Vietnam National University - Ho Chi Minh

Optimization, Machine Learning and Kernel Methods.

Optimization I

Marco Cuturi - Princeton University

Outline of this module

- Short historical introduction to **mathematical programming**
- Start with **linear programming**.
  - introduce convexity,
  - an important algorithm: simplex
- Follow with **convex programming**
  - study convex programs,
  - define **duality**,
  - study algorithms.

Mathematical Programming

- The term *programming* in *mathematical programming* is actually **not** related to computer programs.
- Dantzig explains the jargon in 2002 (document available on BB)
  - `The military refer to their various plans or proposed schedules of training, logistical supply and deployment of combat units as a program. When I first analyzed the Air Force planning problem and saw that it could be formulated as a system of linear inequalities, I called my paper Programming in a Linear Structure. Note that the term program was used for linear programs long before it was used as the set of instructions used by a computer. In the early days, these instructions were called codes.`

Mathematical Programming

- In the summer of 1948, Koopmans and I visited the Rand Corporation. One day we took a stroll along the Santa Monica beach. Koopmans said: Why not shorten Programming in a Linear Structure to Linear Programming? I replied: Thats it! From now on that will be its name. Later that day I gave a talk at Rand, entitled Linear Programming; years later Tucker shortened it to Linear Program.

- The term Mathematical Programming is due to Robert Dorfman of Harvard, who felt as early as 1949 that the term Linear Programming was too restrictive.

Mathematical Programming

- Today mathematical programming is synonymous with **optimization**. A relatively **new discipline** and one that has had significant impact.
  - `What seems to characterize the pre-1947 era was lack of any interest in trying to optimize. T. Motzkin in his scholarly thesis written in 1936 cites only 42 papers on linear inequality systems, none of which mentioned an objective function.`

Origins & Success

- **Monge**'s 1781 memoir is the earliest known anticipation of Linear Programming type of problems, in particular of the transportation problem (moving piles of dirt into holes).
- In the early 40's significant work can also be attributed to **Kantorovich** in the USSR (Nobel 75) on transport planning as well. More dramatic application: *road of life* from & to Leningrad during WW2.

- **Dantzig** proposed a general method to solve LP's in 1947, the **simplex method**, which ranks among `the top 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century` according to the journal *Computing in Science & Engineering*.
- Other laureates: metropolis, FFT, quicksort, Krylov subspaces, QR decomposition *etc.*

Mathematical Programming

- A general formulation for a mathematical programming problem is that of defining the unknown variables $x_1, x_2, \cdots, x_n \in \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n$ such that

$$\begin{array}{ll} \text{minimize (or mazimize)} & \boldsymbol{f(x_1, x_2, \cdots, x_n)}, \\ \text{subject to} & \boldsymbol{g_i(x_1, x_2, \cdots, x_n)} \left\{ \begin{array}{c} <,> \\ = \\ \leq,\geq \end{array} \right\} \boldsymbol{b_i}, i = 1, 2, \cdots, m; \end{array}$$

  where the b_i's are real constants and the functions $\boldsymbol{f}$ (the objective) and $\boldsymbol{g_1, g_2, \cdots, g_m}$ (the constraints) are real-valued functions of $\mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n$.

- the sets $\mathcal{X}_i$ need not be the same, as $\mathcal{X}_i$ might be
  - **R** scalar numbers,
  - **Z** integers,
  - $\mathbf{S}_n^+$ positive definite matrices,
  - strings of letters,
  - *etc.*

- When the $\mathcal{X}_i$ are different, the adjective *mixed* usually comes in.

Linear Programs in $\mathbf{R}^n$

- the general form of linear programs in $\mathbf{R}^n$:

$$
\begin{array}{ll}
\text{max } \textit{or} \text{ min} & z = \boldsymbol{c_1} x_1 + \boldsymbol{c_2} x_2 + \cdots + \boldsymbol{c_n} x_n, \\
\text{subject to} & \left\{ \begin{array}{ccccccccc}
\boldsymbol{a_{11}} x_1 & + & \boldsymbol{a_{12}} x_2 & + & \cdots & + & \boldsymbol{a_{1n}} x_n & \left\{ \begin{array}{c} <,> \\ = \\ \leq, \geq \end{array} \right\} & \boldsymbol{b_1}, \\
\vdots & & & & & & & & \vdots \\
\boldsymbol{a_{m1}} x_1 & + & \boldsymbol{a_{m2}} x_2 & + & \cdots & + & \boldsymbol{a_{mn}} x_n & \left\{ \begin{array}{c} <,> \\ = \\ \leq, \geq \end{array} \right\} & \boldsymbol{b_m},
\end{array} \right. \\
\text{where} & x_1 \geq \mathbf{0},\ x_2 \geq \mathbf{0},\ \cdots,\ x_n \geq \mathbf{0}.
\end{array}
$$

- linear objective, linear constraints... simple.
- yet powerful for many problems and one of the first classes of mathematical programs that was solved.

Linear Programs, landmarks in history

- First solution by Dantzig in the late 40's, the simplex.
- At the time, programs were solved by hand, the algorithm reflects this.
- In 1972, Klee and Minty show that the simplex has an exponential worst case complexity.
- Low complexity of linear programming proved (in theory) by Nemirovski, Yudin and Khachiyan in the USSR in 1976.
- First efficient algorithm with provably low complexity discovered by Karmarkar at Bell Labs in 1984.

Mathematical Programming Subfields

- **convex** programming: f is a **convex** function and the constraints g_i, if any, form a **convex** set.
  - **Linear programming**.
  - Second order cone programming (SOCP).
  - Semidefinite Programming, that is linear programs in $\mathbf{S}_n^+$.
  - Conic programming, with more general cones.
- Quadratic programming (QP), with quadratic objectives and linear constraints,
- Nonlinear programming,
- Stochastic programming,
- Combinatorial programming: discrete set of feasible solutions. **integer programming**, that is LP's with integer variables, is a subfield.

Some examples

The Diet Problem

- Most introductions to LP start with the diet problem.
- The reason: historically, one of the first large scale LP's that was computed. More on this later.
- You're a (bad) cook obsessed with numbers trying to come up with a new **cheap** dish that **meets nutrition standards**.
- You summarize your problem in the following way:

Ingredient	Carrot	Cabbage	Cucumber	Required per dish
Vitamin A [mg/kg]	**35**	**0.5**	**0.5**	**0.5mg**
Vitamin C [mg/kg]	**60**	**300**	**10**	**15mg**
Dietary Fiber [g/kg]	**30**	**20**	**10**	**4g**
Price [$/kg]	**0.75**	**0.5**	**0.15**	-

The Diet Problem

- Let x_1, x_2 and x_3 be the amount in kilos of carrot, cabbage and cucumber in the new dish.
- Mathematically,

$$\begin{array}{lrl} \text{minimize} & \mathbf{0.75x_1 + 0.5x_2 + 0.15x_3}, & \textbf{cheap,} \\ \text{subject to} & \mathbf{35x_1 + 0.5x_2 0.5x_3 \geq 0.5}, & \textbf{\textit{nutritious,}} \\ & \mathbf{60x_1 + 300x_2 + 10x_3 \geq 15}, & \\ & \mathbf{30x_1 + 20x_2 + 10x_3 \geq 4}, & \\ & \mathbf{x_1, x_2, x_3 \geq 0}. & \textbf{reality.} \end{array}$$

- The program can be solved by standard methods. The optimal solution yields a price of 0.07$ pre dish, with 9.5g of carrot, 38g of cabbage and 290g of cucumber...

The Diet Problem

- The first large scale experiment for the simplex algorithm: 77 variables (ingredients) and 9 constraints (health guidelines)
- The solution, computed by hand-operated desk calculators took 120 man-days.
- The first recommendation was to drink several *liters* of vinegar every day.
- When vinegar was removed, Dantzig obtained *200 bouillon cubes* as the basis of the diet.
- This illustrates that a clever and careful mathematical **modeling** is always important before **solving** anything.

Flow of packets in Networks

We follow with an example in networks:

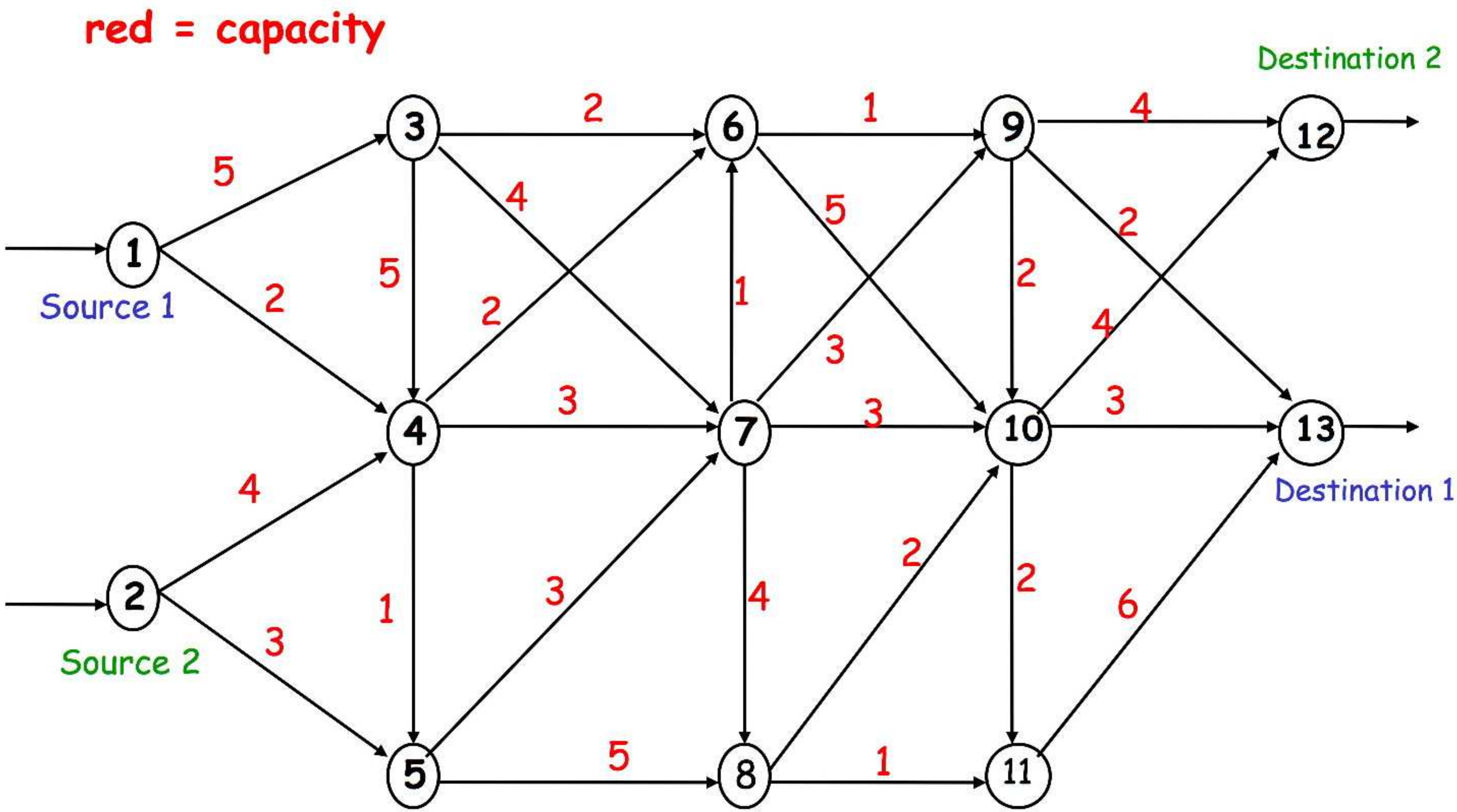
- We use the internet here, but this could be any network (factory floor, transportation, etc).
- Transport data packets from a source to a destination.
- For simplicity: two sources, two destinations.
- Each link in the network has a fixed capacity (bandwidth), shared by all the packets in the network.

Networks: Routing

- When a link is saturated (congestion), packets are simply dropped.
- Packets are dropped at random from those coming through the link.
- Objective: choose a routing algorithm to maximize the **total bandwidth** of the network.

This randomization is not a simplification. TCP/IP, the protocol behind the internet, works according to similar principles.. . .

Networks: Routing



Networks: Routing

A model for the network routing problem: let $N = \{1, 2, \ldots, 13\}$ be the set of network nodes and $L = \{(1,3), \ldots, (11,13)\}$ the set of links.

Variables:

- x_{ij} the flow of packets with origin 1 and destination 1, going through the link between nodes i and j.
- y_{ij} the flow of packets with origin 2 and destination 2, going through the link between nodes i and j.

Parameters:

- u_{ij} the maximum capacity of the link between nodes i and j.

Networks: Routing

In EXCEL. . .

Routing problem: Modeling

Write this as an optimization problem.

Consistency constraints:

- Flow coming out of a node must be less than incoming flow:

$$\sum_{j:\ (i,j)\in L} x_{ij} \leq \sum_{j:\ (j,i)\in L} x_{ij}, \quad \text{for all nodes } i$$

  and

$$\sum_{j:\ (i,j)\in L} y_{ij} \leq \sum_{j:\ (j,i)\in L} y_{ij}, \quad \text{for all nodes } i$$

- Flow has to be positive:

$$x_{ij}, y_{ij} \geq 0, \quad \text{for all } (i,j) \in L$$

Routing problem: Modeling

Capacity constraints:

- Total flow through a link must be less than capacity:

$$x_{ij} + y_{ij} \leq u_{ij}, \quad \text{for all } (i,j) \in L$$

- No packets originate from wrong source:

$$x_{2,4},\ x_{2,5},\ y_{1,3},\ y_{1,4} = 0$$

Objective:

- Maximize total throughput at destinations:

$$\text{maximize } x_{9,13} + x_{10,13} + x_{11,13} + y_{9,12} + y_{10,12}$$

Routing problem: Modelling

The final program is written:

$$
\begin{aligned}
\text{maximize} \quad & x_{9,13} + x_{10,13} + x_{11,13} + y_{9,12} + y_{10,12} \\
\text{subject to} \quad & \sum_{j:\ (i,j)\in L} x_{ij} \leq \sum_{j:\ (j,i)\in L} x_{ij} \\
& \sum_{j:\ (i,j)\in L} y_{ij} \leq \sum_{j:\ (j,i)\in L} y_{ij} \\
& x_{ij} + y_{ij} \leq u_{ij} \\
& x_{2,4}, x_{2,5}, y_{1,3}, y_{1,4} = 0 \\
& x_{ij}, y_{ij} \geq 0, \quad \text{for all } (i,j) \in L
\end{aligned}
$$

Constraints and objective are linear: this is a **linear program**.

Routing problem: Solving

- In this case, the model was written entirely in EXCEL
- EXCEL has a rudimentary linear programming solver (which does not work very well for macs unfortunately)
- This is how the optimal solution was found here. In general, specialized solvers are used (more later).

- Original solution, : network capacity of 3.7
- Optimal capacity: **14 !!**

Typology of Linear Programs

Remember...

- the general form of linear programs:

$$
\begin{array}{ll}
\text{max } \textit{or} \text{ min} & z = \boldsymbol{c_1} x_1 + \boldsymbol{c_2} x_2 + \cdots + \boldsymbol{c_n} x_n, \\
\text{subject to} & \left\{ \begin{array}{ccccccccc}
\boldsymbol{a_{11}} x_1 & + & \boldsymbol{a_{12}} x_2 & + & \cdots & + & \boldsymbol{a_{1n}} x_n & \left\{ \begin{array}{c} <,> \\ = \\ \leq,\geq \end{array} \right\} & \boldsymbol{b_1}, \\
\vdots & & & & & & & & \vdots \\
\boldsymbol{a_{m1}} x_1 & + & \boldsymbol{a_{m2}} x_2 & + & \cdots & + & \boldsymbol{a_{mn}} x_n & \left\{ \begin{array}{c} <,> \\ = \\ \leq,\geq \end{array} \right\} & \boldsymbol{b_m}, \\
\end{array} \right. \\
\text{where} & x_1, x_2, \cdots, x_n \geq \mathbf{0}.
\end{array}
$$

- This form is however too vague to be easily usable.
- First step: get rid of the strict inequalities: do not bring much and would only add numerical noise.
- Second step: use matrix and vectorial notations to alleviate.

Notations

Unless explicitly stated otherwise,

- A, B etc... are matrices whose size is clear from context.
- $\mathbf{x}, \mathbf{b}, \mathbf{a}$ are vectors. $\mathbf{a}_1, \mathbf{a}_k$ are members of a vector family.
- $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ with vector coordinates x_i in **R**.
- $\mathbf{x} \geq 0$ is meant coordinate-wise, that is $x_i \geq 0$ for $1 \geq i \leq n$
- $\mathbf{x} \neq \mathbf{0}$ means that $\mathbf{x}$ is not the zero vector, i.e. there exists at least one index i such that $x_i \neq 0$.
- $\mathbf{x}^T$ is the transpose $\begin{bmatrix} x_1, \cdots, x_n \end{bmatrix}$ of $\mathbf{x}$.

Linear Program

Common representation for all these programs?

- Would help in developing both theory & algorithms.
- Also helps when developing software, solvers, etc

The answer is yes. . .

- There are 2: **standard form** and **canonical form**

Terminology

- A linear program in **canonical** form is the program

$$\begin{array}{ll} \text{max or min} & \mathbf{c}^{\boldsymbol{T}}\mathbf{x} \\ \text{subject to} & \boldsymbol{A}\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}. \end{array}$$

$\mathbf{b} \geq 0 \Rightarrow$ **feasible canonical form** (just a convention)

- A linear program in **standard** form is the program

$$\text{max or min} \quad \mathbf{c}^{\boldsymbol{T}}\mathbf{x} \tag{1}$$

$$\text{subject to} \quad \boldsymbol{A}\mathbf{x} = \mathbf{b}, \tag{2}$$

$$\mathbf{x} \geq \mathbf{0}. \tag{3}$$

Linear Programs: a look at the canonical form

Canonical form linear program

- **Maximize** the objective
- Only **inequality** constraints
- All variables should be **positive**

Example:

$$
\begin{array}{lrcrcrcr}
\text{maximize} & 5x_1 & + & 4x_2 & + & 3x_3 & & \\
\text{subject to} & 2x_1 & + & 3x_2 & + & x_3 & \leq & 5 \\
 & 4x_1 & + & x_2 & + & 2x_3 & \leq & 11 \\
 & 3x_1 & + & 4x_2 & + & 2x_3 & \leq & 8 \\
 & & & & x_1, & x_2, x_3 & \geq & 0.
\end{array}
$$

Linear Programs: canonical form

Although more intuitive than the standard form, the canonical is not the most useful,

- We will formulate the simplex method on problems with **equality constraints**, that is **standard forms**.
- Solvers do not all agree on this input format. MATLAB for example uses:

$$
\begin{array}{ll}
\text{minimize} & \sum_i c_i x_i \\
\text{subject to} & \sum_{j=1}^n A_{ij} x_j \leq b_i, \quad i = 1, \ldots, m_1 \\
& \sum_{j=1}^n B_{ij} x_j = d_i, \quad i = 1, \ldots, m_2 \\
& l_i \leq x_i \leq u_i, \quad i = 1, \ldots, n
\end{array}
$$

- Ultimately: this is a **non-issue**, we can easily switch from one form to the other. . .

