Dr. Z.'s Overview of the Various methods for solving Linear Programming Problems

By Doron Zeilberger

Version of March 30, 2019 (correcting a typo pointed out by William Wong (he won a dollar), and correcting a more serious error, (how to get the RHS at the reconstructive surgery problem) pointed out by Amber Rawson (who won five dollars).

The **general** linear programming problem is

{Maximize, Minimize} Some LINEAR expression in a **set of variables**,

subject to a set of '$\leq$' and/or '$\leq$' and/or '$=$' **constraints**.

(Usually, and definitely in **standard form**, you insist that all the variables are ≥ 0, but sometimes they are 'unrestricted'.)

If there are only two variables, then the quickest way is to use the **graphical method**. Each inequality

$$ax + by \leq c \quad OR \quad ax + by \geq c$$

gives rise to a line in the xy plane, $ax + by = c$, and the region corresponding to the set of points satisfying the inequality is on one of the half planes that the line borders.

To find out which, just pick a random point on either side and see whether the inequality is satisfied.

The **intersection** (common region) of these half-planes is always a **convex polygon** (usually inside the positive quadrant, i.e. in the region where $x \geq 0, y \geq 0$). Its vertices are the **extreme points**. There are only finitely many of them (usually not too many).

These points are the **finalists**. Once you have found them, you plug into the **goal function** and whoever gives the maximal (or minimal, as the case may be) **value**, produces the **optimal solution**, and the value at that record is the **optimal value**.

If the number of variables is three, then you can still do it graphically, with planes, bordering half-spaces that reside in 3D space, but you need a good 3D vision, and I don't recommend it (unless you are requested to do it.)

The graphical method is described in section 1.4 of the textbook (Beck and Kolman, "Introduction to Linear Programming".)

Another method that **always** works, but is not very efficient, is the algebraic method of section 1.5. Like in the simplex algorithm, you must first bring it to **canonical form** by creating **slack variables**, and insisting that all the variables are non-negative (by doing the 'pre-processing': if the condition is $x_i \leq 0$ make a change of variable $x_i \to -x_i$, if the condition is 'x_i unrestricted', replace x_i by two new variables, and write $x_i = x_i' - x_i''$).

Once you do it, the problem is

Maximize or Minimize $\mathbf{c}^T\mathbf{x}$ subject to $A\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$, where A is an $m \times s$ matrix, $\mathbf{c}$ and $\mathbf{x}$ are $s \times 1$ column vectors and $\mathbf{b}$ is a $m \times 1$ column vector.

Since, by a general theorem, an **optimal solution** is always an **extreme point**, and an extreme point always has at least $s - m$ zeros, the 'haystack' is finite, and this gives rise to the algebraic algorithm.

It goes as follows.

• list all $\binom{s}{m}$ possible m-element subsets of the set of variables $\{x_1, \ldots, x_s\}$. For each such choice, find its **complement** (this set is called the set of **non-basic** variables). Now, for each such choice (there are $\binom{s}{m}$ of them) in your system of equations, set all the $s - m$ non-basic variables to 0, getting a system of m equations for the remaining m basic variables. Solve the system. Each such solution is called a **basic solution**. Then see whether there are negative numbers among them. If there are, then it is an **infeasible** basic solution, and you kick it out. The survivors are the set of **basic feasible solutions**, and as before you do the final contest, and plug-in the goal function (this works also for minimization).

A very simple example of using the algebraic method

Maximize $z = x_1 + x_2 + x_3$ subject to the constraints

$$x_1 + 2x_2 - 3x_3 = 6 \quad , \quad x_1 \geq 0 \quad , \quad x_2 \geq 0 \quad , \quad x_3 \geq 0 \quad .$$

Solution using the algebraic approach (section 1.5): There are 3 variables and one equality, hence $s = 3$ and $m = 1$ and there are $\binom{3}{1} = \frac{3!}{1!2!} = 3$ possibilities.

• Basic $\{x_1\}$, Non Basic $\{x_2, x_3\}$. Plugging-in $x_2 = 0, x_3 = 0$ we get the 'system of equations' $\{x_1 = 6\}$ in the 'set of variables' $\{x_1\}$ whose solution is $x_1 = 6$. Hence a basic solution is $(x_1, x_2, x_3) = (6, 0, 0)$. This is feasible. since everything is non-negative.

