COMP3121/9101/3821/9801 Lecture Notes

Linear Programming

LiC: Aleks Ignjatovic

THE UNIVERSITY OF
NEW SOUTH WALES

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Consider the following problem. You are given a list of food sources $f_1, f_2, \ldots, f_n$; for each source f_i you are given its price per gram p_i, the number of calories c_i per gram, and for each of 13 vitamins $V_1, V_2, \ldots, V_{13}$ you are given the content $v(i,j)$ of milligrams of vitamin V_j in one gram of food source f_i. Your goal is to find a combination of quantities of each food source such that:

1. the total number of calories in all of the chosen food is equal to a recommended daily value of 2000 calories;
2. the total intake of each vitamin V_j is at least the recommended amount of w_j milligrams for all $1 \leq j \leq 13$;
3. the price of all food per day is as small as possible.

To obtain the corresponding constraints let us assume that we take x_i grams of each food source f_i. Then:

1. the total number of calories must satisfy
$$\sum_{i=1}^{n} x_i c_i = 2000;$$
2. for each vitamin V_j the total amount in all food must satisfy
$$\sum_{i=1}^{n} x_i v(i,j) \geq w_j \quad (1 \leq j \leq k);$$
3. an implicite assumption is that all the quantities must be non-negative numbers,
$$x_i \geq 0, \quad 1 \leq i \leq n.$$

Our goal is to minimise the objective function which is the total cost $y = \sum_{i=1}^{n} x_i p_i$. Note that here all the equalities and inequalities, as well as the objective function, are **linear**. This problem is a typical example of a Linear Programming problem.

Assume now that you are the Shadow Treasurer and you want to make certain promises to the electorate which will ensure your party will win in the forthcoming elections. You can promise that you will build certain number of bridges, each 3 billion a piece, certain number of rural airports, each 2 billion a piece and a number of olympic swimming pools each a billion a piece. Each bridge you promise brings you 5% of city votes, 7% of suburban votes and 9% of rural votes; each rural airport you promise brings you no city votes, 2% of suburban votes and 15% of rural votes; each olympic swimming pool promised brings you 12% of city votes, 3% of suburban votes and no rural votes. In order to win, you have to get at least 51% of each of the city, suburban and rural votes. You wish to win the election by making a promise that will cost as little budget money as possible.

We can let the number of bridges to be built be x_b, number of airports x_a and the number of swimming pools x_p. We now see that the problem amounts to minimising the objective $y = 3x_b + 3x_a + x_p$, while making sure that the following constraints are satisfied:

$$\begin{aligned}
0.05x_b \qquad\qquad + 0.12x_p &\geq 0.51 \quad \text{(securing majority of city votes)}\\
0.07x_b + 0.02x_a + 0.03x_p &\geq 0.51 \quad \text{(securing majority of suburban votes)}\\
0.09x_b + 0.15x_a \qquad\qquad &\geq 0.51 \quad \text{(securing majority of rural votes)}\\
x_b, x_a, x_p &\geq 0.
\end{aligned}$$

However, there is a very significant difference with the first example: you can eat 1.56 grams of chocolate but (presumably) you cannot promise to build 1.56 bridges, 2.83 airports and 0.57 swimming pools! The second example is an example of an Integer Linear Programming problem, which requires all the solutions to be integers. Such problems are MUCH harder to solve than the plain Linear Programming problems whose solutions can be real numbers.

We now follow closely $COMP3121$ textbook by Cormen, Leiserson, Rivest and Stein, *Introduction to Algorithms.* We start by defining a common representations of Linear Programming optimisation problems.

In the **standard form** the *objective* to be maximized is given by

$$f(\mathbf{x}) = \sum_{j=1}^{n} c_j\, x_j,$$

and the *constraints* are of the form

$$\begin{aligned}
g_i(\mathbf{x}) &\leq b_i, && 1 \leq i \leq m; && (1)\\
x_j &\geq 0, && 1 \leq j \leq n, && (2)
\end{aligned}$$

where

$$g_i(\mathbf{x}) = \sum_{j=1}^{n} a_{ij} x_j; \qquad 1 \leq i \leq m. \qquad (3)$$

Note that $\mathbf{x}$ represents a vector, $\mathbf{x} = \langle x_1 \dots x_n \rangle^{\intercal}$, and a_{ij} are reals. The above somewhat messy formulation can be reformulated in a much more compact matrix form. To get a very compact representation of linear programs let us introduce a partial ordering on vectors $\mathbf{x} \in \mathbb{R}^n$ by $\mathbf{x} \leq \mathbf{y}$ if and only if such inequalities hold coordinate-wise, i.e., if and only if $x_j \leq y_j$ for all $1 \leq i \leq n$. Recall that vectors are written as a single column matrices; thus, letting $\mathbf{c} = \langle c_1, \dots, c_n \rangle^{\intercal} \in \mathbb{R}^n$ and $\mathbf{b} = \langle b_1, \dots b_m \rangle^{\intercal} \in \mathbb{R}^m$, and letting A be the matrix $A = (a_{ij})$ of size $m \times n$, we get that the above problem can be formulated simply as:

maximize $\mathbf{c}^{\intercal}\mathbf{x}$

subject to the following two (matrix-vector) constraints:[1] $Ax \leq b$ and $x \geq 0$.

[1]Note that the inequality involved in the constraints is the partial ordering on vectors which we have just introduced above. Also, we are slightly abusing the notation because in the non-negativity constraint, 0 actually represents a vector $\mathbf{0} \in \mathbb{R}^n$ with all coordinates equal to 0.