Linear Programs: standard & canonical form

equalities $\Rightarrow$ inequalities

- What if the original problem has equality constraints?
- Replace equality constraints by two inequality constraints.
- The inequality

$$2x_1 + 3x_2 + x_3 = 5,$$

  is equivalent to

$$2x_1 + 3x_2 + x_3 \leq 5 \quad \text{and} \quad 2x_1 + 3x_2 + x_3 \geq 5$$

- The new problem is **equivalent** to the previous one. . .

Linear Programs: standard & canonical form

inequalities $\Rightarrow$ equalities

- The opposite direction works too. . .
- Turn inequality constraints into equality constraints by **adding variables**.
- The inequality

$$2x_1 + 3x_2 + x_3 \leq 5,$$

  is equivalent to

$$2x_1 + 3x_2 + x_3 + w_1 = 5 \quad \text{and} \quad w_1 \geq 0,$$

- The new variable is called a **slack** variable (one for each inequality in the program). . .
- The new problem is **equivalent** to the previous one. . .

Linear Programs: standard & canonical form

free variable $\Rightarrow$ positive variables

- What about free variables?
- A free variable is simply the difference of its positive and negative parts. Again the solution is again **adding variables**.
- If the variable y is free, we can write it

$$y_1 = y_2 - y_3 \quad \text{and} \quad y_2, y_3 \geq 0,$$

- We add two positive variables for each free variable in the program.
- Again, the new problem is **equivalent** to the previous one.

Linear Programs: standard & canonical form

minimizing $\Rightarrow$ maximizing

- What happens when the objective is to minimize? We can use the fact that

$$\min_x f(x) = -\max_x -f(x)$$

- In a linear program this means

$$\text{minimize} \quad 6x_1 \quad - \quad 3x_2 \quad + \quad 5x_3$$

becomes:

$$-\ \text{maximize} \quad -6x_1 \quad + \quad 3x_2 \quad - \quad 5x_3$$

That's all we need to convert all linear programs in standard form. . .

Linear Programs: standard & canonical form

Example. . .

$$
\begin{array}{lrcrcrcr}
\text{minimize} & 2x_1 & - & 4x_2 & + & x_3 & & \\
\text{subject to} & 2x_1 & + & 7x_2 & + & x_3 & = & 5 \\
 & 4x_1 & + & x_2 & + & 9x_3 & \leq & 11 \\
 & 3x_1 & + & 4x_2 & + & 2x_3 & = & 8 \\
 & & & & & x_1,\ x_2 & \geq & 0.
\end{array}
$$

This program has one free variable (x_3) and one inequality constraint. It's a minimization problem. . .

Linear Programs: standard & canonical form

We first turn it into a **maximization**. . .

$$\begin{array}{lrcrcrcr} -\text{ maximize} & -2x_1 & + & 4x_2 & - & x_3 & & \\ \text{subject to} & 2x_1 & + & 7x_2 & + & x_3 & = & 5 \\ & 4x_1 & + & x_2 & + & 9x_3 & \leq & 11 \\ & 3x_1 & + & 4x_2 & + & 2x_3 & = & 8 \\ & & & & & x_1,\ x_2 & \geq & 0. \end{array}$$

Just switch the signs in the objective. . .

Linear Programs: standard & canonical form

We then turn the inequality into an **equality** constraint by adding a slack variable. . .

$$
\begin{array}{lrcrcrcrcr}
-\text{ maximize} & -2x_1 & + & 4x_2 & - & x_3 & & & & \\
\text{subject to} & 2x_1 & + & 7x_2 & + & x_3 & & & = & 5 \\
 & 4x_1 & + & x_2 & + & 9x_3 & + & w_1 & = & 11 \\
 & 3x_1 & + & 4x_2 & + & 2x_3 & & & = & 8 \\
 & & & & & x_1,\ x_2,\ w_1 & & & \geq & 0.
\end{array}
$$

Now, we only need to get rid of the free variable. . .

Linear Programs: standard & canonical form

We replace the free variable by a difference of two **positive** ones:

$$\begin{array}{lrcrcrcrcr} -\text{ maximize} & -2x_1 & + & 4x_2 & - & (x_4 - x_5) & & & & \\ \text{subject to} & 2x_1 & + & 7x_2 & + & x_4 - x_5 & & & = & 5 \\ & 4x_1 & + & x_2 & + & 9x_4 - 9x_5 & + & w_1 & = & 11 \\ & 3x_1 & + & 4x_2 & + & 2x_4 - 2x_5 & & & = & 8 \\ & & & & & x_1,\ x_2,\ x_4,\ x_5,\ w_1 & & & \geq & 0. \end{array}$$

- That's it, we've reached a standard form.
- The simplex algorithm is easier to write with this form.

To sum up...

- A linear program in **standard** form is the program

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}. \end{array} \tag{4}$$

  where

  - $\mathbf{c}, \mathbf{x} \in \mathbf{R}^n$ – the objective,
  - $A \in \mathbf{R}^{m \times n}$ and $\mathbf{b} \in \mathbf{R}^m$ – the equality constraints,
  - $\mathbf{x} \geq \mathbf{0}$ means that for $\mathbf{x} = (x_1, \ldots, x_n), x_i \geq 0$ for $1 \leq i \leq n$.

- From now on we focus on

  - **linear constraints** $A\mathbf{x} = \mathbf{b}$,
  - **objective function** $\mathbf{c}^T\mathbf{x}$,

  separately.

- $\mathbf{x} \geq \mathbf{0}$ will reappear when we study convexity.

Linear Equations

Linear Equations

The usual linear equations we know, $m = n$

- In the usual linear algebra setting, A is square of size n and invertible.
- Straightforward: $\{\mathbf{x} \in \mathbf{R}^n | A\mathbf{x} = \mathbf{b}\}$ **is a singleton,** $\{A^{-1}\mathbf{b}\}$.
- Focus: find **efficiently** that **unique** solution. Many methods (Gaussian pivot, Conjugate gradient *etc.*)

In classic statistics, most often $m \gg n$

- A few explicative variables, a lot of observations.
- Generally $\{\mathbf{x} \in \mathbf{R}^n | A\mathbf{x} = \mathbf{b}\} = \emptyset$ so we need to tweak the problem
- Least-squares regression: select $\mathbf{x}_0 \mid \mathbf{x}_0 = \operatorname{argmin} |A\mathbf{x} - \mathbf{b}|^2$
- More advanced, penalized LS regression: $\mathbf{x}_0 = \operatorname{argmin}(|A\mathbf{x} - \mathbf{b}|^2 + \lambda \|\mathbf{x}\|)$

Linear Equations

On the other hand, in an LP setting where usually $m < n$

- $\{\mathbf{x} \in \mathbf{R}^n | A\mathbf{x} = \mathbf{b}\}$ is a wider set of candidates, a convex set.
- In LP, a linear criterion is used to choose one of them.
- In other fields, such as **compressed sensing**, other criterions are used.
- Today we start studying some simple properties of the set $\{\mathbf{x} \in \mathbf{R}^n | A\mathbf{x} = \mathbf{b}\}$.

Linear Equations

- **Linear Equation**: $A\mathrm{x} = \mathrm{b}$, m equations.

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1, \\ \vdots & & & & & & & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & = & b_m. \end{array}$$

- Writing $A = [\mathbf{a}_1, \cdots, \mathbf{a}_n]$ we have n columns $\in \mathbf{R}^m$.
- Add now $\mathbf{b}$: $A_b = [A, b] \in \mathbf{R}^{m \times n+1}$.
- remember: a solution to $A\mathbf{x} = \mathbf{b}$ is a vector $\mathbf{x}$ such that

$$\sum_{i=1}^{n} x_i \mathbf{a}_i = \mathbf{b},$$

that is the $\mathbf{b}$ and $\mathbf{a}$'s should be **linearly dependent** (l.d.) for everything to work.

Linear Equations

Two cases (note that $\mathbf{Rank}(A)$ cannot be $> \mathbf{Rank}(A_b)$)

- (i) $\mathbf{Rank}(A) < \mathbf{Rank}(A_b)$; $\mathbf{b}$ and $\mathbf{a}$'s are **linearly independent** (l.i.). *no solution*.
- (ii) $\mathbf{Rank}(A) = \mathbf{Rank}(A_b) = k$; every column of A_b, $\mathbf{b}$ in particular, can be expressed as a linear combination of k other columns of the matrix $a_{i_1}, \cdots, a_{i_k}$. Namely, $\exists \mathbf{x}$ such that

$$\sum_{j=1}^{k} x_{i_j} \mathbf{a}_{i_j} = \mathbf{b}.$$

In practice

- if $m = n = k$, then there is a unique solution: $\mathbf{x} = A^{-1}\mathbf{b}$;
- Usually $\mathbf{Rank}(A) = k \leq m < n$ and we have a plenty of solutions;
- We assume from now on that $\mathbf{Rank}(A) = \mathbf{Rank}(A_b) = m$.

Linear Equation Solutions

- if $\mathbf{x}_1$ and $\mathbf{x}_2$ are two different solutions, then $\forall \lambda \in \mathbf{R}, \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$ is a solution.
- $\mathbf{Rank}(A) = m$. There are m independent columns. Suppose we reorder them so that $\mathbf{a}_1, \cdots, \mathbf{a}_m$ are linearly independent.
- Then

$$A = \left[\begin{array}{cccc|cccc} a_{11} & a_{12} & \cdots & a_{1m} & a_{1m+1} & a_{1m+2} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2m} & a_{2m+1} & a_{2m+2} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} & a_{mm+1} & a_{mm+2} & \cdots & a_{mn} \end{array}\right] = [B, R]$$

- B is $m \times m$ square, R is $m \times (n-m)$ rectangular.

Linear Equation Solutions

- suppose we divide $\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_\beta \end{bmatrix}$ where $\mathbf{x}_B \in \mathbf{R}^m$ and $\mathbf{x}_\beta \in \mathbf{R}^{m-n}$
- If $A\mathbf{x} = \mathbf{b}$ then $B\mathbf{x}_B + R\mathbf{x}_\beta = \mathbf{b}$. Since B is non-singular, we have

$$\mathbf{x}_B = B^{-1}(\mathbf{b} - R\mathbf{x}_\beta),$$

  which shows that we can assign **arbitrary** values to $\mathbf{x}_\beta$ and obtain different points $\mathbf{x}$ such that $A\mathbf{x} = \mathbf{b}$.

- Solutions are parameterized by $\mathbf{x}_\beta$... a bit problematic since R is the "discarded" part.
- We choose $\mathbf{x}_\beta = \mathbf{0}$ and focus on the choice of B.

Basic Solutions

Basic Solutions

Definition 1. *Consider* $A\mathbf{x} = \mathbf{b}$ *and suppose* $\mathbf{Rank}(A) = m < n$*. Let* $\mathbf{I} = (i_1, \cdots, i_m)$ *be a list of indexes corresponding to* m ***linearly independent*** *columns taken among the* n *columns of* A*.*

- *We call the* m *variables* $\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \cdots, \mathbf{x}_{i_m}$ *of* $\mathbf{x}$ *its* **basic variables**,
- *the other variables are called* ***non-basic****.*

If $\mathbf{x}$ *is a vector such that* $A\mathbf{x} = \mathbf{b}$ *and all* ***its non-basic variables*** ***are equal to 0*** *then* $\mathbf{x}$ *is a basic solution.*

Basic Solutions

- When reordering variables as in the previous slide, and defining $B = [\mathbf{a}_{i_1}, \cdots, \mathbf{a}_{i_m}]$ we can set $\mathbf{x}_\beta = \mathbf{0}$. Then $\mathbf{x}_B = B^{-1}\mathbf{b}$ and

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{0} \end{bmatrix},$$

  and we have a **basic solution**.

- Sidenote: a **basic feasible solution** to an LP Equation (4) is such that $\mathbf{x}$ is basic **and** $\mathbf{x} \geq 0$.

Basic Solutions

- More generally, let
$$B_{\mathbf{I}} = [\mathbf{a}_{i_1}, \cdots, \mathbf{a}_{i_m}],$$
$$R_{\mathbf{O}} = [\mathbf{a}_{o_1}, \cdots, \mathbf{a}_{o_{m-n}}],$$
where $\mathbf{O} = \{1, \cdots, n\} \setminus \mathbf{I} = (o_1, \cdots, o_{m-n})$ is the complementary of $\mathbf{I}$ in $\{1, \cdots, n\}$ in increasing order.

- $\mathbf{I}$ contains the indexes of vectors **in** the basis, $\mathbf{O}$ contains the indexes of vectors **outside** the basis.

- Equivalently set $\mathbf{x_I} = \begin{bmatrix} x_{i_1} \\ \vdots \\ x_{i_m} \end{bmatrix}, \mathbf{x_O} = \begin{bmatrix} x_{o_1} \\ \vdots \\ x_{o_{n-m}} \end{bmatrix}$.

- $A\mathbf{x} = B_{\mathbf{I}}\mathbf{x_I} + R_{\mathbf{O}}\mathbf{x_O}$

Basic Solutions

The two things to remember so far:

- **A list $\mathbf{I}$ of m independent columns $\leftrightarrow$ One basic solution $\mathbf{x}$, with $\mathbf{x}_\mathbf{I} = B_\mathbf{I}^{-1}\mathbf{b}$ and $\mathbf{x}_\mathbf{O} = \mathbf{0}$**
- We are **not** interested in **all** basic solutions, only a subset: **basic feasible solutions**.

Basic Solutions: Degeneracy

Definition 2. *A* ***basic*** *solution to* $A\mathbf{x} = \mathbf{b}$ *is* ***degenerate*** *if one or more of the* m ***basic*** *variables is equal to zero.*

- For a **basic solution**, $\mathbf{x_O}$ is always $\mathbf{0}$. On the other hand, we do not expect elements of $\mathbf{x_I}$ to be zero.
- This is **degeneracy** which appears whenever there is one or more components of $\mathbf{x_I}$ which are zero.

Basic Solutions: Example

- Consider $A\mathbf{x} = \mathbf{b}$ where

$$A = \begin{bmatrix} 1 & 2 & 1 & 0 & 3 \\ 0 & 1 & 2 & 1 & 3 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

We start by choosing $\mathbf{I}$:

- $\mathbf{I} = (1, 2)$. $B_{\mathbf{I}} = [\mathbf{a}_1, \mathbf{a}_2] = \left[\begin{smallmatrix} 1 & 2 \\ 0 & 1 \end{smallmatrix}\right] \rightarrow \mathbf{x}_{\mathbf{I}} = \left[\begin{smallmatrix} -1 \\ 1 \end{smallmatrix}\right]; \mathbf{x} = \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ is **basic**.

- $\mathbf{I} = (1, 4)$. $B_{\mathbf{I}} = [\mathbf{a}_1, \mathbf{a}_4] = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right] \rightarrow \mathbf{x}_{\mathbf{I}} = \left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right]; \mathbf{x} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ is **basic**.

- $\mathbf{I} = (2, 5)$. $B_{\mathbf{I}} = [\mathbf{a}_2, \mathbf{a}_5] = \left[\begin{smallmatrix} 1 & 3 \\ 0 & 3 \end{smallmatrix}\right] \rightarrow \mathbf{x}_{\mathbf{I}} = \begin{bmatrix} 0 \\ \frac{1}{3} \end{bmatrix}; \mathbf{x} = \begin{bmatrix} 0 \\ \mathbf{0} \\ 0 \\ 0 \\ \frac{1}{3} \end{bmatrix}$ is **degenerate basic**

  note that $\mathbf{a}_5$ and $\mathbf{b}$ are colinear...

Non-degeneracy

Theorem 1. *A necessary and sufficient condition for the existence and non-degeneracy of all basic solutions of $A\mathbf{x} = \mathbf{b}$ is the* ***linear independence of every set of*** m ***columns of*** A_b*, the augmented matrix.*

Proof. • **Proof strategy**: $\Rightarrow$ the existence of all possible basic solutions is already a good sign: all families of m columns of A are l.i. What we need is show that $m-1$ columns of A plus $\mathbf{b}$ are also l.i.

• $\Leftarrow$ if all m columns choices are independent, basic solutions exist, and are non-degenerate because $\mathbf{b}$ is l.i. with any combination of $m-1$ columns.

■

Non-degeneracy

Proof.

- $\Rightarrow$: Let $I = (i_1, \cdots, i_m)$ a family of indexes.
  - The basic solution associated with I exists and is non-degenerate. $\mathbf{b} \neq 0$
  - Hence by definition $\{\mathbf{a}_{i_1}, \cdots, \mathbf{a}_{i_m}\}$ is l.i. and $\mathbf{b} = \sum_{k=1}^{m} x_k \mathbf{a}_{i_k}$.
  - For a given r, suppose $\{\mathbf{a}_{i_1}, \cdots, \mathbf{a}_{i_{r-1}}, \mathbf{a}_{i_{r+1}}, \cdots, \mathbf{a}_{i_m}, \mathbf{b}\}$ is l.d.
  - Then $\exists(\alpha_1, \cdots, \alpha_{r-1}, \alpha_{r+1}, \alpha_m)$ and β such that

$$\beta \mathbf{b} + \sum_{k=1, k\neq r}^{m} \alpha_k \mathbf{a}_{i_k} = \mathbf{0}.$$

    Note that necessarily $\beta \neq 0$ (otherwise $\{\mathbf{a}_{i_1}, \cdots, \mathbf{a}_{i_{r-1}}, \mathbf{a}_{i_{r+1}}, \cdots, \mathbf{a}_{i_m}\}$ is l.d)
  - Contradiction: degenerate solution for I, $(-\frac{\alpha_1}{\beta}, \cdots, -\frac{\alpha_{r-1}}{\beta}, 0, -\frac{\alpha_{r+1}}{\beta}, -\frac{\alpha_m}{\beta})$
- $\Leftarrow$: Let $I = (i_1, \cdots, i_m)$ a family of indexes.
  - A basic solution exists, $\sum_{k=1}^{m} x_k \mathbf{a}_{i_k} = \mathbf{b}$
  - Suppose it is degenerate, i.e. $x_r = 0$. Then $\sum_{k=1, k\neq r}^{m} x_k \mathbf{a}_{i_k} - \mathbf{b} = \mathbf{0}$
  - Contradiction: $\{\mathbf{a}_{i_1}, \cdots, \mathbf{a}_{i_{r-1}}, \mathbf{a}_{i_{r+1}}, \cdots, \mathbf{a}_{i_m}, \mathbf{b}\}$, of size m, is l.d.

