeneous, its optimal value is either 0, or $-\infty$. By Conic Duality, the optimal value is finite (i.e., 0) if and only if the dual problem

$$\max_{\lambda, Y} \{0 : B = \lambda A + Y, \lambda \geq 0, Y \succeq 0\}$$

is solvable, which is exactly (III).

- Given that $x^TAx \geq 0$ implies $x^TBx \geq 0$ we should prove that

$$\mathrm{Tr}(BX) \geq 0 \textit{ whenever } \mathrm{Tr}(AX) \geq 0 \textit{ and } X \succeq 0$$

- Let $X \succeq 0$ be such that $\mathrm{Tr}(AX) \geq 0$, and let us prove that $\mathrm{Tr}(BX) \geq 0$.

There exists *orthogonal* U such that $U^TX^{1/2}AX^{1/2}U$ is diagonal

$\Rightarrow$ For every vector ξ with ± 1 entries:

$$\begin{aligned}[X^{1/2}U\xi]^TA[X^{1/2}U\xi] &= \xi^T\underbrace{[U^TX^{1/2}AX^{1/2}U]}_{\text{diagonal}}\xi \\ &= \mathrm{Tr}(U^TX^{1/2}AX^{1/2}U) \\ &= \mathrm{Tr}(AX) \geq 0\end{aligned}$$

$\Rightarrow$ For every vector ξ with ± 1 entries:

$$0 \leq [X^{1/2}U\xi]^TB[X^{1/2}U\xi] = \xi^T[U^TX^{1/2}BX^{1/2}U]\xi$$

$\Rightarrow$ [Taking average over ± 1 vectors ξ]

$$0 \leq \mathrm{Tr}(U^TX^{1/2}BX^{1/2}U) = \mathrm{Tr}(BX)$$

Thus, $\mathrm{Tr}(BX) \geq 0$, as claimed.

Note: We have used the standard fact of Linear Algebra: *if the product PQ of matrices P, Q makes sense and is a square matrix, then* $\mathrm{Tr}(PQ) = \mathrm{Tr}(QP)$.

Lecture III.3
Interior Point Algorithms
for
Linear and Semidefinite Optimization

Interior Point Methods for LO and SDO

♣ **Interior Point Methods** (IPM's) are state-of-the-art theoretically and practically efficient polynomial time algorithms for solving well-structured convex optimization programs, primarily Linear, Conic Quadratic and Semidefinite ones.

Modern IPMs were first developed for LO, and the words "Interior Point" are aimed at stressing the fact that instead of traveling along the vertices of the feasible set, as in the Simplex algorithm, the new methods work in the interior of the feasible domain.

♠ Basic theory of IPMs remains the same when passing from LO to SDO

$\Rightarrow$ It makes sense to study this theory in the more general SDO case.

Primal-Dual Pair of SDO Programs

♣ Consider an SDO program in the form

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \mathcal{A}x := \sum_{j=1}^n x_j A_j \succeq B \right\} \quad (P)$$

where A_j, B are $m \times m$ block diagonal symmetric matrices of a given block-diagonal structure $\nu = (\nu_1, ..., \nu_K)$ (i.e., with a given number K and given sizes ν_k, $k \leq K$, of diagonal blocks). (P) can be thought of as a conic problem on the self-dual and regular positive semidefinite cone $\mathbf{S}^\nu_+$ in the space $\mathbf{S}^\nu$ of symmetric block diagonal $m \times m$ matrices with block-diagonal structure ν.

Note: In the diagonal case (with the block-diagonal structure in question, all diagonal blocks are of size 1), (P) becomes a LO program with m linear inequality constraints and n variables.

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \mathcal{A}x := \sum_{j=1}^n x_j A_j \succeq B \right\} \qquad (P)$$

♠ **Standing Assumption A:** The mapping $x \mapsto \mathcal{A}x$ has trivial kernel, or, equivalently, the matrices $A_1, ..., A_n$ are linearly independent.

♠ The problem dual to (P) is

$$\mathrm{Opt}(D) = \max_{S \in \mathbf{S}^\nu} \{ \mathrm{Tr}(BS) : S \succeq 0, \mathrm{Tr}(A_j S) = c_j \, \forall j \} \qquad (D)$$

♠ **Standing Assumption B:** Both (P) and (D) are strictly feasible ($\Rightarrow$ both problems are solvable with equal optimal values).

♠ Let $C \in \mathbf{S}^\nu$ satisfy the equality constraint in (D). Passing in (P) from x to the primal slack $X = \mathcal{A}x - B$, (P) becomes the problem

$$\mathrm{Opt}(\mathcal{P}) = \min_{X \in \mathbf{S}^\nu} \left\{ \mathrm{Tr}(CX) : X \in [\mathcal{L}_P - B] \cap \mathbf{S}^\nu_+ \right\} \qquad (\mathcal{P})$$
$$\mathcal{L}_P = \{X = \mathcal{A}x\} = \mathrm{Lin}\{A_1, ..., A_n\}$$

while (D) is the problem

$$\mathrm{Opt}(D) = \max_{S \in \mathbf{S}^\nu} \left\{ \mathrm{Tr}(BS) : S \in [\mathcal{L}_D + C] \cap \mathbf{S}^\nu_+ \right\} \qquad (D)$$
$$\mathcal{L}_D = \mathcal{L}_P^\perp = \{S \in \mathbf{S}^\nu : \mathrm{Tr}(A_j S) = 0, \, 1 \le j \le n\}$$

$$
\begin{array}{rl}
\mathrm{Opt}(P) = \min\limits_{x} \left\{ c^T x : \mathcal{A}x := \sum_{j=1}^n x_j A_j \succeq B \right\} & (P) \\
\Leftrightarrow \mathrm{Opt}(\mathcal{P}) = \min\limits_{X} \left\{ \mathrm{Tr}(CX) : X \in [\mathcal{L}_P - B] \cap \mathbf{S}^\nu_+ \right\} & (\mathcal{P}) \\
\Leftarrow \mathrm{Opt}(D) = \max\limits_{S} \left\{ \mathrm{Tr}(BS) : S \in [\mathcal{L}_D + C] \cap \mathbf{S}^\nu_+ \right\} & (D) \\
\left[\mathcal{L}_P = \mathrm{Im}\mathcal{A},\ \mathcal{L}_D = \mathcal{L}_P^\perp \right] &
\end{array}
$$

Since (P) and (D) are strictly feasible, both problems are solvable with equal optimal values, and a pair of feasible solutions X to $(\mathcal{P})$ and S to $(\mathcal{D})$ is composed of optimal solutions to the respective problems iff $\mathrm{Tr}(XS) = 0$.

Fact: *For positive semidefinite X, S, $\mathrm{Tr}(XS) = 0$ if and only if $XS = SX = 0$.*

Proof: • **Standard Fact of Linear Algebra:** *For every matrix $A \succeq 0$ there exists exactly one matrix $B \succeq 0$ such that $A = B^2$; B is denoted $A^{1/2}$.*

• **Standard Fact of Linear Algebra:** *Whenever A, B are matrices such that the product AB makes sense and is a square matrix,* $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$.

• **Standard Fact of Linear Algebra:** *Whenever $A \succeq 0$ and QAQ^T makes sense, we have $QAQ^T \succeq 0$.*

• Standard Facts of LA $\Rightarrow$ Claim:

$0 = \mathrm{Tr}(XS) = \mathrm{Tr}(X^{1/2}X^{1/2}S) = \mathrm{Tr}(X^{1/2}SX^{1/2}) \Rightarrow$ All diagonal entries in the positive semidefinite matrix $X^{1/2}SX^{1/2}$ are zeros $\Rightarrow X^{1/2}SX^{1/2} = 0$

$\Rightarrow (S^{1/2}X^{1/2})^T(S^{1/2}X^{1/2}) = 0 \Rightarrow S^{1/2}X^{1/2} = 0$

$\Rightarrow SX = S^{1/2}[S^{1/2}X^{1/2}]X^{1/2} = 0 \Rightarrow XS = (SX)^T = 0.$ □

$$\begin{array}{rl}
\mathrm{Opt}(P) = \min_x \left\{ c^T x : \mathcal{A}x := \sum_{j=1}^n x_j A_j \succeq B \right\} & (P) \\
\Leftrightarrow \mathrm{Opt}(\mathcal{P}) = \min_X \left\{ \mathrm{Tr}(CX) : X \in [\mathcal{L}_P - B] \cap \mathbf{S}^\nu_+ \right\} & (\mathcal{P}) \\
\mathrm{Opt}(D) = \max_S \left\{ \mathrm{Tr}(BS) : S \in [\mathcal{L}_D + C] \cap \mathbf{S}^\nu_+ \right\} & (D) \\
\left[\mathcal{L}_P = \mathrm{Im}\mathcal{A},\ \mathcal{L}_D = \mathcal{L}_P^\perp \right] &
\end{array}$$

Theorem: *Assuming (P), (D) strictly feasible, feasible solutions X for $(\mathcal{P})$ and S for (D) are optimal for the respective problems if and only if*

$$XS = SX = 0$$

("SDO Complementary Slackness").

Logarithmic Barrier for the Semidefinite Cone $\mathbf{S}^\nu_+$

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \mathcal{A}x := \sum_{j=1}^n x_j A_j \succeq B \right\} \qquad (P)$$

$$\Leftrightarrow \mathrm{Opt}(\mathcal{P}) = \min_X \left\{ \mathrm{Tr}(CX) : X \in [\mathcal{L}_P - B] \cap \mathbf{S}^\nu_+ \right\} \qquad (\mathcal{P})$$

$$\mathrm{Opt}(D) = \max_S \left\{ \mathrm{Tr}(BS) : S \in [\mathcal{L}_D + C] \cap \mathbf{S}^\nu_+ \right\} \qquad (D)$$

$$\left[\mathcal{L}_P = \mathrm{Im}\mathcal{A},\ \mathcal{L}_D = \mathcal{L}_P^\perp \right]$$

♣ A crucial role in building IPMs for (P), (D) is played by the *logarithmic barrier for the positive semidefinite cone:*

$$K(X) = -\ln \mathrm{Det}(X) : \mathrm{int}\, \mathbf{S}^\nu_+ \to \mathbb{R}$$

Back to Basic Analysis: Gradient and Hessian

♣ Consider a smooth (3 times continuously differentiable) function $f(x) : D \to \mathbb{R}$ defined on an open subset D of Euclidean space E.

♠ The *first order directional derivative of* f taken at a point $x \in D$ along a direction $h \in E$ is the quantity

$$Df(x)[h] := \frac{d}{dt}\Big|_{t=0} f(x+th)$$

Fact: *For a smooth* f, $Df(x)[h]$ *is linear in* h *and thus*

$$Df(x)[h] = \langle \nabla f(x), h \rangle \ \forall h$$

for a uniquely defined vector $\nabla f(x)$ *called the gradient of* f *at* x.

If E *is* $\mathbb{R}^n$ *with the standard Euclidean structure, then*

$$[\nabla f(x)]_i = \frac{\partial}{\partial x_i} f(x),\ 1 \le i \le n$$

♠ The *second order directional derivative* of f taken at a point $x \in D$ along a *pair* of directions g, h is defined as

$$D^2 f(x)[g,h] = \frac{d}{dt}\Big|_{t=0} [Df(x+tg)[h]]$$

Fact: *For a smooth f, $D^2 f(x)[g,h]$ is bilinear and symmetric in g, h, and therefore*

$$D^2 f(x)[g,h] = \langle g, \nabla^2 f(x) h\rangle = \langle \nabla^2 f(x) g, h\rangle \forall g, h \in E$$

for a uniquely defined linear mapping $h \mapsto \nabla^2 f(x) h : E \to E$, called the Hessian of f at x.