Thus, to specify a Linear Programming optimization problem we just have to provide a triplet (A, b, c), and the information that the triplet represents an LP problem in the standard form.

Wile in general the "natural formulation" of a problem into a Linear Program does not necessarily produce the non-negativity constrains (2) for all of the variables, in the standard form such constraints are indeed required for all of the variables. This poses no problem, because each occurrence of an unconstrained variable x_j can be replaced by the expression $x'_j - x^*_j$ where x'_j, x^*_j are new variables satisfying the constraints $x'_j \geq 0,\ x^*_j \geq 0$.

Any vector x which satisfies the two constraints is called a *feasible solution*, regardless of what the corresponding objective value $c^T x$ might be. Note that the set of feasible solutions, i.e., the domain over which we seek to maximize the objective, is a convex set because it is an intersection of the half spaces defined by each of the constraints.

As an example, let us consider the following optimization problem:

$$\text{maximize} \qquad 3x_1 + x_2 + 2x_3 \tag{4}$$

subject to the constraints

$$x_1 + x_2 + 3x_3 \leq 30 \tag{5}$$
$$2x_1 + 2x_2 + 5x_3 \leq 24 \tag{6}$$
$$4x_1 + x_2 + 2x_3 \leq 36 \tag{7}$$
$$x_1, x_2, x_3 \geq 0 \tag{8}$$

How large can the value of the objective $z(x_1, x_2, x_3) = 3x_1 + x_2 + 2x_3$ be, without violating the constraints? If we add inequalities (5) and (6), we get

$$3x_1 + 3x_2 + 8x_3 \leq 54$$

Since all variables are constrained to be non-negative, we are assured that

$$3x_1 + x_2 + 2x_3 \leq 3x_1 + 3x_2 + 8x_3 \leq 54$$

The far left hand side of this equation is just the objective (4); thus $z(x_1, x_2, x_3)$ is bounded above by 54, i.e., $z(x_1, x_2, x_3) \leq 54$.

Can we obtain a tighter bound? We could try to look for coefficients $y_1, y_2, y_3 \geq 0$ to be used to for a linear combination of the constraints:

$$y_1(x_1 + x_2 + 3x_3) \leq 30y_1$$
$$y_2(2x_1 + 2x_2 + 5x_3) \leq 24y_2$$
$$y_3(4x_1 + x_2 + 2x_3) \leq 36y_3$$

Then, summing up all these inequalities and factoring, we get

$$\begin{aligned} x_1(y_1 + 2y_2 + 4y_3) + x_2(y_1 + 2y_2 + y_3) + x_3(3y_1 + 5y_2 + 2y_3) \\ \leq 30y_1 + 24y_2 + 36y_3 \end{aligned} \tag{9}$$

If we compare this with our objective (4) we see that if we choose y_1, y_2 and y_3 so that:

$$\begin{aligned} y_1 + 2y_2 + 4y_3 &\geq 3 \\ y_1 + 2y_2 + y_3 &\geq 1 \\ 3y_1 + 5y_2 + 2y_3 &\geq 2 \end{aligned}$$

then

$$x_1(y_1 + 2y_2 + 4y_3) + x_2(y_1 + 2y_2 + y_3) + x_3(3y_1 + 5y_2 + 2y_3) \geq 3x_3 + x_2 + 2x_3$$

Combining this with (9) we get:

$$30y_1 + 24y_2 + 36y_3 \geq 3x_1 + x_2 + 2x_3 = z(x_1, x_2, x_3)$$

Consequently, in order to find as tight upper bound for our objective $z(x_1, x_2, x_3)$ we have to look for y_1, y_2, y_3 which produce the smallest possible value of $z^*(y_1, y_2, y_3) = 30y_1 + 24y_2 + 36y_3$, but which do not violate the constraints

$$y_1 + 2y_2 + 4y_3 \geq 3 \tag{10}$$
$$y_1 + 2y_2 + y_3 \geq 1 \tag{11}$$
$$3y_1 + 5y_2 + 2y_3 \geq 2 \tag{12}$$
$$y_1, y_2, y_3 \geq 0 \tag{13}$$

Thus, trying to find the best upper bound for our objective $z(x_1, x_2, x_3)$ obtained by forming a linear combination of the constraints only reduces the original maximization problem to a minimization problem:

minimize $\quad 30y_1 + 24y_2 + 36y_3$
subject to the constraints (10-13)

Such minimization problem P^* is called *the dual problem* of the initial problem P.

Let us now repeat the whole procedure with P^* in place of P, i.e., let us find the dual program $(P^*)^*$ of P^*. We are now looking for non negative multipliers $z_1, z_2, z_3 \geq 0$ to multiply inequalities (10-12) and obtain

$$\begin{aligned} z_1(y_1 + 2y_2 + 4y_3) &\geq 3z_1 \\ z_2(y_1 + 2y_2 + y_3) &\geq z_2 \\ z_3(3y_1 + 5y_2 + 2y_3) &\geq 2z_3 \end{aligned}$$

Summing these up and factoring produces

$$y_1(z_1 + z_2 + 3z_3) + y_2(2z_1 + 2z_2 + 5z_3) + y_3(4z_1 + z_2 + 2z_3) \geq 3z_1 + z_2 + 2z_3 \tag{14}$$

If we choose these multiples so that

$$z_1 + z_2 + 3z_3 \leq 30 \tag{15}$$
$$2z_1 + 2z_2 + 5z_3 \leq 24 \tag{16}$$
$$4z_1 + z_2 + 2z_3 \leq 36 \tag{17}$$

we will have:

$$y_1(z_1 + z_2 + 3z_3) + y_2(2z_1 + 2z_2 + 5z_3) + y_3(4z_1 + z_1 + 2z_3) \leq 30y_1 + 24y_2 + 36y_3$$

Combining this with (14) we get

$$3z_1 + z_2 + 2z_3 \leq 30y_1 + 24y_2 + 36y_3$$

Consequently, finding the dual program $(P^*)^*$ of P^* amounts to maximizing the objective $3z_1 + z_2 + 2z_3$ subject to the constraints (15-17). But notice that, except for having different variables, $(P^*)^*$ is exactly our starting program P! Thus, the dual program $(P^*)^*$ for program P^* is just P itself, i.e., $(P^*)^* = P$.