■

Non-degeneracy

Theorem 2. *Given a basic solution to $A\mathbf{x} = \mathbf{b}$ with basic variables $x_{i_1}, \cdots, x_{i_m}$, a necessary and sufficient condition for the solution to be non-degenerate is the l.i. of $\mathbf{b}$ with every subset of $m-1$ columns of $\{\mathbf{a}_{i_1}, \cdots, \mathbf{a}_{i_m}\}$*

- In our previous example,

$$A = \begin{bmatrix} 1 & 2 & 1 & 0 & 3 \\ 0 & 1 & 2 & 1 & 3 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \; m = 2.$$

- Hence if $\mathbf{I} = (2,5)$, $[\mathbf{b}, \mathbf{a}_2]$ and $[\mathbf{b}, \mathbf{a}_5]$ should be of rank 2 for the solution not to be degenerate. Yet $[\mathbf{b}, \mathbf{a}_5] = \left[\begin{smallmatrix} 1 & 3 \\ 1 & 3 \end{smallmatrix}\right]$ is clearly of rank 1.

Hyperplanes

Hyperplane

Definition 3. *A hyperplane in* $\mathbf{R}^n$ *is defined by a vector* $\mathbf{c} \neq \mathbf{0} \in \mathbf{R}^n$ *and a scalar* $z \in \mathbf{R}$ *as the set* $\{\mathbf{x} \in \mathbf{R}^n | \mathbf{c}^T\mathbf{x} = z\}$.

$z = 0$**,**

- A hyperplane $H_{\mathbf{c},z}$ contains $\mathbf{0}$ iff $z = 0$.
- In that case $H_{\mathbf{c},0}$ is a **vector subspace** and $\dim(H_{\mathbf{c},0}) = n - 1$

$z \neq 0$**,**

- For $\mathbf{x}_1, \mathbf{x}_2$ easy to check that $\mathbf{c}^T(\mathbf{x}_1 - \mathbf{x}_2) = 0$. In other words $\mathbf{c}$ is orthogonal to vectors lying in the hyperplane.
- $\mathbf{c}$ is called the **normal** of the hyperplane

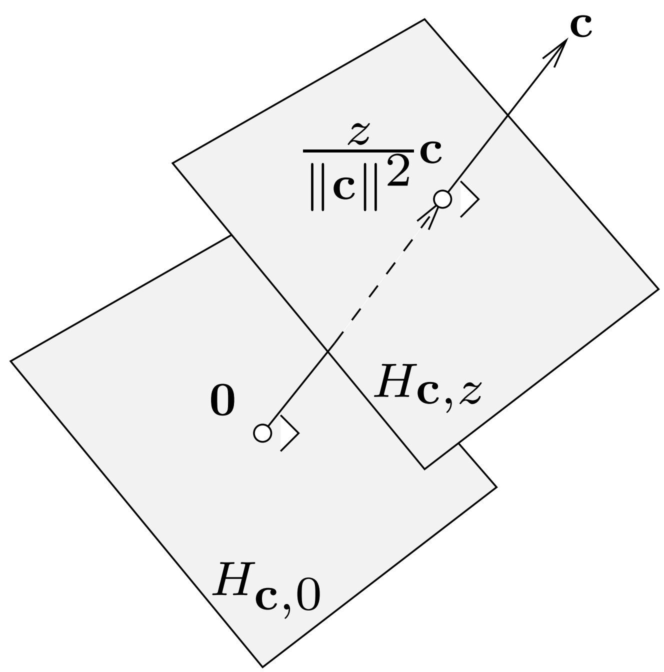
Affine Subspace

Definition 4. *Let V be a vector space and let L be a vector subspace of V. Then given $\mathbf{x} \in V$, the translation $T = L + \mathbf{x} = \{\mathbf{u} + \mathbf{x}, \mathbf{u} \in L\}$ is called an affine subspace of V.*

- the **dimension** of T is the dimension of L.
- T is **parallel** to L.

Affine Hyperplane

- For $\mathbf{c} \neq \mathbf{0}$, $H_{\mathbf{c},0}$ is a Vector subspace of $\mathbf{R}^n$ of dimension $n-1$.
- When $z \neq 0$, $H_{\mathbf{c},z}$ is an affine **hyperplane**: it's easy to see that $H_{\mathbf{c},z} = H_{\mathbf{c},0} + \frac{z}{\|\mathbf{c}\|^2}\mathbf{c}$



A bit of Topology and Halfspaces

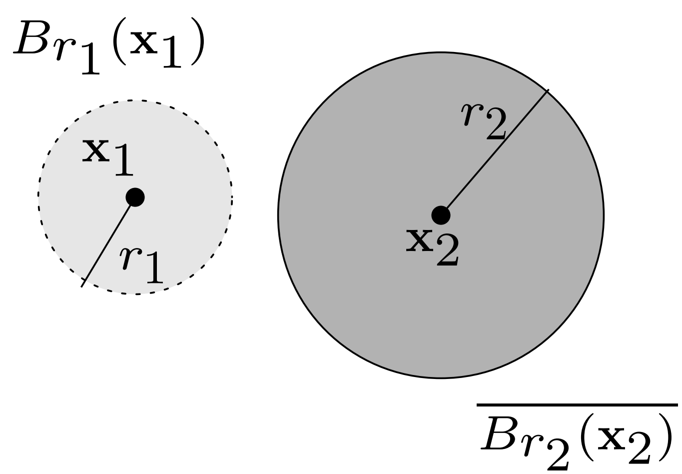
A bit of topology: open and closed balls

- The n dimensional open ball centered at $\mathbf{x}_0$ with radius r is defined as

$$B_r(\mathbf{x}_0) = \{x \in \mathbf{R}^n \text{ s.t. } |\mathbf{x} - \mathbf{x}_0| < r\},$$

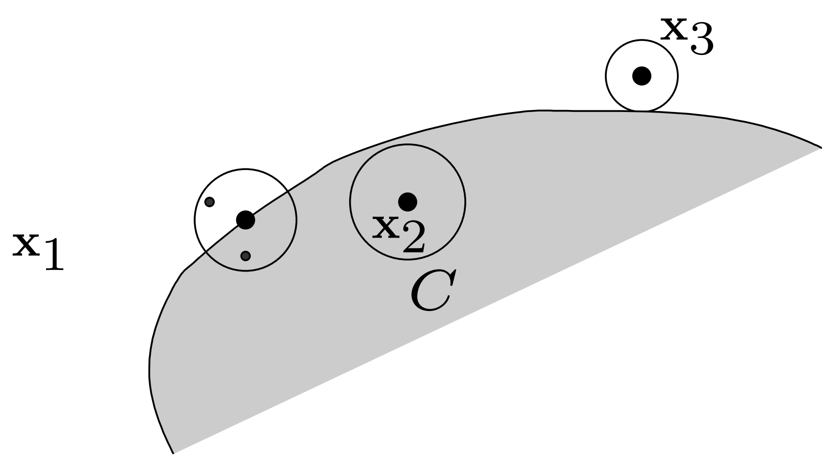
- its closure

$$\overline{B_r(\mathbf{x}_0)} = \{x \in \mathbf{R}^n \text{ s.t. } |\mathbf{x} - \mathbf{x}_0| \leq r\},$$



A bit of topology: boundary

- Let $S \subset \mathbf{R}^n$. A point $\mathbf{x}$ is a **boundary point** of S if every open ball centered at $\mathbf{x}$ contains both a point in S and a point in $\mathbf{R}^n \setminus S$.
- A boundary point can either be in S or not in S.



- x_1 is a boundary point, x_2 and x_3 are not.

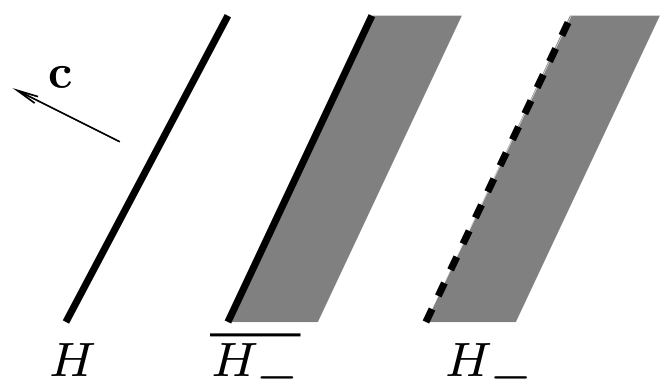
A bit of topology: open and closed sets

- The set of all boundary points of S is the **boundary** ∂S of S.
- A set is **closed** if $\partial S \subset S$. A set is *open* if $\mathbf{R}^n \setminus S$ is closed.
- Note that there are sets that are **neither** open nor close.
- The **closure** $\overline{S}$ of a set S is $S \cup \partial S$
- The **interior** S^o of a set S is $S \setminus \partial S$
- A set S is *closed* iff $S = \overline{S}$ and *open* iff $S = S^o$.

Halfspaces

- For a hyperplane H, its complement in $\mathbf{R}^n$ is the union of two sets called **open halfspaces**;
$$\mathbf{R}^n \setminus H = H_+ \cup H_-$$
where
$$\begin{aligned} H_+ &= \{\mathbf{x} \in \mathbf{R}^m | \mathbf{c}^T\mathbf{x} > z\} \\ H_- &= \{\mathbf{x} \in \mathbf{R}^m | \mathbf{c}^T\mathbf{x} < z\} \end{aligned}$$

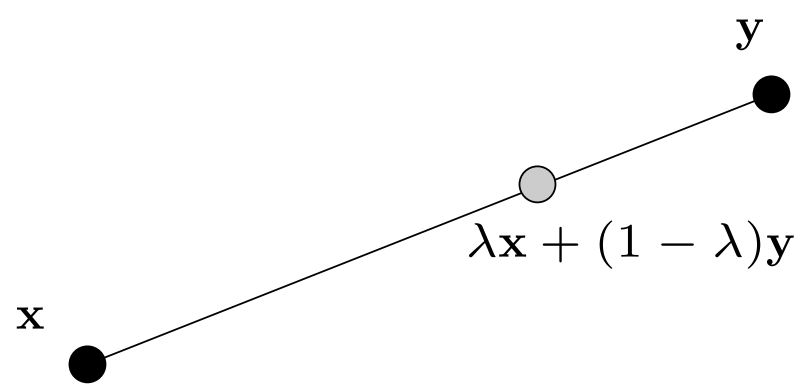
- $\overline{H_+} = H_+ \cup H$ and $\overline{H_-} = H_- \cup H$ are **closed** halfspaces.



Convex sets & extreme points

Definition

- Convexity starts by defining segments



$$[\mathbf{x}, \mathbf{y}] = \lambda\mathbf{x} + (1-\lambda)\mathbf{y}, \lambda \in [0, 1]$$

.

Definition 5. *A set C is said to be* **convex** *if for all* $\mathbf{x}$ *and* $\mathbf{y}$ *in C the segment* $[\mathbf{x}, \mathbf{y}] \subset C$.

Examples

- $\mathbf{R}^n$ is trivially convex and so is any vector subspace V of $\mathbf{R}^n$.
- For $\mathbf{R}^n \ni \mathbf{c} \neq \mathbf{0}$ and $z \in \mathbf{R}$, $H_{\mathbf{c},z}$ is convex
- The halfspaces $H^+_{\mathbf{c},z}$ and $H^-_{\mathbf{c},z}$ are **open convex sets**, their respective closures are **closed convex sets**.
- Let $\mathbf{x}_1, \mathbf{x}_2 \in B_r(\mathbf{x}_0), \lambda \in [0,1]$ then

$$|(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) - \mathbf{x}_0| = |\lambda(\mathbf{x}_1 - \mathbf{x}_0) + (1-\lambda)(\mathbf{x}_2 - \mathbf{x}_0)| < \lambda r + (1-\lambda) r = r.$$

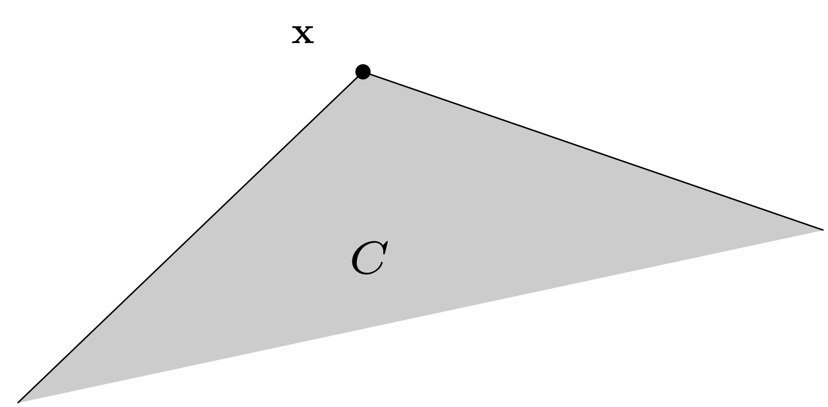
hence $B_r(\mathbf{x}_0)$ and similarly $\overline{B_r(\mathbf{x}_0)}$ are convex

Extreme points

Definition 6. *A point* $\mathbf{x}$ *of a convex set* C *is said to be an* ***extreme point*** *of* C *if*

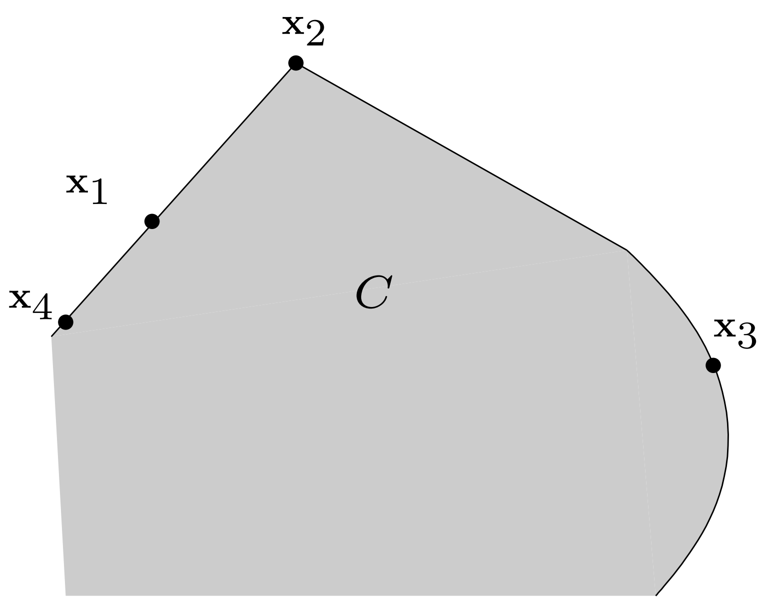
$$\left(\exists \mathbf{x}_1, \mathbf{x}_2 \in C \mid \mathbf{x} = \tfrac{\mathbf{x}_1+\mathbf{x}_2}{2}\right) \Rightarrow \mathbf{x}_1 = \mathbf{x}_2 = \mathbf{x}.$$

- intuitively $\mathbf{x}$ is not part of an **open** segment of two other points $\mathbf{x}_1, \mathbf{x}_2$.
- other definitions use $0 < \lambda < 1, \mathbf{x} = \lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$ but the one above is equivalent & easier to remember.



Extreme points

- an extreme point is a boundary point but **the converse is not true in general**.



- $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ are all boundary points. Only $\mathbf{x}_2$ and $\mathbf{x}_3$ are extreme. $\mathbf{x}_1$ for instance can be written as $\lambda \mathbf{x}_2 + (1 - \lambda)\mathbf{x}_4$

Hyperplanes and Convexity: Isolation and Support

Boundaries of Hyperplanes and Halfspaces

- Hyperplanes are closed
  - We can actually show that $H_{\mathbf{c},z} \subset \partial H_{\mathbf{c},z}$, namely any point of $H_{\mathbf{c},z}$ is a boundary point:
    - ▷ let $\mathbf{x} \in H_{\mathbf{c},z}$ and $B_r(\mathbf{x})$ an open ball centered in $\mathbf{x}$.
    - ▷ let $\mathbf{y}_1 = \mathbf{x} + \frac{r}{2|\mathbf{c}|^2}\mathbf{c}$. Then $\mathbf{c}^T\mathbf{y}_1 = z + \frac{r}{2} > z$ hence $\mathbf{y}_1 \notin H_{\mathbf{c},z}$ but $\mathbf{y}_1 \in B_r(\mathbf{x})$,
    - ▷ let $\mathbf{z} \in H_{\mathbf{c},z}, \mathbf{z} \neq \mathbf{x}$, and $\mathbf{y}_2 = \mathbf{x} + r\frac{\mathbf{x}-\mathbf{z}}{2|\mathbf{x}-\mathbf{z}|}$, hence $\mathbf{y}_2 \in H_{\mathbf{c},z}$ and $\mathbf{y}_2 \in B_r(\mathbf{x})$.
  - We could also have raised the fact that for $\mathbf{x}_i$ a converging sequence of $H_{\mathbf{c},z}$ we have that $\mathbf{c}^T \lim_{i\to\infty} \mathbf{x}_i = \lim_{i\to\infty} \mathbf{c}^T\mathbf{x}_i = z$.

- The boundary of a halfspace is the corresponding hyperplane, i.e.

$$\partial H_- = \partial H_+ = H.$$

- The interior H^o of a hyperplane is empty as $H^o = H \setminus \partial H$.

