The simplex algorithm

A solution to linear programming

Hugues Talbot

Centrale Supélec
Centre de Vision Numérique

March 28, 2018

Outline

Matrices, inverses, etc

- To follow this course it is mandatory to know Linear Algebra at a basic level: matrix manipulation, addition, multiplication, vector space, inverse, etc.
- In practice, the largest matrix I may ask you to invert by hand is 3×3.
- A complete course on linear algebra : `http://joshua.smcvt.edu/linearalgebra/` : 440 pages, free, with all the proofs and solutions to exercises.

Example - belt factory

- A belt factory produces two kinds of leather belts: luxury and standard
- Each kind requires $1m^2$ of leather
- A standard belt requires 1h of work
- A luxury belt requires 2h
- Our weekly resources are $40m^2$ of leather and 60h of work
- Each standard belt generates a profit of 3 Euros
- Each luxury belt generates a profit of 4 Euros.
- Maximize the weekly profit.

Formulation

- $x_1 =$ number of *luxury* belts produced each week
- $x_2 =$ number of *standard* belts produced each week
- Maximize $z = 4x_1 + 3x_2$, with

leather constraint (1)

work constraint (2)

sign constraint (3)

Standard form conversion

- We want to convert all the inequalities into equalities
- For each $\leq$ constraint we define a “slack” variable s_i. All the s_i are positives. Here

$$s_1 = 40 - x_1 - x_2 \tag{4}$$
$$s_2 = 60 - 2x_1 - x_2 \tag{5}$$

- This is the *standard form* of an LP : Maximize z, with :

$$\begin{array}{rcrcrcrcrcr} z & = & 4x_1 & + & 3x_2 & & & & & & \\ & & x_1 & + & x_2 & + & s_1 & & & = & 40 \\ & & 2x_1 & + & x_2 & & & + & s_2 & = & 60 \\ & & & & & & x_1,\ x_2, & s_1, & s_2 & \geq & 0 \end{array}$$

The famous diet problem

- We want to follow a diet (regimen) that imposes to eat from the 4 fundamental groups : chocolate, ice-cream, soda and cake.
- A chocolate bar costs 50 centimes, an ice-cream scoop costs 20 centimes, each soda can costs 30 centimes and a portion of cake costs 80 centimes.
- Each day one must eat 500 calories, 60g of chocolate, 100g of sugar et 80g de lipids.
- The nutritional contribution of each kind of food is given in the following matrix:

	Cal.	Choc. (g)	Sug. (g)	Lip. (g)
Choc. bar	400	30	20	20
Ice cream	200	20	20	40
Soda	150	0	40	10
Cake	500	0	40	50

Formulation

- We want to *minimize* the cost of this diet
- How many variables ?
- Express the objective function
- Express the constraints
- Put the problem in standard form

Formulation – diet

- Objective= minimize $z = 50x_1 + 20x_2 + 30x_3 + 80x_4$
- Total calories = $400x_1 + 200x_2 + 150x_3 + 500x_4 \geq 500$
- Total chocolate = $30x_1 + 20x_2 \geq 60$
- Total sugar = $20x_1 + 20x_2 + 40x_3 + 40x_4 \geq 100$
- Total lipids = $20x_1 + 40x_2 + 10x_3 + 50x_4 \geq 80$
- Also, all the x_i are positive.

Formulation – standard form

- For the $\geq$ constraints, one defines *excess variables* e_i, all positives
- We obtain :

$$
\begin{array}{rcrcrcrcrcrcrcrcrcl}
z & = & 50x_1 & + & 20x_2 & + & 30x_3 & + & 80x_4 & & & & & & & & & & \\
 & & 400x_1 & + & 200x_2 & + & 150x_3 & + & 500x_4 & -e1 & & & & & & & & = & 500 \\
 & & 30x_1 & + & 20x_2 & & & & & & & -e2 & & & & & & = & 60 \\
 & & 20x_1 & + & 20x_2 & + & 40x_3 & + & 40x_4 & & & & & -e3 & & & & = & 100 \\
 & & 20x_1 & + & 40x_2 & + & 10x_3 & + & 50x_4 & & & & & & & -e4 & & = & 80
\end{array}
$$

with the x_i et e_i all positive

- With mixed constraints (i.e. both $\leq$ et $\geq$) we have both s_i et e_i.
- The s_i and e_i variables have the same status as the x_i variables.

