Linear Programming (LP)

Prof. Cesar de Prada
ISA-UVA
prada@autom.uva.es

Outline

- Introduction, LP problems
- Geometric solutions
- Definitions
- Simplex algorithm
- Dual problem
- Sensibility of the solucions
- LP typical problems
- Software

Introduction

- In most of the practical problems, the decision variables can not be chosen freely, but they must complain with a set of constraints expressed as equality or inequality equations.
- When the cost function and the constraint equations are linear in the decision variables, the optimization problem is called linear programming (LP)
- The term mathematical programming is related to the techniques developed during the II World War with the purpose of optimizing the planning (programming) of flights of military planes.

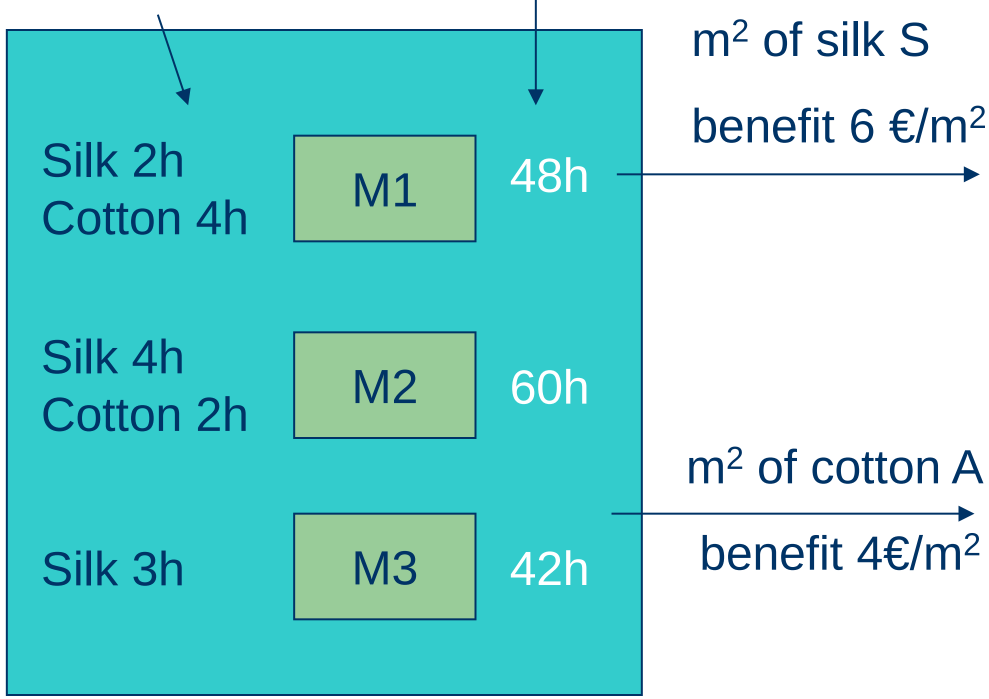
Example

Every m^2 of silk need to be processed in M!, M2, M3. The same with cotton in M1, M2

(teal box)	Factory
Mi	Machine i

Processing time of every m^2 (h)

availability in h per week of every machine (h)



How many m^2 of each type must be manufactured weekly in order to achieve the maximum benefit?

x_1 m^2 of silk per week

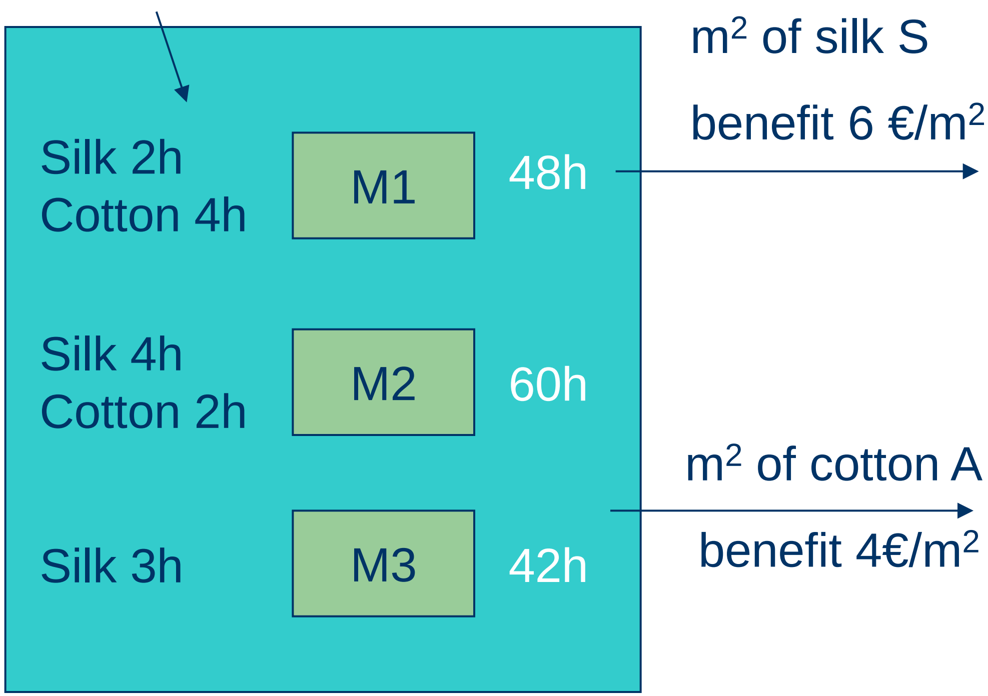
x_2 m^2 of cotton per week

Cesar de Prada ISA-UVA

Example

Processing time of every m² (h)

availability in h per week of every machine (h)



$$\max 6x_1 + 4x_2$$

$$\text{sujeto a:}$$

$$2x_1 + 4x_2 \le 48$$

$$4x_1 + 2x_2 \le 60$$

$$3x_1 \le 42$$

$$x_1 \ge 0$$

$$x_2 \ge 0$$

LP Problems

$$\max 6x_1 + 4x_2$$

under:

$$2x_1 + 4x_2 \le 48$$
$$4x_1 + 2x_2 \le 60$$
$$3x_1 \le 42$$
$$x_1 \ge 0$$
$$x_2 \ge 0$$

$$\Rightarrow \quad \max_x (6,4)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 4 \\ 4 & 2 \\ 3 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \le \begin{pmatrix} 48 \\ 60 \\ 42 \end{pmatrix}$$

$$\mathbf{x} \ge 0$$

$$\Rightarrow \quad \max_{\mathbf{x}} \mathbf{c}'\mathbf{x} \quad \mathbf{Ax} \le \mathbf{b} \quad \mathbf{x} \ge \mathbf{0}$$

The same LP problem can be formulated in several equivalent formats

$$\min_{\mathbf{x}} \mathbf{c}'\mathbf{x} \quad \mathbf{Ax} \le \mathbf{b} \quad \mathbf{x} \ge 0$$

Standard format

$$\max_{\mathbf{x}} \mathbf{c}'\mathbf{x} \quad \mathbf{Ax} = \mathbf{b} \quad \mathbf{x} \ge 0$$

$$\max_{\mathbf{x}} \mathbf{c}'\mathbf{x} \quad \mathbf{Ax} \ge \mathbf{b} \quad \mathbf{x} \ge 0$$

…

LP transformations

Maximize / minimize $$\max_{\mathbf{x}} \mathbf{c}'\mathbf{x} = \min_{\mathbf{x}} (-\mathbf{c}'\mathbf{x})$$

Inequalities $$3x_1 + 5x_2 \le 7 \quad \Rightarrow \quad -3x_1 - 5x_2 \ge -7$$

Equalities / Inequalities

$$3x_1 + 5x_2 \le 7 \quad \Rightarrow \quad \begin{cases} 3x_1 + 5x_2 + \varepsilon = 7 \\ \varepsilon \ge 0 \end{cases}$$

$$3x_1 + 5x_2 = 7 \quad \Rightarrow \quad \begin{cases} 3x_1 + 5x_2 - \varepsilon \le 7 \\ \varepsilon \ge 0 \end{cases}$$

A new (slack) variable ε is added

Unconstraint variable $$x_1 \quad \Rightarrow \quad x_1 = x_2 - x_3 \quad x_2 > 0,\ x_3 > 0$$

LP Transformations

$$\max_x (6,4)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \max_x 6x_1 + 4x_2 + 0x_3 + 0x_4 + 0x_5 = \max_{\mathbf{x}} \bar{\mathbf{c}}'\mathbf{x}$$

$$\begin{pmatrix} 2 & 4 \\ 4 & 2 \\ 3 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \le \begin{pmatrix} 48 \\ 60 \\ 42 \end{pmatrix} \Rightarrow \begin{matrix} 2x_1 + 4x_2 + x_3 = 48 \\ 4x_1 + 2x_2 + x_4 = 60 \\ 3x_1 + 0x_2 + x_5 = 42 \end{matrix} \Rightarrow \begin{pmatrix} 2 & 4 & 1 & 0 & 0 \\ 4 & 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 48 \\ 60 \\ 42 \end{pmatrix}$$

$$x_1 \ge 0 \quad x_2 \ge 0 \qquad x_3 \ge 0 \quad x_4 \ge 0 \quad x_5 \ge 0$$

$$\mathbf{x} \ge \mathbf{0}$$

$$\max_{\mathbf{x}} \mathbf{c}'\mathbf{x}$$

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

$$\mathbf{x} \ge \mathbf{0}, \quad \mathbf{b} \ge \mathbf{0}$$

The original problem is converted into standard form increasing the number of decision variables with the three slack variables x_3, x_4, x_5

LP Transformations

$$\max_{x} (3,2)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \max_{x} 3x_1 + 0x_3 + 2x_4 - 2x_5 = \max_{\mathbf{x}} \bar{\mathbf{c}}'\mathbf{x}$$

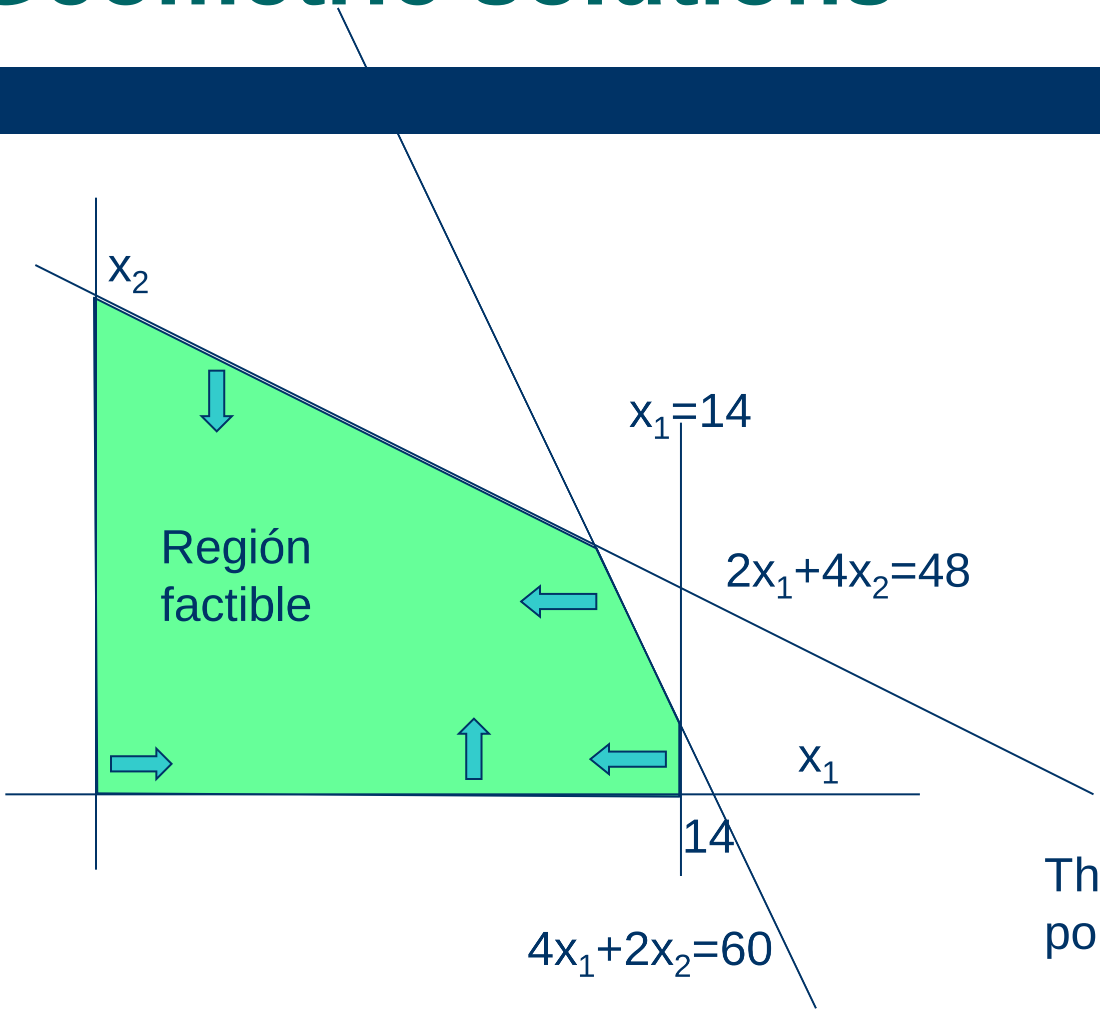
$$\left.\begin{array}{c} 2x_1 - 4x_2 \le -48 \\ 4x_1 + 2x_2 = 60 \\ x_1 \ge 0 \end{array}\right\} \Rightarrow \left.\begin{array}{c} 2x_1 - 4x_4 + x_3 + 4x_5 = -48 \\ 4x_1 + 2x_4 - 2x_5 = 60 \\ x_2 = x_4 - x_5 \quad \Rightarrow \end{array}\right\} \Rightarrow \left.\begin{array}{c} -2x_1 + 4x_4 - 4x_5 - x_3 = 48 \\ 4x_1 + 2x_4 - 2x_5 = 60 \\ (x_1, x_3, x_4, x_5)' \ge \mathbf{0} \end{array}\right\}$$

$$x_2 \text{ unconstraint} \qquad x_3 \ge 0 \quad x_4 \ge 0 \quad x_5 \ge 0$$

$$\begin{pmatrix} -2 & -1 & 4 & -4 \\ 4 & 0 & 2 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 48 \\ 60 \end{pmatrix}$$

$$\max_{\mathbf{x}} \mathbf{c}'\mathbf{x}$$
$$\mathbf{A}\mathbf{x} = \mathbf{b}$$
$$\mathbf{x} \ge \mathbf{0}, \quad \mathbf{b} \ge \mathbf{0}$$