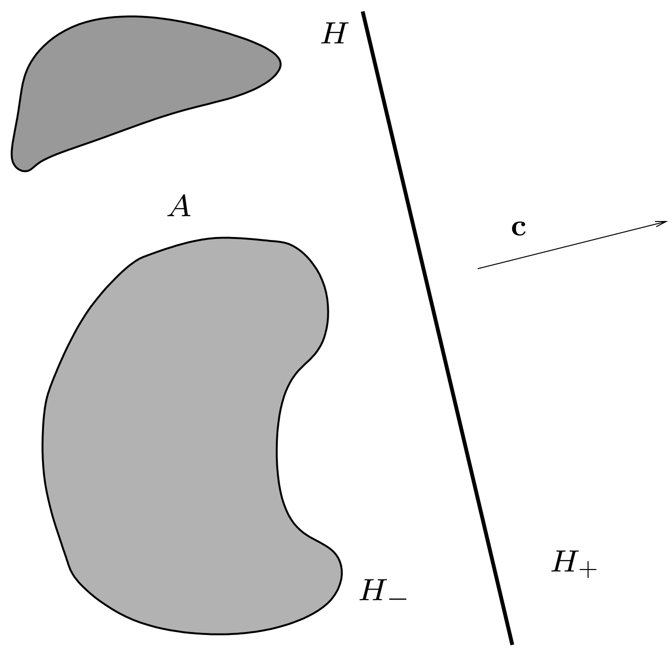
Hyperplanes, halfspaces and convexity

Lemma 1. *(i) All hyperplanes are convex;*
(ii) The halfspaces $H^+_{\mathbf{c},z}, H^-_{\mathbf{c},z}, \overline{H^+_{\mathbf{c},z}}, \overline{H^-_{\mathbf{c},z}}$ *are convex;*
(iii) Any ***intersection*** *of convex sets is convex;*
(iv) The set of all ***feasible solutions of a linear program*** *is a* ***convex*** *set.*

Proof. (i) $c^T(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) = (\lambda + (1-\lambda))\, z = z$.
(ii) same as above by replacing equality by inequalities.
(iii) Let $C = \cap_{i \in I} C_i$. Let $\mathbf{x}_1, \mathbf{x}_2 \in C$. Then for $\lambda \in [0,1], \forall i \in I, (\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \in C_i$, hence $(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \in C$.
(iv) The set of feasible points to an LP problem is the intersection of hyperplanes $\mathbf{r}_i^T \mathbf{x} = b_i$ and halfspaces $\mathbf{r}_j^T \mathbf{x} \gtreqless b_j$ and is hence convex by (iii). ∎

Isolation

Definition 7. *Let $A \subset \mathbf{R}^n$ be a set and let $H \subset \mathbf{R}^n$ be a affine hyperplane. H is said to* ***isolate*** *A if A is contained in one of the closed subspaces $\overline{H_-}$ or $\overline{H_+}$. H* ***strictly isolates*** *A if A is contained in one of the open halfspaces H_- or H_+.*



Isolation Theorem

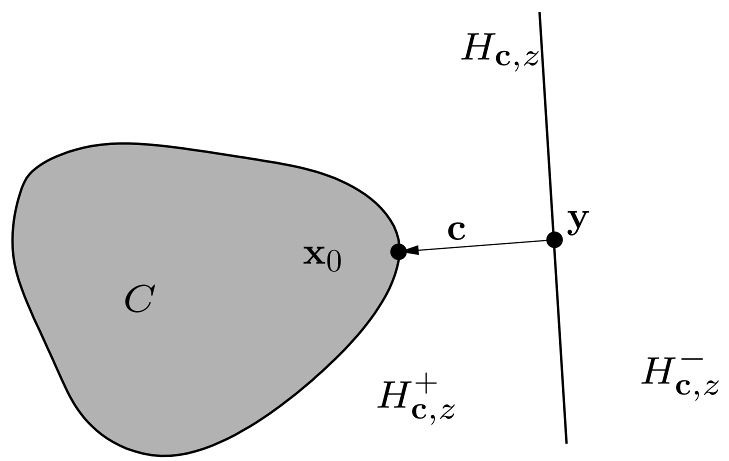
Theorem 3. *Let C be a* ***closed convex set*** *and $\mathbf{y}$ a point not in C. Then there is a hyperplane $H_{\mathbf{c},z}$ that contains $\mathbf{y}$ and such that $C \subset H^{-}_{\mathbf{c},z}$ or $C \subset H^{+}_{\mathbf{c},z}$*

- (Bar02,II.1.6) has a more general result when C is **open**. The proof is longer and we won't use it.
- **Proof strategy**: build a suitable hyperplane and show it satisfies the property.

Isolation Theorem : Proof

Proof. • **Define the hyperplane**:

- Let $\delta = \inf_{x \in C} |\mathbf{x} - \mathbf{y}| > 0$.
- The continuous function $\mathbf{x} \to |\mathbf{x} - \mathbf{y}|$ on the closed set $\overline{B_{2\delta}(\mathbf{y})}$ achieves its minimum at a point $\mathbf{x}_0 \in C$.
- One can prove that necessarily $\mathbf{x} \in \partial C$.
- Let $\mathbf{c} = \mathbf{x}_0 - \mathbf{y}, z = \mathbf{c}^T\mathbf{y}$ and consider $H_{\mathbf{c},z}$. Clearly $\mathbf{y} \in H_{\mathbf{c},z}$.



Isolation Theorem : Proof

- **Show that** $C \subset H^+_{\mathbf{c},z}$:
  - Let $\mathbf{x} \in C$. Since $\mathbf{x}_0 \in C$, for $\lambda \in [0,1]$,

$$\lambda \mathbf{x} + (1-\lambda)\mathbf{x}_0 = \mathbf{x}_0 + \lambda(\mathbf{x} - \mathbf{x}_0) \in C.$$

  - By definition of $\mathbf{x}_0$, $|\,(\mathbf{x}_0 + \lambda(\mathbf{x} - \mathbf{x}_0)) - \mathbf{y}|^2 \geq |\mathbf{x}_0 - \mathbf{y}|^2$,
  - thus by definition of $\mathbf{c} = \mathbf{x}_0 - \mathbf{y}$,

$$|\lambda(\mathbf{x} - \mathbf{x}_0) + \mathbf{c}|^2 \geq |\mathbf{c}|^2,$$

  - thus $2\lambda \mathbf{c}^T(\mathbf{x} - \mathbf{x}_0) + \lambda^2 |\mathbf{x} - \mathbf{x}_0|^2 \geq 0$,
  - Letting $\lambda \to 0$ we have that $\mathbf{c}^T(\mathbf{x} - \mathbf{x}_0) \geq 0$, hence

$$\mathbf{c}^T\mathbf{x} \geq \mathbf{c}^T\mathbf{x}_0 = \mathbf{c}^T(\mathbf{y} + \mathbf{c}) = z + |\mathbf{c}|^2 = z + \delta^2 > z$$

■

Supporting Hyperplane

Definition 8. *Let* $\mathbf{y}$ *be a* ***boundary*** *point of a convex set* C. *A hyperplane* $H_{\mathbf{c},z}$ *is called a* ***supporting hyperplane*** *of* C *at* $\mathbf{y}$ *if* $\mathbf{y} \in H_{\mathbf{c},z}$ *and either* $C \subseteq \overline{H_{\mathbf{c},z}^{+}}$ *or* $C \subseteq \overline{H_{\mathbf{c},z}^{-}}$.

Theorem 4. *If* $\mathbf{y}$ *is a boundary point of a closed convex set* C *then there is at least one supporting hyperplane at* $\mathbf{y}$.

- **Proof strategy**: use the isolation theorem on a sequence of points that converge to a boundary point.

Supporting Hyperplane : Proof

Proof. Since $\mathbf{y} \in \partial C, \forall k \in \mathbf{N}$, $\exists \mathbf{y}_k \in B_{\frac{1}{k}}(\mathbf{y})$ such that $\mathbf{y}_k \notin C$. $(\mathbf{y}_k)$ is thus a sequence of $\mathbf{R}^n \setminus C$ that converges to $\mathbf{y}$. Let $\mathbf{c}_k$ be the sequence of corresponding normal vectors constructed according to the proof of Theorem 3, normalized so that $|\mathbf{c}_k| = 1$ and C is in the halfspace $\{\mathbf{x} \,|\, \mathbf{c}_k^T \mathbf{x} \geq \mathbf{c}_k^T \mathbf{y}_k\}$. Since $(\mathbf{c}_k)$ is a bounded sequence in a compact space, there exists a subsequence $\mathbf{c}_{k_j}$ that converges to a point $\mathbf{c}$. Let $z = \mathbf{c}^T \mathbf{y}$. For any $\mathbf{x} \in C$,

$$\mathbf{c}^T \mathbf{x} = \lim_{j \to \infty} \mathbf{c}_{k_j}^T \mathbf{x} \geq \lim_{j \to \infty} \mathbf{c}_{k_j}^T \mathbf{y}_{k_j} = \mathbf{c}^T \mathbf{y} = z,$$

thus $C \subset \overline{H_{\mathbf{c},z}^{+}}$ ■

Definition 9. *A set $A \subset \mathbf{R}^n$ is said to be* ***bounded from below*** *if for all $1 \leq j \leq n$,*

$$\inf \left\{ \mathbf{x}_j \,|A \ni \mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T \right\} > -\infty.$$

- Any bounded set is bounded from below
- More importantly, $\mathbf{R}^n_+ = \{\mathbf{x}|\mathbf{x} \geq 0\}$ is bounded from below.
- the LP set of solutions $\{\mathbf{x} \in \mathbf{R}^n \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ is **convex** & **bounded from below**.

Supporting Hyperplane and Extreme Points

Theorem 5. *Let C be a closed convex set which is bounded from below. Then every supporting hyperplane of C contains an extreme point of C .*

- **Proof strategy**: Show that for a supporting hyperplane H, an extreme point of the convex subset $H \cap C$ is an extreme point of C. Find an extreme point of $H \cap C$.

Supporting Hyperplane and Extreme Points: Proof

Proof. • Let $H_{\mathbf{c},z}$ be a supporting hyperplane at $\mathbf{y} \in C$. Let us write $A = H_{\mathbf{c},z} \cap C$ which is non-empty since it contains $\mathbf{y}$.

- **an extreme point of A is an extreme point of C**
  - suppose $\mathbf{x} \in A$, that is $\mathbf{c}^T\mathbf{x} = z$, is **not** an ext. point of C, i.e $\exists \mathbf{x}_1 \neq \mathbf{x}_2 \in C$ such that $\mathbf{x} = \frac{\mathbf{x}_1+\mathbf{x}_2}{2}$.
  - If $\mathbf{x}_1 \notin A$ **or** $\mathbf{x}_2 \notin A$ then $\frac{1}{2}\mathbf{c}^T(\mathbf{x}_1 + \mathbf{x}_2) > z = \mathbf{c}^T\mathbf{x}$ hence $\mathbf{x}_1, \mathbf{x}_2 \in A$ and thus $\mathbf{x}$ is **not** an ext. point of A.

Supporting Hyperplane and Extreme Points: Proof

- **look now for an extreme point of A. We use mainly $A \subset H_{\mathbf{c},z} \cap \mathbf{R}^m_+$**
  - if A is a singleton, namely $A = \{\mathbf{y}\}$, then $\mathbf{y}$ is obviously extreme.
  - if not, **narrow down recursively**:
    - ▷ $A^1 = \operatorname{argmin}\{\mathbf{a}_1 \mid \mathbf{a} \in A\}$. Since $A \subset C$ and C is bounded from below the closed set A^1 is well defined as the set of points which achieve this minimum.
    - ▷ If A^1 is still not a singleton, we narrow further:

$$A^j = \operatorname{argmin}\{\mathbf{a}_j \mid \mathbf{a} \in A^{j-1}\}.$$

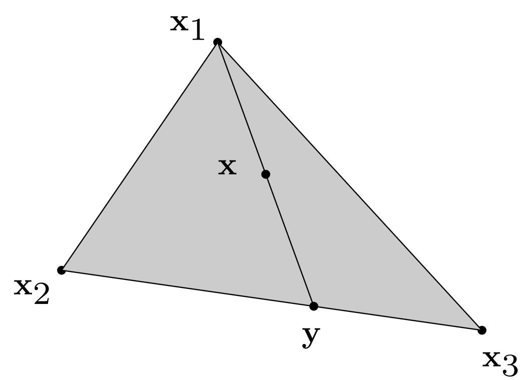
    - ▷ Since $A \subset \mathbf{R}^n$, this process must stop after $k \leq n$ iterations (after n iterations the n variables of points in A^n are uniquely defined). We have $A^k \subseteq A^{k-1} \subseteq A^1 \subseteq A$ and write $A^k = \{\mathbf{a}^k\}$.
  - Suppose $\exists \mathbf{x}^1 \neq \mathbf{x}^2 \in A$ such that $\mathbf{a}^k = \frac{\mathbf{x}^1+\mathbf{x}^2}{2}$. In particular $\forall i \leq k, \mathbf{a}^k_i = \frac{\mathbf{x}^1_i+\mathbf{x}^2_i}{2}$.
  - Since $\mathbf{a}^k_1$ is an **infimum**, $\mathbf{x}^1_i = \mathbf{x}^2_i = \mathbf{a}^k_1$ and $\mathbf{x}^1, \mathbf{x}^2 \in A^1$.
  - By the same argument **applied recursively** we have that $\mathbf{x}^1, \mathbf{x}^2 \in A^j$ and finally A^k which by construction is $\{\mathbf{a}^k\}$, hence $\mathbf{x}_1 = \mathbf{x}_2 = \mathbf{a}^k$, a contradiction, and $\mathbf{a}^k$ is our extreme point.

■

Convex Hulls & Carathéodory's Theorem

Convex combinations

Definition 10. *Let* $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k\}$ *be a set of points. Let* $\alpha_1, \cdots, \alpha_k$ *be a family of nonnegative weights such that* $\sum_1^k \alpha_i = 1$. *Then* $\mathbf{x} = \sum_1^k \alpha_i \mathbf{x}_i$ *is called a* ***convex combination*** *of the points* $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k$.



Let's illustrate this statement with a point $\mathbf{x}$ in a triangle $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$.

- Let $\mathbf{y}$ be the intersection of $(\mathbf{x}_1, \mathbf{x})$ with $[\mathbf{x}_2, \mathbf{x}_3]$. $\mathbf{y} = p\mathbf{x}_2 + q\mathbf{x}_3$ with $p = \frac{|\mathbf{x}_2 - \mathbf{y}|}{|\mathbf{x}_3 - \mathbf{x}_2|}$ and $q = \frac{|\mathbf{x}_3 - \mathbf{y}|}{|\mathbf{x}_3 - \mathbf{x}_2|}$.
- On the other hand, $\mathbf{x} = l\mathbf{x}_1 + k\mathbf{y}$ with $l = \frac{|\mathbf{x}_1 - \mathbf{x}|}{|\mathbf{x}_1 - \mathbf{y}|}$ and $k = \frac{|\mathbf{y} - \mathbf{x}|}{|\mathbf{x}_1 - \mathbf{y}|}$.
- Finally $\mathbf{x} = l\mathbf{x}_1 + pk\mathbf{x}_2 + qk\mathbf{x}_3$, and $l + pk + qk = 1$.

Convex hull

Definition 11. *The* ***convex hull*** $\langle A \rangle$ *of a set* A *is the minimal convex set that contains* A.

Lemma 2. *(i) if* $A \neq \emptyset$ *then* $\langle A \rangle \neq \emptyset$
(ii) if $A \subset B$ *then* $\langle A \rangle \subset \langle B \rangle$
(iii) $\langle A \rangle$ *is the intersection of all convex sets that contain* A.
(iv) if A *is convex then* $\langle A \rangle = A$

Convex hull $\Leftrightarrow$ all convex combinations

Theorem 6. *The convex hull of a set of points* $\{\mathbf{x}_1, \cdots, \mathbf{x}_k\}$ *is the set of all convex combinations of* $\mathbf{x}_1, \cdots, \mathbf{x}_k$.

Proof.

- Let $A = \{\mathbf{x} \,|\, \mathbf{x} = \sum_1^k \alpha_i \mathbf{x}_i, \alpha_i \geq 0, \sum_1^k \alpha_i = 1\}$; $B = \langle \{\mathbf{x}_1, \cdots, \mathbf{x}_k\} \rangle$
  - It's easy to prove that A is convex: Let $\mathbf{x} = \sum_1^k \alpha_i \mathbf{x}_i$ and $\mathbf{y} = \sum_1^k \beta_i \mathbf{x}_i$ be two points of A. Then $\lambda \mathbf{x} + (1-\lambda)\mathbf{y}$ can be written as

$$\sum_{i=1}^{k} (\lambda \alpha_i + (1-\lambda)\beta_i)\, \mathbf{x}_i \in A$$

  - $B \subseteq A$: A is convex and contains each point $\mathbf{x}_i$ since

$$\mathbf{x}_i = \sum_{j=1}^{k} \delta_{ij} \mathbf{x}_j.$$

Convex hull ⇔ all convex combinations

- $A \subseteq B$: by induction on k. if $k = 1$ then $B_1 = \langle\{\mathbf{x}_1\}\rangle$ and $A_1 = \{\mathbf{x}_1\}$. By Lemma 2 $A_1 \subseteq B_1$. Suppose that the claim holds for any family of $k-1$ points, i.e. $A_{k-1} \subseteq B_{k-1}$. Let now $\mathbf{x} \in A_k$ such that

$$\mathbf{x} = \sum_{i=1}^{k} \alpha_i \mathbf{x}_i.$$

If $\mathbf{x} = \mathbf{x}_k$ then trivially $\mathbf{x} \in B_k$. If $\mathbf{x} \neq \mathbf{x}_k$ then $\alpha_k \neq 1$ and we have that

$$\frac{\sum_{i=1}^{k-1} \alpha_i}{1 - \alpha_k} = 1.$$

Consider $\mathbf{y} = \sum_{i=1}^{k-1} \frac{\alpha_i}{1-\alpha_k} \mathbf{x}_i$. $\mathbf{y} \in B_{k-1}$ by the induction hypothesis. Since $\{\mathbf{x}_1, \cdots, \mathbf{x}_{k-1}\} \subset \{\mathbf{x}_1, \cdots, \mathbf{x}_k\}$, $B_{k-1} \subseteq B_k$ by Lemma2. Since B_k is convex and both $\mathbf{y}, \mathbf{x}_k \in B_k$, so is $\mathbf{x} = (1-\alpha_k)\mathbf{y} + \alpha_k \mathbf{x}_k$.