• Basic $\{x_2\}$, Non Basic $\{x_1, x_3\}$. Plugging-in $x_1 = 0, x_3 = 0$ we get the 'system of equations' $\{2x_2 = 6\}$ in the 'set of variables' $\{x_2\}$ whose solution is $x_2 = 3$. Hence a basic solution is $(x_1, x_2, x_3) = (0, 3, 0)$. This is feasible. since everything is non-negative.

• Basic $\{x_3\}$, Non Basic $\{x_1, x_2\}$. Plugging-in $x_1 = 0, x_2 = 0$ we get the 'system of equations' $\{-3x_3 = 6\}$ in the 'set of variables' $\{x_3\}$ whose solution is $x_3 = -2$. Hence a basic solution is $(x_1, x_2, x_3) = (0, 0, -2)$. This is **not feasible**. since one of the coordinates is negative.

Hence the set of **finalists** is $\{(6, 0, 0), (0, 3, 0)\}$. Now we examine them

$z(6, 0, 0) = 6 + 0 + 0 = 6$,

$z(0, 3, 0) = 0 + 3 + 0$.

Since 6 is the largest, the **optimal solution** is $(x_1, x_2, x_3) = (6, 0, 0)$ and the **optimal value** is 6.

The Amazing Simplex Algorithm

The above algorithm is **effective** (it always terminates) but is **not efficient** (when s and m are large $s!/(m!(s-m)!)$ are usually [unless s is close to m] very large). The reason that it is so inefficient is that we always stick to the same problem. When we solve a system of linear equations in linear algebra, we use the Gauss-Jordan algorithm, to transform a complicated system to simpler and simpler systems, until the system is so simple that we can solve it by glancing. That the point of the three elementary row operations. In the simplex algorithm we use a similar approach, but now we have an extra complication with maximizing z, the goal function. We also have the important notion of **basic variables**.

Each of these intermediate problems are "encoded" using **simplex tableaux**, staring with the **initial simplex tableau**.

The first thing to do if the goal function is

$$z = c_1 x_1 + \ldots + c_s x_s \quad ,$$

is write it as

$$-c_1 x_1 - \ldots - c_s x_s + z \;=\; 0 \quad .$$

You treat z as an equal-rights variable.

The simplex algorithm is very fussy, it requires that

• The numbers in the right hand sides are all **non-negative**

• Each and every equality has a variable that **only shows up** in it and nowhere else (**including not showing up in the objective row!**)

That special variable, is called the **basic variable** for that row (alias equality).

• The coefficient of that basic variable is **positive** (so it can be made 1 by rescaling, i.e. dividing by it), since it is positive, the RHS will not get ruined.

Correction to what I said in class: In class, when I said that it is stupid to introduce artificial variables if you don't have to, I gave an example of the following kind:

Maximize $z = x_1 + x_2 + x_3$ subject to the following constraints

$$x_1 + 2x_2 = 4 \quad ,$$

$$x_1 + 2x_2 + x_3 = 5 \quad ,$$

$$x_1 \geq 0 \quad , \quad x_2 \geq 0 \quad , \quad x_3 \geq 0 \quad .$$

I said (wrongly!) that since x_3 only shows up in the second equation it can be taken as the basic variable for that equation. This is **NONSENSE**. It is true that x_3 does not show up in the first equation, but it does show up in the goal equation, so we need to introduce artificial variables for both equalities, and use the big-M method with **two** artificial variables.

On the other hand, if the problem would have been

Maximize $z = x_1 + x_2$ subject to

$$x_1 + 2x_2 = 4 \quad ,$$

$$x_1 + 2x_2 + x_3 = 5 \quad ,$$

$$x_1 \geq 0 \quad , \quad x_2 \geq 0 \quad , \quad x_3 \geq 0 \quad ,$$

then it would have been OK to take x_3 to be the basic variable of the second equation, since it shows up neither in the first equation nor in the equation for $z = x_1 + x_2$.

Once you write the system in the tableau format, then a basic variable should be all 0 in its column except for 1 at the row for which it serves as the basic variable. At any step, you should check that the current simplex tableau has the property that each row has an entry that is 1 and all the other entries in its column are 0.

The reason that we often do not need to introduce artificial variables is that *in real life*, the problems thar are inputted into the simplex algorithm came from a linear programming problem in **standard form**

$$A\mathbf{x} \leq \mathbf{b} \quad , \quad \mathbf{x} \geq \mathbf{0}.$$

Using the method of transforming it to **canonical form**, by introducing **slack variables**, we get **automatically** that the slack variables (by construction!) may serve as basic variables for the **initial tableau**, since each equation has its own slack variable (that is not shared with the other equations, and of course, is not part of the goal expression).

