JUNIA Grande école d'ingénieurs
HEI·ISEN·ISA

#Class 3

Linear Programming

Amina El Yaagoubi
Samuel Deleplanque

September 2024

Before we start…

Some keywords

- Linear Program;
- Feasible region;
- Feasible solution;
- Optimal solution;
- Optimal objective value;
- Standard form;
- Canonical form;
- Slack variables;
- .lp files;
- Primal program;
- Dual program.

Why Linear Programming (LP)?

- Although reality is often far from linear, a large number of problems can be formulated in linear form, either **directly** or as a first **simplification**. Furthermore, a very large number of models are **extensions** of linear programs. Understanding linear programming is essential for comprehending more sophisticated models.

https://www.vox.com/2018/4/28/17292244/flat-earthers-explain-philosophy

More formally, … an LP is:

- … a specific case of mathematical programming, i.e., optimizing a function subject to constraints, most commonly encountered in the industry.

$$\begin{array}{rc} Max\ or\ Min & f(x) \\ s.t. & g(x) = 0 \\ & h(x) \leq 0 \\ & x \in X \end{array}$$

- Where, $\mathrm{X} \subset \mathbb{R}^n$ $(\neq \emptyset)$, $f\colon X \to \mathbb{R}$, $g\colon X \to \mathbb{R}^p$, $h\colon X \to \mathbb{R}^q$.
- **f is linear, and both g and h are affine.**

Framework of LP

LP makes four assumptions:

- **Linearity**: implies that the objective function and all of the constraints are linear relationship.
  - A function $(x_1, x_2, \dots, x_n) \mapsto f(x_1, x_2, \dots, x_n)$ is linear if and only if there exist a set of constants $c_1, c_2, \dots, c_n$ such that the following condition is satisfied:

$$f(x_1, x_2, \dots, x_n) = c_1 x_1 + c_2 x_2 + \dots + c_n x_n$$

$$\cancel{\begin{array}{l} \min 39(x_1)^2 + 69(x_2)^2 \\ 2{,}5\, x_1 + 2{,}5\, x_2 \leq 240 \end{array}} \qquad \cancel{\begin{array}{l} \min 39\, x_1 - x_2 \\ 2{,}5\, (x_1)^2 + x_2 \leq 240 \end{array}} \qquad \cancel{\begin{array}{l} \max 39\, x_1 + x_2 x_3 \\ x_1 + 5x_2 + x_3 \geq 240 \end{array}} \qquad \dots$$

  - Nowhere can we have terms such as $x^3, \log(x)$ or $x_1 x_2$ appearing.
  - Proportionality and additivity are consequences of the linear assumption.
    - Proportionality : the contribution of each variable to the value of the objective function/left term of the constraint is proportional to the value of that variable.
    - Additivity : the objective function/left term of the constraint is composed of the sum of the individual contributions of each variable.

Framework of LP

LP makes four assumptions:

- **Linearity**: implies that the objective function and all of the constraints are linear relationship.
- **Divisibility**: means that the optimal values of decision variables may be fractional depending upon the application.
- **Certainty**: requires that the parameters of LP model are known or can be accurately estimated.
- **Non-negativity**: simply means that all decision variables cannot be less than zero.

Note:

- LP has a finite number of real variables.
- Big difference from systems of linear equations is the existence of objective function and linear inequalities (instead of equalities).

LP forms

- A linear program is formulated in the following way:

$$
\begin{array}{llllll}
\max_x & c_1 x_1 & + \ \dots \ + & c_n x_n & & \\
 & a_{11} x_1 & + \ \dots \ + & a_{1n} x_n & \left\{\begin{array}{c} \leq \\ \geq \\ = \end{array}\right\} & b_1 \\
 & \vdots & & \vdots & & \\
 & a_{m1} x_1 & + \ \dots \ + & a_{mn} x_n & & b_m, \\
 & \multicolumn{5}{l}{x_i \geq 0, \ i = 1, \dots, n,}
\end{array}
$$

- Or, in a more compact form:

$$
\begin{aligned}
&\max_x \sum_{j=1}^{n} c_j x_j \\
&\sum_{j=1}^{n} a_{ij} x_j \left\{\begin{array}{c} \leq \\ \geq \\ = \end{array}\right\} b_i, \quad i = 1, \dots, m. \\
&x_j \geq 0, \ j = 1, \dots, n,
\end{aligned}
$$

- Or, in a matrix form, where:
- $x, c \in \mathbb{R}^n, b \in \mathbb{R}^m$, b : right-hand side (RHS)

$$
\begin{aligned}
&\max_x c^T x \\
&Ax \left\{\begin{array}{c} \leq \\ \geq \\ = \end{array}\right\} b, \\
&x \geq 0
\end{aligned}
$$

$$
x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}
$$

and

$$
A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}
$$

All the mathematical models in our previous class were LP.

Solving LP models using Graphical Method

Applicable to ***linear*** programming involving less than three variables …

Some terminologies and tools

- Solution: the assignment of values to variables.
- Feasible solution: a solution whose values satisfy all of the constraints. It is a valid solution within the problem's limitations.
- Feasible region: the set of all feasible solutions. It represents all the valid combinations of variable values.
- Optimal solution : a feasible solution that either maximizes or minimizes the objective function, depending on the nature of the optimization problem. It is the best solution within the given constraints.
- **Useful tools:**
  - Let us use GeoGebra and/or Desmos to search for some optimums…

GeoGebra desmos

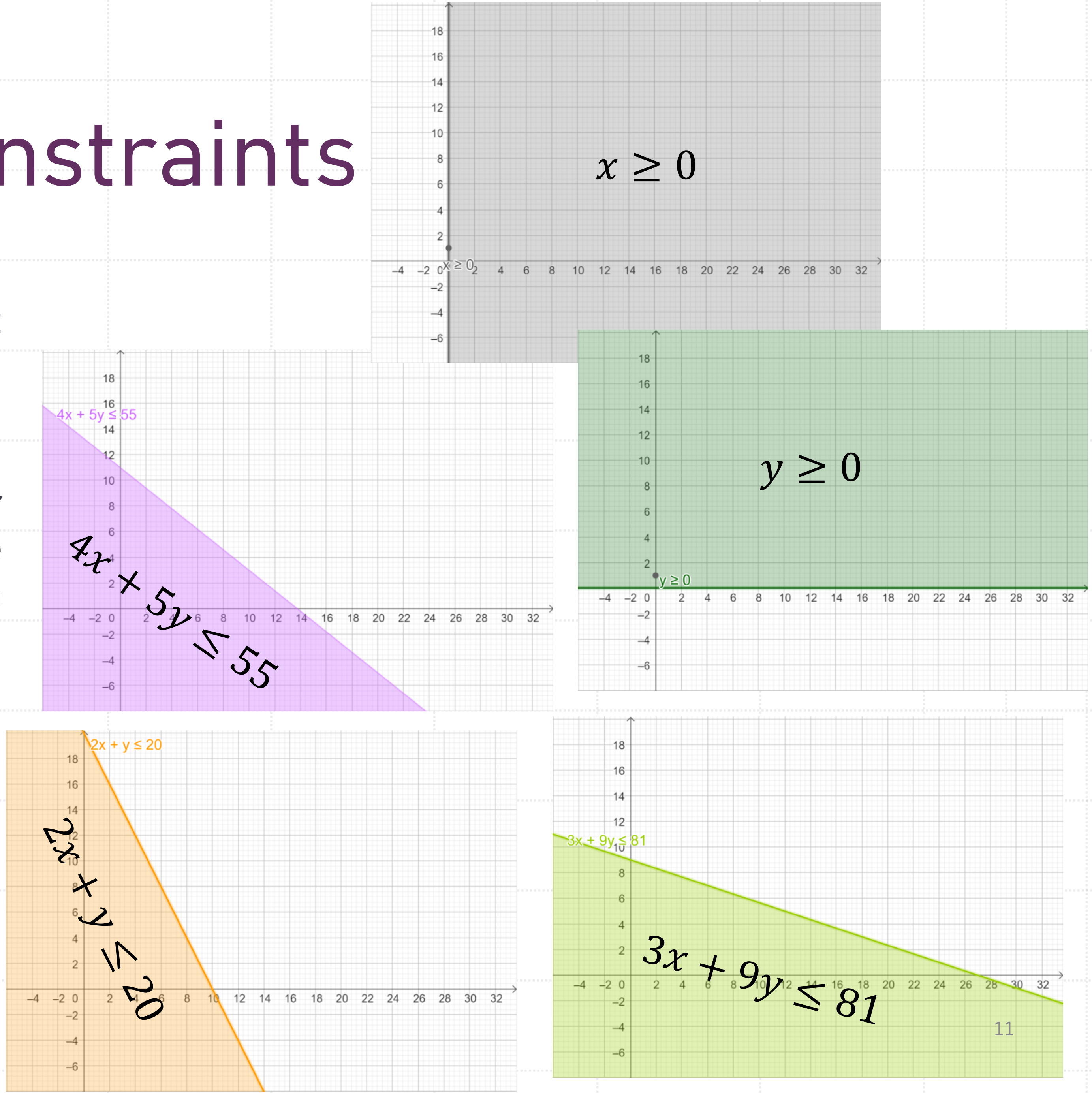
Interpreting the constraints

We will now examine the constraints of "our first mathematical program (**LP**)" (slide 14 #Class2).