If E is $\mathbb{R}^n$ with the standard Euclidean structure, then

$$[\nabla^2 f(x)]_{ij} = \frac{\partial^2}{\partial x_i \partial x_j} f(x)$$

Fact: *Hessian is the derivative of the gradient:*

$$\nabla f(x+h) = \nabla f(x) + [\nabla^2 f(x)]h + R_x(h),$$
$$\|R_x(h)\| \le C_x \|h\|^2 \forall (h : \|h\| \le \rho_x), \rho_x > 0$$

Fact: *Gradient and Hessian define the second order Taylor expansion*

$$\widehat{f}(y) = f(x) + \langle y - x, \nabla f(x)\rangle + \tfrac{1}{2}\langle y - x, \nabla^2 f(x)[y-x]\rangle$$

of f at x which is a quadratic function of y with the same gradient and Hessian at x as those of f. This expansion approximates f around x, specifically,

$$|f(y) - \widehat{f}(y)| \le C_x \|y - x\|^3$$
$$\forall (y : \|y - x\| \le \rho_x), \rho_x > 0$$

Back to SDO

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \mathcal{A}x := \sum_{j=1}^n x_j A_j \succeq B \right\} \qquad (P)$$

$$\Leftrightarrow \mathrm{Opt}(\mathcal{P}) = \min_X \left\{ \mathrm{Tr}(CX) : X \in [\mathcal{L}_P - B] \cap \mathbf{S}^\nu_+ \right\} \qquad (\mathcal{P})$$

$$\mathrm{Opt}(D) = \max_S \left\{ \mathrm{Tr}(BS) : S \in [\mathcal{L}_D + C] \cap \mathbf{S}^\nu_+ \right\} \qquad (D)$$

$$\left[\mathcal{L}_P = \mathrm{Im}\mathcal{A},\ \mathcal{L}_D = \mathcal{L}_P^\perp \right]$$

$$K(X) = -\ln \mathrm{Det} X : \mathbf{S}^\nu_{++} := \{X \in \mathbf{S}^\nu : X \succ 0\} \to \mathbb{R}$$

Facts: *$K(X)$ is a smooth function on its domain $\mathbf{S}^\nu_{++} = \{X \in \mathbf{S}^\nu : X \succ 0\}$. The first- and the second order directional derivatives of this function taken at a point $X \in \mathrm{dom} K$ along a direction $H \in \mathbf{S}^\nu$ are given by*

$$\begin{array}{rcl} \frac{d}{dt}\big|_{t=0} K(X+tH) & = & -\mathrm{Tr}(X^{-1}H) \quad \left[\Leftrightarrow \nabla K(X) = -X^{-1}\right] \\ \frac{d^2}{dt^2}\big|_{t=0} K(X+tH) & = & \mathrm{Tr}(H[X^{-1}HX^{-1}]) = \mathrm{Tr}([X^{-1/2}HX^{-1/2}]^2) \end{array}$$

In particular, K is strongly convex:

$$X \in \mathrm{Dom} K, 0 \neq H \in \mathbf{S}^\nu \Rightarrow \tfrac{d^2}{dt^2}\big|_{t=0} K(X+tH) > 0$$

Proof:

$$\begin{aligned}
&\tfrac{d}{dt}\big|_{t=0}[-\ln\mathrm{Det}(X+tH)] = \tfrac{d}{dt}\big|_{t=0}[-\ln\mathrm{Det}(X[I+tX^{-1}H)]\\
&= \tfrac{d}{dt}\big|_{t=0}[-\ln\mathrm{Det}(X) - \ln\mathrm{Det}(I+tX^{-1}H)]\\
&= \tfrac{d}{dt}\big|_{t=0}[-\ln\mathrm{Det}(I+tX^{-1}H)]\\
&= -\tfrac{d}{dt}\big|_{t=0}[\mathrm{Det}(I+tX^{-1}H)] \text{ [chain rule]}\\
&= -\mathrm{Tr}(X^{-1}H)
\end{aligned}$$

$$\begin{aligned}
&\tfrac{d}{dt}\big|_{t=0}\left[-\mathrm{Tr}([X+tG]^{-1}H)\right] = \tfrac{d}{dt}\big|_{t=0}\left[-\mathrm{Tr}([X[I+tX^{-1}G]]^{-1}H)\right]\\
&\tfrac{d}{dt}\big|_{t=0}\left[-\mathrm{Tr}([I+tX^{-1}G]^{-1}X^{-1}H)\right]\\
&= -\mathrm{Tr}\left(\left[\tfrac{d}{dt}\big|_{t=0}[I+tX^{-1}G]^{-1}\right]X^{-1}H\right)\\
&= \mathrm{Tr}(X^{-1}GX^{-1}H)
\end{aligned}$$

In particular, when $X \succ 0$ and $H \in \mathbf{S}^\nu$, $H \neq 0$, we have

$$\begin{aligned}
&\tfrac{d^2}{dt^2}\big|_{t=0}K(X+tH) = \mathrm{Tr}(X^{-1}HX^{-1}H)\\
&= \mathrm{Tr}(X^{-1/2}[X^{-1/2}HX^{-1/2}]X^{-1/2}H)\\
&= \mathrm{Tr}([X^{-1/2}HX^{-1/2}]X^{-1/2}HX^{-1/2})\\
&= \langle X^{-1/2}HX^{-1/2}, X^{-1/2}HX^{-1/2}\rangle > 0.
\end{aligned}$$

Additional properties of $K(\cdot)$:

- $\nabla K(tX) = -[tX]^{-1} = -t^{-1}X^{-1} = t^{-1}\nabla K(X)$
- The mapping $X \mapsto -\nabla K(X) = X^{-1}$ maps the domain $\mathbf{S}^\nu_{++}$ of K onto itself and is self-inverse:

$$S = -\nabla K(X) \Leftrightarrow X = -\nabla K(S) \Leftrightarrow XS = SX = I$$

- The function $K(X)$ is an *interior penalty* for the positive semidefinite cone $\mathbf{S}^\nu_+$: whenever points $X_i \in \mathrm{Dom}K = \mathbf{S}^\nu_{++}$ converge to a boundary point of $\mathbf{S}^\nu_+$, one has $K(X_i) \to \infty$ as $i \to \infty$.

Primal-Dual Central Path

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \mathcal{A}x := \sum_{j=1}^n x_j A_j \succeq B \right\} \qquad (P)$$
$$\Leftrightarrow \mathrm{Opt}(\mathcal{P}) = \min_X \left\{ \mathrm{Tr}(CX) : X \in [\mathcal{L}_P - B] \cap \mathbf{S}^\nu_+ \right\} \qquad (\mathcal{P})$$
$$\mathrm{Opt}(D) = \max_S \left\{ \mathrm{Tr}(BS) : S \in [\mathcal{L}_D + C] \cap \mathbf{S}^\nu_+ \right\} \qquad (D)$$
$$K(X) = -\ln \mathrm{Det}(X)$$

Let

$$\mathcal{X} = \{X \in \mathcal{L}_P - B : X \succ 0\}$$
$$\mathcal{S} = \{S \in \mathcal{L}_D + C : S \succ 0\}.$$

be the (nonempty!) sets of strictly feasible solutions to $(\mathcal{P})$ and (D), respectively. Given *path parameter* $\mu > 0$, consider the functions

$$P_\mu(X) = \mathrm{Tr}(CX) + \mu K(X) : \mathcal{X} \to \mathbb{R}$$
$$D_\mu(S) = -\mathrm{Tr}(BS) + \mu K(S) : \mathcal{S} \to \mathbb{R} \ .$$

Fact: *For every $\mu > 0$, the function $P_\mu(X)$ achieves its minimum at $\mathcal{X}$ at a unique point $X_*(\mu)$, and the function $D_\mu(S)$ achieves its minimum on $\mathcal{S}$ at a unique point $S_*(\mu)$. These points are related to each other:*

$$X_*(\mu) = \mu S_*^{-1}(\mu) \Leftrightarrow S_*(\mu) = \mu X_*^{-1}(\mu)$$
$$\Leftrightarrow X_*(\mu) S_*(\mu) = S_*(\mu) X_*(\mu) = \mu I$$

We associate with $(\mathcal{P})$, (D) the primal-dual central path – the curve $\{X_*(\mu), S_*(\mu)\}_{\mu>0}$; for every $\mu > 0$, $X_*(\mu)$ is a strictly feasible solution to $(\mathcal{P})$, and $S_*(\mu)$ is a strictly feasible solution to (D).

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \mathcal{A}x := \sum_{j=1}^n x_j A_j \succeq B \right\} \qquad (P)$$
$$\Leftrightarrow \mathrm{Opt}(\mathcal{P}) = \min_X \left\{ \mathrm{Tr}(CX) : X \in [\mathcal{L}_P - B] \cap \mathbf{S}^\nu_+ \right\} \qquad (\mathcal{P})$$
$$\mathrm{Opt}(D) = \max_S \left\{ \mathrm{Tr}(BS) : S \in [\mathcal{L}_D + C] \cap \mathbf{S}^\nu_+ \right\} \qquad (D)$$
$$\left[\mathcal{L}_P = \mathrm{Im}\mathcal{A},\ \mathcal{L}_D = \mathcal{L}_P^\perp \right]$$

Proof of the fact: A. Let us prove that the primal-dual central path is well defined. Let $\bar{S}$ be a strictly feasible solution to (D). For every pair of feasible solutions X, X' to $(\mathcal{P})$ we have

$$\langle X - X', \bar{S} \rangle = \langle X - X', C \rangle + \langle \underbrace{X - X'}_{\in \mathcal{L}_P}, \underbrace{S - C}_{\in \mathcal{L}_D = \mathcal{L}_P^\perp} \rangle = \langle X - X', C \rangle$$

$\Rightarrow$ On the feasible plane of $(\mathcal{P})$, the linear functions $\mathrm{Tr}(CX)$ and $\mathrm{Tr}(\bar{S}X)$ of X differ by a constant

$\Rightarrow$ To prove the existence of $X_*(\mu)$ is the same as to prove that the feasible problem

$$\mathrm{Opt} = \min_{X \in \mathcal{X}} \left[\mathrm{Tr}(\bar{S}X) + \mu K(X) \right] \qquad (R)$$

is solvable.

$$\text{Opt} = \min_{X \in \mathcal{X}} \left[\text{Tr}(\bar{S}X) + \mu K(X)\right] \quad (R)$$

Let $X_i \in \mathcal{X}$ be such that

$$\left[\text{Tr}(\bar{S}X_i) + \mu K(X_i)\right] \to \text{Opt as } i \to \infty.$$

We claim that a properly selected subsequence $\{X_{i_j}\}_{j=1}^{\infty}$ of the sequence $\{X_i\}$ has a limit $\bar{X} \succ 0$.

Claim $\Rightarrow$ Solvability of (R): Since $X_{i_j} \to \bar{X} \succ 0$ as $j \to \infty$, we have $\bar{X} \in \mathcal{X}$ and

$$\text{Opt} = \lim_{j\to\infty} \left[\text{Tr}(\bar{S}X_{i_j}) + \mu K(X_{i_j})\right] = \left[\text{Tr}(\bar{S}\bar{X}) + \mu K(\bar{X})\right]$$

$\Rightarrow$ $\bar{X}$ is an optimal solution to (R).

Proof of Claim *"Let $X_i \succ 0$ be such that*

$$\lim_{i\to\infty} [\mathrm{Tr}(\bar{S}X_i) + \mu K(X_i)] < +\infty.$$

Then a properly selected subsequence of $\{X_i\}_{i=1}^{\infty}$ has a limit which is $\succ 0$":

First step: Let us prove that X_i form a bounded sequence.

Lemma: *Let $\bar{S} \succ 0$. Then there exists $c = c(\bar{S}) > 0$ such that $\mathrm{Tr}(X\bar{S}) \geq c\|X\|$ for all $X \succeq 0$.*

Indeed, there exists $\rho > 0$ such that $\bar{S} - \rho U \succeq 0$ for all $U \in \mathbf{S}^\nu$, $\|U\| \leq 1$. Therefore for every $X \succeq 0$ we have

$$\begin{array}{l}\forall(U, \|U\| \leq 1) : \mathrm{Tr}([\bar{S} - \rho U]X) \geq 0 \\ \Rightarrow \mathrm{Tr}(\bar{S}X) \geq \rho \max\limits_{U:\|U\|\leq 1} \mathrm{Tr}(UX) = \rho\|X\|.\end{array}$$

Now let X_i satisfy the premise of our claim. Then

$$\mathrm{Tr}(\bar{S}X_i) + \mu K(X_i) \geq c(\bar{S})\|X_i\| - \mu \ln(\|X_i\|^m).$$

Since the left hand side sequence is above bounded and $cr - \mu \ln(r^m) \to \infty$ as $r \to +\infty$, the sequence $\|X_i\|$ indeed is bounded.