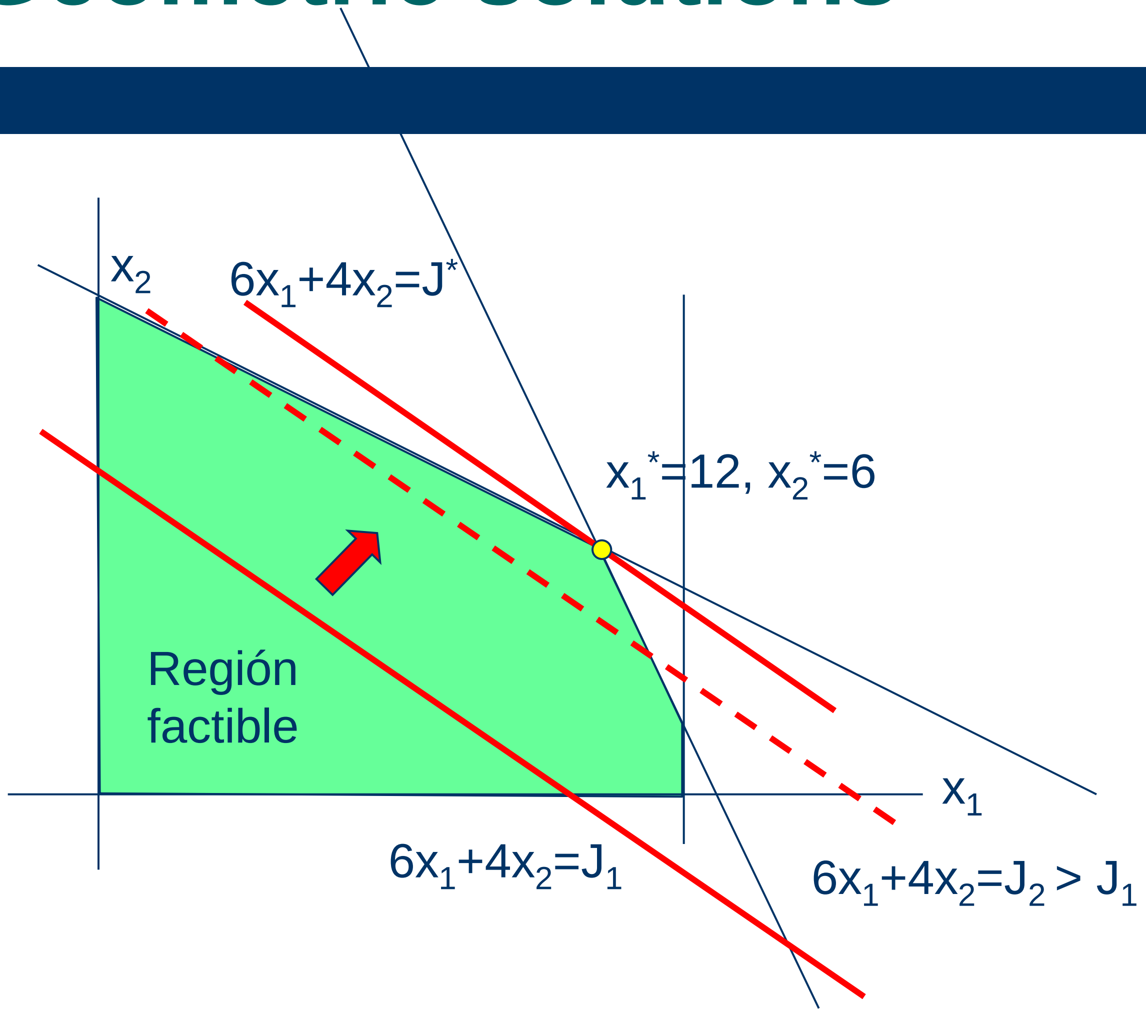
Geometric solutions



$$\max 6x_1 + 4x_2$$
$$\text{under:}$$
$$2x_1 + 4x_2 \leq 48$$
$$4x_1 + 2x_2 \leq 60$$
$$3x_1 \leq 42$$
$$x_1 \geq 0$$
$$x_2 \geq 0$$

The feasible region is a polytope

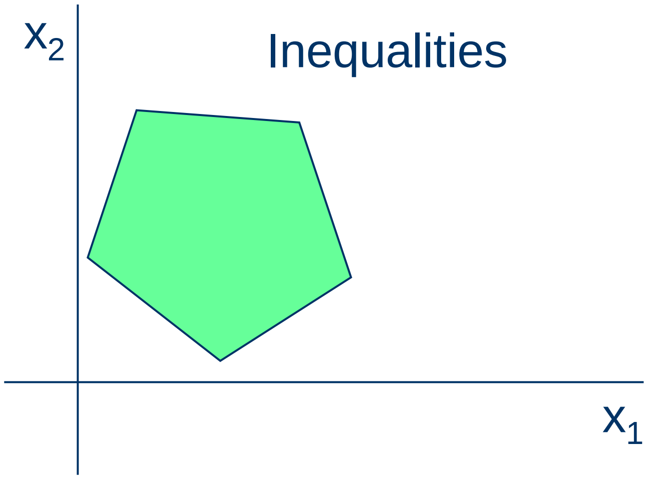
Geometric solutions



$$\max 6x_1 + 4x_2$$
$$\text{under:}$$
$$2x_1 + 4x_2 \le 48$$
$$4x_1 + 2x_2 \le 60$$
$$3x_1 \le 42$$
$$x_1 \ge 0$$
$$x_2 \ge 0$$

The solution is located in a vertex of the feasible region

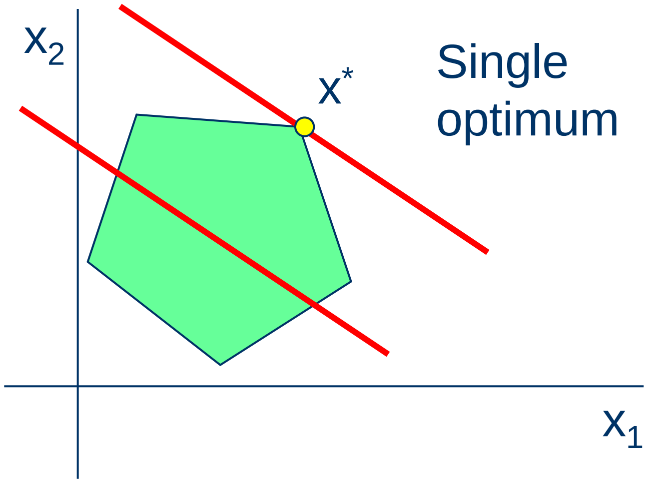
Feasible region

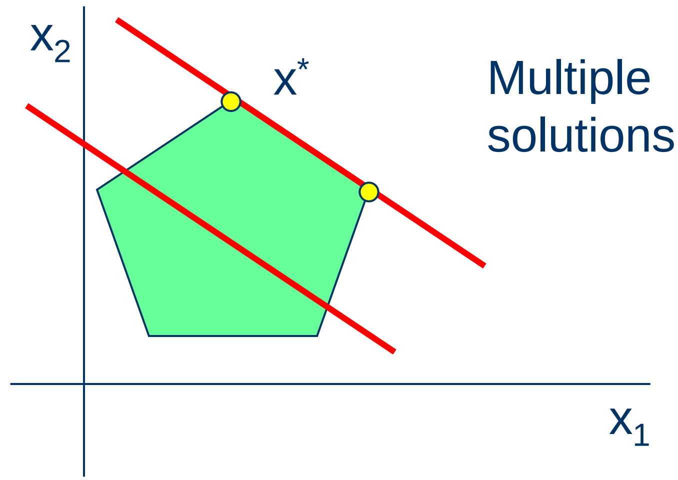


The feasible region is a polytope in R^n

$$\min_{\mathbf{x}} \mathbf{c}'\mathbf{x}$$
$$\mathbf{A}\mathbf{x} \leq \mathbf{b}$$
$$\mathbf{x} \geq 0$$

LP Problems

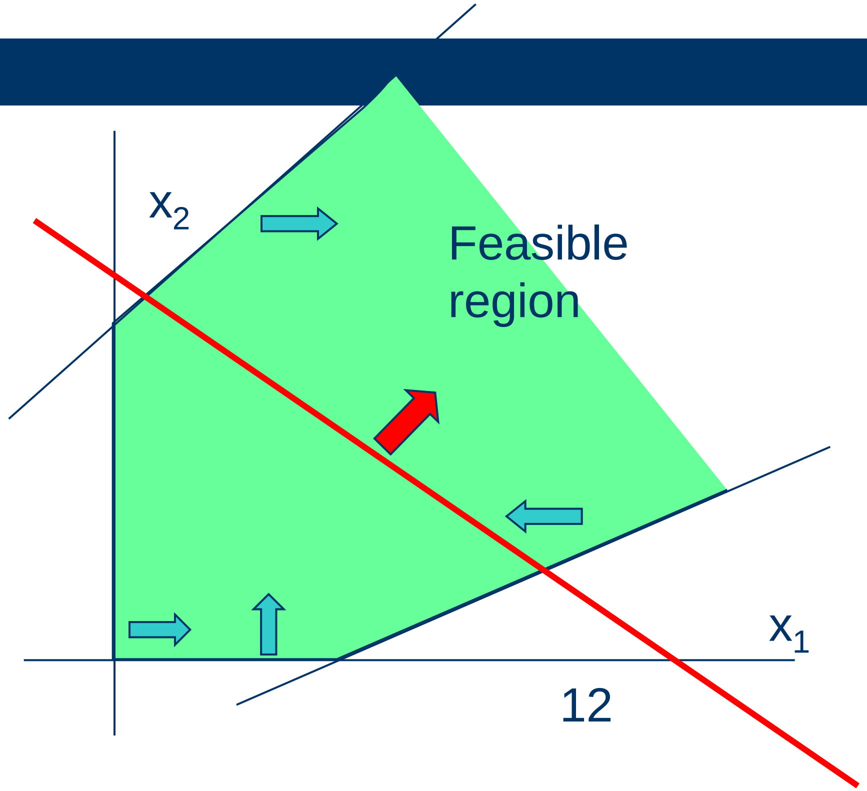




The feasible region is a polytope in R^n

The optimal solution, if it exist, is located in a vertex of the polytope

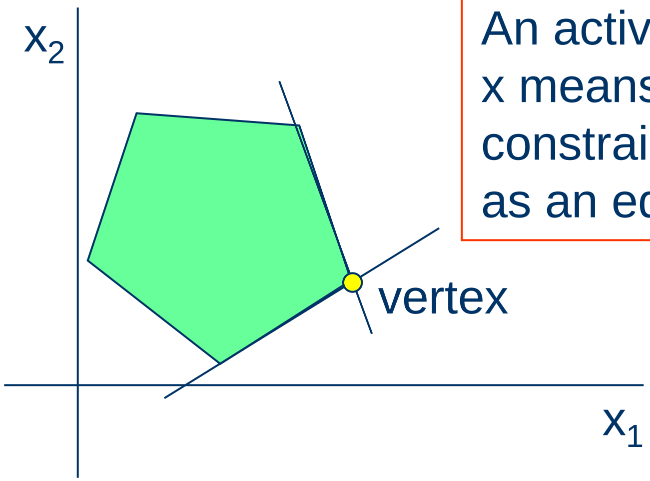
Constraints



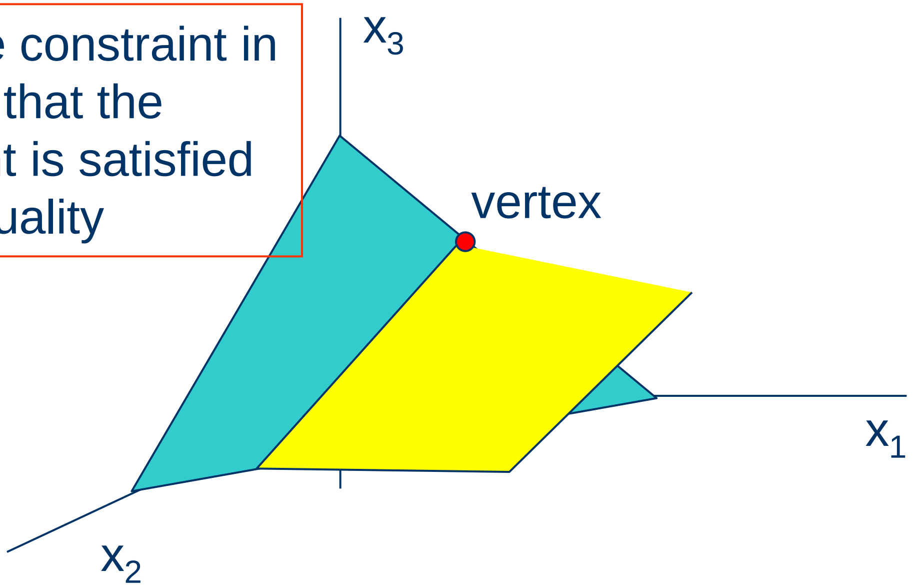
LP problem with unbounded cost function. There is no solution: The LP problem has no solution: there is no feasible **x** such that J(**x**) is greater than the value of the function at any other point

An LP problem may have also no solution because the feasible set is empty

Vertices



An active constraint in x means that the constraint is satisfied as an equality



If **x** is a vertex, it is placed in the intersection of two active and independent constraints

In R^n a vertex is defined as the common point of, at least, n independent and active constraints

Standard LP Problem

$$\max_{\mathbf{x}} J = \mathbf{c}'\mathbf{x}$$
$$\mathbf{A}\mathbf{x} = \mathbf{b}$$
$$\mathbf{x} \geq \mathbf{0}$$

A(m x n)

x(n x 1)

Rank(A)=m

n > m

b ≥ 0

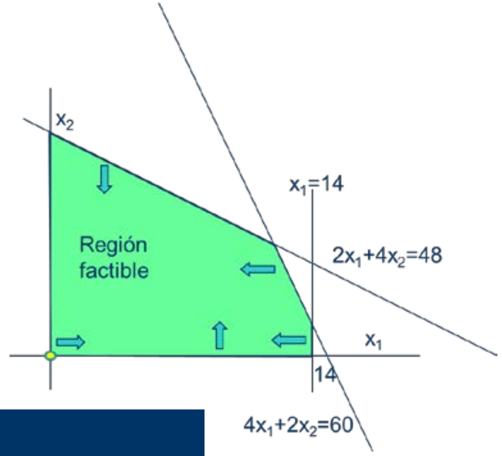
$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2$$
$$\ldots..$$
$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m$$

If n = m there is only a single solution and if n < m likely there will be no solution at all. So, the only case that is worth to consider is when n > m

A constraint that is linear combination of other ones is redundant and can be suppressed. This explains the condition rank(A) = m

The standard LP problem has n+m constraints, m are equality constraints and n are inequality ones. Notice that it is formulated as a maximization one.

Example



$$\max_{\mathbf{x}} J = 6x_1 + 4x_2 + 0x_3 + 0x_4 + 0x_5 = \max_{\mathbf{x}} J = \bar{\mathbf{c}}'\mathbf{x}$$

$$\begin{pmatrix} 2 & 4 & 1 & 0 & 0 \\ 4 & 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 48 \\ 60 \\ 42 \end{pmatrix} \Rightarrow \begin{aligned} 2x_1 + 4x_2 + x_3 \quad &= 48 \\ 4x_1 + 2x_2 + \quad x_4 &= 60 \\ 3x_1 + \quad\quad x_5 &= 42 \end{aligned}$$

$$\mathbf{x} \geq \mathbf{0}$$

The degrees of freedom of the problem are
n – m = 5 - 3 =2

Definitions

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2$$
$$\ldots..$$
$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m$$

A basic matrix is a set of m linearly independent columns of A.

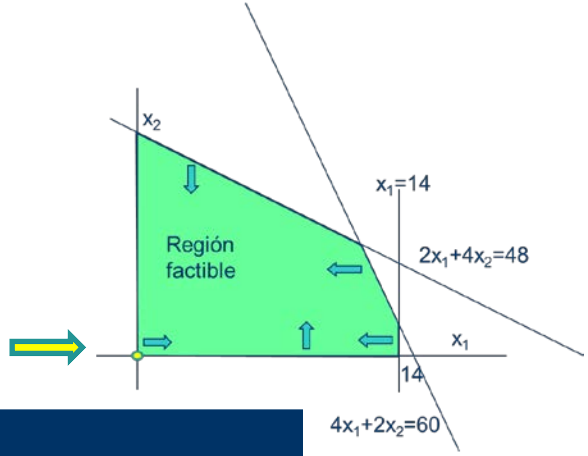
A basic variable is a decision variable associated to one of these columns.

As rank(A) = m it is always possible to find, at least, a basic matrix

A basic solution is a solution of A**x**= **b** that is obtained fixing the value of the n-m non basic variables to zero and solving the equation A**x** = **b** for the m basic variables.