Each line in these figures is represented by a constraint expressed as an equality. The color area associated with each line shows the direction indicated by the inequality sign in each constraint.

$$\begin{cases} maximize\ f(x,y) = 6x + 4y \\ s.t. \\ \qquad 3x + 9y \leq 81 \\ \qquad 4x + 5y \leq 55 \\ \qquad 2x + y \leq 20 \\ \qquad x, y \geq 0 \end{cases}$$

Half-planes in $\mathbb{R}^2$



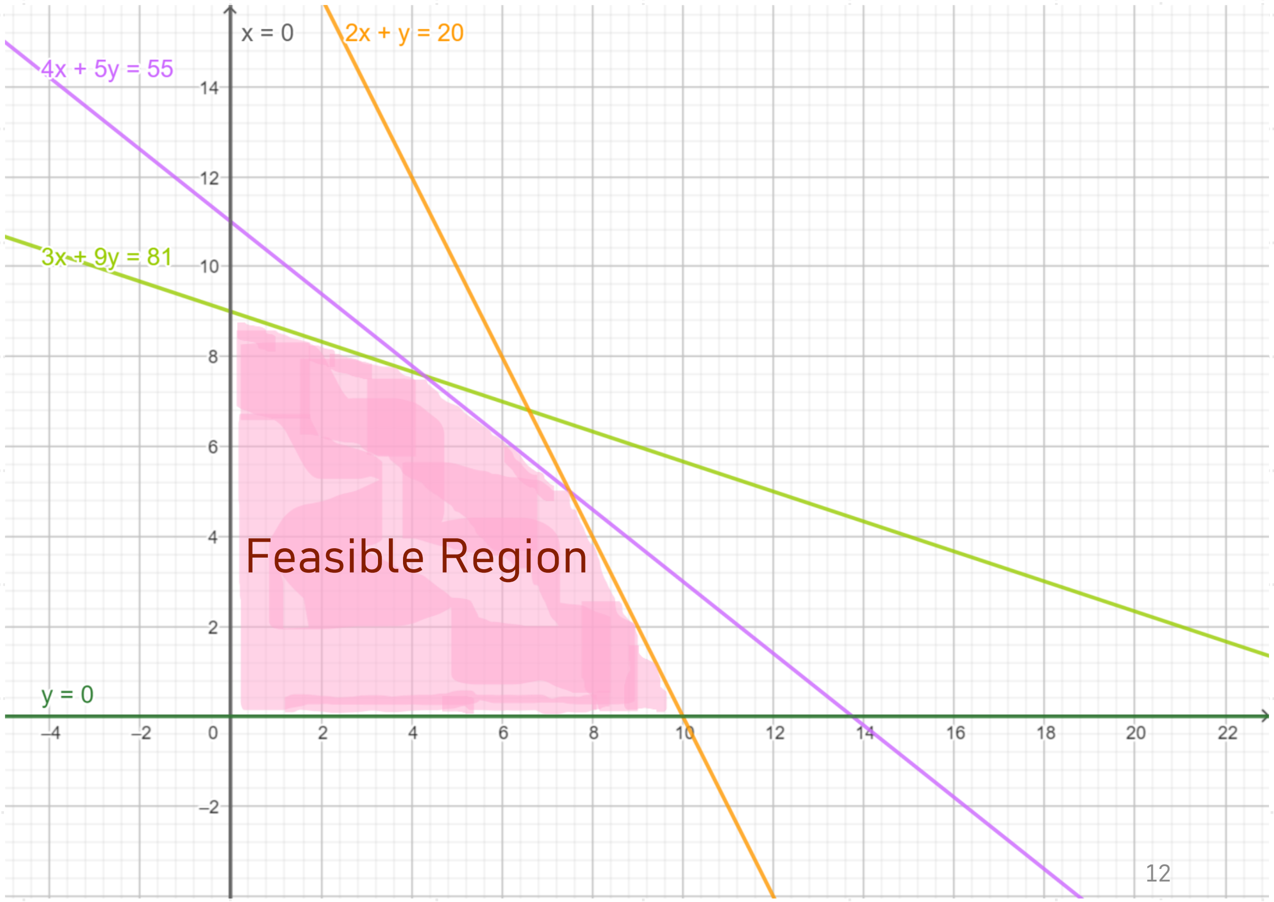
Interpreting the constraints

The shaded area in the accompanying figure represents all the values of the variables x and y that satisfy these constraints simultaneously : feasible solutions to the problem.

Feasible region (feasible solution set) is the intersection of the colored half-planes.

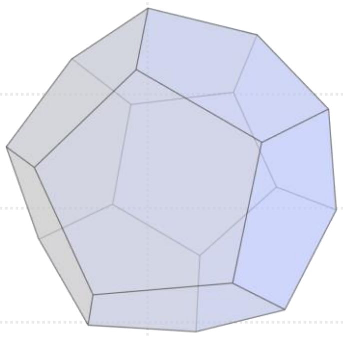
Any alternative solution not within the feasible region must violate at least one of the constraints of the problem.

… Give an example of non-feasible solution.

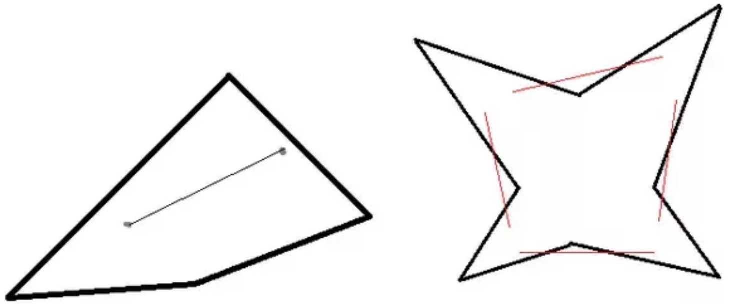


A Bit of Geometry…

- Inequalities represent half-planes (in 2D) or hyperplanes.
- **The set of feasible solutions** → a set of inequalities can construct a bounded feasible space, known as a convex polygon (2D) or convex polyhedron.



- **A quick reminder about convexity:**
- $f: X \to \mathbb{R}$ is convex : $\forall t \in [0,1], \forall x_1, x_2 \in X$:
$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$
- … Let x_1 and x_2 be two feasible solutions of a convex polyhedron, then $(x_1 + x_2)/2$ is also a solution of this polyhedron.
- It can be easily verified that a polyhedron is convex, meaning that for any pair of points x_1 and x_2 within a polyhedron P, all points along the line segment $[x_1, x_2]$ are contained within P.



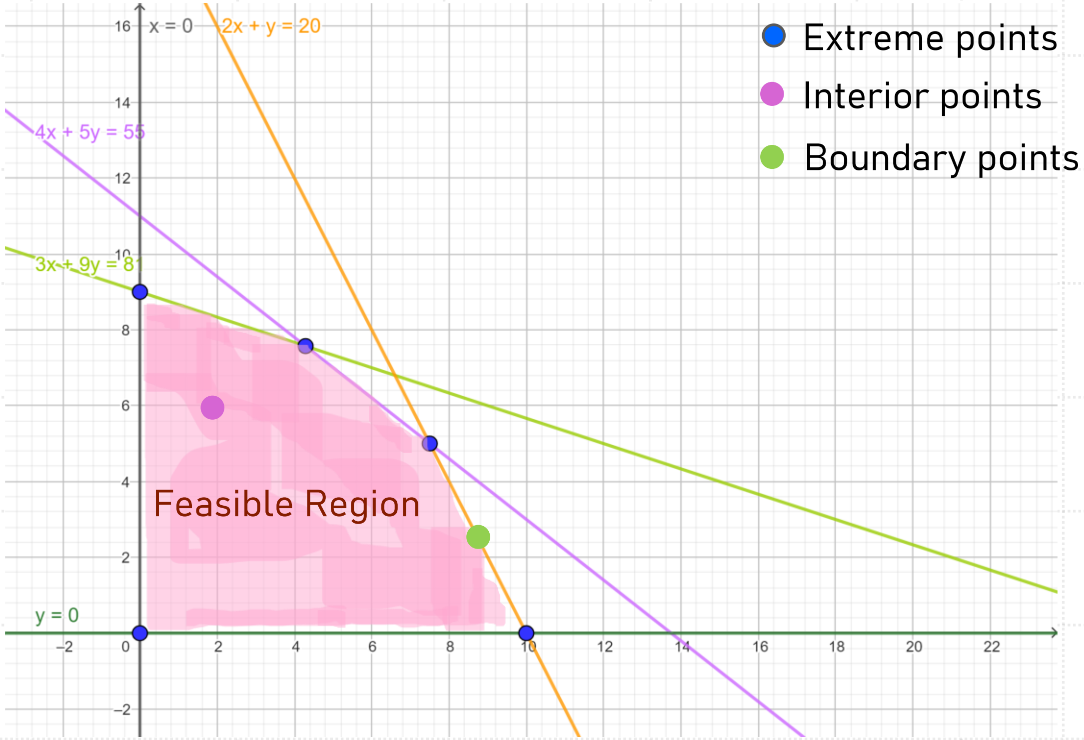
Convex Polygon

Non convex polygon

A Bit of Geometry…

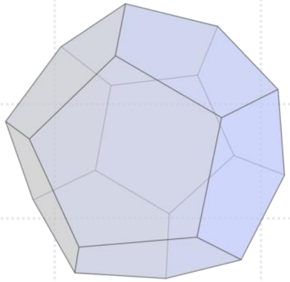
- The good news: we can only consider the extreme points!
  - **Optimal solution achieved at the boundary of the feasible region:** the optimum of the objective function, if it exists, is achieved at (at least) one vertex of the polyhedron (**extreme points**).
- The bad news: the number of extreme points can be exponentially high compared to the number of decision variables.
- The consolation prize: convexity ensures that local optima are global optima. In other words, an extreme point is optimal if all its neighbors are worse than it. But beware of trays “plateaux”…

3 types of feasible points:



In the general case

- **Fact**: even if the case of alternative solutions does arise, there will always be an optimal solution that lies at a vertex (extreme point).
- This fact generalizes to problems with more variables (which would need more dimensions to be represented geometrically).
- It is this fact that makes **linear** programming models comparatively easy to solve.