"Let $X_i \succ 0$ be such that $\lim_{i\to\infty}\left[\mathrm{Tr}(\bar{S}X_i)+\mu K(X_i)\right] < +\infty$. Then a properly selected subsequence of $\{X_i\}_{i=1}^{\infty}$ has a limit which is $\succ 0$"

Second step: Let us complete the proof of the claim. We have seen that the sequence $\{X_i\}_{i=1}^{\infty}$ is bounded, and thus we can select from it a converging subsequence X_{i_j}. Let $\bar{X} = \lim_{j\to\infty} X_{i_j}$. If $\bar{X}$ were a boundary point of $\mathbf{S}_+^\nu$, we would have

$$\mathrm{Tr}(\bar{S}X_{i_j}) + \mu K(X_{i_j}) \to +\infty, j \to \infty$$

which is not the case. Thus, $\bar{X}$ is an interior point of $\mathbf{S}_+^\nu$, that is, $\bar{X} \succ 0$.
The existence of $S_*(\mu)$ is proved similarly, with (D) in the role of $(\mathcal{P})$.
The uniqueness of $X_*(\mu)$ and $S_*(\mu)$ follows from the fact that these points are minimizers of strongly convex functions.

B. Let us prove that $S_*(\mu) = \mu X_*^{-1}(\mu)$. Indeed, since $X_*(\mu) \succ 0$ is the minimizer of $P_\mu(X) = \mathrm{Tr}(CX) + \mu K(X)$ on $\mathcal{X} = \{X \in [\mathcal{L}_P - B] \cap \mathbf{S}^\nu_{++}\}$, the first order directional derivatives of $P_\mu(X)$ taken at $X_*(\mu)$ along directions from $\mathcal{L}_P$ should be zero, that is, $\nabla P_\mu(X_*(\mu))$ should belong to $\mathcal{L}_D = \mathcal{L}_P^\perp$. Thus,

$$C - \mu X_*^{-1}(\mu) \in \mathcal{L}_D \Rightarrow S := \mu X_*^{-1}(\mu) \in C + \mathcal{L}_D \,\&\, S \succ 0$$

$\Rightarrow S \in \mathcal{S}$. Besides this,

$$\nabla K(S) = -S^{-1} = -\mu^{-1} X_*(\mu) \Rightarrow \mu \nabla K(S) = -X_*(\mu)$$
$$\Rightarrow \mu \nabla K(S) \in -[\mathcal{L}_P - B] \Rightarrow \mu \nabla K(S) - B \in \mathcal{L}_P = \mathcal{L}_D^\perp$$

$\Rightarrow \nabla D_\mu(S)$ is orthogonal to $\mathcal{L}_D$

$\Rightarrow S$ is the minimizer of $D_\mu(\cdot)$ on $\mathcal{S} = [\mathcal{L}_D + C] \cap \mathbf{S}^\nu_{++}$.

$\Rightarrow \mu X_*^{-1}(\mu) =: S = S_*(\mu)$. $\square$

Duality Gap on the Central Path

$$\begin{array}{rl}
\mathrm{Opt}(P) = \min\limits_{x}\left\{c^Tx : \mathcal{A}x := \sum_{j=1}^n x_jA_j \succeq B\right\} & (P)\\
\Leftrightarrow \mathrm{Opt}(\mathcal{P}) = \min\limits_{X}\left\{\mathrm{Tr}(CX) : X \in [\mathcal{L}_P - B] \cap \mathbf{S}^\nu_+\right\} & (\mathcal{P})\\
\mathrm{Opt}(D) = \max\limits_{S}\left\{\mathrm{Tr}(BS) : S \in [\mathcal{L}_D + C] \cap \mathbf{S}^\nu_+\right\} & (D)
\end{array}$$

$$\Rightarrow \left\{\begin{array}{l} X_*(\mu) \in [\mathcal{L}_P - B] \cap \mathbf{S}^\nu_{++} \\ S_*(\mu) \in [\mathcal{L}_D + C] \cap \mathbf{S}^\nu_{++}\end{array}\right\} : X_*(\mu)S_*(\mu) = \mu I$$

Observation: *On the primal-dual central path, the duality gap is*

$$\mathrm{Tr}(X_*(\mu)S_*(\mu)) = \mathrm{Tr}(\mu I) = \mu m.$$

Therefore sum of non-optimalities of the strictly feasible solution $X_*(\mu)$ to $(\mathcal{P})$ and the strictly feasible solution $S_*(\mu)$ to (D) in terms of the respective objectives is equal to μm and goes to 0 as $\mu \to +0$.

$\Rightarrow$ *Our ideal goal would be to move along the primal-dual central path, pushing the path parameter μ to 0 and thus approaching primal-dual optimality, while maintaining primal-dual feasibility.*

♠ Our ideal goal is not achievable – how could we move along a curve? A *realistic* goal could be to move in a neighborhood of the primal-dual central path, staying close to it. A good notion of "closeness to the path" is given by the *proximity measure* of a triple $\mu > 0, X \in \mathcal{X}, S \in \mathcal{S}$ to the point $(X_*(\mu), S_*(\mu))$ on the path:

$$\begin{aligned}
&\mathrm{dist}(X,S,\mu) = \sqrt{\mathrm{Tr}(X[X^{-1}-\mu^{-1}S]X[X^{-1}-\mu^{-1}S])}\\
&= \sqrt{\mathrm{Tr}(X^{1/2}[X^{1/2}[X^{-1}-\mu^{-1}S]X^{1/2}]\left[X^{1/2}[X^{-1}-\mu^{-1}S]\right])}\\
&= \sqrt{\mathrm{Tr}([X^{1/2}[X^{-1}-\mu^{-1}S]X^{1/2}]\left[X^{1/2}[X^{-1}-\mu^{-1}S]X^{1/2}\right])}\\
&= \sqrt{\mathrm{Tr}([X^{1/2}[X^{-1}-\mu^{-1}S]X^{1/2}]^2)}\\
&= \sqrt{\mathrm{Tr}([I-\mu^{-1}X^{1/2}SX^{1/2}]^2)}.
\end{aligned}$$

Note: We see that $\mathrm{dist}(X,S,\mu)$ is well defined and $\mathrm{dist}(X,S,\mu) = 0$ iff $X^{1/2}SX^{1/2} = \mu I$, or, which is the same,

$$SX = X^{-1/2}[X^{1/2}SX^{1/2}]X^{1/2} = \mu X^{-1/2}X^{1/2} = \mu I,$$

i.e., iff $X = X_*(\mu)$ and $S = S_*(\mu)$.

Note: We have

$$\begin{aligned}
&\mathrm{dist}(X,S,\mu) = \sqrt{\mathrm{Tr}(X[X^{-1}-\mu^{-1}S]X[X^{-1}-\mu^{-1}S])}\\
&= \sqrt{\mathrm{Tr}([I-\mu^{-1}XS][I-\mu^{-1}XS])}\\
&= \sqrt{\mathrm{Tr}(\left[[I-\mu^{-1}XS][I-\mu^{-1}XS]\right]^T)}\\
&= \sqrt{\mathrm{Tr}([I-\mu^{-1}SX][I-\mu^{-1}SX])}\\
&= \sqrt{\mathrm{Tr}(S[S^{-1}-\mu^{-1}X]S[S^{-1}-\mu^{-1}X])},
\end{aligned}$$

$\Rightarrow$ The proximity is defined in a symmetric w.r.t. X, S fashion.

Fact: *Whenever $X \in \mathcal{X}$, $S \in \mathcal{S}$ and $\mu > 0$, one has*

$$\mathrm{Tr}(XS) \le \mu[m + \sqrt{m}\mathrm{dist}(X, S, \mu)]$$

Indeed, we have seen that

$$d := \mathrm{dist}(X, S, \mu) = \sqrt{\mathrm{Tr}([I - \mu^{-1}X^{1/2}SX^{1/2}]^2)}.$$

Denoting by λ_i the eigenvalues of $X^{1/2}SX^{1/2}$, we have

$$\begin{array}{rl} & d^2 = \mathrm{Tr}([I - \mu^{-1}X^{1/2}SX^{1/2}]^2) = \sum_i [1 - \mu^{-1}\lambda_i]^2 \\ \Rightarrow & \sum_i |1 - \mu^{-1}\lambda_i| \le \sqrt{m}\sqrt{\sum_i [1 - \mu^{-1}\lambda_i]^2} \\ & = \sqrt{m}d \\ \Rightarrow & \sum_i \lambda_i \le \mu[m + \sqrt{m}d] \\ \Rightarrow & \mathrm{Tr}(XS) = \mathrm{Tr}(X^{1/2}SX^{1/2}) = \sum_i \lambda_i \le \mu[m + \sqrt{m}d] \end{array}$$

Corollary. Let us say that a triple (X, S, μ) is *close to the path*, if $X \in \mathcal{X}$, $S \in \mathcal{S}$, $\mu > 0$ and $\mathrm{dist}(X, S, \mu) \le 0.1$. Whenever (X, S, μ) is close to the path, one has

$$\mathrm{Tr}(XS) \le 2\mu m,$$

that is, if (X, S, μ) is close to the path, then X is at most $2\mu m$-nonoptimal strictly feasible solution to $(\mathcal{P})$, and S is at most $2\mu m$-nonoptimal strictly feasible solution to (D).

How to Trace the Central Path?

♣ **The goal:** To follow the central path, staying close to it and pushing μ to 0 as fast as possible.

♣ **Question.** *Assume we are given a triple $(\bar{X}, \bar{S}, \bar{\mu})$ close to the path. How to update it into a triple (X_+, S_+, μ_+), also close to the path, with $\mu_+ < \mu$?*

♠ **Conceptual answer:** Let us choose μ_+, $0 < \mu_+ < \bar{\mu}$, and try to update $\bar{X}, \bar{S}$ into $X_+ = \bar{X} + \Delta X$, $S_+ = \bar{S} + \Delta S$ in order to make the triple (X_+, S_+, μ_+) close to the path. Our goal is to ensure that

$$\begin{array}{rcl} X_+ = \bar{X} + \Delta X & \in & \mathcal{L}_P - B \quad \& \quad X_+ \succ 0 \quad (a) \\ S_+ = \bar{S} + \Delta S & \in & \mathcal{L}_D + C \quad \& \quad S_+ \succ 0 \quad (b) \\ G_{\mu_+}(X_+, S_+) \approx 0 & & \qquad\qquad\qquad\quad (c) \end{array}$$

where $G_\mu(X, S) = 0$ expresses equivalently the *augmented slackness* condition $XS = \mu I$. For example, we can take

$$\begin{array}{l} G_\mu(X, S) = S - \mu X^{-1}, \text{ or} \\ G_\mu(X, S) = X - \mu S^{-1}, \text{ or} \\ G_\mu(X, S) = XS + SX - 2\mu I, \text{ or...} \end{array}$$

$$
\begin{array}{rcl}
X_+ = \bar{X} + \Delta X & \in & \mathcal{L}_P - B \quad \& \quad X_+ \succ 0 \quad (a) \\
S_+ = \bar{S} + \Delta S & \in & \mathcal{L}_D + C \quad \& \quad S_+ \succ 0 \quad (b) \\
G_{\mu_+}(X_+, S_+) \approx 0 & & \qquad\qquad\qquad\quad (c)
\end{array}
$$

♠ Since $\bar{X} \in \mathcal{L}_P - B$ and $\bar{X} \succ 0$, (a) amounts to $\Delta X \in \mathcal{L}_P$, which is a system of linear equations on ΔX, and to $\bar{X} + \Delta X \succ 0$. Similarly, (b) amounts to the system $\Delta S \in \mathcal{L}_D$ of linear equations on ΔS, and to $\bar{S} + \Delta S \succ 0$. To handle the troublemaking *nonlinear in* $\Delta X, \Delta S$ condition (c), we *linearize* G_{μ_+} *in* ΔX *and* ΔS*:*

$$
\begin{array}{l}
G_{\mu_+}(X_+, S_+) \approx G_{\mu_+}(\bar{X}, \bar{S}) \\
+ \left.\frac{\partial G_{\mu_+}(X,S)}{\partial X}\right|_{(X,S)=(\bar{X},\bar{S})} \Delta X + \left.\frac{\partial G_{\mu_+}(X,S)}{\partial S}\right|_{(X,S)=(\bar{X},\bar{S})} \Delta S
\end{array}
$$

and enforce the lumbarization, as evaluated at ΔX*,* ΔS*, to be zero.* We arrive at the *Newton system*

$$
\left\{
\begin{array}{l}
\Delta X \in \mathcal{L}_P, \;\; \Delta S \in \mathcal{L}_D \\
\frac{\partial G_{\mu_+}}{\partial X} \Delta X + \frac{\partial G_{\mu_+}}{\partial S} \Delta S = -G_{\mu_+}
\end{array}
\right. \qquad (N)
$$

(the value and the partial derivatives of $G_{\mu_+}(X, S)$ are taken at the point $(\bar{X}, \bar{S})$).