General standard form

- We suppose we have a problem with m constraints and n variables in standard form.
- The form of the problem is the following *minimize* (or *maximize*) z with

$$
\begin{array}{rcrcrcrcrcl}
z = & & c_1x_1 & + & c_2x_2 & + \dots + & c_nx_n & & \\
& & a_{11}x_1 & + & a_{12}x_2 & + \dots + & a_{1n}x_n & = & b_1 \\
& & a_{21}x_1 & + & a_{22}x_2 & + \dots + & a_{2n}x_n & = & b_2 \\
& & \vdots & & \vdots & & \vdots & & \\
& & a_{m1}x_1 & + & a_{m2}x_2 & + \dots + & a_{mn}x_n & = & b_m
\end{array}
$$

et $\forall i, x_i \geq 0$

Principal Matrix

We define :

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

Note : we have $n \geq m$, otherwise the system is over-constrained

Matrix of the variables and constraints

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}.$$

LP in matrix form

The linear program can be written in matrix form:

$$\text{min (ou max) } \mathbf{c}^T \mathbf{x} \tag{6}$$

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{7}$$

$$\mathbf{x} \geq 0 \tag{8}$$

Basis variables

- A *Basis* is a regular sub-matrix of $\mathbf{A}$. The rank of the matrix $\mathbf{A}(m, n)$ must be exactly m.
- A *basis solution* is obtained by setting $n - m$ variables to 0, and by resolving the problem for the m remaining variables, called the *in-basis variables* (IBV).
- The $n - m$ variables set to 0 are called the *non-basis variables* (NBV).
- Varying choices of IBV/NBV yield different *basis solutions*
- Note: we always reorder the basis variables to the left of the matrix.

Representation

x_b^t	x_e^t

c_b^t	c_e^t

B basis	E non-basis
m columns	$n - m$ columns

Matrix representation

- We have

$$\mathbf{A} = [\mathbf{B}\mathbf{E}]\,, \mathbf{x} = \begin{bmatrix} x_b \\ x_e \end{bmatrix}, \mathbf{c}^T = \begin{bmatrix} \mathbf{c}_b{}^T \mathbf{c}_e{}^T \end{bmatrix}$$

- Which yields
  $z = \mathbf{c}^T\mathbf{x} = \mathbf{c}_b{}^T\mathbf{x}_b + \mathbf{c}_e{}^T\mathbf{x}_e$
  $\mathbf{A}\mathbf{x} = \mathbf{b} \rightarrow \mathbf{B}\mathbf{x}_b + \mathbf{E}\mathbf{x}_e = \mathbf{b}$
- A *basis* solution is such that

$$\mathbf{x}_e = 0 \tag{9}$$

$$\mathbf{B}\mathbf{x}_b = \mathbf{b} \tag{10}$$

$$\mathbf{x}_b = \mathbf{B}^{-1}\mathbf{b} \tag{11}$$

Example

- Consider the following system

$$\begin{array}{rcrcrcr} x_1 & + & x_2 & & & = & 3 \\ & & -x_2 & + & x_3 & = & -1 \end{array}$$

- If we specify NBV $= \{x_3\}$, then IBV $= \{x_1, x_2\}$. We solve for these, we obtain

$$\begin{array}{rcrcr} x_1 & + & x_2 & = & 3 \\ & & -x_2 & = & -1 \end{array}$$

  Which yields $x_1 = 2$ et $x_2 = 1$.
- Some choices of variables may not yield a basis solution.

Feasible basis solution

- A basis solution is said to be *feasible* (FBS) if

$$\mathbf{x}_b = \mathbf{B}^{-1}\mathbf{b} \geq 0$$

- If the vector $\mathbf{x}_b$ contains null terms, we call this solution a degenerate basis solution.

FBS example - 1

- Consider the problem :

$$\begin{array}{lrcrcl} \text{min} & - & x_1 & - & 2x_2 & \\ \text{with} & & & & & \\ & & x_1 & + & 2x_2 & \leq 4 \\ & & 2x_1 & + & x_2 & \leq 5 \\ & & x_1 & , & x_2 & \geq 0 \end{array}$$

- In standard form, we have

$$\begin{array}{lrcrcrcrcl} \text{min} & - & x_1 & - & 2x_2 & & & & & \\ \text{with} & & & & & & & & & \\ & & x_1 & + & 2x_2 & + & x_3 & & & = 4 \\ & & 2x_1 & + & x_2 & & & + & x_4 & = 5 \\ & & x_1 & , & x_2 & , & x_3 & , & x_4 & \geq 0 \end{array}$$

FBS - 2

- We can try a basis with B = {1, 3},

$$\mathbf{B} = [\mathbf{A}_1 \mathbf{A}_3] = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}$$

$$\mathbf{E} = [\mathbf{A}_2 \mathbf{A}_4] = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}$$

$$\mathbf{x}_b = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}, \mathbf{x}_e = \begin{bmatrix} x_2 \\ x_4 \end{bmatrix}$$

- The inverse of $\mathbf{B}$ exists, so $\mathbf{B}$ correspond to a basis

$$\mathbf{B}^{-1} = \begin{bmatrix} 0 & 1/2 \\ 1 & -1/2 \end{bmatrix}$$

FBS -3

- The corresponding basis solution is therefore

$$\mathbf{x}_b = \mathbf{B}^{-1}\mathbf{b} = \begin{bmatrix} 0 & 1/2 \\ 1 & -1/2 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \end{bmatrix} = \begin{bmatrix} 5/2 \\ 3/2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} > 0$$

- This solution is indeed a FBS.