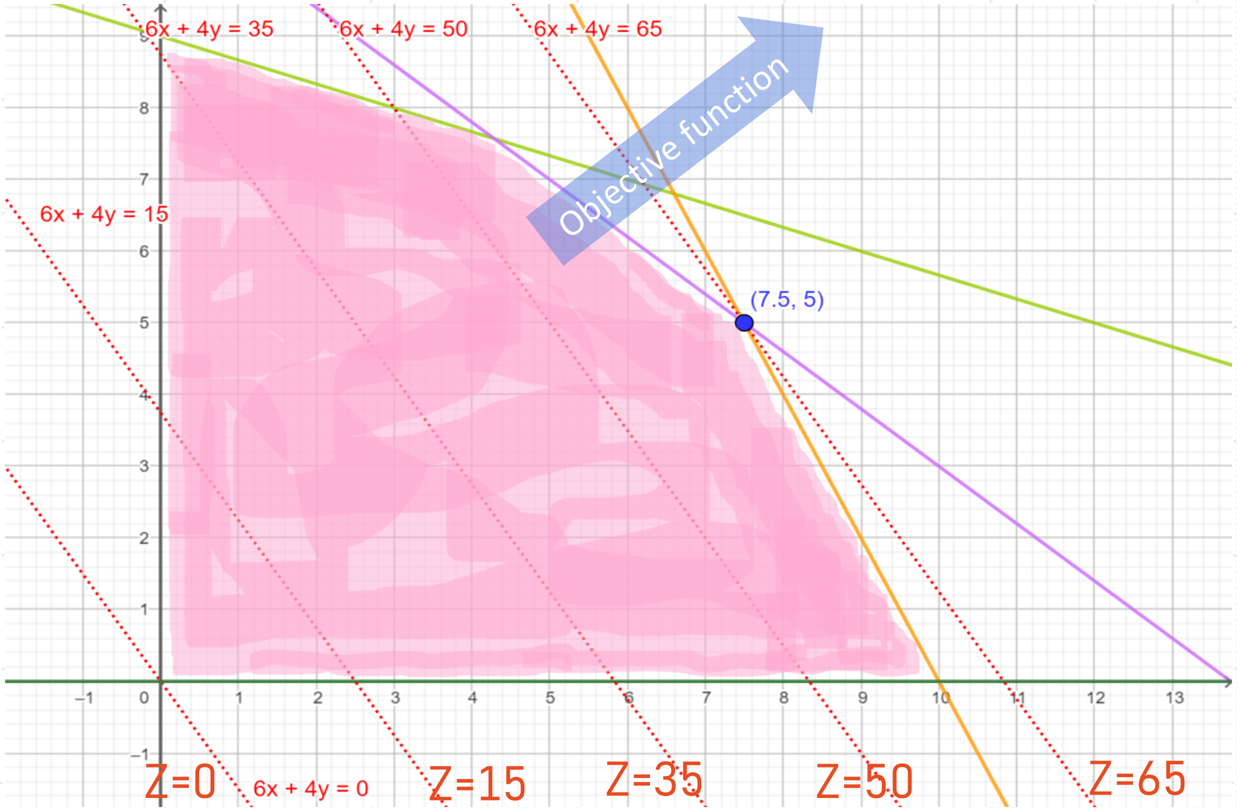
- **Good to know**: an algorithm works by only examining vertex solutions, i.,e. extreme points (rather than the generally infinite set of feasible solutions): The ***simplex*** algorithm.

... Back to our model

Interpreting the objective function : first method

- Among the feasible solutions, we want to find the values of the variables x and y that maximize the total profit from the sale of the two products.
- We draw the lines corresponding to changes in the objective:
  - **Level curves** $\{6x + 4y = z\}$ are parallel lines. Maximizing is equivalent to increasing z (maximizing as long as we remain feasible).
  - **In general:** Level curves $\{f = constant\}$ of the objective function f are affine hyperplanes ($n = 2$ => line, $n = 3$ => plane...).

$$maximize\ f(x, y) = 6x + 4y$$

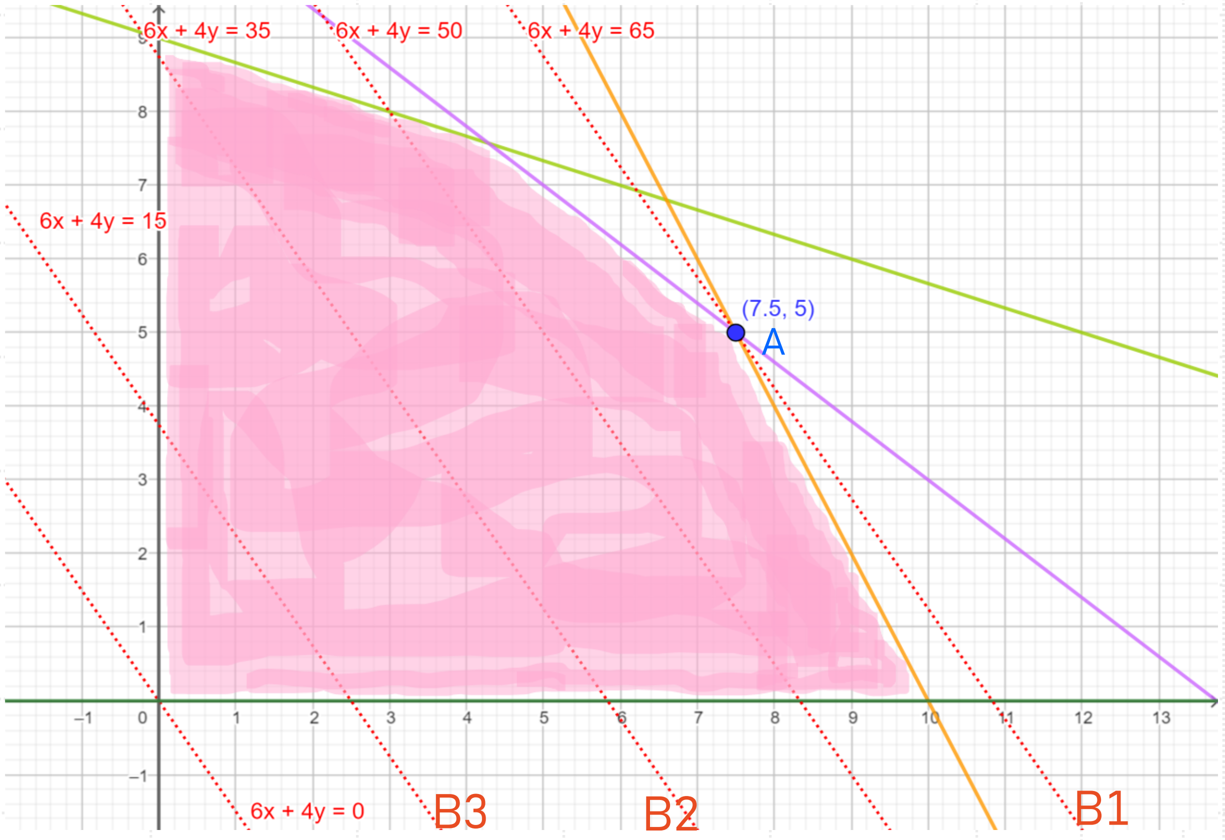


Interpreting the objective function

The optimal solution is represented by the intersection of the second and third constraints in the model. Evaluating the coordinates of this point is equivalent to solving the following linear system:

$$\begin{cases} 4x + 5y = 55 \\ 2x + \ \ y \ = 20 \end{cases}$$

- According to the graph, it corresponds to the point in blue A where $6x + 4y$ has a value of 65.
- Any point on the same broken line B1 in the accompanying figure will give the objective $6x + 4y$ this value (65).
- Other values of the objective correspond to lines parallel to this:
  - 35 for B2
  - 15 for B3, …etc.



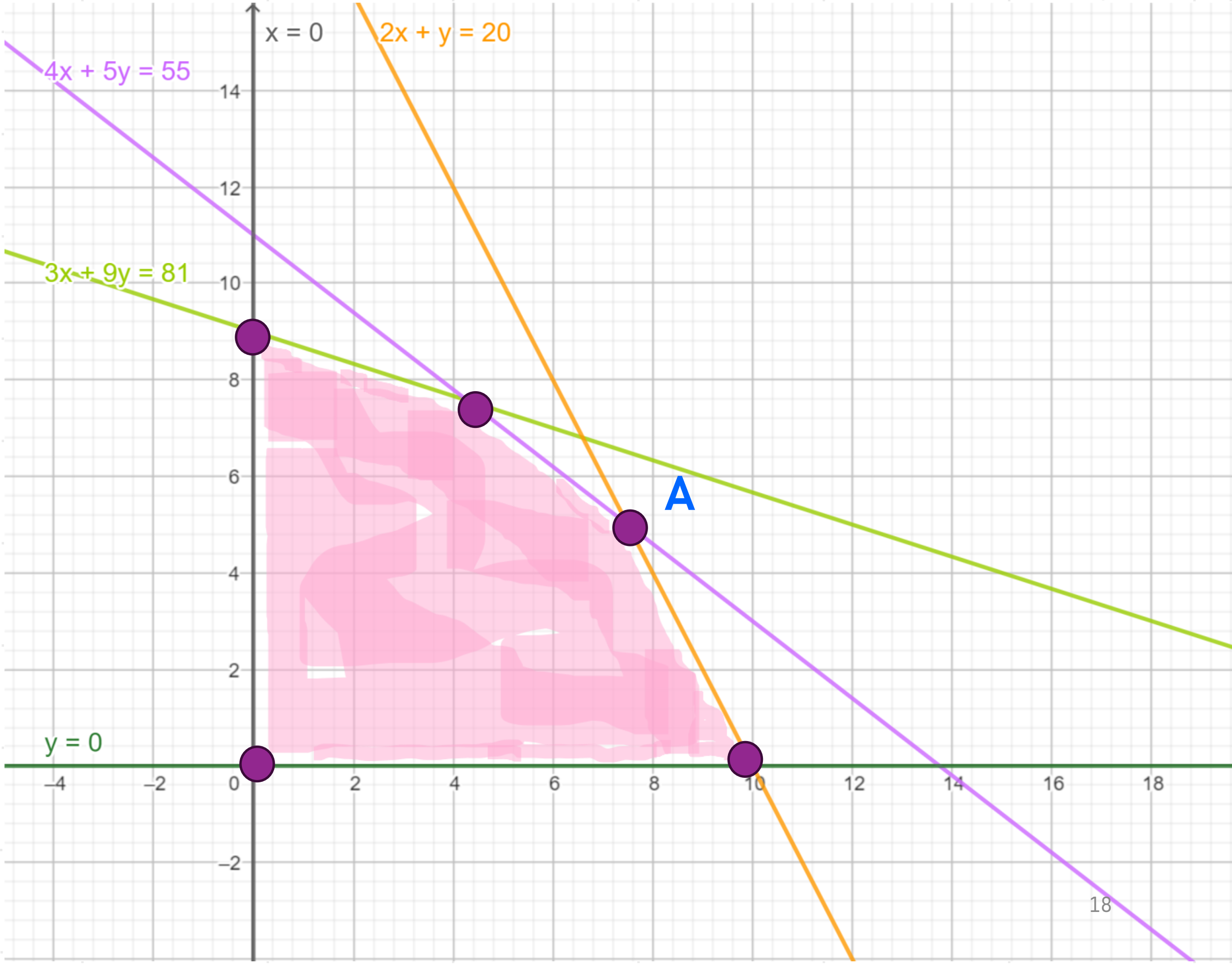
Question: What is the difference between optimal solution and optimal objective value?