So, at the first sight, looking for the multipliers y_1, y_2, y_3 did not help much, because it only reduced a maximization problem to an equally hard minimization problem.[2]

To find the maximal value of $3x_1 + x_2 + 2x_3$ subject to the constraints

$$\begin{aligned} x_1 + x_2 + 3x_3 &\leq 30 \\ 2x_1 + 2x_2 + 5x_3 &\leq 24 \\ 4x_1 + x_2 + 2x_3 &\leq 36 \end{aligned}$$

let us start with $x_1 = x_2 = x_3 = 0$ and ask ourselves: how much can we increase x_1 without violating the constraints? Since $x_1 + x_2 + 3x_3 \leq 30$ we can introduce a new variable x_4 such that $x_4 \geq 0$ and

$$x_4 = 30 - (x_1 + x_2 + 3x_3) \tag{18}$$

Since such variable measures how much "slack" we got between the actual value of $x_1 + x_2 + 3x_3$ and its upper limit 30, such x_4 is called a slack variable. Similarly we introduce new variables x_5, x_6 requiring them to satisfy $x_5, x_6 \geq 0$ and let

$$x_5 = 24 - 2x_1 - 2x_2 - 5x_3 \tag{19}$$

$$x_6 = 36 - 4x_1 - x_2 - 2x_3 \tag{20}$$

Since we started with the values $x_1 = x_2 = x_3 = 0$, this implies that these new slack variables must have values $x_4 = 30$, $x_5 = 24$, $x_6 = 36$, or as a single vector, the initial *basic feasible solution* is $(\overset{x_1}{0}, \overset{x_2}{0}, \overset{x_3}{0}, \overset{x_4}{30}, \overset{x_5}{24}, \overset{x_6}{36})$. Note that "feasible" refers merely to the fact that all of the constraints are satisfied.[3]

Now we see that (18) implies that x_1 cannot exceed 30, and (19) implies that $2x_1 \leq 24$, i.e., $x_1 \leq 12$, while (20) implies that $4x_1 \leq 36$, i.e., $x_1 \leq 9$. Since

[2] It is now useful to remember how we proved that the Ford - Fulkerson Max Flow algorithm in fact produces a maximal flow, by showing that it terminates only when we reach the capacity of a (minimal) cut...

[3] Clearly, setting all variables to 0 does not always produce a basic feasible solution because this might violate some of the constraints; this would happen, for example, if we had a constraint of the form $-x_1 + x_2 + x_3 \leq -3$; choosing an initial basic feasible solution requires a separate algorithm to "bootstrap" the *SIMPLEX* algorithm - see for the details our (C-L-R-S) textbbok.

all of these conditions must be satisfied we conclude that x_1 cannot exceed 9, which is the upper limit coming from the constraint (20).

If we set $x_1 = 9$, this forces $x_6 = 0$. We now swap the roles of x_1 and x_6: since we cannot increase x_1 any more, we eliminate x_1 from the right hand sides of the equations (18-20) and from the objective, introducing instead variable x_6 to the right hand side of the constraints and into the objective. To do that, we solve equation (20) for x_1:

$$x_1 = 9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4}$$

and eliminate x_1 from the right hand side of the remaining constraints and the objective to get:

$$\begin{aligned}
z &= 3\left(9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4}\right) + x_2 + 2x_3 \\
&= 27 - \frac{3}{4}x_2 - \frac{3}{2}x_3 - \frac{3}{4}x_6 + x_2 + 2x_3 \\
&= 27 + \frac{1}{4}x_2 + \frac{1}{2}x_3 - \frac{3}{4}x_6 \\
x_5 &= 24 - 2\left(9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4}\right) - 2x_2 - 5x_3 \\
&= 6 + \frac{x_2}{2} + x_3 + \frac{x_6}{2} - 2x_2 - 5x_3 \\
&= 6 - \frac{3}{2}x_2 - 4x_3 + \frac{x_6}{2} \\
x_4 &= 30 - \left(9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4}\right) - x_2 - 3x_3 \\
&= 21 + \frac{x_2}{4} + \frac{x_3}{2} + \frac{x_6}{4} - x_2 - 3x_3 \\
&= 21 - \frac{3}{4}x_2 + \frac{5}{2}x_3 + \frac{x_6}{4}
\end{aligned}$$

To summarise: the "new" objective is

$$z = 27 + \frac{1}{4}x_2 + \frac{1}{2}x_3 - \frac{3}{4}x_6$$

and the "new constraints" are

$$x_1 = 9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{5} \tag{21}$$

$$x_4 = 21 - \frac{3}{4}x_2 - \frac{5}{2}x_3 + \frac{x_6}{4} \tag{22}$$

$$x_5 = 6 - \frac{3}{2}x_2 - 4x_3 + \frac{x_6}{2} \tag{23}$$

Our new basic feasible solution replacing $(\overset{x_1}{0}, \overset{x_1}{0}, \overset{x_3}{0}, \overset{x_4}{30}, \overset{x_5}{24}, \overset{x_6}{36})$ is obtained by setting all the variables on the right to zero, thus obtaining $(\overset{x_1}{9}, \overset{x_2}{0}, \overset{x_3}{0}, \overset{x_4}{21}, \overset{x_5}{6}, \overset{x_6}{0})$.