But watch out, this is true only if the entries of the RHS vector, $\mathbf{b}$, are all positive entries. Whenever b_i is positive, indeed, you should take the slack variable as the basic variable for that row, but if b_i happens to be negative, too bad. In addition to the slack variable that is already there, you need a brand-new artificial variable.

An example of using the big-M method

Consider the problem:

Maximize $z = x_1 + 2x_2$ subject to

$$x_1 + x_2 \leq 2 \quad ,$$

$$x_1 + 2x_2 \geq 4 \quad ,$$

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

Before we can use the simplex method, we must introduce slack variables, x_3 and x_4, and write it as

Maximize $z = x_1 + 2x_2$ subject to

$$x_1 + x_2 + x_3 = 2 \quad ,$$

$$x_1 + 2x_2 - x_4 = 4 \quad .$$

$$x_1 \geq 0 \quad , \quad x_2 \geq 0 \quad , \quad x_3 \geq 0 \quad , \quad x_4 \geq 0 \quad .$$

Note that x_4, in the second equation, has a minus sign, since it came from a $\geq$ inequality.

Since x_3 only shows up in the first equation, and it is neither in the second, nor in the goal equation, it may serve as the basic variable of the first equation, since its coefficient is positive (namey 1). On the other hand, the variable x_4 may **not** serve as the basic variable of the second equation, even though it is exclusive to the second equation, since its coefficient is negative, hence we need to introduce an artificial variable for the second equation, let's call it y_1.

Using the big-M method, our new problem is

Maximize $z = x_1 + 2x_2 - My_1$ subject to

$$x_1 + x_2 + x_3 = 2 \quad ,$$

$$x_1 + 2x_2 - x_4 + y_1 = 4 \quad .$$

$$x_1 \geq 0 \quad , \quad x_2 \geq 0 \quad , \quad x_3 \geq 0 \quad , \quad x_4 \geq 0 \quad , \quad y_1 \geq 0 \quad .$$

Since y_1 is going to be the basic variable of the second equation, it may not show up in the goal equation, so the **first** thing to do is get rid of it, by using

$$y_1 \;=\; 4 - x_1 - 2x_2 + x_4 \quad .$$

We get

$$z = x_1 + 2x_2 - My_1 = x_1 + 2x_2 - M(4 - x_1 - 2x_2 + x_4) = x_1 + 2x_2 - 4M + Mx_1 + 2Mx_2 - Mx_4$$

$$= \; (M+1)x_1 + (2M+2)x_2 - 4M + 0 \cdot x_3 - Mx_4 \quad ,$$

Moving everything, except for the number $(-4M)$ to the left, we get

$$-(M+1)x_1 - (2M+2)x_2 + 0 \cdot x_3 + Mx_4 + z = -4M \quad .$$

Now we are ready to set-up the *initial tableau*

$$\begin{array}{c|cccccc|c} BASIC & x_1 & x_2 & x_3 & x_4 & y_1 & z & RHS \\ & & & & & & & \\ x_3 & 1 & 1 & 1 & 0 & 0 & 0 & 2 \\ y_1 & 1 & 2 & 0 & -1 & 1 & 0 & 4 \\ & & & & & & & \\ & -(M+1) & -(2M+2) & 0 & M & 0 & 1 & -4M \end{array} .$$

Now we use the simplex algorithm, keeping in mind that M is a HUGE (but fixed!) number.

The **most negative** entry at the bottom row is $-(2M+2)$, belonging to the x_2 column. Hence the **entering variable** is x_2. To decide on the **departing variable** we form the $\theta-$ratios, $2/1 = 2$ and $4/2$. They are both the same, so we can take either x_3 or y_1 as the departing variables. Let's take x_3. The pivot entry is the $(1,2)$ entry, and since it is 1 it needs no scaling. Putting arrows we have

$$\begin{array}{c|cccccc|c} BASIC & x_1 & x_2^{\downarrow} & x_3 & x_4 & y_1 & z & RHS \\ & & & & & & & \\ ^{\leftarrow}x_3 & 1 & 1 & 1 & 0 & 0 & 0 & 2 \\ y_1 & 1 & 2 & 0 & -1 & 1 & 0 & 4 \\ & & & & & & & \\ & -(M+1) & -(2M+2) & 0 & M & 0 & 1 & -4M \end{array} .$$

To make x_2 a basic variable, we have to make the $(2,2)$ and $(3,2)$ entries 0. To that end we perform the elementary row operations