■

Polytope, Polyhedrons

Definition 12. *The convex hull of a finite set of points in* $\mathbf{R}^n$ *is called a* ***polytope****.*

Let $\mathbf{r}_1, \cdots, \mathbf{r}_m$ *be vectors from* $\mathbf{R}^n$ *and* $b_1, \cdots, b_m$ *be numbers. The set*

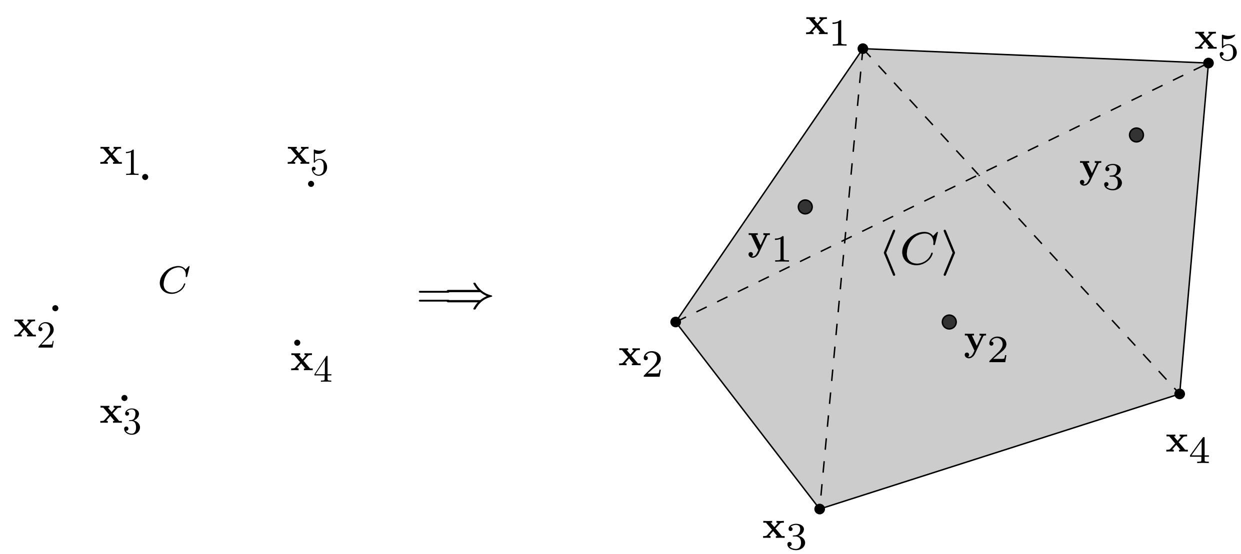
$$P = \left\{ \mathbf{x} \in \mathbf{R}^n \mid \mathbf{r}_i^T \mathbf{x} \leq b_i \ , \ i = 1, \cdots, n \right\}$$

is called a ***polyhedron****.*

- A few comments:
  - **bounded polyhedron** $\Leftrightarrow$ **polytope**: TBP **Weyl-Minkowski** theorem.
  - polytopes are generated by a finite set of points. $\overline{B_r(\mathbf{x})}$ is **not** a polytope.
  - a polyhedron is exactly the set of **feasible solutions of an LP**.

Carathéodory's Theorem

- Start with the example of $C = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} \subset \mathbf{R}^2$ and its hull $\langle C \rangle$.



  - $\mathbf{y}_1$ can be written as a convex combination of $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ (or $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_5$);
  - $\mathbf{y}_2$ can be written as a convex combination of $\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4$;
  - $\mathbf{y}_3$ can be written as a convex combination of $\mathbf{x}_1, \mathbf{x}_4, \mathbf{x}_5$;

- For a set C of 5 points in $\mathbf{R}^{\mathbf{2}}$ there seems to be always a way to write a point $\mathbf{y} \in \langle C \rangle$ as the convex combination of $\mathbf{2+1=3}$ of such points.

- Is this result still valid for **general hulls** $\langle S \rangle$ (not necessarily polytopes but also balls etc..) and **higher dimensions**?

Carathéodory's Theorem

Theorem 7. *Let $S \subset \mathbf{R}^{n}$. Then every point $\mathbf{x}$ of $\langle S\rangle$ can be represented as a convex combination of $\boldsymbol{n+1}$ points from S,*

$$\mathbf{x} = \alpha_1 \mathbf{x}_1 + \cdots + \alpha_{n+1}\mathbf{x}_{n+1}, \sum_{i=1}^{n+1} \alpha_i = 1, \alpha_i \geq 0.$$

alternative formulation:

$$\langle S\rangle = \bigcup_{C\subset S, \mathrm{card}(C)=n+1} \langle C\rangle.$$

- **Proof strategy**: show that when a point is written as a combination of m points and $m > n+1$, it is possible to write it as a combination of $m-1$ points by solving a homogeneous linear equation of $n+1$ equations in $\mathbf{R}^m$.

Proof.

- $(\supset)$ is direct.
- $(\subset)$ any $\mathbf{x} \in \langle S \rangle$ can be written as a convex combination of p points, $\mathbf{x} = \alpha_1 \mathbf{x}_1 + \cdots \alpha_p \mathbf{x}_p$. We can assume $\alpha_i > 0$ for $i = 1, \cdots, p$.
  - If $p < n+1$ then we add terms $0\mathbf{x}_{p+1} + 0\mathbf{x}_{p+2} + \cdots$ to get $n+1$ terms.
  - If $p > n+1$, we build a new combination with one term less:
    - ▷ let $A = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_m \\ 1 & 1 & \cdots & 1 \end{bmatrix} \in \mathbf{R}^{n+1 \times p}$.
    - ▷ **The key here is that since** $p > n+1$ **there exists a solution** $\eta \in \mathbf{R}^m \neq \mathbf{0}$ **to** $A\eta = \mathbf{0}$.
    - ▷ By the last row of A, $\eta_1 + \eta_2 + \cdots + \eta_m = 0$, thus η has both $+$ and - coordinates.
    - ▷ Let $\tau = \min\{\frac{\alpha_i}{\eta_i}, \eta_i > 0\} = \frac{\alpha_{i_0}}{\eta_{i_0}}$.
    - ▷ **Let** $\mathbf{m}\tilde{\alpha}_i = \alpha_i - \tau\eta_i$. Hence $\tilde{\alpha}_i \geq 0$ and $\tilde{\alpha}_{i_0} = 0$.
    - ▷ $\tilde{\alpha}_1 + \cdots + \tilde{\alpha}_p = (\alpha_1 + \cdots + \alpha_p) - \tau(\eta_1 + \cdots + \eta_p) = 1,$
      $\tilde{\alpha}_1 \mathbf{x}_1 + \cdots + \tilde{\alpha}_p \mathbf{x}_p = \alpha_1 \mathbf{x}_1 + \cdots + \alpha_p \mathbf{x}_p - \tau(\eta_1 \mathbf{x}_1 + \cdots + \eta_p \mathbf{x}_p) = \mathbf{x}.$
    - ▷ Thus $\mathbf{x} = \sum_{i \neq i_0} \alpha_i \mathbf{x}_i$ of $\mathbf{m}p - 1$ points $\{\mathbf{x}_i, i \neq i_0\}$.
    - ▷ Iterate this procedure until $\mathbf{x}$ is a convex combin. of $n+1$ points of S.

Basic Solutions, Extreme Points and Optima of Linear Programs

Terminology

- A linear program is **a mathematical program** with **linear objectives** and **linear constraints**.
- A linear program in **canonical** form is the program

$$\begin{array}{ll}\text{maximize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}.\end{array}$$

  $\mathbf{b} \geq 0 \Rightarrow$ **feasible** canonical form. Initial feasible point: $\mathbf{x} = \mathbf{0}$.

- In broad terms:
  - In resource allocation problems canonical is more adapted,
  - in flow problems standard is usually more natural.
- However our algorithms work in **standard** form.

Terminology

- A linear program in **standard** form is the program

$$\begin{aligned} \text{maximize} \quad & \mathbf{c}^{T}\mathbf{x} && (5)\\ \text{subject to} \quad & A\mathbf{x} = \mathbf{b}, && (6)\\ & \mathbf{x} \geq \mathbf{0}. && (7) \end{aligned}$$

- Easy to go from one to the other but dimensions of $\mathbf{x}, \mathbf{c}, A, \mathbf{b}$ may change.
- Ultimately, **all** LP can be written in **standard form**.

Terminology

Definition 13. *(i) A* ***feasible solution*** *to an LP in* **standard form** *is a vector* $\mathbf{x}$ *that satisfies constraints* (6)(7).

(ii) The set of all feasible solutions is called the ***feasible set*** *or* ***feasible region****.*

(iii) A feasible solution to an LP is an ***optimal solution*** *if it maximizes the objective function of the LP.*

(iv) A feasible solution to an LP in **standard form** *is said to be a* ***basic feasible solution (BFS)*** *if it is a basic solution with respect to Equation* (6).

(v) If a basic solution is non-degenerate, we call it a ***non-degenerate basic feasible solution****.*

- note that an optimal solution may not be unique, but the optimal value of the problem is.
- Anytime **"basic"** is quoted, we are implicitly using the **standard form**.

$\exists$ feasible solutions $\Rightarrow \exists$ *basic* feasible solutions

Theorem 8. *The* feasible region *to an LP is* ***convex, closed, bounded from below****.*

Theorem 9. *If there is a* feasible solution *to a* ***LP in standard form****, then there is a* ***basic feasible*** *solution.*

- **Proof idea**:
  - if $\mathbf{x}$ is such that $\sum_{i \in I} x_i \mathbf{a}_i = \mathbf{b}$ and where $\mathrm{card}(I) > m$ then we show we can have an expansion of $\mathbf{x}$ with a smaller family I'.
  - Eventually by making I smaller we turn it into a basis $\mathbf{I}$.
  - Some of the simplex's algorithm ideas are contained in the proof.
- **Remarks**:
  - Finding an **initial** feasible solution might be **a problem to solve by itself**.
  - We **assume** in the next slides **we have one**. More on this later.

Proof

Assume $\mathbf{x}$ is a solution with $p \leq n$ positive variables. Up to a reordering and for convenience, assume that such variables are the p first variables, hence $\mathbf{x} = (x_1, \cdots, x_p, 0, \cdots, 0)$ and $\sum_{i=1}^{p} x_i \mathbf{a}_i = \mathbf{b}$.

- if $\{\mathbf{a}_i\}_{i=1}^{p}$ is linearly independent, then necessarily $p \leq m$. If $p = m$ then the solution is *basic*. If $p < m$ it is *basic* and *degenerate*.
- Suppose $\{\mathbf{a}_i\}_{i=1}^{p}$ is *linearly dependent*.
  - Assume all $\mathbf{a}_i, i \leq p$ are non-zero. If there is a zero vector we can remove it from the start. Hence we have $\sum_{i=1}^{p} \alpha_i \mathbf{a}_i = \mathbf{0}$ with $\alpha \neq \mathbf{0}$.
  - If $\alpha_r \neq 0$, then $\mathbf{a}_r = \sum_{j=1, j\neq r}^{p} \left(-\frac{\alpha_j}{\alpha_r}\right) \mathbf{a}_j$, which, when substituted in $\mathbf{x}$'s expansion,
$$\sum_{j=1, j\neq r}^{p} \left(x_j - x_r \frac{\alpha_j}{\alpha_r}\right) \mathbf{a}_j = \mathbf{b},$$
with has now no more than $p-1$ **non-zero** variables.
  - **non-zero** is not enough, since we need **feasibility**.

Proof

- ◦ We need to choose r carefully such that

$$x_j - x_r \frac{\alpha_j}{\alpha_r} \geq 0, j = 1, 2, \cdots, p. \tag{8}$$

- ◦ For indexes j such that $\alpha_j = 0$ condition (8) is ok. For those $\alpha_j \neq 0$, (8) becomes

$$\frac{x_j}{\alpha_j} - \frac{x_r}{\alpha_r} \geq 0 \quad \text{for } \alpha_j > 0, \tag{9}$$

$$\frac{x_j}{\alpha_j} - \frac{x_r}{\alpha_r} \leq 0 \quad \text{for } \alpha_j < 0, \tag{10}$$

- ◦ Let's select r among the indexes $\{k \,|\alpha_k > 0\}$ is positive. (10) always holds, and we set $r = \text{argmin}_k \left\{\frac{x_k}{\alpha_k} | \alpha_k > 0\right\}$ for (9) to hold.

- **Finally**: when $p > m$, we can show that there exists a **feasible** solution which can be written as a combination of $p - 1$ vectors $\mathbf{a}_i \Rightarrow$ only need to reiterate.
- **Remark** we could have chosen r among $\{k \,|\alpha_k < 0\}$.(9) would always hold, and we need to choose $r = \text{argmin}_k \left\{\frac{x_k}{\alpha_k} | \alpha_k < 0\right\}$ for (10). **both cases** are valid. Of course, different choices will give different expansions.

Basic feasible solutions of an LP $\subset$ Extreme points of the feasible region

Theorem 10. *The* ***basic feasible*** *solutions of an LP in standard form are* ***extreme points*** *of the corresponding feasible region.*

- **Proof idea**: basic solutions means that $\mathbf{x_I}$ is uniquely defined by $B_\mathbf{I}$'s invertibility, that is $\mathbf{x_I}$ is uniquely defined as $B_\mathbf{I}^{-1}\mathbf{b}$. This helps to prove that $\mathbf{x}$ is extreme.

Proof

- Suppose $\mathbf{x}$ is a basic feasible solution, that is with proper reordering $\mathbf{x}$ has the form $\mathbf{x} = \left[\begin{smallmatrix}\mathbf{x}_B \\ \mathbf{0}\end{smallmatrix}\right]$ with $\mathbf{x}_B = B^{-1}\mathbf{b}$ and $B \in \mathbf{R}^{m\times m}$ an invertible matrix made of l.i. columns of A.
- Suppose $\exists \mathbf{x}_1, \mathbf{x}_2$ s.t. $\mathbf{x} = \frac{\mathbf{x}_1+\mathbf{x}_2}{2}$.
- Write $\mathbf{x}_1 = \left[\begin{smallmatrix}\mathbf{u}_1 \\ \mathbf{v}_1\end{smallmatrix}\right], \mathbf{x}_2 = \left[\begin{smallmatrix}\mathbf{u}_2 \\ \mathbf{v}_2\end{smallmatrix}\right]$
- since $\mathbf{v}_1, \mathbf{v}_2 \geq 0$ and $\frac{\mathbf{v}_1+\mathbf{v}_2}{2} = \mathbf{0}$ necessarily $\mathbf{v}_1 = \mathbf{v}_2 = \mathbf{0}$.
- Since $\mathbf{x}_1$ and $\mathbf{x}_2$ are feasible, $B\mathbf{u}_1 = \mathbf{b}$ and $B\mathbf{u}_2 = \mathbf{b}$ hence $\mathbf{u}_1 = \mathbf{u}_2 = B^{-1}\mathbf{b} = \mathbf{x}_B$ which proves that $\mathbf{x}_1 = \mathbf{x}_2 = \mathbf{x}$.

Basic feasible solutions of an LP $\supset$ Extreme points of the feasible region

Theorem 11. *The* ***extreme points*** *of the feasible region of an LP in standard form are* ***basic feasible*** *solutions of the LP.*

- **Proof idea**: Similar to the previous proof, the fact that a point is extreme helps show that it only has m or less non-zero components.

Proof

Let $\mathbf{x}$ be an extreme point of the feasible region of an LP, with $r \leq n$ zero variables. We reorder variables such that $x_i, i \leq r$ are positive and $x_i = 0$ for $r+1 \leq i \leq n$.

- As usual $\sum_{i=1}^{r} x_i \mathbf{a}_i = \mathbf{b}$.
- Let us prove by contradiction that $\{\mathbf{a}_i\}_{i=1}^{r}$ are linearly independent.
- if not, $\exists (\alpha_1, \cdots, \alpha_r) \neq \mathbf{0}$ such that $\sum_{i=1}^{r} \alpha_i \mathbf{a}_i = \mathbf{0}$. We show how to use the family α to create two distinct feasible points $\mathbf{x}_1$ and $\mathbf{x}_2$ such that $\mathbf{x}$ is their center.
- Let $0 < \varepsilon < \min_{\alpha_i \neq 0} \frac{x_i}{|\alpha_i|}$. Then $x_i \pm \varepsilon \alpha_i > 0$ for $i \leq r$ and set $\mathbf{x}_1 = \mathbf{x} + \varepsilon \alpha$ and $\mathbf{x}_2 = \mathbf{x} - \varepsilon \alpha$ with $\alpha = (\alpha_1, \cdots, \alpha_r, 0, \cdots, 0) \in \mathbf{R}^n$.
- $\mathbf{x}_1, \mathbf{x}_2$ are feasible: by definition of $\varepsilon, \mathbf{x}_1, \mathbf{x}_2 \geq 0$. Furthermore, $A\mathbf{x}_1 = A\mathbf{x}_2 = A\mathbf{x} \pm \varepsilon A\alpha = \mathbf{b}$ since $A\alpha = \mathbf{0}$
- We have $\frac{\mathbf{x}_1 + \mathbf{x}_2}{2} = \mathbf{x}$ which is a contradiction.

$\exists$ extreme point in the set of optimal solutions.

Theorem 12. *The* ***optimal*** *solution to an LP in standard form occurs at an* ***extreme point*** *of the feasible region.*

Proof. Suppose the optimal value of an LP is $z^\star$ and suppose the objective is to maximize $\mathbf{c}^T\mathbf{x}$.

- Any optimal solution $\mathbf{x}$ is necessarily in the boundary of the feasible region. If not, $\exists\varepsilon > 0$ such that $\mathbf{x}+\varepsilon\mathbf{c}$ is still feasible, and $\mathbf{c}^T(\mathbf{x}+\varepsilon\mathbf{c}) = z^\star + \varepsilon|c|^2 > z^\star$.
- The set of solutions is the intersection of $H_{\mathbf{c},z^\star}$ and the feasible region C which is *convex & bounded from below*. $H_{\mathbf{c},z^\star}$ is a supporting plane of C on the boundary point $\mathbf{x}$, thus $H_{\mathbf{c},z^\star}$ contains an extreme point (Thm. 3,lecture 3).

■

... but some solutions that are not extreme points might be optimal.

Wrap-up

(i) a feasible solution exists $\Rightarrow$ we know how to turn it into a **basic feasible** solution;

(ii) **basic feasible** solutions $\Leftrightarrow$ **extreme** points of the feasible region;

(iii) **Optimum** of an LP occurs at an **extreme** point of the feasible region;

That's it for basic convex analysis and LP's

Major Recap

- A Linear Program is a program with **linear constraints** and **objectives**.
- Equivalent formulations for LP's: **canonical** (inequalities) and **standard** (equalities) form.
- Both have feasible **convex** sets that are **bounded from below**.
- **Simplex Algorithm** to solve LP's only works in **standard form**.

- In **standard form**, the optimum occurs on an **extreme point** of the feasible set.
- All **extreme points** are **basic feasible solutions**.
- **basic feasible solutions** are of the type $\mathbf{x}_\mathbf{I} = B_\mathbf{I}^{-1}\mathbf{b}$ for a subset $\mathbf{I}$ of m coordinates in n, zero elsewhere.
- Looking for an optimum? **only need to check extreme points** $\Leftrightarrow$ **BFS**.
- Looking for an optimum? $\exists$ **a basis** $\mathbf{I}$ **which realizes that optimum**.

The essence of The Simplex Algorithm: Improving the Objective From a Basic Feasible Solution

Improving a BFS

- Remember that a **standard form** LP is

$$\begin{array}{rc} \text{maximize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}. \end{array}$$

- Given $\mathbf{I} = (i_1, \cdots, i_m)$, the base $B_\mathbf{I} = [\mathbf{a}_{i_1}, \mathbf{a}_{i_2}, \cdots, \mathbf{a}_{i_m}]$, suppose we have a **basic feasible solution** where $\mathbf{x}_\mathbf{I} = B^{-1}\mathbf{b}$, that is an **extreme** point of the feasible polyhedron.
- **We know that the optimum is reached on an optimal $\mathbf{I}^\star$.**
- There is finite number of families $\{\mathbf{I}|B_\mathbf{I}$ is invertible, $x_\mathbf{I}$ is feasible$\}$.
- How can we find a family $\mathbf{I}'$ such that $\mathbf{x}_{\mathbf{I}'}$ is still feasible and $\mathbf{c}_{\mathbf{I}'}^T\mathbf{x}_{\mathbf{I}'} > \mathbf{c}_\mathbf{I}^T\mathbf{x}_\mathbf{I}$?.
- The **simplex algorithm** provides an answer, where an index of $\mathbf{I}$ is replaced by a new integer in $\mathbf{O} = [1, \cdots, n] \setminus \mathbf{I}$.
- Note that we only have methods that change **one index at a time**.