Interpreting the objective function : second method

By hand:

After identifying the feasible solution region

1. Determine the coordinates of the extreme points by solving the equations of the intersecting lines at each of these points.
2. Evaluate the objective function at extreme points to get the required maximum/minimum value of the objective function.



What about minimization problems?

- Solve the following program graphically :

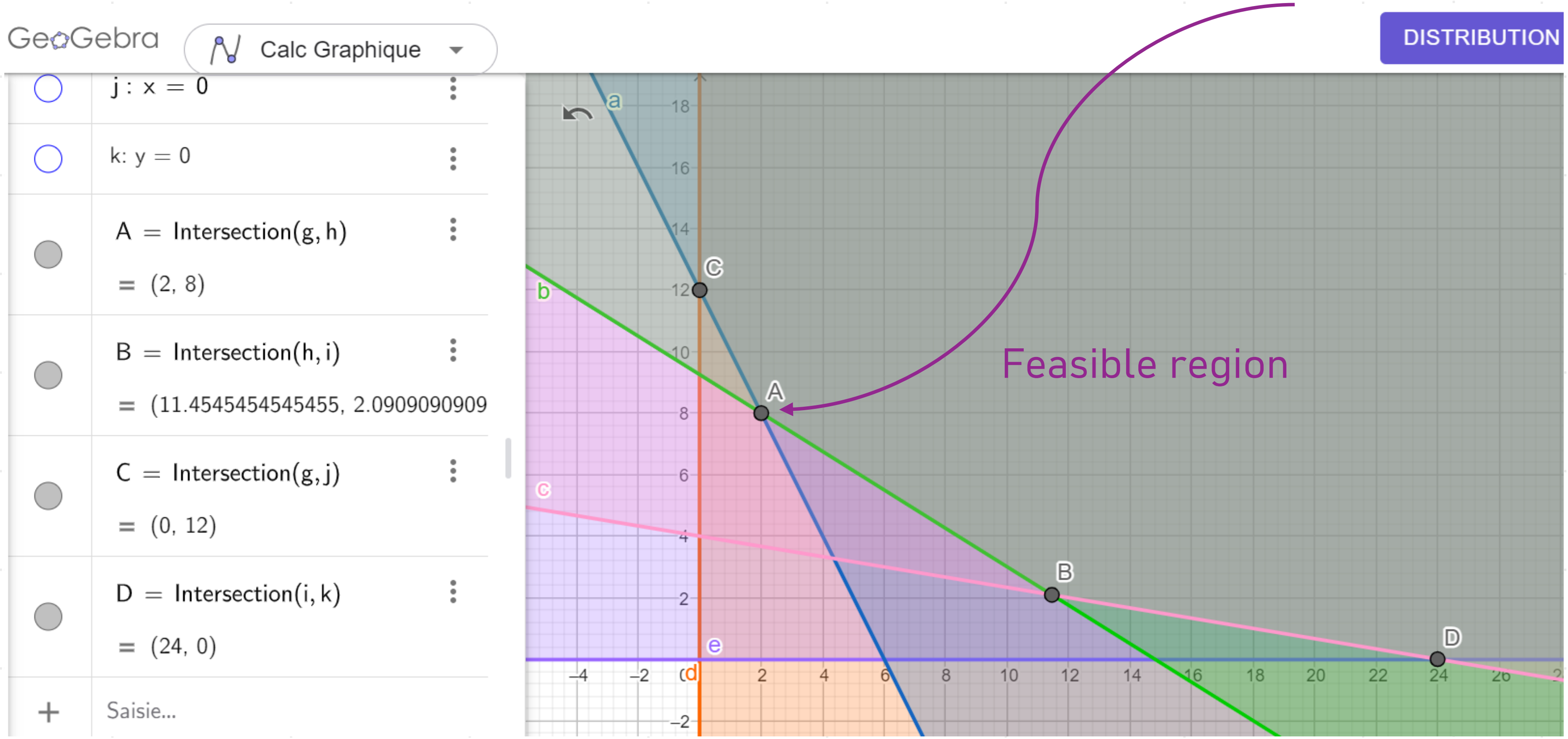
$$
\begin{aligned}
&Min && x_1 + x_2 \\
&s.t. && \\
& && 2x_1 + x_2 \geq 12 \\
& && 5x_1 + 8x_2 \geq 74 \\
& && x_1 + 6x_2 \geq 24 \\
& && x_1, x_2 \geq 0
\end{aligned}
$$

What about minimization problems?

Overview of the solution ->

$$
\begin{aligned}
&Min \quad && x_1 + x_2 \\
&s.t. && \\
& && 2x_1 + x_2 \geq 12 \\
& && 5x_1 + 8x_2 \geq 74 \\
& && x_1 + 6x_2 \geq 24 \\
& && x_1, x_2 \geq 0
\end{aligned}
$$

The optimal solution



Feasible region

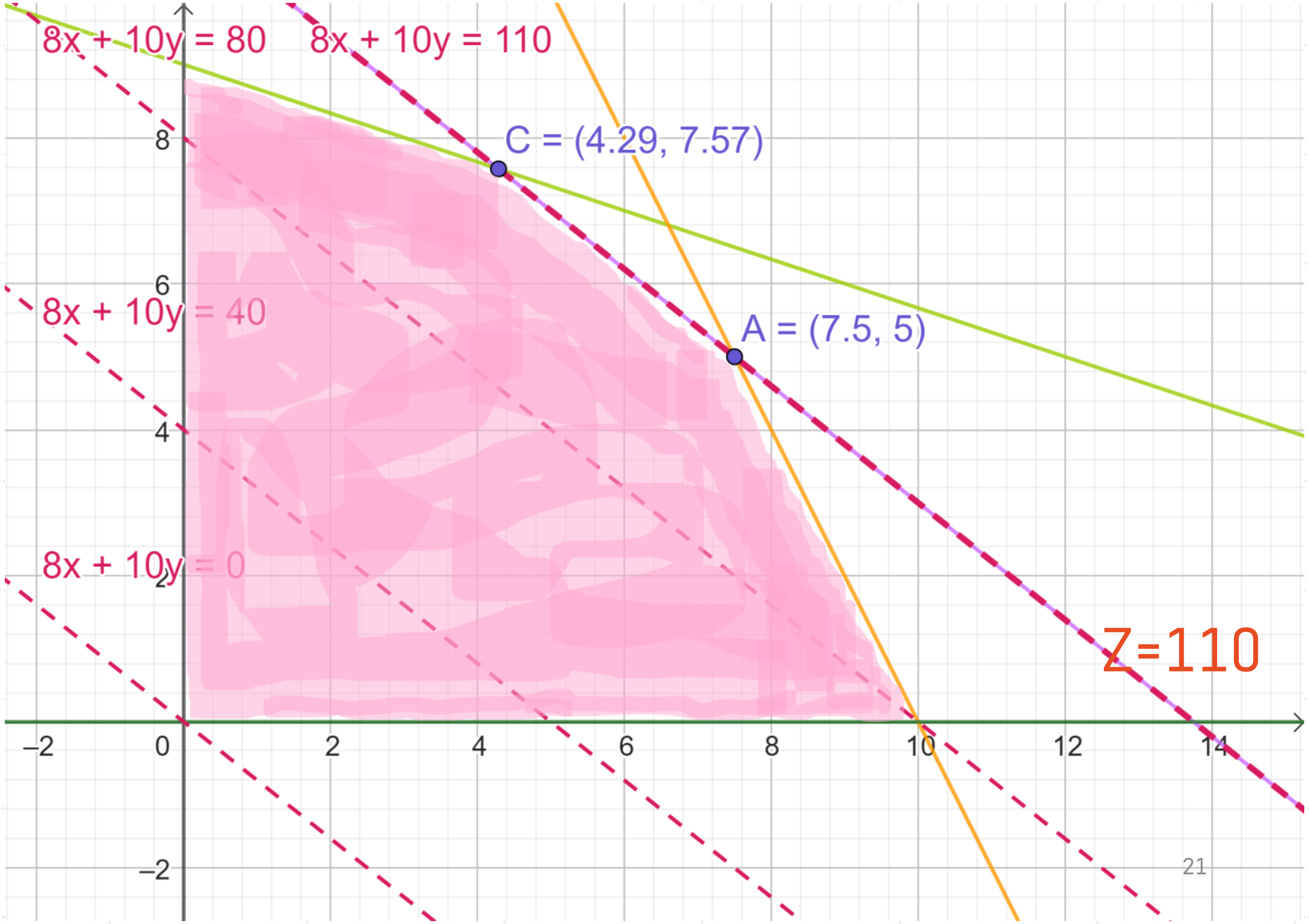
What if…?

Remember the “But beware of trays “plateaux”…” in slide 14, was it that clear?

- It is possible that the objective lines might be parallel to one of the sides of the feasible region.
- For example, if the objective function in the previous example were $8x + 10y$, the objective lines would be parallel to (AC). The point A would then still be an optimal solution, but C would be also, and any point between A and C.
- →Any weighted average of multiple optimal solutions is also an optimal solution

- What is the set of optimal solutions in this case?

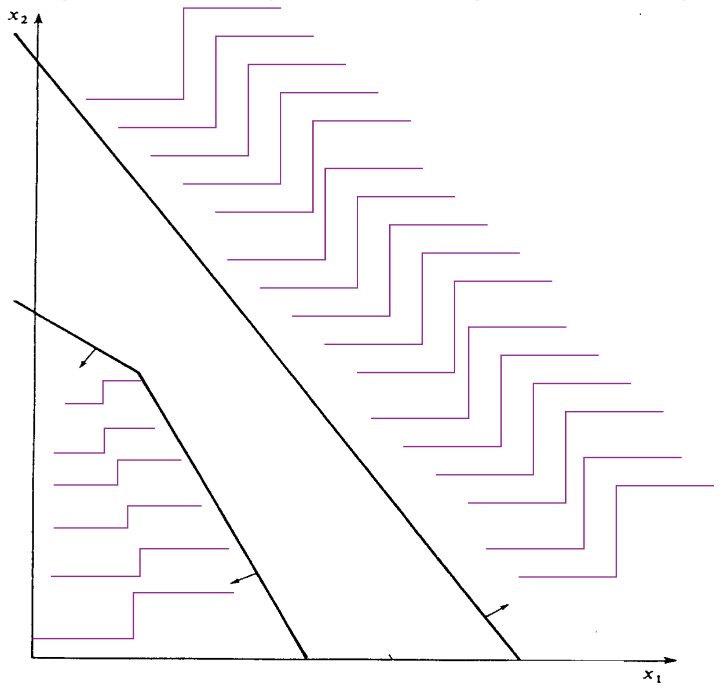
→ **Multiple optimal solutions case.**



Other irregular types of models

- **Non-existent feasible solutions:**
  - The solution does not exist: the case of an open polygon or an empty polygon.
  - There is no point that satisfies all the constraints simultaneously.