♠ We arrive at conceptual *primal-dual path-following method* where one iterates the updatings

$$(X_i, S_i, \mu_i) \mapsto (X_{i+1} = X_i + \Delta X_i, S_{i+1} = S_i + \Delta S_i, \mu_{i+1})$$

where $\mu_{i+1} \in (0, \mu_i)$ and $\Delta X_i, \Delta S_i$ are the solution to the Newton system

$$\begin{cases} \Delta X_i \in \mathcal{L}_P, \ \ \Delta S_i \in \mathcal{L}_D \\ \frac{\partial G^{(i)}_{\mu_{i+1}}}{\partial X} \Delta X_i + \frac{\partial G^{(i)}_{\mu_{i+1}}}{\partial S} \Delta S_i = -G^{(i)}_{\mu_{i+1}} \end{cases} \qquad (N_i)$$

and $G^{(i)}_\mu(X, S) = 0$ represents equivalently the augmented complementary slackness condition $XS = \mu I$ and the value and the partial derivatives of $G^{(i)}_{\mu_{i+1}}$ are evaluated at (X_i, S_i).

♠ Initialized by a close to the path triple (X_0, S_0, μ_0), this conceptual algorithm should

- be well-defined: (N_i) should remain solvable, X_i should remain strictly feasible for $(\mathcal{P})$, S_i should remain strictly feasible for (D), and
- maintain closeness to the path: for every i, (X_i, S_i, μ_i) should remain close to the path.

Under these limitations, we want to push μ_i to 0 as fast as possible.

Example: Primal Path-Following Method

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \mathcal{A}x := \sum_{j=1}^n x_j A_j \succeq B \right\} \qquad (P)$$
$$\Leftrightarrow \mathrm{Opt}(\mathcal{P}) = \min_X \left\{ \mathrm{Tr}(CX) : X \in [\mathcal{L}_P - B] \cap \mathbf{S}^\nu_+ \right\} \qquad (\mathcal{P})$$
$$\mathrm{Opt}(D) = \max_S \left\{ \mathrm{Tr}(BS) : S \in [\mathcal{L}_D + C] \cap \mathbf{S}^\nu_+ \right\} \qquad (D)$$
$$\left[\mathcal{L}_P = \mathrm{Im}\mathcal{A},\ \mathcal{L}_D = \mathcal{L}_P^\perp \right]$$

♣ Let us choose

$$G_\mu(X, S) = S + \mu \nabla K(X) = S - \mu X^{-1}$$

Then the Newton system becomes

$$\begin{array}{rcl} \Delta X_i \in \mathcal{L}_P & \Leftrightarrow & \Delta X_i = \mathcal{A}\Delta x_i \\ \Delta S_i \in \mathcal{L}_D & \Leftrightarrow & \mathcal{A}^* \Delta S_i = 0 \\ & & \mathcal{A}^* U = [\mathrm{Tr}(A_1 U); ...; \mathrm{Tr}(A_n U)] \end{array} \qquad (N_i)$$

$$(!) \quad \Delta S_i + \mu_{i+1} \nabla^2 K(X_i) \Delta X_i = -[S_i + \mu_{i+1} \nabla K(X_i)]$$

♠ Substituting $\Delta X_i = \mathcal{A}\Delta x_i$ and applying $\mathcal{A}^*$ to both sides in (!), we get

$$(*) \quad \mu_{i+1} \underbrace{[\mathcal{A}^* \nabla^2 K(X_i) \mathcal{A}]}_{\mathcal{H}} \Delta x_i = -\big[\underbrace{\mathcal{A}^* S_i}_{=c} + \mathcal{A}^* \nabla K(X_i) \big]$$

$$\Delta X_i = \mathcal{A}\Delta x_i$$
$$S_{i+1} = \mu_{i+1} \left[\nabla K(X_i) - \nabla^2 K(X_i) \mathcal{A}\Delta x_i \right]$$

The mappings $h \mapsto \mathcal{A}h$, $H \mapsto \nabla^2 K(X_i) H$ have trivial kernels $\Rightarrow$ $\mathcal{H}$ is nonsingular $\Rightarrow$ (N_i) has a unique solution given by

$$\Delta x_i = -\mathcal{H}^{-1} \left[\mu_{i+1}^{-1} c + \mathcal{A}^* \nabla K(X_i) \right]$$
$$\Delta X_i = \mathcal{A}\Delta x_i$$
$$S_{i+1} = S_i + \Delta S_i = \mu_{i+1} \left[\nabla K(X_i) - \nabla^2 K(X_i) \mathcal{A}\Delta x_i \right]$$

$$\begin{array}{rc}
\mathrm{Opt}(P) = \min_x \left\{ c^T x : \mathcal{A}x := \sum_{j=1}^n x_j A_j \succeq B \right\} & (P) \\
\Leftrightarrow \mathrm{Opt}(\mathcal{P}) = \min_X \left\{ \mathrm{Tr}(CX) : X \in [\mathcal{L}_P - B] \cap \mathbf{S}_+^\nu \right\} & (\mathcal{P}) \\
\mathrm{Opt}(D) = \max_S \left\{ \mathrm{Tr}(BS) : S \in [\mathcal{L}_D + C] \cap \mathbf{S}_+^\nu \right\} & (D)
\end{array}$$

$$\left[\mathcal{L}_P = \mathrm{Im}\mathcal{A},\ \mathcal{L}_D = \mathcal{L}_P^\perp\right]$$

$$\Rightarrow \left\{ \begin{array}{l}
\Delta x_i = -\mathcal{H}^{-1} \left[\mu_{i+1}^{-1} c + \mathcal{A}^* \nabla K(X_i)\right] \\
\Delta X_i = \mathcal{A}\Delta x_i \\
S_{i+1} = S_i + \Delta S_i = \mu_{i+1} \left[\nabla K(X_i) - \nabla^2 K(X_i)\mathcal{A}\Delta x_i\right]
\end{array} \right.$$

♠ $X_i = \mathcal{A}x_i - B$ for a (uniquely defined by X_i) strictly feasible solution x_i to (P). Setting

$$F(x) = K(\mathcal{A}x - B),$$

we have $\mathcal{A}^*\nabla K(X_i) = \nabla F(x_i),\ \mathcal{H} = \nabla^2 F(x_i)$

⇒ *The above recurrence can be written solely in terms of x_i and F:*

$$(\#) \quad \left\{ \begin{array}{l}
\mu_i \mapsto \mu_{i+1} < \mu_i \\
x_{i+1} = x_i - [\nabla^2 F(x_i)]^{-1} \left[\mu_{i+1}^{-1} c + \nabla F(x_i)\right]
\end{array} \right.$$
$$X_{i+1} = \mathcal{A}x_{i+1} - B$$
$$S_{i+1} = \mu_{i+1} \left[\nabla K(X_i) - \nabla^2 K(X_i)\mathcal{A}[x_{i+1} - x_i]\right]$$

Recurrence (#) is called the *primal path-following method.*

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \mathcal{A}x := \sum_{j=1}^n x_j A_j \succeq B \right\} \qquad (P)$$
$$\Leftrightarrow \mathrm{Opt}(\mathcal{P}) = \min_X \left\{ \mathrm{Tr}(CX) : X \in [\mathcal{L}_P - B] \cap \mathbf{S}^\nu_+ \right\} \qquad (\mathcal{P})$$
$$\mathrm{Opt}(D) = \max_S \left\{ \mathrm{Tr}(BS) : S \in [\mathcal{L}_D + C] \cap \mathbf{S}^\nu_+ \right\} \qquad (D)$$
$$\left[\mathcal{L}_P = \mathrm{Im}\mathcal{A},\ \mathcal{L}_D = \mathcal{L}_P^\perp\right]$$

♠ The primal path-following method can be explained as follows:

• The barrier $K(X) = -\ln \mathrm{Det} X$ induces the barrier $F(x) = K(\mathcal{A}x - B)$ for the interior P^o of the feasible domain of (P).

• The primal central path

$$X_*(\mu) = \mathrm{argmin}_{X = \mathcal{A}x - B \succ 0} \left[\mathrm{Tr}(CX) + \mu K(X)\right]$$

induces the path

$$x_*(\mu) \in P^o:\ X_*(\mu) = \mathcal{A}x_*(\mu) + \mu F(x).$$

Observing that

$$\mathrm{Tr}(C[\mathcal{A}x - B]) + \mu K(\mathcal{A}x - B) = c^T x + \mu F(x) + \mathrm{const},$$

we have

$$x_*(\mu) = \mathrm{argmin}_{x \in P^o} F_\mu(x),\ F_\mu(x) = c^T x + \mu F(x).$$

• The method works as follows: given $x_i \in P^o, \mu_i > 0$, we

— replace μ_i with $\mu_{i+1} < \mu_i$

— convert x_i into x_{i+1} by applying to the function $F_{\mu_{i+1}}(\cdot)$ a single step of the *Newton minimization method*

$$x_i \mapsto x_{i+1} - [\nabla^2 F_{\mu_{i+1}}(x_i)]^{-1} \nabla F_{\mu_{i+1}}(x_i)$$

$$\begin{array}{rl}
\mathrm{Opt}(P) = \min\limits_x \left\{c^T x : \mathcal{A}x := \sum_{j=1}^n x_j A_j \succeq B\right\} & (P)\\
\Leftrightarrow \mathrm{Opt}(\mathcal{P}) = \min\limits_X \left\{\mathrm{Tr}(CX) : X \in [\mathcal{L}_P - B] \cap \mathbf{S}^\nu_+\right\} & (\mathcal{P})\\
\mathrm{Opt}(D) = \max\limits_S \left\{\mathrm{Tr}(BS) : S \in [\mathcal{L}_D + C] \cap \mathbf{S}^\nu_+\right\} & (D)\\
\left[\mathcal{L}_P = \mathrm{Im}\mathcal{A},\ \mathcal{L}_D = \mathcal{L}_P^\perp\right] &
\end{array}$$

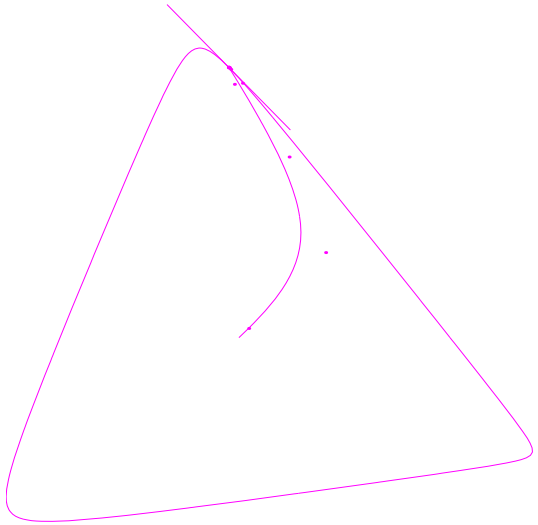
Theorem. *Let $(X_0 = \mathcal{A}x_0 - B, S_0, \mu_0)$ be close to the primal-dual central path, and let (P) be solved by the Primal path-following method where the path parameter μ is updated according to*

$$\mu_{i+1} = \left(1 - \frac{0.1}{\sqrt{m}}\right)\mu_i. \qquad (*)$$

Then the method is well defined and all triples $(X_i = \mathcal{A}x_i - B, S_i, \mu_i)$ are close to the path.