NOTE: These are EQUIVALENT constraints and objectives; the old ones were only *transformed* to an equivalent form. Any values of the variables will produce exactly the same value in both forms of the objective and they will satisfy the first set of constraints if and only if they satisfy the second set.

So x_1 and x_6 have switched their roles; x_1 acts as a new basic variable, and the new basic feasible solution is: $(\overset{x_1}{9}, \overset{x_2}{0}, \overset{x_3}{0}, \overset{x_4}{21}, \overset{x_5}{6}, \overset{x_6}{0})$; the new value of the objective is $z = 27 + \frac{1}{4}0 + \frac{1}{2}0 - \frac{3}{4}0 = 27$. We will continue this process of finding basic feasible solutions which increase the value of the objective, switching which variables are used to measure the slack and which are on the right hand side of the constraint equations and in the objective. The variables on the left are called *the basic variables* and the variables on the right are the *non basic variables.*

We now choose another variable with a positive coefficient in the objective, say x_3 (we could also have chosen x_2). How much can we increase it?

From (21) we see that $\frac{x_3}{2}$ must not exceed 9, otherwise x_1 will become negative. Thus x_3 cannot be larger than 18. Similarly, $\frac{5}{2}x_3$ cannot exceed 21, otherwise x_4 will become negative, and so $x_3 \leq \frac{42}{5}$; similarly, $4x_3$ cannot exceed 6, ie, $x_3 \leq \frac{3}{2}$. Thus, in order for all constraints to remain valid, x_3 cannot exceed $\frac{3}{2}$. Thus we increase x_3 to $\frac{3}{2}$; equation (23) now forces x_5 to zero. We now eliminate x_3 from the right hand side of the constraints and from the objective, taking it as a new basic variable:

$$4x_3 = 6 - \frac{3}{2}x_2 - x_5 + \frac{x_6}{2}$$

i.e.,

$$x_3 = \frac{3}{2} - \frac{3}{8}x_2 - \frac{1}{4}x_5 + \frac{1}{8}x_6 \tag{24}$$

After eliminating x_3 by substitution using (24), the objective now becomes:

$$\begin{aligned} z =& 27 - \frac{1}{4}x_2 + \frac{1}{2}\left(\frac{3}{2} - \frac{3}{8}x_2 - \frac{1}{4}x_5 + \frac{1}{8}x_6\right) - \frac{3}{4}x_6 = \\ & \frac{111}{4} + \frac{1}{16}x_2 - \frac{1}{8}x_5 - \frac{11}{6}x_6 \end{aligned} \tag{25}$$

Using (24) to eliminate x_3 from the constraints, after simplifications we get the new constraints:

$$x_1 = \frac{33}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \tag{26}$$

$$x_3 = \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} + \frac{x_6}{8} \tag{27}$$

$$x_4 = \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_3}{8} - \frac{x_6}{16} \tag{28}$$

Our new basic solution is again obtained using the fact that all variables on the right and in the objective, including the newly introduced non-basic variable x_5, are equal to zero, i.e., the new basic feasible solution is $(\overset{x_1}{\frac{33}{4}}, \overset{x_2}{0}, \overset{x_3}{\frac{3}{2}}, \overset{x_4}{\frac{69}{4}}, \overset{x_5}{0}, \overset{x_6}{0})$ and the new value of the objective is $z = \frac{111}{4}$.

Comparing this with the previous basic feasible solution $(\overset{x_1}{9}, \overset{x_2}{0}, \overset{x_3}{0}, \overset{x_4}{21}, \overset{x_5}{6}, \overset{x_6}{0})$ we see that in the new basic feasible solution the value of x_1 dropped from 9 to

33/4; however, this now has no effect on the value of the objective because x_1 no longer appears in the objective; all the variables appearing in the objective (thus, the *non-basic* variables) always have value 0.

We now see that the only variable in the objective (25) appearing with a positive coefficient is x_2. How much can we increase it without violating the new constraints? The first constraint (26) implies that $\frac{x_2}{16} \leq \frac{33}{4}$, i.e., that $x_2 \leq 132$; the second constraint (27) implies that $\frac{3x_2}{8} \leq \frac{3}{2}$, i.e., that $x_2 \leq 4$. Note that the third constraint (28) does not impose any restrictions on how large x_2 can be, for as long as it is positive; thus, we conclude that the largest possible value of x_2 which does not cause violation of any of the constraints is $x_2 = 4$, which corresponds to constraint (27). The value $x_2 = 4$ forces $x_3 = 0$; we now switch the roles of x_2 and x_3 (this operation of switching the roles of two variables is called *pivoting*) by solving (27) for x_2:

$$x_2 = 4 - \frac{8x_2}{3} - \frac{2x_5}{3} + \frac{x_6}{3}$$

and then using this to eliminate x_2 from the objective, obtaining

$$z = 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3}$$

as well as from the constraints, obtaining after simplification

$$x_1 = 8 + \frac{x_3}{6} + \frac{x_5}{6} - \frac{x_6}{3} \tag{29}$$

$$x_2 = 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \tag{30}$$

$$x_4 = 18 - \frac{x_3}{2} + \frac{x_5}{2} \tag{31}$$

which produces the new basic feasible solution $(\overset{x_1}{8}, \overset{x_2}{4}, \overset{x_3}{0}, \overset{x_4}{18}, \overset{x_5}{0}, \overset{x_6}{0})$ with the new value of the objective $z = 28$. Note that in the new objective all the variables appear with a negative coefficient; thus our procedure terminates, but did it find the maximum value of the objective? Maybe with a different choices of variables in pivoting we would have come up with another basic feasible solution which would have different basic variables, also with all non basic variables in the objective appearing with a negative coefficient, but for which the obtained value of the objective is larger?