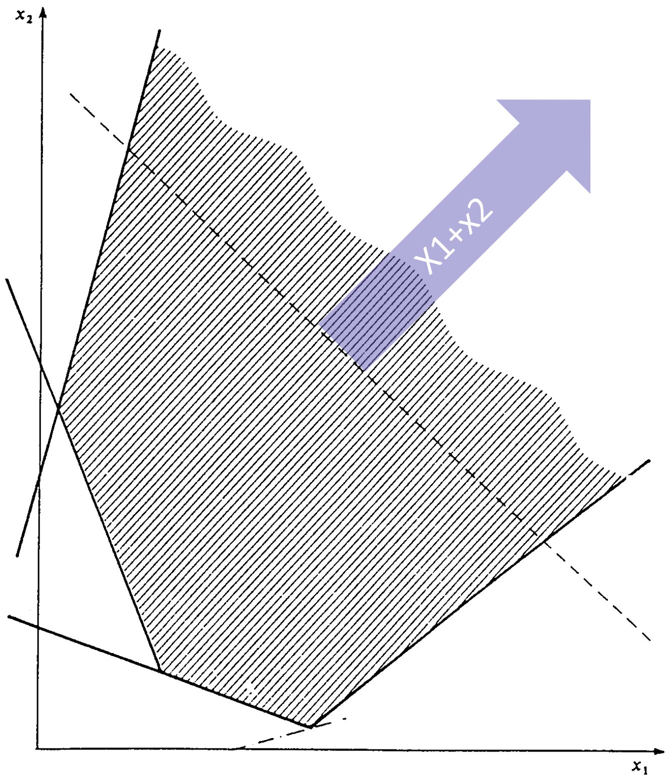
$$
\begin{aligned}
Max \quad & 4x_1 + 3x_2 \\
s.t. \quad & \\
& 3x_1 + 4x_2 \leq 12 \\
& x_1 + x_2 \geq 4 \\
& 4x_1 + 2x_2 \leq 8 \\
& x_1, x_2 \geq 0
\end{aligned}
$$



Other irregular types of models

- **Unbounded problem**
  - No optimal solution: One can increase the value of the objective function in the direction of the arrow indefinitely, so the solution is unbounded.

$$
\begin{array}{ll}
Max & x_1 + x_2 \\
s.t. & \\
& -4x_1 + x_2 \leq 2 \\
& x_1 + x_2 \geq 3 \\
& x_1 + 2x_2 \geq 4 \\
& x_1 - x_2 \leq 2 \\
& x_1, x_2 \geq 0
\end{array}
$$



Good to know

It is worth giving an indication of why linear programming models can be solved more easily than non-linear ones:

- The simple property of *linear* programming models (**which one?**) may not apply well to non-linear programming models:
  - For models with a non-linear objective function, the objective lines in two dimensions would no longer be straight lines.
  - If there were non-linearities in the constraints, the feasible region might not be bounded by straight lines either.
- In these circumstances, the optimal solution might well not lie at a vertex. It might even lie in the interior of the feasible region. Moreover, having found a solution, it may be rather difficult to be sure that it is optimal (so-called local optima may exist).

The use of ‘black box’ solvers…

- **NEOS Server**
  - https://neos-server.org/neos/solvers/milp:CPLEX/LP.html (for both real and/or integer variables)
  - Consult the **CPLEX** .lp file documentation (CPLEX lp files (mit.edu) and LP file format: algebraic representation - IBM Documentation) to properly define and submit the correct variable types to the solver.

We need .lp files …

- .lp files are text files that contain the formulation of an optimization problem, typically in the context of linear programming or mixed-integer linear programming (which we will cover in the next class). These files follow a specific format recognized by optimization solvers such as CPLEX, Gurobi, and GLPK.
- They serve as input for optimization solvers : the solver reads the .lp file, interprets the problem structure, and computes the optimal solution according to the given objective function and constraints.

- An .lp file describes:
  - Objective function (min or max)
  - Constraints
  - Bounds
  - Variables type (continuous, integer, or binary)

Defines the upper or lower limits of variables (optional if the default is $0 \leq x \leq \infty$).

Example of .lp file

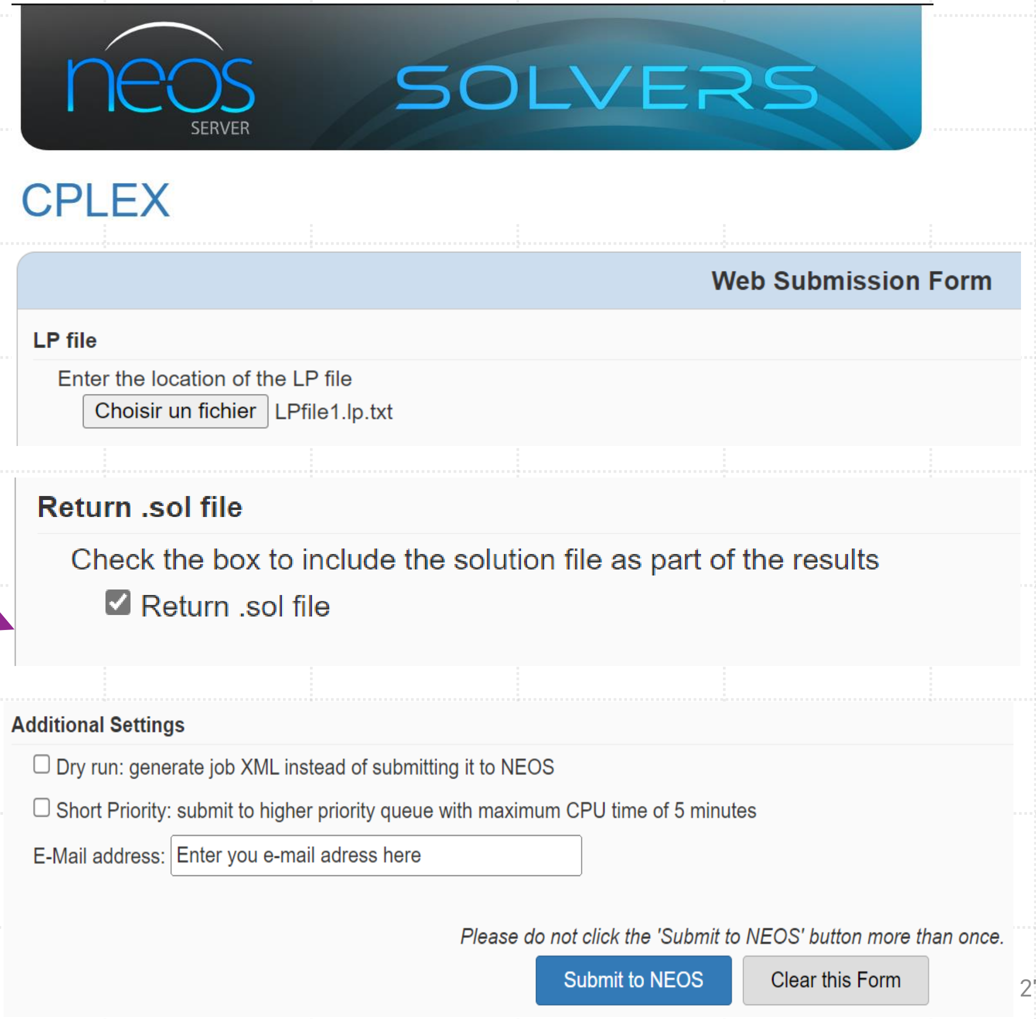
$$
\begin{aligned}
&Max \quad x_1 + 2x_2 + 3x_3 + x_4 \\
&s.t. \\
&-x_1 + x_2 + x_3 + 10x_4 \leq 20 \\
&x_1 - 3x_2 + x_3 \leq 30 \\
&x_2 - 3.5x_4 = 0 \\
&x_1 \leq 40 \\
&x_4 \leq 3 \\
&x_4 \geq 2 \\
&x_1, x_2, x_3, x_4 \geq 0
\end{aligned}
$$