The simplex does three things

Given a BFS $\mathbf{I}$

- shows how to select a **base** $\mathbf{I}'$ by changing one index in $\mathbf{I}$ (an index goes out, an index goes in)
- check how to select an **improved basic** solution by telling which index to include.
- check how we can select a **improved basic feasible** solution linked to $\mathbf{I}'$ by telling which index to remove.

In practice, given a BFS $\mathbf{I}$, the 3 steps of the simplex

1. Look for an index that would **improve** the objective.
2. check we can **improve** and obtain a valid **base** $\mathbf{I}'$ by incorporating that index and checking there is at least one we can remove.
3. **basic** & **improve** objective accomplished, ensure now $\mathbf{x}_{\mathbf{I}'}$ is **feasible** by choosing the index we remove.

Initial Setting

- Let $\mathbf{I} = (i_1, \cdots, i_m)$, the base $B_{\mathbf{I}} = [\mathbf{a}_{i_1}, \mathbf{a}_{i_2}, \cdots, \mathbf{a}_{i_m}]$ and suppose we have a **basic feasible solution** $\mathbf{x}_{\mathbf{I}} = B_{\mathbf{I}}^{-1}\mathbf{b}$.
- The **column vectors** of B are **l.i.**, and can thus be used as a basis of $\mathbf{R}^m$. Thus $\exists Y \in \mathbf{R}^{m\times n} \mid A = BY$, namely $Y = B^{-1}A$, the coordinates of all vectors of A in base B.

$$m\left\{ \overbrace{\begin{bmatrix} \vdots & \vdots & \cdots\cdots & \vdots \\ \mathbf{a}_1 & \mathbf{a}_2 & \cdots\cdots & \mathbf{a}_n \\ \vdots & \vdots & \cdots\cdots & \vdots \end{bmatrix}}^{n} \right. = \overbrace{\begin{bmatrix} \vdots & \vdots & \cdot\cdot & \vdots \\ \mathbf{a}_{i_1} & \mathbf{a}_{i_2} & \cdot\cdot & \mathbf{a}_{i_m} \\ \vdots & \vdots & \cdot\cdot & \vdots \end{bmatrix}}^{m} \overbrace{\begin{bmatrix} \vdots & \vdots & \cdots\cdots & \vdots \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots\cdots & \mathbf{y}_n \\ \vdots & \vdots & \cdots\cdots & \vdots \end{bmatrix}}^{n}$$

or individually $\mathbf{a}_j = \sum_{k=1}^{m} y_{k,j}\, \mathbf{a}_{i_k}$. We write $\mathbf{y}_j = \begin{bmatrix} y_{1,j} \\ \vdots \\ y_{m,j} \end{bmatrix}$ and $\mathbf{a}_j = B\mathbf{y}_j$.

- Hence $\mathbf{y}_j = B^{-1}\mathbf{a}_j$ and B^{-1} is a change of coordinate matrix from the canonical base to the base in B.

Change an element in the basis and still have a *basic* solution

- Change an index in $\mathbf{I}$? everything depends on

$$Y = \begin{bmatrix} \vdots & \vdots & \cdots\cdots & \vdots \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots\cdots & \mathbf{y}_n \\ \vdots & \vdots & \cdots\cdots & \vdots \end{bmatrix} \in \mathbf{R}^{m\times n}$$

- Claim: if $y_{r,e} \neq 0$ for two indices, $r \leq m$, $e \leq n$ and not in $\mathbf{I}$,
  - r **for remove,** e **for enter**,
  - one can substitute the $\boldsymbol{r}^{\text{th}}$ column of $B, \mathbf{a}_{i_r}$, for the $\boldsymbol{e}^{\text{th}}$ column of $A, \mathbf{a}_e$.
  - That is we can select the basis $\hat{\mathbf{I}} = (\mathbf{I} \setminus i_r) \cup e$ and we are sure that
    - ▷ $B_{\hat{\mathbf{I}}}$ is invertible,
    - ▷ $\mathbf{x}_{\hat{\mathbf{I}}}$ is a basic solution.

basic solution

- **Proof** if $y_{r,e} \neq 0$, $\mathbf{a}_e = \boldsymbol{y_{r,e}}\, \mathbf{a}_{i_r} + \sum_{k \neq r} y_{k,j}\, \mathbf{a}_{i_k} \Rightarrow \mathbf{a}_{i_r} = \frac{1}{\boldsymbol{y_{r,e}}} \mathbf{a}_e - \sum_{k \neq r} \frac{y_{k,j}}{\boldsymbol{y_{r,e}}} \mathbf{a}_{i_k}$.

  Thus
$$B_{\mathbf{I}} \mathbf{x}_{\mathbf{I}} = \sum_{k=1}^{m} x_{i_k} \mathbf{a}_{i_k} = \boldsymbol{x_{i_r}} \mathbf{a}_{i_r} + \sum_{k=1, k \neq r}^{m} x_{i_k} \mathbf{a}_{i_k} = \mathbf{b}$$
  is replaced by
$$\frac{\boldsymbol{x_{i_r}}}{\boldsymbol{y_{r,e}}} \mathbf{a}_e + \sum_{k=1}^{m} \left(x_{i_k} - \boldsymbol{x_{i_r}} \frac{y_{k,e}}{\boldsymbol{y_{r,e}}} \right) \mathbf{a}_{i_k} = \mathbf{b}$$
  and we have a new solution $\hat{\mathbf{x}}$ with $\hat{I} = (i_1, \cdots, i_{r-1}, e, i_{r+1}, \cdots, i_m)$ and
$$\begin{aligned} \hat{x}_{i_k} &= x_{i_k} - \boldsymbol{x_{i_r}} \frac{y_{k,e}}{\boldsymbol{y_{r,e}}} \quad \text{for } 1 \leq k \leq m, \ (k \neq r) \\ \hat{x}_e &= \frac{\boldsymbol{x_{i_r}}}{\boldsymbol{y_{r,e}}} \end{aligned}$$
  note that $\hat{x}_{i_r} = 0$ and we still have a **basic** solution.

basic & *better*: restriction on e

- The objective value, $\mathbf{c}_{\mathbf{I}}^T \mathbf{x}_{\mathbf{I}}$ becomes $\mathbf{c}_{\hat{I}}^T \hat{\mathbf{x}}_{\hat{I}}$ with $\hat{c}_{i_k} = c_{i_k}$ for $k \neq r$ and $\hat{c}_e = \boldsymbol{c_e}$. Thus

$$\begin{aligned}\hat{z} &= \mathbf{c}_{\hat{I}}^T \hat{\mathbf{x}}_{\hat{I}} = \textstyle\sum_{k\neq r} c_{i_k} \hat{x}_{i_k} + \boldsymbol{c_e} \hat{x}_e \\ &= \textstyle\sum_{k\neq r} c_{i_k} \left(x_{i_k} - \boldsymbol{x_{i_r}} \frac{y_{k,e}}{\boldsymbol{y_{r,e}}} \right) + \boldsymbol{c_e} \frac{\boldsymbol{x_{i_r}}}{\boldsymbol{y_{r,e}}} \\ &= \textstyle\sum_k c_{i_k} x_{i_k} - \frac{\boldsymbol{x_{i_r}}}{\boldsymbol{y_{r,e}}} \sum_k c_{i_k} y_{k,e} + \boldsymbol{c_e} \frac{\boldsymbol{x_{i_r}}}{\boldsymbol{y_{r,e}}} \\ &= z - \frac{\boldsymbol{x_{i_r}}}{\boldsymbol{y_{r,e}}} \mathbf{c}_{\mathbf{I}}^T \mathbf{y}_e + \boldsymbol{c_e} \frac{\boldsymbol{x_{i_r}}}{\boldsymbol{y_{r,e}}} \\ &= z + \frac{\boldsymbol{x_{i_r}}}{\boldsymbol{y_{r,e}}} (\boldsymbol{c_e} - z_e),\end{aligned}$$

  where $z_e = \mathbf{c}_{\mathbf{I}}^T \mathbf{y}_e = \mathbf{c}_{\mathbf{I}}^T B^{-1} \mathbf{a}_e$.

- $\hat{z} > z$ if $y_{r,e} > 0$ and $\boldsymbol{c_e} - \boldsymbol{z_e} > 0$, hence we choose a column e such that
  - $\boldsymbol{c_e} - \boldsymbol{z_e} > 0$
  - there exists $y_{i,e} > 0$

- **Important Remark** if $\mathbf{x}_{\mathbf{I}}$ is **non-degenerate**, $x_{i_r} > 0$ and hence $\hat{z} > z$.

- Much better than $\hat{z} \geq z$ as it implies convergence.

basic & better & *feasible*: restriction on r

- We require $\hat{x}_i \geq 0$ for all i. In particular, for basic variables we need that

$$\begin{cases} \hat{x}_{i_k} = x_{i_k} - \boldsymbol{x_{i_r}} \frac{y_{k,e}}{\boldsymbol{y_{r,e}}} \geq 0 & \text{for } 1 \leq k \leq m \ (k \neq r) \\ \hat{x}_e = \frac{\boldsymbol{x_{i_r}}}{\boldsymbol{y_{r,e}}} \geq 0 \end{cases}$$

- Let r be chosen such that

$$\frac{x_{i_r}}{y_{r,e}} = \min_{k=1,..,m} \left\{ \frac{x_{i_k}}{y_{k,e}} \middle| \ y_{k,e} > 0 \right\}$$

From one basic feasible solution to a better one

Theorem 13. *Let* $\mathbf{x}$ *be a* ***basic feasible solution (BFS)*** *to a LP with index set* $\mathbf{I}$ *and objective value* z*. If there exists* $e \notin \mathbf{I}, 1 \leq e \leq n$ *such that*

(i) *a* ***reduced cost coefficient*** $c_e - z_e > 0$,

(ii) *at least* ***one coordinate of*** $\mathbf{y}_e$ ***is positive***, $\exists i$ *such that* $y_{i,e} > 0$,

then it is possible to obtain ***a new BFS*** *by replacing an index in* $\mathbf{I}$ *by* e*, and the new value of the objective value* $\hat{z}$ *is such that* $\hat{z} \geq z$, ***strictly*** *if* $x_{\mathbf{I}}$ *is non-degenerate.*

From one basic feasible solution to a better one

- **Remark**: coefficients $\boldsymbol{c_e} - \boldsymbol{z_e}$ are called reduced cost coefficients.
- **Remark** "$e \notin \mathbf{I}$" is redundant: if $e \in \mathbf{I}$, that is $\exists k, i_k = e$ then $\boldsymbol{c_e} - \boldsymbol{z_e} = 0$. Indeed, $\boldsymbol{c_e} - \boldsymbol{z_e} = c_e - \mathbf{c}_{\mathbf{I}}^T B^{-1} \mathbf{a}_e = c_e - \mathbf{c}_{\mathbf{I}}^T \mathbf{e}_{i_k} = c_e - c_e = 0$ where $\mathbf{e}_i$ is the i^{th} canonical vector of $\mathbf{R}^m$. Indeed, if $B\mathbf{x} = \mathbf{a}$ and $\mathbf{a}$ is the k^{th} vector of B then necessarily $\mathbf{x} = \mathbf{e}_k$.
- **Remember**: if $k \in \mathbf{I}$ then **necessarily the reduced cost** $(c_k - z_k)$ **is 0**.

Testing for Optimality

Optimality: $c_i - z_i \leq 0$ for all i

Theorem 14. *Let* $\mathbf{x}^\star$ *be a* ***basic feasible solution (BFS)*** *to a LP with index set* $\mathbf{I}^\star$ *and objective value* $z^\star$*. If* $c_i - z_i^\star \leq 0$ *for all* $1 \leq i \leq n$ *then* $\mathbf{x}^\star$ *is optimal.*

- **Proof idea**: the conditions $c_i - z_i^\star \leq 0$ allow us to write that $\sum c_i x_i$ is smaller than $\sum z_i^\star x_i$ for all $\mathbf{x}$ in $\mathbf{R}_+^m$. Moreover, $z_i^\star$ integrates information about the base $\mathbf{I}^\star$ and we show that the point that realizes $\sum z_i^\star x_i = \mathbf{c}^T\mathbf{x}$ is necessarily $\mathbf{x}^\star$ and thus every $\mathbf{c}^T\mathbf{x}$ is smaller than $\mathbf{c}^T\mathbf{x}^\star$.

Proof

- For any *feasible solution* $\mathbf{x}$ we have $\sum_{k=1}^n c_k x_k \leq \sum_{k=1}^n \boldsymbol{z_k^\star} x_k$. Yet,

$$\sum_{k=1}^n \boldsymbol{z_k^\star} x_k = \sum_{k=1}^n \mathbf{c}_{\mathbf{I}^\star}^T \mathbf{y}_{\boldsymbol{k}}\, x_k = \sum_{k=1}^n \left(\sum_{j=1}^m \boldsymbol{c_{i_j} y_{j,k}} \right) x_k = \sum_{j=1}^m \boldsymbol{c_{i_j}} \left(\sum_{k=1}^n \boldsymbol{y_{j,k}} x_k \right)$$

- We have found a maxima of $\mathbf{c}^T\mathbf{x}$ with base $\mathbf{I}^\star$...
- The terms $u_j \stackrel{\text{def}}{=} \sum_{k=1}^n \boldsymbol{y_{j,k}}\, x_k$ **are actually equal to** $x_{i_j}^\star$. Indeed, remember $\sum_{j=1}^m \boldsymbol{x_{i_j}^\star \mathbf{a}_{i_j}} = \mathbf{b}$ and that since $\mathbf{x}$ is feasible, $\sum_{k=1}^n x_k \mathbf{a}_k = \mathbf{b}$. Yet,

$$\sum_{k=1}^n x_k (\boldsymbol{B_{\mathbf{I}^\star} \mathbf{y}_k}) = \sum_{k=1}^n \left(\sum_{j=1}^m \boldsymbol{y_{k,j} \mathbf{a}_{i_j}} \right) x_k = \sum_{j=1}^m \left(\sum_{k=1}^n \boldsymbol{y_{k,j}} x_k \right) \boldsymbol{\mathbf{a}_{i_j}} = \sum_{j=1}^m u_j \boldsymbol{\mathbf{a}_{i_j}} = \mathbf{b}.$$

Hence

$$z \leq \sum_{j=1}^m \boldsymbol{c_{i_j}\, x_{i_j}^\star} = z^\star.$$

Testing for Boundedness

(un)boundedness

- Sometimes programs are trivially unbounded

$$\begin{array}{ll} \text{maximize} & \mathbf{1}^T \mathbf{x} \\ \text{subject to} & \mathbf{x} \geq \mathbf{0}. \end{array}$$

- Here **both** the feasible set and the objective on that feasible set are **unbounded**.
- Feasible set is **bounded** $\Rightarrow$ objective is bounded.
- Feasible set is **unbounded**, optimum might be bounded **or** unbounded, no implication.
- Two different issues.
- Can we check quickly?

(un)boundedness of the feasible set *and/or* of the objective.

Theorem 15. *Consider an LP in* **standard** *form and a basic feasible index set* $\mathbf{I}$*. If there exists an index* $e \notin \mathbf{I}$ *such that* $\mathbf{y}_e \leq 0$ *then the* ***feasible region*** *is* ***unbounded****. If moreover for* e *the reduced cost* $c_e - z_e > 0$ *then there exists a feasible solution with at most* $\mathbf{m+1}$ *nonzero variables and an* ***arbitrary large objective function****.*

Proof sketch:

- Take advantage of $\mathbf{y}_e \leq 0$ to modify a BFS $\mathbf{b} = \sum x_{i_j}\mathbf{a}_{i_j}$ to get a new **nonbasic** feasible solution using $\mathbf{a}_e$, $\mathbf{b} = \sum x_{i_j}\mathbf{a}_{i_j} - \theta\mathbf{a}_e + \theta\mathbf{a}_e$. This solution is arbitrarily large.
- If for that e, $c_e > z_e$ then it is easy to prove that we can have an arbitrarily high objective.

(un)boundedness of the feasible set *and/or* of the objective.

Proof. • Let $\mathbf{I}$ be an index set and $\mathbf{x_I}$ the corresponding BFS.

- Remember that for any index, e in particular, $\mathbf{a}_e = B_\mathbf{I}\mathbf{y}_e = \sum_{j=1}^m y_{j,e}\mathbf{a}_{i_j}$.
- Let's play with $\mathbf{a}_e$: $\mathbf{b} = \sum_{j=1}^m x_{i_j}\mathbf{a}_{i_j} - \theta\mathbf{a}_e + \theta\mathbf{a}_e$.
- $\mathbf{b} = \sum_{j=1}^m \left(x_{i_j} - \theta y_{j,e}\right)\mathbf{a}_{i_j} + \theta\mathbf{a}_e$
- Since $y_{j,e} \leq 0$ is negative we have a **nonbasic** & **feasible** solution with $m+1$ nonzero variables.
- θ can be set arbitrarily large: $\mathbf{x_I} + \theta\mathbf{a}_e$ is feasible $\Rightarrow$ **unboundedness**.
- If moreover $c_e > z_e$ then writing $\hat{z}$ for the objective of the point above,

$$\begin{aligned}
\hat{z} &= \sum_{j=1}^m (x_{i_j} - \theta y_{j,e})c_{i_j} + \theta c_e,\\
&= \sum_{j=1}^m x_{i_j}c_{i_j} - \theta\sum_{j=1}^m y_{j,e}c_{i_j} + \theta c_e,\\
&= c_\mathbf{I}^T\mathbf{x_I} - \theta c_\mathbf{I}^T\mathbf{y}_e + \theta c_e = z - \theta z_e + \theta c_e,\\
&= z + \theta(c_e - z_e).
\end{aligned}$$

■

A simple example

An example

- Let's consider the following example:

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 2 \\ 5 \\ 6 \\ 8 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}.$$

- Let us choose the starting $\mathbf{I}$ as $(1, 4)$. $B_{\mathbf{I}} = \begin{bmatrix} 1 & 4 \\ 1 & 1 \end{bmatrix}$, and we check easily that $\mathbf{x}_{\mathbf{I}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ which is feasible (lucky here) with objective

$$z = c_{\mathbf{I}}^T x_{\mathbf{I}} = \begin{bmatrix} 2 & 8 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 10.$$

An example: 4 out, 2 in

- Here $B_{\mathbf{I}}^{-1} = \frac{1}{3}\begin{bmatrix} -1 & 4 \\ 1 & -1 \end{bmatrix}$ the y_{ij} are given by $B_{\mathbf{I}}^{-1}A = \begin{bmatrix} 1 & -\frac{2}{3} & -1 & 0 \\ 0 & \frac{2}{3} & 1 & 1 \end{bmatrix}$, namely
$$\mathbf{y}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{y}_2 = \begin{bmatrix} -\frac{2}{3} \\ \frac{2}{3} \end{bmatrix}, \mathbf{y}_3 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \mathbf{y}_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
- Hence, $z_2 = \begin{bmatrix} 2 & 8 \end{bmatrix}\begin{bmatrix} -\frac{2}{3} \\ \frac{2}{3} \end{bmatrix} = 4, z_3 = \begin{bmatrix} 2 & 8 \end{bmatrix}\begin{bmatrix} -1 \\ 1 \end{bmatrix} = 6.$
- Because $\mathbf{I} = [1, 4]$, we know $z_1 - c_1 = z_4 - c_4 = 0.$
- We have $c_2 - z_2 = \mathbf{1}$; $c_3 - z_3 = 0$ so only one choice for e, that is 2.
- We check $\mathbf{y}_2$ and see that y_{22} is the only positive entry. Hence we remove the second index of $\mathbf{I}$, $i_2 = 4$. $\mathbf{I}' = (1, 2)$ and $B_{\mathbf{I}'} = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix}$
- The corresponding basic solution is $\mathbf{x}_{\mathbf{I}'} = \begin{bmatrix} 2 \\ \frac{3}{2} \end{bmatrix}$, **feasible as expected**.
- The objective is now $z' = \begin{bmatrix} 2 & 5 \end{bmatrix}\begin{bmatrix} 2 \\ \frac{3}{2} \end{bmatrix} = 11.5 > z$, **better, as expected**.