Fundamental theorems

Theorem

The feasible region for any linear programming problem is convex. If a LP has an optimal solution, then an extremal point of this region must be optimal.

Theorem

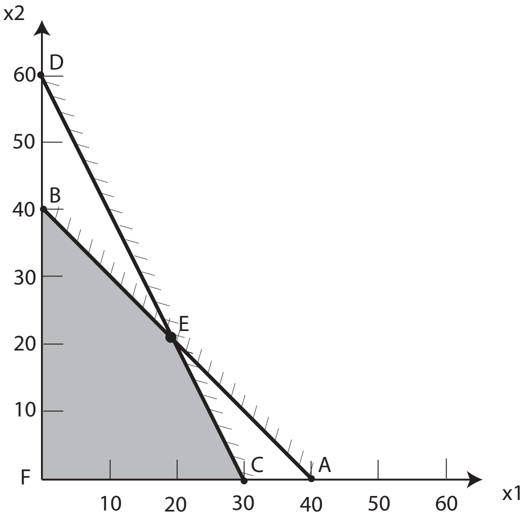
for every LP, there exists a unique extremal point of the feasible region that correspond to every feasible basis solution. Also there exists at least one feasible basis solution that corresponds to every extremal point of the feasible region

Illustration of the theorems

We revisit the leather belt problem, i.e. maximize z, under :

$$\begin{aligned} z = 4x_1 + 3x_2 & \\ x_1 + x_2 + s_1 \qquad &= 40 \\ 2x_1 + x_2 \qquad + s_2 &= 60 \\ x_1,\ x_2,\ s_1,\ s_2 &\geq 0 \end{aligned}$$

Leather belts- 1



Leather belts - 2

We have the equivalence between FBS and the following extremal points:

Basis	Non-base	FBS	Extremal point
x_1, x_2	s_1, s_2	$s_1 = s_2 = 0, x_1 = x_2 = 20$	E
x_1, s_1	x_2, s_2	$x_2 = s_2 = 0, x_1 = 30, s_1 = 10$	C
x_1, s_2	x_2, s_1	$x_2 = s_1 = 0, x_1 = 40, s_2 = -20$	Non feasible, $s_2 < 0$
x_2, s_1	x_1, s_2	$x_1 = s_2 = 0, s_1 = -20, x_2 = 60$	Non feasible, $s_1 < 0$
x_2, s_2	x_1, s_1	$x_1 = s_1 = 0, x_2 = 40, s_2 = 20$	B
s_1, s_2	x_1, x_2	$x_1 = x_2 = 0, s_1 = 40, s_2 = 60$	F

Proof

Let $\mathbf{x}$ be a FBS, of the form $\mathbf{x} = \{x_1, x_2, \dots, x_m, 0, 0, \dots, 0]^T$. If $\mathbf{x}$ is not an extremal point, there exists two points (solutions) α and β both distinct of $\mathbf{x}$ and a scalar λ such that:

$$\mathbf{x} = \lambda\alpha + (1-\lambda)\beta, 0 < \lambda < 1$$

in other words

$$\alpha = [\alpha_1, \alpha_2, \dots, \alpha_m, \alpha_{m+1}, \dots, \alpha_n]^T = \begin{bmatrix} \alpha_b \\ \alpha_e \end{bmatrix}$$

$$\beta = [\beta_1, \beta_2, \dots, \beta_m, \beta_{m+1}, \dots, \beta_n]^T = \begin{bmatrix} \beta_b \\ \beta_e \end{bmatrix}$$

Therefore

$$\lambda\alpha_i + (1-\lambda)\beta_i = 0 \forall i \in [m+1, n]$$

Since $\lambda > 0, (1-\lambda) > 0, \alpha_i > 0$ and $\beta_i > 0$, we have $\alpha_i = \beta_i = 0$, in other words $\mathbf{x} = \alpha = \beta$, which is a contradiction.

Number of possible solutions

- The number of candidate bases is $C_n^m = \frac{n!}{(n-m)!m!}$. All the candidate bases are not invertible, so this is a higher bound.
- Exploring all the extremal point would be *non-polynomial*
- Experimentally, we can explore n variables with m constraints, in such a way that a solution is found on average in fewer than $3m$ operations.