LPfile1.lp - Bloc-notes

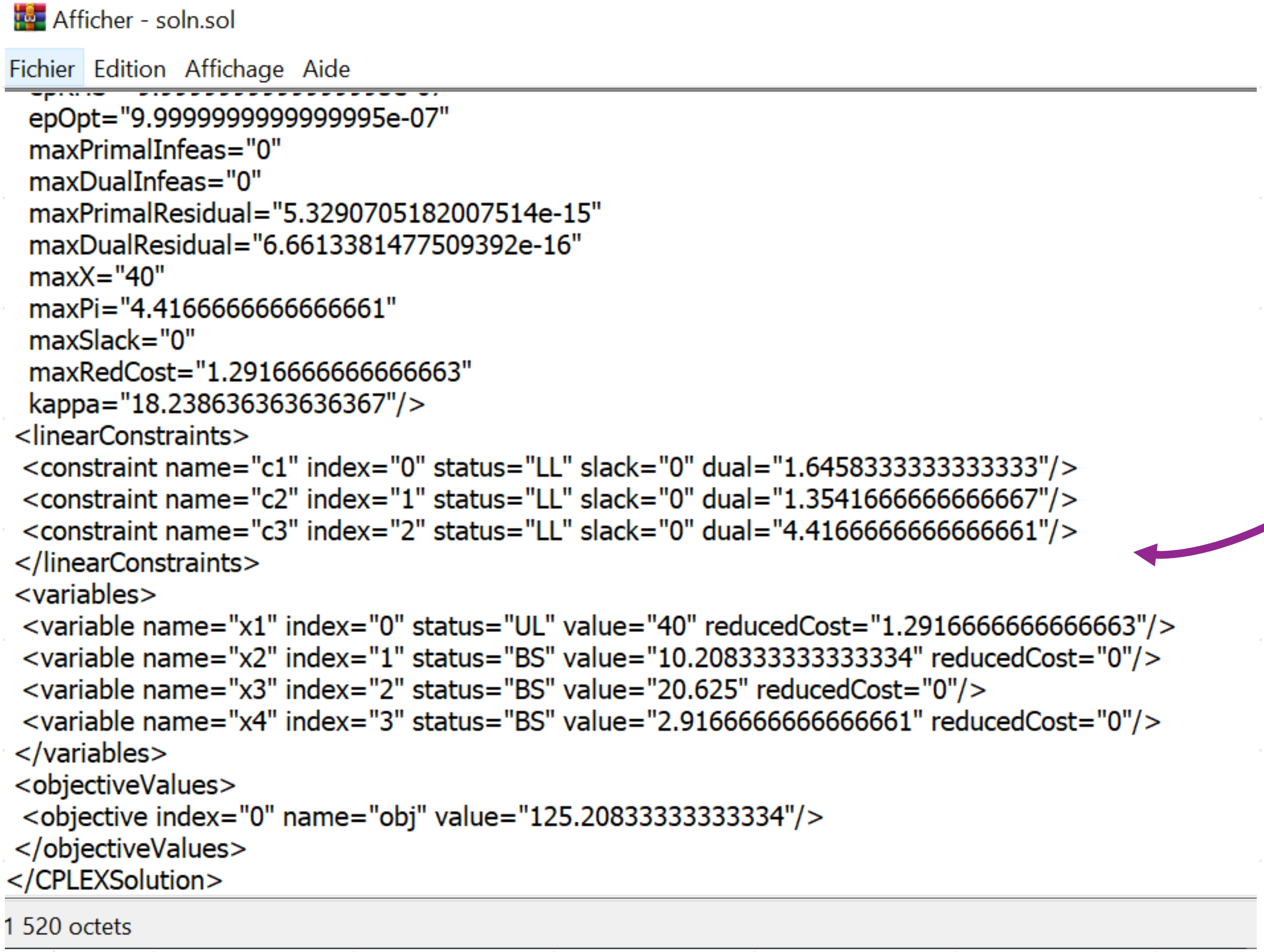
Fichier Edition Format Affichage Aide

```
Maximize
 obj: x1 + 2 x2 + 3 x3 + x4
Subject To
 c1: - x1 + x2 + x3 + 10 x4 <= 20
 c2: x1 - 3 x2 + x3 <= 30
 c3: x2 - 3.5 x4 = 0
Bounds
 0 <= x1 <= 40
 2 <= x4 <= 3
End
```

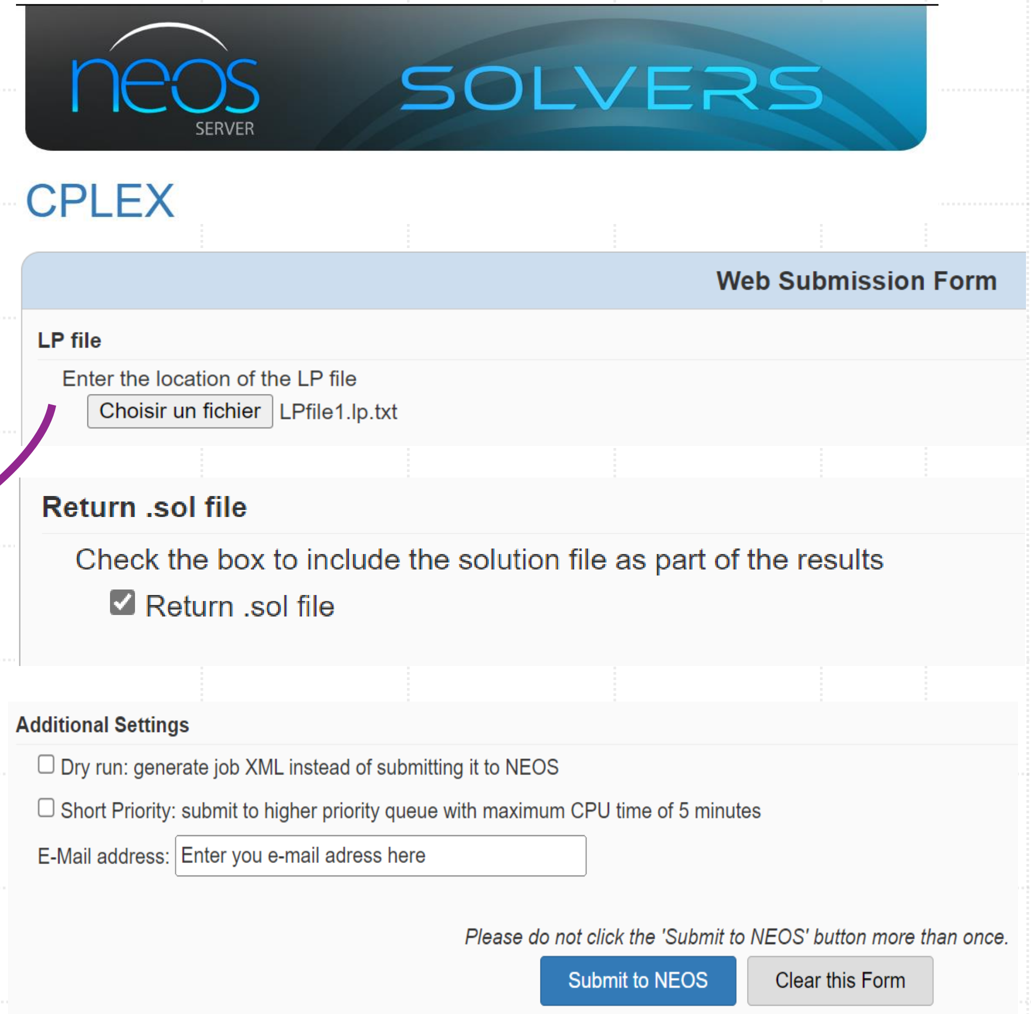
A simple technique is to create a similar file using a standard text editor and to read it into the solver.



The solution file




```
  epOpt="9.9999999999999995e-07"
  maxPrimalInfeas="0"
  maxDualInfeas="0"
  maxPrimalResidual="5.3290705182007514e-15"
  maxDualResidual="6.6613381477509392e-16"
  maxX="40"
  maxPi="4.4166666666666661"
  maxSlack="0"
  maxRedCost="1.2916666666666663"
  kappa="18.238636363636367"/>
 <linearConstraints>
  <constraint name="c1" index="0" status="LL" slack="0" dual="1.6458333333333333"/>
  <constraint name="c2" index="1" status="LL" slack="0" dual="1.3541666666666667"/>
  <constraint name="c3" index="2" status="LL" slack="0" dual="4.4166666666666661"/>
 </linearConstraints>
 <variables>
  <variable name="x1" index="0" status="UL" value="40" reducedCost="1.2916666666666663"/>
  <variable name="x2" index="1" status="BS" value="10.208333333333334" reducedCost="0"/>
  <variable name="x3" index="2" status="BS" value="20.625" reducedCost="0"/>
  <variable name="x4" index="3" status="BS" value="2.9166666666666661" reducedCost="0"/>
 </variables>
 <objectiveValues>
  <objective index="0" name="obj" value="125.20833333333334"/>
 </objectiveValues>
</CPLEXSolution>
```

Some theorical elements

Mixed canonical form of LP

- Maximization,
- All variables are non-negative,
- Both inequality '≤' and equality '=' constraints.

$$
\begin{aligned}
&Max \quad f(x_1,\dots,x_n) = c_1x_1 + \cdots + c_nx_n = \sum_{j=1}^{n} c_jx_j \\
&s.t. \quad \sum_{j=1}^{n} a_{ij}x_j \le b_i, \forall i \in I_1 \\
&\qquad \sum_{j=1}^{n} a_{ij}x_j = b_i, \forall i \in I_2 \\
&\qquad x_j \ge 0, \forall j \in \{1,\dots,n\}
\end{aligned}
$$

Pure canonical form of LP

- Maximization,
- All variables are non-negative,
- All constraints are inequalities of the type '≤'.

$$\max z = \sum_j c_j x_j$$
$$\begin{aligned} s.t. \quad \sum_j a_{ij} x_j &\leq b_i \qquad i = 1, 2 \ldots m \\ x_j &\geq 0 \qquad j = 1, 2 \ldots n \end{aligned}$$

- Note that in this form, **no equality** constraints: $I_2 = \emptyset$

- In matrix form:

$$\begin{aligned} \max z = \; & \boldsymbol{c}^T \boldsymbol{x} \\ s.t. \; & A\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}. \end{aligned}$$

Where $x, c \in \mathbb{R}^n, b \in \mathbb{R}^m$, A is $m \times n$ real matrix and 0 is the vector 0_n.

Note:
Minimization → Constraints type '≥'

Standard form of PL

- Maximization,
- All variables are non-negative,
- All constraints are equations.

$$\max z = \textstyle\sum_j c_j x_j$$

$$\begin{array}{rcccl} s.t. & \sum_j a_{ij} x_j & = & b_i & i = 1, 2 \ldots m \\ & x_j & \geq & 0 & j = 1, 2 \ldots n \end{array}$$

- Note that in this form, **no inequality** constraints: $I_1 = \emptyset$

- In a matrix form:

$$\begin{array}{rl} \max z = & \boldsymbol{c}^T \boldsymbol{x} \\ s.t. & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}. \end{array}$$

Equivalence of forms

These three forms are equivalent: starting from a feasible solution of a program in one of the forms, it is easy to construct a feasible solution for the other two forms, yielding the same value for the objective function.

Transition between forms

- Any constraint " $\geq$ " can be written as a constraint " $\leq$ " : $\sum_{i=1}^{n} a_i x_i \geq b \equiv \sum_{i=1}^{n} -a_i x_i \leq -b$

- Any equality constraint can be written as two inequalities: $\sum_{i=1}^{n} a_i x_i = b \equiv \begin{cases} \sum_{i=1}^{n} a_i x_i \leq b \\ \sum_{i=1}^{n} a_i x_i \geq b \end{cases}$

- We solve the maximization (resp. minimization) problem by changing the signs of coefficients in the objective. The optimal value of the minimization (resp. maximization) problem is the negation (opposite) of that of the maximization (resp. minimization) problem.

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

Inequality $\rightarrow$ Equality: Add a slack/surplus variable

- Transformation of inequalities into equalities:
- If $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$, we have two cases:
  - $b_i \geq 0$: we add a non-negative slack variable x_{n+i}: $\sum_{j=1}^{n} a_{ij}x_j + x_{n+i} = b_i$
  - $b_i < 0$: we multiply the inequality by -1 to reduce it to the case developed below.