This is not the case: just as in the case of the Ford Fulkerson algorithm for Max Flow, once the pivoting terminates, the solution must be optimal regardless of which particular variables where swapped in pivoting, because the pivoting terminates when the corresponding basic feasible solution of the program becomes equal to a basic feasible solution of the dual program. Since ***every*** feasible solution of the dual is larger than every feasible solution of the starting (or *primal*) program, we get that the SIMPLEX algorithm must return the optimal value after it terminates. We now explain this in more detail.

1 LP Duality

General setup

Comparing our initial program P with its dual P^*:

$$
\begin{aligned}
P: \quad & \text{maximize} && 3x_1 + x_2 + 2x_3, \\
& \text{subject to the constraints} && \\
& && x_1 + x_2 + 3x_3 \leq 30 \\
& && 2x_1 + 2x_3 + 5x_3 \leq 24 \\
& && 4x_1 + x_2 + 2x_3 \leq 36 \\
& && x_1, x_2, x_3 \geq 0;
\end{aligned}
$$

$$
\begin{aligned}
P^*: \quad & \text{minimize} && 30y_1 + 24y_2 + 36y_3, \\
& \text{subject to the constraints} && \\
& && y_1 + 2y_2 + 4y_3 \geq 3 \\
& && y_1 + 2y_2 + y_3 \geq 1 \\
& && 3y_1 + 5y_2 + 2y_3 \geq 2 \\
& && y_1, y_2, y_3 \geq 0.
\end{aligned}
$$

we see that the original, *primal* Linear Program P and its *dual* Linear Program are related as follows

$$
\begin{aligned}
P: \quad & \text{maximize} && z(x) = \sum_{j=1}^{n} c_j x_j, \\
& \text{subject to the constraints} && \sum_{j=1}^{n} a_{ij} x_j \leq b_i; \quad 1 \leq i \leq m \\
& && x_1, \ldots, x_n \geq 0;
\end{aligned}
$$

$$
\begin{aligned}
P^*: \quad & \text{minimize} && z^*(y) = \sum_{i=1}^{m} b_i y_i, \\
& \text{subject to the constraints} && \sum_{i=1}^{m} a_{ij} y_i \geq c_j; \quad 1 \leq j \leq n \\
& && y_1, \ldots, y_m \geq 0,
\end{aligned}
$$

or, in matrix form,

$$
\begin{aligned}
P: \quad & \text{maximize} \quad z(x) = c^{\mathsf{T}} x, \quad \text{subject to the constraints} \quad Ax \leq b \quad \text{and} \quad x \geq 0; \\
P^*: \quad & \text{minimize} \quad z^*(y) = b^{\mathsf{T}} y, \quad \text{subject to the constraints} \quad A^{\mathsf{T}} y \geq c \quad \text{and} \quad y \geq 0.
\end{aligned}
$$

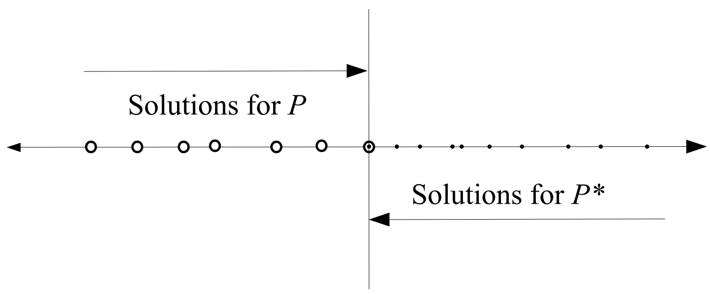
Weak Duality Theorem If $x = \langle x_1, \ldots, x_n \rangle$ is any basic feasible solution for P and $y = \langle y_1, \ldots, y_m \rangle$ is any basic feasible solution for P^*, then:

$$z(x) = \sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{n} b_i y_i = z^*(y)$$

Proof: Since x and y are basic feasible solutions for P and P^* respectively, we can use the constraint inequalities, first from P^* and then from P to obtain

$$z(x) = \sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i \right) x_j = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \right) y_i \leq \sum_{i=1}^{n} b_i y_i = z^*(y)$$

Thus, every feasible solution of P^* is an upper bound for the set of all feasible solutions of P, and every feasible solution of P is a lower bound for the set of feasible solutions for P^*; see the figure below.



Consequently, if we find a feasible solution of P which is also a feasible solution for P^*, such solution must be maximal feasible solution for P and minimal feasible solution for P^*.

We now show that when the *SIMPLEX* algorithms terminates that it produces a basic feasible solution $\overline{x}$ for P, and, implicitly, a basic feasible solution $\overline{y}$ for the dual P^* for which $z(\overline{x}) = z^*(\overline{y})$; by the above, this will imply that $z(\overline{x})$ is the maximal value for the objective of P and that $z^*(\overline{y})$ is the minimal value of the objective for P^*.