A basic feasible solution is a basic solution that verifies all constraints

Example



$$\begin{pmatrix} 2 & 4 & 1 & 0 & 0 \\ 4 & 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 48 \\ 60 \\ 42 \end{pmatrix} \Rightarrow \begin{aligned} 2x_1 + 4x_2 + x_3 \qquad &= 48 \\ 4x_1 + 2x_2 + \quad x_4 \quad &= 60 \\ 3x_1 + \qquad\qquad x_5 &= 42 \end{aligned}$$

Basic matrix

$$\mathbf{x} \geq \mathbf{0}$$

$$\left.\begin{aligned} 2.\ 0 + 4.\ 0 + x_3 \qquad &= 48 \\ 4.\ 0 + 2.\ 0 + \quad x_4 \quad &= 60 \\ 3.\ 0 + \qquad\qquad x_5 &= 42 \end{aligned}\right\} \Rightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 48 \\ 60 \\ 42 \end{pmatrix}$$

Basic variables

Basic solution and also basic feasible solution

Vertices = Basic feasible solutions

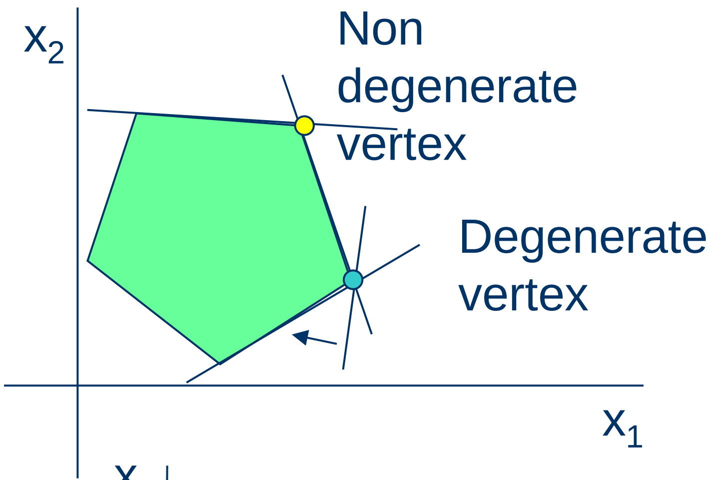
$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2$$
$$\ldots..$$
$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m$$

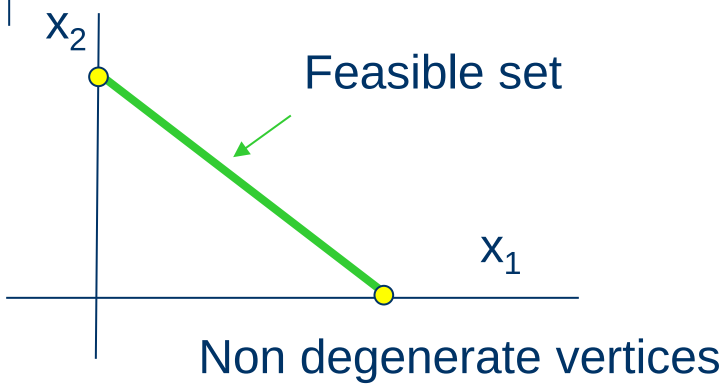
A vertex satisfies n independent and active constraints. If x is feasible, it must satisfy all equations. There is only m equality constraints, Ax = b, hence, it must satisfy too other n-m, $x_i \geq 0$ as equalities. So, x corresponds to a feasible basic solution.

A basic feasible solution x, verifies m equations Ax = b, so, it satisfies m active constraints, as well as n-m relations $x_i = 0$. Hence, it satisfies n – m + m = n constraints in active form and consequently is a vertex

Vertices= basic feasible solutions

Degenerate Vertices





A vertex x, or basic feasible solution, is non degenerate if it satisfies exactly n active constraints.

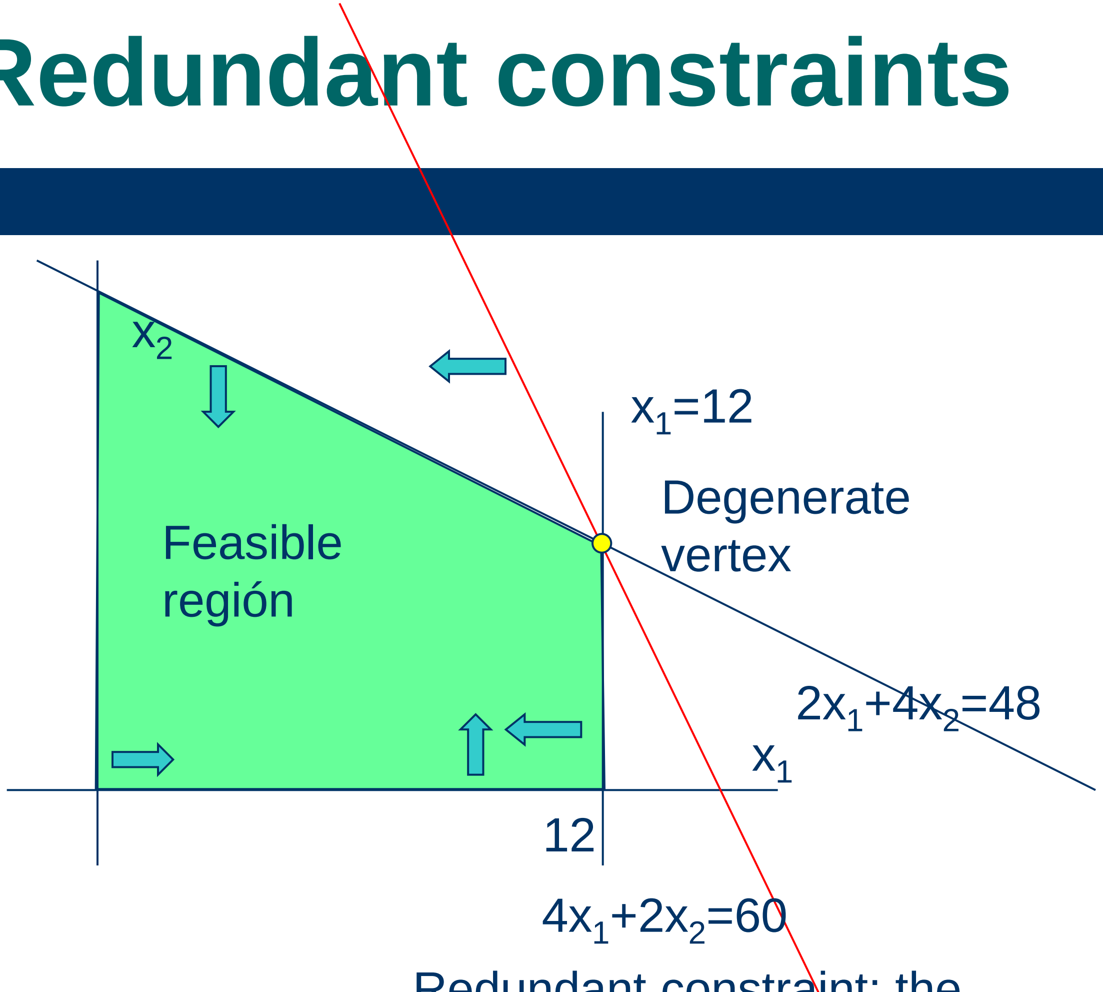
If it satisfies more than n active constraints it is called degenerate

$$\max\ 2x_1 + 3x_2$$

$$4x_1 + 5x_2 = 7$$

$$x_1 \geq 0, \quad x_2 \geq 0$$

Redundant constraints

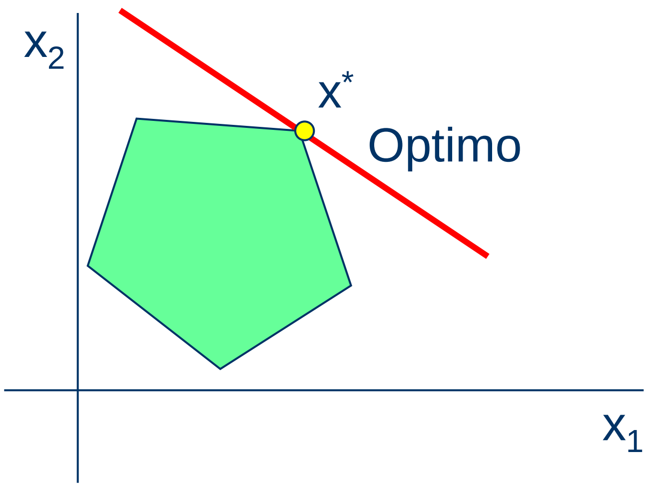


$$\max 6x_1 + 4x_2$$
$$\text{under:}$$
$$2x_1 + 4x_2 \le 48$$
$$4x_1 + 2x_2 \le 60$$
$$3x_1 \le 36$$
$$x_1 \ge 0$$
$$x_2 \ge 0$$

change

Redundant constraint: the feasible region doesn't change if the constraint is omitted

LP problem



If the solution is located in a vertex and a vertex is a basic feasible solution, one could think in a solution method that would evaluate J = c'x at each basic feasible solution and would choose that best one, if it exist.

The maximum number of vertices corresponds to the different groups of m columns that we can form using the n ones of A, that is:

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

If n is big, this can be an enormous number: e.g. for n=100, m=50 there are 10^{29} combinations!

The Simplex Algorithm (Dantzig 1947)

The Simplex algorithm is an intelligent way of travelling through the vertices of the feasible region such that

✓It founds a vertex

✓Check if it is optimum

✓If not, it moves to another neighbouring vertex having a better value of J

✓It also detects the absence of solution due to a unfeasible set or an unbounded cost function.

Hence, in a finite number of steps, the algorithm can found the optimum

It operates in two phases:

I Finds the initial vertex or detects that there is no solution

II Finds the optimum, or detects that the problem is unbounded

Phase I of the simplex algorithm

In its first step, the simplex algorithm transforms the original LP problem into the following canonical format:

$$x_1 \qquad\qquad + \bar{a}_{1,m+1} x_{m+1} + \ldots + \bar{a}_{1n} x_n = \bar{b}_1$$

$$x_2 \qquad\qquad + \bar{a}_{2,m+1} x_{m+1} + \ldots + \bar{a}_{2n} x_n = \bar{b}_2$$

$$\ldots\ldots$$

$$x_m + \bar{a}_{m,m+1} x_{m+1} + \ldots + \bar{a}_{mn} x_n = \bar{b}_m$$

where m variables (in the example the first m) appear only in an equation and with coefficient one, and also $\bar{b}_i \geq 0$

or concludes that there is no feasible solution

Transformation to canonical form

The conversion can be made choosing a base B. For simplicity, we will assume that it corresponds to the first m variables. (It is always possible to switch columns in order to place the selected variables in these positions). Then, one can operate as:

$$\mathbf{Ax} = \mathbf{b} \;\Rightarrow\; [\mathbf{B} \quad \mathbf{N}]\mathbf{x} = \mathbf{b} \;\Rightarrow \mathbf{B}^{-1}[\mathbf{B} \quad \mathbf{N}]\mathbf{x} = \mathbf{B}^{-1}\mathbf{b} \;\Rightarrow\; [\mathbf{I} \quad \mathbf{B}^{-1}\mathbf{N}]\mathbf{x} = \overline{\mathbf{b}}$$

So that the system is in the canonical form format.

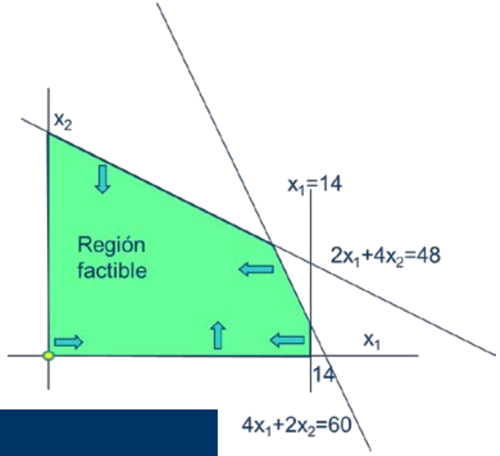
Alternatively, the Gauss-Jordan elimination can be used for the same purpose. In this context, the operations performed with linear combinations of the files are called pivot operations.

From this format, a right away solution is:

(This point is discussed later on)

$$(\overline{b}_1, \overline{b}_2, ..., \overline{b}_m, 0, ..., 0) \quad \text{if } \overline{b}_i \geq 0 \quad \text{it is feasible in addition}$$

Example



$$\begin{pmatrix} 2 & 4 & 1 & 0 & 0 \\ 4 & 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 48 \\ 60 \\ 42 \end{pmatrix} \Rightarrow \begin{array}{l} 2x_1 + 4x_2 + x_3 = 48 \\ 4x_1 + 2x_2 + x_4 = 60 \\ 3x_1 + x_5 = 42 \end{array}$$

Basic matrix

$$\mathbf{x} \geq \mathbf{0}$$

$$\left.\begin{array}{l} 2.\ 0 + 4.\ 0 + x_3 = 48 \\ 4.\ 0 + 2.\ 0 + x_4 = 60 \\ 3.\ 0 + x_5 = 42 \end{array}\right\} \Rightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 48 \\ 60 \\ 42 \end{pmatrix}$$

Basic variables

Basic feasible solution

$$\left.\begin{array}{ll} 2.\;0+4.\;0+x_3 & =48 \\ 4.\;0+2.\;0+\quad x_4 & =60 \\ 3.\;0+\qquad\quad x_5 & =42 \end{array}\right\} \Rightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 48 \\ 60 \\ 42 \end{pmatrix}$$

Variables básicas

Solución básica factible

Cesar

Simplex, phase II

$$\begin{array}{l} x_1 \qquad\qquad + \bar{a}_{1,m+1} x_{m+1} + \ldots + \bar{a}_{1n} x_n = \bar{b}_1 \\ \quad x_2 \qquad\quad + \bar{a}_{2,m+1} x_{m+1} + \ldots + \bar{a}_{2n} x_n = \bar{b}_2 \\ \qquad \ldots\ldots \\ \qquad\qquad x_m + \bar{a}_{m,m+1} x_{m+1} + \ldots + \bar{a}_{mn} x_n = \bar{b}_m \end{array}$$

Basic (or dependent)

Non-basic (or independent)

An initial vertex is generated easily:

$$\left.\begin{array}{ll} \text{basic} & x_i = \bar{b}_i \geq 0 \\ \text{non basic} & x_j = 0 \end{array}\right\} \begin{array}{l} i = 1,2,\ldots,m \\ j = m+1,\ldots,n \end{array}$$