$r_2 - 2r_1 \to r_2$ and $r_3 + (2M+2)r_1 \to r_3$ getting the new simplex tableau

$$\begin{array}{c|cccccc|c} BASIC & x_1 & x_2 & x_3 & x_4 & y_1 & z & RHS \\ & & & & & & & \\ x_2 & 1 & 1 & 1 & 0 & 0 & 0 & 2 \\ y_1 & -1 & 0 & -2 & -1 & 1 & 0 & 0 \\ & & & & & & & \\ & (M+1) & 0 & 2M+2 & M & 0 & 1 & 4 \end{array} .$$

In the new tableau there are no more negative entries, hence it is the **final tableau**, and the **optimal solution** is $x_2 = 2$ and $y_1 = 0$, and of course $x_1 = 0$, $x_3 = 0$ and $x_4 = 0$ (since x_1, x_3, x_4 are non-basic variables). Note that at the end, the artificial variable, y_1 is 0 (as it should!, if that does not happen, you messed up). Going back to the original problem, before we introduced slack and artificial variables, we only care about x_1 and x_2, hence the **optimal solution** to the original problem is $x_1 = 0$, $x_2 = 2$, and the **optimal value** is 4.

How to Reconstruct the Simplex Tableau at ANY stage (inluding the final one) From the Initial Tableau?

Consider the initial tableau (Tableau 2.13 in the Kolman-Beck book, page 124)

$BASIC$	x_1	x_2	x_3	x_4	x_5	z	RHS
x_3	1	-1	1	0	0	0	2
x_4	2	1	0	1	0	0	4
x_5	-3	2	0	0	1	0	6
	-5	-3	0	0	0	1	0

.

Suppose that at the final (or any intermediate) stage the BASIC column is something, in that order $x_{i_1}, x_{i_2}, \ldots, x_{i_m}$.

Comments: What we have here is an **algorithm** that have

inputs

- The initial tableau
- The righmost column of the current tableau

Output

- the full current tableau

(Note that the 'current tableau' may be the final tableau, but this algorithm applies to any intermediate tableau (and even to the initial tableau, but then there is nothing to do)

In other words, you are told (that's part of the input) that the left-most "BASIC" column of the current tableau is

$BASIC$	x_1	x_2	x_3	x_4	x_5	z	RHS
x_{i_1}						0	
x_{i_2}						0	
x_{i_3}						0	
						1	

.

and you have to *reconstruct* this tableau without actually peforming the simplex algorithm , **only** using linear algebra.

We call the **columns** $A_1, A_2, A_3, A_4, A_5, \ldots$.

Step 1: Form the $m \times m$ B matrix consising, **in that order**, of columns $A_{i1}, \dots, A_{i_m}$

$$B = \mathbf{A}_{i_1} \dots \mathbf{A}_{i_m}$$

Also from the $m \times 1$ column vector (gotten from $\mathbf{c}$)

$$\mathbf{c}_B = \begin{bmatrix} c_{i_1} \\ \\ . \\ . \\ . \\ c_{i_m} \end{bmatrix} \quad .$$

First, you must use linear algebra to find B^{-1}. Now you can resconstuct all the columns of the current tableau.

To get $\mathbf{t}_j$, the j-th column of the **current tableau** you do

$$\mathbf{t}_j = B^{-1}\mathbf{A}_j \quad .$$

To get the current RHS, you do

$$B^{-1}\mathbf{b} \quad ,$$

And finally to get the j-th entry at the **bottom line** you do

$$\mathbf{c}_B^T \mathbf{t}_j - c_j$$

(recall that the original entries at the bottom line were $-c_j$).

Let's reconstuct tableau 2.15a (p. 125) using this method, by only looking at the BASIC column.

$BASIC$	x_1	x_2	x_3	x_4	x_5	z	RHS
x_3						0	
x_1						0	
x_2						0	
						1	

.

It is

$$\begin{bmatrix} x_3 \\ x_1 \\ x_2 \end{bmatrix}$$

Note: One student asked me "how come you knew the basic variables column"?, I did not! This is part of input for this problem. It so happened that I "stole" it from the book, and it happens to be a final tableau, but this fact is irrelevant.