Assume that the *SIMPLEX* algorithm has terminated; let B be such that the basic variables (variables on the left hand side of the constraint equations) in the final form of P are variables x_i for which $i \in B$; let $N = \{1, 2, \ldots, n+m\} \setminus B$; then x_j for $j \in N$ are all the non-basic variables in the final form of P. Since the *SIMPLEX* algorithm has terminated, we have also obtained a set of coefficients $\overline{c}_j \leq 0$ for $j \in N$, as well as $\overline{v}$ such that the final form of the objective is

$$z(x) = \overline{v} + \sum_{j \in N} \overline{c}_j x_j$$

If we set all the final non-basic variables x_j, $j \in N$, to zero, we obtain a basic feasible solution $\overline{x}$ for which $z(\overline{x}) = \overline{v}$.

Let us define $\overline{c}_j = 0$ for all $j \in B$; then

$$\begin{aligned}
z(x) &= \sum_{j=1}^{n} c_j x_j \\
&= \overline{v} + \sum_{j=1}^{n+m} \overline{c}_j x_j \\
&= \overline{v} + \sum_{j=1}^{n} \overline{c}_j x_j + \sum_{i=n+1}^{n+m} \overline{c}_i x_i \\
&= \overline{v} + \sum_{j=1}^{n} \overline{c}_j x_j + \sum_{i=1}^{m} \overline{c}_{n+i} x_{n+i}
\end{aligned}$$

Since the variables x_{n+i}, $(1 \le i \le m)$, are the initial slack variables, they satisfy $x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j$; thus we get

$$\begin{aligned}
z(x) &= \sum_{j=1}^{n} c_j x_j \\
&= \overline{v} + \sum_{j=1}^{n} \overline{c}_j x_j + \sum_{i=1}^{m} \overline{c}_{n+i} \left(b_i - \sum_{j=1}^{n} a_{ij} x_j \right) \\
&= \overline{v} + \sum_{j=1}^{n} \overline{c}_j x_j + \sum_{i=1}^{m} \overline{c}_{n+i} b_i - \sum_{i=1}^{m} \sum_{j=1}^{n} \overline{c}_{n+i} a_{ij} x_j \\
&= \overline{v} + \sum_{j=1}^{n} \overline{c}_j x_j + \sum_{i=1}^{m} \overline{c}_{n+i} b_i - \sum_{j=1}^{n} \sum_{i=1}^{m} \overline{c}_{n+i} a_{ij} x_j \\
&= \overline{v} + \sum_{i=1}^{m} \overline{c}_{n+i} b_i + \sum_{j=1}^{n} \left(\overline{c}_j - \sum_{i=1}^{m} \overline{c}_{n+i} a_{ij} \right) x_j
\end{aligned}$$

The above equations hold true for all values of x; thus, comparing the first and the last equation we conclude that

$$\begin{aligned}
\overline{v} + \sum_{i=1}^{m} \overline{c}_{n+i} b_i &= 0; \\
\overline{c}_j - \sum_{i=1}^{m} \overline{c}_{n+i} a_{ij} &= c_j, \quad (1 \le j \le n).
\end{aligned}$$

i.e.,

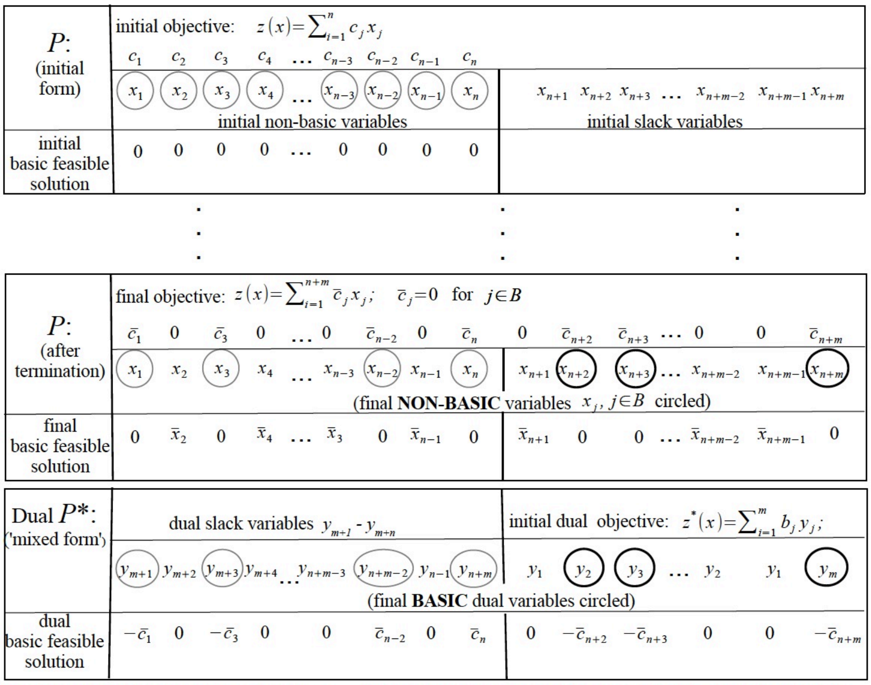
$$\begin{aligned}
\sum_{i=1}^{m} b_i(-\overline{c}_{n+i}) &= \overline{v}; \\
\sum_{i=1}^{m} a_{ij}(-\overline{c}_{n+i}) &= c_j + (-\overline{c}_j), \quad (1 \le j \le n).
\end{aligned}$$

We now see that if we set $\overline{y}_i = -\overline{c}_{n+i}$ for all $1 \le i \le m$, then, since the *SIMPLEX* terminates when all coefficients of the objective are either negative or zero, then such $\overline{y}$ satisfies:

$$\sum_{i=1}^{m} b_i \overline{y}_i = \overline{v};$$

$$\sum_{i=1}^{m} a_{ij} \overline{y}_i = c_j - \overline{c}_j \ge c_j, \quad (1 \le j \le n),$$

$$\overline{y}_i \ge 0, \quad (1 \le i \le m).$$



For all final non-basic variables of the form x_{n+j} (in the right box and circled) the corresponding basic dual variable y_j is set to $-\overline{c}_{n+j}$; for all final basic variables of the form x_{n+j} (in the right box and NOT circled) the value of y_j is set to 0.