- If $\sum_{j=1}^{n} a_{ij}x_j \geq b_i$, there are two possible cases:
  - $b_i \leq 0$: we multiply the inequality by -1 to reduce it to the case developed above.
  - $b_i > 0$: we subtract a positive surplus variable x_{0i}: $\sum_{j=1}^{n} a_{ij}x_j - x_{0i} = b_i$

    This brings us back to the case of adding artificial variables.

Unrestricted variable → Non-negative variable

- Unrestricted variables: variables with arbitrary values ($\in \mathbb{R}$).
- If a variable x_j can take negative values, we introduce two variables $x_j^+ \geq 0$ and $x_j^- \geq 0$. We then set:

$$x_j = x_j^+ - x_j^-$$

- Another possible approach, if $x_j \geq r_j$, where r_j is a negative constant, is to set:

$$x_j^+ = x_j - r_j \geq 0$$

Transition to standard form

- We can add slack variables: $\sum_{j=1}^{n} a_{ij}x_j \leq b_i \Leftrightarrow \sum_{j=1}^{n} a_{ij}x_j + e_i = b_i \, , e_i \geq 0$

- We are working in a higher-dimensional space, but all the constraints are equalities.
- Easier algebraic manipulations.
- Standard PL:

$$\begin{cases} Max & z(x) = cx \\ s.t. & Ax = b \\ & x \geq 0 \end{cases}$$

Note: Every linear program in standard form can be equivalently transformed into a linear program in pure canonical form, and vice versa.

Example

- For the following LP, write :
  - The pure canonical form
  - The standard form

$$
\begin{aligned}
&Max \quad 3x_1 + 2x_2 + 4x_3 \\
&s.t. \\
&\qquad x_1 + x_2 + 2x_3 \leq 4 \\
&\qquad -2x_1 - 3x_3 \geq -5 \\
&\qquad 2x_1 + x_2 + 3x_3 \leq 7 \\
&\qquad x_1, x_2, x_3 \geq 0
\end{aligned}
$$

Duality concept

Duality

- We now approach the fundamental theoretical property of linear programming, which has countless applications. This property is known as duality.
- We can associate another linear program with any linear program, known as its dual. The first program is called the primal program.
- During the transformation, the variables of the primal become the constraints of the dual, and the constraints of the primal become the variables of the dual.
- In certain circumstances, it is more advantageous to solve the dual problem. Subsequently, the solution to the initial LP can be easily deduced through the solution of the dual.
- One of the most fruitful consequences of the concept of duality is the dual simplex algorithm, which was developed in the 1950s by Lemke. This algorithm is of great significance in linear programming.

$$\underset{x \in \mathbb{R}^n}{(\mathcal{P})} \begin{cases} Min & z = c^T x \\ s.t. & Ax \geq b \\ & x \geq 0 \end{cases}$$

$$\underset{y \in \mathbb{R}^m}{(\mathcal{D})} \begin{cases} Max & w = b^T y \\ s.t. & A^T y \leq c \\ & y \geq 0 \end{cases}$$

Duality

$$\begin{cases} \textbf{Min} \quad z = \sum_{j=1}^{n} c_j x_j \\ s.t. \quad \sum_{j=1}^{n} a_{ij} x_j \geq b_i \quad \forall i \in \{1, \dots, m\} \\ \qquad x_j \geq 0, \qquad \forall j \in \{1, \dots, n\} \end{cases}$$

$$(\mathcal{P}) \quad x \in \mathbb{R}^n \quad \begin{cases} Min & z = c^T x \\ s.t. & Ax \geq b \\ & x \geq 0 \end{cases}$$

$$\begin{cases} \textbf{Max} \quad w = \sum_{i=1}^{m} b_i y_i \\ s.t. \quad \sum_{i=1}^{m} a_{ji} y_i \leq c_j \quad \forall j \in \{1, \dots, n\} \\ \qquad y_i \geq 0, \qquad \forall i \in \{1, \dots, m\} \end{cases}$$

$$(\mathcal{D}) \quad y \in \mathbb{R}^m \quad \begin{cases} Max & w = b^T y \\ s.t. & A^T y \leq c \\ & y \geq 0 \end{cases}$$

The strong duality theorem

- In reality, we have even more: in general, for linear programming, the optimal values of the primal and dual problems coincide. More precisely:

- If the primal and dual LPs have feasible solutions, their respective optimal solutions x^* and y^* satisfy $c^T x^* = b^T y^*$.
- In other words, if the primal (dual) problem has a finite optimal solution, then so does the dual (primal) problem, and these two values are equal.

- For the curious minds among you: for the proof, you can use a classic lemma in geometry—Farkas' lemma.

Transition from primal to dual

- Each constraint in the primal becomes a variable in the dual,
- Each variable in the primal becomes a constraint in the dual,
- Reverse the coefficients of the objective function and the RHS of constraints,
- Use the transpose of the constraint matrix,
- Change the direction of inequalities in the constraints.

Duality *recipe* in all its forms

Maximize/Minimize Z	Minimize/Maximize W
Constraints matrix (m, n)	Transpose of the constraint matrix (n, m)
Right-side vector of constraints	Coefficients of variables in objective function
Coefficients of variables in objective function	Right-side vector of constraints
Number of variables n	Number of constraints n
Number of constraints m	Number of variables m
i^{th} constraint of type $\geq$ i^{th} constraint of type $\leq$ i^{th} constraint of type $=$	i^{th} variable of type ≤ 0 i^{th} variable of type ≥ 0 i^{th} variable of type 'unconstrained' $\in \mathbb{R}$
j^{th} variable of type ≤ 0 j^{th} variable of type ≥ 0 j^{th} variable of type 'unconstrained' $\in \mathbb{R}$	j^{th} constraint of type $\leq$ j^{th} constraint of type $\geq$ j^{th} constraint of type $=$

Proposition. The dual of the dual is equivalent to the primal.

Some applications… (1)

- Give the dual problem of the following LPs:

$$\begin{aligned} Max\ & \tfrac{1}{2} x_1 + x_2 \\ s.t.\ & x_1 + x_2 \leq 3 \\ & -x_1 + x_2 \leq 1 \\ & x_1 \leq 2 \\ & x_1 \geq 0,\ x_2 \geq 0 \end{aligned}$$

$$\begin{aligned} Min\ & -x_1 + x_2 \\ s.t.\ & 2x_1 - x_2 \geq 2 \\ & -x_1 + 2x_2 \geq -2 \\ & x_1 + x_2 \leq 5 \\ & x_1 \geq 0,\ x_2 \geq 0 \end{aligned}$$

$$\begin{aligned} Max\ & 2x_1 - x_2 \\ s.t.\ & x_1 - 2x_2 \leq 2 \\ & x_1 + x_2 = 6 \\ & x_2 \leq 5 \\ & x_1 \in IR,\ x_2 \in IR \end{aligned}$$

$$\begin{aligned} Max\ & z = 100x_1 + 200x_2 \\ s.t.\ & x_1 + x_2 \leq 150 \\ & 4x_1 + 2x_2 \leq 440 \\ & x_1 + 4x_2 \leq 480 \\ & x_1 \leq 90 \\ & x_1 \geq 0\ ,\ x_2 \geq 0 \end{aligned}$$

$$\begin{aligned} Max\ & 2x_1 - x_2 \\ s.t.\ & x_1 - x_2 = 3 \\ & x_1 \leq 4 \\ & x_1 \geq 0,\ x_2 \geq 0 \end{aligned}$$

Some applications… (2)

- Use the graphical method to solve the following LP:

$$
\begin{aligned}
Min \quad & z = -x_1 - x_2 \\
& -x_1 - 3x_2 \geq -3 \\
& -2x_1 - x_2 \geq -2 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

- Write the matrix form.
- Give the associated dual problem.
- Using the same method, confirm the strong duality theorem result.

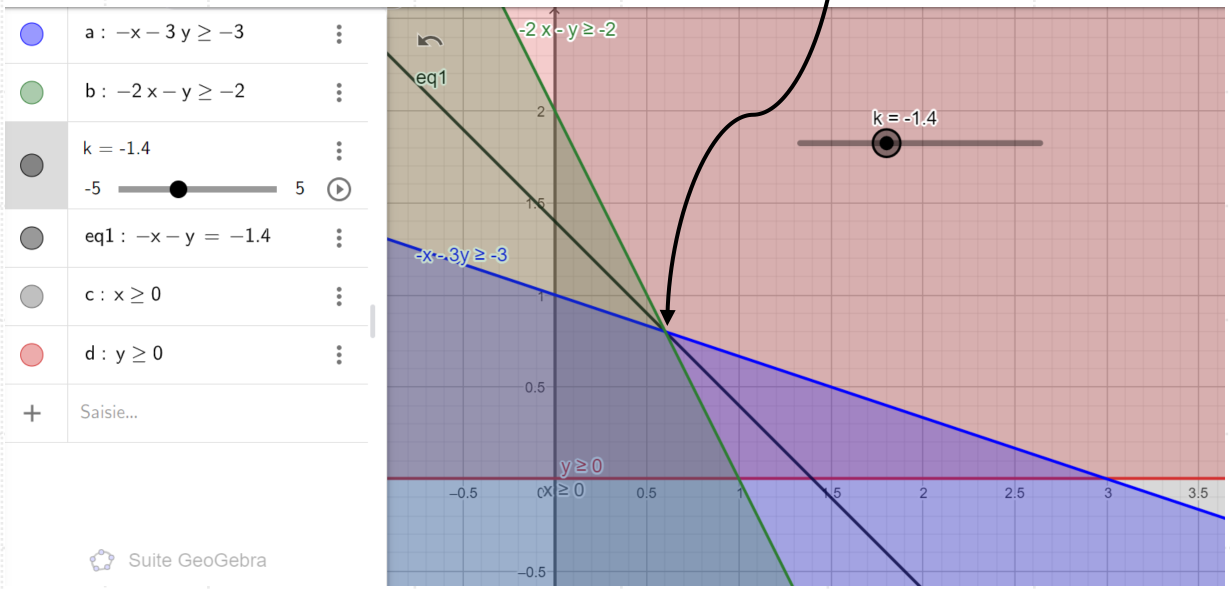
Some applications… (2)