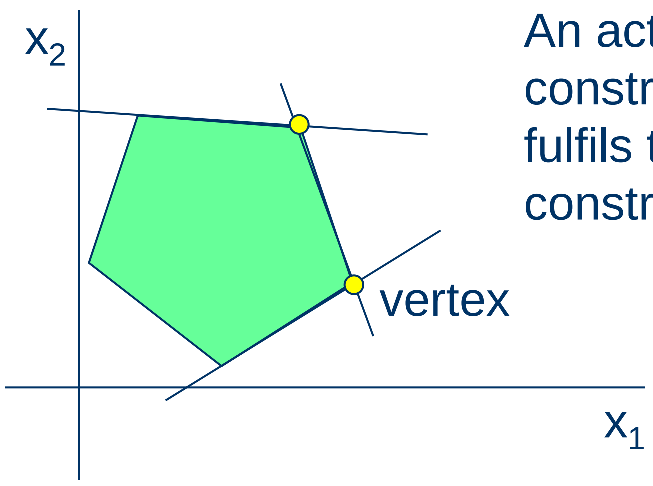
The set of basic variables is called a base x_B

$$\mathbf{x}_B = (x_1, x_2, \ldots, x_m)'$$

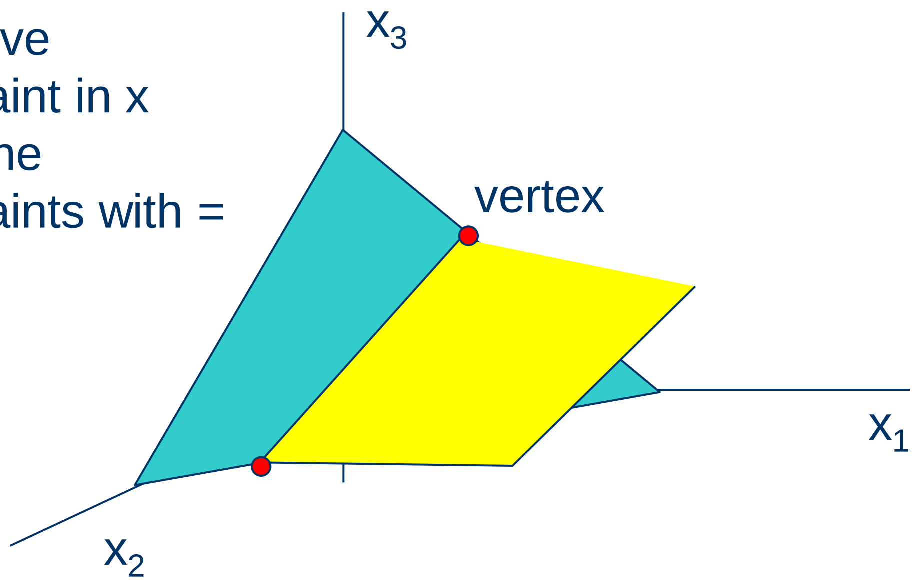
As the non-basic variables are 0, if $c_B=(c_1,c_2,\ldots,c_m)$ then:

$$J = \mathbf{c}_B' \mathbf{x}_B$$

Adjacent Vertex

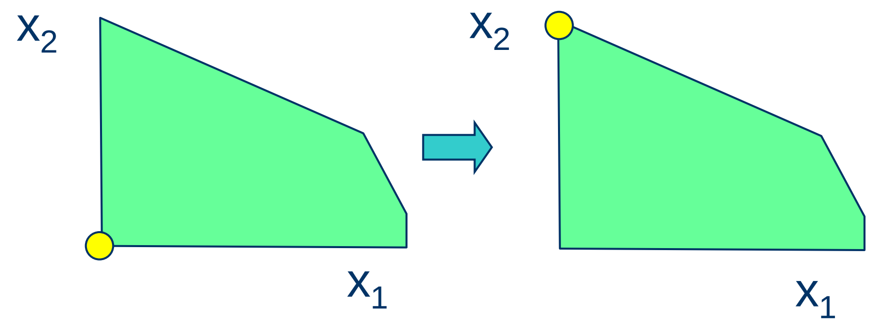


An active constraint in x fulfils the constraints with =



Adjacent vertices differ just in one constraint

Adjacent vertices



$$\begin{pmatrix}2&4&1&0&0\\4&2&0&1&0\\3&0&0&0&1\end{pmatrix}\begin{pmatrix}x_1\\x_2\\x_3\\x_4\\x_5\end{pmatrix}=\begin{pmatrix}48\\60\\42\end{pmatrix} \Rightarrow \begin{array}{l} 2x_1+4x_2+x_3 = 48\\ 4x_1+2x_2+x_4=60\\ 3x_1+x_5=42\end{array}$$

$\mathbf{x} \geq \mathbf{0}$

Basic matrix

$$\left.\begin{array}{l} 2.\ 0+4.\ x_2+0 = 48\\ 4.\ 0+2.\ x_2+x_4=60\\ 3.\ 0+x_5=42\end{array}\right\} \Rightarrow \begin{pmatrix}x_1\\x_2\\x_3\\x_4\\x_5\end{pmatrix}=\begin{pmatrix}0\\12\\0\\36\\42\end{pmatrix}$$

Basic variables

Basic feasible solution

The active constraint $x_2 = 0$ is changed by $x_3 = 0$ in order to generate an adjacent vertex. This means that x_3 moves out of the basis and x_2 comes in

Adjacent vertex

An adjacent vertex differs from its neighbours in just one active constraint.

In order to obtain an adjacent vertex, the simplex method moves one non-basic variable to the basis and swaps it with a basic variable.

The variables to be swapped are chosen so that the cost function improves as much as possible

$$
\begin{aligned}
x_1 \qquad\qquad & + \bar{a}_{1,m+1}x_{m+1} + \ldots + \bar{a}_{1n}x_n = \bar{b}_1 \\
x_2 \qquad & + \bar{a}_{2,m+1}x_{m+1} + \ldots + \bar{a}_{2n}x_n = \bar{b}_2 \\
\ldots\ldots & \\
x_m & + \bar{a}_{m,m+1}x_{m+1} + \ldots + \bar{a}_{mn}x_n = \bar{b}_m
\end{aligned}
$$

Assign the value zero to n-m variables and solve Ax=b for the others. Which are the best two variables to swap?

Adjacent vertex

$$\left.\begin{array}{l} x_1 \quad + \bar{a}_{1,m+1} x_{m+1} + \ldots + \bar{a}_{1s} x_s + \ldots + \bar{a}_{1n} x_n = \bar{b}_1 \\ \quad \ldots\ldots \\ \quad x_r \quad + \bar{a}_{r,m+1} x_{m+1} + \ldots + \bar{a}_{rs} x_s + \ldots + \bar{a}_{rn} x_n = \bar{b}_r \\ \qquad \ldots\ldots \\ \qquad x_m + \bar{a}_{m,m+1} x_{m+1} + \ldots + \bar{a}_{ms} x_s + \ldots + \bar{a}_{mn} x_n = \bar{b}_m \end{array}\right\}$$

Assume that we increase the value of a non-basic variable x_s from 0 to 1,maintaining the remaining ones in 0

$$\left.\begin{array}{l} x_1 \quad + \bar{a}_{1s} x_s = \bar{b}_1 \\ \quad \ldots\ldots \\ \quad x_r \quad + \bar{a}_{rs} x_s = \bar{b}_r \\ \qquad \ldots\ldots \\ \qquad x_m + \bar{a}_{1s} x_s = \bar{b}_m \end{array}\right\} \text{if } x_s = 1 \Rightarrow \begin{array}{c} x_i = \bar{b}_i - \bar{a}_{is} \quad i = 1,\ldots., m \\ x_s = 1 \\ x_j = 0 \quad j = m+1,\ldots, n \quad j \neq i \end{array}$$

Relative gain

$$\text{if } x_s = 1 \Rightarrow \left.\begin{array}{c} x_i = \bar{b}_i - \bar{a}_{is} \quad i = 1,\dots,m \\ x_s = 1 \\ x_j = 0 \quad j = m+1,\dots,n \quad j \neq i \end{array}\right\}$$

Of course, the change in x_s should be such that one basic variable becomes zero, making it non-basic in this way, but, for the moment, let's keep the value $x_s = 1$

The change in the cost function J would be:

$$\Delta J = \left(\sum_{i=1}^{m} c_i (\bar{b}_i - \bar{a}_{is}) + c_s \right) - \sum_{i=1}^{m} c_i \bar{b}_i = \boxed{c_s - \sum_{i=1}^{m} c_i \bar{a}_{is}} = c_s - \mathbf{c}_B' \mathbf{P}_s$$

If the relative gain (change of J per unit change in x_i) of a non-basic variable x_s is > 0, then J will improve when x_s is converted into a basic variable because its value will change from 0 to a positive one.

$\mathbf{P}_s$ non-basic column of A

Extremum conditions

If the relative gains of all non-basic variables are negative or zero, then all adjacent vertex to the current one have values of J lower than the current one and this vertex is a local minimum, but as the problem is convex (linear) the local optimum is also a global one.

$$c_s - \mathbf{c}_B{}'\mathbf{P}_s \leq 0 \quad s = m+1,\ldots,n$$

$$\mathbf{c_N}' - \mathbf{c}_B{}'\mathbf{B}^{-1}\mathbf{N} \leq \mathbf{0}'$$

If a non-basic variable x_s has a relative gain >0, then, J will improve if the value of this variable is increased by converting it into a basic variable.

Which is the non-basic variable that should be selected? The one with the highest relative gain.

How much should its value be increased? Which basic variable should be removed from the basis?

How much should its value be increased? Which basic variable should be removed from the basis? Rule of the lower ratio

If the value of a non-basic variable x_s is changed, the value of the basic variables changes to:

$$x_i = \bar{b}_i - \bar{a}_{is} x_s \quad i = 1,\dots,m \begin{cases} \text{if } \bar{a}_{is} < 0 & x_s \uparrow \Rightarrow x_i \uparrow \\ \text{if } \bar{a}_{is} = 0 & x_s \uparrow \Rightarrow x_i \text{ cte} \\ \text{if } \bar{a}_{is} > 0 & x_s \uparrow \Rightarrow x_i \downarrow \end{cases}$$

Notice that x_s only can change from 0 to a positive value

The maximum change in x_s that respects the constraint $x_i \geq 0$, for all basic variables is:

$$\min_{\bar{a}_{is} > 0} \left[\frac{\bar{b}_i}{\bar{a}_{is}} \right] = \frac{\bar{b}_r}{\bar{a}_{rs}}$$

When x_s is assigned the value b_r/a_{rs}, the basic variable x_r will became 0 and, hence, will be converted into a non-basic one, swapping roles with x_s. The change in J will be given by:

$$\bar{c}_s \left(\frac{\bar{b}_r}{\bar{a}_{rs}} \right) > 0$$

Unbounded solutions

If the value of a non-basic variable x_s is changed, the value of the basic variables changes to:

$$x_i = \bar{b}_i - \bar{a}_{is} x_s \qquad i = 1,\ldots,m \begin{cases} \text{si } \bar{a}_{is} < 0 & x_s \uparrow \quad \Rightarrow x_i \uparrow \\ \text{si } \bar{a}_{is} = 0 & x_s \uparrow \quad \Rightarrow x_i \; cte \\ \text{si } \bar{a}_{is} > 0 & x_s \uparrow \quad \Rightarrow x_i \downarrow \end{cases}$$

Notice that if all elements of the colum s verify:

$$\bar{a}_{is} \leq 0 \quad \forall i = 1,\ldots, m$$

Then the value of x_s can be increased as much as one wish without any risk that any basic variable x_i become negative. So, as J increases when x_s increases, there is no upper bound in J and the LP problem has no solution.

Degenerate solutions

If it happens that after computing a new basic solution, any of the basic variables is zero, then the vertex (or the basic feasible solution) is called degenerate.

They can appear in the initial vertex when a $\bar{b}_i$ is zero, or when computing a new vertex. In this case as

$$\min_{\bar{a}_{is}>0}\left[\frac{\bar{b}_i}{\bar{a}_{is}}\right]=\frac{\bar{b}_r}{\bar{a}_{rs}}=0$$

The increment in J will be:

$$\bar{c}_s\left(\frac{\bar{b}_r}{\bar{a}_{rs}}\right)=0$$

And no improvement will be achieved in this iteration. In theory, it may happens that after several changes without improvements, one returns to a previous visited vertex, creating a cycle and stopping the convergence of the algorithm. Nevertheless, in practice, this can be avoided in a well programmed algorithm.

Summary

1 Formulate the problem in canonical form, write the associated table and choose a basis.

2 For every non-basic variable, compute the relative gain and choose the variable with the highest one. If the relative gain is ≤ 0, the current vertex is optimum. If not, select this variable x_s to be moved to the basis. If all a_{is} are < 0, then there is no solution.

3 Compute the lowest of all ratios b_i/a_{is} ($a_{is}>0$) in order to select which basic variable is moved from the basis.

		c1	c2	c3	c4	c5			
		6	4	0	0	0			
c_B	Basis	x1	x2	x3	x4	x5	b		(ais>0) bi/ais
0	x3	2	4	1	0	0	48		24
0	x4	4	2	0	1	0	60		15
0	x5	3	0	0	0	1	42		14
c_s- sum($c_i a_{is}$)	Relative gain	6	4				0	J	

4 Pivot on the element of this file and row in order to formulate again the problem in canonical form and repeat the process

Example

Manufacture of silk and cotton

$$\max 6x_1 + 4x_2$$
$$\text{under:}$$
$$2x_1 + 4x_2 \le 48$$
$$4x_1 + 2x_2 \le 60$$
$$3x_1 \le 42$$
$$x_1 \ge 0$$
$$x_2 \ge 0$$

Standard canonical form

$$\max_{\mathbf{x}} J = 6x_1 + 4x_2 + 0x_3 + 0x_4 + 0x_5 = \max_{\mathbf{x}} J = \mathbf{c}'\mathbf{x}$$

$$\begin{pmatrix} 2 & 4 & 1 & 0 & 0 \\ 4 & 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 48 \\ 60 \\ 42 \end{pmatrix}$$

$$\mathbf{x} \ge \mathbf{0}$$

Excel

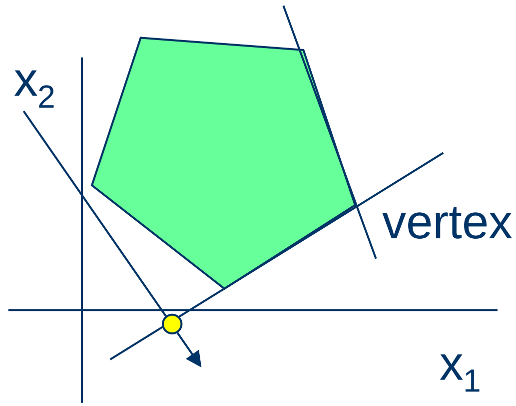
Phase I of the simplex algorithm

We have seen that selecting m independent columns of A, it is always possible to convert the LP problem to canonical form:

$$\mathbf{Ax} = \mathbf{b} \quad \Rightarrow \quad [\mathbf{B} \quad \mathbf{N}]\mathbf{x} = \mathbf{b} \quad \Rightarrow \mathbf{B}^{-1}[\mathbf{B} \quad \mathbf{N}]\mathbf{x} = \mathbf{B}^{-1}\mathbf{b} \quad \Rightarrow \quad [\mathbf{I} \quad \mathbf{B}^{-1}\mathbf{N}]\mathbf{x} = \bar{\mathbf{b}}$$

And find a basic solution: $(\bar{b}_1, \bar{b}_2, \ldots., \bar{b}_m, 0, \ldots, 0)$

if all $\bar{b}_i \geq 0$ A basic feasible solution has been found and Phase II can start



But if some b_i is < 0 the previous basic solution is not feasible and an alternative must be found

Phase I of the simplex algorithm

If some $b_i < 0$, then both sides of the corresponding equation can be multiplied by -1, so that all b_i will be positive. (but the solution will remain unfeasible). Next the following associated LP problem can be formulated and solved:

$$\max_{\mathbf{x},\lambda} -(1,1,1\ldots,1)\mathbf{v}$$

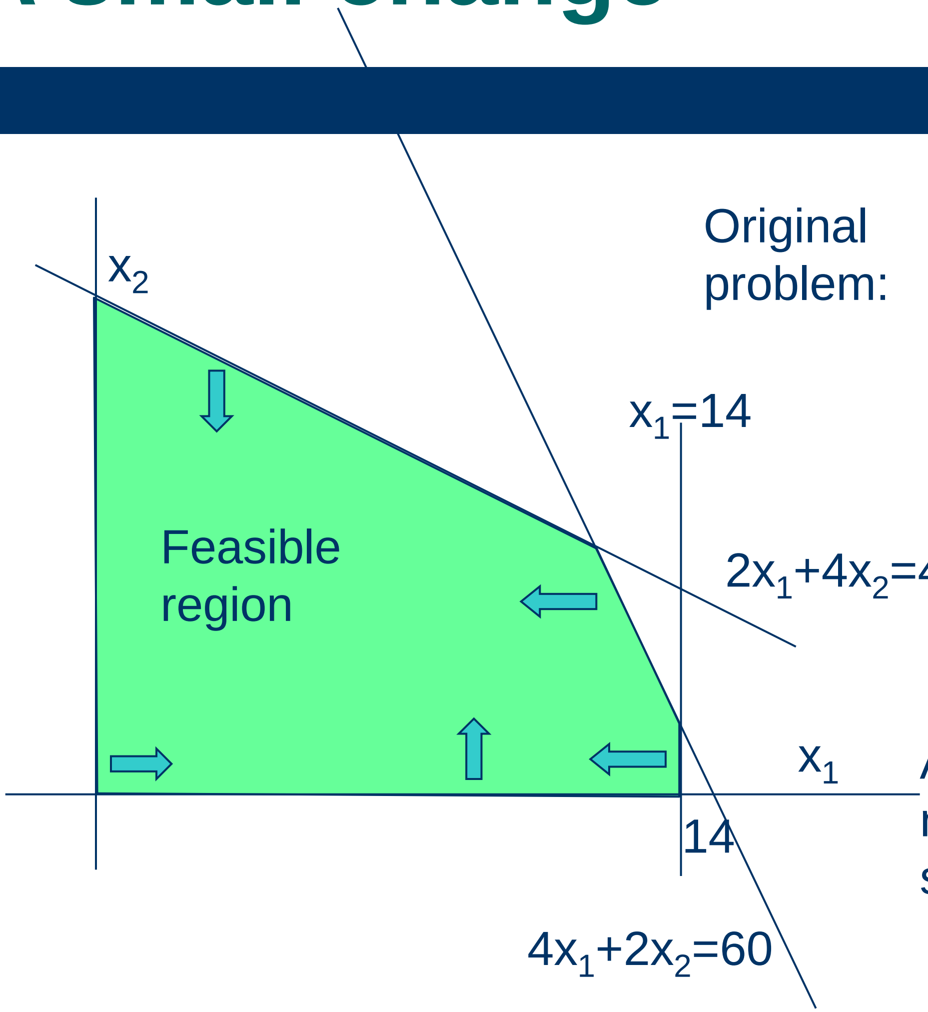
$$\begin{bmatrix} \mathbf{I} & \tilde{\mathbf{I}} & \tilde{\mathbf{M}} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{x} \end{bmatrix} = \boldsymbol{\beta} \qquad \mathbf{v} \geq \mathbf{0}, \quad \mathbf{x} \geq \mathbf{0}$$