♠ With the rule $(*)$ it takes $O(\sqrt{m})$ steps to reduce the path parameter μ by an absolute constant factor. Since the method stays close to the path, the duality gap $\mathrm{Tr}(X_i S_i)$ of i-th iterate does not exceed $2m\mu_i$.

⇒ *The number of steps to make the duality gap $\leq \epsilon$ does not exceed* $O(1)\sqrt{m}\ln\left(1 + \frac{2m\mu_0}{\epsilon}\right)$.



2D feasible set of a toy SDO ($\mathbf{K} = \mathbf{S}^3_+$).
"Continuous curve" is the primal central path
Dots are iterates x_i of the Primal Path-Following method.

Itr#	Objective	Gap	Itr#	Objective	Gap
1	-0.100000	2.96	7	-1.359870	8.4e-4
2	-0.906963	0.51	8	-1.360259	2.1e-4
3	-1.212689	0.19	9	-1.360374	5.3e-5
4	-1.301082	6.9e-2	10	-1.360397	1.4e-5
5	-1.349584	2.1e-2	11	-1.360404	3.8e-6
6	-1.356463	4.7e-3	12	-1.360406	9.5e-7

Duality gap along the iterations

♣ The Primal path-following method is yielded by Conceptual Path-Following Scheme when the Augmented Complementary Slackness condition is represented as

$$G_\mu(X,S) := S + \mu\nabla K(X) = 0.$$

Passing to the representation

$$G\mu(X,S) := X + \mu\nabla K(S) = 0,$$

we arrive at the *Dual path-following method* with the same theoretical properties as those of the primal method. the Primal and the Dual path-following methods imply the best known so far complexity bounds for LO and SDO.

♠ In spite of being "theoretically perfect", Primal and Dual path-following methods in practice are inferior as compared with the methods based on less straightforward and more symmetric forms of the Augmented Complementary Slackness condition.

♠ The Augmented Complementary Slackness condition is

$$XS = SX = \mu I \qquad (*)$$

Fact: For $X, S \in \mathbf{S}^\nu_{++}$, $(*)$ is equivalent to

$$XS + SX = 2\mu I$$

Indeed, if $XS = SX = \mu I$, then clearly $XS + SX = 2\mu I$. On the other hand,

$$\begin{array}{l} X, S \succ 0, XS + SX = 2\mu I \\ \Rightarrow S + X^{-1}SX = 2\mu X^{-1} \\ \Rightarrow X^{-1}SX = 2\mu X^{-1} - S \\ \Rightarrow X^{-1}SX = [X^{-1}SX]^T = XSX^{-1} \\ \Rightarrow X^2S = SX^2 \end{array}$$

We see that $X^2S = SX^2$. Since $X \succ 0$, X is a polynomial of X^2, whence X and S commute, whence $XS = SX = \mu I$. □

Fact: Let $Q \in \mathbf{S}^\nu$ be nonsingular, and let $X, S \succ 0$. Then $XS = \mu I$ if and only if

$$QXSQ^{-1} + Q^{-1}SXQ = 2\mu I$$

Indeed, it suffices to apply the previous fact to the matrices $\widehat{X} = QXQ \succ 0$, $\widetilde{S} = Q^{-1}SQ^{-1} \succ 0$. □

♠ In practical path-following methods, at step i the Augmented Complementary Slackness condition is written down as

$$G_{\mu_{i+1}}(X,S) := Q_i X S Q_i^{-1} + Q_i^{-1} S X Q_i - 2\mu_{i+1} I = 0$$

with properly chosen varying from step to step nonsingular matrices $Q_i \in \mathbf{S}^\nu$.

Explanation: Let $Q \in \mathbf{S}^\nu$ be nonsingular. The *Q-scaling* $X \mapsto QXQ$ is a one-to-one linear mapping of $\mathbf{S}^\nu$ onto itself, the inverse being the mapping $X \mapsto Q^{-1}XQ^{-1}$. *Q-scaling is a symmetry of the positive semidefinite cone – it maps the cone onto itself.*

⇒ Given a primal-dual pair of semidefinite programs

$$\mathrm{Opt}(\mathcal{P}) = \min_X \left\{ \mathrm{Tr}(CX) : X \in [\mathcal{L}_P - B] \cap \mathbf{S}^\nu_+ \right\} \quad (\mathcal{P})$$

$$\mathrm{Opt}(\mathcal{D}) = \max_S \left\{ \mathrm{Tr}(BS) : S \in [\mathcal{L}_D + C] \cap \mathbf{S}^\nu_+ \right\} \quad (\mathcal{D})$$

and a nonsingular matrix $Q \in \mathbf{S}^\nu$, one can pass in $(\mathcal{P})$ from variable X to variables $\widehat{X} = QXQ$, while passing in $(\mathcal{D})$ from variable S to variable $\widetilde{S} = Q^{-1}SQ^{-1}$. The resulting problems are

$$\mathrm{Opt}(\mathcal{P}) = \min_{\widehat{X}} \left\{ \mathrm{Tr}(\widetilde{C}\widehat{X}) : \widehat{X} \in [\widehat{\mathcal{L}}_P - \widehat{B}] \cap \mathbf{S}^\nu_+ \right\} \quad (\widehat{\mathcal{P}})$$

$$\mathrm{Opt}(\mathcal{D}) = \max_{\widetilde{S}} \left\{ \mathrm{Tr}(\widehat{B}\widetilde{S}) : \widetilde{S} \in [\widetilde{\mathcal{L}}_D + \widetilde{C}] \cap \mathbf{S}^\nu_+ \right\} \quad (\widetilde{\mathcal{D}})$$

$$\left[\begin{array}{c} \widehat{B} = QBQ, \widehat{\mathcal{L}}_P = \{QXQ : X \in \mathcal{L}_P\}, \\ \widetilde{C} = Q^{-1}CQ^{-1}, \widetilde{\mathcal{L}}_D = \{Q^{-1}SQ^{-1} : S \in \mathcal{L}_D\} \end{array} \right]$$

$$\mathrm{Opt}(\mathcal{P}) = \min_{\widehat{X}} \left\{ \mathrm{Tr}(\widetilde{C}\widehat{X}) : \widehat{X} \in [\widehat{\mathcal{L}}_P - \widehat{B}] \cap \mathbf{S}^\nu_+ \right\} \quad (\widehat{\mathcal{P}})$$

$$\mathrm{Opt}(\mathcal{D}) = \max_{\widetilde{S}} \left\{ \mathrm{Tr}(\widehat{B}\widetilde{S}) : \widetilde{S} \in [\widetilde{\mathcal{L}}_D + \widetilde{C}] \cap \mathbf{S}^\nu_+ \right\} \quad (\widetilde{\mathcal{D}})$$

$$\left[\begin{array}{c} \widehat{B} = QBQ, \widehat{\mathcal{L}}_P = \{QXQ : X \in \mathbf{L}_P\}, \\ \widetilde{C} = Q^{-1}CQ^{-1}, \widetilde{\mathcal{L}}_D = \{Q^{-1}SQ^{-1} : S \in \mathcal{L}_D\} \end{array} \right]$$

$\widehat{\mathcal{P}}$ and $\widetilde{\mathcal{D}}$ are dual to each other, the primal-dual central path of this pair is the image of the primal-dual path of $(\mathcal{P})$, $(\mathcal{D})$ under the *primal-dual Q-scaling*

$$(X, S) \mapsto (\widehat{X} = QXQ, \widetilde{S} = Q^{-1}SQ^{-1})$$

$\mathcal{Q}$ preserves closeness to the path, etc.

Writing down the Augmented Complementary Slackness condition as

$$QXSQ^{-1} + Q^{-1}SXQ = 2\mu I \qquad (!)$$

we in fact

- pass from $(\mathcal{P})$, $(\mathcal{D})$ to the equivalent primal-dual pair of problems $(\widehat{\mathcal{P}})$, $(\widetilde{\mathcal{D}})$
- write down the Augmented Complementary Slackness condition for the latter pair in the simplest primal-dual symmetric form

$$\widehat{X}\widetilde{S} + \widetilde{S}\widehat{X} = 2\mu I,$$

- "scale back" to the original primal-dual variables X, S, thus arriving at (!).

Note: In the LO case $\mathbf{S}^\nu$ is composed of diagonal matrices, so that (!) is exactly the same as the "unscaled" condition $XS = \mu I$.

$$G_{\mu_{i+1}}(X,S) := Q_i X S Q_i^{-1} + Q_i^{-1} S X Q_i - 2\mu_{i+1} I = 0 \qquad (!)$$

With (!), the Newton system becomes

$$\Delta X \in \mathcal{L}_P,\ \Delta S \in \mathcal{L}_D$$
$$Q_i \Delta X S_i Q_i^{-1} + Q_i^{-1} S_i \Delta X Q_i + Q_i X_i \Delta S Q_i^{-1} + Q_i^{-1} \Delta S X_i Q_i$$
$$= 2\mu_{i+1} I - Q_i X_i S_i Q_i^{-1} - Q_i^{-1} S_i X_i Q_i$$

♣ Theoretical analysis of path-following methods simplifies a lot when the scaling (!) is *commutative*, meaning that the matrices $\widehat{X}_i = Q_i X_i Q_i$ and $\widehat{S}_i = Q_i^{-1} S_i Q_i^{-1}$ commute.

Popular choices of commuting scalings are:

- $Q_i = S_i^{1/2}$ ("XS-method," $\widetilde{S} = I$)
- $Q_i = X_i^{-1/2}$ ("SX-method, $\widehat{X} = I$)
- $Q_i = \left(X^{-1/2}(X^{1/2} S X^{1/2})^{-1/2} X^{1/2} S\right)^{1/2}$

(famous *Nesterov-Todd* method, $\widehat{X} = \widetilde{S}$).

$$\begin{array}{rl}
\mathrm{Opt}(P) = \min_x \left\{ c^T x : \mathcal{A}x := \sum_{j=1}^n x_j A_j \succeq B \right\} & (P) \\
\Leftrightarrow \mathrm{Opt}(\mathcal{P}) = \min_X \left\{ \mathrm{Tr}(CX) : X \in [\mathcal{L}_P - B] \cap \mathbf{S}^\nu_+ \right\} & (\mathcal{P}) \\
\mathrm{Opt}(D) = \max_S \left\{ \mathrm{Tr}(BS) : S \in [\mathcal{L}_D + C] \cap \mathbf{S}^\nu_+ \right\} & (D) \\
\left[\mathcal{L}_P = \mathrm{Im}\mathcal{A},\ \mathcal{L}_D = \mathcal{L}_P^\perp \right] &
\end{array}$$

Theorem: *Let a strictly-feasible primal-dual pair (P), (D) of semidefinite programs be solved by a primal-dual path-following method based on commutative scalings. Assume that the method is initialized by a close to the path triple $(X_0, S_0, \mu_0 = \mathrm{Tr}(X_0 S_0)/m)$ and let the policy for updating μ be*

$$\mu_{i+1} = \left(1 - \frac{0.1}{\sqrt{m}}\right) \mu_i.$$

The the trajectory is well defined and stays close to the path.
As a result, every $O(\sqrt{m})$ steps of the method reduce duality gap by an absolute constant factor, and it takes $O(1)\sqrt{m}\,\mathrm{In}\left(1 + \frac{m\mu_0}{\epsilon}\right)$ steps to make the duality gap $\leq \epsilon$.