Thus, such $\overline{y}$ is a basic feasible solution for the dual program P^* for which the dual objective has the same value $\overline{v}$ which the original, primal program P achieves for the basic feasible solution $\overline{x}$. Thus, by the Weak Duality Theorem, we conclude that $\overline{v}$ is the maximal feasible value of P and minimal feasible value for P^*.

Note also that the basic and non basic variables of the final primal form of the problem and of its dual are complementary: for every $1 \le i \le m$, variable x_{n+i} is basic for the final form of the primal if and only if y_i is non basic for the final form of the dual; (similarly, for every $1 \le j \le n$, variable x_j is basic for the final form of the primal if and only if y_{m+j} is not basic for the final form of the dual). Since the basic variables measure the slack of the corresponding basic feasible solution, we get that if $\overline{x}$ and $\overline{y}$ are the extremal feasible solutions for P and P^*, respectively, then for all $1 \le j \le n$ and all $1 \le i \le m$,

$$\begin{array}{lllll} \text{either} & \overline{x}_j = 0 & \text{or} & y_{m+j} = 0, & \text{i.e.,} \quad \sum_{i=0}^{m} a_{ij}\overline{y}_i = c_j; \\ \text{either} & \overline{y}_i = 0 & \text{or} & x_{n+i} = 0, & \text{i.e.,} \quad \sum_{j=0}^{n} a_{ij}\overline{x}_j = b_i. \end{array}$$

Note that any equivalent form of P which is obtained through a pivoting operation is uniquely determined by its corresponding set of basic variables. Assuming that we have n variables and m equations, then there are $\binom{n+m}{m}$ choices for the set of basic variables. Using the Stirling formula

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \quad \Rightarrow \quad \ln n! \approx \frac{\ln(2\pi n)}{2} + n\ln n - n = n\ln n - n + O(\ln n)$$

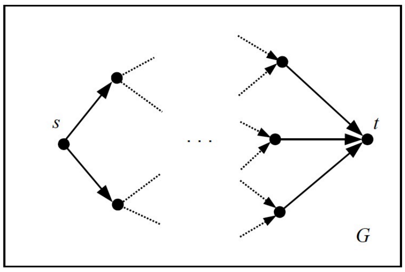
we get

$$\begin{aligned} \ln\binom{n+m}{m} &= \ln\frac{(m+n)!}{m!\,n!} = \ln(m+n)! - \ln m! - \ln n! \\ &= (m+n)\ln(m+n) - (m+n) - n\ln n - m\ln m + m + n \\ &= m(\ln(m+n) - \ln m) + n(\ln(m+n) - \ln n) \\ &\ge m + n \end{aligned}$$

Thus, the total number of choices for the set of the basic variables is $\binom{n+m}{m} > e^{m+n}$. This implies that the *SIMPLEX* algorithm could potentially run in exponential time, and in fact, one can construct examples of LP on which the run time of the *SIMPLEX* algorithm is exponential. However, in practice the *SIMPLEX* algorithm is extremely efficient, even for large problems with thousands of variables and constraints, and it tends to outperform algorithms for LP which do run in polynomial time (the Ellipsoid Method and the Interior Points Method).

1.1 Examples of dual programs: max flow

We would now like to formulate the Max Flow problem in a flow network as a Linear Program.

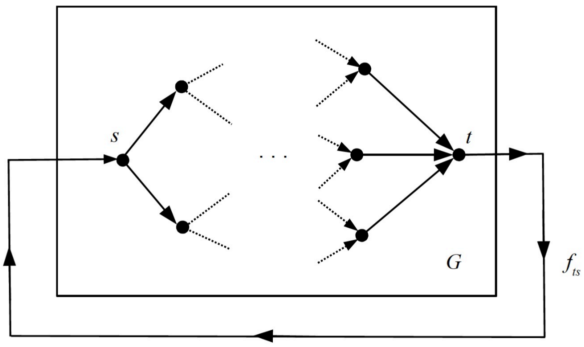


Thus, assume we are given a flow network G with capacities κ_{ij} of edges $(i,j) \in G$. The max flow problem seeks to maximise the total flow $\sum_{j:(s,j)\in E} f_{sj}$

through a flow network graph G, subject to the constraints:

$$C^* \begin{cases} f_{ij} & \leq \kappa_{ij};\ (i,j) \in G;\ \text{(flow smaller than pipe's capacity)} \\ \sum_{i:(i,j)\in G} f_{ij} & = \sum_{k:(j,k)\in G} f_{jk};\ j \in G - \{s,t\};\ \text{(incoming flow equals outgoing)} \\ f_{ij} & \geq 0;\ (i,j) \in G\ \text{(no negative flows).} \end{cases}$$

To eliminate the equality in the second constraint in C^* while introducing only one rather than two inequalities, we use a "trick": we make the flow circular, by connecting the sink t with the source s with a pipe of infinite capacity. Thus, we now have a new graph G', $G \subset G'$, with an additional edge $(t,s) \in G'$ with capacity ∞.