If the original LP problem has a feasible solution x, then the associated LP will have a solution [0 , x]

where $\begin{bmatrix} \tilde{\mathbf{I}} & \tilde{\mathbf{M}} \end{bmatrix} \mathbf{x} = \boldsymbol{\beta}$ is $\begin{bmatrix} \mathbf{I} & \mathbf{B}^{-1}\mathbf{N} \end{bmatrix} \mathbf{x} = \bar{\mathbf{b}}$ after converting all $\bar{b}_i$ to positive #

Notice that the associated LP have always an initial basic feasible solution [ν,**x**]=(β,**0**). If the solution of the associated LP is [ν^*,**x***]=(**0**,**x**$^\$$), then **x**$^\$$ is an initial basic feasible solution of the original LP. Otherwise, there is no feasible solution to the original LP

A small change

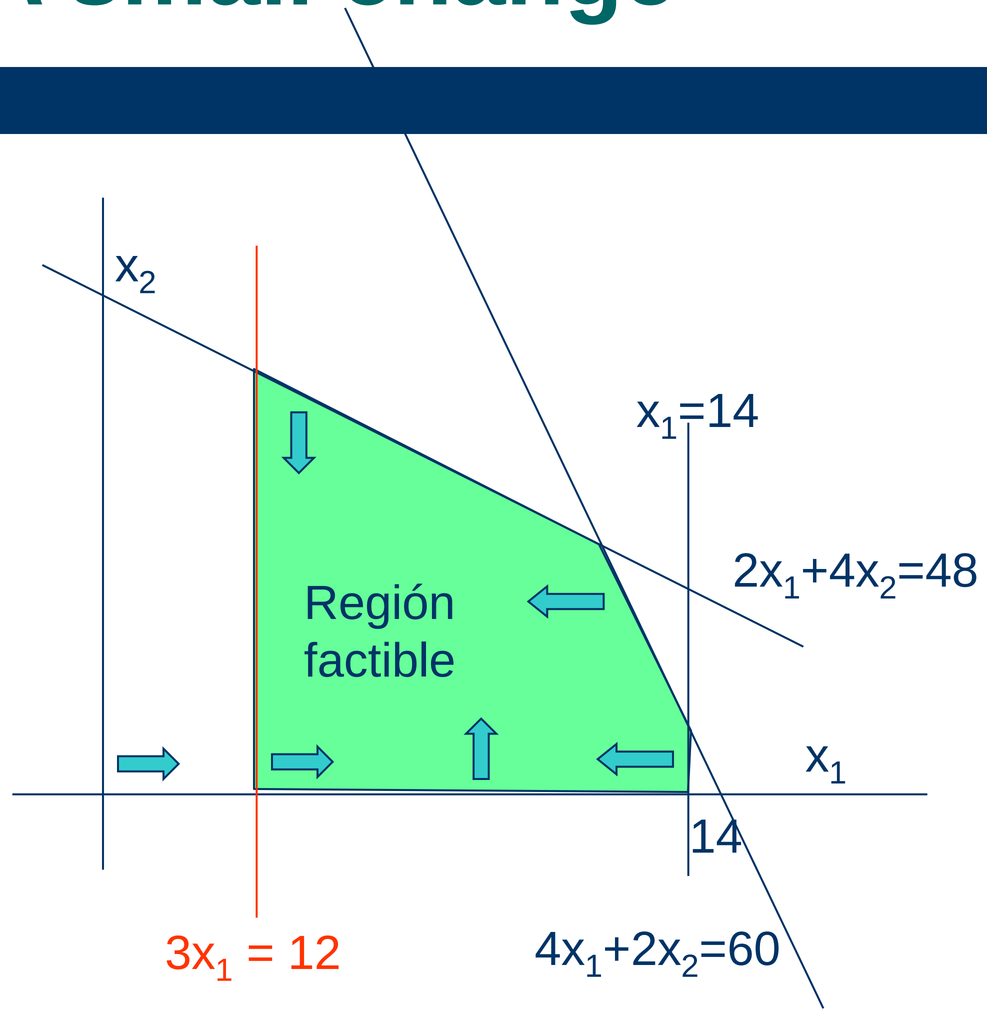


Original problem:

$$\max 6x_1 + 4x_2$$
$$\text{under:}$$
$$2x_1 + 4x_2 \leq 48$$
$$4x_1 + 2x_2 \leq 60$$
$$3x_1 \leq 42$$
$$x_1 \geq 0$$
$$x_2 \geq 0$$

Assume now that the machine number 3 must be processing silk at least 12h. per week

A small change



$$\max 6x_1 + 4x_2$$

under:

$$2x_1 + 4x_2 \leq 48$$

$$4x_1 + 2x_2 \leq 60$$

$$3x_1 \leq 42$$

$$3x_1 \geq 12$$

$$x_1 \geq 0, \quad x_2 \geq 0$$

Initial basic feasible solution

$$\max 6x_1 + 4x_2$$

sujeto a:

$$2x_1 + 4x_2 \leq 48$$
$$4x_1 + 2x_2 \leq 60$$
$$3x_1 \leq 42$$
$$3x_1 \geq 12$$
$$x_1 \geq 0, \quad x_2 \geq 0$$

$$\max (6,4,0,0,0,0)'\mathbf{x}$$

sujeto a:

$$\begin{pmatrix} 2 & 4 & 1 & 0 & 0 & 0 \\ 4 & 2 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 1 & 0 \\ -3 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 48 \\ 60 \\ 42 \\ -12 \end{bmatrix}$$

$$\mathbf{x} \geq \mathbf{0}$$

(0,0,48,60,42,-12) is not feasible, so, the phase I of the simplex algorithm is needed

Example phase I

All positive

$$\max_{\mathbf{x},\lambda} (-1,-1,-1,-1,0,0,0,0,0,0)' \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{x} \end{bmatrix} \quad \begin{pmatrix} 1 & 0 & 0 & 0 & 2 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 4 & 2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 3 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 3 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 48 \\ 60 \\ 42 \\ 12 \end{bmatrix}$$

$$\boldsymbol{\lambda} \geq \mathbf{0}, \ \mathbf{x} \geq \mathbf{0}$$

Provides an initial vertex for:

$$\max_{\mathbf{x}} (6,4,0,0,0,0)'\mathbf{x} \quad \begin{pmatrix} 2 & 4 & 1 & 0 & 0 & 0 \\ 4 & 2 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 1 & 0 \\ -3 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 48 \\ 60 \\ 42 \\ -12 \end{bmatrix}$$

$$\mathbf{x} \geq \mathbf{0}$$

Excel

Mixtures (diet)

The following table provides prices and nutrient content of several foods:

Daily minimum	Nutrient	Milk (cup)	Eggs (unit)	Meet (100g)	Bread (piece)	Cheese (100g)
75	Proteins g	8	7	20	2	7
1.2	Calcium g	0.3	0.03	0.01	0.01	0.25
1.2	Iron mg	0.1	1.5	3	0.6	0.1
3600	Calories	175	75	150	75	100
	Price €	0.3	0.1	0.8	0.1	0.6

As well as the minimum daily dose of each one. Which is the cheapest menu that covers the minimum daily amount of each nutrient, assuming that at least two pieces of bread must be included?

Mixtures

x_j amount of each type of food (x_1= cups of milk, x_2=number of eggs, x_3= grams of meet/100, x_4 = pieces of bread, x_5= grams of chease/100) for the menu

n_i minimum daily amount that must be eaten of every nutrient i

c_{ij} content of nutrient i in every unit of food j

p_j price of a unit of food j

$$\min_{x_j} \sum_j p_j x_j$$

$$\sum_j c_{ij} x_j \geq n_i \quad i = 1,\ldots,4$$

$$x_j \geq 0, \quad x_4 \geq 2$$

Blending

Two types of kerosene A and B, and two types of car naphtha A and B, are manufactured in a refinery by mixing alkylate, basic gasoline and cracked gasoline. Its physical properties and daily production are given in the table:

Row Product	PVR	Octane number (0)	Octane number (250)	Production m^3/day
Alkylate	5	94	108	4000
Basic Gasoline	4	74	86	4000
Cracked Gasoline	8	84	94	2500

With or without 250 mg/m^3 of tetraethyl lead (TEL)

TEL: mg/m^3 de tetraethyl lead

Blending

And the mixtures must have the following properties:

Product	PVR	TEL	Octane number	Benefit €/ m^3
Kerosene A	≤7	0	≥ 80	100
Kerosene B	≤7	250	≥ 91	110
Leaded naphtha A	---	250	≥ 87	95
Unleaded naphtha B	---	0	≥ 91	95

Decide which must be the daily production of each product and the blending that provides the maximum benefit per day

Blending- Nomenclature

x_{aA} = m^3 of alkylate spent daily in the manufacturing of kerosene A

x_{aB} = m^3 of alkylate with TEL spent daily in the manufacturing of kerosene B

x_{aS} = m^3 of alkylate spent daily in the manufacturing of unleaded naphtha A

x_{aP} = m^3 of alkylate with TEL spent daily in the manufacturing of leaded naphtha B

x_{bA}, x_{bB}, x_{bS}, x_{bP} m^3 of basic gasoline spent.. ....
x_{cA}, x_{cB}, x_{cS}, x_{cP} m^3 of cracked gasoline spent…

Blending - Aim

Find the blend that, satisfying the specifications of quality (in PVR and octane index) and the availability of products, maximizes the daily benefit

$$\max_{\mathbf{x}} 100(x_{aA}+x_{bA}+x_{cA})+110(x_{aB}+x_{bB}+x_{cB})+95(x_{aS}+x_{bS}+x_{cS})+95(x_{aP}+x_{bP}+x_{cP})$$

The amount of each type of products used in the blend cannot be bigger than its daily availability

$$x_{aA}+x_{aB}+x_{aS}+x_{aP}\le 4000$$
$$x_{bA}+x_{bB}+x_{bS}+x_{bP}\le 4000$$
$$x_{cA}+x_{cB}+x_{cS}+x_{cP}\le 2500$$

Blending - PVR

The kerosene of each type must comply with the minimum specifications of PVR:

$$\frac{x_{aA}}{x_{aA}+x_{bA}+x_{cA}}5+\frac{x_{bA}}{x_{aA}+x_{bA}+x_{cA}}4+\frac{x_{cA}}{x_{aA}+x_{bA}+x_{cA}}8 \le 7$$

$$\frac{x_{aB}}{x_{aB}+x_{bB}+x_{cB}}5+\frac{x_{bB}}{x_{aB}+x_{bB}+x_{cB}}4+\frac{x_{cA}}{x_{aB}+x_{bB}+x_{cB}}8 \le 7$$

Which can be written in linear form:

$$5x_{aA}+4x_{bA}+8x_{cA} \le 7(x_{aA}+x_{bA}+x_{cA})$$

$$5x_{aB}+4x_{bB}+8x_{cA} \le 7(x_{aB}+x_{bB}+x_{cB})$$

Blending – Octane number

The kerosene and naphtha of each type must comply with the minimum specifications of octane number and TEL content:

$$\frac{x_{aA}}{x_{aA}+x_{bA}+x_{cA}}94+\frac{x_{bA}}{x_{aA}+x_{bA}+x_{cA}}74+\frac{x_{cA}}{x_{aA}+x_{bA}+x_{cA}}84\geq 80$$

$$\frac{x_{aB}}{x_{aB}+x_{bB}+x_{cB}}108+\frac{x_{bB}}{x_{aB}+x_{bB}+x_{cB}}86+\frac{x_{cB}}{x_{aB}+x_{bB}+x_{cB}}94\geq 91$$

$$\frac{x_{aS}}{x_{aS}+x_{bS}+x_{cS}}94+\frac{x_{bS}}{x_{aS}+x_{bS}+x_{cS}}74+\frac{x_{cS}}{x_{aS}+x_{bS}+x_{cS}}84\geq 91$$

$$\frac{x_{aP}}{x_{aP}+x_{bP}+x_{cP}}108+\frac{x_{bP}}{x_{aP}+x_{bP}+x_{cP}}86+\frac{x_{cP}}{x_{aP}+x_{bP}+x_{cP}}94\geq 87$$

Which can be written also in linear form

Blending

The final problem can be formulated as a LP one

$$\max_{\mathbf{x}} 100(x_{aA}+x_{bA}+x_{cA})+110(x_{aB}+x_{bB}+x_{cB})+95(x_{aS}+x_{bS}+x_{cS})+95(x_{aP}+x_{bP}+x_{cP})$$

$$x_{aA}+x_{aB}+x_{aS}+x_{aP}\le 4000 \qquad \mathbf{x}\ge\mathbf{0}$$

$$x_{bA}+x_{bB}+x_{bS}+x_{bP}\le 4000$$

$$x_{cA}+x_{cB}+x_{cS}+x_{cP}\le 2500$$

$$5x_{aA}+4x_{bA}+8x_{cA}\le 7(x_{aA}+x_{bA}+x_{cA})$$

$$5x_{aB}+4x_{bB}+8x_{cB}\le 7(x_{aB}+x_{bB}+x_{cB})$$

$$94x_{aA}+74x_{bA}+84x_{cA}\ge 80(x_{aA}+x_{bA}+x_{cA})$$

$$108x_{aB}+86x_{bB}+94x_{cB}\ge 91(x_{aB}+x_{bB}+x_{cB})$$

$$94x_{aS}+74x_{bS}+84x_{cS}\ge 91(x_{aS}+x_{bS}+x_{cS})$$

$$108x_{aP}+86x_{bP}+94x_{cP}\ge 87(x_{aP}+x_{bP}+x_{cP})$$

Blending

The total amount consumed of each row product will be:

$$x_a = x_{aA} + x_{aB} + x_{aS} + x_{aP}$$

$$x_b = x_{bA} + x_{bB} + x_{bS} + x_{bP}$$

$$x_c = x_{cA} + x_{cB} + x_{cS} + x_{cP}$$

$$TEL = 250(x_{aB} + x_{bB} + x_{cB} + x_{aP} + x_{bP} + x_{cP})\ mg$$

And the total amount produced of every final product:

$$x_A = x_{aA} + x_{bA} + x_{cA}$$

$$x_B = x_{aB} + x_{bB} + x_{cB}$$

$$x_S = x_{aS} + x_{bS} + x_{cS}$$

$$x_P = x_{aP} + x_{bP} + x_{cP}$$

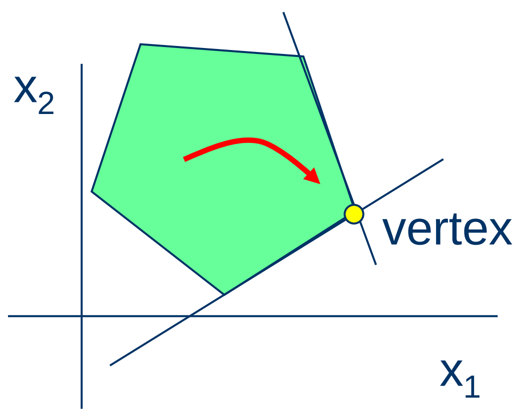
Complexity

- The relation between the time spent by an algorithm in finding a solution and the size (n) of the decision vector x is called complexity
- The number of vertices that the simplex algorithm must visit depends on the initial vertex. In the worst case in may be 2^n-1, so the algorithm has an exponential complexity O(2^n-1)
- A desirable property for an algorithm to be useful is having polynomial complexity

Other LP algorithms

- There are other methods for solving LP problems:
  - Revised simplex. It has exponential complexity but reduces the number of computations on the columns.
  - Khachiyan Algorithm. It has polynomial complexity $O(n^4L)$ where L depends on the required precision.
  - Karmarkar Algorithm. It is an interior point method of polynomial complexity $O(n^{3.5}L)$. It is an efficient method for large scale problems. It does not travel through vertices, but it generate a sequence of points starting from an feasible point in the interior of the feasible set.