♠ To improve the practical performance of primal-dual path-following methods, in actual computations

• the path parameter is updated in a more aggressive fashion than $\mu \mapsto \left(1 - \frac{0.1}{\sqrt{m}}\right)\mu$;

• the method is allowed to travel in a wider neighborhood of the primal-dual central path than the neighborhood given by our "close to the path" restriction $\mathrm{dist}(X, S, \mu) \leq 0.1$;

• instead of updating $X_{i+1} = X_i + \Delta X_i$, $S_{i+1} = S_i + \Delta S_i$, one uses the more flexible updating

$$X_{i+1} = X_i + \alpha_i \Delta X_i, \; S_{i+1} = S_i + \alpha_i \Delta S_i$$

with α_i given by appropriate line search.

♣ The constructions and the complexity results we have presented are incomplete — they do not take into account the necessity to come close to the central path before starting path-tracing and do not take care of the case when the pair (P), (D) is not strictly feasible. All these "gaps" can be easily closed via the same path-following technique as applied to appropriate augmented versions of the problem of interest.

SUMMARY
I. WHAT IS LO

♣ **An LO program** is an optimization problem of the form

> Optimize a *linear objective* $c^T x$ over $x \in \mathbb{R}^n$ satisfying a system (S) of finitely many linear equality and (nonstrict) inequality constraints.

♠ There are several "universal" forms of an LO program, universality meaning that every LO program can be straightforwardly converted to an equivalent problem in the desired form.
Examples of universal forms are:

- **Canonical form:**

$$\mathrm{Opt} = \max_x \left\{ c^T x : Ax - b \geq 0 \right\} \qquad [A : m \times n]$$

- **Standard form:**

$$\mathrm{Opt} = \max_x \left\{ c^T x : Ax = b, x \geq 0 \right\} \qquad [A : m \times n]$$

EXTENSION: CONIC PROBLEMS

♣ An LO program is

$$\mathrm{Opt} = \max_x \left\{ c^T x : Ax - b \in \mathbb{R}^m_+ \right\}$$
$$\mathbb{R}^m_+ = \{x \in \mathbb{R}^m : x_i \geq 0,\ 1 \leq i \leq m\}$$

Note: *The nonnegative orthant $\mathbb{R}^m_+$ is a regular cone* – closed convex pointed cone with a nonempty interior.

♠ Replacing in the definition of an LO the nonnegative orthant with another regular cone $\mathbf{K}$, we arrive at a *conic problem on a cone* $\mathbf{K}$*:*

$$\mathrm{Opt} = \max_x \left\{ c^T x : Ax - b \in \mathbf{K} \right\}$$

In this problem,

- c, A, b *form problem's data*
- $\mathbf{K}$ *"summarizes" problem's structure.*

♠ *Essentially, the entire Convex Programming is covered by just 3 generic conic problems:*

- $\mathcal{LO}$ *– cones* $\mathbf{K}$ *are nonnegative orthants – direct products of nonnegative rays,*
- $\mathcal{CQO}$ *– cones* $\mathbf{K}$ *are direct products of Lorentz cones,*
- $\mathcal{SDO}$ *– cones* $\mathbf{K}$ *are direct products of cones of symmetric positive semidefinite matrices*

- **Note:**

$$\boxed{\mathcal{LO} \subset \mathcal{CQO} \subset \mathcal{SDO}}$$

and CQO can be approximated *in a polynomial time fashion* by LO.

II. WHAT CAN BE REDUCED TO LO

♣ Every optimization problem $\max\limits_{y\in Y} f(y)$ can be straightforwardly converted to an equivalent problem with *linear* objective, specifically, to the problem

$$\max_{x\in X} c^T x \qquad (P)$$
$$\left[X = \{x = [y;t] : y \in Y, t \le f(y)\}, c^T x := t\right]$$

♠ The possibility to convert (P) into an LO program depends on the geometry of the feasible set X. When X is *polyhedrally representable:*

$$X = \{x \in \mathbb{R}^n : \exists w \in \mathbb{R}^k : Px + Qw - r \ge 0\}$$

for properly chosen P, Q, r, (P) reduces to the LO program

$$\max_{x,w} \left\{c^T x : Px + Qw - r \ge 0\right\}.$$

♣ Similarly, given a family $\mathcal{K}$ of regular cones closed w.r.t. taking direct products and passing from a cone to its dual, we can define the notion of *$\mathcal{K}$-representation* of a set X:

$$X = \{x \in \mathbb{R}^n : \exists w \in \mathbb{R}^k : Px + Qw - r \in \mathbf{K}\}$$

with $\mathbf{K} \in \mathcal{K}$. Given such a representation, (P) reduces to the conic problem

$$\max_{x,w} \left\{c^T x : Px + Qw - r \in \mathbf{K}\right\}.$$

on a cone from family $\mathcal{K}$.

♣ By Fourier-Motzkin elimination scheme, *every polyhedrally representable set X is in fact polyhedral* – it can be represented as the solution set of a system of linear inequalities *without slack variables* w:

$$X = \{x : \exists w : Px + Qw - r \geq 0\}$$
$$\Leftrightarrow \exists A, b : X = \{x : Ax - b \geq 0\}$$

This does not make the notion of a polyhedral representation "void" — a polyhedral set with a "short" polyhedral representation involving slack variables can require astronomically many inequalities in a slack-variable-free representation.

♠ *There is no "Fourier-Motzkin elimination" for general $\mathcal{K}$-representability: it may happen that a set which admits conic quadratic (or semidefinite) representation utilizing slack variables does not admit such a representation without slack variables.*

♠ *Every polyhedral* ($\equiv$ polyhedrally representable) *set is convex* (but not vice versa), and *all basic convexity-preserving operations with sets* (taking finite intersections, affine images, inverse affine images, arithmetic sums, direct products) *as applied to polyhedral operands yield polyhedral results.*

• Moreover, *for all basic convexity-preserving operations, a polyhedral representation of the result is readily given by polyhedral representations of the operands.*

♠ All this holds true for $\mathcal{K}$-representations. *"The calculus" of representations via a particular family of regular cones $\mathcal{K}$* (closed w.r.t. taking direct products and duality) *is completely independent of $\mathcal{K}$; what does depend on $\mathcal{K}$, are the "raw materials."*

- **Example:** The problem

	minimize $\sum_{\ell=1}^{n} x_\ell$
(a)	$x \geq 0;$
(b)	$a_\ell^T x \leq b_\ell,\ \ell = 1, ..., n;$
(c)	$\|Px - p\|_2 \leq c^T x + d;$
(d)	$x_\ell^{\frac{\ell+1}{\ell}} \leq e_\ell^T x + f_\ell,\ \ell = 1, ..., n;$
(e)	$x_\ell^{\frac{\ell}{\ell+3}} x_{l+1}^{\frac{1}{\ell+3}} \geq g_\ell^T x + h_\ell,\ \ell = 1, ..., n-1;$
(f)	$\mathrm{Det}\begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_2 & x_1 & x_2 & \cdots & x_{n-1} \\ x_3 & x_2 & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n & x_{n-1} & x_{n-2} & \cdots & x_1 \end{bmatrix} \geq 1;$
(g)	$1 \leq \sum_{\ell=1}^{n} x_\ell \cos(\ell\omega) \leq 1 + \sin^2(5\omega)\ \forall \omega \in \left[-\frac{\pi}{7}, 1.3\right]$

can be converted, *in a systematic way*, into an equivalent SDO problem

$$\min_x \left\{ c^T x : Ax - b \succeq 0 \right\}.$$

Dropping red constraints (f), (g), the remaining problem can be converted into an equivalent CQO. Further dropping magenta constraints (c), (d), (e), we arrive at LO problem.

♣ A "counterpart" of the notion of polyhedral ($\equiv$ polyhedrally representable) set is the notion of a *polyhedrally representable function*. A function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is called *polyhedrally representable*, if its epigraph is a polyhedrally representable set:

$$\begin{array}{rcl} \mathrm{Epi}\{f\} & := & \{[x;\tau] : \tau \geq f(x)]\} \\ & = & \{[x;\tau] : \exists w : Px + \tau p + Qw - r \geq 0\} \end{array}$$

♠ *A polyhedrally representable function always is convex* (but not vice versa);
♠ *A function f is polyhedrally representable if and only if its domain is a polyhedral set, and in the domain f is the maximum of finitely many affine functions:*

$$f(x) = \begin{cases} \max\limits_{1 \leq \ell \leq L} [a_\ell^T x + b_\ell], & x \in \mathrm{Dom} f := \{x : Cx \leq d\} \\ +\infty, & \text{otherwise} \end{cases}$$

♠ *A level set $\{x : f(x) \leq a\}$ of a polyhedrally representable function is polyhedral.*

♠ *All basic convexity-preserving operations with functions* (taking finite maxima, linear combinations with nonnegative coefficients, affine substitution of argument) *as applied to polyhedrally representable operands yield polyhedrally representable results.*

• For all basic convexity-preserving operations with functions, *a polyhedral representation of the result is readily given by polyhedral representations of the operands.*

♣ Given a family $\mathcal{K}$ of regular cones closed w.r.t. taking direct products and passing from a cone to its dual, we can define the notion of $\mathcal{K}$-representation of a function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ as $\mathcal{K}$-representation of its epigraph:

$$\begin{array}{rcl} \mathrm{Epi}\{f\} & := & \{[x;\tau] : \tau \geq f(x)]\} \\ & = & \{[x;\tau] : \exists w : Px + \tau p + Qw - r \in \mathbf{K}\} \end{array}$$

with $\mathbf{K} \in \mathcal{K}$. *All what was said on polyhedral representability of functions, except for piecewise linearity, specific for polyhedral representability, holds true for $\mathcal{K}$-representability.*

• For "rich" $\mathcal{K}$, like SDO, the family of $\mathcal{K}$-representable sets/functions is incomparably wider than the family of polyhedral sets/functions.

III. STRUCTURE OF A POLYHEDRAL SET
III.A: Extreme Points

♣ **Extreme points:** A point $v \in X = \{x \in \mathbb{R}^n : Ax \leq b\}$ is called an *extreme point* of X, if v is not a midpoint of a nontrivial segment belonging to X:

$$v \in \mathrm{Ext}(X) \Leftrightarrow v \in X \ \& \ v \pm h \in X \Rightarrow h = 0.$$

♠ **Facts:** Let $X = \{x \in \mathbb{R}^n : a_i^T x \leq b_i, 1 \leq i \leq m\}$ be a nonempty polyhedral set.

- X has extreme points if and only if it does not contain lines;
- The set $\mathrm{Ext}(X)$ of the extreme points of X is finite;
- X is bounded iff $X = \mathrm{Conv}(\mathrm{Ext}(X))$;
- A point $v \in X$ is an extreme point of X if and only if among the inequalities $a_i^T x \leq b_i$ which are *active* at v: $a_i^T v = b_i$, there are $n = \dim x$ inequalities with linearly independent vectors of coefficients:

$$v \in \mathrm{Ext}(X)$$
$$\Leftrightarrow v \in X \ \& \ \mathrm{Rank}\{a_i : a_i^T v = b_i\} = n.$$

♠ The notion of extreme point naturally extends from polyhedral to *closed* convex sets X. It is still true that

- The set $\mathrm{Ext}(X)$ of extreme points of a closed convex set X is nonempty if and only if X is nonempty and does not contain lnes
- When closed convex set X is bounded, one has

$$X = \mathrm{Conv}(\mathrm{Ext}(X))$$

♡ When X is not polyhedral, the set $\mathrm{Ext}(X)$ can be (and typically is) infinite

III.B. Recessive Directions

♣ Let $X = \{x \in \mathbb{R}^n : Ax \leq b\}$ be a nonempty polyhedral set. A vector $d \in \mathbb{R}^n$ is called a *recessive direction* of X, if there exists $\bar{x} \in X$ such that the ray $\mathbb{R}_+ \cdot d := \{\bar{x} + td : t \geq 0\}$ is contained in X.

♠ **Facts:** • The set of all recessive directions of X form a cone, called the *recessive cone* $\mathrm{Rec}(X)$ of X.