Overview of the solution

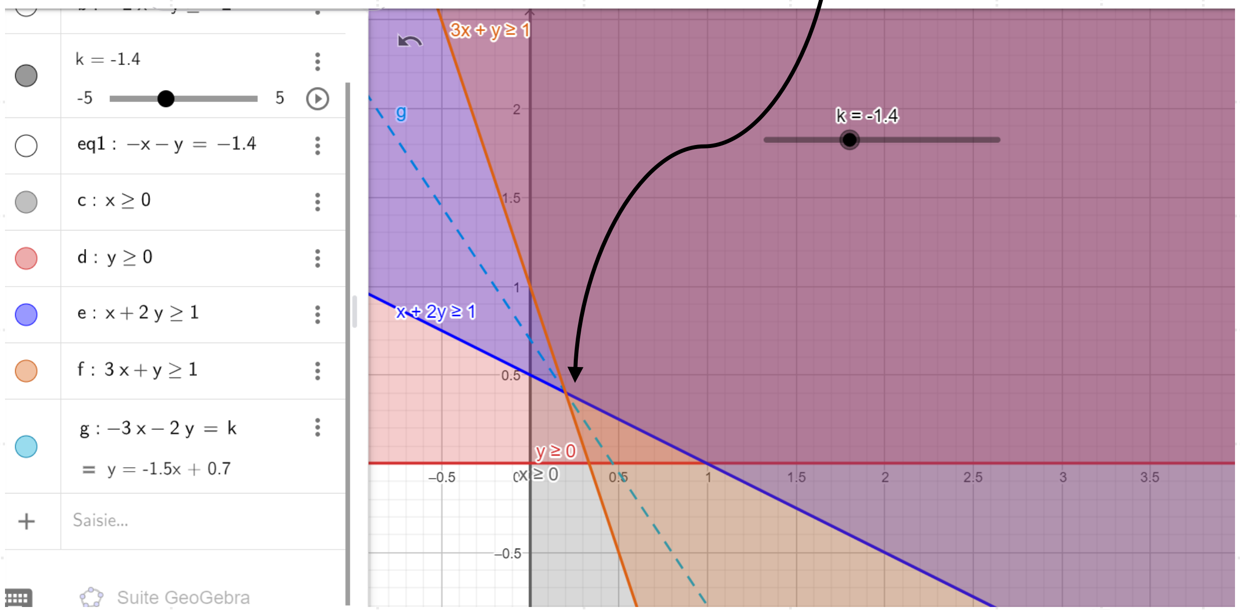
The optimal solution $\left(\frac{3}{5}, \frac{4}{5}\right)$ with optimal value $z = -\frac{7}{5}$

The optimal solution $\left(\frac{1}{5}, \frac{2}{5}\right)$ with optimal value $w = -\frac{7}{5}$

The primal (P)



The dual (D)



A very

Brief overview of LP resolution methods

This section is optional; it is for the curious minds among you ;).

Brief overview

Practical Plan: Many software programs (both open-source and commercial) can solve large-scale linear programming problems. Therefore, when a problem is formulated as a linear program, there's no need to write an algorithm to solve it from scratch. There are primarily two algorithms used in these software programs: the simplex algorithm and the interior point algorithm. It's worth noting that these algorithms come in various variants, so we can refer to them as algorithm families.

- Some examples of linear programming software include CPLEX, XPRESS, MINOS, GLPK, LPSOLVE, CLP, SCIP, SOPLEX, and even Excel has its implementation of the simplex algorithm.
- For detailed descriptions of these algorithms, you can refer to various books and resources, such as:
  - J. Matousek and B. Gärtner, Understanding and using linear programming, Sringer, Berlin Heidelberg New York, 2007.
  - Ficken, F. A. (2015). *The simplex method of linear programming*. Courier Dover Publications.
  - T. Terlaky, An easy way to teach interior-point methods, European Journal of Operational Research 130 (2001), 119.

Brief overview

Practical Plan: Many software programs (both open-source and commercial) can solve large-scale linear programming problems. Therefore, when a problem is formulated as a linear program, there's no need to write an algorithm to solve it from scratch. There are primarily two algorithms used in these software programs: the simplex algorithm and the interior point algorithm. It's worth noting that these algorithms come in various variants, so we can refer to them as algorithm families.

Theoretical Plan: Linear programming belongs to the class P, meaning there exist polynomial-time algorithms to solve linear programming problems. The interior point algorithm is one such polynomial-time algorithm. On the other hand, while the simplex algorithm is very efficient in practice, it doesn't have a known polynomial-time version. For each of its variants, there are instances where it requires an exponential number of steps to solve.

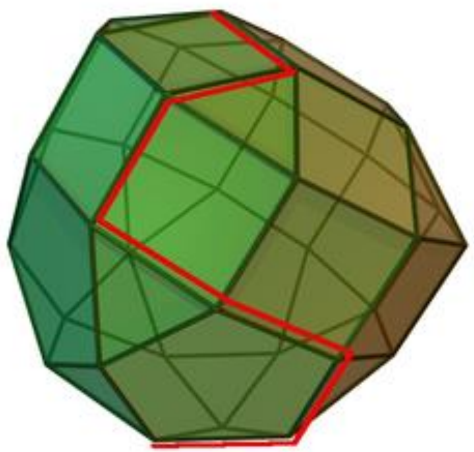
- (Note: "P" refers to the complexity class of decision problems that can be solved in polynomial time.)

Brief overview

Simplex algorithm (Dantzig, 1947) :

- Developed just after World War II,
- Used to plan the Berlin Airlift in 1948,
- It systematically searches the extreme points until the values of the objective function cannot be improved.

- The main steps of the algorithm are as follows:
  - The search starts from an extreme point.
  - It moves (pivots) to a neighboring extreme point with a better value according to the objective function by sliding along an edge of the polyhedron.
  - This process is repeated until the optimal solution is obtained.



Brief overview

Interior Points (Karmarkar, 1984):

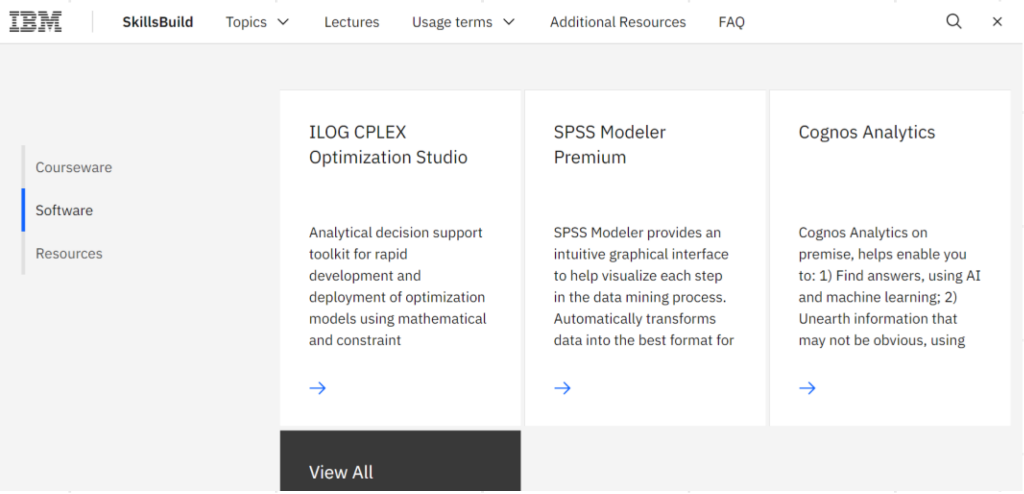
- In contrast to the simplex algorithm, the interior point algorithm aims to avoid the edges of the polyhedron and stay as much as possible inside it. It's only in the last step that the algorithm reaches the edge, precisely at an optimal solution.

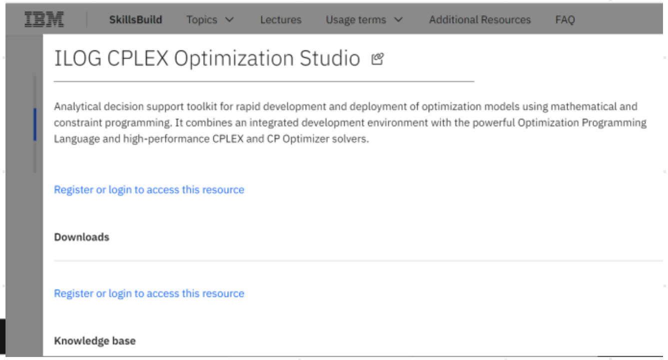
- In practice, interior point methods perform better when the space is smoother, whereas the simplex method appears more efficient when the edges are more pronounced.

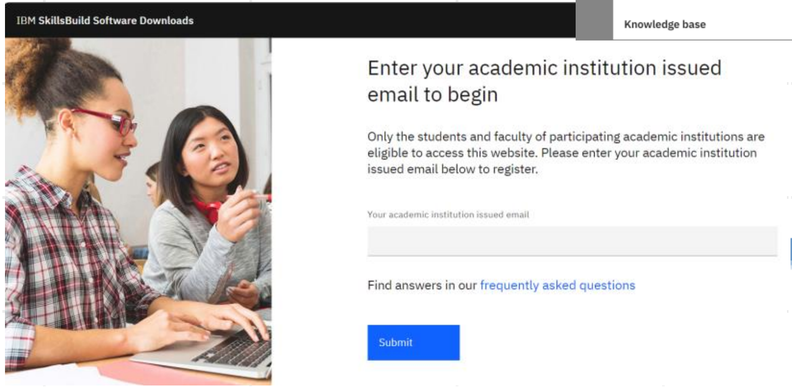
Try a good solver:

How to install ILOG CPLEX Optimization Studio

- For quick access to CPLEX Optimization Studio through the Academic Initiative (AI) program, go to IBM SkillsBuild Software Downloads. Click on Software, then you'll find, in the ILOG CPLEX Optimization Studio card, a link to register. Once your registration is accepted, you will see a link to download of the AI version using JUNIA email address.
- Some useful links:
  - How To Install CPLEX with IBM Academic Initiative? | Mert Bakır (mertbakir.gitlab.io)
  - Installing CPLEX Optimization Studio – IBM Documentation
  - IBM Academic Initiative
  - Use CPLEX® Optimizers – IBM Documentation to get some help (ILOG CPLEX Interactive Optimizer)







Return to keywords

- Linear Program; Feasible region; Feasible solution; Optimal solution; Optimal objective value; Standard form; Canonical form; Slack variables; .lp files; Primal program; Dual program.

- Provide a concise explanation and suggest a visual diagram that incorporates the majority of the mentioned keywords.