Interior point methods



Breakthrough in Problem Solving

By JAMES GLEICK

A 28-year-old mathematician at A.T.&T. Bell Laboratories has made a startling theoretical breakthrough in the solving of systems of equations that often grow too vast and complex for the most powerful computers.

The discovery, which is to be formally published next month, is already circulating rapidly through the mathematical world. It has also set off a deluge of inquiries from brokerage houses, oil companies and airlines, industries with millions of dollars at stake in problems known as linear programming.

Faster Solutions Seen

These problems are fiendishly complicated systems, often with thousands of variables. They arise in a variety of commercial and government applications, ranging from allocating time on a communications satellite to routing millions of telephone calls over long distances, or whenever a limited, expensive resource must be spread most efficiently among competing users. And investment companies use them in creating portfolios with the best mix of stocks and bonds.

The Bell Labs mathematician, Dr. Narendra Karmarkar, has devised a radically new procedure that may speed the routine handling of such problems by businesses and Government agencies and also make it possible to tackle problems that are now far out of reach.

"This is a path-breaking result," said Dr. Ronald L. Graham, director of mathematical sciences for Bell Labs in Murray Hill, N.J.

"Science has its moments of great progress, and this may well be one of them."

Because problems in linear programming can have billions or more possible answers, even high-speed computers cannot check every one. So computers must use a special procedure, an algorithm, to examine as few answers as possible before finding the best one — typically the one that minimizes cost or maximizes efficiency.

A procedure devised in 1947, the simplex method, is now used for such problems,

Continued on Page A19, Column 1

THE NEW YORK TIMES, November 19, 1984

Interior points methods

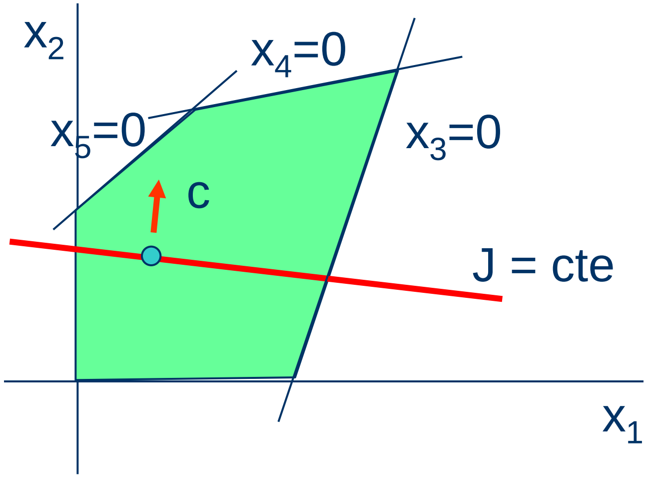
- Karmarkar algorithm (1984) is the best known
- For simplicity, the Dikin method, which can be applied to LP problems in standard form will be described briefly:

$$\max_{\mathbf{x}} J = \mathbf{c}'\mathbf{x}$$

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

$$\mathbf{x} \geq \mathbf{0}$$

Starting from a point in the interior of the feasible set, **x** > **0**, the point is moved to other places that respect the constraints and improve the cost J



Interior point method (Dikin)

$$\max_{\mathbf{x}} J = \mathbf{c}'\mathbf{x}$$
$$\mathbf{Ax} = \mathbf{b}$$
$$\mathbf{x} \geq \mathbf{0}$$

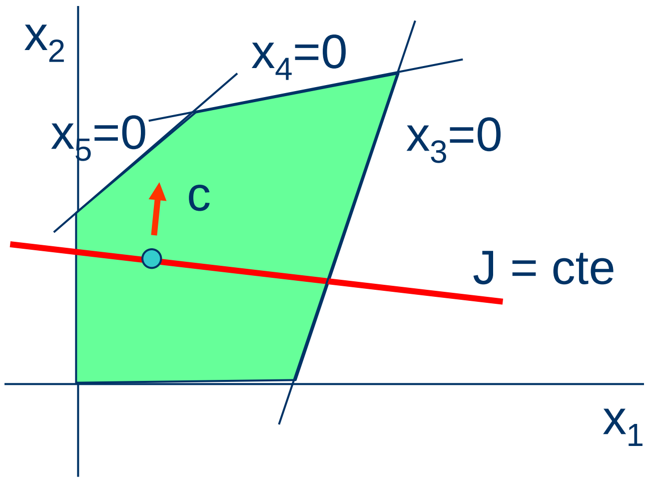
Any move respecting the constraints must fulfil:

$$\mathbf{A}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} \quad \Rightarrow \quad \mathbf{A}\,\Delta\mathbf{x} = \mathbf{0}$$

At any feasible x, the vector c points to an improvement direction of J, but in order to guarantee that Δx fulfils the constraints, the move should follow a direction perpendicular to A given by its orthogonal projection

$$\prod\nolimits_{A}^{\perp} = I - A^T(AA^T)^{-1}A$$

$$A\Delta x = A\left[\mathbf{I} - \mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{A}\right]\mathbf{c} =$$
$$= Ac - A\mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{A}\,\mathbf{c} = 0$$



Moving x in the direction:

$$\Delta x = \left[\mathbf{I} - \mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{A}\right]\mathbf{c}$$

The equality constraints are satisfied

Interior point method (Dikin)

$$\max_{\mathbf{x}} J = \mathbf{c}'\mathbf{x}$$
$$\mathbf{A}\mathbf{x} = \mathbf{b}$$
$$\mathbf{x} \geq \mathbf{0}$$

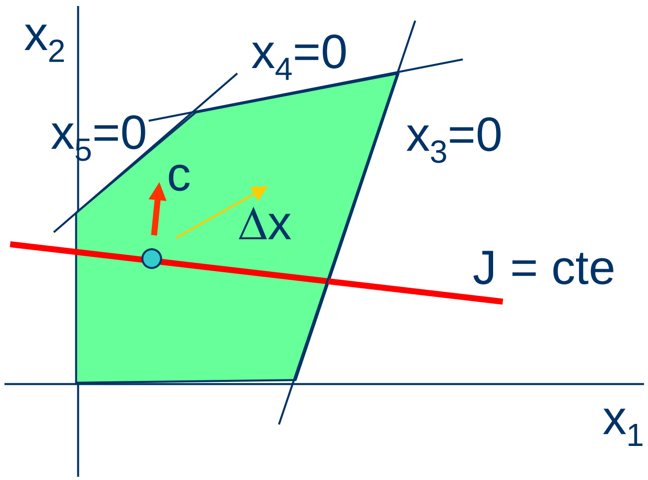
Moving x in the direction:

$$\Delta x = \left[\mathbf{I} - \mathbf{A}^{\mathbf{T}}(\mathbf{A}\mathbf{A}^{\mathbf{T}})^{-1}\mathbf{A}\right]\mathbf{c}$$

J is increased while the constraints are respected.

In order to assure that **x ≥ 0** it is important also to choose an adequate step lenght σ

Usually, a previous scaling facilitate the selection of σ based on the most negative component of Δx



$$\mathbf{x_{k+1}} = \mathbf{x_k} + \sigma \Delta x$$

If all components of Δx are positive, then the problem is unbounded and there is no maximum

Dual Problem

There exists a dual LP problem associated to every primal LP one

There is a relation between both problems that can be used to analyse the solutions and think in alternative solution paths

Primal:

(1xn)(nx1)

$$\max_{\mathbf{x}} \mathbf{c}'\mathbf{x}$$

$$\mathbf{A}\mathbf{x} \leq \mathbf{b}$$

$$\mathbf{x} \geq \mathbf{0} \quad \text{(nx1)}$$

(mxn)(nx1)

Dual:

(1xm)(mx1)

$$\min_{\mathbf{z}} \mathbf{b}'\mathbf{z}$$

$$\mathbf{A}'\mathbf{z} \geq \mathbf{c}$$

$$\mathbf{z} \geq \mathbf{0} \quad \text{(mx1)}$$

(nxm)(mx1)

The dual of the dual is the original LP problem

Example

Primal

$$\max_{\mathbf{x}} 6x_1 + 4x_2$$

under:

$$2x_1 + 4x_2 \leq 48$$

$$4x_1 + 2x_2 \leq 60$$

$$3x_1 \leq 42$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

Dual

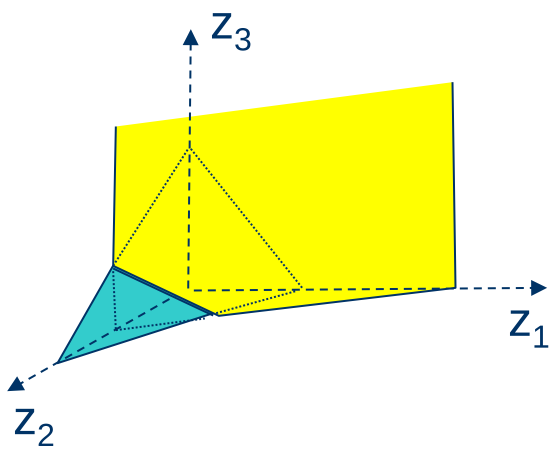
$$\min_{\mathbf{z}} 48z_1 + 60z_2 + 42z_3$$

under:

$$2z_1 + 4z_2 + 3z_3 \geq 6$$

$$4z_1 + 2z_2 + 0z_3 \geq 4$$

$$z_1 \geq 0, \quad z_2 \geq 0, \quad z_3 \geq 0$$



Dual problem of the LP standard one

Primal

$$\max_{\mathbf{x}} \mathbf{c}'\mathbf{x} \quad \mathbf{A}\mathbf{x} = \mathbf{b} \quad \mathbf{x} \geq 0$$

$$\Rightarrow \quad \max_{\mathbf{x}} \mathbf{c}'\mathbf{x} \quad \mathbf{A}\mathbf{x} \leq \mathbf{b} \quad \mathbf{A}\mathbf{x} \geq \mathbf{b} \quad \mathbf{x} \geq 0$$

$$\Rightarrow \quad \max_{\mathbf{x}} \mathbf{c}'\mathbf{x} \quad \mathbf{A}\mathbf{x} \leq \mathbf{b} \quad -\mathbf{A}\mathbf{x} \leq -\mathbf{b} \quad \mathbf{x} \geq 0$$

$$\Rightarrow \quad \max_{\mathbf{x}} \mathbf{c}'\mathbf{x} \quad \begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix} \quad \mathbf{x} \geq 0$$

Dual

$$\min_{\boldsymbol{\lambda},\boldsymbol{\nu}} [\mathbf{b}' \quad -\mathbf{b}'] \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\nu} \end{bmatrix} = \min_{\boldsymbol{\lambda},\boldsymbol{\nu}} \mathbf{b}'(\boldsymbol{\lambda} - \boldsymbol{\nu})$$

$$[\mathbf{A}' \quad -\mathbf{A}'] \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\nu} \end{bmatrix} \geq \mathbf{c} \quad \Rightarrow \mathbf{A}'(\boldsymbol{\lambda} - \boldsymbol{\nu}) \geq \mathbf{c}$$

$$\begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\nu} \end{bmatrix} \geq \mathbf{0} \qquad \mathbf{z} = \boldsymbol{\lambda} - \boldsymbol{\nu}$$

Dual

$$\min_{\mathbf{z}} \mathbf{b}'\mathbf{z} \quad \mathbf{A}'\mathbf{z} \geq \mathbf{c}$$

z unconstraint

Example

Primal

$$\max_{\mathbf{x}} J = 6x_1 + 4x_2 + 0x_3 + 0x_4 + 0x_5$$

$$\begin{pmatrix} 2 & 4 & 1 & 0 & 0 \\ 4 & 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 48 \\ 60 \\ 42 \end{pmatrix}$$

$$\mathbf{x} \geq \mathbf{0}$$

Dual

$$\min_{\mathbf{z}} 48z_1 + 60z_2 + 42z_3$$

$$\begin{pmatrix} 2 & 4 & 3 \\ 4 & 2 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \geq \begin{bmatrix} 6 \\ 4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Duality Lemma

If x and z are two feasible solutions of the primal and dual LP problems respectively, then:

$$\mathbf{c'x \leq b'z}$$

So, any solution of the primal is a lower bound of any solution of the dual

In fact, if x and z are feasible points: $Ax = b$, $x \geq 0$, $A'z \geq c$

$A'z \geq c$, $\Rightarrow x'A'z \geq x'c$ but $x'A' = b'$, $\Rightarrow b'z \geq x'c$

So, if one of the problems is unbounded, the other has no feasible solution

The result can be applied also when the LP problem is formulated with inequalities

Duality Lemma

According to the duality lemma, if x and y are two feasible solutions of the primal and dual LP problems respectively, then:

$$\mathbf{c'x \le b'z}$$

Hence, if x_0 and z_0 are two feasible points of the primal and dual LP problems respectively and they verify **$c'x_0 = b'z_0$,** then both are optimal for these problems because they reach an upper /lower bound

The opposite is also true and jointly they form the duality theorem

The duality Theorem

If the primal LP has an optimal solution x^*, then the dual has also a solution and they verify: $c'x^* = b'z^*$

Primal
$$\max_{\mathbf{x}} \mathbf{c'x}$$
$$\mathbf{Ax} = \mathbf{b}$$
$$\mathbf{x} \geq 0$$

Dual
$$\min_{\mathbf{z}} \mathbf{b'z}$$
$$\mathbf{A'z} \geq \mathbf{c}$$

Every optimal solution of the primal LP must be a basic feasible solution verifying: (the columns have been reordered so that B is a basis)

$$\mathbf{Ax}^* = \mathbf{b} \Rightarrow [\mathbf{B} \quad \mathbf{N}]\mathbf{x}^* = \mathbf{b} \Rightarrow \mathbf{B}^{-1}[\mathbf{B} \quad \mathbf{N}]\mathbf{x}^* = \mathbf{B}^{-1}\mathbf{b} \Rightarrow [\mathbf{I} \quad \mathbf{B}^{-1}\mathbf{N}]\mathbf{x}^* = \mathbf{B}^{-1}\mathbf{b}$$

$$\mathbf{c'x}^* = [\mathbf{c}_\mathbf{B} \quad \mathbf{c}_\mathrm{N}]\begin{bmatrix}\mathbf{x}_\mathbf{B}^* \\ \mathbf{x}_\mathrm{N}^*\end{bmatrix} \qquad \mathbf{c}'_\mathrm{N} - \mathbf{c}'_\mathrm{B}\mathbf{B}^{-1}\mathrm{N} \leq \mathbf{0}'$$

← Extremum condition

Duality Theorem

$$\mathbf{c_B}'\mathbf{B}^{-1}\mathbf{N} \geq \mathbf{c_N}' \qquad \text{defining } \mathbf{z}' = \mathbf{c_B}'\mathbf{B}^{-1} \quad \Rightarrow \quad \mathbf{z}'\mathbf{N} \geq \mathbf{c_N}'$$

z is a basic feasible solution of the dual problem. In fact:

$$\mathbf{z}'\mathbf{A} = \mathbf{z}'[\mathbf{B} \quad \mathbf{N}] = \mathbf{c_B}'\mathbf{B}^{-1}[\mathbf{B} \quad \mathbf{N}] = [\mathbf{c_B}' \quad \mathbf{c_B}'\mathbf{B}^{-1}\mathbf{N}] \geq [\mathbf{c_B}' \quad \mathbf{c_N}'] = \mathbf{c}'$$

So it verifies z'A ≥ c' and it is feasible in the dual. In addition:

$$\mathbf{z}'\mathbf{b} = \mathbf{c_B}'\mathbf{B}^{-1}\mathbf{b} = \mathbf{c_B}'\mathbf{x_B^*}$$

So, because of the duality lemma, z is optimum for the dual as it reaches an upper bound, because ($B^{-1}b$, 0) is a solution of the primal

The result can be applied also when the LP problem is formulated with inequalities

Solution of the dual problem

If the optimal solution of the primal LP problem is known, then it is possible to compute the solution of the dual (and viceversa)

In order to compute z^*, one must use the expression Ax = b where the columns have been reordered so that the m first ones correspond to the base B of the optimum.