An example: that's it

- Since $B_{\mathbf{I}'}^{-1} = \frac{1}{2}\left[\begin{smallmatrix} 0 & 2 \\ 1 & -1 \end{smallmatrix}\right]$ the new coefficients y'_{ij} in

$$B_{\mathbf{I}'}^{-1}A == \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & \frac{3}{2} & \frac{3}{2} \end{bmatrix}$$

  are given by

$$\mathbf{y}'_1 = \left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right],\ \mathbf{y}'_2 = \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right],\ \mathbf{y}'_3 = \left[\begin{smallmatrix} 0 \\ 3/2 \end{smallmatrix}\right],\ \mathbf{y}'_4 = \left[\begin{smallmatrix} 1 \\ 3/2 \end{smallmatrix}\right],$$

- Now $c_3 - z_3 = 6 - \left[\begin{smallmatrix} 2 & 5 \end{smallmatrix}\right]\begin{bmatrix} 0 \\ \frac{3}{2} \end{bmatrix} = \mathbf{-1.5}$ and $c_4 - z_4 = 8 - \left[\begin{smallmatrix} 2 & 5 \end{smallmatrix}\right]\begin{bmatrix} 1 \\ \frac{3}{2} \end{bmatrix} = \mathbf{-1.5}$.
- since all $c_j - z_j \geq 0$, the set of indices $1, 2$ is optimal.
- The solution is $\mathbf{x}^\star = \begin{bmatrix} 2 \\ \frac{3}{2} \\ 0 \\ 0 \end{bmatrix}$.

Nice algorithm but...

Issues with the previous example

- Clean mathematically, but **very heavy notation-wise**.
- **Worse**: lots of **redundant** computations: we only change one column from $B_{\mathbf{I}}$ to $B_{\mathbf{I}'}$ but always recompute at each iteration:
  - the inverse $B_{\mathbf{I}}^{-1}$,
  - the $\mathbf{y}_i$'s, that is the matrix $Y = B_{\mathbf{I}}^{-1}A$,
  - the z_i's which can be found through $c_{\mathbf{I}}^T Y = c_{\mathbf{I}}^T B_{\mathbf{I}}^{-1} A$ and the reduced costs.
- **Plus** we *assumed* we had an initial feasible solution immediately... what if?
- Imagine someone solves the problem $(\mathbf{c}, A, \mathbf{b})$ before us and finds $\mathbf{x}^\star$ as the optimal solution such that $\mathbf{c}^T\mathbf{x}^\star = z^\star$.
- He gives it back to us adding the constraint $\mathbf{c}^T\mathbf{x} \geq z^\star$. Finding an initial feasible solution is as hard as finding **the optimal solution** itself!

A simpler formulation?

- For all these reasons, we look for a
  - compact (less redundant variables and notations),
  - fast computationally (rank one updates),

  methodology: the tableaux and dictionaries methods to go through the simplex step by step.

- We also study how to find an **initial** BFS and address additional issues.

- **YET** The simplex is **not** just a dictionary or a tableau method.

- The latter are **tools**. The **simplex algorithm is 100% algebraic and combinatorial**.

- The truth is that it is just an "optimization tool in disguise".

The simplex algorithm

Back to Basics: Basic Feasible Solutions, Extreme points, Optima

Three fundamental theorems:

- Let $\mathbf{x}$ be a **basic feasible solution (BFS)** to a LP with index set I and objective value z. If $\exists e, 1 \leq e \leq n, \boldsymbol{e \notin I}$ such that $c_e - \boldsymbol{z_e} > 0$ **and** at least one $\boldsymbol{y_{i,e} > 0}$, then we can have a **better basic feasible** solution by replacing an index in I by e with a new objective $\hat{z} \geq z$, **strictly** if x_I is non-degenerate.

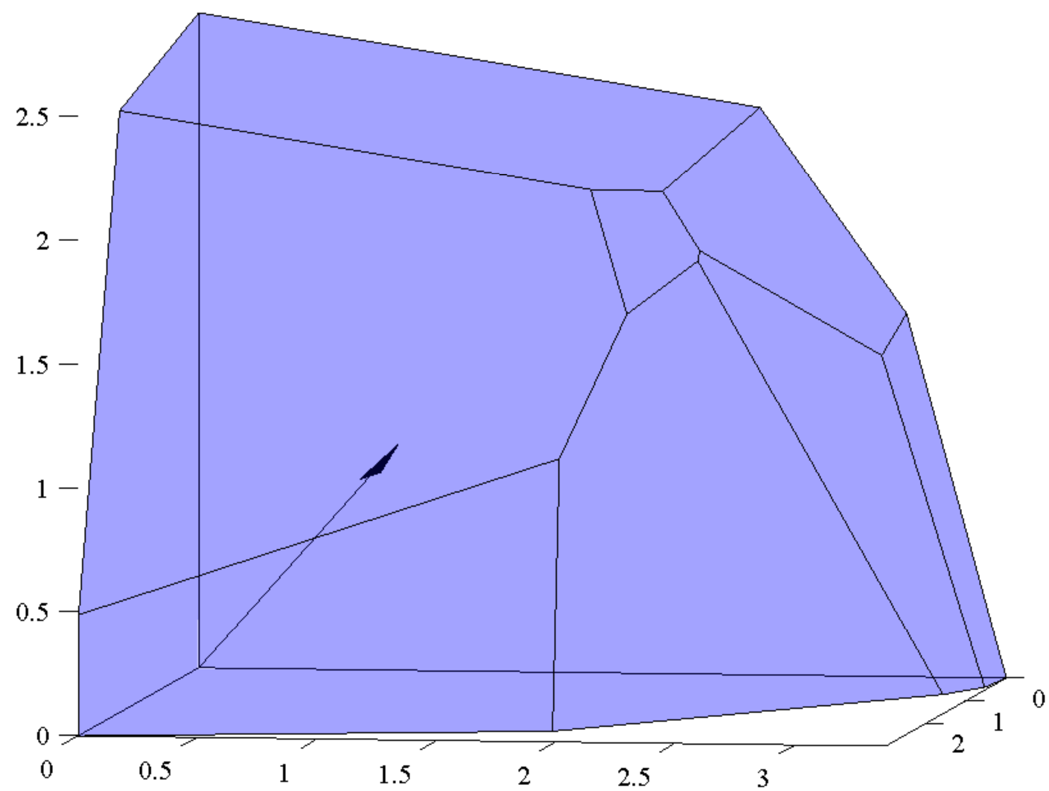
- Let $\mathbf{x}^\star$ be a **basic feasible solution (BFS)** to a LP with index set I and objective value $\boldsymbol{z}^\star$. If $c_i - \boldsymbol{z_i^\star} \leq 0$ for all $1 \leq i \leq n$ then $\mathbf{x}^\star$ is optimal.

- Let $\mathbf{x}$ be a **basic feasible solution (BFS)** to a LP with index set I. If $\exists$ an index $e \notin \mathbf{I}$ such that $\mathbf{y}_e \leq 0$ then the **feasible region** is **unbounded**. If moreover for e the reduced cost $c_e - z_e > 0$ then there exists a feasible solution with at most $\mathbf{m+1}$ nonzero variables and an **arbitrary large objective function**.

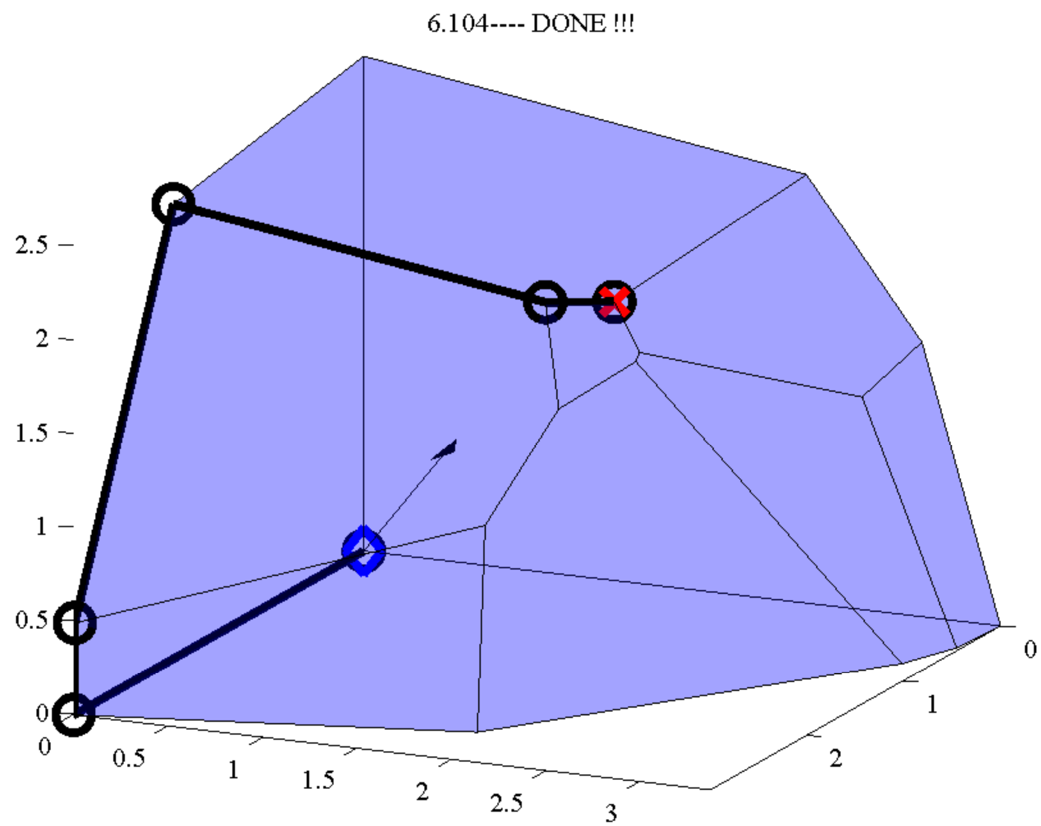
So far, what is the simplex?

- The simplex is a family of algorithms which do the following:
  1. Starts from an **initial** Basic feasible solution. **more on that later**.
  2. iterates: move from one BFS $\mathbf{I}$ to a **better** BFS $\mathbf{I}'$:
     - check reduced cost coefficients $c_j - c_{\mathbf{I}}^T B_{\mathbf{I}}^{-1}\mathbf{a}_j$, $j \in \mathbf{O}$. **if all negative $\mathbf{I}$ is *optimal*, OVER**.
     - otherwise, pick **one** index $\boldsymbol{e}$ for which it is positive. this will enter $\mathbf{I}$.
     - Check coordinates of $\mathbf{y}_{\boldsymbol{e}} = B_{\mathbf{I}}^{-1}\mathbf{a}_e$. **if all ≥ 0 then optimum is *unbounded*, OVER**.
     - otherwise, take the index $\boldsymbol{r}$ such that it achieves the minimum in $\{\frac{x_{i_j}}{y_{j,\boldsymbol{e}}} | y_{j,\boldsymbol{e}} > 0, 1 \leq j \leq m\}$, this will ensure feasibility. The $\boldsymbol{r}$th index of the base $\mathbf{I}$ is $i_r \leq n$.
     - $\mathbf{I}' = \{\mathbf{I} \setminus \boldsymbol{i_r}\} \cup \boldsymbol{e}$.
     - We have improved on the objective. If $x_{\mathbf{I}}$ was **not** degenerate, we have **strictly** improved.
     - $\mathbf{I} \leftarrow \mathbf{I}'$
- The loop is on a finite set of extreme points. it either exits early (unbounded), exits giving an answer (optimum $\mathbf{I}^\star$ and corresponding solution $\mathbf{x}^\star$) or loops indefinitely (degeneracy).

A Matlab Demo With Polyhedrons Containing the Origin



A Matlab Demo With Polyhedrons Containing the Origin



now with the real matlab demo...

A very important slide... WHY tableaux ?

- Last time: an example where we move from a base $\mathbf{I}$ to a new base $\mathbf{I}'$, compute $B_{\mathbf{I}'}^{-1}$, do the multiplications etc.. and reach the optimum. This is the simplex.
- **Double issue**:
  - **Computational 1**: inverting matrices costs time & money. One column is different between $B_{\mathbf{I}}$ and $B_{\mathbf{I}'}$, can we do better than inverting everything again?
  - **Computational 2**: multiplying matrices costs time & money. $B_{\mathbf{I}}^{-1}A$ and $B_{\mathbf{I}'}^{-1}A$ are related.
- **Down to what we really need at each iteration**:
  - reduced cost coefficients vector $(c_i - z_i)$ of $\mathbf{R}^n$ to pick an index $\boldsymbol{e}$ and check optimality,
  - All column vectors of A in the base $\mathbf{I}$, that is Y, to check boundedness and choose $\boldsymbol{r}$, namely all coordinates of $\mathbf{y}_{\boldsymbol{e}} = B_{\mathbf{I}}^{-1}a_{\boldsymbol{e}}$ in particular.
  - The current basic solution vector, $B_{\mathbf{I}}^{-1}\mathbf{b}$ both to choose $\boldsymbol{r}$ and on exit.
- Tableaux and Dictionaries only keep track of the last elements efficiently.

Simplex Method with Canonical Feasible Form

Canonical Feasible Form: We know an initial BFS to corresponding Standard Form

- let's **standardize** a **feasible** $(i.e. \mathbf{b} \geq 0)$ canonical form:

$$\begin{array}{ll} \text{maximize} & \alpha^T \mathbf{y} \\ \text{subject to} & \left\{ \begin{array}{ccc} M\mathbf{x} & \leq & \mathbf{b} \\ \mathbf{y} & \geq & 0 \end{array} \right. \end{array}$$

- We assume that $\mathbf{y}, \alpha \in \mathbf{R}^d$ for a d dimensional objective and $M \in \mathbf{R}^{m \times d}$ and $\mathbf{b} \in \mathbf{R}^m$ for m constraints.

Canonical Feasible Form: We know an initial BFS to corresponding Standard Form

- Slack variables $x_{d+1}, \cdots, x_{d+m}$ can be added so that $[A, I_m] \begin{bmatrix} \mathbf{y} \\ x_{d+1} \\ \vdots \\ x_{d+m} \end{bmatrix} = \mathbf{b}$ and the problem is now with $\mathbf{c} = [\alpha, \underbrace{0, \cdots, 0}_{m}] \in \mathbf{R}^{d+m}$

$$\begin{array}{ll} \text{maximize} & x_0 = \mathbf{c}^T \mathbf{x} \\ \text{subject to} & \left\{ \begin{array}{ccc} [M, I_m]\mathbf{x} & = & \mathbf{b} \\ \mathbf{x} & \geq & 0 \end{array} \right. \end{array}$$

- $\mathbf{x}, \mathbf{c} \in \mathbf{R}^{m+d}$, $\mathbf{c} = \left[\begin{smallmatrix} \alpha \\ \mathbf{0} \end{smallmatrix} \right]$, $A = [M, I_m] \in \mathbf{R}^{m \times (m+d)}$ and same $\mathbf{b} \in \mathbf{R}^m$.
- The dimensionality of the problem is now $n = d + m$.

Simplex Method: Tableau

Let us represent this by an (annotated) tableau:

	$\mathbf{O}$						$\mathbf{I}$						
	x_1	x_2	$\cdots$	x_e	$\cdots$	x_d	x_{d+1}	x_{d+2}	$\cdots$	x_{d+r}	$\cdots$	x_{d+m}	$\mathbf{b}$
x_{d+1}	m_{11}	m_{12}	$\cdots$	m_{1e}	$\cdots$	m_{1d}	**1**	0	$\cdots$	0	$\cdots$	0	b_1
x_{d+2}	m_{21}	m_{22}	$\cdots$	m_{2e}	$\cdots$	m_{2d}	0	**1**	$\cdots$	0	$\cdots$	0	b_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x_{d+r}	m_{r1}	m_{r2}	$\cdots$	m_{re}	$\cdots$	m_{rd}	0	0	$\cdots$	**1**	$\cdots$	0	b_r
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathbf{e}_{d+m}$	m_{m1}	m_{m2}	$\cdots$	m_{me}	$\cdots$	m_{md}	0	0	$\cdots$	0	$\cdots$	**1**	b_m
x_0	c_1	c_2	$\cdots$	c_e	$\cdots$	c_d	0	0	$\cdots$	0	$\cdots$	0	0

- Since $\mathbf{b} \geq 0$, take an original BFS as $\left[\underbrace{0, \cdots, 0}_{d}, b_1, b_2, \cdots, b_m\right]^T$
- Why:
  - **basic**: $I = \{d+1, \ldots, d+m\}$
  - **feasible**: $[0, \cdots, 0, b_1, b_2, \cdots, b_m]^T \geq 0$.