Adjacent FBS

- For all LP problem, two FBS are *adjacent* if their respective basis variable set have exactly $m - 1$ variables in common.

A geometrical interpretation is that two FBS are adjacent if they are linked by a single edge of the feasible polytope.

General description of the algorithm

The simplex algorithm follows these steps:

1. Find a FBS for the LP, called the initial FBS;
2. find if the current FBS is optimal. If yes stop ; if not find an adjacent FBS that has a better z;
3. go to (2) with the new FBS as current FBS.

The remaining questions are: how to detect optimality and how to move along an edge of the feasible polytope.

Reduced costs

- For any FBS, we can write:

$$z = {\mathbf{c}_b}^T \mathbf{x}_b + {\mathbf{c}_e}^T \mathbf{x}_e$$

  and

$$\mathbf{B}\mathbf{x}_b + \mathbf{E}\mathbf{x}_e = \mathbf{b}$$

- Therefore

$$\mathbf{x}_b = \mathbf{B}^{-1}(\mathbf{b} - \mathbf{E}\mathbf{x}_e)$$

- Subsituting

$$z = {\mathbf{c_b}}^T \mathbf{B}^{-1}\mathbf{b} + ({\mathbf{c_e}}^T - {\mathbf{c_b}}^T \mathbf{B}^{-1}\mathbf{E})\mathbf{x_e}$$

- We set :

$$\bar{\mathbf{c}}_e^T = ({\mathbf{c_e}}^T - {\mathbf{c_b}}^T \mathbf{B}^{-1}\mathbf{E})$$

Reduced costs - 2

- For this FBS, $\mathbf{x}_e = 0$, however this second term corresponds to an increase in cost for an augmentation of the variables in $\mathbf{x}_e$.
- If all the costs are negative (for a maximization), any increase of the variables in $\mathbf{x}_e$ will reduce the value of z, and so the current value is necessarily optimal.
- Conversely for a minimization
- So we have an effective optimality test.

Example

- In the case of the leather belts, considering the FBS=$\{s_1, s_2\}$ with the IBS=$\{x_1, x_2\}$.
- In this case, ${\mathbf{c_b}}^T = [0\ 0]$, we have very simply $\bar{\mathbf{c}}_e^T = {\mathbf{c_e}}^T = [4\ 3]$
- in order to augment z most efficiently, we must let x_1 enter the basis, since its coefficient is the highest
- We must still decide which variable should exit the basis. In order to achieve this, we must augment x_1 while keeping x_2 at zero, and see which basis variable becomes zero first.
- In our case s_2 becomes zero the first (see drawing). This is the variable that must exit the basis.
- If we do not do this correctly, we end up with a non-feasible basis.

Improving a basis solution

- Considering a maximization, if our basis is such that $\bar{\mathbf{c}}_e^T$ is not strictly negative or zero, then there exists a variable x_k in $\mathbf{x}_e$ so that $\bar{c}_k > 0$. Augmenting x_k may then improve z.
- **Change of basis** If for a variable x_k of $\mathbf{x}_e$, $\bar{c}_k > 0$, the solution may be improved by augmenting x_k.

$$\mathbf{x}_b = \mathbf{B}^{-1}(\mathbf{b} - \mathbf{A}_k x_k - \mathbf{E}'\mathbf{x}'_e)$$

By fixing $\mathbf{x}'_e = 0$, and by varying x_k only :

$$\mathbf{x}_b = \mathbf{B}^{-1}(\mathbf{b} - A_k x_k) = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{A}_k x_k \tag{12}$$

$$\mathbf{x}_b = \bar{\mathbf{b}} - \mathbf{P} x_k \tag{13}$$

Augmentation

- As originally x_k is zero, we can only increase its value. There are two cases:
- **case 1** $\forall i, P_i \leq 0$. In this case the solution is unbounded: x_k tends to $+\infty$ and z towards $-\infty$ (for a minimization) or $+\infty$ for a maximization.
- **case 2**, there are two possibilities for each i :
  1. if $P_i \leq 0$, $x_{bi} \geq 0$ for all $x_k \geq 0$, this is a non-critical case: this variable cannot be used in any new basis.
  2. if $P_i > 0$, then $x_{bi} \leq 0$ for $x_k \geq \bar{b_i}/P_i$, therefore for for all $P_i > 0$, there exists a maximum value of $x_k = \bar{b_i}/P_i$, allowing $x_b \geq 0$.
     We choose the variable l so that:

$$l = \text{min}_{i/P_i>0}\left[\frac{\bar{b_i}}{P_i}\right]$$

In summary

- We have shown an algorithm that uses linear algebra instead of graphical intuition.
- We need to be able to model a problem
- We need to understand the simplex algorithm
- We need to be able to make it run on simple problems.