The optimal solution z^* of the dual can be obtained from:

$$\mathbf{z}^{*}{}' = \mathbf{c}_{\mathbf{B}}^{'}\mathbf{B}^{-1}$$

Which can always be computed as B is a basis

Sensitivities of the optimum

$$\max_{\mathbf{x}} J = \mathbf{c}'\mathbf{x}$$
$$\mathbf{A}\mathbf{x} = \mathbf{b}$$
$$\mathbf{x} \geq 0$$

How does the optimal cost J^* change if the constraint vector b changes?

$$\frac{\partial J^*}{\partial \mathbf{b}} = \frac{\partial \mathbf{c}'\mathbf{x}^*}{\partial \mathbf{b}} = \frac{\partial \mathbf{b}'\mathbf{z}^*}{\partial \mathbf{b}} = \mathbf{z}^{*\prime}$$

The solution z^* of the dual LP problem provides the sensitivity of the optimal cost of the primal with respect to the constraint vector b

The values z^* are called sometimes shadow prices

Obviously, it also happens:

$$\frac{\partial J^*}{\partial \mathbf{c}} = \frac{\partial \mathbf{c}'\mathbf{x}^*}{\partial \mathbf{c}} = \mathbf{x}^{*\prime}$$

Example: How much changes J^*?

Primal

$$\max_{\mathbf{x}} 6x_1 + 4x_2$$

under:

$$2x_1 + 4x_2 \leq 48$$

$$4x_1 + 2x_2 \leq 60$$

$$3x_1 \leq 42$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

How much changes the optimal cost 96 if machine 1 can work for 50 h. per week?

$$\frac{\partial J^*}{\partial b_1} = z_1^*$$

In order to answer this question the solution of the dual problem must be computed, either directly or, in a more efficient way, from the base B of the optimal solution of the primal LP

$$\mathbf{z}' = \mathbf{c}_{\mathbf{B}}' \mathbf{B}^{-1}$$

Example: How much changes J*?

$$\max_{\mathbf{x}} 6x_1 + 4x_2$$

$$\left.\begin{array}{l} 2x_1 + 4x_2 \le 48 \\ 4x_1 + 2x_2 \le 60 \\ 3x_1 \le 42 \end{array}\right\}$$

$$\mathbf{x} \ge 0$$

Excel

$$\max_{\mathbf{x}} J = 6x_1 + 4x_2 + 0x_5 + 0x_4 + 0x_3$$

$$\begin{pmatrix} 2 & 4 & 0 & 0 & 1 \\ 4 & 2 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_5 \\ x_4 \\ x_3 \end{pmatrix} = \begin{pmatrix} 48 \\ 60 \\ 42 \end{pmatrix} \quad \mathbf{x} \ge \mathbf{0}$$

B

$$\mathbf{z}' = \mathbf{c}_B' \mathbf{B}^{-1}$$

$$[z_1 \quad z_2 \quad z_3] = [6 \quad 4 \quad 0] \begin{bmatrix} 2 & 4 & 0 \\ 4 & 2 & 0 \\ 3 & 0 & 1 \end{bmatrix}^{-1} = [1/3 \quad 4/3 \quad 0]$$

$$\mathbf{z}^* = [1/3 \quad 4/3 \quad 0]$$

$$(50-48)\frac{\partial J^*}{\partial b_1} = 2z_1^* = 2/3$$

Cesar de Prada ISA-UVA

Transport problems

A firm has factories in Galicia, La Rioja and Murcia, and warehouses in Seville, Madrid, Barcelona, Santander and Bilbao. The cost of sending a unity of a product from a factory to a warehouse is given in the adjoint table, as well as the stocks in every factory and the demands from the warehouses:

		Sevilla	*Madrid*	*Barcelona*	*Santander*	*Bilbao*
Demandas por almacén-->		180	80	200	160	220
Plantas:	*Existencias*	*Costos de envío de la planta "x" al almacén "y" (en la intersecció*				
Galicia	310	10	8	6	5	4
La Rioja	260	6	5	4	3	6
Murcia	280	3	4	5	5	9

Which is the amount of product that must be sent from each factory to every warehouse in order to minimize the transport costs while satisfying the demand of each warehouse?

Transport problems

i set of factories   j set of warehouses

x_{ij} amount of product sent from factory i to warehouse j

c_{ij} cost of sending a unity of product from factory i to warehouse j

e_i stock of product in factory i

d_j demand of warehouse j

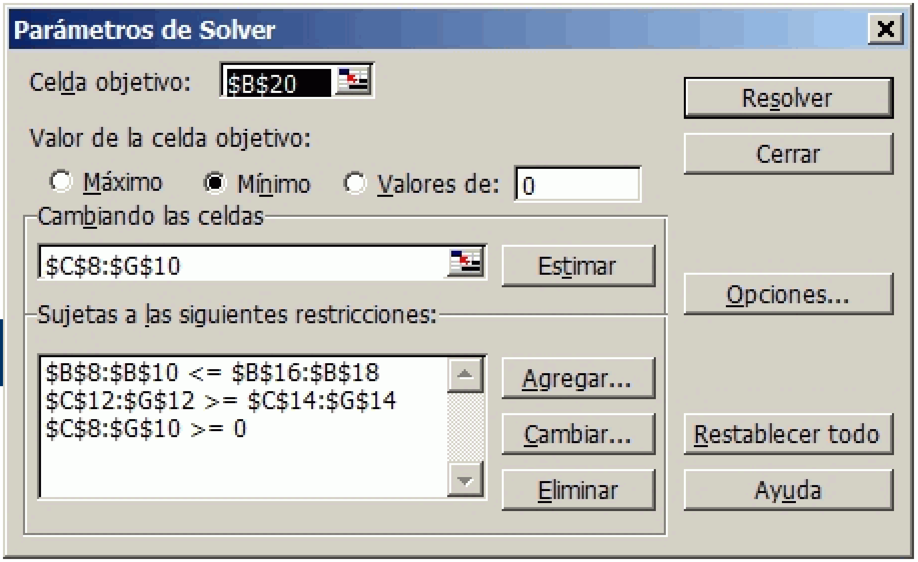
$$\min_{\mathbf{x}} \sum_{i,j} c_{ij} x_{ij}$$

$$\sum_i x_{ij} \geq d_j \qquad \sum_j x_{ij} \leq e_i \qquad x_{ij} \geq 0$$

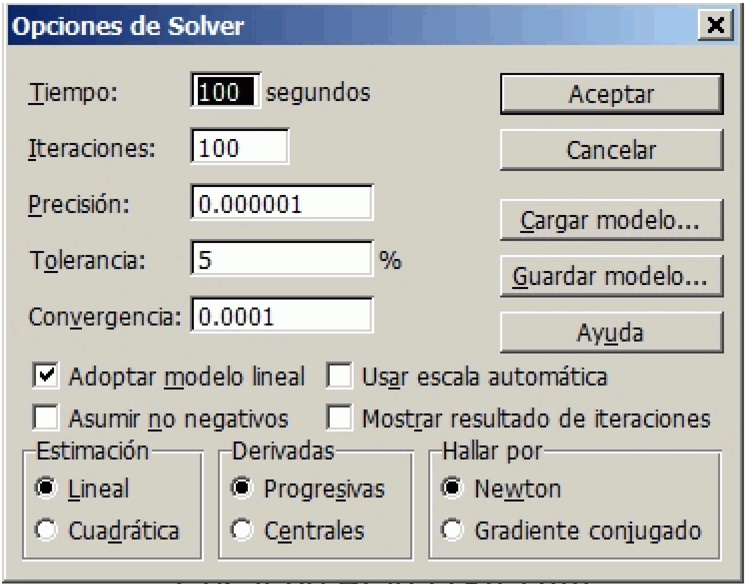
$$i = 1,2,3 \qquad j = 1,2,3,4,5$$

Excel

Excel



		Cantidad a enviar de la planta "x" al almacén "y' (en la intersección):				
Plantas	*Total*	*Sevilla*	*Madrid*	*Barcelona*	*Santander*	*Bilbao*
Galicia	300	0	0	0	80	220
La Rioja	260	0	0	180	80	0
Murcia	280	180	80	20	0	0
		---	---	---	---	---
TOTAL:		180	80	200	160	220
Demandas por almacén-->		180	80	200	160	220
Plantas:	*Existencias*	*Costos de envío de la planta "x" al almacén "y" (en la intersección):*				
Galicia	310	10	8	6	5	4
La Rioja	260	6	5	4	3	6
Murcia	280	3	4	5	5	9
Envío:	3 200 $	540 $	320 $	820 $	640 $	880 $



Precision: Refers to the error in the constraints

Convergence: Refers to the error in the cost

Tolerance: Refers to the error in the cost (MILP)

Sensibility of the constraints

Constraints

Admissible change in the constraints before the shadow price changes

Decrease

Restricciones

Celda	Nombre	Valor Igual	Sombra precio	Restricción lado derecho	Aumento permisible	Aumento permisible
B8	Galicia Total	300	0	310	1E+30	10
B9	La Rioja Total	260	-2	260	80	10
B10	Murcia Total	280	-1	280	80	10
C12	TOTAL: ---	180	4	180	10	80
D12	TOTAL: ---	80	5	80	10	80
E12	TOTAL: ---	200	6	200	10	80
F12	TOTAL: ---	160	5	160	10	80
G12	TOTAL: ---	220	4	220	10	220

Sensibility (cost function)

Optimal solution

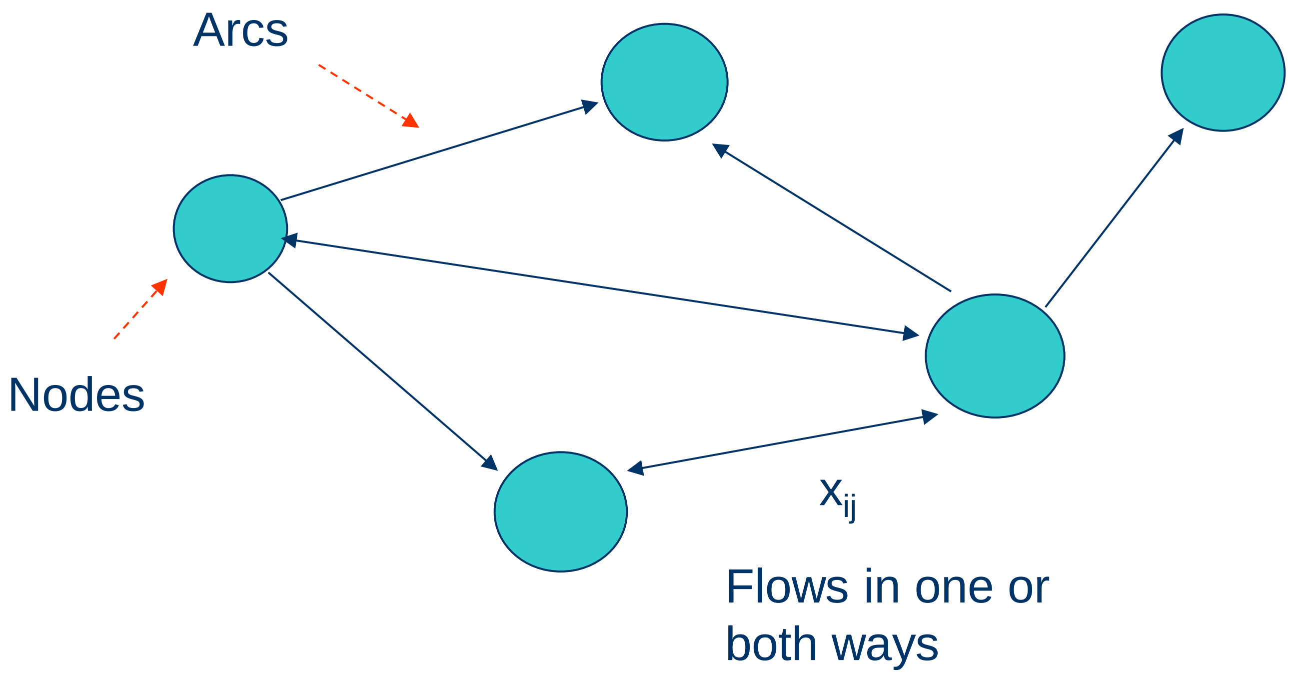
Relative gains (min ≥0)

decrease

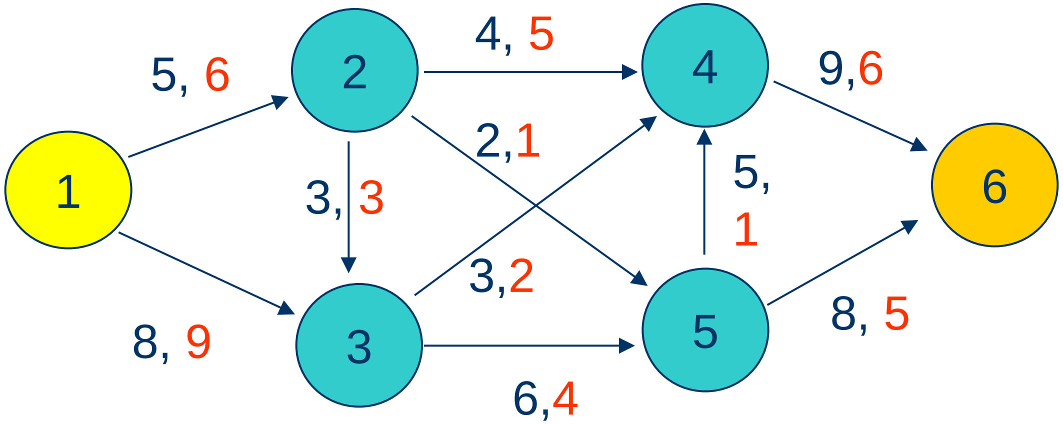
Celda	Nombre	Valor Igual	Gradiente reducido	Coeficiente objetivo	Aumento permisible	Aumento permisible
C8	Galicia Sevilla	0	6	10	1E+30	6
D8	Galicia Madrid	0	3	8	1E+30	3
E8	Galicia Barcelona	0	0	6	1E+30	0
F8	Galicia Santander	80	0	5	0	1
G8	Galicia Bilbao	220	0	4	4	4
C9	La Rioja Sevilla	0	4	6	1E+30	4
D9	La Rioja Madrid	0	2	5	1E+30	2
E9	La Rioja Barcelona	180	0	4	0	1
F9	La Rioja Santander	80	0	3	1	0
G9	La Rioja Bilbao	0	4	6	1E+30	4
C10	Murcia Sevilla	180	0	3	4	4
D10	Murcia Madrid	80	0	4	2	5
E10	Murcia Barcelona	20	0	5	1	2
F10	Murcia Santander	0	1	5	1E+30	1
G10	Murcia Bilbao	0	6	9	1E+30	6