Simplex Method: Tableau

- the structure of the tableau so far,

A	$\mathbf{b}$
$(\mathbf{c}-\mathbf{z})'$	$\mathbf{0}$

- The index set $\mathbf{I}$ so far $\{d+1, d+2, \cdots, d+m\}$.
- $B_{\mathbf{I}} = I_m$, $B_{\mathbf{I}}^{-1}\mathbf{b} = \mathbf{b}$, $B^{-1}A = A$ *etc.*.

Simplex Method without non-negativity and objectives...

- Remember: a basis $\mathbf{I}$ gives a sparse solution $\mathbf{x_I}$.
- **there's one basis** $\mathbf{I}^\star$ which is the good one.
- The solution is $\mathbf{x}$ such that $\mathbf{x}_\mathbf{I}^\star = B_{\mathbf{I}^\star}^{-1}\mathbf{b}$ and the rest is zero.
- We can start with the **slack variables** as a basis in canonical **feasible** form.
- Under this form, the first matrix basis is $B = I$ the identity matrix.
- We will **move** from one basis to the other. We've proved this is possible.
- In doing so, we also have to recast the cost.
- Let's check how it looks in practice, without looking at feasibility and objective related concepts.

...the Gauss pivot...

- Consider now taking a variable out of $\mathbf{I}$ to replace it by a variable in $\mathbf{O}$.
- r initially in $\mathbf{I}$ leaves the basis, e initially in $\mathbf{O}$ is removed.
- all terms expressed so far in x_r need to be removed from all but one equation, and x_e enters instead.

...the Gauss pivot

- This is achieved through a pivot in the tableau.
- Once the indexes r and e are agreed upon, the rules to update the tableau are:

(a) in pivot row $a_{rj} \leftarrow a_{rj}/a_{re}$.
(b) in pivot column $a_{re} \leftarrow 1, a_{ie} = 0$ for $i = 1, \cdots, m, i \neq r$: the eth column becomes a matrix of zeros and a one.
(c) for all other elements $a_{ij} \leftarrow a_{ij} - \frac{a_{rj}a_{ie}}{a_{re}}$

The Gauss pivot

- Graphically,

$$\begin{array}{c} \\ \vdots \\ i \\ \vdots \\ r \\ \vdots \end{array}\begin{array}{c} \begin{array}{ccccc} \cdots & j & \cdots & e & \cdots \end{array} \\ \begin{bmatrix} \ddots & \vdots & \ddots & \vdots & \ddots \\ \cdots & a_{ij} & \cdots & \boldsymbol{a_{ie}} & \cdots \\ \ddots & \vdots & \ddots & \vdots & \ddots \\ \cdots & \boldsymbol{a_{rj}} & \cdots & \boldsymbol{a_{re}} & \cdots \\ \ddots & \vdots & \ddots & \vdots & \ddots \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ \vdots \\ i \\ \vdots \\ r \\ \vdots \end{array}\begin{array}{c} \begin{array}{ccccc} \cdots & j & \cdots & e & \cdots \end{array} \\ \begin{bmatrix} \ddots & \vdots & \ddots & \vdots & \ddots \\ \cdots & a_{ij} - \frac{\boldsymbol{a_{rj}}\boldsymbol{a_{ie}}}{\boldsymbol{a_{re}}} & \cdots & 0 & \cdots \\ \ddots & \vdots & \ddots & \vdots & \ddots \\ \cdots & \boldsymbol{a_{rj}}/\boldsymbol{a_{re}} & \cdots & \mathbf{1} & \cdots \\ \ddots & \vdots & \ddots & \vdots & \ddots \end{bmatrix} \end{array}$$

- Look at how the column e is now a column of 0 and 1's. This makes sense since $B^{-1}\mathbf{a}_e = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ with 1 in eth position means $\mathbf{a}_e$ is in the basis.

Linear system and pivoting

- Consider the linear system

$$\left\{\begin{array}{rcrcrcrcrcrcl} x_1 & + & x_2 & - & x_3 & + & x_4 & & & & & = & 5 \\ 2x_1 & - & 3x_2 & + & x_3 & & & + & x_5 & & & = & 3 \\ -x_1 & + & 2x_2 & - & x_3 & & & & & + & x_6 & = & 1 \end{array}\right.$$

- The corresponding tableau

$$\begin{array}{ccccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{a}_6 & \mathbf{b} \end{array}$$
$$\left[\begin{array}{ccccccc} 1 & 1 & -1 & 1 & 0 & 0 & 5 \\ 2 & -3 & 1 & 0 & 1 & 0 & 3 \\ -1 & 2 & -1 & 0 & 0 & 1 & 1 \end{array}\right]$$

Simplex Method: Swapping Indexes

- in the corresponding tableau,

$$\begin{array}{c} \\ \mathbf{a}_4 \\ \mathbf{a}_5 \\ \mathbf{a}_6 \end{array} \begin{array}{ccccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{a}_6 & \mathbf{b} \\ 1 & 1 & -1 & 1 & 0 & 0 & 5 \\ 2 & -3 & 1 & 0 & 1 & 0 & 3 \\ -1 & 2 & -1 & 0 & 0 & 1 & 1 \end{array}$$

notice the structure:

$$\boxed{\begin{array}{ccccccc} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & M & \vdots & \vdots & I_3 & \vdots & \mathbf{b} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \end{array}}$$

- And the fact that by taking the obvious basis $\mathbf{I} = \{4, 5, 6\}$ we have $B_\mathbf{I} = I_3$ and $B_\mathbf{I}^{-1} = I_3$

Simplex Method: Let's pivot

- Let's pivot arbitrarily. We put **1** in the base and remove **4**.

$$\begin{array}{c} \\ \mathbf{a}_4 \\ \mathbf{a}_5 \\ \mathbf{a}_6 \end{array} \begin{array}{c} \begin{array}{ccccccc} \boldsymbol{x}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{a}_6 & \mathbf{b} \end{array} \\ \left[\begin{array}{ccccccc} 1 & 1 & -1 & 1 & 0 & 0 & 5 \\ 2 & -3 & 1 & 0 & 1 & 0 & 3 \\ -1 & 2 & -1 & 0 & 0 & 1 & 1 \end{array}\right] \end{array}$$

which yields

$$\begin{array}{c} \\ \mathbf{a}_1 \\ \mathbf{a}_5 \\ \mathbf{a}_6 \end{array} \begin{array}{c} \begin{array}{ccccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{a}_6 & \mathbf{b} \end{array} \\ \left[\begin{array}{ccccccc} \mathbf{1} & 1 & -1 & 1 & 0 & 0 & 5 \\ \mathbf{0} & -5 & 3 & -2 & 1 & 0 & -7 \\ \mathbf{0} & 3 & -2 & 1 & 0 & 1 & 6 \end{array}\right] \end{array}$$

- $\mathbf{I} = \{1, 5, 6\}$, that is $B_\mathbf{I} = \left[\begin{smallmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{smallmatrix}\right]$. The basic solution is such that $\mathbf{x_I} = B_\mathbf{I}^{-1}\mathbf{b}$
- Note that all coordinates of $\mathbf{a}_1, \cdots, \mathbf{a}_6, \mathbf{b}$ in the table are given with respect to $\mathbf{a}_1, \mathbf{a}_5, \mathbf{a}_6$. In particular the last column corresponds to $B_\mathbf{I}^{-1}\mathbf{b}$...not feasible here BTW.

Simplex Method: again...

- Let's pivot arbitrarily again, this time inserting **2** and removing the **second** variable of the basis, **5**.

$$\begin{array}{c} \\ \mathbf{a}_1 \\ \mathbf{a}_5 \\ \mathbf{a}_6 \end{array} \begin{array}{c} \begin{array}{ccccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{a}_6 & \mathbf{b} \end{array} \\ \left[\begin{array}{ccccccc} 1 & 1 & -1 & 1 & 0 & 0 & 5 \\ 0 & \mathbf{-5} & 3 & -2 & 1 & 0 & -7 \\ 0 & 3 & -2 & 1 & 0 & 1 & 6 \end{array}\right] \end{array}$$

$$\begin{array}{c} \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_6 \end{array} \begin{array}{c} \begin{array}{ccccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{a}_6 & \mathbf{b} \end{array} \\ \left[\begin{array}{ccccccc} 1 & \mathbf{0} & -\frac{2}{5} & \frac{3}{5} & \frac{1}{5} & 0 & \frac{18}{5} \\ 0 & \mathbf{1} & -\frac{3}{5} & \frac{2}{5} & -\frac{1}{5} & 0 & \frac{7}{5} \\ 0 & \mathbf{0} & -\frac{1}{5} & -\frac{1}{5} & \frac{3}{5} & 1 & \frac{9}{5} \end{array}\right] \end{array}$$

- Notice how one can keep track of who is in the basis by checking where 0/1's columns are.

- The solution is now feasible... pure luck.

Simplex Method: and again...

- once again, pivot inserting **3** and removing the **third** variable of the basis, **6**.

$$\begin{array}{c} \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_6 \end{array} \begin{array}{c} \begin{array}{ccccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{a}_6 & \mathbf{b} \end{array} \\ \left[\begin{array}{ccccccc} 1 & 0 & -\frac{2}{5} & \frac{3}{5} & \frac{1}{5} & 0 & \frac{18}{5} \\ 0 & 1 & -\frac{3}{5} & \frac{2}{5} & -\frac{1}{5} & 0 & \frac{7}{5} \\ 0 & 0 & -\frac{1}{5} & -\frac{1}{5} & \frac{3}{5} & 1 & \frac{9}{5} \end{array}\right] \end{array}$$

$$\begin{array}{c} \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{array} \begin{array}{c} \begin{array}{ccccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 & \mathbf{a}_6 & \mathbf{b} \end{array} \\ \left[\begin{array}{ccccccc} 1 & 0 & \mathbf{0} & 1 & -1 & -2 & 0 \\ 0 & 1 & \mathbf{0} & 1 & -2 & -3 & -4 \\ 0 & 0 & \mathbf{1} & 1 & -3 & -5 & -9 \end{array}\right] \end{array}$$

- horrible. moving randomly we have a now non-feasible degenerate basic solution.
- yet we knew that pivoting randomly based only on $y_{r,e} \neq 0$ would lead us nowhere.

Adding the reduced costs

- What happens when we also pivot the last line?
- Remember the last line is equal to $(\mathbf{c}-\mathbf{z})'$ in the beginning.
- Remember also that

(a) in pivot row $a_{rj} \leftarrow a_{rj}/a_{re}$.
(b) in pivot column $a_{re} \leftarrow 1, a_{ie} = 0$ for $i = \mathbf{0}, 1, \cdots, m, i \neq r$: the eth column becomes a matrix of zeros and a one.
(c) for all other elements $a_{ij} \leftarrow a_{ij} - \frac{a_{rj}a_{ie}}{a_{re}}$

- Here, (a) does not apply, we cannot be in the pivot row.
- we have
  - in pivot column $a_{m+1,e} = 0$: makes sense, reduced cost is zero for basis elements.
  - for all other elements $a_{m+1,j} \leftarrow a_{m+1,j} - \frac{a_{rj}a_{m+1,e}}{a_{re}}$

Adding the reduced costs

- Recapitulating, at each iteration of the pivot the matrix is exactly

$$\begin{array}{|cccccc|c|}
\hline
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\
\vdots & B_{\mathbf{I}}^{-1}M & \vdots & \vdots & B_{\mathbf{I}}^{-1} & \vdots & B_{\mathbf{I}}^{-1}\mathbf{b} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\
\hline
\cdots & & (\mathbf{c}-\mathbf{z}) & & & \cdots & -x_0 \\
\hline
\end{array}$$

- The pivot is thus applied on the $m+1 \times n+1$ tableau.
- The tableau contains **everything we need**, reduced costs, (minus)objective, the coordinates of $B_{\mathbf{I}}^{-1}\mathbf{b}$ and $B_{\mathbf{I}}^{-1}A$

A quick comment on the initialization of the simplex

- We have seen that the simplex works when we **know an initial** feasible point.
- Sometimes, finding a feasible point is **as difficult as the problem itself.**
- How can we solve this?

Initialization methods exist. See lecture 7 of my course ORF522.

An Example from Portfolio Optimization

Simple Portfolio Theory

- n traded financial assets.
- For each asset a (random) return R_j at horizon T. $R = \frac{p_T}{p_0} - 1$.
- R_j is a $[-1, \infty)$-valued random variable. not much more...

Simple Portfolio Theory

- A (long) **portfolio** is a **vector** of $\mathbf{R}^n$ which represents the proportion of wealth invested in each asset.
- Namely $\mathbf{x}$ such that $x_1, \cdots, x_n \geq 0$ and $\sum_{x_i} = 1$.
- In \$ terms, Given M dollars, hold $M \cdot x_i$ of asset i.
- The performance of the portfolio is a random variable, $\rho(\mathbf{x}) = \sum_{i=1}^{n} x_i R_i$.

- Suppose $\mathbf{x} = \left[\frac{1}{3}\,\frac{1}{3}\,\frac{1}{3}\right]^T$ in the previous example.
- the realized value for $\rho(\mathbf{x})$ is $\frac{4.1\%}{3} + \frac{5.8\%}{3} + \frac{4.2\%}{3} = 4.7\% = 0.047$.

Simple Portfolio Theory

- For a second, imagine we **know** the actual return **realizations** r_j.
- Where would you invest?

- A bit ambitious.. we're not likely to be able see the future.
- Imagine we can **guess** realistically the expected returns $E(R_j)$.
- For instance, $\mathbb{E}[R_{\text{goog}}] = .5 = 50\%, \mathbb{E}[R_{\text{ibm}}] = .05 = 5\%, \mathbb{E}[R_{\text{dow}}] = .01 = 1\%$.
- If your goal is to maximize expected return,

$$\mathbf{x} = \text{argmax}(\mathbb{E}(\rho(\mathbf{x})),$$

  where would you put your money?

- The other question... **is that really what you want** in the first place?

Risk?

- **PHARMA** is a pharmaceutical company working on a new drug.
  - its researchers (or you) think there is a 50% probability that the new drug works
  - Let's do a binary scenario to keep things simple.
    - **the drug works and is approved by FDA**: PHARMA's market value is multiplied by 3. $R = 2$
    - **the drug does not work**: PHARMA goes bankrupt $R = -1$.
  - **Expected return**: $\mathbb{E}[R_{\text{PHARMA}}] = \frac{2 + -1}{2} = 1 = 100\%$. You are *expecting* to double your bet.

- **BORING** is a company that produces and sells screwdrivers.
  - The return is uniformly distributed between $-.01 = -1\%$ and $.02 = 2\%$
  - **Expected return** is $.0005$, that is 0.5%.

- Would you bet everything on PHARMA with these cards? **something is missing in our formulation**

Risk?

- Portfolio optimization needs to input the investor's aversion to risk.
- If the investor uses $\mathbf{x} = \operatorname{argmax}(\mathbb{E}(\rho(\mathbf{x})))$, he **forgot about risk**.
- Solution: include risk in the program. Risk is vaguely a **quantification** of the **dispersion / entropy** of the returns of a portfolio.
- Different choices:
  - **Variance**:
    - ▷ C is the covariance matrix of the vector r.v. R takes values in $\mathbf{R}^n$, $C = \mathbb{E}[(R - \mathbb{E}[R])(R - \mathbb{E}[R])^T]$.
    - ▷ The variance of $\rho(\mathbf{x})$ is simply $\mathbf{x}^T C \mathbf{x}$.
    - ▷ Maximal expected return under variance constraints = mean-variance optimization.
  - **Mean-absolute deviation (MAD)**:
    - ▷ Namely $\mathbb{E}\left[|(\rho(\mathbf{x}) - \mathbb{E}[\rho(\mathbf{x})])|\right] = E[|\mathbf{x}^T \bar{R}|]$ where $\bar{R} = R - \mathbb{E}[R]$.
    - ▷ Penalized estimation: $\mathbf{x} = \underset{\mathbf{x}\geq 0, \mathbf{x}^T \mathbf{1}_n = 1}{\operatorname{argmax}} \underbrace{\lambda}_{\text{trade-off}} \cdot \underbrace{\mathbb{E}[\rho(\mathbf{x})]}_{\text{expected return}} - \underbrace{\mathbb{E}[|\mathbf{x}^T \bar{R}|]}_{\text{risk}}$.

Risk

- The **variance** formulation leads to a quadratic program:

$$\begin{array}{ll} \text{maximize} & \mathbf{x}^T \mathbb{E}[R] \\ \text{subject to} & \mathbf{x} \geq 0, \mathbf{x}^T \mathbf{1}_n = 1 \\ & \mathbf{x}^T C \mathbf{x} \leq \lambda \end{array}$$

- The **MAD** formulation leads to something closer to linear programming:

$$\begin{array}{ll} \text{maximize} & \lambda \mathbf{x}^T \mathbb{E}[R] - \mathbb{E}[|\mathbf{x}^T \bar{R}|] \\ \text{subject to} & \mathbf{x} \geq 0, \\ & \mathbf{x}^T \mathbf{1}_n = 1 \end{array}$$

- **Problem**: lots of expectations $\mathbb{E}$...
- We need to fill in some expected values above by some **guesses**.

Approximations

- We write $\tilde{\mathbf{r}}$ for $\mathbb{E}[R]$ which can be guessed according to...
  - research, analysts playing with excel, valuation models.
  - historical returns.
- We also need to approximate $\mathbb{E}[|\mathbf{x}^T \bar{R}|]$.
- Suppose we have a history of N returns $(\mathbf{r}^1, \cdots, \mathbf{r}^N)$ where each $\mathbf{r} \in \mathbf{R}^n$.
  - Write $\bar{\mathbf{r}} = \sum_{j=1}^N \mathbf{r}^j$.
  - in practice, approximate $\mathbb{E}[|\mathbf{x}^T \bar{R}| \approx \sum_{j=1}^N |\mathbf{x}^T(\mathbf{r}^j - \bar{\mathbf{r}})|$
- this becomes:

$$\begin{array}{ll} \text{maximize} & \lambda \mathbf{x}^T \mathbf{r} - \frac{1}{N} \sum_{j=1}^N |\mathbf{x}^T(\mathbf{r}^j - \bar{\mathbf{r}})| \\ \text{subject to} & \mathbf{x} \geq 0, \mathbf{x}^T \mathbf{1}_n = 1 \end{array}$$

Approximations

- Now add artificial variables $y_j = |\mathbf{x}^T(\mathbf{r}^j - \bar{\mathbf{r}})|$. One for each observation. Now,

$$
\begin{array}{ll}
\text{maximize} & \lambda \mathbf{x}^T \mathbf{r} - \frac{1}{N}\sum_{j=1}^{N} y_j \\
\text{subject to} & \mathbf{x} \geq 0, \\
& y_j \geq 0, \\
& \mathbf{x}^T \mathbf{1}_n = 1, \\
& -y_j \leq \mathbf{x}^T(\mathbf{r}^j - \bar{\mathbf{r}}) \leq y_j, \quad j = 1, \cdots, N
\end{array}
$$

- Use the simplex...