• The recessive cone of X is the polyhedral set given by $\mathrm{Rec}(X) = \{d : Ad \leq 0\}$

• Adding to an $x \in X$ a recessive direction, we get a point in X:

$$X + \mathrm{Rec}(X) = X.$$

• The recessive cone of X is trivial (i.e., $\mathrm{Rec}(X) = \{0\}$) if and only if X is bounded.

• X contains a line if and only if $\mathrm{Rec}(X)$ is *not* pointed, that is, $\mathrm{Rec}(X)$ contains a line, or, equivalently, there exists $d \neq 0$ such that $\pm d \in \mathrm{Rec}(X)$, or, equivalently, the null space of A is nontrivial: $\mathrm{Ker}A = \{d : Ad = 0\} \neq \{0\}$.

• A line directed by a vector d belongs to X if and only if it crosses X and $d \in \mathrm{Ker}A = \mathrm{Rec}(X) \cap [-\mathrm{Rec}(X)]$.

• One has

$$X = X + \mathrm{Ker}A.$$

• X can be represented as

$$X = \widehat{X} + \mathrm{Ker}A$$

where $\widehat{X}$ is a nonempty polyhedral set which does *not* contain lines. One can take

$$\widehat{X} = X \cap [\mathrm{Ker}A]^{\perp}.$$

♠ The notion of recessive direction naturally extends from polyhedral to nonempty *closed* convex sets X. It is still true that

• $\mathrm{Rec}(X)$ is a closed convex cone (not necessarily polyhedral)

• One has

$$X + \mathrm{Rec}(X) = X$$

• $\mathrm{Rec}(X)$ is trivial if and only if X is bounded

III.C. Extreme Rays of a Cone

♣ Let K be a polyhedral cone, that is,

$$\begin{array}{rcl} K & = & \{x \in \mathbb{R}^n : Ax \leq 0\} \\ & = & \{x \in \mathbb{R}^n : a_i^T x \leq 0,\ 1 \leq i \leq m\} \end{array}$$

♠ A direction $d \in K$ is called an *extreme direction of* K, if $d \neq 0$ and in any representation $d = d_1 + d_2$ *with* $d_1, d_2 \in K$ both d_1 and d_2 are nonnegative multiples of d.

• Given $d \in K$, the ray $R = \mathbb{R}_+ \cdot d \subset K$ is called the *ray generated by* d, and d is called a *generator* of the ray. When $d \neq 0$, all generators of the ray $R = \mathbb{R}_+ \cdot d$ are positive multiples of d, and vice versa – every positive multiple of d is a generator of R.

• Rays generated by extreme directions of K are called *extreme rays of* K.

♠ **Facts:**

• *A direction* $d \neq 0$ *is an extreme direction of a pointed polyhedral cone* $K = \{x \in \mathbb{R}^n : a_i^T x \leq 0, 1 \leq i \leq m\}$ *if and only if* $d \in K$ *and among the inequalities* $a_i^T x \leq 0$ *defining* K *there are* $n-1$ *which are active at* d $a_i^T d = 0$ *and have linearly independent vectors of coefficients:*

$$\begin{array}{c} d \text{ is an extreme direction of } K \\ \Leftrightarrow d \in K \backslash \{0\} \ \& \ \text{Rank}\,\{a_i : a_i^T d = 0\} = n - 1 \end{array}$$

• K has extreme rays if and only if K is nontrivial ($K \neq \{0\}$) and pointed ($K \cap [-K] = \{0\}$), and in this case

- the number of extreme rays of K is finite, and
- if $r_1, ..., r_M$ are generators of extreme rays of K, then

$$K = \text{Cone}\,(\{r_1, ..., r_M\}).$$

♠ The notion of extreme ray extends naturally from polyhedral cones to *closed* convex cones K. It still is true that

• K possesses extreme rays if and only if K is nontrivial and pointed, and in this case K is the conic hull of the set of its extreme rays.

♡ What is lost in the general case, is finiteness of the set of extreme rays.

III.D. Structure of a Polyhedral Set

♣ **Theorem:** *Every nonempty polyhedral set X can be represented in the form*

$$X = \mathrm{Conv}(\{v_1, ..., v_N\}) + \mathrm{Cone}\,(\{r_1, ..., r_M\}) \qquad (!)$$

where $\{v_1, ..., v_N\}$ is a nonempty finite set, and $\{r_1, ..., r_M\}$ is a finite (possibly empty) set. Vice versa, every set of the form (!) is a nonempty polyhedral set. In addition,

♠ *In a representation (!), one always has*

$$\mathrm{Cone}\,(\{r_1, ..., r_M\}) = \mathrm{Rec}(X)$$

♠ *Let X do not contain lines. Then one can take in (!) $\{v_1, ..., v_N\} = \mathrm{Ext}(X)$ and to choose, as $r_1, ..., r_M$, the generators of the extreme rays of $\mathrm{Rec}(X)$. The resulting representation is "minimal": for every representation of X in the form of (!), it holds $\mathrm{Ext}(X) \subset \{v_1, ..., v_N\}$ and every extreme ray of $\mathrm{Rec}(X)$, is any, has a generator from the set $\{r_1, ..., r_M\}$.*

IV. FUNDAMENTAL PROPERTIES OF LO PROGRAMS

♣ Consider a *feasible* LO program

$$\mathrm{Opt} = \max_x \left\{ c^T x : Ax \leq b \right\} \qquad (P)$$

♠ **Facts:**

- (P) is solvable (i.e., admits an optimal solution) *if and only if* (P) *is bounded* (i.e., the objective is bounded from above on the feasible set)
- (P) *is bounded if and only if* $c^T d \leq 0$ *for every* $d \in \mathrm{Rec}(X)$
- Let the feasible set of (P) do not contain lines. Then (P) *is bounded if and only if* $c^T d \leq 0$ *for generator of every extreme ray of* $\mathrm{Rec}(X)$*, and in this case there exists an optimal solution which is an extreme point of the feasible set.*

V. GTA AND DUALITY

♣ **GTA:** *Consider a finite system (S) of nonstrict and strict linear inequalities and linear equations in variables $x \in \mathbb{R}^n$:*

$$\begin{array}{c} a_i^T x \ \Omega_i \ b_i,\ 1 \le i \le m \\ \Omega_i \in \{" \le ", " < ", " \ge ", " > ", " = "\} \end{array} \qquad (S)$$

(S) has no solutions if and only if a legitimate weighted sum of the inequalities of the system is a contradictory inequality, that is

One can assign the constraints of the system with weights λ_i so that

- *the weights of the "$<$" and "$\le$" inequalities are nonnegative, while the weights of the "$>$" and "$\ge$" inequalities are nonpositive,*
- *$\sum_{i=1}^m \lambda_i a_i = 0$, and*
- *either $\sum_{i=1}^m \lambda_i b_i < 0$,*
  or $\sum_{i=1}^m \lambda_i b_i = 0$ and at least one of the strict inequalities gets a nonzero weight.

♠ **Particular case: Homogeneous Farkas Lemma.** *A homogeneous linear inequality* $a^T x \leq 0$ *is a consequence of a system of homogeneous linear inequalities* $a_i^T x \leq 0$, $1 \leq i \leq m$, *if and only if the inequality is a combination, with nonnegative weights, of the inequalities from the system, or, which is the same, if and only if* $a \in \mathrm{Cone}\,(\{a_1, ..., a_m\})$.

♠ **Particular case: Inhomogeneous Farkas Lemma.** *A linear inequality* $a^T x \leq b$ *is a consequence of a solvable system of inequalities* $a_i^T x \leq b_i$, $1 \leq i \leq m$, *if and only if the inequality is a combination, with nonnegative weights, of the inequalities from the system and the trivial identically true inequality* $0^T x \leq 1$, *or, which is the same, if and only if there exist nonnegative* λ_i, $1 \leq i \leq m$, *such that* $\sum_{i=1}^m \lambda_i a_i = a$ *and* $\sum_{i=1}^m \lambda_i b_i \leq b$.

LO DUALITY

♣ *Given an LO program in the form*

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \left\{ \begin{array}{l} Px - p \geq 0 \\ Rx = r \end{array} \right. \right\} \qquad (P)$$

we associate with it the dual problem

$$\mathrm{Opt}(D) = \max_{\lambda,\mu} \left\{ p^T\lambda + r^T\mu : \left\{ \begin{array}{l} \lambda \geq 0 \\ P^T\lambda + R^T\mu = c \end{array} \right. \right\}. \qquad (D)$$

LO Duality Theorem:

(i) [symmetry] *Duality is symmetric: the problem dual to (D) is (equivalent to) the primal problem (P)*

(ii) [weak duality] $\mathrm{Opt}(D) \leq \mathrm{Opt}(P)$

(iii) [strong duality] *The following 3 properties are equivalent to each other:*

- *one of the problems (P), (D) is feasible and bounded*
- *both (P) and (D) are solvable*
- *both (P) and (D) are feasible*

Whenever one of (and then – all) these properties takes place, we have

$$\mathrm{Opt}(D) = \mathrm{Opt}(P).$$

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \left\{ \begin{array}{l} Px - p \geq 0 \\ Rx = r \end{array} \right\} \right. \qquad (P)$$

$$\mathrm{Opt}(D) = \max_{\lambda,\mu} \left\{ p^T\lambda + r^T\mu : \left\{ \begin{array}{l} \lambda \geq 0 \\ P^T\lambda + R^T\mu = c \end{array} \right\} \right. . \qquad (D)$$

♠ **LO Optimality Conditions:** *Let*

$$x, (\lambda, \mu)$$

be feasible solutions to the respective problems (P), (D). *Then* $x, (\lambda, \mu)$ *are optimal solutions to the respective problems*

- *if and only if the associated duality gap is zero:*

$$\begin{array}{rcl} \mathrm{DualityGap}(x, (\lambda, \mu)) & := & c^T x - [p^T\lambda + r^T\mu] \\ & = & 0 \end{array}$$

- *if and only if the complementary slackness condition holds:*

$$\lambda^T [Px - p] = 0,$$

or, equivalently, nonzero Lagrange multipliers λ_i *are associated only with the primal constraints which are active at* x*:*

$$\forall i : \lambda_i [Px - p]_i = 0$$

♣ As a corollary of Duality Theorem, *the optimal value in a feasible maximization LO program is a polyhedral function of the objective admitting an explicit polyhedral representation:*

$$\mathrm{Opt}(x) := \max_u \left\{x^T u : Pu - p \geq 0, Ru = r\right\} \leq \tau$$
$$\Updownarrow$$
$$\exists \lambda, \mu : \lambda \geq 0, -P^T\lambda + R^T\mu = x, -p^T\lambda + r^T\mu \leq \tau$$

♠ As a result, *the solution set of a semi-infinite scalar linear inequality*

$$a^T x \leq b \; \forall (a, b) \in \mathcal{U}$$

with polyhedral *uncertainty set* $\mathcal{U}$ is polyhedrally representable, with polyhedral representation readily given by the one of $\mathcal{U}$

$\Rightarrow$ *The robust counterpart*

$$\min\{c^T x : a_i^T x \leq b_i \; \forall (a_i, b_i) \in \mathcal{U}_i, \; 1 \leq i \leq m\}$$

with polyhedral uncertainty sets $\mathcal{U}_i$ *is equivalent to an LO problem readily given by polyhedral representations of the uncertainty sets* $\mathcal{U}_1, ..., \mathcal{U}_m$.