Possible multiplicity of the optimal solution

Directed Graphs

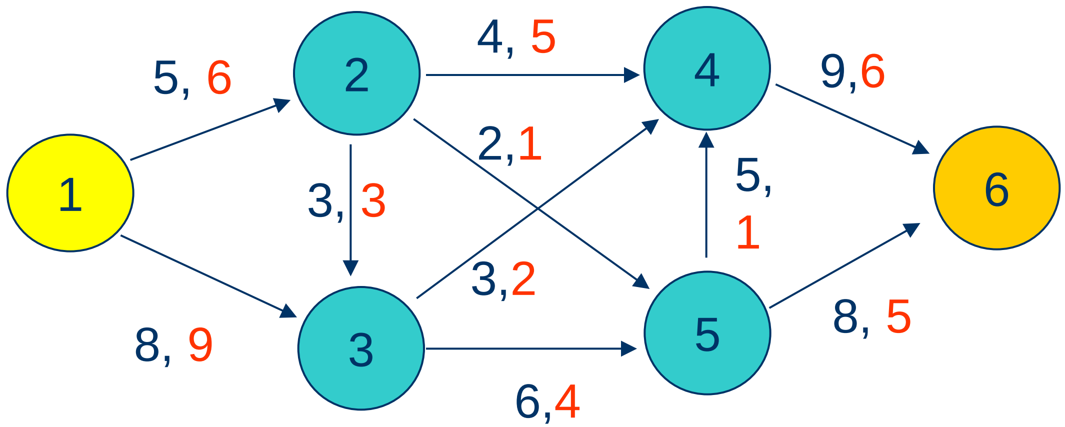


Network flows problems



A chemical company has several plants connected by pipelines whose maximum capacity in m^3/min, direction and architecture can be seen in the figure, besides the cost of sending a unit flow (in red). It wishes to send a flow of 9 m3/min from plant number 1 to plant number 6. Which is the best route in order to minimize the transport costs? Assume that no accumulation or generation of product takes place in the intermediate plants.

Flow in networks



$$\min_{x_{ij}} \sum_{i,j=1}^{N} c_{ij} x_{ij}$$

x_{ij} amount sent from node i to node j

$$\sum_{k=1}^{6} x_{ik} = \sum_{k=1}^{6} x_{ki} \quad \forall i \neq 1,6 \qquad \sum_{j=1}^{6} x_{1j} = 9 \qquad \sum_{i=1}^{6} x_{i6} = 9$$

$$0 \leq x_{ij} \leq U_{ij} \qquad x_{ij} = 0 \text{ if i and j are not connected}$$

U capacity

c cost

Flow in networks GAMS

```
SET
n nodos   /nod1,nod2,nod3,nod4,nod5, nod6/
i(n) nodos de salida  /nod1,nod2,nod3,nod4,nod5/
j(n) nodos de llegada  /nod2,nod3,nod4,nod5,nod6/
k(n) nodos intermedios /nod2,nod3,nod4,nod5/;
alias (n,nn)

Table c(n,nn) costes de envio
        nod1  nod2   nod3 nod4  nod5  nod6
nod1      0     6      9     0     0    0
nod2      0     0      3     5     1    0
nod3      0     0      0     2     4    0
nod4      0     0      0     0     0    6
nod5      0     0      0     1     0    5
nod6      0     0      0     0     0    0;
```

GAMS

```
Table b(n,nn) capacidad de envio
          nod1  nod2  nod3  nod4  nod5 nod6
nod1       0     5     8     0     0     0
nod2       0     0     3     4     2     0
nod3       0     0     0     3     6     0
nod4       0     0     0     0     0     9
nod5       0     0     0     5     0     8
nod6       0     0     0     0     0     0;

Variables
x(n,nn) cantidades de envio
z;
Positive Variables x(n,nn);
x.up(n,nn) = b(n,nn);
```

GAMS

```
Equations
obj      defines the cost function
const1   defines the constraint on the first output node
const2 defines the constraint on the last arrival node
const3(n)   balance in one node;

obj..          z =e= sum((i,j), c(i,j)*x(i,j));
const1..    sum(j, x('nod1',j)) =e= 9;
const2..    sum(i, x(i, 'nod6')) =e= 9;
const3(k)..    sum(i, x(i,k))=e= sum(j, x(k,j));

Model redes /all/;
Solve redes using lp minimizing z;
Display x.l;
```

Results GAMS

```
EXECUTION TIME      =       0.125 SECONDS      4 Mb  WEX236-236 Apr  6, 2011

GAMS Rev 236  WEX-WEI 23.6.5 x86_64/MS Windows         10/25/11 03:12:45 Page 5
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
Solution Report     SOLVE redes Using LP From line 57

          S O L V E     S U M M A R Y

    MODEL   redes               OBJECTIVE  z
    TYPE    LP                  DIRECTION  MINIMIZE
    SOLVER  CPLEX               FROM LINE  57

**** SOLVER STATUS        1 Normal Completion
**** MODEL STATUS         1 Optimal
**** OBJECTIVE VALUE      144.0000

 RESOURCE USAGE, LIMIT        0.149   (sg. CPU)   1000.000
 ITERATION COUNT, LIMIT        2     2000000000
```

Results GAMS

```
----     58 VARIABLE  x.L  cantidades de envio
            nod2        nod3        nod4        nod5        nod6
nod1   5.000       4.000
nod2                            3.000       2.000
nod3                            3.000       1.000
nod4                                                    6.000
nod5                                                    3.000

---- VAR      z            -INF    144.000     +INF

                        LOWER     LEVEL     UPPER    MARGINAL
---- EQU obj              .         .          .       1.000
---- EQU const1         9.000     9.000      9.000     9.000
---- EQU const2         9.000     9.000      9.000     9.000
……
```

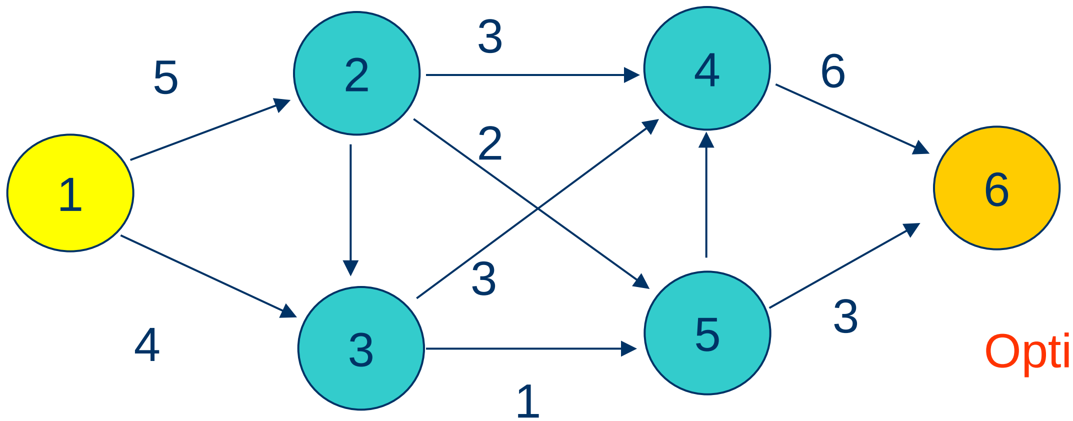
Results GAMS

---- VAR x cantidades de envio

	LOWER	LEVEL	UPPER	MARGINAL
nod1.nod2	.	5.000	5.000	-1.000
nod1.nod3	.	4.000	8.000	.
nod1.nod4	.	.	.	-12.000
nod1.nod5	.	.	.	-13.000
nod1.nod6	.	.	.	-18.000
nod2.nod3	.	.	3.000	1.000
nod2.nod4	.	3.000	4.000	.
nod2.nod5	.	2.000	2.000	-5.000

…….

Flow in networks, solution



Optimal Cost 144

$$\min_{x_{ij}} \sum_{i,j=1}^{N} c_{ij} x_{ij}$$

x_{ij} amount sent from node i to node j

$$\sum_{k=1}^{6} x_{ik} = \sum_{k=1}^{6} x_{ki} \quad \forall i \neq 1,6 \qquad \sum_{j=1}^{6} x_{1j} = 9 \qquad \sum_{i=1}^{6} x_{i6} = 9$$

U capacity

c cost

$$0 \leq x_{ij} \leq U_{ij} \qquad x_{ij} = 0 \text{ if } i \text{ and } j \text{ are not connected}$$

Heat exchanger network synthesis

Let's assume that a network is optimal if it has the following characteristics:

1. Minimum utility cost
2. Minimum number of matches (units)
3. Minimum investment cost (configuration and sizes)

Sequential synthesis of a heat exchanger network: 1 Minimum utility cost

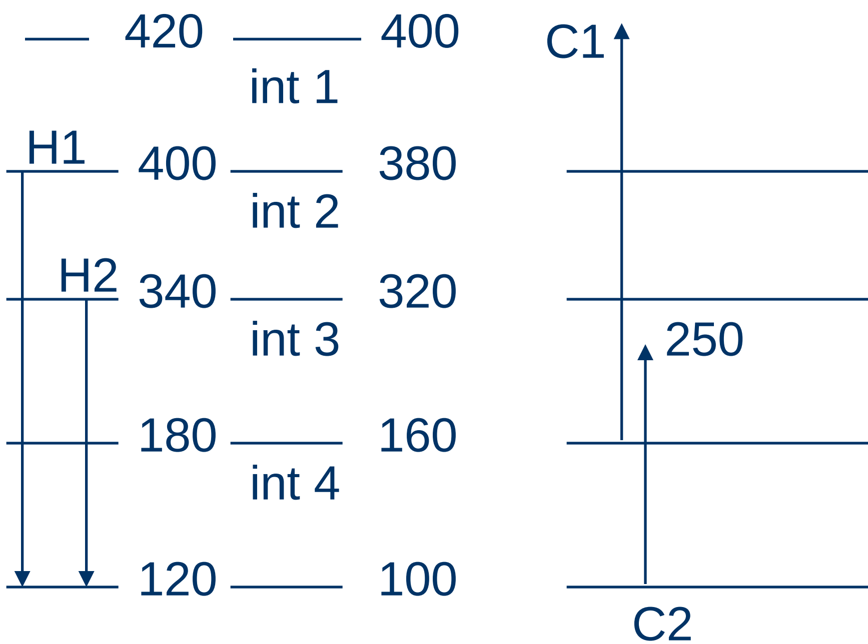
Which is the minimum utility cost in the heat exchanger problem ?:

	Fc_p(kW/ºC)	T_{in}(C)	T_{out}(C)
H1	1	400	120
H2	2	340	120
C1	1.5	160	400
C2	1.3	100	250

Steam: 500ºC Cold water: 20-30ºC Temp. difference = 20ºC

Sequential synthesis of a heat exchanger network

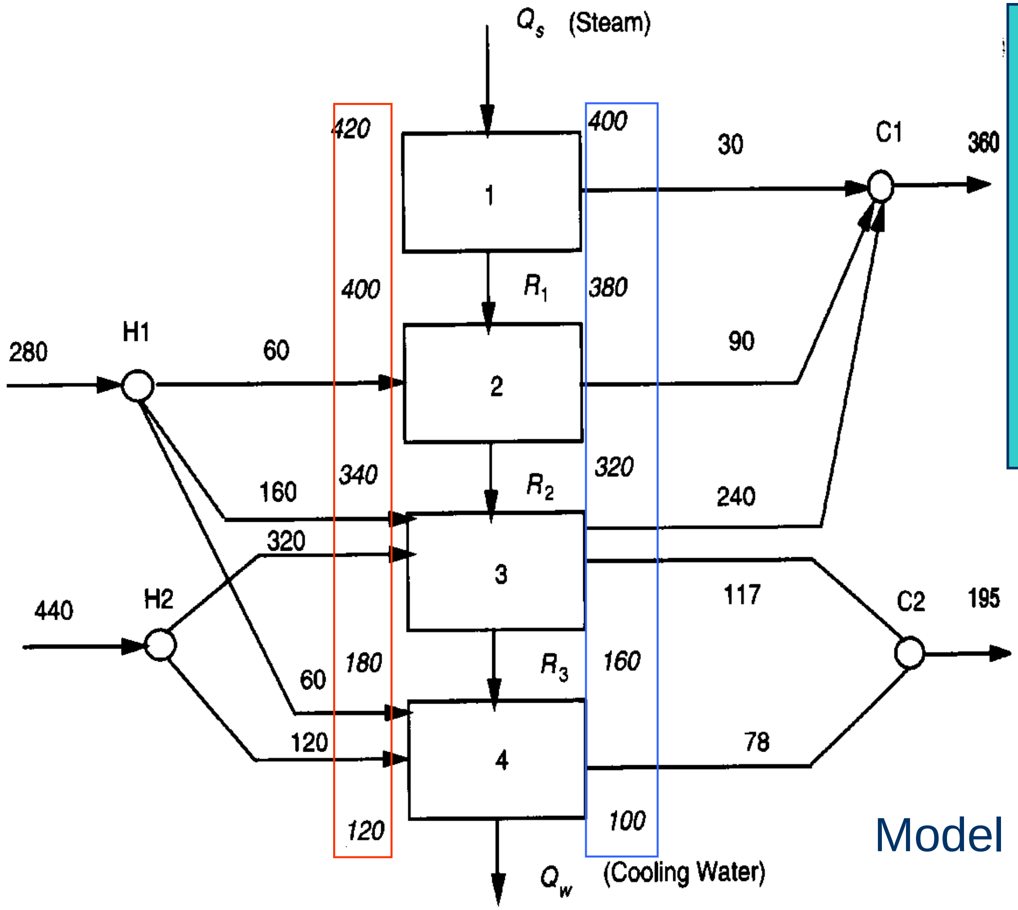
Temperature intervals



Heat content

	H1	H2	C1	C2
int 1			30	
int 2	60		90	
int 3	160	320	240	117
int 4	60	120		78
	280	440	360	195

Utility requirement in the heat exchanger network using LP



Balance in each temp. interval

$$R_1 + 30 = Q_s$$

$$R_2 + 90 = R_1 + 60$$

$$R_3 + 357 = R_2 + 480$$

$$Q_w + 78 = R_3 + 180$$

Model of Papoulias and Grossmann

Utility requirement in the heat exchanger network using LP

Linear programming problem

$$\min Z = Q_s + Q_w$$
$$s.a.$$
$$R_1 - Q_s = -30$$
$$R_2 - R_1 = -30$$
$$R_3 - R_2 = 123$$
$$Q_w - R_3 = 102$$
$$Q_s, Q_w, R_1, R_2, R_3 \geq 0$$

SOLUTION

Q_s = 60kW

Q_w = 225 kW

R_1 = 30 kW

R_2 = 0

R_3 = 123 kW

R_2 = 0 there is a pinch in the temperature interval 340º-320ºC

Software

- There are two main families of LP software :
  - **Solvers** : routines that implement algorithms and can be called from another software environment or language as dll's and provide the LP optimum (CPLEX, LINDO, OSL, Matlab, NAG,...)
  - **Modelling systems**: Software environments that facilitate the description, solution, analysis and management of the LP problem. They allow to formulate the problem in a certain language (CPLEX, GAMS, XPRESS-MP, AIMMS, GUROBI,...) or structure (Excel). They call different solvers for finding the optimum.
- The size of a problem very often is expressed as the number of non-zero elements of the A matrix (sparsity) A LP problem with less that 1000 non-zero elements is considered a small problem, and one with more than 50000 a big one. Modern solvers can solve this problems in short times, from 1 second to one hour.