We can now formulate the Max Flow problem as a Linear Program by replacing the equality in the second constraint with a single but equivalent inequality:

P: *maximize:* f_{ts}
subject to the constraints:

$$\begin{aligned} f_{ij} &\leq \kappa_{ij}; & (i,j) &\in G; \\ \sum_{i:(i,j)\in G'} f_{ij} - \sum_{k:(j,k)\in G'} f_{jk} &\leq 0; & j &\in G; \\ f_{ij} &\geq 0; & (i,j) &\in G'. \end{aligned}$$

Thus, the coefficients c_{ij} of the objective of the primal P are zero for all variables f_{ij} except for f_{ts} which is equal to 1, i.e.,

$$z(f) = \sum_{ij} 0 \cdot f_{ij} + 1 \cdot f_{ts} \tag{1.1}$$

To obtain the dual of P we look for coefficients d_{ij}, $(i,j) \in G$ corresponding to the first set of constraints, and coefficients p_j, $j \in G$ corresponding to the second set of constraints to use as multipliers of the constraints:

$$f_{ij} d_{ij} \leq \kappa_{ij}\, d_{ij}; \qquad (i,j) \in G; \tag{1.2}$$

$$\sum_{i:(i,j)\in G'} f_{ij} p_j - \sum_{k:(j,k)\in G'} f_{jk} p_j \leq 0; \qquad j \in G. \tag{1.3}$$

note the two special cases of the second inequality involving f_{ts} are

$$\sum_{i\,:\,(i,t)\in G} f_{it}p_t - f_{ts}p_t \le 0;$$

$$f_{ts}p_s - \sum_{k\,:\,(s,k)\in G} f_{sk}p_s \le 0.$$

Summing all of inequalities from (1.2) and (1.3) and factoring out, we get

$$\sum_{(i,j)\in G} (d_{ij} - p_i + p_j)f_{ij} + (p_s - p_t)f_{ts} \le \sum_{(i,j)\in G} \kappa_{ij}d_{ij}$$

Thus, the dual objective is the right hand side of the above inequality, and, as before, the dual constraints are obtained by comparing the coefficients of the left hand side with the coefficients of the objective of P:

P^* : *minimize:* $\sum_{(i,j)\in G} \kappa_{ij}d_{ij}$

subject to the constraints:

$$\begin{array}{rll} d_{ij} - p_i + p_j & \ge 0 & (i,j)\in G \\ p_s - p_t & \ge 1 & \\ & & \\ d_{ij} & \ge 0 & (i,j)\in G \\ p_i & \ge 0 & i\in G \end{array}$$

Let us write the constraints of P using new slack variables ψ_{ij}, φ_j:

$$\begin{array}{ll} \psi_{ij} = \kappa_{ij} - f_{ij}; & (i,j)\in G; \\ \varphi_j = \displaystyle\sum_{k\,:\,(j,k)\in G'} f_{jk} - \sum_{i\,:\,(i,j)\in G'} f_{ij}; & j\in G; \\ f_{ij} \ge 0; & (i,j)\in G'; \\ \psi_{ij} \ge 0; & (i,j)\in G'; \\ \varphi_j \ge 0; & j\in G \end{array}$$

Note now an important feature of both P and P^*: all variables $f_{ij}, \psi_{ij}, \varphi_j$ in P (and also all variables d_{ij}, p_i in P^*) appear only with the coefficient ± 1. One can now see that, if we solve the above constraint equations for any subset of the set of all variables $\{f_{ij}, \psi_{ij}, \varphi_j \,:\, (i,j)\in G^*,\ j\in G\}$ as the set of the basic variables, all the coefficients c_{ij} in the new objective which multiply ψ_{ij} or φ_j and which are obtained after the corresponding substitutions removing the basic variables from the objective will have coefficients either 0 or 1.[1] This means that in the corresponding basic feasible solution of P' at which the minimal value of the dual program is obtained, also all values $\overline{d}_{ij}$ of d_{ij} and all values $\overline{p}_j$ of p_j will be either 0 or 1.

What is the interpretation of such solution of the dual Linear Program P^*? Let us consider the set A of all vertices j of G for which $\overline{p}_j = 1$ and the set B of

[1]This corresponds to the fact that such constraints define a polyhedra whose all vertices have coordinates which are all either 0 or 1.

all vertices j for which $\overline{p}_j = 0$. Then $A \cup B = G$ and $A \cap B = \emptyset$. The constraint $p_s - p_t \geq 1$ of P^* implies that $\overline{p}_s = 1$ and $\overline{p}_t = 0$, i.e., $s \in A$ and $t \in B$. Thus, A and B define a cut in the flow network. Since at points $\overline{p}, \overline{d}$ the objective $\sum_{(i,j) \in G} \kappa_{ij} d_{ij}$ achieves the minimal value, the constraint $d_{ij} - p_i + p_j \geq 0$ implies that $\overline{d}_{ij} = 1$ if and only if $p_i = 1$ and $p_j = 0$, i.e., if and only the edge (i, j) has crossed from set A into set B. Thus, the minimum value of the dual objective $\sum_{(i,j) \in G} \kappa_{ij} d_{ij}$ precisely corresponds to the capacity of the cut defined by A, B. Since such value is equal to the maximal value of the flow defined by the primal problem, we have obtained a maximal flow and minimal cut in G!

As we have mentioned, the extreme values of linear optimization problems are always obtained on the vertices of the corresponding constraint polyhedra. In *this particular case* all vertices are with $0, 1$ coordinates; however, in general this is *false*. For example, NP hard problems formulated as Linear Programming problems always result in polyhedra with non-integer vertices.

Theorem: Solving an Integer Linear Program (ILP), i.e., an LP with additional constraints that the values of all variables must be integers is NP hard, i.e., there cannot be a polynomial time algorithm for solving $ILPs$ (unless an extremely unlikely thing happens, namely that $P = NP$).