CONIC DUALITY

♣ Given conic program in the form

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \left\{ \begin{array}{l} Px - p \geq 0 \\ Qx - q \in \mathbf{K} \\ Rx = r \end{array} \right\} \right. \quad (P)$$

we associate with it the *dual problem*

$$\mathrm{Opt}(D) = \max_{\lambda,\omega,\mu} \left\{ p^T\lambda + q^T\omega + r^T\mu : \left\{ \begin{array}{l} \lambda \geq 0 \\ \omega \in \mathbf{K}_* \\ P^T\lambda + Q^T\omega + R^T\mu = c \end{array} \right\} \right. . \quad (D)$$

Conic Duality Theorem:

(i) [symmetry] *Duality is symmetric: the problem dual to* (D) *is (equivalent to) the primal problem* (P)

(ii) [weak duality] $\mathrm{Opt}(D) \leq \mathrm{Opt}(P)$

(iii) [strong duality] *Let one of the problems* (P), (D) *be essentially strictly feasible*, meaning that it has a feasible solution where the nonpolyhedral ("brown") conic constraint is satisfied strictly: ... $\in$ int .... *Then the other problem is solvable, and* $\mathrm{Opt}(P) = \mathrm{Opt}(D)$. *In particular, if both problems are essentially strictly feasible, both are solvable with equal optimal values.*

$$\mathrm{Opt}(P) = \min_x \left\{ c^T x : \left\{ \begin{array}{l} Px - p \geq 0 \\ Qx - q \in \mathbf{K} \\ Rx = r \end{array} \right\} \right. \quad (P)$$

$$\mathrm{Opt}(D) = \max_{\lambda,\omega,\mu} \left\{ p^T\lambda + q^T\omega + r^T\mu : \left\{ \begin{array}{l} \lambda \geq 0 \\ \omega \in \mathbf{K}_* \\ P^T\lambda + Q^T\omega + R^T\mu = c \end{array} \right\}. \right. \quad (D)$$

♠ **Conic Programming Optimality Conditions:** *Let*

$$x, (\lambda, \omega, \mu)$$

be feasible solutions to the respective problems (P), (D), *and let one of the problems be essentially strictly feasible. Then* $x, (\lambda, \omega, \mu)$ *are optimal solutions to the respective problems*

- *if and only if the associated duality gap is zero:*

$$\begin{array}{rcl} \mathrm{DualityGap}(x, (\lambda, \omega, \mu)) & := & c^T x - [p^T\lambda + q^T\omega + r^T\mu] \\ & = & 0 \end{array}$$

- *if and only if the complementary slackness condition holds:*

$$\underbrace{\lambda^T[Px - p] = 0}_{\Leftrightarrow \lambda_i [Px-p]_i = 0 \; \forall i} \; \& \; \omega^T[Qx - q] = 0.$$

♣ *Conic Duality Theorem underlies several important "nonlinear versions" of the basic polyhedral results, like*

"Conic" Farkas Lemma: *A scalar linear inequality* $a^T x \geq b$ *is a consequence of a strictly feasible system*

$$A_i x - b_i \in \mathbf{K}_i,\ i = 1, ..., m, \qquad (S)$$

of conic inequalities if and only if it can be obtained by taking "legitimate weighted sum" of inequalities from the system and the trivial identically true inequality $0 \leq 1$*:*

Scalar linear inequality $a^T x \geq b$ *is a consequence of strictly feasible system* (S) *iff*

$$\exists \lambda_i \in \mathbf{K}_i^* : \ \sum_i A_i^T \lambda_i = a \ \& \ \sum_i \lambda_i^T b_i \geq b$$

GEOMETRY OF PRIMAL-DUAL PAIR OF CONIC PROBLEMS

♣ **Geometrically**, *a primal-dual pair of conic problems is specified by*

- *a regular cone* $\mathbf{K}$ *in some* $E = \mathbb{R}^N$ *and its dual cone* $\mathbf{K}_*$
- *a pair of linear subspaces* $\mathcal{L}_P$, $\mathcal{L}_D$ *in* E *which are orthogonal complements to each other*
- *a pair of shift vectors* $b \in E, c \in E$

and requires to find in the primal feasible set

$$[\mathcal{L}_P - b] \bigcap \mathbf{K}$$

and the dual feasible set

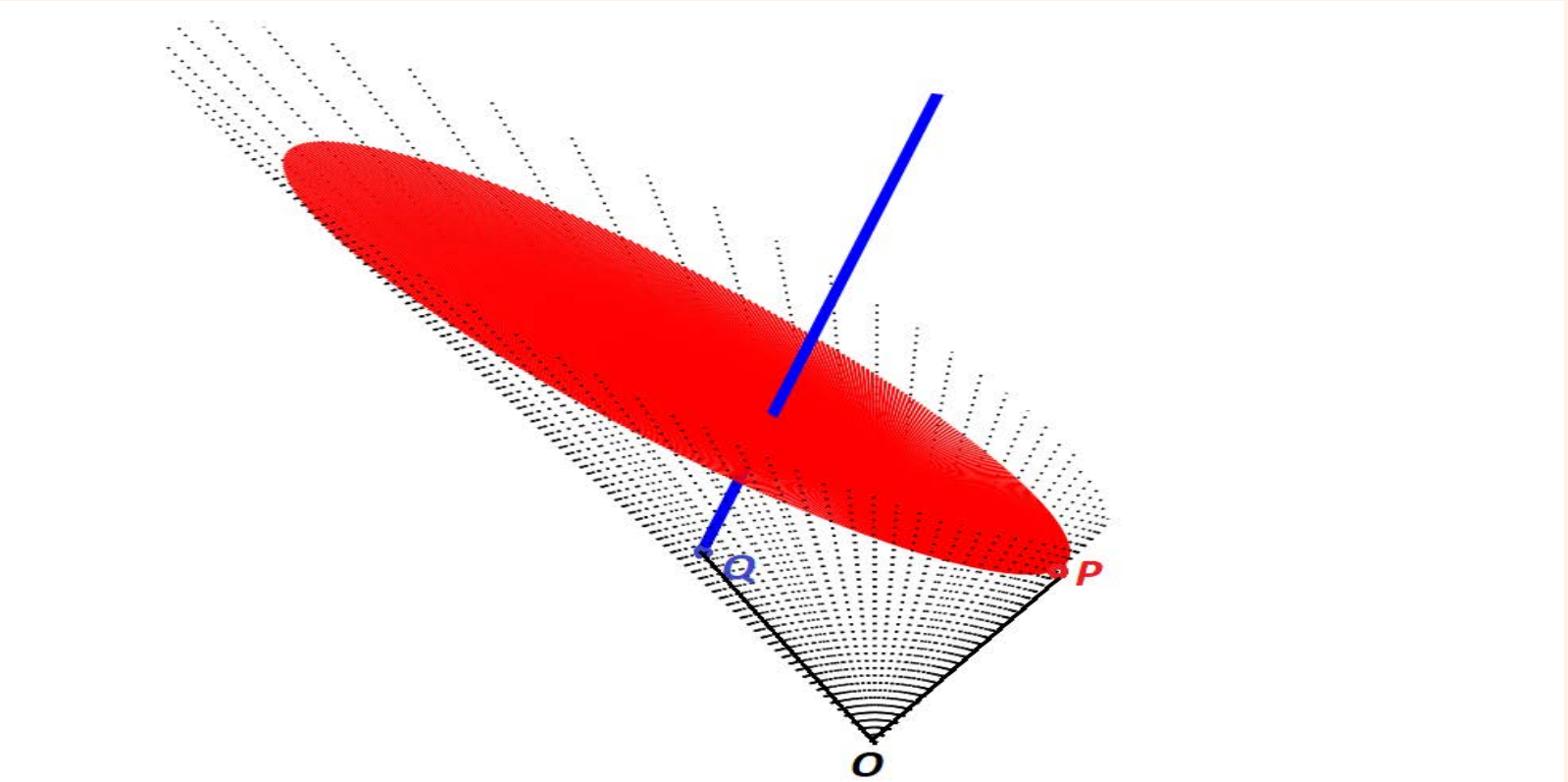
$$[\mathcal{L}_D + c] \bigcap \mathbf{K}_*$$

a pair of orthogonal to each other vectors x_*, λ_**; whenever this is possible,* x_*, λ_* *are optimal solutions to the problems*

$$\begin{array}{rcll} \mathrm{Opt}(\mathcal{P}) & = & \min_x \left\{c^T x : x \in [\mathcal{L}_P - b] \bigcap \mathbf{K}\right\} & (\mathcal{P}) \\ \mathrm{Opt}(\mathcal{D}) & = & \max_\lambda \left\{b^T \lambda : \lambda \in [\mathcal{L}_D + c] \bigcap \mathbf{K}_*\right\} & (\mathcal{D}) \end{array}$$

and $\mathrm{Opt}(\mathcal{P}) - \mathrm{Opt}(\mathcal{D}) + b^T c = 0$ *(strong duality). Besides this,*

- *one always has* $\mathrm{Opt}(\mathcal{P}) - \mathrm{Opt}(\mathcal{D}) + b^T c \geq 0$ *(weak duality);*
- *under strong duality,* $[c^T x - \mathrm{Opt}(P)] + [\mathrm{Opt}(D) - b^T \lambda] = \lambda^T x$ *for all primal-dual feasible* (x, λ)*,*
- *under strong duality, a primal-dual feasible pair* (x, λ) *is composed of optimal solutions iff the solutions in the pair are orthogonal;*
- *if* (P), (D) *are essentially strictly feasible, both problems are solvable and strong duality takes place. When* $\mathbf{K} = \mathbb{R}^N_+$*, the conclusion remains true when* (P), (D) *are just feasible.*



Primal-dual pair of conic problems on 3D Lorentz cone (self-dual)
Red: feasible set of $(\mathcal{P})$ *Blue: feasible set of* $(\mathcal{D})$
P *– optimal solution to* $(\mathcal{P})$; Q *– optimal solution to* $(\mathcal{D})$.
Pay attention to orthogonality of $\overrightarrow{OP}$ *to* $\overrightarrow{OQ}$

Tractability of *LO, CQO, SDO*

♠ *Convex Programming in general, and the generic conic problems LO, CQO, SDO in particular, form a "solvable case" in Optimization: in theory, and to some extent also in practice, globally optimal solutions to these problems can be approximated within whatever high accuracy in reasonable, polynomial in the sizes of the instances and in a desired number of accuracy digits, time.*

How It Works

♠ **Computational mini-study, LO:**
A. Dense unstructured LOs – ℓ_1 minimization problems $\min_x \{\|x\|_1 : \|Ax - b\|_\infty \leq \delta\}$ with randomly generated fully dense $m \times n$ matrices A

$m \times n$	Method	Iterations	CPU, sec
100×500	PSM	2006	0.2
	DSM	820	0.1
	IPM	15	0.1
500×2500	PSM	24512	52.0
	DSM	5094	12.2
	IPM	17	3.2
1000×5000	PSM	76926	564.4
	DSM	11018	102.5
	IPM	15	13.9

B. Sparse well structured LOs – Maximum Flow on randomly generated m-node networks with n arcs

$m \times n$	Method	Iterations	CPU, sec
1000×5000	PSM	38	0.0
	DSM	28	0.0
	IPM	4	0.1
$5000 \times 25,000$	PSM	8	0.1
	DSM	16	0.3
	IPM	4	2.1
$10,000 \times 50,000$	PSM	64	0.2
	DSM	38	1.3
	IPM	4	12.4
$100,000 \times 500,000$	PSM	166	2.7
	DSM	200	205.2
	IPM	4	11325.5

♠ **Computational mini-study: SDO:** Computing *Lovasz ϑ-function* of a graph.

♡ For an undirected m-node graph G, its ϑ-function $\theta(G)$ is defined as follows:

• one associates with G variable $m \times m$ symmetric matrix X where:

— diagonal entries X_{ii} and entries X_{ij} with distinct i, j *not* linked by arc in G are set to 1

— entries $X_{ij} = X_{ji}$ with distinct i, j linked by arc in G are arbitrary

• Just defined matrices form affine plane $\mathcal{X}_G$ in the space $\mathbf{S}^m$ of $m \times m$ symmetric matrices. By definition,

$$\vartheta(G) = \min_{\lambda, X} \{\lambda : \lambda I_m - X \succeq 0, X \in \mathcal{X}_G\}$$

♡ $\vartheta(G)$ is efficiently computable upper bound on the difficult to compute combinatorial quantity – the *stability number* of G, that is, the maximum cardinality of independent subsets (i.e., with no two nodes linked by an arc) of the nodal set of G.

$m \times n$	Iterations	CPU, sec
64×512	11	1.3
128×1024	11	1.6
256×2048	11	4.7
512×4096	12	29.8
1024×8192	12	178.7
2048×16384	12	1284.3