The matix B is the matrix consisting, **in that order**, of the third, first and second columns of the initial tableau. So

$$B = \begin{bmatrix} 1 & 1 & -1 \\ 0 & 2 & 1 \\ 0 & -3 & 2 \end{bmatrix}$$

Using linear algebra (you do it!) we get

$$B^{-1} = \frac{1}{7}\begin{bmatrix} 7 & 1 & 3 \\ 0 & 2 & -1 \\ 0 & 3 & 2 \end{bmatrix}$$

We also need, for later.

$$\mathbf{c}_B = \begin{bmatrix} 0 \\ 5 \\ 3 \end{bmatrix}$$

We now have

$$\mathbf{t}_1 = B^{-1}\mathbf{A}_1 = \frac{1}{7}\begin{bmatrix} 7 & 1 & 3 \\ 0 & 2 & -1 \\ 0 & 3 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \\ -3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\mathbf{t}_2 = B^{-1}\mathbf{A}_2 = \frac{1}{7}\begin{bmatrix} 7 & 1 & 3 \\ 0 & 2 & -1 \\ 0 & 3 & 2 \end{bmatrix}\begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{t}_3 = B^{-1}\mathbf{A}_3 = \frac{1}{7}\begin{bmatrix} 7 & 1 & 3 \\ 0 & 2 & -1 \\ 0 & 3 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{t}_4 = B^{-1}\mathbf{A}_4 = \frac{1}{7}\begin{bmatrix} 7 & 1 & 3 \\ 0 & 2 & -1 \\ 0 & 3 & 2 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{7} \\ \frac{2}{7} \\ \frac{3}{7} \end{bmatrix}$$

$$\mathbf{t}_5 = B^{-1}\mathbf{A}_5 = \frac{1}{7}\begin{bmatrix} 7 & 1 & 3 \\ 0 & 2 & -1 \\ 0 & 3 & 2 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{3}{7} \\ -\frac{1}{7} \\ \frac{2}{7} \end{bmatrix}$$

The new RHS is (the original RHS is $\mathbf{b} = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}$)

$$B^{-1}\mathbf{b} = \frac{1}{7}\begin{bmatrix} 7 & 1 & 3 \\ 0 & 2 & -1 \\ 0 & 3 & 2 \end{bmatrix}\begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} = \begin{bmatrix} \frac{36}{7} \\ \frac{2}{7} \\ \frac{24}{7} \end{bmatrix}$$

Finally, the entries at the bottom (objective) row, in order, are

$$\mathbf{c}_B^T\mathbf{t_1} - c_1 = [0 \quad 5 \quad 3] \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} - c_1 = 5 - 5 = 0$$

$$\mathbf{c}_B^T\mathbf{t_2} - c_2 = [0 \quad 5 \quad 3] \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - c_2 = 3 - 3 = 0$$

$$\mathbf{c}_B^T\mathbf{t_3} - c_3 = [0 \quad 5 \quad 3] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - c_3 = 0 - 0 = 0$$

$$\mathbf{c}_B^T\mathbf{t_4} - c_4 = [0 \quad 5 \quad 3] \begin{bmatrix} \frac{1}{7} \\ \frac{2}{7} \\ \frac{3}{7} \end{bmatrix} - c_4 = \frac{19}{7} - 0 = \frac{19}{7}$$

$$\mathbf{c}_B^T\mathbf{t_5} - c_5 = [0 \quad 5 \quad 3] \begin{bmatrix} \frac{3}{7} \\ -\frac{1}{7} \\ \frac{2}{7} \end{bmatrix} - c_5 = \frac{1}{7} - 0 = \frac{1}{7}$$

Finally, calling the rightmost column of the current tableau $\mathbf{x}_B$ (RHS) , the **value of the objective function** (the bottom-right number) is

$$\mathbf{c}_B^T\mathbf{x}_B = [0 \quad 5 \quad 3] \begin{bmatrix} \frac{36}{7} \\ \frac{2}{7} \\ \frac{24}{7} \end{bmatrix} = \frac{82}{7} \quad .$$

Putting everything together, we have that the current (that happens to be the final, but the same method works in general) tableau is

$$\begin{array}{c|cccccc|c} BASIC & x_1 & x_2 & x_3 & x_4 & x_5 & z & RHS \\ & & & & & & & \\ x_3 & 0 & 0 & 1 & \frac{1}{7} & \frac{3}{7} & 0 & \frac{36}{7} \\ x_1 & 1 & 0 & 0 & \frac{2}{7} & -\frac{1}{7} & 0 & \frac{2}{7} \\ x_2 & 0 & 1 & 0 & \frac{3}{7} & \frac{2}{7} & 0 & \frac{24}{7} \\ & & & & & & & \\ & 0 & 0 & 0 & \frac{19}{7} & \frac{1}{7} & 1 & \frac{82}{7} \end{array} \quad .$$